Clustering-Based Colour Image Segmentation

by
Rose H. Turi
B.Comp.(Hons)

School of Computer Science and Software Engineering
Monash University
Australia

Thesis Submitted for Examination
for the Degree of
Doctor of Philosophy

January 2001

Abstract

Many clustering techniques have been developed over the years for different types of data. In this thesis the focus is on colour image segmentation. Many of the existing clustering techniques, such as *K-means* and *fuzzy K-means*, require the number of clusters to be supplied as a parameter, and other methods, such as ISODATA, while allowing the number of clusters to be increased or decreased by incorporating splitting and merging procedures, rely on too many parameters in order to determine when you can split or merge, and when you can stop. In order to overcome these limitations, it becomes necessary to develop some criteria to measure the quality of a clustering so that a clustering program can automatically determine the number of clusters, and it should ideally rely on as few parameters as possible so that the method can produce results which are not biased by the choice of parameter values.

The intra-cluster distance and inter-cluster distance measures are examined, and criteria are developed based on these measures to incorporate into clustering methods to enable the number of clusters to be determined automatically. There was found to be a statistically significant correlation between the intra-cluster distance and inter-cluster distance measures, leading to the idea that either of these criteria can be used individually within a method. Two methods were developed which incorporate a merging process and which use criteria based on the inter-cluster distance measure to indicate when the merging process can stop. Three methods were developed which

incorporate a splitting process and use criteria based on the intra-cluster distance measure to indicate when the splitting process can occur.

The methods using the inter-cluster distance measure ensure that there is good cluster separation, while the methods using the intra-cluster distance measure ensure that the clusters are compact. Ideally, the clustering method should aim to achieve both of these goals. A validity measure has been proposed which is based on the ratio between the average intra-cluster distance and the minimum inter-cluster distance. Obviously, the aim would be to find the clustering which minimizes this measure, however, for small numbers of clusters, the minimum inter-cluster distance is very big, resulting in the clustering with only two or three clusters being chosen, which doesn't match with a good segmentation from a visual point of view. Consequently, a multiplier function is incorporated based on the Gaussian function which penalizes results with too few clusters in favour of results containing more clusters which are also more visually appealing. The performance of this measure was compared with existing measures, including the Davies-Bouldin and the modified Dunn's indexes, and the proposed measure proved to select an adequate number of clusters for all of the test images, whereas the other measures were only successful some of the time.

For each of the methods, two different cluster centre representations, mean and median, and two different distance measures, Euclidean and absolute, were used. This gave four possible combinations of cluster centre and distance type.

The context of the image was also incorporated into the clustering methods. The clustering takes place in the colour domain, without any regard about the actual location of the colours within the image. For this reason, a linear function of the neighbouring eight pixels can be used to estimate the current pixel value. Four ways in which we can do this have been proposed, namely, *mean*, *median*, *weighted*, in

which we consider the distance of the neighbouring pixels from the current pixel, and *regression*, for which a linear regression function is determined from the data in the entire image. Most of the results obtained are based on the RGB (red, green, blue) colour space. However, alternative colour spaces were also investigated, including XYZ, KL (Karhunen-Loève) and HSV (hue, saturation, value), for which findings indicate a similarity between the RGB and KL results, which produced superior results. The XYZ colour space produced reasonable results only for a few images and the HSV colour space results had a tendency to appear noisy.

A visual analysis survey was conducted with a group of ten people to determine what the correct number of clusters should be, which would provide a framework for analysing the results by the proposed methods. Additionally, for each of the methods developed, and for some existing ones, a second visual analysis survey was conducted with the same group of people to compare the results by the proposed methods to determine which method produces better overall results.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university or other institution and to the best of my knowledge, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Rose Turi,

School of Computer Science and Software Engineering,

Monash University, Australia

January 2001

Acknowledgements

First, I would like to thank my supervisor, Dr. Sid Ray, whose tireless guidance and support has been invaluable over the course of completing this thesis. Thanks are also due to Dr. Peter Tischer, whose advice and input much enriched this thesis.

I also express thanks to all the staff of the School of Computer Science and Software Engineering, who have provided an enjoyable working and research environment. I would also like to thank the people who participated in my visual analysis survey for their time.

Finally, I would like to thank my family for all their support during my studies, and especially to Helena Turi for her tireless effort in proof-reading my thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Image Segmentation

Image segmentation is the process of subdividing an image into its constituent parts and extracting these parts of interest, which are the objects. Many techniques are available for image segmentation [PP93, FM81, HS85, Nev86, RD79], especially for grey scale images. The basic idea for segmentation can be described as follows. Given a set of p-dimensional data $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subset \Re^p$ and a uniformity predicate P, we wish to obtain a partition of the data into disjoint nonempty groups $\{C_1, C_2, \ldots, C_K\}$ subject to the following conditions:

(1) $\bigcup_{i=1}^{K} C_i = \mathbf{X}$

(2) $C_i \bigcap C_j = \phi, \qquad i \neq j$

(3) $\mathrm{P}(C_i) = TRUE, \qquad i = 1, 2, \ldots, K$

(4) $\mathrm{P}(C_i \bigcup C_j) = FALSE, \qquad i \neq j$

The first condition ensures that every data value must be assigned to a group, while the second condition ensures that a data value can be assigned to only one group.

The third and fourth conditions imply that every data value in one group must satisfy the uniformity predicate while data values from two different groups must fail the uniformity criterion.

The simplest segmentation method is *thresholding* [GW92] where an image is divided into "object" and "background" depending on whether the pixel values are above or below a given threshold. However, more complex images may have more than one object and, in this instance, the image can be divided into more than two segments by choosing more than one threshold. This process is referred to as *multilevel thresholding*. Many methods have been proposed for thresholding [SSWC88, PG87, KSW85, PP89, Pun80, Ray93, WS89, PC99, RT94, TRT95].

Edge detection [PM82, GW92, KC97] is a method which aims to segment an image by finding the edges of each region by locating the points in the image where the intensity values change dramatically. These discontinuities are usually found by running a mask through the image. The most common form of mask is a 3×3 mask for which the procedure involves computing the sum of products of the coefficients in the mask with the intensity values contained in the region encompassed by the mask, and the value obtained corresponds to the pixel located at the centre of the mask. By using different values for the coefficients in the mask, different forms of edges could be sought [GW92]. It may also be necessary to perform some edge linking as the edges obtained by applying various masks to the image may not give complete boundaries. The use of masks to detect edges has a long history, and many methods have been developed for this task [PM82].

Region growing [GW92, FCE00] is a process which aims to group pixels into larger and larger regions. A set of seed points are chosen as a starting point, and the regions grow by appending to each seed point those neighbouring pixels that have

similar properties, such as colour or texture. Problems inherent in this approach are how to select appropriate seed points, and also how to select suitable properties which can be used to determine whether points can be included in various regions during the growing process. Another region-based method is *region splitting and merging* [GW92, HP78, HP74]. In this approach an image is initially subdivided into a set of disjointed regions and then they are merged and/or split until each region satisfies some condition indicating it is one segment. For example, for a square image it is possible to divide the image into four quadrants. If the condition is not satisfied for any quadrant, then that quadrant is further split into four smaller quadrants. It is also possible to merge neighbouring regions whose combined pixels satisfy the condition of connectedness. The process stops when no more splitting or merging is possible. The *region-based* approach to image segmentation has been applied by many researchers [BP87, OPR78, TB97]. More recently, there has been interest in combining a *region-based* approach with an *edge detection* scheme [TA97, ZL98, MB97].

1.2 Clustering-based Image Segmentation Methods

The other major category of segmentation techniques are *clustering-based* image segmentation methods, although clustering methods have had applications on many other kinds of data and many different approaches have been proposed [JD88, Jai86, JMF99, AHDS96, CA79, AM96, DK97, SS73, MMR96, Kun99, MMR96, PP97, Che95]. The goal of clustering methods is to locate clusters from intensity values which are similar.

Everitt [Eve74] defined a cluster in the following ways:

- "A cluster is a set of entities which are *alike*, and entities from different clusters are not *alike*."

- "A cluster is an aggregate of points in the test space such that the *distance* between any two points in the cluster is less than the distance between any point in the cluster and any point not in it."

- "Clusters may be described as continuous regions of a multi-dimensional space containing a relatively high density of points, separated from other such regions by regions containing a relatively low density of points."

Kowalewski [Kow95] proposed an algorithm aimed at achieving the third definition of a cluster given above, which is a method involving a gradient procedure for determining clusters having a relatively high density of points.

Based on the definitions of a cluster given above, it is clear that a method is required to determine how alike different intensity values are. With regard to the four rules describing the basic idea of segmentation mentioned in the previous section, when applied to clustering, the predicate used is usually some form of similarity or dissimilarity measure so that two data values from one cluster must be more similar to each other than two data values from two distinct clusters, for which the dissimilarity would be larger. This similarity/dissimilarity criterion usually takes the form of some sort of distance measure [JMF99].

Colour images are more complex than grey scale images as instead of a single intensity value for a pixel, each pixel is usually denoted by three component values, such as red, green and blue. Clustering based methods are ideal to use for segmentation of colour images as the methods used for grey scale images can be easily extended to cope with higher dimensionality, although, the increased dimensionality also leads

to a more computationally expensive process. Clustering algorithms can be either *hierarchical* [JD88, Eve74, AK88, BA98] or *partitional* [JD88, Eve74, TG74]. *Hierarchical* techniques involve the clusters themselves being classified into groups, where the process is repeated at different levels to form a tree. *Partitional* techniques form clusters by optimizing a clustering criterion, where the classes are mutually exclusive, thus forming a partition of the data.

Statistical clustering is the most common form of clustering, which involves assigning each occurrence of a particular colour to one particular cluster regardless of whether each pixel having the colour are located near each other spatially in the image, or are scattered throughout the image, in which case these pixels would not form a valid segment from the visual point of view. With regard to the four rules describing the basic idea of image segmentation mentioned in the previous section, where K is the number of groups into which the image is divided, for clustering, K represents the number of groups in the colour space in which the clustering takes place, rather than the number of regions in the image. In fact, when constructing the segmented image, the number of regions in the spatial domain will be greater than or equal to the number of clusters in the colour domain. This is due to the colours being assigned to the same cluster even though they may not be located next to each other from a spatial point of view. For *region-based* or *edge detection* methods, K will correspond to the number of regions in the image from a spatial point of view.

1.3 Motivation of the Research

Many clustering-based segmentation methods have been proposed and the most popular of these are the *K-means* [TG74], *fuzzy K-means* [TB86], *thresholding and fuzzy K-means* [LL90], ISODATA [TG74] and *snob* [WB68, WD94] methods. Most of the

clustering methods have a common failing, which is that the number of clusters usually must be supplied, or if splitting or merging is incorporated, the stopping criterion is usually based on too many parameters. A good clustering-based segmentation method should automatically determine the number of clusters present in the image, and it should rely on as few parameters as possible, preferably none, so that the final result is not biased by the choice of parameter values. In this thesis, clustering-based methods will be presented, of both the *hierarchical* and *partitional* kinds, many of which have their foundations in existing clustering methods. However, they also incorporate various criteria to enable the procedures to objectively determine when to stop and therefore, automatically detect the number of clusters present in the image. These new methods also have the advantage that they do not involve the use of too many parameters, with the most being two.

1.4 Contribution of the Thesis

The most important aspect of any clustering method is determining the number of clusters. In terms of colour image segmentation, visual analysis can be very subjective in evaluating a segmentation result as two different people could have two different opinions about what the correct number of clusters is. We therefore require a good clustering method to automatically determine the number of clusters objectively. Consequently, we require some criteria which would indicate when the correct number of clusters has been achieved.

In detail, the thesis contains the following contributions:

- The introduction of some criteria based on the inter-cluster distance measure, which measures the separation between clusters, where the bigger the separa-

tion, the better the segmentation. These measures are combined with clustering methods which employ a merging step to allow the number of clusters to be decreased, and the criteria indicate when the process can stop and when the correct number of clusters has been found.

- The introduction of some criteria based on the intra-cluster distance measure, which measures the compactness of a cluster, where the more compact the clusters are, the better the segmentation. These measures are combined with clustering methods which employ a splitting step to allow the number of clusters to be increased, and the criteria indicate when the process can stop and when the correct number of clusters has been found.

- The introduction of a validity measure based on the ratio between the average intra-cluster distance and the minimum inter-cluster distance. We use this measure to determine the optimum number of clusters by finding the segmentation which produces the minimum value for this ratio. The problem with this measure, and other similar existing measures, is that there is a tendency to select the clustering result with the smallest number of clusters, such as 2 or 3, which is usually small in terms of colour images. This is due to the inter-cluster distance being very large when only a few clusters are formed. Due to this fact, a multiplier function is incorporated, based on the Gaussian function, which penalizes results with too few clusters in favour of results containing more clusters which are also more visually appealing. The performance of this ratio is compared with other similar ratios such as the Davies-Bouldin index [DB79] and Dunn's indexes [Dun74, BP98], and it was found to correctly detect the number of clusters for a wider range of images than either of these existing measures.

- The use of context information. A particular pixel's value is highly correlated with its neighbouring pixels values. However, the clustering methods take place in the colour space domain, and don't take into account the position of the pixels in the image or the relationship between neighbouring pixels. Therefore, we must incorporate this information in some way. We can use a linear function of the eight neighbouring pixels to estimate the current pixel value. Four methods have been developed to accomplish this, which are
  - mean of the neighbours;
  - median of the neighbours;
  - weighted sum of the neighbours, where the pixels in the north, east, south and west directions are considered to be a distance of 1 away from the current pixel, and the pixels in the north-east, south-east, south-west and north-west directions are considered to be a distance of $\sqrt{2}$ away by Pythagoras' Theorem, and weights are assigned accordingly;
  - linear regression of the neighbours, coefficients of the regression function being estimated from the entire image.
- The investigation of the effect of different methods to calculate the cluster centres, which are mean and median, and also two different distance measures to determine the closeness of two colours, which are Euclidean and absolute. Results for each of the proposed methods are obtained using all four of the possible combinations of cluster centre and distance type in order to determine whether any of the four combinations lead to superior clustering results.
- The application of the clustering methods to a number of colour spaces including RGB (red, green, blue), HSV (hue, saturation, value), XYZ and KL (Karhunen-

Loève). Results for each of the proposed methods are obtained using each of these colour spaces in order to determine whether the use of the HSV, XYZ or KL colour spaces produce superior segmentation results in comparison to the RGB colour space.

1.5 Test Data

This research is based on the clustering of colour image data. Bearing this in mind, two sets of test images have been selected, *synthetic* images and *natural* images. The use of the *synthetic* images allows the methods to be tested for accuracy, as the ideal clustering for these images is known. The set of *synthetic* images have colours generated by a Gaussian distribution, and include images with two, four, six, eight, ten and fifteen clusters. The images were generated by a modified form of the method, described by Milligan, for generating artificial test clusters [Mil85], and the Gaussian values were generated from the method of transforming uniform values in the range [0,1] to standard normal deviates described by Rubinstein [Rub81], where these values were then modified to the required mean and standard deviation. The images with four, six and eight clusters have all the clusters with the same variability, however, the images with four and six clusters have different sized clusters and both have a cluster having only one colour. The images with two, ten and fifteen clusters have clusters with different variabilities with each segment having the same size. These *synthetic* images are shown in Figure 1.1. The corresponding histograms for each of these images, for each of the red, green and blue colour components, are shown in appendix A. The size of each of these images is 256×256. After each method is described, the results for the *synthetic* images are shown first. Most of the methods, which will be described later, rely on one or at most two parameter values to help

determine when the correct number of clusters has been reached. The results from the *synthetic* images will allow the ideal parameter values to be determined, which can then be applied to the *natural* images.

A total of eleven *natural* images were selected which represent a wide variety of colour images from the segmentation point of view. These images are shown in Figure 1.2. The corresponding histograms for each of these images, for each of the red, green and blue colour components, are shown in appendix A. Each of these images are 256 $\times$ 256 in size, except for *molecule* which has a size of 257 $\times$ 257 and *mouse* and *rose* have a size of 200 $\times$ 320. Once the analysis of the results for the *synthetic* images is complete, each of the methods are then applied to the *natural* images.

1.6 Thesis Outline

In chapter 2, a review of existing clustering methods is given, in which some of the existing clustering-based segmentation methods are described, stating the advantages and disadvantages of each.

Chapter 3 will discuss a few considerations which must be taken into account to perform clustering, such as what distance measure should be used to determine similarity, how the cluster centre should be represented, how should context be taken into account, what colour space should be used and most importantly, how do we determine the number of clusters. In determining the number of clusters, results from a visual analysis survey are presented which aims to provide a benchmark for the number of clusters for each of the natural colour images which can then be used to determine the accuracy of the proposed clustering methods.

In chapter 4, an investigation of the intra-cluster and inter-cluster distance measures for possible use for cluster validity will be given, and the relationship between

these measures is also established.

Chapter 5 presents two methods which make use of criteria based on the inter-cluster distance measure, to determine the number of clusters, and which incorporate a merging process to join similar clusters together. The first method only allows two clusters to be merged during one iteration, however, the second method allows more than two clusters to be merged during a given iteration.

Chapter 6 contains a description of clustering methods which make use of criteria, based on the intra-cluster distance measure, to determine the number of clusters, and which also incorporate a splitting process to allow the number of clusters to be increased. The cluster having maximum variability is the one which is always chosen to be split into two smaller clusters. Also, a comparison of the results obtained from the proposed methods in chapters 5 and 6 is given in the form of a visual analysis survey.

Chapter 7 introduces a validity measure based on the ratio between the intra-cluster distance and the inter-cluster distance which can be used to determine what the optimal number of clusters is. This measure also incorporates a multiplier function, based on the Gaussian function, to allow results with small numbers of clusters to be penalized in favour of results with more clusters which are also more visually appealing. The superiority of this method is based on the fact that both the intra-cluster and inter-cluster distances are considered to determine the validity of the results.

Finally, in chapter 8, the concluding observations and remarks will be given, indicating some possible further areas of research.

Chapter 2

Review of Clustering Techniques

Clustering algorithms have had many applications over the years, for example, numerical taxonomy [SS73, Fis36], vector or colour image quantization [KFN98, Xia97, BJLS98, PMJ96, Sch97], image segmentation [JD88, LL90, TB86, AK88, PYL98, PC96, SSN98, HPB98, HB97, BA98], and image retrieval [KMW96], but to name a few. As mentioned in the previous chapter, clustering techniques fall into two main categories, *hierarchical* and *partitional*. In this chapter an overview of these two types of clustering methods is given, as well as a detailed description and analysis of some of the more popular methods, of both the *hierarchical* and the *partitional* kinds, stating the advantages and disadvantages of each of these methods.

2.1 Hierarchical Clustering Techniques

Hierarchical techniques are based on the use of a proximity matrix indicating the similarity between every pair of data points to be clustered. The end result is a "*dendrogram* representing the nested grouping of patterns and similarity levels at which groupings change" [JMF99]. Many *hierarchical* methods have been proposed [JD88, Eve74, DH73, And73] of which some are *agglomerative* and some are

divisive. *Agglomerative hierarchical* methods begin with each of the entities being assigned to a unique cluster and then repeatedly finding the two clusters which are the most similar, which is indicated by the proximity matrix, and merging the corresponding two clusters. The proximity matrix is then updated to reflect the new clustering. A *divisive* approach can also be applied to *hierarchical* clustering in which the algorithm would begin by assigning all the data points to one cluster and splitting a cluster at each stage until the number of clusters is equal to the number of data points, so this is simply the reverse of *agglomerative hierarchical* clustering. Gowda and Ravi [GR95] proposed a *divisive hierarchical* algorithm for symbolic clustering.

Focusing on the *agglomerative* approach, the basic *hierarchical* method has the following steps:

1. Construct a similarity matrix showing the distance between each pair of entities.

2. Let $K = N$, where N is the number of entities.

3. Find the nearest pair of distinct clusters, say C_i and C_j.

4. Merge C_i and C_j, delete C_j, and decrement K by one.

5. If $K > 1$, go to step 3, otherwise stop.

The main difference between *hierarchical* methods is in the way they determine which pair of clusters are the most similar in step 3 above. The *single link* method [Sne57, Joh67, JD88, Eve74, And73] merges together the groups whose distance between their closest members is smallest. The *complete link* method [JMF99, JD88, Eve74, And73] is the exact opposite as this method defines the distance between groups as the distance between their most remote pair of entities and merges together the two groups for which this is the smallest. The *centroid* method [Kin67, Eve74, And73]

replaces groups by their centroid, assuming Euclidean space, so the distance between groups is simply the distance between their centroids. In step 3 above, this method would merge together the two clusters whose distance between their centroids is smallest. A disadvantage of this approach is that if a very small cluster is merged with a very large cluster, the characteristics of the small cluster may be lost. The *median* method [Gow67, Eve74] assumes the clusters are of equal size so the new group will always be between the two groups being merged. If the centroids of the groups are $\mathbf{z}_i$ and $\mathbf{z}_j$, then the distance of the centroid of a third group $\mathbf{z}_h$ from the group formed by the merging of $\mathbf{z}_i$ and $\mathbf{z}_j$ lies along the median of the triangle defined by $\mathbf{z}_i$, $\mathbf{z}_j$ and $\mathbf{z}_h$. The *group average* method [Eve74, LW66] defines the distance between groups as the average of the distances between all pairs of entities in the two groups. *Ward's* method [War63, Eve74, And73] measures the loss of information resulting from the grouping of individuals as the sum of squared deviations of every point from the mean of the cluster to which it belongs. The two clusters whose fusion results in the minimum increase in the error sum of squares are merged.

One of the drawbacks of *hierarchical* algorithms is the time complexity, which according to Kurita [Kur91], is $\mathrm{O}(N^2 \log N)$ for hierarchical agglomerative methods. The space complexity is also a problem, being $\mathrm{O}(N^2)$ due to the similarity matrix of size $N \times N$ needing to be stored. This can be problematic for very large data sets. Amadasun and King [AK88] proposed a segmentation method based on the *agglomerative* clustering of uniform neighbourhoods which is "computationally efficient, and requires minimal memory" [AK88]. They combined the *agglomerative hierarchical* clustering technique with a region-based approach, where they divide the image into non-overlapping blocks and determine which of the blocks are considered uniform. Mean feature vectors are then computed for each of the uniform blocks.

The next step is where the *hierarchical* clustering is applied. The two most similar mean feature vectors are determined and these two vectors are considered to belong to the same category or cluster. The two vectors are merged resulting in one fewer mean vectors. This is repeated until the number of mean feature vectors equals the specified number of categories. The main advantage of this approach, as compared with the standard *agglomerative hierarchical* technique, is that the number of mean feature vectors is much smaller than the number of pixels in the image, leading to a much more computationally efficient approach since there is much less pairs of mean feature vectors for which the similarity has to be calculated compared with the number of pairs of pixels. The disadvantages are that the number of categories or clusters must be pre-specified and the result can depend on the block size and the uniformity criterion used.

2.2 Partitional Clustering Techniques

The defining feature of a *hierarchical* clustering technique is that once a data point is assigned to a particular cluster, it cannot be moved. Therefore, if a data point is incorrectly assigned to a cluster at an early stage, there is no way to correct this mistake. This is where the *partitional* clustering techniques have an advantage over the *hierarchical* techniques, as *partitional* techniques allow a data point to be reassigned to a different cluster if it improves the clustering. Generally, *partitional* clustering techniques produce a partition of the data points which optimizes some criterion function. One of the disadvantages of some of these techniques is that the number of clusters must be supplied and they often depend on the choice of the initial partition. The form of clustering where a class label is assigned to each data value identifying its class is referred to by some authors as *hard* or *crisp* clustering [JMF99,

KK93]. In recent years *fuzzy* clustering approaches have been developed where a fractional degree of membership for each cluster is assigned to each data value. In this section a description of some of the *hard* and *fuzzy* approaches taken to clustering will be given, stating the advantages and disadvantages of each approach.

2.2.1 Hard Clustering

2.2.1.1 K-means

The *K-means* method [Mac67, For65, TG74, And73] is one of the simplest clustering techniques, for which the objective is to find the partition of the data which minimizes the squared-error, or the sum of the squared distances between all points and their respective cluster centres. This procedure consists of the following steps, as described by Tou and Gonzalez [TG74].

1. Choose K initial cluster centres $\mathbf{z}_1(1)$, $\mathbf{z}_2(1)$,..., $\mathbf{z}_K(1)$.

2. At the kth iterative step, distribute the samples $\{\mathbf{x}\}$ among the K clusters using the relation,

$$\mathbf{x} \in C_j(k) \quad \text{if} \quad \| \mathbf{x} - \mathbf{z}_j(k) \| < \| \mathbf{x} - \mathbf{z}_i(k) \| \tag{2.1}$$

   for all $i = 1, 2, \ldots, K$, $i \neq j$, where $C_j(k)$ denotes the set of samples whose cluster centre is $\mathbf{z}_j(k)$.

3. Compute the new cluster centres $\mathbf{z}_j(k+1)$, $j = 1, 2, \ldots, K$, such that the sum of the squared distances from all points in $C_j(k)$ to the new cluster centre is minimized. The measure which minimizes this is simply the sample mean of

$C_j(k)$. Therefore, the new cluster centre is given by

$$\mathbf{z}_j(k+1) = \frac{1}{N_j} \sum_{\mathbf{x} \in C_j(k)} \mathbf{x}, \quad j = 1, 2, \ldots, K \tag{2.2}$$

where N_j is the number of samples in $C_j(k)$.

4. If $\mathbf{z}_j(k+1) = \mathbf{z}_j(k)$ for $j = 1, 2, \ldots, K$, the algorithm has converged and the procedure is terminated. Otherwise go to step 2.

In the above algorithm, the cluster centres are only updated once all points have been allocated to their closest cluster centre, which was the approach taken by Forgy [For65]. An alternative would be to allow the cluster centre gaining a new point to be updated after the addition of the point [Mac67]. The former version is the one which will be used in chapter 4 and is the basis for the proposed method in chapter 6. The advantages of *K-means* are that it is a very simple method, and it is based on intuition about the nature of a cluster, which is that the within cluster error should be as small as possible. The disadvantage of this method is that the number of clusters must be supplied as a parameter, leading to the user having to decide what the best number of clusters for the image is. Also, different results can be achieved by selecting different initial clusters and there is no way of knowing if the best solution has been found.

Pappas [Pap92] proposed a generalization of the *K-means* clustering algorithm and applied this procedure on grey-scale images. This algorithm aims to separate the pixels in the image into clusters based not only on their intensity, but on their relative location as well. Pappas indicated two problems with the *K-means* algorithm, which are that no spatial constraints are used and it assumes that each cluster is characterized by a constant intensity. Therefore, the technique which Pappas developed is

a generalization of the *K-means* algorithm in that it is adaptive and includes spatial constraints by the use of a Gibbs random field model. The proposed algorithm is iterative and each region is characterized by a slowly varying intensity function. The *a´posteriori* probability density function is then defined, which includes two components, one constrains the region intensity to be close to the data and the other imposes spatial continuity. The basic procedure involves alternating between maximizing the *a´posteriori* probability density and estimating the intensity functions, which are initially constant in each region and equal to the *K-means* cluster centres. The intensities are updated by averaging over a sliding window whose size progressively decreases. Therefore, the algorithm starts with global estimates and slowly adapts to the local characteristics of each region. Chen *et al.* [CLP98] applied a similar adaptive *K-means* clustering approach on biomedical images. The main differences between this approach and that of Pappas [Pap92] is the introduction of iterative estimation of cluster variances in the process of optimization, the choice of the number of clusters corresponds to the actual number of clusters present in the given image as they are dealing with biomedical images and the number of clusters are usually known for particular regions of human anatomy, while Pappas did not select the number of clusters by an objective manner, and finally, the selection of the parameter of the Gibbs random field was chosen "such that the spatial constraint is strong enough to smooth out the noise while still preserving the structural details" [CLP98]. Knowledge-based morphological operations are then applied to the segmented regions to identify the desired regions according to the *a´priori* anatomical knowledge of the regions of interest. The adaptive *K-means* type of approach still suffers from the same major disadvantage as the standard *K-means* in that the number of clusters must be supplied rather than determined.

2.2.1.2 ISODATA

The ISODATA method [TG74, BH67, And73] requires the specification of six parameters by the user which are K, the number of cluster centres desired, θ_N, a parameter against which the number of samples in a cluster domain is compared, θ_s, the standard deviation parameter, θ_c, the lumping parameter, L, the maximum number of pairs of cluster centres which can be lumped, and I, the number of iterations allowed. As with *K-means*, this procedure is iterative where, during each iteration, the data is assigned to the present cluster to whose centre it is closest. This algorithm also allows small clusters to be discarded, clusters with large within cluster variation to be split into two smaller clusters, and also for two clusters whose centres are close to each other to be lumped into one larger cluster. The parameters supplied by the user determine whether any of these things are to be carried out. The advantage of the ISODATA method is that it allows the number of clusters to increase or decrease by splitting or merging clusters throughout the program based on certain criteria being met. This method therefore will determine the number of clusters. While this is a clear advantage over *K-means*, which is referred to by some people as *basic* ISODATA [DH73], it has the problem that a large number of parameters are required, and the result will ultimately depend on the values of these parameters that the user wishes to use, making the result very subjective.

2.2.1.3 DYNOC

Tou [Tou79] describes a split-and-merge technique, similar to the ISODATA algorithm [TG74], called DYNOC (DYNamic Optimal Cluster-seeking), where the objec-

tive is to find a global maximum of

$$\lambda(K) = \frac{\min D_{ij}}{\max D_i} \tag{2.3}$$

where K is the number of clusters, D_{ij} is the interset distance between clusters C_i and C_j defined as

$$\begin{aligned} D_{ij} = &\| \mathbf{z}_i - \mathbf{z}_j \|, \quad i = 1, 2, \ldots, K-1 \\ &\qquad\qquad\quad j = i+1, \ldots, K \end{aligned} \tag{2.4}$$

where $\mathbf{z}_i$ and $\mathbf{z}_j$ are the cluster centres of cluster C_i and C_j, respectively, and D_i is the intraset distance for cluster C_i defined as

$$D_i = \sqrt{2 \sum_{j=1}^{p} (\sigma_{ij})^2} \tag{2.5}$$

where p is the dimensionality of the data and σ_{ij} is the standard deviation of component j in cluster C_i defined as

$$\sigma_{ij} = \sqrt{\frac{1}{N_i} \sum_{\mathbf{x} \in C_i} (x_j - z_{ij})^2} \tag{2.6}$$

where N_i is the number of points in cluster C_i and $\mathbf{x}$ is the data point $(x_1, x_2, \ldots, x_p)^T$, where T denotes vector transpose. As with the ISODATA algorithm, this is mainly an iterative procedure which also allows large clusters to be split into two smaller clusters and for two small clusters with similar centroids to be merged together. This method also requires parameter values to help determine whether splitting is required, and while a small range of possible values is given for these parameters, the final result will still ultimately rely on the parameter values used.

2.2.1.4 Techniques for Colour Image/Vector Quantization

Kaukoranta *et al.* [KFN98] describe a split-and-merge technique which they applied to the generation of a codebook in vector quantization. The general procedure is described by the following steps:

1. Generate an initial codebook by any algorithm.
2. Iterate the following
   - Split phase: (repeat h times)
     2.1. Select a cluster to be split.
     2.2. Split the selected cluster.
   - Merge phase: (repeat h times)
     2.3. Select two clusters to be merged.
     2.4. Merge the selected clusters.

   Until no improvement achieved.

The split and merge phases can be done in any order. If the split phase is done first, this will increase the number of clusters from K to $K+h$, and after the merge phase the number of clusters will once again be K. If the merge phase is done first, the number of clusters will decrease from K to K-h and the split phase will result with K clusters. This algorithm would stop once no more improvement is observed. The improvement is measured by the distortion of the codebook, defined as

$$D = \frac{1}{N} \sum_{i=1}^{N} \mathrm{d}(\mathbf{x}_i, Q(\mathbf{x}_i)) \tag{2.7}$$

where $\mathbf{x}_i$ is the ith training vector, Q is a mapping that gives the nearest code vector in the codebook and d gives the squared Euclidean distance between two vectors.

After one merge phase, the total distortion is increased by ΔD_M and after one split phase the total distortion is decreased by ΔD_S. Therefore, an iteration improves the D-value if $\Delta D_S > \Delta D_M$. For this method, the number of clusters is required as a parameter, and while this is fine when it is applied to vector quantization for which the target number of clusters is known, for image segmentation it may not be suitable if the ideal number of clusters is not known, which is the case for most natural images.

Xiang [Xia97] describes a colour image quantization technique which is basically a clustering technique which aims to minimize the maximum inter-cluster distance, or in other words, the aim is to minimize the maximum distance between colour points in each cluster. Consequently, it might feel more appropriate to use the term intra-cluster instead of inter-cluster, however, Xiang takes the view that each colour point is a singleton cluster, so the procedure is indeed minimizing the maximum inter-cluster distance. The algorithm is as follows, where $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ is the set of colour points to be clustered, h_i is a designated point in cluster C_i (h_i is called the head of C_i), $\mathrm{d}(a, b)$ is the distance between point a and point b, and K is the number of clusters required:

$C_1 = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$;

$h_1 = \mathbf{x}_1$;

for ($l = 1$; $l < K$; l++) {

$d = \max\{\mathrm{d}(\mathbf{x}_i, h_j) \mid \mathbf{x}_i \in C_j \text{ and } 1 \leq i \leq N \text{ and } 1 \leq j \leq l\}$;

$\mathbf{x}$ = one of the points whose distance to the head of the cluster it belongs to is d;

move $\mathbf{x}$ to C_{l+1};

$h_{l+1} = \mathbf{x}$;

for each $\mathbf{x}' \in (C_1 \bigcup \cdots \bigcup C_l)$ {

let j be such that $\mathbf{x}' \in C_j$;

if $(\mathrm{d}(\mathbf{x}', h_j) \geq \mathrm{d}(\mathbf{x}', \mathbf{x}))$ move $\mathbf{x}'$ from C_j to C_{l+1};

}

}

Initially all points are assigned to one cluster and the selection of its head is arbitrary. Then an iterative procedure is undertaken until the desired number of clusters, K, is obtained. A new cluster is formed by selecting a point that is the farthest away from its corresponding cluster head and moving this point to the new cluster as its head. Every other colour which is closer to this new cluster head than to its own cluster head is moved to this new cluster. Clearly, this approach also suffers from the disadvantage that the number of clusters must be specified. While this may be suitable for colour image quantization, for colour image segmentation in general, it is not always possible to know the ideal number of clusters.

Baek *et al.* [BJLS98] proposed an alternative to the *K-means* algorithm which they applied to vector quantization. The main aim of this method was to obtain a procedure which outperformed the *K-means* algorithm in terms of computational time. Generally, the more iterations of the *K-means* algorithm that have passed, the more training vectors, or data points, become stable and fixed to certain stable cluster memberships. In this procedure, the encoding step is modified so that "training vectors are encoded only when the distortion between them and their newly updated codeword is larger than their previous distortion. The other training vectors are left unencoded and keep their previous memberships" [BJLS98]. The algorithm is as follows, where $S(y)$ represents the set of training vectors that have the same codeword

y in the codebook, the average distortion between training vectors and their closest codewords is given by

$$D = \frac{1}{N} \sum_{x \in \mathbf{X}} \mathrm{d}(x, \hat{x}) \tag{2.8}$$

where $\mathbf{X}$ is the training set, N is the number of items in the training set and $\hat{x}$ is the closest codeword of the training vector x, B_i represents the codebook in the ith iteration, and ϵ is the distortion threshold to decide when to terminate the iterative process:

1. An initial codebook B_0 is given. Set $i = 0$ and $D_{-1} = \infty$.

2. For each training vector x, find its closest codeword in the codebook B_0.

3. Update B_{i+1} with $y^{i+1} = \mathrm{centroid}(\mathrm{S}(y^i))$.

4. Compute the average distortion D_i. If $(D_{i-1} - D_i)/D_i \leq \epsilon$ then halt. Otherwise, set $i = i + 1$ and go to step 5.

5. For each training vector x calculate $\mathrm{d}(x, y^i)$ and compare it to $\mathrm{d}(x, y^{i-1})$, where y^{i-1} is a codeword of x in the previous iteration and y^i is an updated codeword from y^{i-1}. If $(\mathrm{d}(x, y^i) < \mathrm{d}(x, y^{i-1})$, y^i is regarded as the codeword of x. Otherwise, find its closest codeword in the codebook B_i and go to step 3.

2.2.1.5 Snob

The Minimum Message Length (MML) information criterion, also known as the Wallace Information Measure [WB68, BW73, OH94], provides a means of carrying out model selection. Given a set of observations and a set of possible models which can be used to explain those observations, MML can be used to provide an objective criterion for comparing how well each model describes the observations. To do this,

a two part message is constructed, where the first part is a description of the model, and the second part is a description of the data in terms of the model. MML selects the model which leads to the overall shortest description. The descriptions can be encoded as a binary string, and the length of this string provides the measure of how well the model describes the observations. The advantage of this measure is that it can be used to objectively compare models of greatly differing complexity.

Clustering is one of the first areas where MML was applied, and an effective program which applies the MML principle to the problem of clustering is a program called *snob* [WD94, Wal84, BW70]. The following description gives the basic idea of this technique, as described by Patrick [Pat91]:

> Informally the act of classification can be described as allocating objects into classes such that each class is characterized by approximately like values for attribute measurements of the member objects. Snob uses various methods of intelligently assigning objects to classes. After each assignment the Wallace Information Measure is calculated. If the WIM has diminished in making the assignment then it is retained, otherwise the assignment is rejected. Snob has the ability to split and combine classes as well as move individuals from one class to another. Hence on the criterion of the WIM, which assigns a cost to creating a new class, Snob is able to reliably determine within a wide but not comprehensive search space, the following:
>
> a. the optimum number of classes for subdividing the objects;
>
> b. identify the class membership of each object;
>
> c. estimate the parameters of each class;

d. identify the statistical significance of each attribute in its contribution to a classification.

Snob creates a message for encoding the model and the observations given knowledge of the model. The message has the following form:

1. The number of classes.
2. A segment for each class describing
    a. the relative abundance of the class (i.e. the fraction of the observations that belong to the class).
    b. the distribution of attribute values within a class stating for each continuous attribute, its mean and standard deviation, or for each discrete attribute the relative probability of each value that attribute may take.
3. A segment for the description of each object stating
    a. the class to which the object belongs.
    b. the deviation of the object's attribute values from the class distribution.

Kanungo *et al.* [KDNS96] describe a technique based on a similar principle as MML, which is the Minimum Description Length (MDL) principle. They applied this technique on multi-band image segmentation.

2.2.1.6 Mixture Resolving Techniques

One approach to clustering is based on mixture resolving where we assume that the items to be clustered are taken from one of several distributions, and the aim is to determine the parameters of each distribution. For example, if we assume that the individual components of the mixture density are Gaussian, then the aim would be to

estimate the parameters of the individual Gaussian distributions. The Expectation Maximization (EM) algorithm [DLR77, MK97] has been applied to the problem of parameter estimation, where the parameters of the component densities, as well as the mixing parameters, are estimated from the patterns. The basic procedure involves obtaining an initial estimate of the parameters, in the form of a vector, which is followed by an iterative process consisting of two steps, the expectation step, or E-step, and the maximization step, the M-step. During these steps, the patterns are rescored against the mixture density produced by the parameter vector. These rescored patterns can then be used to produce new estimates of the parameters. This type of approach has links to Bayesian clustering [Ber94].

The choice of the initial parameters can greatly affect the result from the EM algorithm. Consequently, Fwu and Djurić [FD97] combined the EM algorithm with a tree structure scheme [PFM93] to obtain the initial estimates, with the objective of applying the process to image segmentation. The basic procedure involves a series of binary searches. Initially, all pixels are considered to belong to one class, and at each iteration, the number of classes is increased by one, and the results obtained from the previous stage are used to construct initial estimates for the current stage. The algorithm stops once the predefined number of classes have been reached.

Parameter estimation and mixture resolving has become a popular technique for image segmentation. Caillol *et al.* [CPH97] proposed a method for estimating fuzzy Gaussian mixtures which was based on an adaption of the iterative conditional estimation (ICE) algorithm [BPM93, Pie92] to the fuzzy framework.

Delignon *et al.* [DMP97] introduced the notion of a generalized mixture, being that the exact nature of components of the distribution mixture is unknown, but each belongs to a finite known set of families of distributions. The added difficulty

with estimating such a mixture is that the components have to be labelled. They show that mixture estimation algorithms, such as EM and ICE, can be adapted to such situations.

2.2.1.7 Other Techniques

Due to colour image clustering being a computationally expensive process, Celenk [Cel90] developed a method involving projecting the colour components into one dimension. Basically the idea is "to project the colour space onto the selected coordinate axes repeatedly until the image clusters are enclosed in some specified volume elements. From each projection of the space, we determine one particular surface of the decision elements. This estimates the clusters' distributions in the 3D colour space without imposing any constraints on their forms" [Cel90]. This method was applied on the CIE (L^*, a^*, b^*) uniform colour coordinate system.

Another popular clustering technique is the Watershed algorithm [SPK97, SPK98, NS96, VS91]. The following description gives the basic idea of this technique, as described by Shafarenko *et al.* [SPK98]:

> The idea of watershed is drawn from a topographic analogy. Consider a 2-D histogram of features as a topographic relief. Find all local minima and "pierce" them. Immerse the whole relief into water. As the relief goes deeper into the water, the regions surrounding the seeds become flooded. Eventually two or more such regions expand to a point at which they would come into contact unless the waters are separated. This is the moment that a dam is raised. In the watershed method, the dams are all infinitely tall and are arbitrary complex sets of pixels depending on the line of contact. This is a very informal definition though, since in the

situation of discrete altitude of the relief, which is laid out on a discrete grid, there is no way of gradually bringing the flooding water up to the point of contact. However, it helps to visualize the procedure.

The watershed algorithm can be used to partition an image into areas associated with each seed, where each area is nonintersecting and the boundaries are given by the watershed lines, which are where the dams would be, and the outer border of the image. However, it is important to note that the number of areas must be equal to the number of seed points chosen, so, as with the *K-means* method, where the number of clusters must be supplied, care must also be taken when selecting the number of seed points, and their values, for the watershed technique.

2.2.2 Fuzzy Clustering

Fuzzy clustering has become quite popular more recently and has had many applications [BB99a, BB99b, TB86, LL90]. For *fuzzy* clustering, fuzzy membership values are needed, which are based on the fuzzy set theory developed by Zadeh [Zad65]. The membership value for the jth value in cluster C_i, denoted by u_{ij}, must satisfy the following conditions [BB99a]:

(1) $u_{ij} \in [0,1], \qquad i = 1, \ldots, K, \quad j = 1, \ldots, N$

(2) $\sum_{i=1}^{K} u_{ij} = 1, \qquad j = 1, \ldots, N$

(3) $0 < \sum_{j=1}^{N} u_{ij} < N, \qquad i = 1, \ldots, K$

Because of condition (2), u_{ij} "values are relative numbers dependent on the absolute membership of the pattern in all other classes, thus indirectly on the total number of classes" [BB99a]. The *hard* clustering where each data value is only assigned to one cluster is a subset or a special case of the above rules. In this case, for a given data

value $\mathbf{x}_j$, u_{ij} will have the value of 1 for only one value of i and 0 for all the rest, thus satisfying each of the three rules above. Possibilistic clustering techniques are similar to fuzzy techniques in that they satisfy conditions (1) and (3), however, they relax condition (2) by not requiring that the membership values sum to 1 [BB99a, BG99]. In this case, according to Baraldi and Blonda [BB99a], u_{ij} "is an absolute similarity value depending on fuzzy state C_i exclusively, given input pattern $\mathbf{x}_j$. In other words," u_{ij} "is context-insensitive, since it is not affected by any other state."

2.2.2.1 Fuzzy K-means

The *fuzzy K-means*[1] algorithm [TB86, LL90] attempts to search for local minima of a generalized within group sum of squared error objective function defined as

$$J_m(u, \mathbf{v}) = \sum_{j=1}^{N} \sum_{i=1}^{K} (u_{ij})^m \| \mathbf{x}_j - \mathbf{v}_i \|_A^2, \qquad 1 \leq m < \infty \tag{2.9}$$

where $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ is a finite data set in $\Re^p$; $\mathbf{x}_j \in \Re^p$, $1 \leq j \leq N$ is a p-dimensional measurement vector; $\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K\}$ is a set of class centres, $\mathbf{v}_i \in \Re^p$, $1 \leq i \leq K$, represents the p-dimensional ith cluster centre; and $m \in [1, \infty)$ is the membership weighting exponent. For m greater than one, under the assumption that $\mathbf{x}_j \neq \mathbf{v}_i$, a local minimum can be found from

$$u_{ij} = \left(\sum_{l=1}^{K} \left(\frac{\| \mathbf{x}_j - \mathbf{v}_i \|}{\| \mathbf{x}_j - \mathbf{v}_l \|} \right)^{2/(m-1)} \right)^{-1}, \qquad \forall i, \forall j \tag{2.10}$$

and

$$\mathbf{v}_i = \frac{\sum_{j=1}^{N} (u_{ij})^m \mathbf{x}_j}{\sum_{j=1}^{N} (u_{ij})^m}, \qquad \forall i \tag{2.11}$$

[1]This method is often referred to as *fuzzy c-means* in the literature, where c represents the number of clusters.

The algorithm is iterative where during each iteration new cluster centres are calculated using equation 2.11 and then the u_{ij} values are updated using equation 2.10. This is repeated until the cluster centres stabilize. The advantage of the *fuzzy K-means* method is that it allows a value to belong partially to a number of clusters simultaneously. However, as with the *K-means* algorithm, the number of clusters must be supplied as a parameter.

According to Bensaid *et al.* [BHBC96], clustering algorithms suffer from two main problems, the first being the choice of the number of clusters and the second being the assignment of physical labels to the classes at termination. Techniques such as *fuzzy K-means* and also the hard *K-means* algorithm, which aim to minimize the squared error, suffer from an additional problem of tending to produce solutions which equalize cluster populations. In order to overcome these problems Bensaid *et al.* [BHBC96] proposed a semi-supervised *fuzzy K-means* algorithm in which some high quality labelled data from each class are introduced, leaving a majority of the data in unlabelled form. Equation 2.11 is updated to reflect the two sets of data, labelled and unlabelled, and in order to alleviate the problem of equalized cluster populations, a further modification is made by weighting the few samples that are labelled more heavily than their unlabelled counterparts. Equation 2.11 is modified as follows

$$\mathbf{v}_i = \left(\frac{\sum_{j=1}^{N_l} w_j (u_{ij}^l)^m \mathbf{x}_j^l + \sum_{j=1}^{N_u} (u_{ij}^u)^m \mathbf{x}_j^u}{\sum_{j=1}^{N_l} w_j (u_{ij}^l)^m + \sum_{j=1}^{N_u} (u_{ij}^u)^m} \right), \qquad \forall i \tag{2.12}$$

where the subscript and superscript l represents the labelled data and the subscript and superscript u represents the unlabelled data.

Liew *et al.* [LLL00] proposed a spatial *fuzzy K-means* clustering algorithm for image data which makes use of both the feature space information and the spatial contextual information. They introduce a weighting function which is adaptive as it

allows the influence of the neighbouring pixels to be suppressed in nonhomogeneous regions in the image. This procedure also makes use of a merging process to determine the optimal number of clusters. Merging is possible due to the incorporation of the spatial contextual information as this results in the membership values being smoothed locally leading to the centroids of two clusters which are close in the feature space and which share the same spatial context to move toward each other.

Wu *et al.* [WYC94] also proposed a fuzzy clustering technique, with the objective of segmenting colour map images, which was based on the *fuzzy K-means* algorithm and included supervised learning. The *fuzzy K-means* algorithm is first applied, giving a set of prototypes which must satisfy a set of validation criteria. These prototypes satisfying the validation criteria are then optimized using a neural network with supervised learning. The image can then be segmented using the optimized prototypes according to the nearest neighbour rule.

2.2.2.2 Thresholding and fuzzy K-means

The *thresholding and fuzzy K-means* method [LL90] is a two stage process involving a *coarse* segmentation followed by a *fine* segmentation. The *coarse* segmentation involves smoothing the histogram of each of the colour components by the Gaussian convolution

$$\begin{aligned} F(x,\tau) &= f(x) * g(x,\tau) \\ &= \int_{-\infty}^{\infty} f(u) \frac{1}{(2\pi)^{1/2}\tau} exp\left[-\frac{(x-u)^2}{2\tau^2}\right] \mathrm{d}u \end{aligned} \tag{2.13}$$

where "$*$" implies a 1-D convolution. We then use the first and second derivatives of the smoothed histograms to find the valleys which will then be the thresholds. A safe-area surrounding the thresholds is then determined, and every pixel not falling into any safe-areas is assigned to a cluster based on its red, green and blue values, and

cluster centres are calculated. The *fine* segmentation involves assigning each pixel which belongs to a safe-area to its closest cluster by calculating fuzzy membership functions, given by equation 2.10 above. The advantages of the *thresholding and fuzzy K-means* method is that the number of clusters is determined by the algorithm based on the number of thresholds for different colour components, and an extra threshold used to decide if a cluster is large enough to be considered valid. However, this number of clusters determined by the program is dependent on the parameter in the smoothing function and the size of the safe margin used to determine the pixels on which the fine segmentation is carried out.

2.2.2.3 Possibilistic K-means

Krishnapuram and Keller [KK93, KK96] proposed a possibilistic clustering algorithm in which the membership values for a given feature point across all clusters was not constrained to add to one. The membership function they use is

$$u_{ij} = \frac{1}{1 + (\frac{d_{ij}^2}{\eta_i})^{\frac{1}{m-1}}} \tag{2.14}$$

where d_{ij}^2 is the squared distance of feature point $\mathbf{x}_j$ to the cluster centre of cluster C_i and η_i are suitable positive numbers defined as

$$\eta_i = A \frac{\sum_{j=1}^{N} u_{ij}^m d_{ij}^2}{\sum_{j=1}^{N} u_{ij}^m} \tag{2.15}$$

where A is typically set to 1. The cluster centres, or prototypes, are updated using equation 2.11 from the *fuzzy K-means* algorithm. While this algorithm performs well in a noisy environment, it suffers from the same disadvantage as the *fuzzy K-means* algorithm in that the number of clusters must be specified. Additionally, Barni *et*

al. [BCM96] indicate three major drawbacks of this approach including the tendency to produce coincident clusters, a bad set of initial clusters leading to bad results and the high likelihood of convergence to only a local minimum.

Barni and Gualtieri [BG99] make use of the above equations in a possibilistic clustering algorithm which they use for line detection in real world imagery. They incorporate cluster validity by defining measures of fuzzy length and fuzzy thickness for each cluster and also allow for splitting, when the edge points are from different lines, and merging, when lines detected at different times during the execution of the algorithm are clearly part of the same line.

2.2.2.4 Other fuzzy clustering techniques

Chi and Yan [CY95] developed a method of deriving fuzzy rules from *K-means* clusters based on three measures, difference intensity, local standard deviation and a measure of the local contrast of a darker pixel against its background. They use these fuzzy rules to determine whether pixels from a map image either belong to the foreground, which includes characters, symbols and line patterns that consist of roads, streets and boundaries, or the background.

Frigui and Krishnapuram [FK97, FK99] proposed a method, *competitive agglomeration*, based on fuzzy clustering which includes an objective function that incorporates the advantages of both *hierarchical* and *partitional* clustering, with the aim being to minimize this function. Initially the number of clusters are over specified and then they are gradually decreased until the final result is obtained which has the "optimal" number of clusters. One advantage of this method is that by incorporating different distance measures, it allows clusters of various shapes to be found.

2.3 Conclusion

Based on all of the techniques discussed in this chapter, it is clear that clustering is a very popular approach to finding natural groupings in data, and it has many applications. The large amount of methods is an indication that the performance of a clustering technique is dependent on the particular data on which it is applied. Many of the techniques, such as *K-means*, *fuzzy K-means* and *possibilistic K-means*, require the number of clusters to be supplied, and while this may be suitable for some applications such as vector quantization in which the desired number of clusters is known, for colour image segmentation in general, this is a major disadvantage. Techniques such as ISODATA or DYNOC incorporate split and merge processes which allow the number of clusters to be modified during the execution of the algorithm with the aim to determine the optimal number of clusters, however, they tend to rely on the specification of too many parameters and ultimately are very subjective because the results will depend on the choice of the parameter values. Therefore, the ideal clustering algorithm is one which would objectively determine the optimal number of clusters and not rely on too many parameters to bias the results.

Chapter 3

Considerations for Clustering

When applying clustering methods to any types of data, many aspects need to be considered. In this chapter an overview of the considerations taken into account in this thesis will be given. The items covered in this chapter include the choice of distance measure to calculate similarity and the representation of the cluster centre. Considerations which are necessary specifically for colour image data are the incorporation of the context of the image, due to pixels being highly correlated with their neighbouring pixels, and the choice of colour space in which to perform the clustering. Finally, an overview of methods to determine the number of clusters is given, most of these methods being related to the concept of cluster validity.

3.1 Distance Measures to Calculate Similarity

One important aspect of clustering is the determination of the similarity of two data points, where in our case the data points are 3-tuple values representing colours. In order to perform a clustering, some form of proximity index is required to determine the similarity of data points $\mathbf{x}_i$ and $\mathbf{x}_j$, which is denoted by $\mathrm{d}(i,j)$. Generally, a proximity index must satisfy the following three properties [JD88]:

1. (a) For a dissimilarity: $\mathrm{d}(i,i) = 0, \ \forall i$

   (b) For a similarity: $\mathrm{d}(i,i) \geq \underbrace{\max}_{j} \mathrm{d}(i,j), \ \forall i$

2. $\mathrm{d}(i,j) = \mathrm{d}(j,i), \ \forall i,j$

3. $\mathrm{d}(i,j) \geq 0, \ \forall i,j$

One of the most common proximity indices is the Minkowski metric which is a dissimilarity measure [JD88]. Let the ith data point be defined as

$$\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{ip})^T, \qquad i = 1, 2, \ldots, N \tag{3.1}$$

where the data are considered p-dimensional and N is the number of data points. Then the Minkowski metric to determine the dissimilarity of data points $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined by

$$\mathrm{d}(i,j) = \left(\sum_{l=1}^{p} \mid x_{il} - x_{jl} \mid^q \right)^{1/q}, \qquad \text{where } q \geq 1 \tag{3.2}$$

In addition to satisfying properties 1-3 given above, the Minkowski metrics also satisfy the following two properties [JD88]:

4. $\mathrm{d}(i,j) = 0$ only if $\mathbf{x}_i = \mathbf{x}_j$

5. $\mathrm{d}(i,j) \leq \mathrm{d}(i,m) + \mathrm{d}(m,j), \ \forall i,j,m$

Property 5 is known as the *triangle inequality*.

The most common Minkowski metric is the Euclidean distance, where $q = 2$, which is given by

$$\mathrm{d}(i,j) = \sqrt{\sum_{l=1}^{p} (x_{il} - x_{jl})^2} = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)} \tag{3.3}$$

Another commonly used Minkowski metric is the absolute distance, sometimes referred to as Manhattan, taxicab or city block distance, where $q = 1$, which is given

by

$$\mathrm{d}(i,j) = \sum_{l=1}^{p} \mid x_{il} - x_{jl} \mid \tag{3.4}$$

The Minkowski metric as $q \to \infty$ is known as the "sup" distance which is given by

$$\mathrm{d}(i,j) = \underbrace{\max}_{1 \leq l \leq p} \mid x_{il} - x_{jl} \mid \tag{3.5}$$

Not all proximity indices are metrics, and thus do not always satisfy the properties of symmetry or transitivity. An example of such an index is the squared Mahalanobis distance, which is defined as

$$\mathrm{d}(i,j) = (\mathbf{x}_i - \mathbf{x}_j)^T \Sigma^{-1} (\mathbf{x}_i - \mathbf{x}_j) \tag{3.6}$$

where the matrix Σ is the sample covariance matrix. If Σ is the identity matrix, the squared Mahalanobis distance is the same as the squared Euclidean distance.

In this thesis, the performance of the Euclidean and absolute distances are compared in terms of the segmentation results obtained when these distances are used as part of the clustering algorithms presented in later chapters.

3.2 Cluster Centre Representation

The obvious way to represent the cluster centre is by the mean value of all the points in the cluster. Therefore, the cluster centre of cluster C_i, denoted by $\mathbf{z}_i = (z_{i1}, z_{i2}, \ldots, z_{ip})^T$, can be calculated as

$$z_{ij} = \frac{1}{N_i} \sum_{\mathbf{x} \in C_i} x_j, \quad j = 1, 2, \ldots, p \tag{3.7}$$

where N_i is the number of points in cluster C_i, and $\mathbf{x} = (x_1, x_2, \ldots, x_p)^T$ represents the data point, in our case, a three-tuple ($p = 3$) colour value. Another way we could represent the cluster centre is by the median value of the points in the cluster. Consider, for example, the following values

$$18, 19, 19, 20, 20, 20, 20, 21, 21, 22, 196$$

The mean value of these is 36, however, the median value is 20 which is a better representation of the values. From this example, we can see that the mean value can be greatly affected by noise in the data, where just one value is very much different from the rest. The median value does not suffer from this limitation.

A natural grouping of cluster centre type and distance metric is mean with Euclidean and median with absolute. However, in this thesis, all four combinations of cluster centre type and distance metric, including mean and Euclidean, mean and absolute, median and Euclidean and median and absolute, are investigated to determine whether any particular combination provides superior clustering results for colour image segmentation.

3.3 Incorporating the Context of the Image

Clustering, as applied to image segmentation, has an additional concern not suffered by other types of data in that a given pixel's value is expected to be highly correlated with its neighbouring pixels values, and better results may be obtainable by incorporating this information somehow. Therefore, we would like to incorporate the context of the image into the clustering algorithm. We can estimate a pixel value as some linear combination of its eight neighbouring pixel values. Figure 3.1 shows the

representation of the 3 × 3 context, which could represent any of the red, green or blue components.

Y_1	Y_2	Y_3
Y_4	CP	Y_5
Y_6	Y_7	Y_8

CP is the current pixel

$Y_1,\ldots,Y_8$ are the eight neighbouring pixels

Figure 3.1: 3 ×3 context for an image.

The estimate of the pixel CP, say $\hat{\text{CP}}$, can be calculated as the mean of the eight neighbours, which would be given by

$$\hat{\text{CP}} = \frac{1}{8}\sum_{i=1}^{8} Y_i \tag{3.8}$$

It is also possible to represent $\hat{\text{CP}}$ by the median of the eight neighbours, which is calculated by ordering the Y_i's in numerical order and taking the average of the two middle-most values, since there is an even number of neighbours. We could also calculate $\hat{\text{CP}}$ as a weighted sum of the eight neighbours by taking into account the distance of each of the neighbours from the current pixel. In Figure 3.1, the pixels Y_2, Y_4, Y_5 and Y_7 can be considered as being a distance of 1 away from CP, while Y_1, Y_3, Y_6 and Y_8 are a distance of $\sqrt{2}$ away from CP by Pythagoras' Theorem. We then give the weights for each of the neighbours as the reciprocal of the distance from CP and, to determine $\hat{\text{CP}}$, we use the equation

$$\hat{\text{CP}} = \frac{\sum_{i=1}^{8} w_i \times Y_i}{\sum_{i=1}^{8} w_i} \tag{3.9}$$

where w_i is the ith weight value. The final method by which to calculate the value

of $\hat{\text{CP}}$ is to perform a linear regression over the entire image, which would be to find the constants, $a_0,\ldots,a_8$, which best fit the equation

$$\text{CP} = a_0+a_1\times Y_1+a_2\times Y_2+a_3\times Y_3+a_4\times Y_4+a_5\times Y_5+a_6\times Y_6+a_7\times Y_7+a_8\times Y_8 \quad (3.10)$$

for all of the possible 3×3 contexts in the image. Then we use the linear equation obtained to calculate $\hat{\text{CP}}$ for each 3×3 context individually. Four methods of incorporating the context of an image have been described, which are referred to as *mean* context, *median* context, *weighted* context and *regression* context, respectively.

Once we determine the context values for each pixel, the result is that we obtain a new image which is a slightly blurred version of the original image. These context images are shown in Figures 3.2-3.5 for the *mean*, *median*, *weighted* and *regression* contexts, respectively. There are two ways to continue with regard to any clustering method which incorporates context. The first is to use only the context values obtained from the transformation of the image by one of the four procedures described above, in which case it is just a matter of using the images shown in Figures 3.2-3.5 as the input images. The second approach is to use both the original pixel values and the new context values which results in an increase from a three dimensional problem to a six dimensional problem, where we would have six values, R, G, B, R′, G′ and B′, where R, G and B are the original red, green and blue values, and R′, G′ and B′ are the red, green and blue values produced by applying one of the context transformations. Even though the dimensionality is increased, this poses no problems for clustering methods which can easily be extended to cope with any dimensionality.

In this thesis the performance of the four context types described above are compared in terms of the segmentation results obtained when they are used. They are

compared to determine if better results can be obtained compared to when only the original colour values are used, and to determine if any one of the context types provides superior results.

3.4 Colour Spaces

A number of colour spaces exist in which the segmentation of images can occur [OKS80, Hun87, Jai89, ST97, LC99, Hea92, CF97, GS99]. Some colour models have also been proposed which encompass physics-based properties [Jan91]. Colour has become a very useful tool in areas such as automatic medical diagnosis [ECS+94], and one of the more recent investigations of colour involves its use in image representation and retrieval [CB99].

The most commonly used colour space is the RGB (red, green, blue) colour space. Other colour spaces are available, some of which consider the property of human perception such as HSV (hue, saturation, value) [FV84], HSL (hue, saturation, lightness) [FV84] and HSI (hue, saturation, intensity) [GW92], while other transformations aim to produce colour components which are orthogonal or decorrelated, such as KL (Karhunen-Loève) [And70, Fuk90], which involves a transformation which is composed of the eigenvectors of the correlation matrix of the data. Due to the fact that spectral primary sources do not yield a full gamut of reproducible colours, the CIE XYZ colour space was developed "with hypothetical primary sources such that all the spectral tristimulus values are positive" [Jai89]. The desired colour values can usually be obtained by some transformation of the original RGB values.

The XYZ colour values can be obtained by the following transformation [TPV94, Jai89]

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.812 & 0.011 \\ 0.000 & 0.010 & 0.990 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.11}$$

The HSV (hue, saturation, value) model [FV84] is defined by the following equations

$$V = max \tag{3.12}$$

$$S = \begin{cases} 0 & max = 0 \\ (max - min)/max & max \neq 0 \end{cases} \tag{3.13}$$

$$H = \begin{cases} undefined & S = 0 \\ 60 * \{(G - B)/(max - min)\} & R = max \\ 60 * \{2 + (B - R)/(max - min)\} & G = max \\ 60 * \{4 + (R - G)/(max - min)\} & B = max \end{cases} \tag{3.14}$$

If H < 0, then H is set to H + 360. In equations 3.12-3.14 min is the minimum of the R, G and B values and max is the maximum of the R, G, and B values.

In this thesis the RGB, XYZ, KL and HSV colour spaces are compared in terms of the segmentation results which are obtained by the proposed clustering methods.

3.5 Determining the Number of Clusters

3.5.1 Visual Analysis Survey

The most important aspect of clustering is actually determining the number of clusters present. With regard to colour image segmentation, one way to decide which is

the best clustering is to view the results, which would require a person to look at the results in comparison with the original image to see if it is a good segmentation. The problem with this approach is that this can be highly subjective as two different people could have two very different opinions about what a good segmentation is. Consequently, a survey was conducted of a group of ten people based on segmentation results produced by the *K-means* method as implemented in chapter 6, with the objective being to try and determine the desired range for the number of clusters for each of the natural images shown in Figure 1.2. The desired range for the number of clusters is determined based on what the majority of participants consider the best segmentation for a number of results with differing numbers of clusters. Obviously we could show all the image segmentation results from 2 up to, say, 25 clusters and ask the survey participants to choose which result is the best. However, instead we showed only the segmentation results for 3, 6, 9, 12 and 17 clusters, each one representing the mid-point of non-overlapping ranges of numbers of clusters. In doing this we can establish for each image a small range of values in which the number of clusters is expected to lie for each image. Table 3.1 shows the percentage of responses for each range for each of the natural images, as well as the overall optimal range for the number of clusters. Some of these ranges cover two ranges from those shown in Table 3.1 because there was a high number of responses for both of the ranges. Generally, any response of 10% or below was treated as insignificant and the result which produced this response was not considered to belong to the desired range for the number of clusters. Any response of 40% or above was considered significant and the entire range in these cases was considered as making up the optimal range, while any range which obtained a response of 20% or 30% was only considered to partially belong to the optimal range for the number of clusters.

clusters (range)	3 (2-4)	6 (5-7)	9 (8-10)	12 (11-13)	17 (14-20)	optimal range
balls			60%	40%		8-13
Lenna		50%	40%	10%		5-10
molecule		10%	80%	10%		8-10
teapot		20%	70%		10%	6-10
ant		50%	30%	20%		5-10
blond	10%	50%	40%			5-10
jet		100%				5-7
mandrill		60%	40%			5-10
peppers		30%	70%			6-10
mouse		20%	50%	30%		6-12
rose		40%	50%	10%		5-10

Table 3.1: Responses for survey of number of clusters.

This survey establishes a benchmark for the optimal number of clusters for each of the natural images, which can then help in determining if a method which automatically determines the number of clusters is performing well or not. However, this still involves human judgement.

3.5.2 Computational Evaluation of Clustering Results

Ideally, what is required is an objective evaluation criterion which incorporates the ideal aspects of clusters. Over the years, many criteria have been developed for determining the optimal partitioning of data [DB79, Dun74, BP98, Mil96, MC85, CM88, PB95], each of which has a common goal to find the clustering which results in compact clusters which are well separated. Also, with the advent of fuzzy clustering techniques, there has also been recent interest in developing fuzzy validity criteria [BJ81, Bez74, XB91, PB95, ZLE99]. Other validity measures have been proposed and applied on a particular method, such as the measure described by Langan *et al.* [LMZ98], which is a model-fitting technique for validating results by an image segmentation technique based on the EM-algorithm. In this technique, the complete data log-likelihood functional is modelled as an exponential function of the number of classes. There are two main approaches when using a validity measure

to determine the optimal number of clusters. The first approach involves obtaining the clustering for every possible number of clusters and then calculating the validity measure for each of these and determining which is the optimum value, which usually involves either maximizing or minimizing the validity criterion. The second approach involves incorporating the validity measure into the clustering algorithm itself with the determination of the number of clusters being part of the algorithm. Pauwels and Frederix [PF99] undertook the latter approach in which they defined an isolation measure and a connectivity measure to be used as part of a clustering technique based on nonparametric density estimation which would automatically determine the optimal clustering.

Zhang [Zha96] conducted a survey of evaluation methods for image segmentation. In this paper, the evaluation methods are classified into three groups, the analytical, the empirical goodness and the empirical discrepancy groups. The analytical methods "directly examine and assess the segmentation algorithms themselves by analysing their principles and properties" [Zha96]. Zhang reports that the analytical methods only work with some particular models or desirable properties of algorithms and they can be problematic as not all properties of segmentation algorithms can be obtained by analytical studies. The empirical methods generally evaluate the segmentation algorithms indirectly by somehow measuring the quality of segmentation results after applying them to some test images. For the empirical goodness methods, quality measures must be defined, most of which are established to represent the conditions which satisfy an "ideal" segmentation. These methods generally don't take into account the *a´priori* knowledge of the correct segmentation. The empirical discrepancy methods are based on taking the disparity between an actually segmented image, generated by a segmentation algorithm, and the ideal segmentation, and using this

to assess the performance of the algorithm. The ideal segmentation is often referred to as the reference image, and for synthetic images it is easy to generate, however, for real images manually segmented images are often used which are generated with the help of visual inspection, which in general will involve human judgement. In this thesis, the concern is mainly with the "empirical goodness" approach to evaluate the segmentation results and help decide on the correct number of clusters. A number of existing measures based on this approach will now be described.

One criterion which measures the compactness of a cluster is the average intra-cluster distance given by

$$aveintra = \frac{1}{N}\sum_{i=1}^{K}\sum_{\mathbf{x}\in C_i} || \mathbf{x} - \mathbf{z}_i || \tag{3.15}$$

When comparing two results, the one for which this measure is lowest is the best. However, when trying to compare results with different numbers of clusters, problems arise due to the fact that as the number of clusters increases, the value for this measure decreases, until it reaches its minimum value of 0 when the number of clusters is equal to the number of distinct patterns. Therefore, this measure for the average intra-cluster distance cannot be used in this way to determine the correct number of clusters.

Liu and Yang [LY94] defined an evaluation function to evaluate segmentation results. Their evaluation function F is defined as

$$F(I) = \sqrt{K} \times \sum_{i=1}^{K} \frac{e_i^2}{\sqrt{N_i}} \tag{3.16}$$

where I is the image to be segmented, K is the number of regions in the segmented image, N_i is the area or the number of pixels in the ith region and e_i is the colour

error of region i, defined as the sum of the Euclidean distances between the original image and the segmented image for each pixel in the region. The incorporation of the $\sqrt{K}$ term penalizes the segmentation which forms too many regions, and the term $e_i^2/\sqrt{N_i}$ penalizes small regions, or regions with a large colour error. The smaller the value of F, the better the segmentation result. Such a measure only focuses on the within-cluster compactness, while not taking into account the cluster separation.

An improved validity measure would consider both the within-cluster compactness and between-cluster separation. There are two main approaches which can be used for this, statistical and mathematical. Devijver and Kittler [DK82] discuss statistical measures which are based on the construction of a between-cluster scatter matrix, S_b, and a within-cluster scatter matrix, S_w. The between-cluster scatter matrix is given by

$$S_b = \sum_{i=1}^{K} \frac{N_i}{N} (\mathbf{z}_i - \mathbf{z})(\mathbf{z}_i - \mathbf{z})^T \tag{3.17}$$

where N_i is the number of patterns in cluster C_i, N represents the total number of patterns, $\mathbf{z}_i$ is the cluster centre of cluster C_i and $\mathbf{z}$ is the mean of all the patterns. The within-cluster scatter matrix is given by

$$S_w = \frac{1}{N} \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} (\mathbf{x} - \mathbf{z}_i)(\mathbf{x} - \mathbf{z}_i)^T \tag{3.18}$$

where $\mathbf{x}$ represents the pattern. Devijver and Kittler [DK82] go on to describe four basic criteria based on these scatter matrices, the first given by

$$J_1(\mathbf{x}) = \mathrm{tr}(S_w + S_b) \tag{3.19}$$

where $\mathrm{tr}(\cdot)$ indicates the "trace" or sum of the diagonal elements of the matrix.

The ideal clustering is one which leads to a large between-cluster separation and a small within-cluster separation. Therefore, the criterion $J_1(\mathbf{x})$ would not be a good indicator of cluster separation as there would be no way of knowing if this had been achieved based solely on the magnitude of this measure. Therefore, Devijver and Kittler [DK82] describe a second criterion function based on the ratio between the between-cluster and within-cluster scatters, given by

$$J_2(\mathbf{x}) = \frac{\text{tr } S_b}{\text{tr } S_w} \tag{3.20}$$

This measure takes into account the within-cluster variability by taking the trace of the matrices, however, it totally ignores the correlation between the components and the effect these have on cluster separability. Consequently, Devijver and Kittler [DK82] also discuss a third criterion which is based on first preprocessing the pattern vectors by a "suitable transformation U so that the average covariance matrix of the transformed patterns is the identity matrix" [DK82], which is indicated by the equation

$$U^T S_w U = I \tag{3.21}$$

They indicate that the matrix

$$U = S_w^{-1/2} \tag{3.22}$$

satisfies the condition given in equation 3.21. This leads to the third criterion, which is given by

$$J_3(\mathbf{x}) = \text{tr } S_w^{-1/2} S_b S_w^{-1/2} = \text{tr } S_w^{-1} S_b = \sum_{i=1}^{p} \lambda_i \tag{3.23}$$

where $\lambda_i, i = 1, \ldots, p$ are the eigenvalues of matrix $S_w^{-1} S_b$. A fourth measure which Devijver and Kittler [DK82] describe is based on the total scatter, which is simply

the determinant of the mixture population covariance matrix, Σ, which is defined by

$$\Sigma = E\left\{(\mathbf{x}-\mathbf{z})(\mathbf{x}-\mathbf{z})^T\right\} \tag{3.24}$$

Since greater spatial separation of the clusters is achieved when the ratio of the total-scatter and within-cluster scatter is greatest, this leads to the fourth criterion which is given by

$$J_4(\mathbf{x}) = \frac{\mid \Sigma \mid}{\mid S_w \mid} \tag{3.25}$$

Further analysis by Devijver and Kittler [DK82] indicates that both the matrices Σ and S_w are symmetric so there exists a matrix U that diagonalizes both of them so that

$$U^T \Sigma U = \Lambda \tag{3.26}$$

and

$$U^T S_w U = I \tag{3.27}$$

Therefore, criterion $J_4(\mathbf{x})$ can also be expressed in terms of the diagonal elements, λ_i, of matrix Λ as

$$J_4(\mathbf{x}) = \frac{\mid U^T \Sigma U \mid}{\mid U^T S_w U \mid} = \prod_{i=1}^{p} \lambda_i \tag{3.28}$$

Coleman and Andrews [CA79] made use of the scattar matrices in a method they proposed, based on the *K-means* algorithm, which also determined the correct number of clusters. They used a measure based on the within-cluster and between-cluster scatter matrices given by

$$J_5(\mathbf{x}) = \text{tr } S_b \cdot \text{tr } S_w \tag{3.29}$$

The desired number of clusters is indicated by this measure being maximized.

The second main approach is the mathematical approach where cluster separation can be measured by a distance metric. The advantage this has over the statistical approach is that once the distance is calculated it is a single measure which can be easily compared to another, however, the statistical approach requires some ad hoc way of deriving a single number for comparability purposes, such as taking the trace or determinant of the matrices, or doing an eigenvalue analysis. One criterion based on the use of distance metrics is the Davies-Bouldin index [DB79], which is a function of the ratio of the sum of within-cluster scatter to between-cluster separation. For this measure, the within cluster scatter for cluster C_i is defined as

$$S_{i,q} = \left(\frac{1}{N_i} \sum_{\mathbf{x} \in C_i} \| \mathbf{x} - \mathbf{z}_i \|_2^q \right)^{1/q} \tag{3.30}$$

and the distance between clusters C_i and C_j is defined as

$$d_{ij,t} = \left\{ \sum_{s=1}^{p} | z_{is} - z_{js} |^t \right\}^{1/t} = \| \mathbf{z}_i - \mathbf{z}_j \|_t \tag{3.31}$$

Next, define

$$R_{i,qt} = \max_{j, j \neq i} \left\{ \frac{S_{i,q} + S_{j,q}}{d_{ij,t}} \right\} \tag{3.32}$$

The Davies-Bouldin index is then defined as

$$DB = \frac{1}{K} \sum_{i=1}^{K} R_{i,qt} \tag{3.33}$$

The objective is to minimize this measure as we want to minimize the within-cluster scatter and maximize the between-cluster separation.

Bezdek and Pal [BP98] have generalized Dunn's index [Dun74] as follows

$$D_{ij} = \underbrace{\min}_{1\leq s\leq K} \left\{ \underbrace{\min}_{1\leq t\leq K, t\neq s} \left\{ \frac{\delta_i(C_s, C_t)}{\underbrace{\max}_{1\leq l\leq K} \{\Delta_j(C_l)\}} \right\} \right\} \tag{3.34}$$

where $\delta_i(C_s, C_t)$ represents the distance between clusters C_s and C_t and $\Delta_j(C_l)$ represents the diameter of cluster C_l. Bezdek and Pal [BP98] defined various set distance functions that can be used in equation 3.34, which are

$$\delta_1(C_s, C_t) = \underbrace{\min}_{\mathbf{x}\in C_s, \mathbf{y}\in C_t} \{\mathrm{d}(\mathbf{x}, \mathbf{y})\} \tag{3.35}$$

$$\delta_2(C_s, C_t) = \underbrace{\max}_{\mathbf{x}\in C_s, \mathbf{y}\in C_t} \{\mathrm{d}(\mathbf{x}, \mathbf{y})\} \tag{3.36}$$

$$\delta_3(C_s, C_t) = \frac{1}{N_s \times N_t} \sum_{\mathbf{x}\in C_s, \mathbf{y}\in C_t} \mathrm{d}(\mathbf{x}, \mathbf{y}) \tag{3.37}$$

$$\delta_4(C_s, C_t) = \mathrm{d}(\mathbf{z}_s, \mathbf{z}_t) \tag{3.38}$$

where

$$\mathbf{z}_s = \frac{1}{N_s} \sum_{\mathbf{x}\in C_s} \mathbf{x}$$

and

$$\mathbf{z}_t = \frac{1}{N_t} \sum_{\mathbf{y}\in C_t} \mathbf{y}$$

$$\delta_5(C_s, C_t) = \frac{1}{N_s + N_t} \left(\sum_{\mathbf{x}\in C_s} \mathrm{d}(\mathbf{x}, \mathbf{z}_t) + \sum_{\mathbf{y}\in C_t} \mathrm{d}(\mathbf{y}, \mathbf{z}_s) \right) \tag{3.39}$$

$$\delta_6(C_s, C_t) = \max\{\delta(C_s, C_t), \delta(C_t, C_s)\} \tag{3.40}$$

where

$$\delta(C_s, C_t) = \underbrace{\max}_{\mathbf{x}\in C_s} \{\underbrace{\min}_{\mathbf{y}\in C_t} \{\mathrm{d}(\mathbf{x}, \mathbf{y})\}\}$$

$$\delta(C_t, C_s) = \underbrace{\max}_{\mathbf{y} \in C_t} \{ \underbrace{\min}_{\mathbf{x} \in C_s} \{ \mathrm{d}(\mathbf{x}, \mathbf{y}) \} \}$$

The set distance, δ_6, is also known as the Hausdorff metric [PS87]. Bezdek and Pal also defined various diameter measures which can be used in equation 3.34, which are

$$\Delta_1(C_s) = \underbrace{\max}_{\mathbf{x}, \mathbf{y} \in C_s} \{ \mathrm{d}(\mathbf{x}, \mathbf{y}) \} \tag{3.41}$$

$$\Delta_2(C_s) = \frac{1}{N_s \times (N_s - 1)} \sum_{\mathbf{x}, \mathbf{y} \in C_s, \mathbf{x} \neq \mathbf{y}} \mathrm{d}(\mathbf{x}, \mathbf{y}) \tag{3.42}$$

$$\Delta_3(C_s) = 2 \left(\frac{\sum_{\mathbf{x} \in C_s} \mathrm{d}(\mathbf{x}, \mathbf{z}_s)}{N_s} \right) \tag{3.43}$$

Equations 3.35-3.43 give eighteen possible combinations of set distance and diameter for D_{ij} where $1 \leq i \leq 6$ and $1 \leq j \leq 3$. D_{11} represents the original definition of Dunn's index. Pal and Biswas [PB97] generalized both the Davies-Bouldin index and Dunn's index for cluster validation using graph structures, including Gabriel Graphs, Relative Neighbourhood Graphs and Minimal Spanning Trees. The main advantages of this was that these approaches are not sensitive to noisy points in the data and are applicable to both hyperspherical and structural clusters, such as lines, arcs, etc.

Due to the ad hoc nature of the statistical approach, in this thesis the mathematical approach involving distances is used instead. One of the disadvantages of measures such as the Davies-Boudlin [DB79] and Dunn's indexes [Dun74, BP98] is that, when applied to colour image data, there is a tendency to select the result with a small number of clusters, such as 2 or 3, which is shown in chapter 7, and this is well below what we would expect. Due to this problem a new validity criterion is proposed in chapter 7 which incorporates a penalty function, which is a function of the number of clusters, to enable these results to be dismissed in favour of results containing more clusters and which are also more visually appealing.

Chapter 4

Intra-cluster versus Inter-cluster Distance

4.1 Intra-cluster and Inter-cluster Distances as Evaluation Criteria

Validity is an important aspect of image segmentation. Once a segmentation result is obtained from a particular method, it is necessary to determine how good the result actually is and to be able to somehow compare it to the segmentation result from another method. The accuracy in segmentation of a colour image depends not only on the algorithm used but also on the colour space selected [OKS80]. One obvious way to establish the goodness of the segmentation is to visually inspect the segmented image. Naturally, this would require human judgement and is also limited by the fact that two people could have differing views as to whether the result was good or not. This makes an objective evaluation of segmentation results all the more necessary in the analysis of colour images.

A simple measure on which to base the evaluation of clustering-based segmentation results is the intra-cluster distance, which is simply the sum of the squared Euclidean distances between every point and its cluster centre. This, therefore, measures the

within cluster variability. The intra-cluster distance is given by

$$intra_E = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} \sum_{j=1}^{3} (x_j - z_{ij})^2 \tag{4.1}$$

where K is the number of clusters, $\mathbf{x}$ is a 3-tuple pixel value $(x_1, x_2, x_3)^T$, C_i is cluster i and $\mathbf{z}_i = (z_{i1}, z_{i2}, z_{i3})^T$ is the cluster mean of cluster C_i as a 3-tuple value. An alternative is to base the measure on the absolute distance measure in which case equation 4.1 would become

$$intra_A = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} \sum_{j=1}^{3} \mid x_j - z_{ij} \mid \tag{4.2}$$

Ideally, we want the clusters to be as compact as possible, so the within cluster variability should be as small as possible. Therefore, the intra-cluster distance measure can be used as a criterion to measure how good the segmentation is, as the smaller the intra-cluster distance, the more well formed the clusters are and the better the segmentation result.

Another measure, which is complementary to the intra-cluster distance, is the inter-cluster distance. This is simply the sum of the squared distances between each of the cluster centres, and therefore gives an overall measure of separation between the cluster centres. This measure is defined as

$$inter_E = \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \sum_{l=1}^{3} (z_{il} - z_{jl})^2 \tag{4.3}$$

where K is the number of clusters, and $\mathbf{z}_i = (z_{i1}, z_{i2}, z_{i3})^T$ and $\mathbf{z}_j = (z_{j1}, z_{j2}, z_{j3})^T$ are the centres of cluster C_i and cluster C_j, respectively. This is squared so that it may be compared with $intra_E$ given by equation 4.1. As with the intra-cluster distance,

the inter-cluster distance can also be calculated using the absolute distance measure, so equation 4.3 would become

$$inter_A = \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \sum_{l=1}^{3} \mid z_{il} - z_{jl} \mid \tag{4.4}$$

Under the assumption that better separation between cluster centres leads to better segmentation of an image, this simple criterion can then be used as a performance index of the overall segmentation accuracy. Thus, two segmentation results can be compared based on the value of the inter-cluster distance, a higher value indicating the superiority of a result.

4.1.1 An Experimental Comparison of Two Clustering Methods

An experimental study was undertaken which was aimed at being able to decide out of two clustering methods, which was superior. The methods investigated were *K-means* (KM) [TG74] and *thresholding and fuzzy K-means* (TFKM) [LL90], both of which were described in chapter 2. The initial cluster centres for the *K-means* method were chosen to be the first K distinct colours which occur in the image. The TFKM method was chosen as the method automatically detects the number of clusters, then the KM procedure can be executed with the same number of clusters supplied as the parameter.

Results were obtained on the eleven natural colour images shown in Figure 1.2. A number of experiments were conducted in all of which the computations were carried out using the RGB colour space. As mentioned earlier, the TFKM method automatically determines the number of clusters. Table 4.1 shows the number of

clusters obtained by the TFKM method for each of the images by taking the threshold for the cluster size as 0, which results in all clusters with at least 1 pixel to be considered a valid class.

image	No. of Clusters
balls	16
Lenna	12
molecule	13
teapot	16
ant	19
blond	10
jet	14
mandrill	34
peppers	21
mouse	6
rose	9

Table 4.1: Number of clusters determined by TFKM method.

Table 4.2 gives the intra-cluster distances for the two methods using both *Euclidean* and *absolute* distances and Table 4.3 gives the inter-cluster distances for the two methods using both *Euclidean* and *absolute* distances.

method	KM	TFKM	KM	TFKM
distance	Euclidean	Euclidean	absolute	absolute
balls	29,648,855	55,157,107	1,704,421	2,671,157
Lenna	21,906,659	30,831,878	1,583,925	1,943,586
molecule	3,328,666	4,891,063	241,550	292,649
teapot	13,046,172	17,683,112	992,784	1,129,848
ant	31,274,250	23,124,868	1,756,140	1,591,408
blond	17,635,529	22,692,780	1,296,453	1,507,072
jet	14,695,971	29,318,023	1,159,761	1,879,395
mandrill	34,878,747	33,282,386	1,939,293	1,963,920
peppers	26,135,052	30,234,187	1,696,746	1,811,497
mouse	18,941,301	19,333,646	1,238,375	1,278,582
rose	41,451,070	35,557,375	2,079,458	2,061,909

Table 4.2: Intra-cluster distance values.

As expected the *K-means* method results in the lower intra-cluster distance measure for most images, as this method is based on trying to minimize this measure. Therefore, we focus on the inter-cluster distance measure and see if visual analysis of the segmented results matches with the method producing the larger inter-cluster distance. For the most part the visual analysis agreed with the method obtaining the maximum inter-cluster distance. However, in the case of *balls* and *peppers*, the visual

method	KM	TFKM	KM	TFKM
distance	Euclidean	Euclidean	absolute	absolute
balls	4,152,764	2,966,043	29,084	27,042
Lenna	757,499	1,107,354	9,976	12,248
molecule	1,899,435	1,535,345	16,986	15,018
teapot	2,555,357	2,927,844	24,988	26,683
ant	5,350,773	3,895,107	38,414	35,977
blond	439,850	1,035,305	6,133	9,627
jet	1,257,463	1,402,485	14,834	16,152
mandrill	9,235,560	11,801,660	99,832	114,683
peppers	5,004,846	4,412,679	45,047	42,392
mouse	513,977	442,612	4,150	3,845
rose	954,485	892,455	8,226	8,004

Table 4.3: Inter-cluster distance values.

result indicated a mismatch with the inter-cluster distance. The segmented images by the TFKM method are shown in Figure 4.1 and the segmented results for the KM method are shown in Figure 4.2. All of these results are based on the number of clusters indicated in Table 4.1.

The image *balls* was segmented much better using the TFKM method as the background was classified as only one segment, whereas with the KM method, the background was divided into two different segments. This anomaly is due to the nature of the methods as TFKM is mostly a thresholding technique, however, KM is attempting to minimize the within cluster squared distance, and this can only be done by dividing the background into two segments as KM prefers smaller clusters to reduce the within cluster squared distance. The TFKM result also created a separate segment for the yellow ball, which the KM result failed to do as it grouped the yellow ball region together with part of the shadow region.

From the visual perspective, *peppers* was segmented better using TFKM as the regions in the image are more connected and smooth, so that some of the individual peppers are treated as one segment. This disagrees with the result in Table 4.3 above which indicates that KM performed better.

In summary, Table 4.3 indicates that half of the time the KM method is better

and the other half the TFKM method is better. This was true for both *Euclidean* distance and *absolute* distance formulations of the inter-cluster distance. However, visual analysis of segmented images was not always in conformity with the method producing the higher inter-cluster distance measure. The answer lies primarily in the intricate details of the differences between the two methods, the statistical distributions of the R, G, and B components in different image segments, the correlations between these three colour components, and the distance criterion used.

4.2 Correlation Between Intra-cluster and Inter-cluster Distances

Another experimental study was undertaken to establish the exact relationship between the intra-cluster and inter-cluster distances. The relationship between the intra-cluster distance and the inter-cluster distance can be determined by testing the statistical significance of the correlation between them, which is expected to have a negative value. The correlation can be calculated from different values of the intra-cluster and inter-cluster distances obtained by varying the number of clusters. By altering the threshold for the cluster size in the TFKM method, we were able to obtain results for different numbers of clusters, and for the KM method the number of clusters was supplied as a parameter, thus allowing results with varying numbers of clusters to be obtained for both methods, where the number of clusters in Table 4.1 indicates the maximum number of clusters. The correlation between the intra-cluster distance and the inter-cluster distance is given in Table 4.4.

According to Goon *et al.* [GGD71], the equation given by

$$\frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \tag{4.5}$$

method	KM	TFKM	KM	TFKM
distance	Euclidean	Euclidean	absolute	absolute
balls	-0.606159	-0.666351	-0.636276	-0.768319
Lenna	-0.721038	-0.609797	-0.784063	-0.686053
molecule	-0.799667	-0.816780	-0.823355	-0.831155
teapot	-0.626696	-0.596167	-0.660683	-0.663069
ant	-0.470315	-0.544404	-0.558179	-0.640719
blond	-0.898688	-0.603416	-0.894089	-0.597399
jet	-0.726745	-0.787452	-0.761843	-0.830289
mandrill	-0.601290	-0.596420	-0.745109	-0.711207
peppers	-0.619396	-0.617154	-0.694553	-0.690754
mouse	-0.864288	-0.835591	-0.882500	-0.839078
rose	-0.708259	-0.755589	-0.802579	-0.829784

Table 4.4: Correlation between intra-cluster and inter-cluster distances.

where r represents the correlation coefficient and n is the number of pairs of points used in the calculation of the correlation coefficient, has a t-distribution with $n-2$ degrees of freedom. In order to determine whether the correlation between the intra-cluster and inter-cluster distances is statistically significant to a confidence interval of 95%, the value for equation 4.5 must be greater than the value of $t_{0.05,n-2}$. Alternatively, we can calculate the value of r by the following equation

$$r = \sqrt{\frac{t^2_{0.05,n-2}}{n-2+t^2_{0.05,n-2}}} \tag{4.6}$$

and statistical significance is achieved if the absolute value of the given correlation coefficient is greater than r. For each of the test images, the value of n is one less than the number of clusters shown in Table 4.1 since the smallest number of clusters used was 2. There was one exception to this, which occurred for the image *teapot* when the TFKM method was used. Here, n is two less than the number of clusters shown in Table 4.1 due to the fact that two of the clusters contained the same number of pixels resulting in both of these clusters being considered invalid for the same threshold value. Table 4.5 shows the values of r for the relevant values of n. Comparing the relevant values in Table 4.5 to the correlation coefficient values in Table 4.4, we can see that each of the correlation coefficients between the intra-cluster

and inter-cluster distances is highly statistically significant. In general, the images which obtained a higher correlation coefficient are divided evenly between the TFKM and KM methods, regardless of which distance measure is used.

n	r
5	0.805340
8	0.621454
9	0.582291
11	0.521381
12	0.497170
13	0.476180
14	0.457442
15	0.440874
18	0.400049
20	0.378329
33	0.291344

Table 4.5: Correlation values for significance test.

Results seem to indicate that there is a statistically significant relationship between the sum of the inter-cluster distances and the intra-cluster distance measure. Given that they are complementary measures in the sense that as one is increased, the other decreases, and vice versa, it is acceptable to use just one of these measures when assessing the validity of a result. It is also possible to incorporate these measures into the methods themselves, so that the validation process is carried out during the program to obtain the result, rather than after the result has already been obtained. The next few chapters investigate various ways of incorporating these two measures into clustering algorithms.

Chapter 5

Merging Methods Based on the Principle of Cluster Separation

An important aspect of image segmentation is to determine the actual number of segments present in an image. As mentioned in chapter 2, many clustering-based segmentation methods require the number of clusters to be supplied as a parameter, requiring some *a ´priori* knowledge about the image, while other methods require a large list of parameters to allow the number of clusters to be determined, leading to bias in the results. In this chapter some simple criteria will be presented, based on the inter-cluster distance measure, to enable the number of clusters to be determined automatically. The inter-cluster distance measure was chosen, as it allows us to measure the separation between clusters, and the more separated the clusters are, the more confident we can be about the segmentation results.

The first method is based on the *agglomerative hierarchical* technique, where the number of starting clusters is equal to the number of distinct colours in the image, and during each iteration only the pair of clusters whose centres are closest are merged together. The criterion used to determine when to stop the merging process is based on the average inter-cluster distance, which is multiplied by some *weight*

value to determine a threshold which the minimum inter-cluster distance must be bigger than. Due to the fact that the average inter-cluster distance will change from iteration to iteration as the number of clusters decrease, the threshold value also changes for each iteration of the algorithm. The second method allows more than two clusters to be merged during an iteration by allowing every cluster to be merged with its closest cluster provided the inter-cluster distances between the clusters are less than some threshold. More than two clusters can be merged due to the fact that just because cluster C_i is closest to cluster C_j, this does not mean that cluster C_j is closest to cluster C_i, leading to cluster C_j being merged with cluster C_i and cluster C_i being merged with another cluster, which would result in three clusters being merged together. The threshold value is calculated in two ways. The first is to make it the maximum of the minimum inter-cluster distances for each cluster from the first iteration, and the second is to base it on the mean of these values plus a set amount of standard deviations. The second option would allow some of the colours to be treated as outliers, allowing them to remain as separate clusters.

5.1 Merging Method 1

5.1.1 Description of Method

The primary aim of this method is that it should automatically detect the number of clusters present in the image, and the secondary aim is that the process should be entirely automated, with no user input required. This method which has been developed is based on the principle that we would like colours which are "close" to one another to be grouped in the same cluster, and we would also like the clusters to be well separated. It is mainly an iterative technique, but may be thought of as

a two stage process. The first stage requires one pass through the image to collect information about how many actual colours, or red, green, blue combinations, are present in the image. At this time, information about the number of pixels in the image having a certain colour is also obtained. This in fact gives us what may be thought of as initial clusters. This can be described by the following algorithm:

```
for every pixel in the image
    if the (r,g,b) values have not been seen before
        add the new colour to the list of colours already obtained and
        indicate that the colour has only been seen once before
    else
        add one to the total number of pixels seen for that colour
    end if
end for
```

The second stage, which is the iterative stage, continually calculates the minimum inter-cluster distance and records the clusters which produced this, and then merges these two clusters together. It is obvious that some way to determine when to stop is necessary, otherwise this process would continue until only one cluster remained. If the minimum inter-cluster distance measure is small, then the two clusters should be merged. From the opposite view point, if the minimum inter-cluster distance is large enough, no more merging is required as good cluster separation has been achieved, and the process can end. It is therefore necessary to compare the minimum inter-cluster distance against some threshold value to determine if it is small enough to continue merging. The threshold value chosen is based on the average inter-cluster distance, and is actually formed by multiplying this by some *weight* value. Therefore, the two clusters which produce the minimum inter-cluster distance are merged if the minimum inter-cluster distance value is less than some *weight* multiplied by the

average inter-cluster distance. The process stops when the minimum inter-cluster distance is greater than or equal to the threshold at which time the final number of clusters is recorded. Note that since the average inter-cluster distance changes from iteration to iteration, the threshold value used is also changing from iteration to iteration. This iterative stage can be represented by the following algorithm:

```
set K to number of colours from stage 1
loop
    calculate the minimum inter-cluster distance and record the two
    clusters, C_i and C_j, which produced it
    calculate the average inter-cluster distance
    if (minimum < weight × average)
        merge clusters C_i and C_j
        decrement K
    else
        exit loop
    end if
end loop
create segmented image based on K clusters
```

This method is very similar to the single-link *hierarchical* clustering method, and is actually the same as the centroid method variation [Eve74]. The main focus is on the criterion which is used to determine the number of clusters. The number of clusters will be dependent on the *weight* value which is selected. The optimal value for the *weight* can be determined experimentally, by selecting a number of different values for the *weight* and performing the clustering for each of the test images based on these values.

5.1.2 Implementation Details

The method was implemented by allowing two different distance measures to determine the "closeness" of two clusters, namely, Euclidean and absolute distances. For two clusters C_i and C_j whose cluster centres are given by $\mathbf{z}_i = (R_i, G_i, B_i)$ and $\mathbf{z}_j = (R_j, G_j, B_j)$, respectively, the Euclidean distance can be calculated as

$$d_{ij} = \sqrt{(R_i - R_j)^2 + (G_i - G_j)^2 + (B_i - B_j)^2} \tag{5.1}$$

and the absolute distance can be calculated as

$$d_{ij} = |R_i - R_j| + |G_i - G_j| + |B_i - B_j| \tag{5.2}$$

We have also allowed for two representations for the cluster centres. These are mean and median. Therefore, this gives us four possible combinations of distance measure and cluster centre type, namely, mean cluster centres with Euclidean distance, mean cluster centres with absolute distance, median cluster centres with Euclidean distance, and median cluster centres with absolute distance. Trying these different combinations should enable us to determine whether a particular distance measure or a particular representation of the cluster centre has any advantage over the others.

Since there are two ways to represent the cluster centres, there are also two methods for merging two clusters together. If we wished to merge the two clusters C_i and C_j given above, with the number of pixels in each of these clusters being N_i and N_j, respectively, then for the mean cluster centre representation, the new cluster centre

$\mathbf{z}_{ij} = (R_{ij}, G_{ij}, B_{ij})$ would be calculated as

$$R_{ij} = (N_i \times R_i + N_j \times R_j) \div (N_i + N_j) \tag{5.3}$$

$$G_{ij} = (N_i \times G_i + N_j \times G_j) \div (N_i + N_j) \tag{5.4}$$

$$B_{ij} = (N_i \times B_i + N_j \times B_j) \div (N_i + N_j) \tag{5.5}$$

The calculation of the merged cluster centre using the median representation is more complicated. To calculate the median value we must first scan through, and record in numerical order, all the values in either of the two clusters. If the number of pixels in the merged cluster, $N_i + N_j$, is odd, we take the middle-most value as the median, otherwise, if the number of pixels in the new cluster is even, then we represent the median by the average of the two middle-most values. This process must be done for each of the three bands. This is more computationally expensive than the simple calculation shown above for the mean cluster centre, so we would only keep this as our preferred choice of cluster centre representation if the segmentation results were vastly improved by the median cluster centre values.

5.1.3 Results

In the algorithm described earlier, there is an unknown quantity, *weight*, which will ultimately determine the number of clusters in the final segmentation results. Therefore, the procedure as described is not totally automated. However, we can overcome this problem by experimentation with different values for *weight*, thus finding its optimal value, or at the very least to be able to restrict it to a small range of possible values. In undertaking this task, this method was first implemented with synthetic images for which the ideal clustering was known beforehand to establish the range of

weight values which produced the correct results. This method was then applied to natural colour images.

5.1.3.1 Synthetic Images

The six synthetic images shown in Figure 1.1 were used to test this method. Tables 5.1-5.4 show the number of clusters obtained for various values of *weight*. Table 5.1 gives the results obtained when using mean cluster centres and Euclidean distance, Table 5.2 gives the results obtained when using mean cluster centres with absolute distance, Table 5.3 shows the results obtained when using median cluster centres and Euclidean distance, while Table 5.4 shows the results obtained when using median cluster centres with absolute distance. These results are based on the use of the RGB colour space.

weight	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
two	2	2	2	2	2	2	2	2	2	2
four	4	4	4	4	4	4	4	2	2	2
six	6	6	6	6	6	6	6	3	2	2
eight	8	8	8	8	8	8	8	6	3	2
ten	10	10	10	10	10	8	6	6	2	2
fifteen	16	15	15	15	15	11	7	4	3	2

Table 5.1: Number of clusters for synthetic images using mean cluster centres and Euclidean distance.

weight	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
two	2	2	2	2	2	2	2	2	2	2
four	4	4	4	4	3	2	2	2	2	2
six	6	6	6	6	6	4	3	2	2	2
eight	8	8	8	8	8	6	3	2	2	2
ten	12	10	10	10	10	7	4	4	2	2
fifteen	17	15	15	14	14	8	3	3	3	2

Table 5.2: Number of clusters for synthetic images using mean cluster centres and absolute distance.

Table 5.5 shows the range of *weight* values which produced the correct clustering. Generally, the *weight* values ranging from 0.2 to 0.4 produced correct segmentation for the synthetic images.

weight	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
two	2	2	2	2	2	2	2	2	2	2
four	4	4	4	4	4	4	4	2	2	2
six	6	6	6	6	6	6	6	3	2	2
eight	8	8	8	8	8	8	8	6	3	2
ten	10	10	10	10	10	8	8	6	4	2
fifteen	16	15	15	15	15	9	9	4	3	2

Table 5.3: Number of clusters for synthetic images using median cluster centres and Euclidean distance.

weight	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
two	2	2	2	2	2	2	2	2	2	2
four	4	4	4	4	3	2	2	2	2	2
six	6	6	6	6	6	3	3	2	2	2
eight	8	8	8	8	8	6	3	2	2	2
ten	11	10	10	10	10	3	3	2	2	2
fifteen	16	15	15	14	14	8	4	2	2	2

Table 5.4: Number of clusters for synthetic images using median cluster centres and absolute distance.

5.1.3.2 Natural Images

This method was executed for the eleven natural images shown in Figure 1.2. The results in this section are based on the use of the RGB colour space. Table 5.6 shows the number of actual colours present in each image, determined by the first stage of the algorithm. In most cases, there are approximately 250 colours in the images, which is much less than the number of pixels in the images.

Tables 5.7-5.10 show the number of clusters obtained for various values of *weight*. Table 5.7 gives the results obtained when using mean cluster centres and Euclidean distance, Table 5.8 gives the results obtained when using mean cluster centres with absolute distance, Table 5.9 shows the results obtained when using median cluster

cluster centre distance	mean Euclidean	mean absolute	median Euclidean	median absolute
two	0.1-1.0	0.1-1.0	0.1-1.0	0.1-1.0
four	0.1-0.7	0.1-0.4	0.1-0.7	0.1-0.4
six	0.1-0.7	0.1-0.5	0.1-0.7	0.1-0.5
eight	0.1-0.7	0.1-0.5	0.1-0.7	0.1-0.5
ten	0.1-0.5	0.2-0.5	0.1-0.5	0.2-0.5
fifteen	0.2-0.5	0.2-0.3	0.2-0.5	0.2-0.3

Table 5.5: *weight* values producing correct segmentation for synthetic images.

image	No. of colours
balls	252
Lenna	254
molecule	1426
teapot	7039
ant	255
blond	256
jet	256
mandrill	256
peppers	256
mouse	223
rose	240

Table 5.6: Number of original colours.

centres and Euclidean distance, while Table 5.10 shows the results obtained when using median cluster centres with absolute distance.

weight	0.2	0.3	0.4	0.5	0.6
balls	48	23	11	7	4
Lenna	52	21	15	8	4
molecule	34	19	9	8	5
teapot	36	16	11	5	4
ant	47	21	11	6	4
blond	45	20	11	4	4
jet	50	27	15	8	5
mandrill	99	41	21	11	6
peppers	59	22	10	8	7
mouse	16	6	4	3	3
rose	27	12	9	5	3

Table 5.7: Number of clusters using mean cluster centres and Euclidean distance.

weight	0.2	0.3	0.4	0.5	0.6
balls	35	14	8	4	3
Lenna	43	24	14	7	4
molecule	39	17	10	7	4
teapot	35	17	6	4	4
ant	36	14	9	6	3
blond	37	14	7	3	2
jet	45	24	13	8	6
mandrill	66	32	17	10	5
peppers	52	15	9	5	4
mouse	12	6	4	3	2
rose	23	12	7	3	3

Table 5.8: Number of clusters using mean cluster centres and absolute distance.

When comparing the number of clusters obtained for a particular image and a particular *weight*, we can notice that the number of clusters is very similar regardless of which cluster centre representation is used, or which distance measure is used. Therefore, in order to determine which cluster centre representation and distance

weight	0.2	0.3	0.4	0.5	0.6
balls	43	22	10	7	3
Lenna	47	26	17	9	4
molecule	41	17	9	8	6
teapot	31	17	9	4	4
ant	47	22	7	6	4
blond	48	17	11	5	4
jet	53	24	12	9	5
mandrill	94	32	13	7	4
peppers	67	23	11	8	7
mouse	13	5	4	3	3
rose	31	10	9	5	3

Table 5.9: Number of clusters using median cluster centres and Euclidean distance.

weight	0.2	0.3	0.4	0.5	0.6
balls	35	12	7	5	3
Lenna	46	20	11	8	4
molecule	45	18	8	5	3
teapot	34	15	8	5	5
ant	37	17	7	6	6
blond	34	13	7	5	2
jet	40	18	11	6	4
mandrill	67	28	13	8	6
peppers	53	19	11	6	4
mouse	11	6	4	3	3
rose	26	14	7	6	3

Table 5.10: Number of clusters using median cluster centres and absolute distance.

measure is the best, visual analysis of the segmented results is required. Visual analysis is also required to determine if there is an optimum value for *weight*. This is done by comparing the segmented images obtained for the test set of images using each of the different possible values for *weight*. In general, a *weight* of above 0.5 resulted in too few segments to adequately represent each of the regions in the images, and a *weight* below 0.3 resulted in too many segments, although, the more regions that are present in the final segmented images, the more the segmented image looks like the original image. Following is a detailed analysis of the results for each image. The results are summarized in section 5.1.4.

When a *weight* of 0.3 was used for the image *balls*, too many clusters were produced for every segmentation, except the median cluster centres and absolute distance approach, which is especially evident with the mean and absolute and median and

Euclidean segmented images which resulted in the background being considered as two separate segments. Each of the results using a *weight* of 0.5 has too few segments to properly represent the colours of each of the individual balls, as the red ball was classified as belonging to the same cluster as the black ball, and the blue ball was also classified as belonging to the same cluster as the black ball when absolute distance was used. For the four approaches, the best results were obtained with a *weight* of 0.4, except for the median and absolute method for which the best result was obtained with the *weight* of 0.3. The result obtained using the *weight* of 0.4 for the mean cluster centres and Euclidean distance approach achieved the best segmentation as the red ball was correctly classified as a separate cluster from the black ball, the white ball was shown by an outline, even though the outline was from the same cluster as the shadow, and the pink ball was a more smooth and connected region.

The image *Lenna* obtained too many clusters when a *weight* of 0.3 was used. The results obtained using a *weight* of 0.5 were not that good as the hat was incorrectly classified as partly belonging to the background. While the results using a *weight* of 0.4 have more clusters than what we would expect, these results are accepted as the best for each of the four combinations of cluster centre and distance type as the regions with the hat and feathers are handled much better in these results. Each of these four results were very similar.

The image *molecule* is the type of image which has very obvious segments which can be noticed by simply looking at the original image. Using a *weight* of 0.3 produced too many clusters, where the regions which should be considered as one segment were in fact shown as two or even three. The mean and Euclidean approach produced the best results, especially with a *weight* of 0.5, for which each segment matches with one colour in the molecule, except for the grey part which was divided into two clusters.

The mean and Euclidean result using a *weight* of 0.4 was also not too bad, the only difference from the 0.5 result being that the green part was also considered to be two separate segments. All of the other approaches having results with a similar number of clusters failed, especially with the blue part which was incorrectly classified as partly or totally belonging to the same cluster as the black part.

For the image *teapot*, a *weight* of 0.3 produced too many clusters with each of the four approaches. Using a *weight* of 0.5 didn't produce enough clusters to adequately represent the image, as parts of the checkerboard were grouped together with the background. For the mean cluster centres and absolute distance approach a *weight* of 0.4 also didn't produce enough clusters to adequately represent the image. The best results were achieved when using a *weight* value of 0.4 with Euclidean distance and either cluster centre representation, which both produced similar results.

For the image *ant*, a *weight* of 0.3 produced too many clusters with each of the four approaches. A *weight* of 0.5 produced too few clusters with the four approaches as the head of the ant was incorrectly classified as belonging to the same cluster as its body. A *weight* of 0.4 produced the best results, especially for the mean and Euclidean and mean and absolute approaches.

For the image *blond*, a *weight* of 0.5 resulted in too few clusters to adequately represent the segments in the image for each of the four approaches. When using a *weight* of 0.4, the mean and Euclidean approach produced the best result as the lips were shown as a separate segment, however, this result failed to distinguish between the hands, face and hair. The best result was achieved using a *weight* of 0.3 with mean cluster centres and absolute distance, as the other results using a *weight* of 0.3 either had too many clusters or not enough.

For the image *jet*, a *weight* of 0.3 produced too many clusters with each of the four

approaches. The best result was achieved with the use of the *weight* of 0.5, especially when using the mean cluster centres and absolute distance approach, as this resulted in the blue of the plane being treated as a separated region from the blue in the mountains.

For the image *mandrill*, a *weight* of 0.3 produced too many clusters with each of the four approaches. When using median cluster centres, a *weight* of 0.4 gave the best results, and when using mean cluster centres a *weight* of 0.5 gave the best results. Each of these results were very similar.

For the image *peppers*, a *weight* of 0.3 produced too many clusters with each of the four approaches. A *weight* of 0.5 generally produced too few clusters as the large green pepper was partly classified as belonging to the red segment, the only exception occurring for the median cluster centres and Euclidean distance approach which obtained its best result for a *weight* of 0.5. The best results for the remaining combinations of cluster centre and distance type were achieved with a *weight* of 0.4. The overall best results were obtained by the mean cluster centres and Euclidean distance and median cluster centres with either distance approaches, as the green peppers looked smoother as most of them had just one colour to represent them.

For the image *mouse*, each of the four approaches produced similar results for a given *weight*. The results show that a *weight* of 0.3 was best. When the *weight* value was 0.4, part of the mouse was classified as belonging to the background segment, although an outline of the legs were shown. The results using a *weight* of 0.5 were much worse as not even the outline of the legs were shown any more.

For the image *rose*, the results obtained when using a *weight* of 0.5 did not adequately show the outline of each layer of petals. When using a *weight* of 0.3, the segmented results look more like the original, however, the results using a *weight* of

0.4 adequately show the petals, so there is no need to prefer the result which has more clusters. The best segmentation using a *weight* of 0.4 was found by the mean and Euclidean, median and Euclidean and median and absolute approaches.

For each combination of cluster centre and distance type, the best segmentation results obtained are shown in Figures 5.1-5.4, where Figure 5.1 shows the results when using mean cluster centres and Euclidean distance, Figure 5.2 shows the results when using mean cluster centres and absolute distance, Figure 5.3 shows the results when using median cluster centres and Euclidean distance and Figure 5.4 shows the results when using median cluster centres and absolute distance, with the *weight* values producing these results indicated in the figures. Overall, it can be noted that a *weight* of 0.4 produced good segmentation results for most of the images. An anomaly occurred with the images *blond* and *mouse*, for which a *weight* of 0.3 was required, and for *jet*, for which a *weight* of 0.5 was required. At the very least, we can restrict the value of the *weight* to be between 0.3 and 0.5. The mean cluster centres and Euclidean distance approach seems to give the superior results for most cases, however, there were some variations, with *blond* and *jet* preferring the mean cluster centres and absolute distance approach.

5.1.3.3 Results Using Context

Each of the results obtained when using context are based only on the mean cluster centres and Euclidean distance approach and the use of the RGB colour space. Also, comparisons made to the results by the original RGB values refer to the results using mean cluster centres and Euclidean distance presented in section 5.1.3.2.

Tables 5.11-5.14 show the number of clusters obtained for various values of *weight* using only the three context components. Table 5.11 shows the number of clusters

when using the *mean* context, Table 5.12 shows the number of clusters obtained when using the *median* context, Table 5.13 shows the number of clusters obtained when using the *weighted* context and Table 5.14 shows the number of clusters obtained when using the *regression* context.

weight	0.2	0.3	0.4	0.5	0.6
balls	34	16	6	6	3
Lenna	59	15	8	5	3
molecule	65	23	11	8	2
teapot	42	19	12	5	5
ant	40	12	7	5	5
blond	72	15	7	4	3
jet	71	26	11	5	4
mandrill	105	33	15	9	7
peppers	130	30	15	6	3
mouse	10	7	5	3	3
rose	33	14	7	5	3

Table 5.11: Number of clusters using *mean* context values.

weight	0.2	0.3	0.4	0.5	0.6
balls	46	17	8	5	4
Lenna	73	23	9	6	3
molecule	63	22	11	7	4
teapot	44	19	13	5	4
ant	56	14	9	6	6
blond	77	30	13	9	4
jet	88	32	10	7	4
mandrill	115	34	12	9	6
peppers	106	37	17	8	6
mouse	11	7	5	2	2
rose	30	15	8	6	3

Table 5.12: Number of clusters using *median* context values.

weight	0.2	0.3	0.4	0.5	0.6
balls	42	17	10	6	3
Lenna	50	14	7	6	5
molecule	61	25	16	8	2
teapot	44	20	9	7	4
ant	30	15	8	5	5
blond	70	16	4	4	3
jet	76	26	9	5	4
mandrill	96	29	16	11	5
peppers	108	29	15	10	3
mouse	9	6	4	3	2
rose	28	14	8	7	3

Table 5.13: Number of clusters using *weighted* context values.

Following is a detailed analysis of the results for each image. The results are summarized in section 5.1.4.

weight	0.2	0.3	0.4	0.5	0.6
balls	60	16	8	7	3
Lenna	67	25	7	6	3
molecule	65	19	9	8	3
teapot	40	18	11	5	4
ant	42	15	7	5	3
blond	67	19	8	8	4
jet	91	20	9	7	5
mandrill	104	33	15	8	4
peppers	96	31	13	6	6
mouse	10	7	4	3	3
rose	33	15	8	5	5

Table 5.14: Number of clusters using *regression* context values.

For the image *balls*, there was generally a big drop in the number of clusters when changing from a *weight* of 0.3 to 0.4, regardless of which context type was used. When the *mean* context was used, the number of clusters for a *weight* of 0.4 was 6 which is too small to adequately represent the different coloured balls. The best visual result was achieved when the *regression* context was used with a *weight* of 0.3, which resulted in 16 clusters, as this result obtained a separate cluster for the red ball, which no other context type achieved. The results when using a *weight* of 0.4 for the *median*, *weighted* and *regression* contexts were also acceptable, however, they resulted in the bottom four balls being considered as one segment. In comparison to the results using the original RGB values, the context results show more smoother clusters.

For the image *Lenna* there was a big drop in the number of clusters when changing from a *weight* of 0.3 to 0.4, regardless of which context type was used. When using the *median* and *regression* contexts, the number of clusters obtained for a *weight* of 0.3 was too high, while when using the *mean* and *weighted* contexts, the results were acceptable. However, we take the results for all the contexts with a *weight* of 0.4 as optimal as the number of clusters is at an acceptable level. In comparison to the results using the original RGB values, the context results were more smoother, and looked a little blurry. However, the context results correctly classified the feathers as

separate regions even though the number of clusters were smaller. The overall best result was obtained when using the *mean* context as the hat was more clearly visible and the contours of the face showed up better.

For each of the four context types, the best segmentation results for the image *molecule* occurred when a *weight* of 0.5 was used. The best overall result occurred when using the *median* context, although the *regression* context also produced a very similar result, except for a slight blurring effect around the edges of the molecule. These results were very similar to the best result produced when using the original RGB values.

For the image *teapot*, the best results for each of the four context types was achieved when a *weight* of 0.4 was used. The overall best result was achieved when using the *weighted* context which produced a very similar result to that produced by using only the original RGB values, except that in the region containing the reflection of the teapot, there was a blue outline of the checkerboard squares. The result by the other three context types were all very similar.

For the image *ant*, the best results for each of the four context types was achieved when using a *weight* of 0.4. The overall best results were achieved when using the *median* and *weighted* context types as the bottom of the mouse was more clearly shown. In comparison to the results using the original RGB values, the context results had more smooth looking segments.

For the image *blond*, using a *weight* of 0.3 generally gave too many clusters. When using a *weight* of 0.4, the best result was achieved by the *median* context which clearly showed the lips as a separate segment and had more shades in the face and hair regions. The results by the other three context types using a *weight* of 0.4 were generally poor and did not contain enough clusters. The result when using a *weight*

of 0.5 with the *median* context was also reasonable, except that the lips were only partially shown as a separate segment. This result was much better than the results using only the original RGB values, as there were more shades in the hair and face, even though there was a smaller number or clusters.

For the image *jet*, the best results were obtained when using a *weight* of 0.4 for each of the four context types. The overall best result was obtained when using the *regression* context, as the F-16 symbol on the plane was clearer. In comparison to the results using the original RGB values, the context results had a tendency to look blurry, especially for the *mean* and *weighted* contexts. Also, the context results were generally more smoother looking, especially for the *mean*, *median* and *weighted* contexts.

For the image *mandrill*, a *weight* of 0.4 gave the best segmentation for each of the four context types. The best overall result occurred when using the *mean* context which correctly assigned the eye pixels to one segment. The *weighted* context also performed well in that area. In comparison to the results using the original RGB values, the context results appeared a little blurred.

For the image *peppers*, the results when using a *weight* of 0.4 generally produced too many clusters. When using a *weight* of 0.5, the best result was obtained when using the *weighted* context, which had a slightly higher number of clusters than the results for the other context types, and consequently didn't assign part of the green pepper in the front the same colour as the red peppers. In comparison to the results by the original RGB values, the context results had more smoother looking segments.

For the image *mouse*, the results using the *median* and *regression* contexts with a *weight* of 0.2 showed the nose as a separate segment in pink. However, the results when using a *weight* of 0.3 were also acceptable, although the nose was not separate

in these images. The best result when using a *weight* of 0.3 was for the *regression* context as the ears showed up better. In comparison to the results using only the original RGB values, the context results had more smoother segments.

For the image *rose*, the segmented results when using a *weight* of 0.3 for each of the four context types were very similar. In each of these results, the number of clusters was a little high at 14 or 15. Therefore, the best *weight* for this image was 0.4. The overall best result occurred when using the *median* context as the layers of the petals were more visible. The *regression* context result was also acceptable. In comparison to the results using the original RGB values, the context results showed more smoother clusters. The overall best result using the original RGB values showed some white colour in the rose region, especially along the edges of the petals and this did not occur in the context results.

For each context type, the best segmentation results obtained when using only the context components are shown in Figures 5.5-5.8, where Figure 5.5 shows the results when using the *mean* context, Figure 5.6 shows the results when using the *median* context, Figure 5.7 shows the results when using the *weighted* context and Figure 5.8 shows the results when using the *regression* context, with the *weight* values producing these results indicated in the figures.

Generally the context results tended to be slightly blurred and also contained more smooth looking segments. This is not surprising since when we apply the context operation to the original image, the resulting image is a slightly blurred version of the original since we are using the eight neighbours in calculating new colour values. Overall, the *median* and *regression* contexts performed well with the best results occurring for four images each. The *weighted* context only produced the best result for three images and the *mean* context for two images. It is very difficult to prefer

one context type over the others as there is a lot a variation in the results.

Tables 5.15-5.18 show the number of clusters obtained for various values of *weight* using the original colour values and the three context components. Table 5.15 shows the number of clusters when using the original and *mean* context values, Table 5.16 shows the number of clusters obtained when using the original and *median* context values, Table 5.17 shows the number of clusters obtained when using the original and *weighted* context values and Table 5.18 shows the number of clusters obtained when using the original and *regression* context values.

weight	0.2	0.3	0.4	0.5	0.6
balls	177	44	20	6	4
Lenna	289	81	26	13	4
molecule	123	39	19	7	6
teapot	133	39	22	6	5
ant	166	39	18	11	7
blond	278	59	14	8	4
jet	284	84	29	11	9
mandrill	827	161	38	10	5
peppers	387	75	28	17	8
mouse	35	8	4	3	2
rose	92	24	10	7	3

Table 5.15: Number of clusters using original and *mean* context values.

weight	0.2	0.3	0.4	0.5	0.6
balls	191	55	26	6	4
Lenna	340	82	24	12	4
molecule	110	46	16	10	5
teapot	144	46	26	8	5
ant	227	54	20	8	7
blond	329	73	16	8	5
jet	335	92	34	17	5
mandrill	1078	182	43	16	6
peppers	422	111	45	18	8
mouse	25	9	5	4	2
rose	101	38	16	6	3

Table 5.16: Number of clusters using original and *median* context values.

From Tables 5.15-5.18 we can see that the number of clusters is very high for the *weight* values 0.2 and 0.3, and even for a *weight* of 0.4 for many images. There also tends to be a big drop in the number of clusters between certain consecutive *weight* values, where the particular *weight* value resulting in this is different from image to

weight	0.2	0.3	0.4	0.5	0.6
balls	163	44	13	6	4
Lenna	289	85	33	11	8
molecule	118	48	18	7	5
teapot	123	40	20	8	5
ant	169	42	12	9	8
blond	285	61	19	11	6
jet	288	81	36	14	8
mandrill	879	148	39	17	10
peppers	402	84	41	19	8
mouse	29	9	4	3	3
rose	96	29	9	7	3

Table 5.17: Number of clusters using original and *weighted* context values.

weight	0.2	0.3	0.4	0.5	0.6
balls	227	72	21	6	3
Lenna	308	81	30	11	8
molecule	123	38	19	10	4
teapot	123	39	17	6	5
ant	186	36	22	8	4
blond	312	50	17	10	3
jet	322	101	24	18	6
mandrill	891	176	43	15	8
peppers	451	86	35	20	7
mouse	27	7	5	3	3
rose	81	27	9	6	3

Table 5.18: Number of clusters using original and *regression* context values.

image. Following is a detailed analysis of the results for each image. The results are summarized in section 5.1.4.

For the image *balls*, there was generally a big drop in the number of clusters when changing from a *weight* of 0.4 to 0.5, except for the *weighted* context which produced 13 clusters when a *weight* of 0.4 was used. When using a *weight* of 0.5, the number of clusters for each of the context types was 6, which is too small to adequately represent the different coloured balls. The best visual result was achieved when the *regression* context was used with a *weight* of 0.4 which resulted in 21 clusters as this result obtained a separate cluster for the red ball, which no other context type achieved. Generally, the results when using a *weight* of 0.4 contained too many clusters and the results when using a *weight* of 0.5 produced too few clusters to adequately represent the different coloured balls, and the bottom four balls were considered as belonging

to one segment. In comparison to the results using only the original RGB values, the context results usually required a larger number of clusters to achieve the same results visually. For example, the result for the *weighted* context with a *weight* of 0.4 gave 13 clusters, and the result using only the original RGB values resulted in 11 clusters for the same *weight*. In this instance the result using the original RGB values is preferred because it correctly treated the red ball as a separate cluster, unlike the *weighted* context result which treated all four of the lower balls as belonging to the same cluster.

For the image *Lenna* the best results for each of the four context types were achieved when a *weight* of 0.5 was used. The best overall result occurred when using the *median* context as this result showed more facial features, such as the lips and nose, better. In comparison to the results using only the original RGB values, the context results contained more smoother looking regions, especially in the background area. The only problem with this is that there was a tendency for parts of the hat to be considered the same as the background.

For the image *molecule*, the best result was obtained when using the *median* context type with the *weight* set to 0.5. From a visual point of view, this result was virtually identical to the result using only the original RGB values. The *regression* context result with the *weight* set to 0.5 was also acceptable and very similar to these two results. The *mean* and *weighted* contexts performed poorly as when using a *weight* of 0.4 too many clusters were produced and when using a *weight* of 0.5 too few clusters were produced.

For the image *teapot*, there was a big drop in the number of clusters between the *weight* values 0.4 and 0.5. The results when using a *weight* of 0.5 didn't contain enough clusters as the dark checks and the shadow of the teapot were classified together with

the background. However, the result using the *mean* context was the best out of these results as the teapot was treated as one segment. For the results using a *weight* of 0.4, the *regression* context was the best simply because it was a good result and had the smallest number of clusters at 17. The best result using only the original RGB values was better since it was a good result and only contained 11 clusters.

For the image *ant*, the best results for each of the four context types were achieved when using a *weight* of 0.5. The overall best result was achieved when using the *mean* context as the head of the ant was grey instead of the other results for which it was mostly green. The best result using only the original RGB values was better since the result using the *mean* context classified some of the mouse as black.

For the image *blond*, when using a *weight* of 0.4, the *weighted* and *regression* context results were the best as they had more than one shade for the face and hair, although these results gave 19 and 17 clusters, respectively. The best result when using a *weight* of 0.5 was achieved when using the *median* context as the lips were clearly visible and the outline of the hands were visible. Generally, this method performed poorly for this image as the hair, face and hands were assigned to the same region unless a very high number of clusters was selected.

For the image *jet*, there was generally too many clusters produced when using a *weight* of 0.5, especially when using the *median* and *regression* contexts. The best result was obtained when using the *weighted* context with a *weight* of 0.6 as the F-16 symbol on the plane was clearer. This result was very similar to that produced when using only the original RGB values.

For the image *mandrill*, a *weight* of 0.5 gave the best segmentation for each of the four context types. The best overall result occurred when using the *regression* context which contained 15 clusters, and while the *mean* context result only contained

10 clusters, this result was poor as it failed to distinguish between the eyes and nose regions. This result was very similar to the best result when using only the original RGB values.

For the image *peppers*, the results when using a *weight* of 0.5 generally produced too many clusters. When using a *weight* of 0.6, the best result was obtained when using the *regression* context, which had one fewer clusters than the results for the other context types. This result did not assign part of the green pepper in the front the same colour as the red peppers, as the other context types tended to do. In comparison to the results by the original RGB values, the context results had more smoother looking segments.

For the image *mouse*, each of the four context types achieved their best result when using a *weight* of 0.3. The overall best result occurred when using the *regression* context as in the other results part of the mouse was incorrectly assigned to the background, and while the *weighted* context result looked very similar to the *regression* context result, the *regression* context result had two less clusters. The results were very similar in appearance to those using only the original RGB values.

For the image *rose*, the segmented results when using a *weight* of 0.4 for each of the four context types were the best. The overall best results occurred when using the *weighted* and *regression* contexts which both had 9 clusters. These results were very similar to the best result using only the original RGB values, except that there were some white areas in the rose when using only the original values which did not occur in the results using context.

For each context type, the best segmentation results obtained when using the original colour values together with the context values are shown in Figures 5.9-5.12, where Figure 5.9 shows the results when using the original and *mean* context values,

Figure 5.10 shows the results when using the original and *median* context values, Figure 5.11 shows the results when using the original and *weighted* context values and Figure 5.12 shows the results when using the original and *regression* context values, with the *weight* values producing these results indicated in the figures.

Generally, the *weight* values required to obtain the correct level of segmentation when using both the original and context values were higher than when only using the original RGB values or only using the context values, as *weight* values ranging from 0.3-0.5 and sometimes even 0.6 were optimal. Overall the *regression* context performed well with good results occurring for seven out of the eleven natural images. The *median* and *weighted* contexts only produced the best results for three images, and the *mean* context only for one. The *regression* context appears to be the superior context type when used in combination with the original RGB values.

5.1.3.4 Results Using Different Colour Spaces

Each of the results obtained when using different colour spaces are based only on the mean cluster centres and Euclidean distance approach. The results using the RGB colour space with mean cluster centres and Euclidean distance were presented in section 5.1.3.2.

Tables 5.19-5.21 show the number of clusters obtained for various values of *weight* using the XYZ, KL and HSV colour spaces, respectively. Following is a detailed analysis of the results for each image. The results are summarized in section 5.1.4.

For the image *balls*, a *weight* of 0.4 produced the best segmentation for all of the colour spaces used, including RGB. Overall though, the RGB and KL results were the best, both of which were very similar from a visual point of view. The results by the HSV colour space tended to appear noisy, especially in the regions where the

weight	0.2	0.3	0.4	0.5	0.6
balls	31	16	9	6	4
Lenna	27	19	8	7	5
molecule	33	15	8	6	4
teapot	31	16	6	5	5
ant	29	12	8	6	4
blond	27	11	7	5	3
jet	36	16	10	3	3
mandrill	55	25	11	8	5
peppers	38	12	8	5	5
mouse	9	6	4	3	3
rose	16	7	5	3	3

Table 5.19: Number of clusters using XYZ colour space.

weight	0.2	0.3	0.4	0.5	0.6
balls	48	23	11	6	4
Lenna	57	23	15	9	4
molecule	41	17	10	8	5
teapot	25	13	10	3	2
ant	47	23	8	6	6
blond	46	19	9	5	4
jet	52	27	14	8	5
mandrill	99	40	20	12	6
peppers	59	22	10	8	7
mouse	15	6	5	3	3
rose	27	12	8	6	3

Table 5.20: Number of clusters using KL colour space.

shadows of the balls appear and also for part of the pink ball. The XYZ result using a *weight* of 0.4 had a tendency to group the four dark balls together, while the RGB result with the same *weight* was able to separate the red ball from the other dark balls better.

The best results for the image *Lenna* were produced when a *weight* of 0.4 was used for both the RGB and KL colour spaces, which were visually very similar. The results

weight	0.2	0.3	0.4	0.5	0.6
balls	68	36	16	11	6
Lenna	33	15	8	3	2
molecule	55	30	15	7	6
teapot	52	24	16	9	6
ant	71	33	17	10	8
blond	32	13	8	6	3
jet	60	27	12	8	5
mandrill	110	41	21	12	7
peppers	43	19	12	7	4
mouse	37	12	7	5	4
rose	32	14	10	7	5

Table 5.21: Number of clusters using HSV colour space.

when using a *weight* of 0.5 were also similar for both of these colour spaces as well as the XYZ colour space. The only problem with these results is that the hat was incorrectly classified as partly belonging to the background. The results by using the HSV colour space were noisy, especially in the face and hat areas.

The best results for the image *molecule* were produced when using the RGB and KL colour spaces with a *weight* of 0.5, when 8 clusters were obtained. The result for the HSV colour space incorrectly considered two separate parts of the molecule to belong together, as did the XYZ result with an equivalent number of clusters.

For the image *teapot*, the best result was obtained when using the RGB colour space with a *weight* of 0.4. For this image the HSV colour space also did a reasonable job when a *weight* of 0.4 was used except that the shadow of the teapot on the checkerboard was considered to be one region, not showing the individual squares of the board. Also, when a *weight* of 0.5 was used, the result by the HSV colour space assigned all of the checkerboard to one cluster only, except for the region which includes the reflection of the teapot. The XYZ and KL results using a *weight* of 0.3 were also good, however, these results contained more clusters than the RGB result, and the XYZ result produced many clusters for the teapot, whereas the RGB result had mostly just one cluster to represent the teapot.

For the image *ant*, the best results were produced when using the XYZ or KL colour spaces with a *weight* of 0.4. Both of these results were very similar visually. The result by the RGB colour space with a *weight* of 0.4 was also similar, except that it contained a slightly higher number of clusters. The results by the HSV colour space looked noisy.

The best result for the image *blond* was produced by the XYZ colour space when a *weight* of 0.3 was used. For all of the other colour spaces, results with a similar

number of clusters failed to distinguish between the skin and hair regions.

For the image *jet*, the best results were obtained when using the RGB or KL colour spaces using a *weight* of 0.5. The result by the XYZ colour space using a *weight* of 0.4 was also quite good except that the red part of the tail of the plane appeared noisy. The results by the HSV colour space were a little noisy, especially in the snow regions.

For the image *mandrill*, a *weight* of 0.5 produced the best results for the RGB and KL colour spaces, and a *weight* of 0.4 produced the best result when using the XYZ colour space. All of these results were very similar except that when the RGB colour space was used, the eyes were classified in a separate segment from the nose, unlike the other two results, and the XYZ result produced red spots in the fur region. The result when using the HSV colour space with a comparable number of clusters was very poor.

For the image *peppers*, the best results were obtained when using the RGB or KL colour spaces with a *weight* of 0.4, for which the results were visually very similar. The result by the HSV colour space with a *weight* of 0.4 was reasonable, although the red peppers tended to be a little noisy. The XYZ colour space produced its best result using a *weight* of 0.3, however, this result considered part of the big green pepper in the front to belong to the same region as the red peppers.

For the image *mouse*, each of the RGB, XYZ and KL colour spaces produced the best results, all of which were visually similar, when a *weight* of 0.3 was used. The HSV results looked very noisy, especially on the right side of the mouse.

For the image *rose*, the RGB and KL colour spaces produced the best results when a *weight* of 0.4 was used. Both of these results were visually very similar. The XYZ colour space resulted in the background being classified as part of the dark shade in

the rose, and the HSV colour space resulted in most of the rose being one cluster, loosing detail in the layers of the petals, even though it had more clusters.

For each of the colour spaces used, the best segmentation results obtained are shown in Figures 5.13-5.15, where Figure 5.13 shows the results when using the XYZ colour space, Figure 5.14 shows the results when using the KL colour space and Figure 5.15 shows the results when using the HSV colour space, with the *weight* values producing these results indicated in the figures.

Overall, the RGB and KL colour spaces produced very similar results from a segmentation point of view. The HSV colour space in general performed very poorly and the XYZ colour space was often outperformed by the RGB and KL colour spaces. Therefore, there seems to be no clear advantage in choosing a different colour space other than RGB as there is no real benefit in doing so.

5.1.4 Summary of Results

Tables 5.22-5.25 show the *weight* values giving the best segmentations for each of the natural images for the different cluster centre and distance types, the different context types, when used separately and together with the original RGB values, and the different colour spaces, respectively. The result which was visually the best for each image is indicated in bold. The results produced by the *weight* values shown in Tables 5.22-5.25 match with the results shown in Figures 5.1-5.15.

For this new method of determining the number of clusters, a *weight* of about 0.4 seems to give the optimal results from the segmentation point of view. Therefore, if the *weight* is fixed at the value 0.4, we have in fact achieved our aim of a totally automated segmentation technique which also determines the number of clusters present in the image. However, as seen with results for the natural images, some

image	mean & Euc.	mean & abs.	med. & Euc.	med. & abs.
balls	**0.4**	0.4	0.4	0.3
Lenna	**0.4**	**0.4**	**0.4**	**0.4**
molecule	**0.5**	0.4	0.5	0.3
teapot	**0.4**	0.3	**0.4**	0.4
ant	**0.4**	**0.4**	0.4	0.4
blond	0.3	**0.3**	0.3	0.3
jet	0.5	**0.5**	0.5	0.5
mandrill	**0.5**	**0.5**	**0.4**	**0.4**
peppers	**0.4**	0.4	**0.5**	**0.4**
mouse	**0.3**	**0.3**	**0.3**	**0.3**
rose	**0.4**	0.3	**0.4**	**0.4**

Table 5.22: *weight* values producing best results for all four combinations of cluster centre and distance type.

image	*mean*	*median*	*weighted*	*regression*
balls	0.3	0.3	0.3	**0.3**
Lenna	**0.4**	0.4	0.4	0.4
molecule	0.5	**0.5**	0.5	0.5
teapot	0.4	0.4	**0.4**	0.4
ant	0.4	**0.4**	**0.4**	0.4
blond	0.3	**0.4**	0.3	0.3
jet	0.4	0.4	0.4	**0.4**
mandrill	**0.4**	0.4	0.4	0.4
peppers	0.5	0.5	**0.5**	0.5
mouse	0.3	0.3	0.3	**0.3**
rose	0.4	**0.4**	0.4	0.4

Table 5.23: *weight* values producing best results when using only context values.

images require a smaller *weight*, while others require a larger *weight*, depending on which combination of cluster centre and distance type was used. Therefore, 0.4 can be used as a good starting point, and the option of being able to alter this *weight* should be available if necessary. The optimal *weight* value is generally in the range 0.3-0.5.

The *regression* context type is performing quite well for most of the images, espe-

image	*mean*	*median*	*weighted*	*regression*
balls	0.4	0.4	0.4	**0.4**
Lenna	0.5	**0.5**	0.5	0.5
molecule	0.4	**0.5**	0.4	0.5
teapot	0.4	0.4	0.4	**0.4**
ant	**0.5**	0.5	0.5	0.5
blond	0.4	0.4	**0.4**	**0.4**
jet	0.6	0.6	**0.6**	0.6
mandrill	0.5	0.5	0.5	**0.5**
peppers	0.6	0.6	0.6	**0.6**
mouse	0.3	0.3	0.3	**0.3**
rose	0.4	0.4	**0.4**	**0.4**

Table 5.24: *weight* values producing best results when using original and context values.

image	RGB	XYZ	KL	HSV
balls	**0.4**	0.4	**0.4**	0.4
Lenna	**0.4**	0.4	**0.4**	0.3
molecule	**0.5**	0.3	**0.5**	0.4
teapot	**0.4**	0.3	0.3	0.4
ant	0.4	**0.4**	**0.4**	0.4
blond	0.3	**0.3**	0.3	0.3
jet	**0.5**	0.4	**0.5**	0.5
mandrill	**0.5**	0.4	0.5	0.4
peppers	**0.4**	0.3	**0.4**	0.4
mouse	**0.3**	**0.3**	**0.3**	0.3
rose	**0.4**	0.2	**0.4**	0.3

Table 5.25: *weight* values producing best results for different colour spaces.

cially when both the original and context values are taken into account to perform the segmentation. When using both the original and context values a higher *weight* value is needed to achieve the correct level of segmentation. When using the context values, especially when on its own, the segmentation results look smoother and the regions more connected, which is a clear advantage over the results using only the original RGB values. The problem that can occur, especially when the background and foreground colours are similar, is that parts of the objects get grouped together with the background.

The results by different colour spaces indicate that the RGB colour space performs quite well and there is no real benefit in selecting a different colour space in which to perform the segmentation. The visual results indicated that the RGB and KL colour spaces perform very similarly, and the HSV colour space tends to result in the segmented image looking very noisy, while the XYZ colour space is often outperformed by the RGB and KL colour spaces.

One major limitation of this method is the computational time. If the number of initial colours in the image is too large, having to calculate the inter-cluster distance between every pair of clusters becomes time consuming. However, this method is very simple and is not necessarily specific to colour images. It could also be used for other types of images, such as medical images or multispectral images.

5.2 Merging Method 2

The method presented in the previous section has a serious limitation. Since during every iteration only two clusters are merged, the process can be very time consuming, especially if the number of colours present in the image is very large. The process is so slow due to the fact that the inter-cluster distances have to be calculated between every pair of colours or clusters. Considerable speed up can be achieved if we allow more than just two clusters to be merged during any one iteration. In this section a technique which does exactly that will be presented.

5.2.1 Description of Method

The main aim of this method is to automatically determine the number of clusters present in the image, and the process should be totally automated, with no user input required, to remove any bias in the results.

This new method which has been developed is based on the principle that we would like colours which are "close" to one another to be grouped in the same cluster, and we would also like the clusters to be well separated. It is mainly an iterative technique, but may be thought of as a two stage process. The first stage requires one pass through the image to collect information about how many actual colours, or red, green, blue combinations, are present in the image. At this time, information about the number of pixels in the image having a certain colour are also obtained. This in fact gives us what may be thought of as initial clusters. This is in fact the same as the first stage in the method presented in section 5.1.

The second stage, which is the iterative stage, is where this method differs from the method presented in section 5.1. In this stage, the minimum inter-cluster distance is calculated for each cluster and the other cluster with which it was produced is

recorded. During the first iteration, a threshold is determined which will be used to decide whether a cluster should be merged or not. The threshold can be calculated in two ways, referred to as the *maximum* method and the *mean* method, which will be described later. A cluster is merged with its closest cluster if the inter-cluster distance between the clusters is less than or equal to the threshold value determined in the first iteration, and after all the merges are computed, new cluster centres are calculated based on the new clustering which has been formed. This process continues until no more merging is possible, indicated by every inter-cluster distance being greater than the threshold, and at this time the final number of clusters is recorded. Note that the threshold is selected once during the first iteration, and remains constant during the program, as opposed to the previous method which allowed for the threshold to change from iteration to iteration.

To determine the "closeness" of two clusters, we can use either the Euclidean distance measure to calculate the inter-cluster distance, as for merging method 1, which is shown in equation 5.1, or we can use the absolute distance measure, which is shown in equation 5.2. Both mean and median cluster centres were also implemented for this method.

When we merge a cluster C_i with its closest cluster $closest_i$, it may not be just those two clusters which are being merged. This is due to the fact that just because cluster $closest_i$ may be closest to cluster C_i, this does not mean that cluster C_i is closest to cluster $closest_i$. Therefore, cluster $closest_i$ may have already been merged with another cluster, so we simply merge C_i with the cluster to which $closest_i$ now belongs. Consequently, after the merging is completed the new clusters may have been formed by more than two of the original clusters, so instead of calculating a new cluster centre after each merge takes place, we simply make a note of which

clusters should be grouped together based on the minimum inter-cluster distance information. After this is known, we then calculate the new cluster centres as either the mean or median of all the pixels representing the colours which have been assigned to each cluster.

The entire process can be represented by the following algorithm:

loop

for every cluster C_i

set $minimum_i$ to be the minimum inter-cluster distance

set $closest_i$ to be the cluster with which the minimum occurred

end for

if it is the first iteration

set $thresh$ to be the desired threshold value

end if

for every cluster C_i

if $(minimum_i \leq thresh)$

merge clusters C_i and $closest_i$

end if

end for

if no merging took place during this iteration

exit loop

else

calculate new cluster centres based on the new clustering

end if

end loop

5.2.1.1 Threshold Calculation by *maximum* method

The threshold can be calculated as the maximum of all the minimum inter-cluster distances from the first iteration, as we make the assumption that we want to merge all the clusters to their closest cluster in the first iteration and consequently in the subsequent iterations we must also be allowed to merge clusters whose minimum inter-cluster distance is less than or equal to the maximum value from the first iteration, so it is reasonable for the maximum of the minimum inter-cluster distances to be the threshold.

5.2.1.2 Threshold Calculation by *mean* method

The second method to calculate the threshold value is to combine the mean and standard deviation values of the minimum inter-cluster distances, and then calculate the threshold as $mean + 4 \times standard\ deviation$. The multiplier of 4 was chosen based on both theoretical reason as well as experimental results. The theoretical basis comes from Chebyshev's inequality [Rao73] which says that the interval ($mean - 4 \times standard\ deviation$, $mean + 4 \times standard\ deviation$) covers more than 94% of the inter-cluster distance values. The other distances can therefore be treated as outliers, making it possible to leave these outlying colours as separate clusters, which may give some advantage over simply taking the maximum value as the threshold. In experimental results, reported in a later section, the multiplier of 4 has been used because, compared to other multipliers, it gives a reasonable number of clusters for different images.

5.2.2 Results

5.2.2.1 Synthetic Images

To test this new method, the synthetic images shown in Figure 1.1 were first used. Results were obtained based on the use of the RGB colour space. Note that the images *four* and *six* both contain an outlier segment consisting of only one colour value, which corresponds to the red region in the image *four* and the white region in the image *six*. The remaining regions in both of these images have varying colour values based on a Gaussian distribution.

As expected, for the images *four* and *six*, the threshold selected from the *maximum* method was too big due to the existence of an outlier colour, resulting in an inadequate number of clusters to represent these images, regardless of which cluster centre representation or which distance type was used. For most combinations of cluster centre and distance type, the resulting number of clusters was 3, with the only exception being that for the image *six* when using mean cluster centres and absolute distance, 4 clusters were obtained. The results for *two*, *eight*, *ten* and *fifteen* using the *maximum* method were all correct for all combinations of cluster centre and distance type.

For the two synthetic images containing an outlier colour, Table 5.26 shows the *maximum* threshold as well as the mean and standard deviation values which are used to calculate the *mean* threshold.

image	Distance	Maximum	Mean	Std. Dev.
four	Euclidean	236.129202	1.033498	2.635183
	absolute	247.000000	1.039056	2.758132
six	Euclidean	236.105908	1.026206	2.204164
	absolute	246.000000	1.034177	2.299063

Table 5.26: Maximum, mean and standard deviation values.

The *maximum* threshold values shown in Table 5.26 are very big, which is why the

merging process continued further than was required. The *mean* threshold is based on the equation $mean + weight \times standard\ deviation$. Different values for the *weight* were chosen ranging from 1, 2,. . ., 10.

Table 5.27 shows the *weight* values which produced the correct segmentation. The lowest *weight* values giving correct segmentation range from 4 to 8 depending on which combination of cluster centre and distance type was being used. However, considering that natural colour images will in general contain greater variability, we will take the lowest value of 4 and apply this to the natural colour images.

cluster centre	mean	mean	median	median
distance	Euclidean	absolute	Euclidean	absolute
four	4-10	5-10	4-10	6-10
six	4-10	7-10	5-10	8-10

Table 5.27: *weight* values for *mean* method producing correct segmentation.

5.2.2.2 Natural Images

The natural images used to test this new method are the eleven images shown in Figure 1.2. The results in this section are based on the use of the RGB colour space. The number of starting clusters is the same as the number of colours produced from the first stage, as shown in Table 5.6 earlier. The number of clusters resulting from the *maximum* method are shown in Tables 5.28-5.31. Table 5.28 shows the results using mean cluster centres and Euclidean distance, Table 5.29 shows the results using mean cluster centres and absolute distance, Table 5.30 shows the results using median cluster centres and Euclidean distance and Table 5.31 shows the results using median cluster centres and absolute distance. Figures 5.16-5.19 show the segmented results when using the *maximum* method, corresponding to the number of clusters shown in Tables 5.28-5.31, respectively.

Just by looking at the number of clusters shown in Tables 5.28-5.31, we can see

image	Threshold	No. of clusters
balls	52.306787	8
Lenna	57.271284	4
molecule	79.466974	6
teapot	61.032778	8
ant	69.166466	6
blond	95.331002	1
jet	73.102668	3
mandrill	44.899889	15
peppers	85.510233	2
mouse	22.627417	9
rose	23.664319	22

Table 5.28: Thresholds and number of clusters using the *maximum* method with mean cluster centres and Euclidean distance.

image	Threshold	No. of clusters
balls	64	11
Lenna	72	3
molecule	91	5
teapot	88	8
ant	88	6
blond	160	1
jet	88	5
mandrill	64	12
peppers	100	3
mouse	32	10
rose	36	26

Table 5.29: Thresholds and number of clusters using the *maximum* method with mean cluster centres and absolute distance.

that the resulting number of clusters is too low for *Lenna*, *jet*, *peppers* and especially for *blond*, which was reduced to only one cluster. This conclusion is reinforced by looking at the corresponding segmented results in Figures 5.16-5.19, as there are not enough clusters to adequately represent the whole of these images. The reason that such low numbers of clusters were achieved for these images is that during the first iteration we are not taking into account the possibility of there being some colours which are outliers as we take the maximum of the minimum inter-cluster distances as the threshold, thus ensuring that every initial cluster is merged with at least one other cluster. Also, this maximum distance can be very large compared to the other distances, and further iterations may not produce any minimum inter-cluster distances which are greater than the threshold, which is how the segmented result for *blond* produced only one cluster.

image	Threshold	No. of clusters
balls	52.306787	7
Lenna	57.271284	4
molecule	79.466974	6
teapot	61.032778	7
ant	69.166466	5
blond	95.331002	1
jet	73.102668	4
mandrill	44.899889	13
peppers	85.510233	4
mouse	22.627417	9
rose	23.664319	24

Table 5.30: Thresholds and number of clusters using the *maximum* method with median cluster centres and Euclidean distance.

image	Threshold	No. of clusters
balls	64	8
Lenna	72	5
molecule	91	7
teapot	88	7
ant	88	6
blond	160	1
jet	88	4
mandrill	64	11
peppers	100	4
mouse	32	10
rose	36	18

Table 5.31: Thresholds and number of clusters using the *maximum* method with median cluster centres and absolute distance.

The remainder of the results are quite good, especially for *molecule*, when using Euclidean distance with either cluster centre type, which was assigned only one cluster for every individual part in the molecule, despite the way in which the original image shows slowly varying shades because of the lighting effect. For the results which produced adequate numbers of clusters, the best visual segmentation occurred when mean cluster centres and Euclidean distance were used. However, for *molecule*, median cluster centres and Euclidean distance also produced a good result. Good results for *teapot* were achieved in all approaches except with mean cluster centres and absolute distance. All the results for *mandrill* and *mouse* were very similar, no matter which cluster centre type or distance measure was used, and all the results for *rose* produced too many clusters with the result using median cluster centres and absolute distance being the lowest at 18, however, the results by each of the four

combinations of cluster centre and distance type were visually very similar.

To improve the results for images which contain outlying colours, it is necessary to have a lower threshold value. This is when we would employ the *mean* method. The number of clusters resulting from the *mean* method are shown in Tables 5.32-5.35. Table 5.32 shows the results using mean cluster centres and Euclidean distance, Table 5.33 shows the results using mean cluster centres and absolute distance, Table 5.34 shows the results using median cluster centres and Euclidean distance and Table 5.35 shows the results using median cluster centres and absolute distance. Figures 5.20-5.23 show the segmented results when using the *mean* method, corresponding to the number of clusters shown in Tables 5.32-5.35, respectively. In each of these results, the *weight* used in the calculation of the threshold was 4.

image	Threshold	No. of clusters
balls	44.626385	14
Lenna	40.367123	11
molecule	15.537798	60
teapot	5.119287	380
ant	46.171107	10
blond	42.511440	8
jet	43.110337	10
mandrill	49.655660	11
peppers	49.049102	9
mouse	14.028864	24
rose	18.642888	36

Table 5.32: Thresholds and number of clusters using the *mean* method with mean cluster centres and Euclidean distance.

image	Threshold	No. of clusters
balls	58.747233	15
Lenna	53.403923	12
molecule	20.875746	65
teapot	7.511094	378
ant	56.821262	12
blond	60.635745	10
jet	54.172568	14
mandrill	62.407173	15
peppers	61.622057	10
mouse	18.626570	23
rose	28.632843	32

Table 5.33: Thresholds and number of clusters using the *mean* method with mean cluster centres and absolute distance.

image	Threshold	No. of clusters
balls	44.626385	13
Lenna	40.367123	12
molecule	15.537798	63
teapot	5.119287	358
ant	46.171107	9
blond	42.511440	7
jet	43.110337	12
mandrill	49.655660	9
peppers	49.049102	8
mouse	14.028864	21
rose	18.642888	34

Table 5.34: Thresholds and number of clusters using the *mean* method with median cluster centres and Euclidean distance.

image	Threshold	No. of clusters
balls	58.747233	13
Lenna	53.403923	11
molecule	20.875746	66
teapot	7.511094	384
ant	56.821262	12
blond	60.635745	10
jet	54.172568	14
mandrill	62.407173	13
peppers	61.622057	12
mouse	18.626570	20
rose	28.632843	28

Table 5.35: Thresholds and number of clusters using the *mean* method with median cluster centres and absolute distance.

From Tables 5.32-5.35 we can see that the number of clusters for *Lenna*, *blond*, *jet* and *peppers* is much improved, which can also be seen in the segmented results shown in Figures 5.20-5.23. In fact, most of the results are very good, except for *molecule*, *teapot*, *mouse* and *rose* which now have too many clusters. In general, the results for the *mean* method have more clusters than the results for the *maximum* method, with an exception being found for *mandrill* which actually achieved a lower number of clusters for the *mean* method when using Euclidean distance with either cluster centre type. This was due to the fact that the threshold calculated for the *mean* method was higher than the threshold calculated for the *maximum* method in these cases. For the images producing adequate numbers of clusters for the *mean* method, most obtained the best visually segmented images using the mean cluster centres and Euclidean distance approach. The main exception was for the image *blond*, which

produced the best result when median cluster centres and absolute distance was used as the lips were correctly classified as a separate cluster. Also, the results for *jet* indicate that all of the approaches except mean cluster centres with Euclidean distance produced good results. The results for *ant*, *mandrill* and *peppers* produced similar results regardless of the cluster centre or distance type used.

Some images produce better results for the *maximum* method, namely *molecule*, *teapot*, *mouse* and *rose*, while other images produce better results for the *mean* method, such as *Lenna*, *blond*, *jet* and *peppers*. A set of ground rules are necessary to enable the decision about whether an outlying colour exists to be made automatically, thus leading to a decision about how to calculate the threshold.

5.2.2.3 Results Using Context

Each of the results when using context are based only on the mean cluster centres and Euclidean distance approach and the use of the RGB colour space. The number of clusters obtained by the *maximum* method using only the context values are shown in Tables 5.36-5.39. Table 5.36 shows the results when using only *mean* context values, Table 5.37 shows the results when using only *median* context values, Table 5.38 shows the results when using only *weighted* context values, and Table 5.39 shows the results when using only *regression* context values.

image	Threshold	No. of clusters
balls	15.684387	62
Lenna	13.453624	46
molecule	15.165751	86
teapot	9.848858	190
ant	13.341664	73
blond	13.964240	38
jet	13.341664	68
mandrill	14.560220	146
peppers	23.021729	48
mouse	7.071068	80
rose	6.782330	237

Table 5.36: Thresholds and number of clusters using the *maximum* method with only *mean* context values.

image	Threshold	No. of clusters
balls	32.062439	19
Lenna	15.231546	62
molecule	21.840330	46
teapot	15.652476	81
ant	22.891046	31
blond	12.649111	66
jet	28.284271	14
mandrill	17.549929	133
peppers	31.685959	25
mouse	9.848858	35
rose	9.165151	128

Table 5.37: Thresholds and number of clusters using the *maximum* method with only *median* context values.

image	Threshold	No. of clusters
balls	14.247807	79
Lenna	14.177447	46
molecule	20.024984	50
teapot	10.770330	166
ant	13.747727	73
blond	13.964240	40
jet	19.000000	30
mandrill	13.416408	186
peppers	21.189620	62
mouse	6.480741	98
rose	6.403124	269

Table 5.38: Thresholds and number of clusters using the *maximum* method with only *weighted* context values.

image	Threshold	No. of clusters
balls	20.435154	45
Lenna	20.832341	16
molecule	73.512287	6
teapot	8.578289	200
ant	25.915793	17
blond	56.109681	4
jet	41.937227	7
mandrill	21.231665	44
peppers	45.402937	8
mouse	8.359457	55
rose	9.632931	107

Table 5.39: Thresholds and number of clusters using the *maximum* method with only *regression* context values.

If only the three context components are used, the results show that the number of clusters is too high for all the images. As expected, better results are achieved when the context values are used together with the original values. Also, note that when context values are used, the starting number of colours is considerably larger as the context colour values are calculated from eight possible surrounding values, leading to a wider range of colours for the context image.

The number of clusters obtained using the *maximum* method with original and context values are shown in Tables 5.40-5.43. Table 5.40 shows the results when using original and *mean* context values, Table 5.41 shows the results when using original and *median* context values, Table 5.42 shows the results when using original and *weighted* context values, and Table 5.43 shows the results when using original and *regression* context values.

image	Threshold	No. of clusters
balls	48.805737	24
Lenna	57.584720	9
molecule	120.673112	5
teapot	48.528342	23
ant	97.534609	4
blond	76.216796	4
jet	59.874870	12
mandrill	45.299007	54
peppers	60.605280	15
mouse	35.383612	10
rose	36.565011	25

Table 5.40: Thresholds and number of clusters using the *maximum* method with original and *mean* context values.

Using the *regression* context values together with the original colour values resulted in the most reasonable numbers of clusters for most of the images. The segmented results corresponding to the number of clusters shown in Table 5.43 are shown in Figure 5.24. Even though the *maximum* method is used, there is no longer a problem of the number of clusters being too low for *blond*, *jet* and *peppers*, and except for *teapot*, *mandrill*, and possibly *peppers*, the number of clusters for each of the images

image	Threshold	No. of clusters
balls	50.119856	29
Lenna	57.271284	10
molecule	131.144958	5
teapot	66.257075	15
ant	89.397987	7
blond	82.462113	4
jet	60.332413	18
mandrill	49.558047	36
peppers	86.533231	12
mouse	37.947332	8
rose	51.730069	11

Table 5.41: Thresholds and number of clusters using the *maximum* method with original and *median* context values.

image	Threshold	No. of clusters
balls	49.335586	28
Lenna	57.497826	8
molecule	118.220134	5
teapot	46.936127	22
ant	98.173316	5
blond	83.414627	4
jet	59.514704	12
mandrill	45.133136	50
peppers	60.778286	15
mouse	34.322005	9
rose	35.805028	23

Table 5.42: Thresholds and number of clusters using the *maximum* method with original and *weighted* context values.

are quite reasonable, which is shown by the segmented results in Figure 5.24. Each of the four context types, when used in combination with the original RGB values, produced too many clusters for *teapot* and *mandrill*, however, for *peppers*, a reasonable result was obtained by using the *median* context together with the original RGB values, which only resulted in 12 clusters.

5.2.2.4 Results Using Different Colour Spaces

Each of the results obtained when using different colour spaces are based only on the mean cluster centres and Euclidean distance approach. The results using the RGB colour space with mean cluster centres and Euclidean distance were presented in section 5.2.2.2.

The number of clusters resulting from the *maximum* method for the various colour

image	Threshold	No. of clusters
balls	59.589804	15
Lenna	57.437002	8
molecule	108.512932	6
teapot	50.307168	26
ant	94.591681	5
blond	85.229103	5
jet	61.950797	9
mandrill	47.466174	40
peppers	52.485197	24
mouse	28.524372	12
rose	43.911998	12

Table 5.43: Thresholds and number of clusters using the *maximum* method with original and *regression* context values.

spaces are shown in Tables 5.44-5.46. Table 5.44 shows the results using the XYZ colour space, Table 5.45 shows the results using the KL colour space and Table 5.46 shows the results using the HSV colour space. Figures 5.25-5.27 show the segmented results when using the *maximum* method, corresponding to the number of clusters shown in Tables 5.44-5.46, respectively.

image	Threshold	No. of clusters
balls	52.664989	6
Lenna	52.253245	3
molecule	81.206405	5
teapot	45.497133	8
ant	63.859646	5
blond	90.920771	1
jet	37.803759	5
mandrill	40.793260	10
peppers	56.279499	5
mouse	12.994727	21
rose	16.702987	21

Table 5.44: Thresholds and number of clusters for *maximum* method using XYZ colour space.

image	Threshold	No. of clusters
balls	52.304367	7
Lenna	57.225141	3
molecule	79.467001	5
teapot	45.394636	8
ant	69.164852	7
blond	95.312345	1
jet	73.144086	3
mandrill	44.889373	15
peppers	85.515890	4
mouse	22.639556	8
rose	23.673111	23

Table 5.45: Thresholds and number of clusters for *maximum* method using KL colour space.

image	Threshold	No. of clusters
balls	0.305479	8
Lenna	0.283352	3
molecule	0.314278	6
teapot	0.332657	5
ant	0.252745	13
blond	0.311991	3
jet	0.232780	6
mandrill	0.208868	16
peppers	0.256507	7
mouse	0.349274	4
rose	0.328343	6

Table 5.46: Thresholds and number of clusters for *maximum* method using HSV colour space.

As for the RGB results, the various colour spaces used also resulted in the number of clusters being too low for *Lenna*, *blond*, *jet* and *peppers*, especially for the XYZ colour space, however, the results for *peppers* were not too bad for the KL and HSV colour spaces, and the result for *jet* using the HSV colour space was also at an acceptable level. This conclusion is reinforced by looking at the corresponding segmented results in Figures 5.25-5.27, as there are not enough clusters to adequately represent the whole of these images, except in the cases mentioned. In these cases, the threshold selected was too high, resulting in too much merging taking place.

The remainder of the results were quite good, by the *maximum* method, except for *molecule* for which the correct segmentation which was found by the RGB colour space could not be found by any of the other colour spaces. For *balls*, the HSV colour space did a good job of separating the balls from the background, but the white ball was completely classified as belonging to the background and the pink ball appeared noisy. The KL result was similar but a little of the shadow was also shown and the pink ball was more smooth in appearance. The XYZ result did not have enough clusters and the red ball was incorrectly classified together with the black ball. The RGB result differed from these in that the shadows of the balls were shown as a separate segment and also an outline of the white ball was shown. For *teapot*, the KL and XYZ results were reasonable, however, the RGB result was the best out of all

the colour spaces used. For *ant*, the HSV result was very noisy, however, the XYZ and KL results looked good. The RGB result was superior as only one colour was assigned to the whole of the toy mouse in this image, and while this was also true of the XYZ result, part of the ant's head in the XYZ result was green. For *mandrill*, the RGB and KL results were very similar and also the best in appearance, while the XYZ result was also reasonable. For *mouse*, the XYZ result contained too many clusters and the HSV result was very bad. The RGB and KL results were both good for this image. For *rose* each of the XYZ, KL and RGB results were very similar, however, even for the *maximum* method there was too many clusters. The HSV result for this image was bad.

The number of clusters resulting from the *mean* method for the various colour spaces are shown in Tables 5.47-5.49. Table 5.47 shows the results using the XYZ colour space, Table 5.48 shows the results using the KL colour space and Table 5.49 shows the results using the HSV colour space. Figures 5.28-5.30 show the segmented results when using the *mean* method, corresponding to the number of clusters shown in Tables 5.47-5.49, respectively.

image	Threshold	No. of clusters
balls	35.634557	12
Lenna	30.199041	13
molecule	12.853505	59
teapot	4.053660	395
ant	34.635461	16
blond	33.184336	8
jet	28.851776	12
mandrill	32.348710	19
peppers	32.562775	10
mouse	9.143055	34
rose	14.188277	29

Table 5.47: Thresholds and number of clusters for *mean* method using XYZ colour space.

From Tables 5.47-5.49 we can see that the number of clusters for *Lenna*, *blond*, *jet* and *peppers* is much improved, which can also be seen in the segmented results shown in Figures 5.28-5.30. Also, for *balls*, the results from the *mean* method can

image	Threshold	No. of clusters
balls	44.626837	13
Lenna	40.360702	10
molecule	15.537836	62
teapot	3.869186	327
ant	46.171149	9
blond	42.517163	8
jet	43.121327	11
mandrill	49.654645	11
peppers	49.048915	8
mouse	14.022953	20
rose	18.643520	36

Table 5.48: Thresholds and number of clusters for *mean* method using KL colour space.

image	Threshold	No. of clusters
balls	0.267399	14
Lenna	0.154775	14
molecule	0.081691	81
teapot	0.036743	329
ant	0.237866	14
blond	0.179741	11
jet	0.172151	12
mandrill	0.201592	18
peppers	0.193718	13
mouse	0.200945	9
rose	0.169463	18

Table 5.49: Thresholds and number of clusters for *mean* method using HSV colour space.

be considered superior to the results by the *maximum* method as the red ball was correctly classified in its own segment, except for the XYZ result. For *Lenna*, the HSV result was very noisy and in the XYZ result the feathers were incorrectly classified. In the KL result a lot of the hat was grouped together with the background, so the RGB result was the overall best. The number of clusters for *blond* was greatly improved by using the *mean* method. However, the results themselves were not very good. For the XYZ result it was hard to distinguish the hair from the background and the HSV result had the same colour for most of the face, hair and hands. The RGB and KL results were similar, however, most of the face and hair was one colour. In general, the methods presented in this chapter are not very good for this image. For *jet*, the HSV result was very noisy, especially in the background region. Overall the KL result was the best as the RGB result did not clearly show the outline of the bottom of the

plane. For *peppers*, the XYZ result combined part of the green pepper at the front with the same segment as the red peppers, while the RGB result was the overall best, and the KL result was also acceptable. The HSV result had a tendency to appear noisy, especially in the regions with the red peppers.

Some images produce better results for the *maximum* method, namely *molecule*, *teapot*, *ant*, *mandrill*, *mouse* and *rose*, while other images produce better results for the *mean* method, such as *Lenna*, *blond*, *jet* and *peppers*. Overall the RGB colour space produces superior results, and in many cases the KL colour space results are similar to these. The HSV results are generally poor, as are the XYZ results.

5.2.3 Summary

For the *maximum* method, good results can be obtained for some images. However, when outlying colours are present in the image, the final number of clusters produced by the *maximum* method is too small. This can be improved by employing the *mean* method so that a smaller threshold value is used, leading to a larger number of clusters. Therefore, in some cases we want to use the *maximum* method, while in other cases we want to use the *mean* method. Consequently, a set of ground rules are required to enable us to decide whether an image contains any outlying colours, and which will ultimately tell us which method should be used for a particular image.

The context results produced much larger numbers of clusters, especially when only the context values were used. Improvement can be gained when both the original and context values are used, especially in the case of using the *regression* context.

The colour space results indicate that the RGB results are generally the best, leading to the conclusion that there is no real benefit in choosing another colour space in which to perform the clustering.

This method is an improvement on the method presented in section 5.1 as the process requires less iterations due to the fact that many clusters can be merged together during a single iteration. This leads to less inter-cluster distances having to be calculated. This is a simple method which has achieved the aim of automatically detecting the number of clusters and also not requiring any user input apart from the image. It is also not restricted to only colour images, and can in fact be extended to cope with any dimensionality of images, such as medical images or multispectral images.

Chapter 6

Splitting Methods Based on the Principle of Cluster Compactness

Up until now the methods presented have concentrated on using the inter-cluster distance measure in combination with a merging process to enable well separated clusters to be formed. In this chapter the opposite approach is used, which is to incorporate the intra-cluster distance and combine this with a splitting process. By using the intra-cluster distance measure, the aim is to make this as small as possible, leading to compact, well formed clusters. The smaller the intra-cluster distance, the more confident we can be that the clusters are in fact well formed clusters. Some simple criteria based on the intra-cluster distance will be presented, which will enable the number of clusters to be determined automatically.

The *K-means* method [TG74] is a very simple technique which enables the sum of the intra-cluster distances to be minimized. This is the criterion which leads to compact, well formed clusters. The major problem with this method is that the number of clusters must be supplied as a parameter. The ISODATA method [TG74] provides an improvement by allowing splitting and merging to change the number of clusters based on certain criteria being met, however, the number of parameters

required is too large and this method is, therefore, very subjective. In this chapter a method will be presented which incorporates a splitting procedure with the *K-means* approach, and which does not rely on so many parameters.

Three stopping rules are presented in this chapter to enable the number of clusters to be determined automatically. The first stopping rule is based on the initial variance of the image, when considered to consist of only one cluster, which is multiplied by some *weight* value to determine a threshold which the intra-cluster variabilities must be smaller than. The second stopping rule is based on the normalized intra-cluster distance and the percentage decrease in intra-cluster distance, and the stopping rule is based on both of these values being below some given thresholds, in which case the algorithm would stop when the error in the segmented image is small enough, indicated by the normalized intra-cluster distance, and when the segmented image is not changing much, which is indicated by the percentage decrease in the intra-cluster distance. The third stopping rule is based solely on the percentage decrease in intra-cluster distance, and in this case a given number of consecutive percentage decrease values are required to be below a threshold. By requiring there to be a sequence of percentage decrease values which are small, we can avoid the problem of stopping too early and are more likely to find the correct segmentation.

6.1 Description of Method

The method for clustering by splitting is based on the principle that we would like each of the pixels belonging to a particular cluster to be as "close" to their cluster centre as possible, which is exactly what the *K-means* procedure aims to optimize. Therefore, we use the *K-means* procedure as a starting point. However, the *K-means* algorithm requires K, the number of clusters, to be supplied as a parameter.

Therefore, we employ a splitting procedure to allow us to increase from K to $K+1$ clusters between consecutive performances of the *K-means* procedure. Since we could continually increase the number of clusters by splitting, we also require some form of stopping rule which will determine the number of clusters present in the image. Three stopping rules have been developed for this, which will be presented in later sections. In this section the general algorithm will be described.

The algorithm begins by forming one cluster containing all of the pixels in the image. Then an iterative process begins where the cluster having maximum variance is determined and this cluster is split into two smaller clusters. Once the cluster is split, we make use of the *K-means* process to repeatedly reassign each of the pixels in the image to their closest cluster centre until no more reassignments can take place. This iterative process continues until the segmented result has reached the optimum number of clusters, indicated by the stopping rule being satisfied.

Since the *K-means* algorithm aims to minimize the average intra-cluster distance, it is most likely that the cluster having maximum variance will be separated by the *K-means* procedure when the number of clusters is increased. Therefore, when we require the number of clusters to be increased, we split the cluster having maximum variance, so the *K-means* procedure is given good starting cluster centres. This new method of clustering by splitting has been implemented by allowing for two different cluster centre representations, namely mean and median. In calculating the variance of a particular cluster, C_i, represented by the cluster centre $\mathbf{z}_i = (R_i, G_i, B_i)$, we use the equations

$$var_{iR} = \frac{1}{N_i} \sum_{\mathbf{x} \in C_i} (x_R - R_i)^2 \tag{6.1}$$

$$var_{iG} = \frac{1}{N_i} \sum_{\mathbf{x} \in C_i} (x_G - G_i)^2 \tag{6.2}$$

$$var_{iB} = \frac{1}{N_i} \sum_{\mathbf{x} \in C_i} (x_B - B_i)^2 \tag{6.3}$$

where N_i is the number of pixels in cluster C_i, and $\mathbf{x}$ is the vector, (x_R, x_G, x_B), containing the red, green and blue values for the pixels. Obviously, when using the mean cluster centre representation, we can obtain the true values of the variances for each cluster, however, when using the median cluster centre representation, the values obtained for the variances will not be the true values, unless the situation arises in which the median values are the same as the mean values. Equations 6.1-6.3 give the variance of each component separately, however, what is required is a single figure by which we would make the comparison with the other variance values to determine the cluster having maximum variance. We can consider just the maximum variance of the three components or we could take the average variance of the three components by adding them up and dividing the sum by 3.

To split a cluster, the first attempt made was to take all three of the red, green and blue colour components into account. In this case, given the cluster centre $\mathbf{z}_i$ shown above, we obtain two new clusters $\mathbf{z}_i' = (R_i', G_i', B_i')$ and $\mathbf{z}_i'' = (R_i'', G_i'', B_i'')$. We split cluster C_i by creating two new values for each component which are centred around the cluster centre values. The general format of this splitting procedure is given by

$$\mathbf{z}_i' = (R_i - a_R, G_i - a_G, B_i - a_B) \tag{6.4}$$

$$\mathbf{z}_i'' = (R_i + a_R, G_i + a_G, B_i + a_B) \tag{6.5}$$

where a_R, a_G and a_B are constants. The values for these constants are determined by taking into account the minimum and maximum values for each colour component occurring in the cluster. For example, the constant a_R will be the value which is half of the smaller of (R_i - min_R) and (max_R - R_i), where min_R is the minimum value for

the red component and max_R is the maximum value for the red component. Similar computations can be carried out for the blue and green components. When we are using the median cluster centre representation, in general we use the same procedure as just described to calculate the new cluster centres. However, the situation may arise in which the cluster centre value is exactly the same as either of the minimum or maximum values. In this case we calculate the new cluster centres as being evenly spaced between the minimum and maximum values, or in other words the new cluster centres are

$$R_i' = min_R + (max_R - min_R) \div 4 \tag{6.6}$$

$$G_i' = min_G + (max_G - min_G) \div 4 \tag{6.7}$$

$$B_i' = min_B + (max_B - min_B) \div 4 \tag{6.8}$$

$$R_i'' = min_R + (max_R - min_R) \times 3 \div 4 \tag{6.9}$$

$$G_i'' = min_G + (max_G - min_G) \times 3 \div 4 \tag{6.10}$$

$$B_i'' = min_B + (max_B - min_B) \times 3 \div 4 \tag{6.11}$$

where min_R, min_G and min_B are the minimum values for the red, green and blue components, respectively, and max_R, max_G and max_B are the maximum pixel values for the red, green and blue values, respectively.

This method of splitting by taking all three components into account proved to be problematic. In some cases when a cluster was split, the reassignment step only assigned pixels to one of these new cluster centres, taking us back to the same number of clusters as before the split. In some other cases the pixel values were only assigned

to one fewer number of clusters than before the split procedure was executed, leading to a smaller number of clusters. This was seen to occur for most of the synthetic images for various combinations of cluster centre and distance type. The only exception was for the image *two* which produced the correct segmentation. This problem led to the conclusion that this method of producing the two new cluster centres was not necessarily going to split the cluster as required.

Due to this problem, the splitting procedure was modified so that only the colour component having maximum variance is modified. For example, if the red component has the maximum variance, then the two new cluster centres would be

$$\mathbf{z}_i{}' = (R_i - a_R, G_i, B_i) \tag{6.12}$$

$$\mathbf{z}_i{}'' = (R_i + a_R, G_i, B_i) \tag{6.13}$$

The constant, a_R is determined by the procedure describe above. This results in the two new cluster centres being well separated, but also still well within the original cluster.

We can use either the Euclidean or absolute distance to determine which cluster centre a particular pixel is closest to. If we wished to calculate the distance of the pixel given by the vector $\mathbf{x}$ and the cluster C_i, both given above, then the measure of "closeness" using Euclidean distance would be

$$d = \sqrt{(x_R - R_i)^2 + (x_G - G_i)^2 + (x_B - B_i)^2} \tag{6.14}$$

and the measure of "closeness" using absolute distance would be given by

$$d = |x_R - R_i| + |x_G - G_i| + |x_B - B_i| \tag{6.15}$$

An important aspect of this algorithm is the number of passes, through the pixels, that are required when reassigning the pixels to their closest cluster centre. Initially the process was implemented by having just one pass through the data during each iteration. However, when executing this for the synthetic images shown in Figure 1.1, it was observed that for the images *four*, *six*, *ten* and *fifteen*, when the correct number of clusters was reached, the pixels belonging to one cluster were not all correctly assigned to the same cluster. Incorrect segmentation was produced for *four* for all combinations of cluster centre and distance type, *six* and *ten* only produced an incorrect segmentation when median cluster centres were used with either distance type, and for the image *fifteen*, an incorrect segmentation was produced only when median cluster centres and absolute distance was used. These problems occurred regardless of whether the average of the variances was taken or only the maximum variance value was considered. This problem led to the reassignment step to be replaced with a *K-means* step instead, so that we keep reassigning pixels to their closest cluster centre until no more changes take place. While this is more computationally expensive, it is necessary in order to achieve the correct segmentation. Another way this could have been achieved was to use the one pass reassignment until the correct number of clusters were found, and then use a *K-means* step at the end only once. This idea was rejected because deciding what the correct number of clusters is should be based on a clustering which itself is correct.

In summary, we can use any combination of mean or median cluster centres, maximum or average variance of the three components, and either of Euclidean or absolute distance.

The procedure can be described by the following algorithm:

```
form one cluster for the entire image

set K equal to 1

loop

    calculate the maximum variance and record the cluster, C_i, which
    produced it

    split cluster C_i

    add 1 to K

    repeatedly reassign all pixels to their closest cluster centre until no more
    changes occur

    if (stopping rule is satisfied)

        exit loop

    end if

end loop

construct segmented image containing K clusters
```

6.2 Stopping Rule 1

This stopping rule is based on the principle that we would like each of the pixels belonging to a particular cluster to be as "close" to their cluster centre as possible. This "closeness" can be measured by the variance of the pixels from their cluster centre, the lower the variance, the closer, on average, the pixel values are to the cluster centre.

The first thing we must do is calculate the variance of the entire image, when considered as one cluster, which is considered to be the initial variance. Obviously there is no need to split a cluster if the within-cluster variability is small, therefore, the simplest rule to use would be to stop the splitting process once the within-cluster

variability is small enough for every cluster. We therefore compare the cluster having maximum variability to some threshold value to determine if it is small enough to stop. The threshold chosen is simply some *weight* multiplied by the initial variance of the image as one cluster. Therefore, the cluster having the maximum variance is split if the maximum variance is greater than some *weight* multiplied by the initial variance. The process stops once the variances of every cluster is less than or equal to the threshold, at which time the final number of clusters is recorded.

The stopping rule to determine when splitting is no longer required is as follows

$$maximum \leq weight \times initial \tag{6.16}$$

where *maximum* is the variance of the cluster having maximum variability and *initial* is the initial variance of the entire image when considered to be one cluster. The value of *weight* will ultimately determine the threshold value and we therefore need to determine the optimal value for the *weight* which will enable suitable segmentation results to be obtained.

6.2.1 Results

When incorporating stopping rule 1 into the algorithm described in section 6.1, the procedure would not be totally automated as there is an unknown quantity, *weight*, which will ultimately determine the number of clusters in the final segmentation result. However, we can overcome this problem by experimentation with different values for the *weight*, thus finding its optimal value. In undertaking this task, this method was first used for the synthetic images shown in Figure 1.1 to determine what the optimal *weight* values are, and it was then also implemented for the natural

images shown in Figure 1.2.

6.2.1.1 Synthetic Images

The results in this section are based on the use of the RGB colour space. Table 6.1 shows the initial variances, using mean cluster centres, for each of the red, green and blue components as well as the average and maximum values which are used in the procedure to determine the threshold. Similarly, Table 6.2 shows the variances when using median cluster centres. These variances are different from those shown in Table 6.1 as the variances in this case are calculated as deviations from the median of all the colours rather than the mean values.

image	red	green	blue	average	maximum
two	13011.61	13009.65	13010.38	13010.55	13011.61
four	9779.60	9929.30	11725.50	10478.13	11725.50
six	9246.40	10880.52	10872.31	10333.07	10880.52
eight	13356.34	13347.49	13348.10	13350.64	13356.34
ten	12888.91	10093.07	10156.19	11046.06	12888.91
fifteen	8592.69	8510.11	8559.61	8554.14	8592.69

Table 6.1: Initial variance values for synthetic images using mean cluster centres.

image	red	green	blue	average	maximum
two	13015.66	13018.67	13019.51	13017.94	13019.51
four	12827.67	13022.11	16550.92	14133.57	16550.92
six	11846.32	14791.95	14782.15	13806.81	14791.95
eight	13368.56	13347.74	13348.35	13354.88	13368.56
ten	12888.97	10098.49	10160.34	11049.27	12888.97
fifteen	8594.82	8513.52	8562.32	8556.89	8594.82

Table 6.2: Initial variance values for synthetic images using median cluster centres.

Tables 6.3-6.6 show the number of clusters obtained for various values of *weight* using both the average variance and maximum variance for calculating the threshold and for determining which cluster to split. Table 6.3 gives the results obtained when using mean cluster centres and Euclidean distance, Table 6.4 gives the results obtained when using mean cluster centres and absolute distance, Table 6.5 shows the results

obtained when using median cluster centres and Euclidean distance, and Table 6.6 shows the results obtained when using median cluster centres and absolute distance.

image	threshold	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
two	average	2	2	2	2	2	2	2	2	2	2
	maximum	2	2	2	2	2	2	2	2	2	2
four	average	4	4	4	4	4	4	4	4	3	3
	maximum	4	4	4	4	4	4	4	4	4	4
six	average	6	6	6	6	5	5	4	4	3	3
	maximum	6	6	6	6	6	6	6	6	6	6
eight	average	8	8	8	8	8	8	4	4	4	4
	maximum	8	8	8	8	8	8	8	8	8	8
ten	average	10	10	10	8	8	8	6	4	4	4
	maximum	10	10	10	10	8	8	8	8	8	8
fifteen	average	15	15	14	14	11	10	8	8	5	4
	maximum	15	15	15	15	15	15	15	9	8	8

Table 6.3: Number of clusters for synthetic images using stopping rule 1 with mean cluster centres and Euclidean distance.

image	threshold	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
two	average	2	2	2	2	2	2	2	2	2	2
	maximum	2	2	2	2	2	2	2	2	2	2
four	average	4	4	4	4	4	4	4	4	3	3
	maximum	4	4	4	4	4	4	4	4	4	4
six	average	6	6	6	6	6	6	5	4	3	3
	maximum	6	6	6	6	6	6	6	6	6	6
eight	average	8	8	8	8	8	8	4	4	4	4
	maximum	8	8	8	8	8	8	8	8	8	8
ten	average	10	10	10	8	8	8	6	4	4	4
	maximum	10	10	10	10	8	8	8	8	8	8
fifteen	average	15	15	14	14	12	10	8	8	5	4
	maximum	15	15	15	15	15	12	12	11	8	8

Table 6.4: Number of clusters for synthetic images using stopping rule 1 with mean cluster centres and absolute distance.

Table 6.7 shows the range of *weight* values which produced the correct number of clusters. Generally, the *weight* values ranging from 0.05 to 0.20 produced correct segmentation for the synthetic images. Some images also produced the correct segmentation for higher values of the *weight*.

6.2.1.2 Natural Images

The results in this section are based on the use of the RGB colour space. Table 6.8 shows the initial variances, using mean cluster centres, for each of the red, green and blue components as well as the average and maximum values which are used in

image	threshold	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
two	average	2	2	2	2	2	2	2	2	2	2
	maximum	2	2	2	2	2	2	2	2	2	2
four	average	4	4	4	4	4	4	4	4	4	4
	maximum	4	4	4	4	4	4	4	4	4	4
six	average	6	6	6	6	6	6	6	6	6	6
	maximum	6	6	6	6	6	6	6	6	6	6
eight	average	8	8	8	8	8	8	4	4	4	4
	maximum	8	8	8	8	8	8	8	8	8	8
ten	average	10	10	10	10	10	9	8	4	4	4
	maximum	10	10	10	10	10	10	10	8	8	8
fifteen	average	15	15	14	14	14	14	13	12	9	7
	maximum	15	15	15	15	15	15	15	14	14	14

Table 6.5: Number of clusters for synthetic images using stopping rule 1 with median cluster centres and Euclidean distance.

image	threshold	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
two	average	2	2	2	2	2	2	2	2	2	2
	maximum	2	2	2	2	2	2	2	2	2	2
four	average	4	4	4	4	4	4	4	4	4	4
	maximum	4	4	4	4	4	4	4	4	4	4
six	average	6	6	6	6	6	6	3	3	3	3
	maximum	6	6	6	6	6	6	6	6	6	6
eight	average	8	8	8	8	8	8	4	4	4	4
	maximum	8	8	8	8	8	8	8	8	8	8
ten	average	10	10	10	10	10	9	8	4	4	4
	maximum	10	10	10	10	10	10	10	8	8	8
fifteen	average	15	15	14	14	14	14	13	13	9	7
	maximum	15	15	15	15	15	15	14	14	9	9

Table 6.6: Number of clusters for synthetic images using stopping rule 1 with median cluster centres and absolute distance.

cluster centre		mean	mean	median	median
distance		Euclidean	absolute	Euclidean	absolute
two	average	0.05-0.50	0.05-0.50	0.05-0.50	0.05-0.50
	maximum	0.05-0.50	0.05-0.50	0.05-0.50	0.05-0.50
four	average	0.05-0.40	0.05-0.40	0.05-0.50	0.05-0.50
	maximum	0.05-0.50	0.05-0.50	0.05-0.50	0.05-0.50
six	average	0.05-0.20	0.05-0.30	0.05-0.50	0.05-0.30
	maximum	0.05-0.50	0.05-0.50	0.05-0.50	0.05-0.50
eight	average	0.05-0.30	0.05-0.30	0.05-0.30	0.05-0.30
	maximum	0.05-0.50	0.05-0.50	0.05-0.50	0.05-0.50
ten	average	0.05-0.15	0.05-0.15	0.05-0.25	0.05-0.25
	maximum	0.05-0.20	0.05-0.20	0.05-0.35	0.05-0.35
fifteen	average	0.05-0.10	0.05-0.10	0.05-0.10	0.05-0.10
	maximum	0.05-0.35	0.05-0.25	0.05-0.35	0.05-0.30

Table 6.7: *weight* values producing correct segmentation for synthetic images.

the procedure to determine the threshold. Similarly, Table 6.9 shows the variances when using median cluster centres. As for the synthetic images, these variances are different from those shown in Table 6.8 as the variances in this case are calculated as deviations from the median of all the colours rather than the mean values.

image	red	green	blue	average	maximum
balls	3023.94	3790.97	3633.11	3482.67	3790.97
Lenna	2423.60	2832.79	1192.11	2149.50	2832.79
molecule	1608.58	2035.74	6219.90	3288.07	6219.90
teapot	3928.32	1347.19	3038.31	2771.27	3928.32
ant	6582.33	6649.38	7929.28	7053.66	7929.28
blond	810.54	1032.97	583.75	809.08	1032.97
jet	1956.16	2683.34	1020.78	1886.76	2683.34
mandrill	3063.09	2339.90	3894.56	3099.18	3894.56
peppers	2056.79	5734.32	2024.97	3272.03	5734.32
mouse	3794.42	3832.54	4006.96	3877.98	4006.96
rose	8327.96	3888.18	3300.82	5172.32	8327.96

Table 6.8: Initial variance values using mean cluster centres.

image	red	green	blue	average	maximum
balls	3331.94	4578.16	4078.47	3996.19	4578.16
Lenna	2705.53	2836.75	1209.90	2250.72	2836.75
molecule	1868.14	2378.69	7468.45	3905.09	7468.45
teapot	3983.41	1694.94	3972.97	3217.10	3983.41
ant	6899.79	6718.42	10060.43	7892.88	10060.43
blond	1123.46	1034.07	686.72	948.08	1123.46
jet	2201.36	3230.85	1139.73	2190.65	3230.85
mandrill	3063.45	2353.98	4054.40	3157.27	4054.40
peppers	2114.31	6047.27	2056.24	3405.94	6047.27
mouse	4255.29	4452.06	4425.87	4377.74	4452.06
rose	9234.94	4221.18	3545.38	5667.16	9234.94

Table 6.9: Initial variance values using median cluster centres.

Tables 6.10-6.13 show the number of clusters obtained for various values of *weight* using the average variance. Table 6.10 gives the results obtained when using mean cluster centres and Euclidean distance, Table 6.11 gives the results obtained when using mean cluster centres and absolute distance, Table 6.12 shows the results obtained when using median cluster centres and Euclidean distance, and Table 6.13 shows the results obtained when using median cluster centres and absolute distance.

For some of the images, especially *balls*, *Lenna*, *blond*, *jet* and *mandrill*, there was a big drop in the number of clusters between certain successive pairs of *weight* values,

weight	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
balls	24	14	10	8	7	6	5	5	3	3
Lenna	21	12	4	4	4	3	3	2	2	2
molecule	11	8	8	7	5	5	5	4	3	3
teapot	26	14	11	9	6	6	4	4	3	3
ant	9	6	4	3	3	3	2	2	2	2
blond	44	25	13	13	11	11	11	10	8	3
jet	29	16	11	8	5	5	4	4	4	4
mandrill	42	17	9	6	5	4	4	4	3	3
peppers	25	12	6	5	4	4	3	3	2	2
mouse	6	5	3	3	3	3	3	2	2	2
rose	11	5	4	3	3	3	2	2	2	2

Table 6.10: Number of clusters using stopping rule 1 with mean cluster centres, Euclidean distance and average variance.

weight	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
balls	20	14	8	8	8	6	4	3	3	3
Lenna	22	12	4	4	4	3	3	2	2	2
molecule	12	8	8	7	6	5	5	5	3	3
teapot	25	14	10	5	5	3	3	3	3	3
ant	9	5	4	4	3	3	2	2	2	2
blond	45	30	16	12	12	12	12	10	3	3
jet	31	15	11	9	9	6	6	6	5	4
mandrill	44	22	10	6	6	4	4	4	3	3
peppers	29	9	6	5	4	4	4	3	3	2
mouse	6	5	3	3	3	3	3	2	2	2
rose	11	6	4	3	3	3	3	2	2	2

Table 6.11: Number of clusters using stopping rule 1 with mean cluster centres, absolute distance and average variance.

as can be seen in Tables 6.10-6.13. Therefore, for these images, the number of clusters for inter-mediate *weight* values in the range where the big drop occurred are shown in appendix B.

For the most part, the number of clusters obtained was very similar for a given *weight*, regardless of which cluster centre representation, or which distance measure was used. Although they were not exactly the same, they differed by only a small amount.

In general, to determine which *weight* is the optimal value, we must compare the segmentation results obtained for each of the possible values for the *weight*. In general, a *weight* of 0.30 or above resulted in too few segments, and a *weight* of 0.05 resulted in too many segments. Following is a detailed analysis of the results for each image. The results are summarized in section 6.2.2.

weight	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
balls	27	13	10	8	6	6	6	5	3	3
Lenna	28	14	4	4	4	3	3	2	2	2
molecule	11	8	8	7	7	5	4	4	4	3
teapot	24	14	9	6	6	5	4	3	3	3
ant	8	5	4	3	3	3	3	2	2	2
blond	49	28	16	16	16	16	16	3	2	2
jet	31	16	14	6	5	5	5	5	5	3
mandrill	58	19	9	7	5	4	4	4	4	4
peppers	29	9	6	5	5	4	4	3	3	2
mouse	6	5	5	3	3	2	2	2	2	2
rose	11	5	4	3	3	2	2	2	2	2

Table 6.12: Number of clusters using stopping rule 1 with median cluster centres, Euclidean distance and average variance.

weight	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
balls	28	13	10	8	5	5	4	4	3	3
Lenna	28	13	4	4	3	3	3	2	2	2
molecule	12	8	8	6	5	5	5	4	4	4
teapot	24	15	10	7	7	3	3	3	3	3
ant	7	5	4	3	3	3	3	2	2	2
blond	54	31	27	12	12	11	11	3	3	2
jet	37	18	15	5	5	5	5	5	5	3
mandrill	63	26	10	7	5	4	4	4	4	4
peppers	34	10	8	5	5	4	3	3	3	2
mouse	6	5	5	3	3	2	2	2	2	2
rose	12	7	4	3	3	3	2	2	2	2

Table 6.13: Number of clusters using stopping rule 1 with median cluster centres, absolute distance and average variance.

For the image *balls*, a *weight* of 0.20 and above resulted in too few segments to adequately represent the colours of each of the balls. The results using a *weight* of 0.10 looked more like the original image. This is due to the fact that the more segments the result has, the more colours it requires to represent it, so naturally we expect the result to resemble the original image more closely. Despite this, and based on the results shown in Tables 6.10-6.13, the results using a *weight* of 0.15 were the best as they have lower numbers of clusters and still provide good segmentations. The only exception to this is for the mean cluster centres and absolute distance approach which resulted in only 8 clusters. However, considering the number of clusters produced by some intermediate *weight* values, as shown in appendix B, the mean cluster centres and absolute distance approach also produced a result with 10 clusters when the *weight* value was 0.11. In fact, the other combinations of cluster centre and distance types produced

10 clusters for a range of *weight* values, being 0.12-0.16 for mean cluster centres and Euclidean distance, 0.14-0.18 for median cluster centres and Euclidean distance and 0.12-0.17 for median cluster centres and absolute distance. The best overall result was achieved when using median cluster centres with Euclidean distance, which is most apparent when looking at the pink ball which was visually a more smooth cluster, whereas with the mean cluster centres and Euclidean distance result, the pink ball was a mixture of pink and grey, and didn't look much like a ball at all.

Lenna provided some strange results. Firstly, there was a big jump in the number of clusters when changing from a *weight* of 0.10 to 0.15. The results using a *weight* of 0.10 created separate segments for the portion of the image where the feathers are located, however, 12, 13 or 14 clusters may seem a little high for this image. The results when using a *weight* of 0.15 provided a segmented image where the feathers were assigned to the same segment as the hair or part of the background. Consequently, we would conclude that the desired *weight* value lies somewhere in the range 0.10-0.15. When using mean cluster centres and Euclidean distance, the best result occurred when using a *weight* of 0.13 which resulted in 7 clusters, although, the result with 8 clusters was very similar and was produced with a *weight* of 0.12. The best result for the mean cluster centres and absolute distance approach was produced when the *weight* value was in the range 0.10-0.12 which gave 12 clusters. The median cluster centres and Euclidean distance approach produced its best result for 9 clusters when a *weight* of 0.12 was used, and the median cluster centres and absolute distance approach produced its best result when the *weight* was in the range 0.12-0.13, which resulted in 10 clusters. Each of these results were very similar from a visual point of view, however, the mean cluster centres and Euclidean distance approach resulted in the lowest number of clusters and the result was still visually good, so this is taken to be

the best result.

For the image *molecule*, *weight* values of 0.10 and 0.15 gave good results for each of the four combinations of cluster centre and distance type, and each of these results were virtually identical. Any *weight* above 0.15 resulted in too few segments to adequately represent the different parts of the molecule.

For the image *teapot*, the best segmentation results occurred when the number of clusters was 9 or 10. This was achieved when a *weight* of 0.20 was used with mean cluster centres and Euclidean distance and when 0.15 was used with every other cluster centre and distance combination. The best result occurred when using mean cluster centres with Euclidean distance which correctly classified all of the reflection of the teapot to the correct clusters.

For the image *ant*, a *weight* of 0.10 produced the best results for all approaches except median cluster centres and Euclidean distance, where a *weight* of 0.05 was required because the result using a *weight* of 0.10 incorrectly classified part of the ant's head together with the mouse. The problem with the ant's head also occurred for the other three approaches when using a *weight* of 0.15 and above. The results using mean cluster centres with either distance and median cluster centres and absolute distance were all very similar, and were the best, as the ant's head and the mouse had less clusters representing them compared to the result using median cluster centres and Euclidean distance.

The image *blond*, in general, required a larger *weight* value than the other images. As for the image *Lenna*, this image also had a big drop in the number of clusters between particular pairs of successive *weight* values, as can be seen in Tables 6.10-6.13. Taking smaller partitions for the *weight* values, results of which are shown in appendix B, did not change the big drop that occurred. The *weight* values giving

the best results are 0.44-0.45 for mean cluster centres and Euclidean distance, 0.39-0.41 for mean cluster centres and absolute distance, 0.15 - 0.39 for median cluster centres and Euclidean distance, and 0.30 - 0.35 for median cluster centres and absolute distance. The number of clusters produced by the best result for median cluster centres and Euclidean distance was slightly high at 16. However, the remaining results produced quite a similar number of clusters, and the best overall results were obtained when using mean cluster centres with either distance type.

For the image *jet*, there was a big drop in the number of clusters between the *weight* values 0.15-0.20 when using median cluster centres with either distance type. Taking the results from Tables 6.10-6.13 in combination with the extra results shown in appendix B, the overall best results for each combination of cluster centre and distance type occurred when 6 clusters were obtained. The *weight* values producing these results were 0.24 for mean cluster centres and Euclidean distance, 0.29-0.40 for mean cluster centres and absolute distance, 0.20-0.24 for median cluster centres and Euclidean distance, and 0.19 for median cluster centres and absolute distance. Each of these results were in fact very similar, and no preference can be given to any of the four approaches.

For the image *mandrill*, considering the results in Tables 6.10-6.13 in combination with the extra results shown in appendix B, the *weight* values producing the best results were 0.15 for mean cluster centres and Euclidean distance, 0.14-0.15 for mean cluster centres and absolute distance, 0.14 for median cluster centres and Euclidean distance and 0.15-0.16 for median cluster centres and absolute distance. All of the four results were very similar, and no preference can be given to any of the four approaches.

For the image *peppers*, a *weight* of 0.15 gave good results for each of the four

approaches. All of the results for the four approaches were very similar, and no preference can be given to any of the four approaches.

The results when using a *weight* of 0.05 produced the best segmentation results for the image *mouse* with each of the four combinations of cluster centres and distance type, so no preference can be given to any of the four approaches. The results when using a *weight* of 0.10 were also not bad, but it was hard to make out the ears of the mouse. When using a *weight* of 0.15 and above, part of the feet were grouped together with the background, as there were too few segments to adequately represent the mouse.

A *weight* of 0.20 and above resulted in too few segments for the image *rose*, and the resulting segmented images did not adequately show the individual layers of the petals. The best results were obtained when using a *weight* of 0.10 for each of the four approaches, with the individual petal layers being quite clear. The best result was achieved with the median cluster centres and absolute distance approach.

For each combination of cluster centre and distance type, the best segmentation results obtained when using average variance are shown in Figures 6.1-6.4, where Figure 6.1 shows the results when using mean cluster centres and Euclidean distance, Figure 6.2 shows the results when using mean cluster centres and absolute distance, Figure 6.3 shows the results when using median cluster centres and Euclidean distance and Figure 6.4 shows the results when using median cluster centres and absolute distance, with the *weight* values producing these results indicated in the figures. Overall, a *weight* of about 0.15 produced good segmentation results for most of the natural images. However, the images *ant*, *mouse* and *rose* obtained better results using a lower *weight*, the value 0.10, the image *blond* required a larger *weight* of about 0.40, and *jet* required a *weight* value of about 0.20. Therefore, we can restrict

the value of the *weight* to lie between 0.10 and 0.20, with the only image requiring a *weight* outside this range being *blond*. This was mostly due to the fact that the image *blond* as a whole had a very small variability, so an acceptable threshold was calculated by using a larger *weight* value. For most of the images the mean cluster centres and Euclidean distance approach produced good results, except for *balls* for which the median cluster centres and Euclidean distance approach produced the best result, and for *rose*, the median cluster centres and absolute distance approach produced the best result.

Tables 6.14-6.17 show the number of clusters obtained for various values of *weight* using the maximum variance. Table 6.14 shows the results obtained when using mean cluster centres and Euclidean distance, Table 6.15 shows the results obtained when using mean cluster centres and absolute distance, Table 6.16 shows the results obtained when using median cluster centres and Euclidean distance, and Table 6.17 shows the results obtained when using median cluster centres and absolute distance. The number of clusters for the images *balls*, *Lenna*, *blond*, *jet* and *mandrill* produced by some inter-mediate *weight* values are shown in appendix B.

weight	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
balls	31	18	9	8	8	7	6	6	4	4
Lenna	27	15	5	4	3	3	3	3	3	2
molecule	10	10	7	6	6	4	4	2	2	2
teapot	25	16	11	7	6	6	4	4	4	4
ant	10	5	4	4	3	3	3	3	3	3
blond	53	30	24	14	12	11	11	9	9	9
jet	39	16	12	6	5	4	4	4	4	3
mandrill	61	24	10	8	6	4	4	4	3	3
peppers	21	8	6	6	5	3	3	3	3	3
mouse	7	5	4	3	3	3	3	2	2	2
rose	10	5	4	3	3	3	3	3	3	3

Table 6.14: Number of clusters using stopping rule 1 with mean cluster centres, Euclidean distance and maximum variance.

For the most part, the number of clusters obtained was very similar for a given *weight*, regardless of which cluster centre representation or which distance measure

weight	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
balls	38	22	10	9	8	6	6	4	4	3
Lenna	30	18	6	4	4	3	3	3	3	3
molecule	10	9	8	8	8	5	2	2	2	2
teapot	29	17	11	5	4	4	4	3	3	3
ant	11	5	5	5	3	3	3	3	3	3
blond	66	34	30	16	14	13	12	12	4	3
jet	43	24	16	9	9	7	7	7	6	2
mandrill	73	33	12	8	6	5	4	4	4	3
peppers	26	9	6	6	5	5	4	3	3	3
mouse	6	5	4	3	3	3	3	2	2	2
rose	12	6	6	3	3	3	3	3	3	3

Table 6.15: Number of clusters using stopping rule 1 with mean cluster centres, absolute distance and maximum variance.

weight	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
balls	39	15	8	8	8	6	6	4	3	3
Lenna	34	19	5	4	4	3	3	3	3	2
molecule	10	9	7	6	6	6	2	2	2	2
teapot	32	18	12	7	6	5	4	4	4	4
ant	9	6	4	3	3	3	3	3	3	3
blond	74	41	29	15	14	13	13	11	7	3
jet	35	17	11	5	5	5	3	3	2	2
mandrill	87	24	14	8	6	4	4	4	4	4
peppers	33	8	6	5	5	3	3	3	3	3
mouse	6	5	5	5	3	3	2	2	2	2
rose	10	5	4	3	3	3	3	3	3	2

Table 6.16: Number of clusters using stopping rule 1 with median cluster centres, Euclidean distance and maximum variance.

was used. However, there were some exceptions, most notably for *mouse*, when using median cluster centres with either distance measure, the number of clusters was maintained at 5 for more values of the *weight*, whereas for the mean cluster centres representation, the only suitable results were obtained for *weight* values of 0.05 and 0.10, which does not correspond to the best results for any of the other images.

As with the average variance approach, to determine which *weight* is the optimal value, we must compare the segmentation results obtained for each of the possible values for the *weight*. In general, a *weight* of 0.30 or above resulted in too few segments, and a *weight* of 0.05 resulted in too many segments. Following is a detailed analysis of the results for each image. The results are summarized in section 6.2.2.

For the image *balls*, *weight* values of 0.10 or smaller resulted in too many clusters, while *weight* values of 0.20 or more resulted in too few clusters to adequately represent

weight	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
balls	37	19	11	9	9	7	4	4	3	3
Lenna	35	20	5	4	4	4	4	3	3	2
molecule	10	9	6	6	6	5	2	2	2	2
teapot	29	20	15	7	7	7	5	5	3	3
ant	8	5	5	3	3	3	3	3	3	3
blond	75	40	29	28	18	15	15	4	3	3
jet	46	16	12	5	5	5	5	3	2	2
mandrill	80	40	20	9	7	4	4	4	4	4
peppers	39	8	7	6	5	3	3	3	3	3
mouse	6	5	5	5	3	3	2	2	2	2
rose	13	6	4	3	3	3	3	3	3	3

Table 6.17: Number of clusters using stopping rule 1 with median cluster centres, absolute distance and maximum variance.

the image. The best results occurred when the number of clusters was 9 or 10. Taking into account the results in Tables 6.14-6.17 and the extra results shown in appendix B, the *weight* values producing the best results were 0.14-0.16 for mean cluster centres and Euclidean distance, 0.15-0.18 for mean cluster centres and absolute distance, 0.14 for median cluster centres and Euclidean distance and 0.18 for median cluster centres and absolute distance. In each of these cases the number of clusters was 10 except for the mean cluster centres and Euclidean distance result which had 9 clusters. The overall best result occurred when using mean cluster centres and Euclidean distance as the red ball was clearly showing as a separate segment.

As when using the average variance values, the results for the image *Lenna*, shown in Tables 6.14-6.17, indicate that there is a big drop in the number of clusters when changing from a *weight* of 0.10 to 0.15 The results using a *weight* of 0.10 for all four approaches created a separate segment for the portion of the image where the feathers are located, however, cluster numbers of 15, 18, 19 or 20 may seem a little high for this image. The results when using a *weight* of 0.15 for all approaches provided a segmented image where the feathers were assigned to the same segment as the hair or part of the background. Consequently, the ideal *weight* values probably lie somewhere between 0.10 and 0.15. Results using other *weight* values within this

range are shown in appendix B. These results show that a big drop still exists in the number of clusters for most combinations of cluster centre and distance type. For the mean cluster centres and Euclidean distance, mean cluster centres and absolute distance and median cluster centres and absolute distance approaches, the best results occurred when a *weight* value of 0.14 was used, although in the case of median cluster centres and absolute distance, the number of clusters was 17 which is much too big for this image, while the next higher *weight* resulted in 5 clusters which was not enough to represent the image. The median cluster centres and Euclidean distance approach also had similar failings in that the best result occurred for *weight* values of 0.11-0.13 which resulted in 17 clusters and the next higher *weight* value also gave only 5 clusters. The best overall result was produced by the mean cluster centres and Euclidean distance approach, which only resulted in 9 clusters and looked very good.

For the image *molecule*, a *weight* of 0.10 gave good results for every approach except when using mean cluster centres with absolute distance, where a *weight* of 0.15 gave the best result. Also, a *weight* of 0.05 gave good results for the mean cluster centres and Euclidean distance approach, and *weight* values of 0.20-0.25 gave good results for the mean cluster centres and absolute distance approach. Any *weight* above those just mentioned for the four approaches resulted in too few segments to adequately represent the different parts of the molecule. The results obtained using mean cluster centres and absolute distance were the best.

For the image *teapot*, the best segmentation results occurred when the number of clusters was 11 or 12. This was achieved when a *weight* of 0.15 was used for all combinations of cluster centre and distance type, except the median cluster centres and absolute distance approach, for which he number of clusters was slightly on the high side at 15. All of the results, except when using median cluster centres and

absolute distance, were very similar, and they were all acceptable results.

For the image *ant*, a *weight* of 0.10 produced the best result when using mean cluster centres with Euclidean distance. Values of 0.10-0.20 for *weight* produced the best result when using mean cluster centres and absolute distance. For a higher *weight*, part of the head of the ant was incorrectly classified together with the mouse. A *weight* of 0.10 produced the best result when using median cluster centres and Euclidean distance, and the best result when using the median cluster centres and absolute distance approach was achieved with *weight* values of 0.10 and 0.15. The results using mean cluster centres with either distance type and when using median cluster centre with absolute distance were all similar and better than the result obtained when using median cluster centres and Euclidean distance, as less colours were needed to represent the mouse.

The image *blond*, in general, required a larger *weight* value than the other images. The results indicate a big drop in the number of clusters between particular pairs of successive *weight* values, as can be seen in Tables 6.14-6.17. Taking smaller partitions for the *weight* values, results of which are shown in appendix B, did not change the big drop that occurred. The *weight* values giving the best results are 0.40-0.50 for mean cluster centres and Euclidean distance, 0.35-0.43 for mean cluster centres and absolute distance, 0.45 for median cluster centres and Euclidean distance, and 0.30-0.37 for median cluster centres and absolute distance. The number of clusters produced by the best result for median cluster centres and absolute distance was slightly high at 15. However, the remaining results produced quite a similar number of clusters, and all of the results looked very similar.

For the image *jet*, considering the results in Tables 6.14-6.17 in combination with the extra results shown in appendix B, the *weight* values producing the best results

were 0.20-0.21 for mean cluster centres and Euclidean distance, which resulted in 6 clusters, 0.19-0.28 for mean cluster centres and absolute distance, which resulted in 9 clusters, 0.20-0.30 for median cluster centres and Euclidean distance, which resulted in 5 clusters, and 0.18-0.35 for median cluster centres and absolute distance, which resulted in 5 clusters. Each of these results were in fact very similar, and no preference can be given to any of the four approaches.

For the image *mandrill*, considering the results in Tables 6.14-6.17 in combination with the extra results shown in appendix B, the *weight* values producing the best results are 0.15 for mean cluster centres and Euclidean distance, which resulted in 10 clusters, 0.17 for mean cluster centres and absolute distance, which resulted in 10 clusters, 0.16-0.18 for median cluster centres and Euclidean distance, which resulted in 12 clusters, and 0.18-0.21 for median cluster centres and absolute distance, which resulted in 9 clusters. Each of these results were in fact very similar, and no preference can be given to any of the four approaches.

For the image *peppers*, a *weight* of 0.15 gave good results for each of the approaches, except for the median cluster centres and absolute distance approach where the best result was obtained using a *weight* of 0.20. When using mean cluster centres with either distance type a good result was also obtained for a *weight* of 0.20. All of the results for the four approaches were very similar, and no preference can be given to any of the four approaches.

The results when using a *weight* of 0.05 produced the best segmentation results for the image *mouse* with each of the four combinations of cluster centre and distance type, so no preference can be given to any of the four approaches. The results when using a *weight* of 0.10 were also not bad, as well as 0.15 and 0.20 when using median cluster centres with either distance type, but it was hard to make out the ears of

the mouse in these results. When using higher *weight* values, part of the feet were grouped together with the background as there were too few segments to adequately represent the mouse.

A *weight* of 0.20 and above resulted in too few segments for the image *rose*, and the resulting segmented images did not adequately show the individual layers of the petals. The best results were obtained when using a *weight* of 0.10 for each of the four approaches, with the individual petal layers being quite clear. The mean cluster centres and absolute distance approach also obtained a good result for a *weight* of 0.15. The best results were achieved using median cluster centres with either distance type.

For each combination of cluster centre and distance type, the best segmentation results obtained when using maximum variance are shown in Figures 6.5-6.8, where Figure 6.5 shows the results when using mean cluster centres and Euclidean distance, Figure 6.6 shows the results when using mean cluster centres and absolute distance, Figure 6.7 shows the results when using median cluster centres and Euclidean distance and Figure 6.8 shows the results when using median cluster centres and absolute distance, with the *weight* values producing these results indicated in the figures. Overall, a *weight* of about 0.15 produced good segmentation results for most of the natural images. However, the images *mouse* and *rose* obtained better results using a lower *weight*, the value 0.10, the image *blond* required a larger *weight* of about 0.40 and *jet* required a *weight* value of 0.20. Therefore, we can restrict the value of the *weight* to lie between 0.10 and 0.20, with the only image requiring a *weight* outside this range being *blond* due to this image as a whole having a much smaller variance. For most of the images the mean cluster centres and Euclidean distance approach produced good results, except for *molecule* for which the mean cluster centres and

absolute distance approach produced the best result, and for *rose*, using median cluster centres with either distance type produced the best result.

6.2.1.3 Results Using Context

Each of the results incorporating the context of the image are based on using only mean cluster centres and Euclidean distance and the use of the RGB colour space. Also, comparisons made to the results by the original RGB values refer to the results using mean cluster centres and Euclidean distance presented in section 6.2.1.2.

Tables 6.18-6.21 show the results using the average variance approach and only using the context values. Table 6.18 shows the number of clusters when using *mean* context values, Table 6.19 shows the number of clusters when using *median* context values, Table 6.20 shows the number of clusters when using *weighted* context values, and Table 6.21 shows the number of clusters when using *regression* context values. The number of clusters for the images *balls*, *Lenna*, *blond*, *jet* and *mandrill* produced by some inter-mediate *weight* values are shown in appendix B. Following is a detailed analysis of the results for each image. The results are summarized in section 6.2.2.

weight	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
balls	20	10	7	7	7	6	5	3	3	3
Lenna	32	13	7	4	4	3	3	2	2	2
molecule	17	9	8	7	5	5	4	4	4	4
teapot	26	15	10	7	6	4	4	3	3	3
ant	7	5	4	3	3	3	3	2	2	2
blond	103	37	22	16	12	4	3	3	2	2
jet	56	26	15	9	6	6	5	5	5	3
mandrill	85	38	9	7	5	4	4	4	3	3
peppers	92	34	7	5	4	4	3	3	2	2
mouse	6	5	3	3	3	3	3	2	2	2
rose	12	7	4	3	3	3	2	2	2	2

Table 6.18: Number of clusters using stopping rule 1 with *mean* context values only and average variance.

For the image *balls*, there was generally a big drop in the number of clusters when changing from a *weight* of 0.05 to 0.10, regardless of which context type was used.

weight	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
balls	26	10	9	7	6	6	6	3	3	3
Lenna	30	14	4	4	3	3	3	2	2	2
molecule	12	8	8	7	5	5	5	4	3	3
teapot	25	14	11	6	6	6	4	3	3	3
ant	7	5	4	3	3	3	2	2	2	2
blond	90	46	14	11	9	3	3	3	2	2
jet	48	26	12	6	5	5	4	4	4	3
mandrill	65	25	9	6	5	4	4	4	3	3
peppers	54	9	6	5	4	4	3	3	2	2
mouse	6	5	3	3	3	3	3	2	2	2
rose	11	7	4	3	3	3	2	2	2	2

Table 6.19: Number of clusters using stopping rule 1 with *median* context values only and average variance.

weight	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
balls	18	10	7	7	7	6	5	3	3	3
Lenna	35	13	5	4	4	3	3	2	2	2
molecule	16	9	8	7	5	5	4	4	4	4
teapot	26	15	11	7	6	4	4	3	3	3
ant	7	5	4	3	3	3	2	2	2	2
blond	111	37	23	16	14	10	3	3	2	2
jet	56	23	15	10	6	6	5	5	5	3
mandrill	89	41	9	7	5	4	4	4	3	3
peppers	84	32	7	5	4	4	3	3	2	2
mouse	6	5	3	3	3	3	3	2	2	2
rose	12	7	4	3	3	3	2	2	2	2

Table 6.20: Number of clusters using stopping rule 1 with *weighted* context values only and average variance.

Taking into account the results in Tables 6.18-6.21 and the extra results shown in appendix B, the best results for the *mean*, *weighted* and *regression* contexts occurred when using *weight* values in the range 0.09-0.11, and for the *median* context, the best results occurred when *weight* values in the range 0.10-0.12 were used. For the *mean*, *median* and *weighted* contexts, the best results contained 10 clusters, while for the *regression* context the best result contained 12 clusters. The best visual result was achieved when the *mean* or *weighted* contexts were used as the pink ball was mostly assigned to one cluster, however, with the *median* and *regression* contexts, the middle of the pink ball was partly like the background, with bits of yellow as well. In comparison to the results using the original RGB values, the context results showed more smoother and connected clusters, which is a clear advantage as each ball should be classified as one segment.

weight	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
balls	34	12	9	7	6	5	5	3	3	3
Lenna	42	14	5	4	3	3	3	2	2	2
molecule	18	9	8	7	5	5	5	4	4	4
teapot	25	15	11	7	6	6	4	3	3	3
ant	9	5	4	3	3	3	2	2	2	2
blond	164	61	32	24	16	14	11	8	3	2
jet	88	32	18	13	6	6	6	6	6	3
mandrill	83	37	9	7	5	4	4	3	3	3
peppers	107	18	6	5	4	4	3	3	3	2
mouse	6	5	3	3	3	3	3	2	2	2
rose	13	6	4	3	3	3	2	2	2	2

Table 6.21: Number of clusters using stopping rule 1 with *regression* context values only and average variance.

For the image *Lenna*, there was a big drop in the number of clusters when changing from a *weight* of 0.10 to 0.15, regardless of which context type was used. The results using a *weight* of 0.10 produced separate segments for the portion of the image where the feathers are located, however, 13 or 14 clusters may seem a little high for this image. The results when using a *weight* of 0.15 provided a segmented image where the feathers were assigned to the same segment as the hair or background for all cases except the *mean* context result. The ideal *weight* would probably lie in the range 0.10-0.15. The *weight* values giving the best results were 0.13-0.14 for the *mean* context, 0.11-0.13 for the *median* context, 0.13-0.14 for the *weighted* context and 0.13 for the *regression* context. In all of these cases the number of clusters was 8. All of the four results were very similar, and no preference can be given to any of the four context types. In comparison to the results using the original RGB values, the context results were more smoother, and looked a little blurry.

For each of the four context types, the best segmentation results for the image *molecule* occurred when a *weight* of 0.15 was used, as well as a *weight* of 0.10 for the *median* context type. The best overall result occurred when using the *median* context, although the other three context types also produced very similar results except for a slight blurring effect around the edges of the molecule. These results

were very similar to the best result produced when using the original RGB values.

For the image *teapot*, the best results for each of the four context types were achieved when a *weight* of 0.15 was used. The overall best result was achieved when using the *mean* context which produced less shades in the actual teapot. The other three results were very similar and also looked very similar to the best result produced when using only the original RGB values.

For the image *ant*, the best results for each of the four context types were achieved when using a *weight* of 0.05. The overall best results were achieved when using the *mean*, *median* and *weighted* context types, which were all very similar. The *regression* context result had two extra clusters which were not really required for this image. In comparison to the results using the original RGB values, the context results had more smoother looking segments.

For the image *blond*, taking the results from Tables 6.18-6.21 in combination with the extra results in appendix B, the *weight* values giving the best results were 0.29 for the *mean* context, 0.22-0.25 for the *median* context, 0.29-0.30 for the *weighted* context and 0.39 for the *regression* context. The resulting number of clusters was 10 for the *mean* and *weighted* contexts and 9 for the *median* and *regression* contexts. The overall best results were obtained when using the *median* and *regression* contexts as these contained one fewer number of clusters, and the extra cluster in the *mean* and *weighted* context results only added an extra shade in the face and hands areas. Apart from these minor differences, the results were quite similar. These context results were much better than the results using only the original RGB values, as these results correctly identified the hair as a separate cluster in yellow, which did not occur when using only the original RGB values for an equivalent number of clusters.

For the image *jet*, taking the results in Tables 6.18-6.21 in combination with the extra results in appendix B, the *weight* values producing the best results were 0.23-0.30 for the *mean* and *weighted* contexts, 0.19-0.21 for the *median* context and 0.25-0.45 for the *regression* context. In all of these results the number of clusters was 6. The overall best result was obtained when using the *regression* context, as the F-16 symbol and the writing on the plane was clearer. In comparison to the results using the original RGB values, the context results, especially for the *mean*, *median* and *weighted* contexts, were blurry and generally more smoother looking, while the writing on the plane was much clearer when using only the original RGB values.

For the image *mandrill*, considering the results in Tables 6.18-6.21 in combination with the extra results shown in appendix B, the *weight* values producing the best results were 0.14-0.19 for the *mean*, *weighted* and *regression* contexts, and 0.14-0.16 for the *median* context. In each of these cases the number of clusters was 9. The results were all very similar, except when using the *median* context for which the right eye was partly classified together with the green fur segment. In comparison to the results using the original RGB values, the context results appeared a little blurred.

For the image *peppers*, the best results were obtained when using a *weight* of 0.15. Each of the four results by the four different context types were very similar, so no preference can be given to one context type over the others for this image. In comparison to the results by the original RGB values, the context results had more smoother looking segments.

For the image *mouse*, the best results for each of the four context types were obtained when a *weight* of 0.05 was used. Each of these four results were very similar in appearance, however, the *mean* and *weighted* contexts were able to classify more of

the left ear as the same segment. In comparison to the results using only the original RGB values, the context results had more smoother segments.

For the image *rose*, the segmented results when using a *weight* of 0.10 for each of the four context types were the best and each of these results looked very similar, except for the *regression* context result which contained one fewer number of clusters and therefore contained one less shade of red in the rose. In comparison to the results using the original RGB values, the context results showed more smoother clusters.

For each context type, the best segmentation results obtained when using only the context components are shown in Figures 6.9-6.12, where Figure 6.9 shows the results when using the *mean* context, Figure 6.10 shows the results when using the *median* context, Figure 6.11 shows the results when using the *weighted* context and Figure 6.12 shows the results when using the *regression* context, with the *weight* values producing these results indicated in the figures.

Generally, the context results tended to be slightly blurred and also contained more smooth looking segments. This is not surprising since when we apply the context operation to the original image, the resulting image is a slightly blurred version of the original since we are using the eight neighbours in calculating new colour values. Overall, the *mean* and *weighted* contexts performed well with good results occurring for most images. The *median* and *regression* contexts also produced good results for half of the images. It is very difficult to prefer one context type over the others as there was a lot a variation in the results, and in many cases the results by the four contexts looked very similar.

Tables 6.22-6.25 show the results using the average variance approach with both the original and context values. Table 6.22 shows the number of clusters when using the original and *mean* context values, Table 6.23 shows the number of clusters when using

the original and *median* context values, Table 6.24 shows the number of clusters when using the original and *weighted* context values, and Table 6.25 shows the number of clusters when using the original and *regression* context values. The number of clusters for the images *balls*, *Lenna*, *blond*, *jet* and *mandrill* produced by some inter-mediate *weight* values are shown in appendix B.

weight	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
balls	65	19	10	8	7	7	7	6	3	3
Lenna	178	31	23	4	4	3	3	3	2	2
molecule	50	15	12	7	7	5	5	5	4	4
teapot	79	20	13	8	6	6	4	4	3	3
ant	10	7	4	3	3	3	2	2	2	2
blond			101	58	41	22	19	19	3	3
jet	204	76	42	28	21	14	5	5	5	5
mandrill		98	41	10	7	6	4	4	3	3
peppers	202	53	7	5	5	4	3	3	3	2
mouse	6	5	3	3	3	3	3	2	2	2
rose	12	6	4	3	3	3	2	2	2	2

Table 6.22: Number of clusters using stopping rule 1 with original and *mean* context values and average variance.

weight	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
balls	61	19	11	7	7	6	6	3	3	3
Lenna	188	61	18	4	4	3	3	2	2	2
molecule	61	10	9	8	7	5	5	5	5	4
teapot	98	46	13	10	6	6	4	3	3	3
ant	11	5	4	3	3	3	2	2	2	2
blond			142	94	22	18	17	17	3	3
jet	252	114	60	38	13	5	5	5	5	5
mandrill		107	41	13	8	6	4	4	3	3
peppers	236	65	7	5	5	4	3	3	2	2
mouse	6	5	3	3	3	3	3	2	2	2
rose	11	6	4	3	3	3	2	2	2	2

Table 6.23: Number of clusters using stopping rule 1 with original and *median* context values and average variance.

For many of the images, the number of clusters obtained when a *weight* of 0.05 was used was very large. The exceptions to this occurred for the images *ant*, *mouse* and *rose*. Also, for the images *Lenna*, *blond*, *jet*, *mandrill* and *peppers*, the number of clusters when using a *weight* of 0.10 was also quite large, especially for *blond* which produced a high number of clusters for the *weight* values 0.15 and 0.20 as well. Following is a detailed analysis of the results for each image. The results are

weight	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
balls	60	18	10	7	7	7	7	6	3	3
Lenna	142	37	22	4	4	3	3	3	2	2
molecule	49	14	12	7	7	5	5	5	4	4
teapot	73	21	13	8	6	6	4	4	3	3
ant	10	5	4	3	3	3	2	2	2	2
blond			102	52	44	20	19	19	3	3
jet	211	75	36	29	17	7	5	5	5	5
mandrill		79	36	10	7	5	4	4	3	3
peppers	222	35	7	5	5	4	3	3	2	2
mouse	6	5	3	3	3	3	3	2	2	2
rose	12	6	4	3	3	3	2	2	2	2

Table 6.24: Number of clusters using stopping rule 1 with original and *weighted* context values and average variance.

weight	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
balls	97	18	13	7	7	7	7	6	3	3
Lenna		36	17	4	4	3	3	2	2	2
molecule	57	27	8	7	6	5	5	5	4	3
teapot	62	16	12	7	6	6	4	4	3	3
ant	12	5	4	3	3	3	2	2	2	2
blond		232	148	50	39	23	23	20	16	3
jet		87	50	31	19	6	6	5	5	5
mandrill		82	42	10	7	5	4	4	3	3
peppers		41	6	5	4	4	3	3	3	2
mouse	6	5	3	3	3	3	3	2	2	2
rose	11	6	4	3	3	3	2	2	2	2

Table 6.25: Number of clusters using stopping rule 1 with original and *regression* context values and average variance.

summarized in section 6.2.2.

For the image *balls*, considering the results in Tables 6.22-6.25 in combination with the extra results shown in appendix B, the *weight* values producing the best results were 0.15 for the *mean* context, 0.16 for the *median* and *regression* contexts and 0.14-0.15 for the *weighted* context. Each of these results were very similar in appearance, and no preference can be given to any of the four context types when used together with the original colour values. In comparison to the results using the original RGB values with a comparable number of clusters, the context results showed more smoother and connected clusters which is a clear advantage as each ball should be classified as one segment.

For the image *Lenna*, there was a big drop in the number of clusters when changing from a *weight* of 0.15 to 0.20, regardless of which context type was used. The results

using a *weight* of 0.15 produced separate segments for the portion of the image where the feathers are located, however, 17, 18, 22 and 23 clusters is too many for this image. The results when using a *weight* of 0.20 provided a segmented image where the feathers were assigned to the same segment as the hair or background for each of the context types. This leads to the conclusion that the ideal *weight* value should lie in the range 0.15-0.20. The *weight* values giving the best results were 0.16 for the *mean* and *weighted* contexts and 0.15-0.16 for the *median* and *regression* contexts. However, the number of clusters for these results was still quite high at 18, 18, 20 and 17 for the *mean*, *median*, *weighted* and *regression* contexts, respectively. Each of these results by the four contexts were very similar in appearance, however, the *regression* context result was accepted as the best since this resulted in the lowest number of clusters. The results for each of the four contexts produced by the *weight* values 0.17-0.25 each only contained 4 clusters, which is not enough to adequately represent this image. In comparison to the results using the original RGB values, the context results were more smoother, which is most noticeable in the background regions, although, when using only the original RGB values only 12 clusters were identified when using a *weight* of 0.10, which is much more reasonable than 17.

For each of the four context types, the best segmentation results for the image *molecule* occurred when a *weight* of 0.15 was used. The best overall result occurred when using the *regression* context, which resulted in 8 clusters and was virtually identical to the best result when using only the original RGB values.

For the image *teapot*, the best results for each of the *mean*, *median* and *weighted* context types was achieved when a *weight* of 0.20 was used, and the best result when using the *regression* context was achieved when using a *weight* of 0.15. The overall best result was achieved when using the *regression* context for which no blue was

incorporated into the reflection part of the image. The *mean* and *weighted* context results had some blue in the reflection region but the teapot itself had less shades to represent it. The best result by using only the original RGB values also had some blue in the reflection regions, however, the teapot region looked almost identical to the *regression* context result.

For the image *ant*, the best results for the *median*, *weighted* and *regression* context types was achieved when using a *weight* of 0.05. The overall best result was achieved when using the *mean* context type with a *weight* of 0.10, which resulted in fewer clusters and still adequately represented the different regions in the image. In comparison to the results using only the original RGB values, the context results had more smooth looking segments.

For the image *blond*, taking the results from Tables 6.22-6.25 in combination with the extra results in appendix B, the *weight* values giving the best results were 0.35-0.42 for the *mean* and *weighted* contexts, 0.35-0.40 for the *median* context and 0.45 for the *regression* context. These results contained a high number of clusters, with the lowest being 16 for the *regression* context. The results using any higher *weight* values all resulted in only 3 clusters for each of the context types which is not enough to adequately represent the regions in this image. The best results for the *mean* and *weighted* contexts showed the lips as a separate cluster, however, these results contained 19 clusters which is a little high. The overall best result is the *regression* context result which had the lowest number of clusters at 16. These context results were much better than the results using only the original RGB values, as these results correctly identified the hair as a separate cluster in yellow, which did not occur when using only the original RGB values until a much higher number of clusters was obtained.

For the image *jet*, taking the results in Tables 6.22-6.25 in combination with the extra results in appendix B, the *weight* values producing the best results were 0.31 for the *mean* and *weighted* contexts, 0.28 for the *median* context and 0.29-0.35 for the *regression* context. In most of the cases these gave 6 clusters, except for the *median* context result which had 7 clusters. The best overall result was obtained when using the *regression* context, as this resulted in the snow region in the background being more like one region in the segmented image. In comparison to the results using only the original RGB values, the context results were a little blurry, which was mostly visible in relation to the writing on the plane, while the writing on the plane was much clearer when using only the original RGB values.

For the image *mandrill*, when considering the results in Tables 6.22-6.25 in combination with the extra results shown in appendix B, the *weight* values producing the best results were 0.20-0.22 for the *mean* and *weighted* contexts, 0.22 for the *median* context and 0.20-0.21 for the *regression* context. These resulted in 10 clusters for all context types except the *median* context for which 11 clusters were obtained. The results were all very similar, and no preference can be given to any of the four context types. In comparison to the results using the original RGB values, the fur in the context results appeared more smooth looking.

For the image *peppers*, the best results were obtained when using a *weight* of 0.15 for each of the four context types. The overall best result occurred when using the *regression* context which had one fewer number of clusters than the other results as the extra cluster did not add anything significant to the result. However, the *mean*, *median* and *weighted* context results were also acceptable. In comparison to the results by the original RGB values, the context results had more smoother looking segments.

For the image *mouse*, the best results for each of the four context types were obtained when a *weight* of 0.05 was used. Each of these four results were very similar in appearance, however, the *mean* and *weighted* contexts were able to classify more of the left ear as the same segment. In comparison to the results using only the original RGB values, the context results had more smoother looking segments, but otherwise looked very similar.

For the image *rose*, the segmented results when using a *weight* of 0.10 for each of the four context types were the best and each of these results looked very similar. In comparison to the results using only the original RGB values, the context results showed more smoother looking clusters.

For each context type, the best segmentation results obtained when using the original colour values together with the context values are shown in Figures 6.13-6.16, where Figure 6.13 shows the results when using the original and *mean* context values, Figure 6.14 shows the results when using the original and *median* context values, Figure 6.15 shows the results when using the original and *weighted* context values and Figure 6.16 shows the results when using the original and *regression* context values, with the *weight* values producing these results indicated in the figures.

Generally, the context results tended to have more smooth looking segments, which is an advantage over only using the original RGB values as we expect segments to be connected regions in the image. This is not surprising since when we apply the context operation to the original image we are using the eight neighbours in calculating new colour values. Overall, the *regression* context performed very well producing good results for most of the natural images. The *mean* and *weighted* contexts also produced reasonable results for about half of the images, while the *median* context produced reasonable results for only three of the images. However, the *regression* context was

the overall best, as in addition to producing good results for most of the images, it also produced a clearly better segmentation result for four of the images, while for many of the other images very similar results were produced by each of the four context types.

6.2.1.4 Results Using Different Colour Spaces

All of the results in this section are based on only the use of the average variance in calculating the threshold and determining which cluster to split and also when only using mean cluster centres and Euclidean distance. The results using the RGB colour space with mean cluster centres and Euclidean distance were presented in section 6.2.1.2.

Tables 6.26-6.28 show the number of clusters obtained for various values of *weight* using the various colour spaces. Table 6.26 shows the results obtained when using the XYZ colour space, Table 6.27 shows the results obtained when using the KL colour space and Table 6.28 shows the results obtained when using the HSV colour space. The number of clusters for the images *balls*, *Lenna*, *blond*, *jet* and *mandrill* produced by some inter-mediate *weight* values are shown in appendix B. Following is a detailed analysis of the results for each image. The results are summarized in section 6.2.2.

weight	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
balls	17	8	8	8	6	6	5	3	3	3
Lenna	14	5	4	4	3	3	3	2	2	2
molecule	10	7	6	5	5	5	3	3	3	2
teapot	24	13	10	9	5	5	5	4	3	3
ant	7	4	4	3	3	3	2	2	2	2
blond	32	17	16	10	10	10	10	3	2	2
jet	17	9	7	5	5	4	4	3	3	3
mandrill	28	10	7	5	4	3	3	3	2	2
peppers	17	6	5	4	4	3	2	2	2	2
mouse	6	5	3	3	3	3	3	2	2	2
rose	6	4	3	3	3	2	2	2	2	2

Table 6.26: Number of clusters using stopping rule 1 with the XYZ colour space and average variance.

weight	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
balls	21	13	9	7	7	5	5	5	3	3
Lenna	22	11	4	4	4	3	3	2	2	2
molecule	11	8	8	7	5	5	5	4	3	3
teapot	12	9	5	4	3	2	2	2	2	2
ant	9	7	4	3	3	3	2	2	2	2
blond	49	20	13	11	11	11	11	10	8	3
jet	33	17	10	8	5	5	5	5	5	3
mandrill	42	17	12	7	5	4	4	4	3	3
peppers	28	8	6	5	4	4	3	3	2	2
mouse	6	5	3	3	3	3	3	2	2	2
rose	11	5	4	3	3	3	2	2	2	2

Table 6.27: Number of clusters using stopping rule 1 with the KL colour space and average variance.

weight	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
balls	66	44	34	27	20	16	14	13	10	9
Lenna	9	7	3	3	2	2	2	2	2	2
molecule	40	24	15	12	8	6	4	4	4	4
teapot	39	22	16	10	9	7	7	4	3	3
ant	38	18	14	9	7	6	6	6	5	3
blond	43	25	23	15	14	14	13	12	9	9
jet	30	18	11	8	6	5	5	5	3	3
mandrill	51	25	16	13	10	8	8	6	5	4
peppers	27	13	11	7	6	5	4	3	3	3
mouse	15	10	9	7	3	3	3	3	3	2
rose	19	11	8	8	7	6	6	5	5	5

Table 6.28: Number of clusters using stopping rule 1 with the HSV colour space and average variance.

For the image *balls*, a *weight* of 0.12-0.16 produced the best segmentation for the KL and RGB colour spaces, which both produced very similar segmentation results. The XYZ colour space produced its best result with a *weight* of 0.08, which gave 11 clusters, and the HSV result, when using a *weight* of 0.35, did a fairly good job of obtaining the different colours of the balls, as the red and green balls can clearly be seen to be red and green in the segmented image. Overall though, the RGB and KL colour spaces produced the best segmentation results, however, the HSV result was also quite good.

The best results for the image *Lenna* were produced when a *weight* of 0.13 was used for the RGB and KL colour spaces, giving 7 and 8 clusters, respectively, and when a *weight* of 0.08 was used for the XYZ colour space, giving 9 clusters, and when a *weight* of 0.05-0.07 was used for the HSV colour space, giving 9 clusters. The result

using the HSV colour space appeared very noisy and the result by the XYZ colour space also produced a very noisy looking background. The results by the RGB and KL colour spaces were the best, which both contained regions which appeared more connected.

The best results for the image *molecule* were produced when using the RGB and KL colour spaces with a *weight* of 0.10-0.15, when 8 clusters were obtained. The best results when using the XYZ and HSV colour spaces were obtained when a *weight* of 0.05 was used, which resulted in 10 and 40 clusters, respectively. In both of these results there were too many clusters, especially the HSV result, however, results with fewer clusters failed to separate the various parts of the molecule into separate clusters when using the HSV colour space.

For the image *teapot*, the best result was obtained when using the KL colour space with a *weight* of 0.05. The result using a *weight* of 0.20 with the RGB colour space was also good. For this image the HSV colour space also did a reasonable job when a *weight* of 0.15 was used except that the shadow of the teapot on the checkerboard was considered as one region, not showing the individual squares of the board. The result when using the XYZ colour space with a *weight* value of 0.20 was also reasonable except that parts of the reflection in the dark squares of the checker board were grouped together with the dark squares in the shadow region.

For the image *ant*, the best results were produced when using the XYZ or KL colour spaces with a *weight* of 0.05 and 0.10, respectively. Both of these results were visually very similar. The result by the RGB colour space with a *weight* of 0.10 was also similar, except that it contained one fewer clusters. The results by the HSV colour space generally looked noisy, however, the result using a *weight* of 0.20 was not too bad.

The best results for the image *blond* were produced by the RGB and KL colour spaces when the *weight* was in the range 0.44-0.45. The result by the XYZ colour space using a *weight* of 0.20-0.37 was also acceptable. The result by the HSV colour space using a *weight* of 0.20 produced more clusters than the other results mentioned, and the lips were correctly assigned to a separate cluster and appeared pink, however, the remainder of the image including the hair, hands and face were all assigned to the same cluster and were hard to distinguish.

For the image *jet*, the best results were obtained when using the RGB or KL colour spaces with a *weight* of 0.24. The best result for the XYZ colour space occurred when the *weight* was 0.13-0.17, however, the red parts in the plane appeared noisy in this result. The best result for the HSV colour space occurred when using a *weight* of 0.24-0.25, however, the image as a whole, especially in the background, appeared noisy in this result.

For the image *mandrill*, a *weight* of 0.15 produced the best result for the RGB colour space and a *weight* of 0.16 produced the best result for the KL colour space. Both of these were very similar and were the overall best results. The result for the XYZ colour space using a *weight* of 0.07 produced 19 clusters, however, any larger *weight* produced a very poor segmentation. The HSV colour space result when using a *weight* of 0.15-0.16, while producing 16 clusters, was also very poor with red specks appearing in the region which contains mostly green and black fur.

For the image *peppers*, the best results were obtained when using the RGB or KL colour spaces with a *weight* of 0.15, in which case the results were visually very similar. The result by the HSV colour space with a *weight* of 0.15 was a little noisy, especially in the regions containing the red peppers. The result by the XYZ colour space using a *weight* of 0.15 was also very poor with some red being included in the

region containing the large green pepper in the front.

For the image *mouse*, each of the RGB, XYZ and KL colour spaces produced the best results, all of which were visually similar, when a *weight* of 0.05 was used. The best result for the HSV colour space was obtained when a *weight* of 0.15 was used, however, this result looked very noisy, especially in the region on the right side of the mouse.

For the image *rose*, the RGB and KL colour spaces produced very good results when a *weight* of 0.05 was used. Both of these results were visually very similar. The results using a *weight* of 0.10 for both of these colour spaces was also acceptable, except that these results contained some red in the background region, however, these are considered the overall best. The XYZ colour space using a *weight* of 0.05 resulted in the background being classified as part of the dark shade in the rose, and the HSV colour space resulted in most of the rose being one cluster, loosing detail in the layers of the petals, even though it had the same number of clusters. The best result for the HSV colour space occurred when using a *weight* of 0.10.

For each of the colour spaces used, the best segmentation results obtained are shown in Figures 6.17-6.19, where Figure 6.17 shows the results when using the XYZ colour space, Figure 6.18 shows the results when using the KL colour space and Figure 6.19 shows the results when using the HSV colour space, with the *weight* values producing these results indicated in the figures.

Overall, the RGB and KL colour spaces produced very similar results from a segmentation point of view. The HSV and XYZ colour spaces in general performed very poorly. Therefore, there seems to be no clear advantage in choosing a different colour space other than RGB as there is no real benefit in doing so.

6.2.2 Summary of Results

Tables 6.29-6.33 show the *weight* values giving the best segmentations for each of the natural images for the different cluster centre and distance types with average variance and maximum variance, the different context types, when used on their own and together with the original values, and the different colour spaces, respectively. The result which is visually the best for each image is indicated in bold. The results produced by the *weight* values shown in Tables 6.29-6.33 match with the results shown in Figures 6.1-6.19.

image	mean & Euc.	mean & abs.	med. & Euc.	med. & abs.
balls	0.12-0.16	0.11	**0.14-0.18**	0.12-0.17
Lenna	**0.13**	0.10-0.12	0.12	0.12-0.13
molecule	**0.10-0.15**	**0.10-0.15**	**0.10-0.15**	**0.10-0.15**
teapot	**0.20**	0.15	0.15	0.15
ant	**0.10**	**0.10**	0.05	**0.10**
blond	**0.44-0.45**	**0.39-0.41**	0.15-0.39	0.30-0.35
jet	**0.24**	**0.29-0.40**	**0.20-0.24**	**0.19**
mandrill	**0.15**	**0.14-0.15**	**0.14**	**0.15-0.16**
peppers	**0.15**	**0.15**	**0.15**	**0.15**
mouse	**0.05**	**0.05**	**0.05**	**0.05**
rose	0.10	0.10	0.10	**0.10**

Table 6.29: *weight* values producing best results for all four combinations of cluster centre and distance type using average variance.

image	mean & Euc.	mean & abs.	med. & Euc.	med. & abs.
balls	**0.14-0.16**	0.15-0.18	0.14	0.18
Lenna	**0.14**	0.14	0.11-0.13	0.14
molecule	0.05-0.10	**0.15-0.25**	0.10	0.10
teapot	**0.15**	**0.15**	**0.15**	0.15
ant	**0.10**	**0.10-0.20**	0.10	**0.10-0.15**
blond	**0.40-0.50**	**0.35-0.43**	**0.45**	0.30-0.37
jet	**0.20-0.21**	**0.19-0.28**	**0.20-0.30**	**0.18-0.35**
mandrill	**0.15**	**0.17**	**0.16-0.18**	**0.18-0.21**
peppers	**0.15-0.20**	**0.15-0.20**	**0.15**	**0.20**
mouse	**0.05**	**0.05**	**0.05**	**0.05**
rose	0.10	0.10-0.15	**0.10**	**0.10**

Table 6.30: *weight* values producing best results for all four combinations of cluster centre and distance type using maximum variance.

When using stopping rule 1 with the splitting method discussed in section 6.1, a *weight* of 0.15 produces the best segmentation results for both the average variance and maximum variance approaches. However, some images require a slightly lower

image	*mean*	*median*	*weighted*	*regression*
balls	**0.09-0.11**	0.10-0.12	**0.09-0.11**	0.09-0.11
Lenna	**0.13-0.14**	**0.11-0.13**	**0.13-0.14**	**0.13**
molecule	0.15	**0.10-0.15**	0.15	0.15
teapot	**0.15**	0.15	0.15	0.15
ant	**0.05**	**0.05**	**0.05**	0.05
blond	0.29	**0.22-0.25**	0.29-0.30	**0.39**
jet	0.23-0.30	0.19-0.21	0.23-0.30	**0.25-0.45**
mandrill	**0.14-0.19**	0.14-0.16	**0.14-0.19**	**0.14-0.19**
peppers	**0.15**	**0.15**	**0.15**	**0.15**
mouse	**0.05**	0.05	**0.05**	0.05
rose	**0.10**	**0.10**	**0.10**	0.10

Table 6.31: *weight* values producing best results when using only context values.

image	*mean*	*median*	*weighted*	*regression*
balls	**0.15**	**0.16**	**0.14-0.15**	**0.16**
Lenna	0.16	0.15-0.16	0.16	**0.15-0.16**
molecule	0.15	0.15	0.15	**0.15**
teapot	0.20	0.20	0.20	**0.15**
ant	**0.10**	0.05	0.05	0.05
blond	0.35-0.42	0.35-0.40	0.35-0.42	**0.45**
jet	0.31	0.28	0.31	**0.29-0.35**
mandrill	**0.20-0.22**	**0.22**	**0.20-0.22**	**0.20-0.21**
peppers	0.15	0.15	0.15	**0.15**
mouse	**0.05**	0.05	**0.05**	0.05
rose	**0.10**	**0.10**	**0.10**	**0.10**

Table 6.32: *weight* values producing best results when using original and context values.

or higher *weight* value, but generally we can restrict the *weight* to be between 0.10 and 0.20.

The *regression* context type is performing quite well for most of the images, especially when both the original and context values are taken into account to perform the clustering. When using both the original and context values a higher *weight* value is needed to achieve the correct level of segmentation. When using the context values,

image	RGB	XYZ	KL	HSV
balls	**0.12-0.16**	0.08	**0.12-0.16**	0.35
Lenna	**0.13**	0.08	**0.13**	0.05-0.07
molecule	**0.10-0.15**	0.05	**0.10-0.15**	0.05
teapot	0.20	0.20	**0.05**	0.15
ant	0.10	**0.05**	**0.10**	0.20
blond	**0.44-0.45**	**0.20-0.37**	**0.44-0.45**	0.20
jet	**0.24**	0.13-0.17	**0.24**	0.24-0.25
mandrill	**0.15**	0.07	**0.16**	0.15-0.16
peppers	**0.15**	0.15	**0.15**	0.15
mouse	**0.05**	**0.05**	**0.05**	0.15
rose	**0.10**	0.05	**0.10**	0.10

Table 6.33: *weight* values producing best results for different colour spaces.

especially when on their own, the segmentation results look smoother and the regions more connected, which is a clear advantage over the results using only the original RGB values. However, when only the context values are used, the segmented results look slightly blurred, which could be a problem when fine details, such as the writing on the plane in the image *jet*, need to be extracted. Another problem that can occur, especially when the background and foreground colours are similar, is that parts of the objects get grouped together with the background.

The results by different colour spaces indicate that the RGB colour space performs quite well and there is no real benefit in selecting a different colour space in which to perform the clustering. The visual results indicated that the RGB and KL colour spaces perform very similarly, and the HSV colour space tends to result in the segmented images looking very noisy. The XYZ colour space in general performed poorly and only achieved reasonable segmentation results for the images *ant*, *blond* and *mouse*.

The method is a very simple technique, which may have wide ranging applications with respect to medical images by, for example, finding abnormalities, which should be grouped as a separate cluster. Also, this method may be used for multispectral images, since it is very easy to extend this technique to any dimensionality, which was demonstrated by the incorporation of the context of the image which required the clustering to be performed for six components.

6.3 Stopping Rule 2

Stopping rule 2 is based on the average intra-cluster distance which is calculated as

$$\frac{1}{N}\sum_{i=1}^{K}\sum_{\mathbf{x}\in C_i}\sqrt{(x_R - R_i)^2 + (x_G - G_i)^2 + (x_B - B_i)^2} \tag{6.17}$$

where N is the total number of pixels in the image, K is the number of clusters in the image, C_i is the ith cluster having cluster centre $\mathbf{z}_i = (R_i,\ G_i,\ B_i)$ which can be calculated as the mean or median of the pixels in the cluster for each of the components, and $\mathbf{x}$ is the vector, $(x_R,\ x_G,\ x_B)$, containing the red, green and blue values for the pixels. This definition is based on the Euclidean distance measure. The absolute distance measure can also be used to determine the average intra-cluster distance, calculated as

$$\frac{1}{N}\sum_{i=1}^{K}\sum_{\mathbf{x}\in C_i} \mid x_R - R_i \mid + \mid x_G - G_i \mid + \mid x_B - B_i \mid \tag{6.18}$$

We then calculate two other measures based on the average intra-cluster distance, namely, normalized intra-cluster distance and percentage decrease in intra-cluster distance. The normalized intra-cluster distance for K clusters is calculated as

$$\mathrm{norm}_{base}(K) = intra_K / intra_{base} \tag{6.19}$$

where $intra_K$ is the average intra-cluster distance with K clusters, and $intra_{base}$ is the average intra-cluster distance with *base* clusters over which each intra-cluster distance is normalized. Usually $intra_{base}$ would be taken as $intra_1$ which makes the normalized intra-cluster distance a percentage of the intra-cluster distance with only one cluster, but considering an image must have at least two clusters, we have chosen

to use the normalizing factor of $intra_2$. A good segmentation can be considered as having a small error, and this can be measured by the normalized intra-cluster distance, which must be required to be below a threshold, say $perc_1$. The percentage decrease in intra-cluster distance for K clusters is simply the amount of change in going from $K-1$ to K clusters and is calculated as

$$\text{percDec}(K) = (intra_{K-1} - intra_K)/intra_{K-1} \tag{6.20}$$

where $intra_{K-1}$ is the average intra-cluster distance when the image has $K-1$ clusters and $intra_K$ is the average intra-cluster distance when the image has K clusters. The simplest rule which we can use to select the number of clusters is as follows:

Find the smallest value of K satisfying the condition

$$\text{percDec}(K+1) < perc_2 \tag{6.21}$$

where $perc_2$ is the percentage value which the percentage decrease must be smaller than.

This rule basically selects the clustering results after which the segmented image is not changing much. The only potential problem with this rule is that it can result in the selection of a number of clusters which is too small, due to some local property where there is a small percentage decrease. We can overcome this problem by combining the rule given by equation 6.21 with the use of the normalized intra-cluster distance.

Given that the first occurrence of the normalized intra-cluster distance being below a threshold, $perc_1$, is at K clusters, then the number of clusters is taken as K if the

rule

$$\text{percDec}(K) < perc_2 \tag{6.22}$$

is satisfied, otherwise, the number of clusters will be the smallest value of K_2, where $K_2 \geq K$, satisfying the rule

$$\text{percDec}(K_2 + 1) < perc_2 \tag{6.23}$$

The rules given by equations 6.22 and 6.23 only consider a possible valid clustering to occur if the normalized intra-cluster distance is small enough. Therefore, the rule we are trying to satisfy says that when the error in the segmented image is small enough and the segmented image is not changing much, we can stop.

6.3.1 Results

6.3.1.1 Synthetic Images

When using stopping rule 2 in the algorithm described in section 6.1, two threshold values, $perc_1$ and $perc_2$, are needed, which will ultimately determine the number of clusters in the final segmentation results. Therefore, the procedure is not totally automated. However, we can overcome this problem by experimentation with different values for these thresholds, thus finding their optimal values. In undertaking this task, we first used the synthetic images shown in Figure 1.1 to determine what the optimal percentage values are, the results of which are based on the use of the RGB colour space.

Table 6.34 shows the intra-cluster distances obtained for each of the synthetic images for one cluster up to 25 clusters based on the use of mean cluster centres and Euclidean distance. Obviously, the lower the intra-cluster distance, the better the

segmented result as the clusters are more well formed. However, Table 6.34 shows that as the number of clusters increases, the intra-cluster distances decrease, so we cannot simply decide to stop at the certain number of clusters which produces the lowest intra-cluster distance. Therefore, we must determine when the intra-cluster distance is small enough.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
1	197.51	159.98	159.45	200.10	179.16	150.71
2	7.31	110.34	87.15	163.39	128.63	129.56
3	6.74	26.31	70.71	139.48	117.84	118.30
4	6.35	2.38	59.75	115.58	106.98	103.99
5	6.03	2.24	43.61	88.20	85.85	94.11
6	5.62	2.16	2.44	60.83	65.00	84.05
7	5.43	2.06	2.32	33.46	50.98	75.34
8	5.21	1.95	2.28	6.10	36.70	66.06
9	4.93	1.90	2.24	6.00	22.26	58.43
10	4.81	1.82	2.19	5.90	8.12	52.07
11	4.70	1.78	2.11	5.79	7.97	42.27
12	4.47	1.69	2.03	5.69	7.81	32.88
13	4.40	1.65	2.00	5.59	7.71	23.45
14	4.34	1.60	1.97	5.49	7.54	14.20
15	4.25	1.57	1.94	5.39	7.46	7.45
16	4.19	1.54	1.89	5.28	7.28	7.35
17	4.14	1.49	1.81	5.21	7.17	7.24
18	3.99	1.45	1.79	5.14	7.06	7.17
19	3.95	1.43	1.74	5.07	6.87	7.06
20	3.91	1.41	1.72	5.00	6.75	7.00
21	3.86	1.38	1.70	4.93	6.69	6.89
22	3.72	1.37	1.68	4.86	6.62	6.81
23	3.68	1.35	1.64	4.79	6.53	6.73
24	3.65	1.34	1.62	4.72	6.47	6.61
25	3.61	1.31	1.59	4.66	6.42	6.57

Table 6.34: Intra-cluster distances for synthetic images using mean cluster centres and Euclidean distance.

To decide whether the intra-cluster distance is small enough, we first normalize the values by dividing each of the intra-cluster distances, from two clusters upwards, by the intra-cluster distance for two clusters. Table 6.35 shows the normalized intra-cluster distances for two clusters up to 25 clusters based on the use of mean cluster centres and Euclidean distance. If you choose a particular number of clusters and look at the normalized intra-cluster distance for each image, you can notice that the figure differs from image to image, and therefore, on its own, may not lead to an individual figure at which to stop if we were to use a rule based on stopping if the normalized intra-cluster distance is below a given threshold.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
3	0.9232	0.2385	0.8114	0.8537	0.9161	0.9130
4	0.8689	0.0216	0.6856	0.7074	0.8317	0.8027
5	0.8248	0.0203	0.5004	0.5398	0.6674	0.7264
6	0.7697	0.0196	0.0280	0.3723	0.5053	0.6487
7	0.7432	0.0186	0.0266	0.2048	0.3963	0.5815
8	0.7132	0.0177	0.0261	0.0373	0.2853	0.5098
9	0.6744	0.0172	0.0257	0.0367	0.1731	0.4510
10	0.6583	0.0165	0.0252	0.0361	0.0632	0.4019
11	0.6435	0.0161	0.0242	0.0355	0.0620	0.3262
12	0.6116	0.0153	0.0233	0.0348	0.0607	0.2538
13	0.6021	0.0149	0.0229	0.0342	0.0599	0.1810
14	0.5936	0.0145	0.0226	0.0336	0.0586	0.1096
15	0.5814	0.0142	0.0223	0.0330	0.0580	0.0575
16	0.5739	0.0139	0.0216	0.0323	0.0566	0.0567
17	0.5662	0.0135	0.0208	0.0319	0.0557	0.0559
18	0.5462	0.0132	0.0206	0.0315	0.0549	0.0553
19	0.5405	0.0130	0.0200	0.0310	0.0534	0.0545
20	0.5351	0.0128	0.0197	0.0306	0.0524	0.0541
21	0.5289	0.0125	0.0195	0.0302	0.0520	0.0532
22	0.5095	0.0124	0.0193	0.0297	0.0515	0.0526
23	0.5038	0.0122	0.0188	0.0293	0.0508	0.0519
24	0.4996	0.0121	0.0186	0.0289	0.0503	0.0510
25	0.4946	0.0119	0.0183	0.0285	0.0499	0.0507

Table 6.35: Normalized intra-cluster distances for synthetic images with $intra_{base}$ equal to $intra_2$.

To overcome this problem we incorporate the percentage decrease in intra-cluster distance, for which the values based on the use of mean cluster centres and Euclidean distance are shown in Table 6.36. We can use these numbers to determine whether there was a significant change in the segmented image from increasing the number of clusters by one. Visual analysis indicates that when this number is below 0.05, there is no significant change in the segmented image, and anything above 0.05 shows quite a difference in the images. Bearing this in mind, the idea is to combine the normalized intra-cluster distance and the percentage decrease in intra-cluster distance so that the rule we are trying to satisfy says that when the error in the segmented image is small enough and the segmented image is not changing much, we can stop.

Table 6.37 shows the number of clusters obtained when using mean cluster centres and Euclidean distance for various values for $perc_1$ and $perc_2$, where $perc_1$ represents the threshold for the normalized intra-cluster distance and $perc_2$ represents the threshold for the percentage decrease in intra-cluster distance. Similarly, Table 6.38

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	0.9630	0.3103	0.4534	0.1835	0.2820	0.1403
3	0.0768	0.7615	0.1886	0.1463	0.0839	0.0870
4	0.0588	0.9094	0.1550	0.1714	0.0922	0.1209
5	0.0507	0.0606	0.2702	0.2369	0.1975	0.0950
6	0.0668	0.0361	0.9440	0.3103	0.2429	0.1069
7	0.0345	0.0473	0.0509	0.4499	0.2157	0.1037
8	0.0403	0.0502	0.0176	0.8177	0.2801	0.1232
9	0.0545	0.0291	0.0183	0.0167	0.3933	0.1155
10	0.0238	0.0388	0.0187	0.0168	0.6351	0.1089
11	0.0226	0.0249	0.0376	0.0173	0.0183	0.1882
12	0.0496	0.0482	0.0397	0.0176	0.0204	0.2221
13	0.0155	0.0255	0.0147	0.0178	0.0134	0.2867
14	0.0141	0.0312	0.0142	0.0187	0.0214	0.3944
15	0.0205	0.0185	0.0145	0.0183	0.0107	0.4756
16	0.0130	0.0186	0.0284	0.0190	0.0238	0.0130
17	0.0133	0.0317	0.0382	0.0139	0.0159	0.0152
18	0.0354	0.0244	0.0129	0.0135	0.0155	0.0102
19	0.0104	0.0148	0.0270	0.0138	0.0265	0.0151
20	0.0099	0.0120	0.0136	0.0139	0.0182	0.0077
21	0.0116	0.0235	0.0132	0.0142	0.0084	0.0166
22	0.0367	0.0119	0.0095	0.0142	0.0105	0.0112
23	0.0112	0.0105	0.0258	0.0143	0.0127	0.0117
24	0.0083	0.0116	0.0089	0.0145	0.0091	0.0184
25	0.0100	0.0203	0.0177	0.0129	0.0078	0.0054

Table 6.36: Percentage decrease in intra-cluster distances for synthetic images.

shows the number of clusters obtained when using mean cluster centres and absolute distance, Table 6.39 shows the number of clusters obtained when using median cluster centres and Euclidean distance, and Table 6.40 shows the number of clusters obtained when using median cluster centres and absolute distance.

An important note is that since we are using the normalized intra-cluster distance with $base_2$, this means that for two clusters, this measure will be set to 1.0. Consequently, we would not be able to select the results with only two clusters. For the synthetic image with two clusters, we would have to use the normalized intra-cluster distance with $base_1$, in which case the correct result can be obtained. However, the major drop in the intra-cluster distance occurs when changing from 1 cluster to 2, and by using the intra-cluster distance for two clusters as the normalizing factor, we can obtain a more robust measure for the variety of colour images we have.

Generally, for a given value for $perc_2$, the number of clusters is the same for the various values of $perc_1$ shown in Tables 6.37-6.40. The only exceptions to this

$perc_1$	0.30	0.40	0.50	0.60	0.30	0.40	0.50	0.60
$perc_2$	0.05	0.05	0.05	0.05	0.10	0.10	0.10	0.10
four	5	5	5	5	4	4	4	4
six	7	7	7	7	6	6	6	6
eight	8	8	8	8	8	8	8	8
ten	10	10	10	10	10	10	10	10
fifteen	15	15	15	15	15	15	15	15

Table 6.37: Number of clusters for synthetic images using stopping rule 2 with mean cluster centres and Euclidean distance.

$perc_1$	0.30	0.40	0.50	0.60	0.30	0.40	0.50	0.60
$perc_2$	0.05	0.05	0.05	0.05	0.10	0.10	0.10	0.10
four	5	5	5	5	4	4	4	4
six	7	7	7	7	6	6	6	6
eight	8	8	8	8	8	8	8	8
ten	10	10	10	10	10	10	10	10
fifteen	15	15	15	15	15	15	15	7

Table 6.38: Number of clusters for synthetic images using stopping rule 2 with mean cluster centres and absolute distance.

occurred for the image *fifteen* when using mean cluster centres and absolute distance and median cluster centres and Euclidean distance, and also for the image *six* when using median cluster centres and Euclidean distance. The images *four* and *six* required a value of 0.10 for $perc_2$ to correctly identify the number of clusters mainly due to these images having a small number of clusters and there being a tendency for larger percentage decrease values to occur for small numbers of clusters. The other images were able to detect the correct number of clusters for both values of $perc_2$ shown in Tables 6.37-6.40, leading to the conclusion that 0.05 is a big enough value to use for images with higher numbers of clusters. The only image which failed to correctly identify the number of clusters was *fifteen* when using median cluster centres and

$perc_1$	0.30	0.40	0.50	0.60	0.30	0.40	0.50	0.60
$perc_2$	0.05	0.05	0.05	0.05	0.10	0.10	0.10	0.10
four	5	5	5	5	4	4	4	4
six	7	7	7	3	6	6	6	3
eight	8	8	8	8	8	8	8	8
ten	10	10	10	10	10	10	10	10
fifteen	15	10	10	10	15	10	10	10

Table 6.39: Number of clusters for synthetic images using stopping rule 2 with median cluster centres and Euclidean distance.

$perc_1$	0.30	0.40	0.50	0.60	0.30	0.40	0.50	0.60
$perc_2$	0.05	0.05	0.05	0.05	0.10	0.10	0.10	0.10
four	5	5	5	5	4	4	4	4
six	7	7	7	7	6	6	6	6
eight	8	8	8	8	8	8	8	8
ten	10	10	10	10	10	10	10	10
fifteen	9	9	9	9	9	9	9	9

Table 6.40: Number of clusters for synthetic images using stopping rule 2 with median cluster centres and absolute distance.

absolute distance. This was due to there being a percentage decrease value less than 0.05 occurring for the result with 10 clusters. In this case, a value of 0.20 for $perc_1$ would be needed to allow the correct number of clusters to be identified.

6.3.1.2 Natural Images

In order to determine the optimal values of $perc_1$ and $perc_2$ for the natural colour images shown in Figure 1.2, we can experiment with different values for these thresholds, thus finding their optimal values. The results in this section are based on the use of the RGB colour space.

Table 6.41 shows the intra-cluster distances obtained for each of the natural images for one cluster up to 25 clusters based on the use of mean cluster centres and Euclidean distance. Obviously, the lower the intra-cluster distance, the better the segmented results as the clusters are more well formed. However, Table 6.41 shows that as the number of clusters increases, the intra-cluster distances decrease, so we cannot simply decide to stop at the certain number of clusters which produces the lowest intra-cluster distance. Therefore, we must determine when the intra-cluster distance is small enough.

To decide whether the intra-cluster distance is small enough, we first normalize the values by dividing each of the intra-cluster distances from two clusters upwards by the intra-cluster distance for two clusters. Table 6.42 shows the normalized intra-cluster

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
1	71.42	70.44	72.38	83.02	134.87	43.60	60.62	88.16	92.24	84.06	115.42
2	35.27	43.58	15.30	56.74	74.77	27.81	26.72	65.33	55.24	42.07	62.55
3	29.03	33.43	11.41	35.99	39.99	20.91	23.41	51.73	44.31	35.54	40.45
4	20.85	25.93	8.96	31.77	30.41	20.01	22.08	42.67	36.34	17.41	30.91
5	19.78	22.83	7.25	18.95	26.96	19.63	18.85	37.93	31.58	12.86	27.17
6	17.80	21.45	5.37	17.32	25.63	16.76	18.50	35.05	27.95	10.86	24.69
7	17.33	20.26	4.52	15.88	21.45	15.55	18.29	33.64	27.10	10.00	23.05
8	16.53	19.27	3.64	13.53	21.07	15.09	18.14	31.16	25.00	9.25	21.78
9	16.21	18.31	3.38	12.62	19.15	14.71	18.09	29.77	24.71	8.75	20.90
10	15.35	18.09	3.15	12.18	17.21	14.63	15.50	28.44	22.86	8.35	20.60
11	15.24	18.00	2.87	11.51	15.65	14.51	13.57	27.37	22.49	8.14	17.75
12	13.93	17.07	2.58	11.09	14.48	13.95	13.48	26.24	21.42	7.94	16.60
13	13.73	16.63	2.44	10.53	13.99	13.86	13.40	25.61	20.92	7.36	15.77
14	13.50	16.24	2.14	10.35	13.85	12.92	12.36	25.29	20.79	7.11	15.11
15	13.33	15.65	2.05	9.92	13.73	12.87	12.18	24.91	19.64	6.98	14.47
16	13.27	15.45	1.97	9.80	13.59	12.44	11.95	24.10	19.33	6.85	13.85
17	13.17	15.18	1.89	9.19	13.27	12.23	11.68	23.62	19.13	6.79	13.25
18	13.12	15.12	1.84	9.12	13.09	11.35	11.41	23.49	18.79	6.70	13.13
19	12.88	15.09	1.70	8.70	12.85	11.14	11.28	23.11	18.58	6.62	12.79
20	12.82	15.08	1.61	8.57	12.24	11.07	11.12	22.15	18.43	6.61	12.01
21	12.78	14.51	1.57	8.40	11.68	11.02	11.04	21.98	17.80	6.60	11.22
22	12.72	14.13	1.55	8.11	11.59	10.99	11.02	21.40	17.59	6.56	11.06
23	12.66	13.97	1.54	8.08	11.44	10.98	10.81	20.88	17.13	6.48	10.84
24	12.63	13.89	1.54	7.98	11.32	10.95	9.81	20.35	16.51	6.44	10.77
25	11.87	13.18	1.54	6.82	11.24	10.43	9.74	19.97	16.32	5.97	10.59

Table 6.41: Intra-cluster distances.

distances for two clusters up to 25 clusters based on the use of mean cluster centres and Euclidean distance. If you choose a particular number of clusters and look at the normalized intra-cluster distance for each image, you can notice that the figure differs from image to image, and therefore, on its own, may not lead to an individual figure at which to stop if we were to use a rule based on stopping if the normalized intra-cluster distance is below a given threshold.

To overcome this problem we incorporate the percentage decrease in intra-cluster distance, for which the values based on the use of mean cluster centres and Euclidean distance are shown in Table 6.43. We can use these numbers to determine whether there was a significant change in the segmented image from increasing the number of clusters by one. Visual analysis indicates that when this number is below 0.05, there is no significant change in the segmented image, and anything above 0.05 shows quite a difference in the images. Bearing this in mind, the idea is to combine the normalized intra-cluster distance and the percentage decrease in intra-cluster distance so that the

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
3	0.8231	0.7672	0.7461	0.6342	0.5348	0.7517	0.8762	0.7919	0.8021	0.8448	0.6468
4	0.5912	0.5951	0.5858	0.5599	0.4067	0.7195	0.8264	0.6531	0.6579	0.4140	0.4942
5	0.5607	0.5239	0.4741	0.3340	0.3605	0.7059	0.7055	0.5806	0.5718	0.3057	0.4344
6	0.5046	0.4922	0.3510	0.3052	0.3428	0.6024	0.6925	0.5366	0.5059	0.2581	0.3947
7	0.4912	0.4648	0.2952	0.2798	0.2869	0.5589	0.6847	0.5149	0.4906	0.2378	0.3686
8	0.4686	0.4423	0.2377	0.2385	0.2817	0.5425	0.6791	0.4769	0.4526	0.2200	0.3482
9	0.4594	0.4201	0.2210	0.2223	0.2561	0.5288	0.6772	0.4556	0.4473	0.2081	0.3341
10	0.4352	0.4150	0.2057	0.2146	0.2302	0.5261	0.5803	0.4353	0.4139	0.1985	0.3293
11	0.4321	0.4130	0.1873	0.2028	0.2094	0.5217	0.5081	0.4189	0.4071	0.1936	0.2838
12	0.3949	0.3917	0.1689	0.1954	0.1936	0.5016	0.5046	0.4017	0.3877	0.1888	0.2654
13	0.3892	0.3815	0.1596	0.1856	0.1871	0.4982	0.5014	0.3920	0.3787	0.1748	0.2522
14	0.3828	0.3727	0.1398	0.1824	0.1852	0.4645	0.4628	0.3871	0.3764	0.1691	0.2415
15	0.3779	0.3591	0.1343	0.1748	0.1836	0.4627	0.4558	0.3814	0.3556	0.1660	0.2314
16	0.3762	0.3546	0.1288	0.1726	0.1818	0.4472	0.4473	0.3688	0.3499	0.1627	0.2215
17	0.3735	0.3485	0.1238	0.1619	0.1775	0.4396	0.4372	0.3616	0.3463	0.1614	0.2119
18	0.3719	0.3469	0.1204	0.1607	0.1750	0.4079	0.4271	0.3596	0.3401	0.1592	0.2100
19	0.3652	0.3463	0.1113	0.1533	0.1719	0.4004	0.4221	0.3537	0.3363	0.1573	0.2045
20	0.3634	0.3461	0.1050	0.1510	0.1637	0.3978	0.4161	0.3390	0.3336	0.1571	0.1921
21	0.3624	0.3330	0.1026	0.1481	0.1563	0.3964	0.4133	0.3365	0.3223	0.1569	0.1794
22	0.3605	0.3242	0.1012	0.1430	0.1550	0.3953	0.4124	0.3275	0.3185	0.1560	0.1768
23	0.3588	0.3206	0.1009	0.1425	0.1530	0.3949	0.4048	0.3196	0.3101	0.1541	0.1733
24	0.3580	0.3188	0.1007	0.1406	0.1514	0.3936	0.3671	0.3116	0.2989	0.1531	0.1722
25	0.3365	0.3024	0.1006	0.1202	0.1504	0.3751	0.3645	0.3057	0.2954	0.1420	0.1693

Table 6.42: Normalized intra-cluster distances with $intra_{base}$ equal to $intra_2$.

rule we are trying to satisfy says that when the error in the segmented image is small enough and the segmented image is not changing much, we can stop.

Table 6.44 shows the number of clusters obtained when using mean cluster centres and Euclidean distance for various values for $perc_1$ and $perc_2$, where $perc_1$ represents the threshold for the normalized intra-cluster distance and $perc_2$ represents the threshold for the percentage decrease in intra-cluster distance. Similarly, Table 6.45 shows the number of clusters obtained when using mean cluster centres and absolute distance, Table 6.46 shows the number of clusters obtained when using median cluster centres and Euclidean distance, and Table 6.47 shows the number of clusters obtained when using median cluster centres and absolute distance.

As can be seen in Tables 6.44-6.47, increasing the value of $perc_1$ while keeping $perc_2$ constant has the effect of decreasing the number of clusters obtained, and increasing $perc_2$ while keeping $perc_1$ constant has the effect of decreasing the number of clusters. This is not always the case, and in fact increasing either of the values could still lead

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.5061	0.3814	0.7887	0.3165	0.4456	0.3620	0.5593	0.2590	0.4011	0.4996	0.4581
3	0.1769	0.2328	0.2539	0.3658	0.4652	0.2483	0.1238	0.2081	0.1979	0.1552	0.3532
4	0.2817	0.2244	0.2148	0.1171	0.2396	0.0429	0.0569	0.1752	0.1798	0.5100	0.2360
5	0.0517	0.1196	0.1907	0.4035	0.1134	0.0189	0.1462	0.1110	0.1309	0.2615	0.1209
6	0.1000	0.0605	0.2596	0.0861	0.0491	0.1466	0.0185	0.0759	0.1152	0.1557	0.0915
7	0.0265	0.0556	0.1591	0.0833	0.1632	0.0722	0.0113	0.0403	0.0302	0.0785	0.0662
8	0.0461	0.0485	0.1948	0.1478	0.0180	0.0294	0.0081	0.0738	0.0775	0.0751	0.0551
9	0.0196	0.0501	0.0700	0.0676	0.0910	0.0253	0.0028	0.0446	0.0117	0.0541	0.0405
10	0.0528	0.0121	0.0696	0.0346	0.1012	0.0050	0.1431	0.0447	0.0747	0.0460	0.0145
11	0.0071	0.0049	0.0892	0.0553	0.0906	0.0085	0.1243	0.0376	0.0165	0.0247	0.1381
12	0.0860	0.0516	0.0985	0.0366	0.0751	0.0386	0.0070	0.0410	0.0476	0.0247	0.0650
13	0.0144	0.0259	0.0549	0.0502	0.0336	0.0067	0.0062	0.0241	0.0233	0.0740	0.0497
14	0.0166	0.0232	0.1237	0.0168	0.0105	0.0677	0.0772	0.0126	0.0061	0.0329	0.0423
15	0.0128	0.0366	0.0396	0.0419	0.0084	0.0039	0.0150	0.0148	0.0553	0.0186	0.0419
16	0.0044	0.0125	0.0408	0.0123	0.0097	0.0336	0.0186	0.0329	0.0160	0.0194	0.0428
17	0.0072	0.0173	0.0390	0.0620	0.0236	0.0169	0.0227	0.0197	0.0101	0.0080	0.0435
18	0.0043	0.0044	0.0273	0.0078	0.0141	0.0720	0.0231	0.0054	0.0181	0.0140	0.0090
19	0.0180	0.0017	0.0762	0.0456	0.0182	0.0186	0.0116	0.0164	0.0110	0.0116	0.0262
20	0.0048	0.0007	0.0566	0.0155	0.0477	0.0064	0.0144	0.0415	0.0081	0.0018	0.0606
21	0.0028	0.0380	0.0224	0.0192	0.0452	0.0036	0.0067	0.0076	0.0339	0.0011	0.0663
22	0.0053	0.0262	0.0134	0.0344	0.0080	0.0028	0.0021	0.0265	0.0119	0.0058	0.0144
23	0.0045	0.0113	0.0035	0.0036	0.0131	0.0009	0.0186	0.0242	0.0262	0.0122	0.0194
24	0.0024	0.0054	0.0014	0.0132	0.0106	0.0033	0.0931	0.0253	0.0363	0.0062	0.0067
25	0.0600	0.0515	0.0013	0.1452	0.0065	0.0470	0.0072	0.0187	0.0114	0.0725	0.0168

Table 6.43: Percentage decrease in intra-cluster distances.

to the same number of clusters, most noticeable for the images *molecule* and *mouse*, while for *teapot* and *ant* the same number of clusters were produced for most of the values of $perc_1$ while $perc_2$ was constant, where a difference occurred only for one of the values for $perc_1$ shown in Tables 6.44-6.47. The reason that there was no difference in the number of clusters achieved for these images is that the number of clusters obtained matches with the first occurrence of the percentage decrease in intra-cluster distance which is below the value of $perc_2$. Following is a detailed analysis of the

$perc_1$	0.30	0.40	0.50	0.60	0.30	0.40	0.50	0.60
$perc_2$	0.05	0.05	0.05	0.05	0.10	0.10	0.10	0.10
balls	41	12	7	6	41	12	7	4
Lenna	26	12	7	7	26	12	6	5
molecule	14	14	14	14	8	8	8	8
teapot	9	9	9	9	7	5	5	5
ant	7	5	5	5	7	5	5	5
blond	46	20	13	7	46	20	13	7
jet	56	24	14	11	56	24	14	11
mandrill	27	13	8	6	27	13	8	5
peppers	24	12	7	6	24	12	7	6
mouse	9	9	9	9	6	6	6	6
rose	12	8	8	8	11	6	5	5

Table 6.44: Number of clusters using stopping rule 2 with mean cluster centres and Euclidean distance.

$perc_1$	0.30	0.40	0.50	0.60	0.30	0.40	0.50	0.60
$perc_2$	0.05	0.05	0.05	0.05	0.10	0.10	0.10	0.10
balls	32	10	6	6	32	10	6	4
Lenna	23	11	7	7	23	11	6	5
molecule	8	8	8	8	8	8	8	8
teapot	9	9	9	5	9	9	8	4
ant	8	5	5	5	7	5	5	5
blond	41	19	14	7	41	19	14	6
jet	42	19	11	8	42	19	11	8
mandrill	30	13	8	6	30	13	8	6
peppers	24	10	9	9	24	10	6	6
mouse	10	10	10	10	6	6	6	6
rose	11	10	10	10	11	7	6	6

Table 6.45: Number of clusters using stopping rule 2 with mean cluster centres and absolute distance.

$perc_1$	0.30	0.40	0.50	0.60	0.30	0.40	0.50	0.60
$perc_2$	0.05	0.05	0.05	0.05	0.10	0.10	0.10	0.10
balls	11	7	7	7	11	5	4	4
Lenna	28	13	9	5	28	13	7	5
molecule	13	13	13	13	8	8	8	8
teapot	8	8	8	8	8	5	5	5
ant	9	9	9	9	7	6	6	6
blond	43	31	17	8	43	31	17	8
jet	75	30	11	10	75	30	11	8
mandrill	31	13	8	5	31	13	8	5
peppers	27	14	7	7	27	14	7	6
mouse	9	9	9	9	6	6	6	6
rose	13	7	7	7	12	6	5	5

Table 6.46: Number of clusters using stopping rule 2 with median cluster centres and Euclidean distance.

results for each image. The results are summarized in section 6.3.2.

For the image *balls*, the best results occurred when using 0.40 for $perc_1$ and either 0.05 or 0.10 for $perc_2$ with mean cluster centres and either distance. When using median cluster centres with Euclidean distance a value of 0.30 for $perc_1$ and 0.05 or 0.10 for $perc_2$ produced the best result, while when using median cluster centres with absolute distance all of the values 0.30-0.60 for $perc_1$ together with the value 0.05 for $perc_2$ produced the same number of clusters which was the best result in this case. The resulting segmentations for the four approaches were all based on a similar number of clusters ranging from 10 to 12, however, the best overall segmentation was produced by the mean cluster centres and absolute distance approach which resulted in smoother and more connected segments.

$perc_1$	0.30	0.40	0.50	0.60	0.30	0.40	0.50	0.60
$perc_2$	0.05	0.05	0.05	0.05	0.10	0.10	0.10	0.10
balls	10	10	10	10	8	5	5	5
Lenna	24	11	6	6	24	11	6	5
molecule	8	8	8	8	8	8	8	8
teapot	9	9	9	9	8	6	6	6
ant	10	10	10	10	8	5	4	4
blond	32	17	13	7	32	17	12	7
jet	45	15	11	8	45	15	11	8
mandrill	26	14	8	6	26	14	8	5
peppers	24	10	8	8	24	10	7	5
mouse	9	9	9	9	6	6	6	6
rose	10	8	8	8	10	7	7	7

Table 6.47: Number of clusters using stopping rule 2 with median cluster centres and absolute distance.

For the image *Lenna* the best results were achieved when $perc_1$ was set to 0.50 and $perc_2$ was set to 0.10 when Euclidean distance was used with either cluster centre type. When absolute distance was used with either cluster centre type, the best results were achieved when $perc_1$ was set to 0.50 or 0.60 and $perc_2$ was set to 0.05, with the median cluster centres and absolute distance approach also achieving its best result with $perc_1$ set to 0.50 and $perc_2$ set to 0.10. The overall best result was produced when using mean cluster centres and Euclidean distance.

For the image *molecule*, for each of the four approaches, the best results occurred when $perc_2$ was 0.10 and $perc_1$ was any of the values in Tables 6.44-6.47 and also for the value of 0.05 for $perc_2$ with any of the values for $perc_1$ shown in Tables 6.37-6.40 when absolute distance was used with either cluster centre type. All of the best results from the four approaches were virtually identical, so no preference can be given to any of the approaches.

For *teapot*, the best results occurred when using the values 0.30-0.50 for $perc_1$ and 0.05 for $perc_2$ for all of the four approaches. The value 0.60 for $perc_1$ in combination with 0.05 for $perc_2$ also produced good results for all except the mean cluster centres and absolute distance approach. A good result also occurred when $perc_1$ was 0.30 and $perc_2$ was 0.10 when using median cluster centres and Euclidean distance, and

also when using mean cluster centres and absolute distance, setting $perc_1$ to 0.30-0.40 with $perc_2$ set to 0.10 also produced a good result. The best overall result occurred when using mean cluster centres and Euclidean distance.

For the image *ant*, when using mean cluster centres with either distance, the best results occurred when $perc_1$ was in the range 0.40-0.60 in combination with either 0.05 or 0.10 for $perc_2$. When using median cluster centres with either distance, good results were achieved when 0.10 was used for $perc_2$, in combination with values of 0.40-0.60 for $perc_1$ when Euclidean distance was used and 0.40 for $perc_1$ when absolute distance was used. All of these results were very similar, however, the best result occurred when using mean cluster centres with Euclidean distance.

For the image *blond*, good results were obtained when using 0.60 for $perc_1$ and either 0.05 or 0.10 for $perc_2$ for each of the four combinations of cluster centre and distance type, except that the mean cluster centres and absolute distance combination obtained its best result when only using 0.05 for $perc_2$. All of the results for the four approaches were very similar, and no preference can be given to any of the approaches.

For the threshold values shown in Tables 6.44-6.47, the image *jet* obtained its best results when 0.60 was used for $perc_1$ and 0.05 or 0.10 was used for $perc_2$. This was true regardless of which cluster centre or distance type was used. These results produced quite a large number of clusters for this image, and a more suitable result could be achieved if 0.70 was used for $perc_1$ where the number of clusters produced when using mean cluster centre with either distance was 6 and the number of clusters when using median cluster centres and absolute distance was 5 for both values of $perc_2$. Increasing the value of $perc_1$ to 0.70 had no effect on the number of clusters produced by the median cluster centres and Euclidean distance approach. All of the results for the four approaches were very similar and no preference can be given to any of the

approaches.

For the image *mandrill*, good results were obtained when using either 0.05 or 0.10 for $perc_2$ and 0.50 for $perc_1$, regardless of which cluster centre or distance type was used. All of the results for the four approaches were very similar and no preference can be given to any of the approaches.

For the image *peppers*, the best results when using mean cluster centres and Euclidean distance occurred when $perc_1$ was 0.60 and $perc_2$ was either 0.05 or 0.10. When using mean cluster centres and absolute distance, the best result was achieved when using the values 0.50-0.60 for $perc_1$ with 0.10 for $perc_2$. When using median cluster centres with Euclidean distance, the best results were obtained when $perc_1$ was set to 0.50-0.60 and $perc_2$ was set to 0.05 and when $perc_1$ was set to 0.50 and $perc_2$ was set to 0.10. For median cluster centres and absolute distance the best result was obtained when $perc_1$ was 0.50 and $perc_2$ was 0.10. All of the results for the four approaches were very similar and no preference can be given to any of the approaches.

For the image *mouse*, for a given value for $perc_2$ all the results for the different values for $perc_1$ were identical. The best results occurred when $perc_2$ was 0.10. The results when using a value of 0.05 for $perc_2$ were also good, but had a slightly higher number of clusters. All of the results for the four approaches were very similar and no preference can be given to any of the approaches.

For the image *rose*, the best results occurred when 0.40-0.60 was used for $perc_1$ and $perc_2$ was either 0.05 for all except the median cluster centres and absolute distance combination, where the best result occurred for the same values for $perc_1$ but only when $perc_2$ was set to 0.10. All of the results for the four approaches were very similar and no preference can be given to any of the approaches.

Figures 6.20-6.23 show the best segmented results for each of the images when using mean cluster centres and Euclidean distance, mean cluster centres and absolute distance, median cluster centres and Euclidean distance and median cluster centres and absolute distance, respectively, with the values of $perc_1$ and $perc_2$ producing these results indicated in the figures. Generally, setting $perc_1$ equal to 0.40 and $perc_2$ equal to 0.05 works quite well. The main exceptions to this occurred for the images *Lenna*, *blond*, *jet*, *mandrill* and *peppers* which required larger values for $perc_1$. Considering there are cases where the results are not that good when using the values 0.40 for $perc_1$ and 0.05 for $perc_2$, there may be some cases where the number of clusters should be smaller or bigger. To achieve this all that is required is to change the values of $perc_1$ and $perc_2$. If we wanted to decrease the number of clusters, we could increase either $perc_1$ or $perc_2$, or both, and to increase the number of clusters, we can decrease $perc_1$ or $perc_2$, or both. However, the values 0.40 and 0.05 are good starting points, and can be considered default values for this stopping rule.

6.3.1.3 Results Using Context

Each of the results incorporating the context of the image are based on using only mean cluster centres and Euclidean distance and the use of the RGB colour space. Also, comparisons made to the results by the original RGB values refer to the results using mean cluster centres and Euclidean distance presented in section 6.3.1.2.

Tables 6.48-6.51 show the number of clusters obtained, when using the *mean*, *median*, *weighted* and *regression* context values only, respectively, for various values for $perc_1$ and $perc_2$, where $perc_1$ represents the threshold for the normalized intra-cluster distance and $perc_2$ represents the threshold for the percentage decrease in intra-cluster distance. Following is a detailed analysis of the results for each image.

The results are summarized in section 6.3.2.

$perc_1$	0.30	0.40	0.50	0.60	0.30	0.40	0.50	0.60
$perc_2$	0.05	0.05	0.05	0.05	0.10	0.10	0.10	0.10
balls	14	9	9	9	14	8	6	4
Lenna	17	9	8	8	17	9	6	5
molecule	12	12	12	12	8	8	8	8
teapot	9	9	9	9	8	5	5	5
ant	12	12	12	12	6	6	6	6
blond	25	13	8	7	25	13	8	6
jet	18	10	9	9	18	10	7	7
mandrill	20	11	8	8	20	11	7	5
peppers	20	11	7	7	20	11	7	7
mouse	10	10	10	10	6	6	6	6
rose	10	9	9	9	10	6	5	5

Table 6.48: Number of clusters using stopping rule 2 with *mean* context values only.

$perc_1$	0.30	0.40	0.50	0.60	0.30	0.40	0.50	0.60
$perc_2$	0.05	0.05	0.05	0.05	0.10	0.10	0.10	0.10
balls	19	9	9	9	19	8	6	4
Lenna	18	9	6	6	18	9	6	5
molecule	11	11	11	11	8	8	8	8
teapot	8	8	8	8	7	5	5	5
ant	10	10	10	10	6	6	6	6
blond	26	13	8	5	26	13	8	5
jet	49	12	10	10	49	12	8	8
mandrill	21	12	9	9	21	11	7	5
peppers	23	10	7	7	23	10	7	7
mouse	9	9	9	9	6	6	6	6
rose	10	10	10	10	10	6	5	5

Table 6.49: Number of clusters using stopping rule 2 with *median* context values only.

For the image *balls*, the best results occurred when using 0.30 for $perc_1$ and either 0.05 or 0.10 for $perc_2$ for the *mean* and *weighted* context types. When using the *median* or *regression* contexts, a value of 0.40-0.60 for $perc_1$ and 0.05 for $perc_2$ produced the best results. The overall best results occurred when using the *mean* and *weighted* contexts as the red ball was correctly classified as a separate segment. However, these two results contained a higher number of clusters, although the results by the *median* and *regression* contexts did not show such a good classification of the red ball with the same number of clusters. In comparison to the results using the original RGB values, the context results show more smoother and connected clusters which is a clear advantage as each ball should be classified as one segment.

$perc_1$	0.30	0.40	0.50	0.60	0.30	0.40	0.50	0.60
$perc_2$	0.05	0.05	0.05	0.05	0.10	0.10	0.10	0.10
balls	14	9	9	9	14	8	5	4
Lenna	17	9	8	8	17	9	6	5
molecule	10	10	10	10	8	8	8	8
teapot	9	9	9	9	8	5	5	5
ant	12	12	12	12	6	6	6	6
blond	25	13	9	9	25	13	8	6
jet	17	11	9	9	17	11	7	7
mandrill	20	11	8	8	20	11	7	5
peppers	21	11	7	7	21	11	7	7
mouse	10	10	10	10	6	6	6	6
rose	10	9	9	9	10	6	5	5

Table 6.50: Number of clusters using stopping rule 2 with *weighted* context values only.

$perc_1$	0.30	0.40	0.50	0.60	0.30	0.40	0.50	0.60
$perc_2$	0.05	0.05	0.05	0.05	0.10	0.10	0.10	0.10
balls	15	11	11	11	15	9	6	4
Lenna	16	10	8	8	16	10	6	5
molecule	11	11	11	11	8	8	8	8
teapot	8	8	8	8	8	5	5	5
ant	11	11	11	11	6	6	6	6
blond	26	14	9	6	26	14	9	6
jet	23	12	10	6	23	12	9	6
mandrill	23	11	8	8	23	11	7	5
peppers	21	11	7	7	21	11	7	7
mouse	10	10	10	10	6	6	6	6
rose	10	6	6	6	10	6	6	6

Table 6.51: Number of clusters using stopping rule 2 with *regression* context values only.

For the image *Lenna* the best results were achieved when $perc_1$ was set to 0.40 and $perc_2$ was set to 0.05 or 0.10 for each of the four context types. Each of the results by the four contexts were very similar in appearance, however, the *median* context grouped part of the feathers together with the background regions, while the *regression* context showed a better separation of the feathers from any region in the background. In comparison to the results using the original RGB values, the context results were more smoother, and looked a little blurry.

For the image *molecule*, for each of the four context types, the best results occurred when $perc_1$ was 0.30-0.60 and $perc_2$ was 0.10. Each of these results contained the same number of clusters, 8, and looked very similar, except that when using the *mean*, *weighted* and *regression* contexts, the edges of the molecule were a little fuzzy, as opposed to the results by the *median* context and the original RGB values, which

had much sharper edges in the segmented images. Apart from this, all of the results, including the best result when using only the original RGB values, looked very similar.

For *teapot*, the best results occurred when using the values 0.30-0.60 for $perc_1$ and 0.05 for $perc_2$ for all of the four context types, as well as 0.30 for $perc_1$ and 0.10 for $perc_2$ when using the *regression* context. The best results occurred when using the *mean* or *weighted* contexts as the reflection region was represented better, however, the shadow of the teapot was grouped as one colour rather than showing the individual colours of the squares in the checkerboard. The *regression* context result contained some blue in the reflection region, especially as an outline of the dark squares. The *median* context result also showed this phenomenon, however, the shadow regions correctly distinguished between the light and dark squares. In comparison to the best result using the original RGB values, the context values had less shades for the teapot, despite having the same or one less number of clusters, however, the shadow region was handled better when using only the original RGB values.

For the image *ant*, the best results for each of the four context types occurred when $perc_1$ was 0.30-0.60 and $perc_2$ was 0.10. All of these results were very similar. In comparison to the results using the original RGB values, the context results had more smooth looking segments.

For the image *blond*, good results were obtained when using 0.50 for $perc_1$ and either 0.05 or 0.10 for $perc_2$ for the *mean*, *median* and *regression* contexts. For the *weighted* context, the best result occurred when $perc_1$ was 0.50-0.60 and $perc_2$ was 0.05. Each of these results were very similar, except that when the *median* context was used, more of the lips were grouped together. These context results were much better than the results using only the original RGB values, as these results correctly

identified the hair as a separate cluster in yellow, which did not occur when using only the original RGB values for an equivalent number of clusters.

For the image *jet*, the best results occurred when $perc_1$ was 0.50-0.60 and $perc_2$ was 0.05 for the *mean*, *median* and *weighted* contexts, while for the *regression* context, the best result was obtained when $perc_1$ was 0.50 and $perc_2$ was 0.05. The overall best result was obtained when using the *regression* context, as the F-16 symbol was clearer. In comparison to the results using the original RGB values, the context results, especially for the *mean*, *median* and *weighted* contexts, were blurry and generally more smoother looking, while the writing on the plane was much clearer when using only the original RGB values.

For the image *mandrill*, the best results for each of the four context types were obtained when $perc_1$ was 0.40 and $perc_2$ was 0.05, and for the *mean*, *weighted* and *regression* contexts, setting $perc_2$ to 0.10 was also good. The results were all very similar, except when using the *median* context for which the right eye was partly classified together with the green fur segment. In comparison to the results using the original RGB values, the context results appeared a little blurred.

For the image *peppers*, the best results for each of the four context types were obtained when $perc_1$ was 0.50-0.60 and $perc_2$ was either 0.05 or 0.10. All of the results were very similar, however, the *median* context result was slightly better as the other results all missed a little part of the red pepper on the right and grouped it with the background. In comparison to the results using the original RGB values, the context results had more smoother looking segments.

For the image *mouse*, the best results occurred when $perc_1$ was 0.30-0.60 and $perc_2$ was 0.10. The results when using a value of 0.05 for $perc_2$ were also good, but had a slightly higher number of clusters. Each of these four results were very similar in

appearance, however, the *mean* and *weighted* contexts were able to classify more of the left ear as the same segment. In comparison to the results using only the original RGB values, the context results had more smoother segments.

For the image *rose*, the best results for the *mean* and *weighted* contexts occurred when 0.40-0.60 was used for $perc_1$ and $perc_2$ was 0.05. For the *median* context, the best result occurred when $perc_1$ was 0.30-0.60 and $perc_2$ was 0.05 and also when $perc_1$ was 0.30 and $perc_2$ was 0.10. The best result for the *regression* context occurred when $perc_1$ was 0.30 and $perc_2$ was 0.05 or 0.10. Each of these results were very similar from a visual point of view. In comparison to the results using the original RGB values, the context results showed more smoother clusters.

Figures 6.24-6.27 show the segmentation results when using only the *mean*, *median*, *weighted* or *regression* contexts for various values of $perc_1$ and $perc_2$ which produced the best visual segmentation results, respectively, with the values of $perc_1$ and $perc_2$ producing these results indicated in the figures. Generally, the context results tended to be slightly blurred and also contained more smooth looking segments. This is not surprising since when we apply the context operation to the original image, the resulting image is a slightly blurred version of the original since we are using the eight neighbours in calculating new colour values. Overall, the *mean* and *weighted* contexts performed well with good results occurring for most images. The *regression* context was also quite good, and the *median* context only produced good results for half of the images. It is very difficult to prefer one context type over the others as there was a lot a variation in the results. As with using only the original RGB values, generally setting $perc_1$ equal to 0.40 and $perc_2$ equal to 0.05 worked quite well. The main exception to this occurred for the images *blond*, *jet* and *peppers*, which required $perc_1$ to be at least 0.50 and *molecule*, *ant* and *mouse* required $perc_2$ to be 0.10. However,

the values 0.40 and 0.05 are good starting points for this method.

Tables 6.52-6.55 show the number of clusters obtained, when using the *mean*, *median*, *weighted* and *regression* context values together with the original RGB values, respectively, for various values for $perc_1$ and $perc_2$, where $perc_1$ represents the threshold for the normalized intra-cluster distance and $perc_2$ represents the threshold for the percentage decrease in intra-cluster distance. Following is a detailed analysis of the results for each image. The results are summarized in section 6.3.2.

$perc_1$	0.30	0.40	0.50	0.60	0.30	0.40	0.50	0.60
$perc_2$	0.05	0.05	0.05	0.05	0.10	0.10	0.10	0.10
balls	27	10	9	9	27	10	7	4
Lenna	35	12	7	7	35	12	7	5
molecule	11	8	8	8	10	8	6	5
teapot	9	9	9	9	8	5	5	5
ant	11	11	11	11	7	6	6	6
blond	71	23	10	7	71	23	10	6
jet	64	20	11	8	64	20	11	7
mandrill	77	24	11	6	77	24	11	6
peppers	38	15	7	7	38	15	7	5
mouse	10	10	10	10	6	6	6	6
rose	11	7	6	6	11	7	5	5

Table 6.52: Number of clusters using stopping rule 2 with original and *mean* context values.

$perc_1$	0.30	0.40	0.50	0.60	0.30	0.40	0.50	0.60
$perc_2$	0.05	0.05	0.05	0.05	0.10	0.10	0.10	0.10
balls	33	10	9	4	33	10	7	4
Lenna	33	13	8	8	33	13	7	5
molecule	12	12	12	12	9	9	6	6
teapot	9	9	9	9	7	5	5	5
ant	10	10	10	10	7	6	6	6
blond	79	24	11	6	79	24	11	6
jet	87	23	11	8	87	23	11	7
mandrill	85	24	11	6	85	24	11	6
peppers	40	15	7	7	40	15	7	6
mouse	9	9	9	9	6	6	6	6
rose	11	7	7	7	11	6	5	5

Table 6.53: Number of clusters using stopping rule 2 with original and *median* context values.

For the image *balls*, the best results occurred when using 0.40 for $perc_1$ and either 0.05 or 0.10 for $perc_2$ for the *mean*, *median* and *weighted* context types. When using the *regression* context, a value of 0.40-0.60 for $perc_1$ and 0.05 for $perc_2$ or 0.40 for $perc_1$ and 0.10 for $perc_2$ produced the best result. Each of these results contained

$perc_1$	0.30	0.40	0.50	0.60	0.30	0.40	0.50	0.60
$perc_2$	0.05	0.05	0.05	0.05	0.10	0.10	0.10	0.10
balls	21	10	9	9	21	10	7	4
Lenna	34	12	7	7	34	12	7	5
molecule	12	8	8	8	10	8	6	5
teapot	9	9	9	9	8	5	5	5
ant	11	11	11	11	7	6	6	6
blond	76	23	10	7	76	23	10	6
jet	62	20	11	8	62	20	11	7
mandrill	78	23	11	6	78	23	11	6
peppers	39	15	7	7	39	15	7	5
mouse	10	10	10	10	6	6	6	6
rose	11	7	6	6	11	7	5	5

Table 6.54: Number of clusters using stopping rule 2 with original and *weighted* context values.

$perc_1$	0.30	0.40	0.50	0.60	0.30	0.40	0.50	0.60
$perc_2$	0.05	0.05	0.05	0.05	0.10	0.10	0.10	0.10
balls	19	10	10	10	19	10	7	4
Lenna	34	13	7	7	34	13	7	5
molecule	13	13	13	13	8	8	8	8
teapot	9	9	9	9	8	5	5	5
ant	11	11	11	11	7	6	6	6
blond	61	20	10	8	61	20	10	7
jet	69	21	11	9	69	21	11	7
mandrill	81	23	11	7	81	23	11	6
peppers	33	13	7	7	33	13	7	5
mouse	10	10	10	10	6	6	6	6
rose	11	6	6	6	11	6	5	5

Table 6.55: Number of clusters using stopping rule 2 with original and *regression* context values.

the same number of clusters and also looked very similar from a visual point of view. In comparison to the results using only the original RGB values, the results using both the original and context values showed more smoother and connected segments which is a clear advantage as each ball should be classified as one segment.

For the image *Lenna* the best results were achieved when $perc_1$ was set to 0.40 and $perc_2$ was set to 0.05 or 0.10 for each of the four context types. Each of the results by the four contexts were very similar in appearance. In comparison to the results using the original RGB values, the results using both the original and context values had more smoother looking segments. While the results when using only the context values were blurry in appearance, the use of the original values as well overcame this effect.

For the image *molecule*, for each of the four context types, the best results occurred

when $perc_2$ was 0.10, however, different values were required for $perc_1$. For the *mean* and *weighted* context, $perc_1$ was required to be 0.30, for the *median* context, $perc_1$ was required to be 0.30-0.40 and for the *regression* context, $perc_1$ was required to be 0.30-0.60. Each of these results were very similar in appearance, and also look similar to the best result using only the original RGB values, however, the *regression* context resulted in the smallest number of clusters at 8, which was the same as the best result when using only the original RGB values.

For *teapot*, the best results occurred when using the values 0.30-0.60 for $perc_1$ and 0.05 for $perc_2$ for all of the four context types. Each of the results were very similar in appearance, and also similar to the best result using only the original RGB values, except that not all the checkerboard squares were visible in the shadow region in the context results.

For the image *ant*, the best results for each of the four context types occurred when $perc_1$ was 0.30 and $perc_2$ was 0.10. All of these results were very similar. In comparison to the results using the original RGB values, the context results had more smooth looking segments.

For the image *blond*, good results were obtained when using 0.50 for $perc_1$ and either 0.05 or 0.10 for $perc_2$ for each of the four context types. Each of these results were very similar. These results using both the original and context values were much better than the results using only the original RGB values, as these results correctly identified the hair as a separate cluster in yellow, which did not occur when using only the original RGB values. Also, the blurring effect which occurred when using only the context values did not occur when using the original RGB values as well.

For the image *jet*, the best results for each of the four context types occurred when $perc_1$ was 0.60 and $perc_2$ was 0.05. In comparison to the results using only the

context values, these results were much better as the blurring effect which made the writing on the plane illegible did not occur for the results using both the original and context values. The results for the *mean*, *weighted* and *regression* contexts were the best as the writing on the plane in the *median* context result was not as clear as the other results. In comparison to the results using the original RGB values, the context results contained more smooth looking clusters, which is especially noticeable in the tail and wing of the plane.

For the image *mandrill*, the best results for each of the four context types were obtained when $perc_1$ was 0.50 and $perc_2$ was 0.05 or 0.10. The results were all very similar. In comparison to the results using the original RGB values, the fur in the context results appeared more smooth looking.

For the image *peppers*, the best results for each of the four context types were obtained when either $perc_1$ was 0.50-0.60 and $perc_2$ was 0.05 or when $perc_1$ was 0.50 and $perc_2$ was 0.10. All of the results were very similar, and in comparison to the results by the original RGB values, the context results had more smoother looking segments.

For the image *mouse*, the best results occurred when $perc_1$ was 0.30-0.60 and $perc_2$ was 0.10. The results when using a value of 0.05 for $perc_2$ were also good, but had a slightly higher number of clusters. Each of these four results were very similar in appearance, however, the *mean* and *weighted* contexts were able to classify more of the left ear as the same segment. In comparison to the results using only the original RGB values, the context results had more smoother segments, but otherwise looked very similar.

For the image *rose*, the best results for each of the four context types occurred when $perc_1$ was 0.30 and $perc_2$ was 0.05 or 0.10. Each of these results were very

similar from a visual point of view. In comparison to the results using the original RGB values, the context results showed more smoother looking clusters.

Figures 6.28-6.31 show the segmentation results for the *mean*, *median*, *weighted* and *regression* contexts, when used together with the original colour values, for various values of $perc_1$ and $perc_2$ which produced the best visual segmentation results, respectively, with the values of $perc_1$ and $perc_2$ producing these results indicated in the figures. Generally, the context results tended to contain more smooth looking segments, which is an advantage over only using the original RGB values as we expect segments to be connected regions in the image. Overall, the *regression* context performed the best, producing good results for all of the images. The *mean*, *median* and *weighted* contexts also performed well with good results occurring for most images. For most of the cases, the results by the four context types were very similar, and no preference can be given to any one particular context type over the others. Generally, setting $perc_1$ equal to 0.40-0.50 and $perc_2$ equal to 0.05 worked quite well. The main exceptions to this occurred for the images *ant* and *rose*, which required $perc_1$ to be 0.30, and *jet*, which required $perc_1$ to be 0.60. For *molecule*, *ant* and *mouse*, $perc_2$ was required to be 0.10.

6.3.1.4 Results Using Different Colour Spaces

All of the results in this section are based on only the use of the average variance in calculating the threshold and determining which cluster to split and also when only using mean cluster centres and Euclidean distance. The results using the RGB colour space with mean cluster centres and Euclidean distance were presented in section 6.3.1.2.

Tables 6.56-6.58 show the number of clusters obtained, when using the XYZ, KL

and HSV colour spaces, respectively, for various values for $perc_1$ and $perc_2$, where $perc_1$ represents the threshold for the normalized intra-cluster distance and $perc_2$ represents the threshold for the percentage decrease in intra-cluster distance. Following is a detailed analysis of the results for each image. The results are summarized in section 6.3.2.

$perc_1$	0.30	0.40	0.50	0.60	0.30	0.40	0.50	0.60
$perc_2$	0.05	0.05	0.05	0.05	0.10	0.10	0.10	0.10
balls	24	9	6	6	24	9	6	4
Lenna	16	10	6	6	16	10	5	5
molecule	13	13	13	13	7	7	7	7
teapot	9	9	9	9	6	6	4	4
ant	7	7	7	7	6	6	6	6
blond	32	19	14	6	32	19	14	6
jet	38	16	8	8	38	16	8	6
mandrill	25	13	9	9	25	13	8	6
peppers	19	11	7	5	19	11	7	5
mouse	10	10	10	10	6	6	6	6
rose	9	9	9	9	9	7	7	7

Table 6.56: Number of clusters using stopping rule 2 with the XYZ colour space.

$perc_1$	0.30	0.40	0.50	0.60	0.30	0.40	0.50	0.60
$perc_2$	0.05	0.05	0.05	0.05	0.10	0.10	0.10	0.10
balls	45	13	7	7	45	13	7	4
Lenna	24	12	7	7	24	12	6	5
molecule	14	14	14	14	8	8	8	8
teapot	10	6	6	6	8	6	6	6
ant	10	10	10	10	7	6	6	6
blond	47	19	12	7	47	19	12	7
jet	55	21	14	10	55	21	14	10
mandrill	31	13	8	7	31	13	8	5
peppers	27	13	7	7	27	13	7	6
mouse	9	9	9	9	6	6	6	6
rose	11	7	7	7	11	6	5	5

Table 6.57: Number of clusters using stopping rule 2 with the KL colour space.

For the image *balls*, the best results occurred when using 0.40 for $perc_1$, and either 0.05 or 0.10 for $perc_2$ for the XYZ, KL and the RGB colour spaces. When using the HSV colour space a value of 0.50 for $perc_1$ and 0.05 or 0.10 for $perc_2$ produced the best result, although this result contained many more clusters than for any other colour space. The overall best result occurred when using the KL colour space for which the pink ball was represented as a more connected region than for the best

$perc_1$	0.30	0.40	0.50	0.60	0.30	0.40	0.50	0.60
$perc_2$	0.05	0.05	0.05	0.05	0.10	0.10	0.10	0.10
balls	51	29	18	12	51	29	18	12
Lenna	22	14	10	10	22	14	9	9
molecule	5	5	5	5	4	4	4	4
teapot	9	9	9	9	7	6	4	4
ant	14	8	8	8	14	8	6	6
blond	41	24	19	15	41	24	19	15
jet	12	8	8	6	12	8	8	6
mandrill	31	17	11	7	31	17	11	7
peppers	21	11	7	7	21	11	7	5
mouse	12	12	5	4	11	9	5	4
rose	12	10	7	5	12	9	7	5

Table 6.58: Number of clusters using stopping rule 2 with the HSV colour space.

RGB result where the pink ball was represented by a mixture of two colours, pink and grey. The XYZ result failed to distinguish the red ball as a separate cluster and also failed to distinguish between the pink ball and the shadows of the balls.

For the image *Lenna* the best results were achieved when $perc_1$ was set to 0.50 or 0.60 and $perc_2$ was set to 0.05, for the KL and HSV colour spaces, while the RGB colour space achieved its best result when $perc_1$ was 0.50 and $perc_2$ was 0.10. When the XYZ colour space was used a value of 0.40 for $perc_1$ and 0.05 or 0.10 for $perc_2$ produced a good result. The HSV result was quite noisy in appearance, while the XYZ result was reasonable. However, the results by the KL and RGB colour spaces were the overall best, both of which contained the smallest number of clusters, and which still adequately represented the major regions in this image.

For the image *molecule*, the best results were obtained by the RGB and KL colour spaces when using any of the values 0.30-0.60 for $perc_1$ and a value of 0.10 for $perc_2$. The best result for the XYZ colour space occurred for the values 0.30-0.60 for $perc_1$ and 0.05 for $perc_2$, however, this result contained five extra clusters compared to the RGB or KL results, without adding anything of value to the result. The HSV results were very poor with the maximum number of clusters being produced by the various values for $perc_1$ and $perc_2$ being 5 which is too small to adequately represent the

various colours in the molecule. However, even if the number of clusters is increased for the HSV colour space results, the segmented results still failed to capture the individual colours in the molecule.

For *teapot*, each of the four colour spaces required $perc_2$ to be 0.05. However, the RGB, XYZ and HSV colour spaces produced their best results when $perc_1$ was any value in the range 0.30-0.60, while the KL colour space only produced its best result when $perc_1$ was set to 0.30. The overall best results occurred when using the KL colour space for which the teapot contained less shades and the reflection of the teapot was also classified correctly as having only two shades, depending on which square in the checkerboard the reflection was in. The RGB result was also acceptable, however, the XYZ result included some blue in the reflection region and the HSV result combined the whole of the shadow of the teapot together with the dark squares of the checkerboard.

For the image *ant*, the values between 0.40 and 0.60 for $perc_1$ were good for the RGB, KL and HSV colour spaces, with a value of 0.10 for $perc_2$ when the KL colour space was used, 0.05 for $perc_2$ when the HSV colour space was used and either 0.05 or 0.10 for $perc_2$ when the RGB colour space was used, producing the best results. Additionally, when using 0.40 for $perc_1$ and 0.10 for $perc_2$ when the HSV colour space was used was also good. For the XYZ colour space the values 0.30-0.60 for $perc_1$ and 0.10 for $perc_2$ produced the best results, which happened to be the overall best result, since the bottom part of the mouse was classified together with the mouse region, while for the KL colour space and especially the RGB colour space, part of the bottom of the mouse was grouped together with the shadow region. The HSV result was quite noisy in appearance.

For the image *blond*, good results were obtained when using 0.60 for $perc_1$ and

either 0.05 or 0.10 for $perc_2$ for each of the four colour spaces. The results by the RGB, XYZ and KL colour spaces were all very similar, and much better than the result obtained by the HSV colour space, which despite having many more clusters, failed to distinguish between the hair, hands and face and treated the background as more than one segment. The only good thing about this result is the fact that it successfully captured the lips as a separate segment, however, this could be accomplished with any of the other colour spaces with a higher number of clusters, which isn't really necessary for this image.

For the threshold values shown in Table 6.44 and Tables 6.56-6.58, the image *jet* obtained its overall best results when using the RGB and KL colour spaces with values of 0.60 for $perc_1$ and 0.05 or 0.10 for $perc_2$. However, the number of clusters was slightly high at 11 and 10, respectively. Better results were obtained when using a value of 0.70 for $perc_1$ with either 0.05 or 0.10 for $perc_2$ for both of these colour spaces, which resulted in 6 and 7 clusters for the RGB and KL colour spaces, respectively. The best result for the XYZ colour space was obtained for the values 0.50-0.60 for $perc_1$ and 0.05 for $perc_2$ and 0.50 for $perc_1$ and 0.10 for $perc_2$. However, this result contained a large amount of blue in the tail, which is a red region. The best HSV colour space result was produced when 0.40-0.50 was used for $perc_1$ and 0.05 or 0.10 was used for $perc_2$, however, this result was quite noisy in appearance, especially in the background or snow areas.

For the image *mandrill*, the RGB and KL colour spaces produced their best results when a value of 0.50 was used for $perc_1$ and either 0.05 or 0.10 were used for $perc_2$, while the XYZ and HSV colour spaces produced their best results when a value of 0.40 was used for $perc_1$ and either 0.05 or 0.10 were used for $perc_2$. The overall best results were produced by the RGB and KL colour spaces, with a slight favouring of

the RGB result as the KL result produced some yellow dots in the nose region. The XYZ and HSV results were both very poor as the fur region, which predominantly contains shades of green and black, contained some red.

For the image *peppers*, the best results for the RGB colour space occurred when $perc_1$ was 0.6 and $perc_2$ was 0.05 or 0.10. The KL colour space produced its best result when $perc_1$ was 0.50 and $perc_2$ was either 0.05 or 0.10, and also when $perc_1$ was 0.60 and $perc_2$ was 0.05. The best results when using the XYZ or HSV colour spaces occurred when $perc_1$ was 0.40 and $perc_2$ was either 0.05 or 0.10. The overall best results were obtained by the RGB and KL colour spaces, which both had a smaller number of clusters, but enough to adequately represent the regions in the image.

For the image *mouse*, the best results for the RGB, XYZ and KL colour spaces occurred when using values of 0.30-0.60 for $perc_1$ and 0.10 for $perc_2$. For the HSV colour space, the best result occurred when using the values 0.40 for $perc_1$ and 0.10 for $perc_2$, which contained 9 clusters. For the results containing a higher number of clusters, the background became noisy, and for results containing a lower number of clusters, the bottom right side of the mouse and the black section on the bottom became noisy. The result with 9 clusters was also a little noisy on the right side of the mouse. Therefore, the best results occurred for either the RGB, XYZ or KL colour spaces, which all produced very similar results, although, with the XYZ result, the nose was more clearly visible as a separate colour from the parts surrounding the nose.

For the image *rose*, the best results for the RGB and KL colour spaces occurred when 0.40-0.60 was used for $perc_1$ and 0.05 was used for $perc_2$. For the XYZ colour space, the best result was obtained when using 0.30-0.60 for $perc_1$ and 0.05 for $perc_2$ and when using 0.30 for $perc_1$ and 0.10 for $perc_2$. The best result for the HSV colour

space was obtained when using 0.30 for $perc_1$ and 0.05 or 0.10 for $perc_2$. The overall best results were obtained by the RGB and KL colour spaces, with a slight favouring of the RGB result since the KL result contained a little bit of green in one of the petals. The XYZ result was very poor as part of the rose was classified together with the green in the background, and the HSV result, while containing more clusters, did not give any improvement over the RGB and KL results.

Figures 6.32-6.34 show the segmentation results for the XYZ, KL and HSV colour spaces for various values of $perc_1$ and $perc_2$ which produced the best visual segmentation results, respectively, with the values of $perc_1$ and $perc_2$ producing these results indicated in the figures. Generally, the HSV colour space performed very poorly, at times producing segmentation results which were noisy in appearance. The XYZ colour space was also not performing very well, although good results were obtained for *ant*, *blond* and *mouse*. The RGB and KL colour spaces produced similar results, and mostly for the same values of $perc_1$ and $perc_2$. While the KL colour space gave superior results for *balls* and *teapot*, the remainder of the results were very similar to the RGB results. Therefore, there seems to be no real advantage in choosing a colour space other than RGB for colour image segmentation.

6.3.2 Summary of Results

Tables 6.59-6.62 show the $perc_1$ and $perc_2$ values giving the best segmentations for each of the natural images for the different cluster centre and distance types, the different context types, when used on their own and together with the original colour values, and the different colour spaces, respectively. The result which is visually the best for each image is indicated in bold. The results produced by the $perc_1$ and $perc_2$ values shown in Tables 6.59-6.62 match with the results shown in Figures 6.20-6.34.

image	mean & Euc.		mean & abs.		med. & Euc.		med. & abs.	
	$perc_1$	$perc_2$	$perc_1$	$perc_2$	$perc_1$	$perc_2$	$perc_1$	$perc_2$
balls	0.4	0.05-0.1	**0.4**	**0.05-0.1**	0.3	0.05-0.1	0.3-0.6	0.05
Lenna	**0.5**	**0.1**	0.5-0.6	0.05	0.5	0.1	0.5-0.6 0.5	0.05 0.1
molecule	**0.3-0.6**	**0.1**	**0.3-0.6**	**0.05-0.1**	**0.3-0.6**	**0.1**	**0.3-0.6**	**0.05-0.1**
teapot	**0.3-0.6**	**0.05**	0.3-0.4 0.5	0.05-0.1 0.05	0.3-0.6 0.3	0.05 0.1	0.3-0.6	0.05
ant	**0.4-0.6**	**0.05-0.1**	0.4-0.6	0.05-0.1	0.4-0.6	0.1	0.4	0.1
blond	**0.6**	**0.05-0.1**	**0.6**	**0.05**	**0.6**	**0.05-0.1**	**0.6**	**0.05-0.1**
jet	**0.7**	**0.05-0.1**	**0.7**	**0.05-0.1**	**0.6-0.7**	**0.1**	**0.7**	**0.05-0.1**
mandrill	**0.5**	**0.05-0.1**	**0.5**	**0.05-0.1**	**0.5**	**0.05-0.1**	**0.5**	**0.05-0.1**
peppers	**0.6**	**0.05-0.1**	**0.5-0.6**	**0.1**	**0.5-0.6** **0.5**	**0.05** **0.1**	**0.5**	**0.1**
mouse	**0.3-0.6**	**0.1**	**0.3-0.6**	**0.1**	**0.3-0.6**	**0.1**	**0.3-0.6**	**0.1**
rose	**0.4-0.6**	**0.05**	**0.4-0.6**	**0.05**	**0.4-0.6**	**0.05**	**0.4-0.6**	**0.1**

Table 6.59: Values of $perc_1$ and $perc_2$ producing best results for all four combinations of cluster centre and distance type.

image	*mean*		*median*		*weighted*		*regression*	
	$perc_1$	$perc_2$	$perc_1$	$perc_2$	$perc_1$	$perc_2$	$perc_1$	$perc_2$
balls	**0.3**	**0.05-0.1**	0.4-0.6	0.05	**0.3**	**0.05-0.1**	0.4-0.6	0.05
Lenna	0.4	0.05-0.1	0.4	0.05-0.1	0.4	0.05-0.1	**0.4**	**0.05-0.1**
molecule	**0.3-0.6**	**0.1**	**0.3-0.6**	**0.1**	**0.3-0.6**	**0.1**	**0.3-0.6**	**0.1**
teapot	**0.3-0.6**	**0.05**	0.3-0.6	0.05	**0.3-0.6**	**0.05**	0.3-0.6 0.3	0.05 0.1
ant	**0.3-0.6**	**0.1**	**0.3-0.6**	**0.1**	**0.3-0.6**	**0.1**	**0.3-0.6**	**0.1**
blond	0.5	0.05-0.1	**0.5**	**0.05-0.1**	0.5-0.6	0.05	0.5	0.05-0.1
jet	0.5-0.6	0.05	0.5-0.6	0.05	0.5-0.6	0.05	**0.5**	**0.05**
mandrill	**0.4**	**0.05-0.1**	0.4	0.05	**0.4**	**0.05-0.1**	**0.4**	**0.05-0.1**
peppers	0.5-0.6	0.05-0.1	**0.5-0.6**	**0.05-0.1**	0.5-0.6	0.05-0.1	0.5-0.6	0.05-0.1
mouse	**0.3-0.6**	**0.1**	0.3-0.6	0.1	**0.3-0.6**	**0.1**	0.3-0.6	0.1
rose	**0.4-0.6**	**0.05**	**0.3-0.6** **0.3**	**0.05** **0.1**	**0.4-0.6**	**0.05**	**0.3**	**0.05-0.1**

Table 6.60: Values of $perc_1$ and $perc_2$ producing best results when using only context values.

When using both the normalized intra-cluster distance and the percentage decrease in intra-cluster distance, a value of 0.40 for the former and 0.05 for the latter seems to give reasonable results for all the natural images. If we take these values as default values, the result is a method which is totally automated. However, the option of being able to modify these values should be available should the need arise to either increase or decrease the number of clusters. The methodology as described allows us to achieve this by requesting these values as input parameters.

The *regression* context type is performing quite well for most of the images, especially when both the original and context values are taken into account to perform the segmentation. When using both the normalized intra-cluster distance and the per-

image	*mean*		*median*		*weighted*		*regression*	
	$perc_1$	$perc_2$	$perc_1$	$perc_2$	$perc_1$	$perc_2$	$perc_1$	$perc_2$
balls	**0.4**	**0.05-0.1**	**0.4**	**0.05-0.1**	**0.4**	**0.05-0.1**	**0.4-0.6** **0.4**	**0.05** **0.1**
Lenna	**0.4**	**0.05-0.1**	**0.4**	**0.05-0.1**	**0.4**	**0.05-0.1**	**0.4**	**0.05-0.1**
molecule	0.3	0.1	0.3-0.4	0.1	0.3	0.1	**0.3-0.6**	**0.1**
teapot	**0.3-0.6**	**0.05**	**0.3-0.6**	**0.05**	**0.3-0.6**	**0.05**	**0.3-0.6**	**0.05**
ant	**0.3**	**0.1**	**0.3**	**0.1**	**0.3**	**0.1**	**0.3**	**0.1**
blond	**0.5**	**0.05-0.1**	**0.5**	**0.05-0.1**	**0.5**	**0.05-0.1**	**0.5**	**0.05-0.1**
jet	**0.6**	**0.05**	0.6	0.05	**0.6**	**0.05**	**0.6**	**0.05**
mandrill	**0.5**	**0.05-0.1**	**0.5**	**0.05-0.1**	**0.5**	**0.05-0.1**	**0.5**	**0.05-0.1**
peppers	**0.5-0.6** **0.5**	**0.05** **0.1**	**0.5-0.6** **0.5**	**0.05** **0.1**	**0.5-0.6** **0.5**	**0.05** **0.1**	**0.5-0.6** **0.5**	**0.05** **0.1**
mouse	**0.3-0.6**	**0.1**	**0.3-0.6**	**0.1**	**0.3-0.6**	**0.1**	**0.3-0.6**	**0.1**
rose	**0.3**	**0.05-0.1**	**0.3**	**0.05-0.1**	**0.3**	**0.05-0.1**	**0.3**	**0.05-0.1**

Table 6.61: Values of $perc_1$ and $perc_2$ producing best results when using original and context values.

image	RGB		XYZ		KL		HSV	
	$perc_1$	$perc_2$	$perc_1$	$perc_2$	$perc_1$	$perc_2$	$perc_1$	$perc_2$
balls	0.4	0.05-0.1	0.4	0.05-0.1	**0.4**	**0.05-0.1**	0.5	0.05-0.1
Lenna	**0.5**	**0.1**	0.4	0.05-0.1	**0.5-0.6**	**0.05**	0.5-0.6	0.05
molecule	**0.3-0.6**	**0.1**	0.3-0.6	0.05	**0.3-0.6**	**0.1**	0.3-0.6	0.05
teapot	0.3-0.6	0.05	0.3-0.6	0.05	**0.3**	**0.05**	0.3-0.6	0.05
ant	0.4-0.6	0.05-0.1	**0.3-0.6**	**0.1**	0.4-0.6	0.1	0.4-0.6 0.4	0.05 0.1
blond	**0.6**	**0.05-0.1**	**0.6**	**0.05-0.1**	**0.6**	**0.05-0.1**	0.6	0.05-0.1
jet	**0.7**	**0.05-0.1**	0.5-0.6 0.5	0.05 0.1	**0.7**	**0.05-0.1**	0.4-0.5	0.05-0.1
mandrill	**0.5**	**0.05-0.1**	0.4	0.05-0.1	**0.5**	**0.05-0.1**	0.4	0.05-0.1
peppers	**0.6**	**0.05-0.1**	0.4	0.05-0.1	**0.5-0.6** **0.5**	**0.05** **0.1**	0.4	0.05-0.1
mouse	0.3-0.6	0.1	**0.3-0.6**	**0.1**	0.3-0.6	0.1	0.4	0.1
rose	**0.4-0.6**	**0.05**	0.3-0.6 0.3	0.05 0.1	0.4-0.6	0.05	0.3	0.05-0.1

Table 6.62: Values of $perc_1$ and $perc_2$ producing best results for different colour spaces.

centage decrease in intra-clusters distance, generally a value of 0.40 for the former and 0.05 for the latter works well when only using the context values, however, when using both the original and context values, a value of 0.40-0.50 is needed for the normalized intra-cluster distance. When using the context values, especially when on their own, the segmentation results look smoother and the regions more connected, which is a clear advantage over the results using only the original RGB values. However, when only the context values are used, the segmented results look slightly blurred, which could be a problem when fine details, such as the writing on the plane in the image *jet*, need to be extracted. Another problem that can occur, especially when the background and foreground colours are similar, is that parts of the objects get

grouped together with the background.

The results by different colour spaces indicate that the RGB colour space performs quite well and there is no real benefit in selecting a different colour space in which to perform the segmentation. The visual results indicated that the RGB and KL colour spaces perform very similarly, and the HSV colour space tends to result in the segmented images looking very noisy, while the XYZ colour space could only produce good results for *ant*, *blond* and *mouse*.

When using stopping rule 2, the splitting method is a very simple technique which can be easily extended to cope with any dimensionality of images, and for this reason, this method may also be used for multispectral images. These methods may also have wide ranging applications with respect to medical images by, for example, finding abnormalities, which should be grouped as a separate cluster.

6.4 Stopping Rule 3

A variation of the stopping rule presented in the previous section is to require that there be a sequence of small enough percentage decrease values. The modified rule would be as follows:

Find the smallest value of K satisfying the condition

$$\text{percDec}(K+i) < perc, \qquad i = 1, 2, \ldots, max \tag{6.24}$$

where max is the number of consecutive percentage decrease values required to be below $perc$.

This rule ensures that we find a K for which there is not much change in the segmented image for at least the next max segmented images.

6.4.1 Results

When using stopping rule 3 in the algorithm described in section 6.1, a threshold value, *perc*, is needed, which will ultimately determine the number of clusters in the final segmentation results. Therefore, the procedure is not totally automated, however, we can overcome this problem by experimentation with different values for the threshold, thus finding its optimal value. Additionally, we investigate the use of requiring either one, two or three consecutive percentage decrease values to be below the threshold, *perc*, although results indicate that in most cases there is no need to require more than one value to be below the threshold. However, the extra results are obtained to enable local variations in the percentage decrease values to be overlooked. The rule that we are trying to satisfy says that when the segmented image is not changing much, we can stop.

6.4.1.1 Synthetic Images

The synthetic images shown in Figure 1.1 were first used to test the performance of this method. Tables 6.63-6.66 show the number of clusters obtained for various values of *perc* when using the mean cluster centres and Euclidean distance, mean cluster centres and absolute distance, median cluster centres and Euclidean distance and median cluster centres and absolute distance approaches, respectively, where *perc* represents the threshold for the percentage decrease in intra-cluster distance.

Increasing the value of *perc* has the effect of decreasing the number of clusters obtained for a given image, and for a given value of *perc*, a large variation is seen in the number of clusters obtained for the different images. Also, a larger number of clusters is obtained when more consecutive values of the percentage decrease are required to be below the value given by *perc*. The values of *perc* giving the correct

image	no. consec.	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
two	1	19	12	9	6	6	3	3	2	2	2
	2	23	12	9	9	6	3	3	2	2	2
	3	23	18	12	12	9	6	3	2	2	2
four	1	27	14	8	5	5	5	4	4	4	4
	2	32	14	14	8	5	5	4	4	4	4
	3	32	21	17	8	8	5	4	4	4	4
six	1	21	7	7	7	7	6	6	6	6	6
	2	25	7	7	7	7	6	6	6	6	6
	3	32	7	7	7	7	6	6	6	6	6
eight	1	32	8	8	8	8	8	8	8	8	8
	2	32	8	8	8	8	8	8	8	8	8
	3	32	8	8	8	8	8	8	8	8	8
ten	1	20	10	10	10	10	10	10	10	2	2
	2	23	16	10	10	10	10	10	10	10	2
	3	30	19	10	10	10	10	10	10	10	10
fifteen	1	19	15	15	15	15	15	15	15	2	2
	2	24	15	15	15	15	15	15	15	15	15
	3	27	15	15	15	15	15	15	15	15	15

Table 6.63: Number of clusters for synthetic images using stopping rule 3 with mean cluster centres and Euclidean distance.

image	no. consec.	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
two	1	21	10	8	5	5	4	3	3	2	2
	2	27	13	12	7	7	4	3	3	2	2
	3	27	13	12	12	12	7	3	3	2	2
four	1	17	14	9	8	5	5	4	4	4	4
	2	28	14	12	8	8	5	4	4	4	4
	3	28	20	12	12	8	5	4	4	4	4
six	1	21	7	7	7	7	6	6	6	6	6
	2	21	12	7	7	7	6	6	6	6	6
	3	21	12	7	7	7	6	6	6	6	6
eight	1	32	8	8	8	8	8	8	8	8	8
	2	32	8	8	8	8	8	8	8	8	8
	3	32	8	8	8	8	8	8	8	8	8
ten	1	17	12	10	10	10	10	10	10	10	10
	2	24	16	10	10	10	10	10	10	10	10
	3	24	16	10	10	10	10	10	10	10	10
fifteen	1	19	15	15	15	15	6	6	4	4	4
	2	27	15	15	15	15	15	15	4	4	4
	3	27	15	15	15	15	15	15	4	4	4

Table 6.64: Number of clusters for synthetic images using stopping rule 3 with mean cluster centres and absolute distance.

segmentation for each of the synthetic images are shown in Table 6.67. Note that for the image *six*, when using median cluster centres and Euclidean distance, and for the image *fifteen*, when using median cluster centres with either distance, at least two consecutive percentage decrease values were required to be below *perc* to obtain the correct number of clusters.

image	no. consec.	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
two	1	21	12	7	5	4	3	3	2	2	2
	2	23	12	12	7	4	3	3	2	2	2
	3	23	12	12	12	12	3	3	2	2	2
four	1	25	14	8	5	5	4	4	4	4	4
	2	25	16	14	12	5	4	4	4	4	4
	3	25	19	14	12	5	4	4	4	4	4
six	1	21	3	3	3	3	3	3	3	3	3
	2	29	7	7	7	7	6	6	6	6	6
	3	29	7	7	7	7	6	6	6	6	6
eight	1	32	8	8	8	8	8	8	8	8	8
	2	32	8	8	8	8	8	8	8	8	8
	3	32	8	8	8	8	8	8	8	8	8
ten	1	19	10	10	10	10	10	2	2	2	2
	2	23	10	10	10	10	10	10	2	2	2
	3	27	19	10	10	10	10	10	10	10	10
fifteen	1	17	10	10	10	10	10	10	2	2	2
	2	25	15	15	15	15	15	15	2	2	2
	3	25	15	15	15	15	15	15	15	15	15

Table 6.65: Number of clusters for synthetic images using stopping rule 3 with median cluster centres and Euclidean distance.

image	no. consec.	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
two	1	21	9	5	5	5	4	3	3	2	2
	2	25	12	7	7	7	4	3	3	2	2
	3	25	12	7	7	7	7	3	3	2	2
four	1	25	13	8	8	5	5	4	4	4	4
	2	29	16	12	12	8	5	4	4	4	4
	3	29	20	12	12	8	5	4	4	4	4
six	1	22	7	7	7	7	6	6	6	6	6
	2	22	12	7	7	7	6	6	6	6	6
	3	37	15	7	7	7	6	6	6	6	6
eight	1	32	8	8	8	8	8	8	8	8	8
	2	32	8	8	8	8	8	8	8	8	8
	3	32	8	8	8	8	8	8	8	8	8
ten	1	19	11	10	10	10	10	10	10	10	10
	2	22	16	10	10	10	10	10	10	10	10
	3	32	19	10	10	10	10	10	10	10	10
fifteen	1	19	9	9	9	9	9	9	9	9	9
	2	26	15	15	15	15	15	15	15	15	15
	3	26	15	15	15	15	15	15	15	15	15

Table 6.66: Number of clusters for synthetic images using stopping rule 3 with median cluster centres and absolute distance.

6.4.1.2 Natural Images

In order to determine the optimal value of *perc* for the natural colour images shown in Figure 1.2, we can experiment with different values for this threshold, thus finding its optimal value. The results in this section are based on the use of the RGB colour space.

As stated earlier, the intra-cluster distance cannot be used, as Table 6.41 shows that this value is decreasing as the number of clusters increases, so we cannot simply

image	no. consec.	mean & Euc.	mean & abs.	med. & Euc.	med. & abs.
two	1	0.08-0.10	0.09-0.10	0.08-0.10	0.09-0.10
	2	0.08-0.10	0.09-0.10	0.08-0.10	0.09-0.10
	3	0.08-0.10	0.09-0.10	0.08-0.10	0.09-0.10
four	1	0.07-0.10	0.07-0.10	0.06-0.10	0.07-0.10
	2	0.07-0.10	0.07-0.10	0.06-0.10	0.07-0.10
	3	0.07-0.10	0.07-0.10	0.06-0.10	0.07-0.10
six	1	0.06-0.10	0.06-0.10		0.06-0.10
	2	0.06-0.10	0.06-0.10	0.06-0.10	0.06-0.10
	3	0.06-0.10	0.06-0.10	0.06-0.10	0.06-0.10
eight	1	0.02-0.10	0.02-0.10	0.02-0.10	0.02-0.10
	2	0.02-0.10	0.02-0.10	0.02-0.10	0.02-0.10
	3	0.02-0.10	0.02-0.10	0.02-0.10	0.02-0.10
ten	1	0.02-0.08	0.03-0.10	0.02-0.06	0.03-0.10
	2	0.03-0.09	0.03-0.10	0.02-0.07	0.03-0.10
	3	0.03-0.10	0.03-0.10	0.03-0.10	0.03-0.10
fifteen	1	0.02-0.08	0.02-0.05		
	2	0.02-0.10	0.02-0.07	0.02-0.07	0.02-0.10
	3	0.02-0.10	0.02-0.07	0.02-0.10	0.02-0.10

Table 6.67: Values of *perc* giving correct segmentation for synthetic images.

decide to stop at the certain number of clusters which produces the lowest intra-cluster distance. Therefore, we determine when the intra-cluster distance is not changing too much any more. To accomplish this, we incorporate the percentage decrease in intra-cluster distance, shown in Table 6.43. We can use these numbers to determine whether there was a significant change in the segmented image from increasing the number of clusters by one. Visual analysis indicates that when this number is below 0.05, there is no significant change in the segmented image, and anything above 0.05 shows quite a difference in the images. Bearing this in mind, the rule that we are trying to satisfy says that when the segmented image is not changing much, we can stop.

Tables 6.68-6.71 show the number of clusters obtained for various values of *perc* when using the mean cluster centres and Euclidean distance, mean cluster centres and absolute distance, median cluster centres and Euclidean distance and median cluster centres and absolute distance approaches, respectively, where *perc* represents the threshold for the percentage decrease in intra-cluster distance.

Increasing the value of *perc* has the effect of decreasing the number of clusters

image	no. consec.	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
balls	1	10	8	6	6	6	4	4	4	4	4
	2	15	12	12	12	6	6	6	6	6	6
	3	15	12	12	12	6	6	6	6	6	6
Lenna	1	10	9	9	9	7	6	5	5	5	5
	2	17	9	9	9	9	6	5	5	5	5
	3	17	15	15	12	12	6	5	5	5	5
molecule	1	22	21	17	14	14	12	8	8	8	8
	2	22	21	20	16	14	14	8	8	8	8
	3	22	21	20	20	14	14	14	14	8	8
teapot	1	17	13	13	9	9	9	8	8	5	5
	2	26	19	19	19	13	9	8	8	5	5
	3	26	26	25	19	13	9	8	8	8	8
ant	1	14	7	7	7	5	5	5	5	5	5
	2	14	13	13	12	12	12	12	11	11	7
	3	29	13	13	12	12	12	12	11	11	10
blond	1	9	4	4	4	3	3	3	3	3	3
	2	9	9	7	7	3	3	3	3	3	3
	3	19	18	7	7	7	7	7	6	6	6
jet	1	7	5	5	5	5	3	3	3	3	3
	2	7	5	5	5	5	5	5	5	5	5
	3	24	5	5	5	5	5	5	5	5	5
mandrill	1	17	13	12	10	6	6	6	5	5	5
	2	27	13	12	12	8	8	8	5	5	5
	3	27	16	12	12	8	8	8	5	5	5
peppers	1	13	8	8	6	6	6	6	6	6	6
	2	28	15	12	12	10	10	10	6	6	6
	3	28	15	15	15	10	10	10	6	6	6
mouse	1	16	14	10	10	9	8	8	6	6	6
	2	19	14	10	10	9	8	8	6	6	6
	3	19	14	14	13	9	8	8	6	6	6
rose	1	17	9	9	9	8	7	6	6	6	5
	2	36	21	17	17	8	7	6	6	6	5
	3	36	21	21	21	12	7	6	6	6	5

Table 6.68: Number of clusters using stopping rule 3 with mean cluster centres and Euclidean distance.

obtained for a given image, and for a given value of *perc*, a large variation is seen in the number of clusters obtained for different images. Also, a larger number of clusters is obtained when more consecutive values of the percentage decrease are required to be below the value given by *perc*. To determine the optimal value for *perc*, visual inspection of the segmented images is required. Following is a detailed analysis of the results for each image. The results are summarized in section 6.4.2.

The image *balls* achieved the best segmentation when using either mean cluster centres and absolute distance or median cluster centres and Euclidean distance. These two results also gave a similar number of clusters, 11 and 10, respectively. These results were achieved regardless of how many consecutive percentage decrease values were required to be below *perc*. The mean cluster centre and Euclidean distance

image	no. consec	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
balls	1	14	11	6	6	6	6	6	4	4	4
	2	16	11	11	11	8	8	8	8	6	6
	3	16	14	14	11	11	11	11	8	6	6
Lenna	1	13	8	8	8	7	7	7	5	5	5
	2	19	15	8	8	7	7	7	7	5	5
	3	36	15	8	8	7	7	7	7	5	5
molecule	1	19	18	18	13	8	8	8	8	8	8
	2	21	18	18	13	13	13	10	10	10	10
	3	21	21	21	17	16	16	16	16	16	10
teapot	1	20	6	6	5	5	5	4	4	4	3
	2	26	18	18	5	5	5	4	4	4	3
	3	26	18	18	18	16	5	4	4	4	3
ant	1	15	14	8	8	5	5	5	5	5	5
	2	24	14	8	8	8	7	7	7	7	7
	3	24	14	14	13	13	7	7	7	7	7
blond	1	7	7	7	7	3	3	3	3	3	3
	2	7	7	7	7	7	5	5	5	5	5
	3	7	7	7	7	7	5	5	5	5	5
jet	1	8	4	4	4	3	3	3	3	3	3
	2	8	8	8	8	3	3	3	3	3	3
	3	26	23	13	13	13	13	13	13	13	13
mandrill	1	13	11	11	9	6	6	6	4	4	4
	2	15	15	15	13	8	8	8	8	6	6
	3	15	15	15	13	11	8	8	8	6	6
peppers	1	12	9	9	9	9	7	6	6	6	6
	2	20	17	11	9	9	7	6	6	6	6
	3	65	17	17	9	9	7	6	6	6	6
mouse	1	15	12	10	10	10	9	8	8	6	6
	2	23	15	15	12	12	9	8	8	6	6
	3	23	15	15	15	15	9	8	8	6	6
rose	1	25	11	11	11	10	6	6	6	6	6
	2	28	23	22	18	10	10	8	8	8	8
	3	38	23	22	18	18	16	8	8	8	8

Table 6.69: Number of clusters using stopping rule 3 with mean cluster centres and absolute distance.

approach also gave a good result when only taking one percentage decrease value with *perc* set to 0.01. The main difference with this result was that the pink ball was partly classified as belonging to the shadow segment.

The overall best segmentation for the image *Lenna* was produced when using mean cluster centres and Euclidean distance which resulted in 6 clusters. The best results for the other combinations of cluster centre and distance type gave a higher number of clusters, however, the result produced from the mean cluster centres and Euclidean distance approach was visually better. For the best result, the same value of 0.06 for *perc* produced 6 clusters regardless of how many consecutive values of the percentage decrease were required to be below *perc*.

The image *molecule* produced good results for all four combinations of cluster

image	no. consec.	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
balls	1	10	7	7	7	7	7	6	5	5	4
	2	14	10	10	10	10	7	6	5	5	4
	3	14	10	10	10	10	7	6	5	5	4
Lenna	1	11	9	5	5	5	5	5	5	5	5
	2	14	9	9	9	9	5	5	5	5	5
	3	14	9	9	9	9	9	5	5	5	5
molecule	1	21	20	19	13	13	12	12	8	8	8
	2	21	20	19	13	13	12	12	12	8	8
	3	21	20	19	19	19	12	12	12	8	8
teapot	1	17	14	14	9	8	8	8	5	5	5
	2	25	20	16	16	8	8	8	7	7	5
	3	25	20	20	20	13	13	12	7	7	5
ant	1	9	9	9	9	9	6	6	6	6	6
	2	9	9	9	9	9	6	6	6	6	6
	3	9	9	9	9	9	9	9	9	9	9
blond	1	9	8	4	4	3	3	3	3	3	3
	2	12	8	8	8	3	3	3	3	3	3
	3	12	8	8	8	8	8	8	8	8	6
jet	1	5	5	5	5	5	4	4	4	4	3
	2	5	5	5	5	5	4	4	4	4	3
	3	23	11	10	10	10	4	4	4	4	3
mandrill	1	20	16	10	8	5	5	5	5	5	4
	2	20	16	10	10	10	10	5	5	5	4
	3	28	16	13	13	10	10	8	8	5	4
peppers	1	18	9	9	8	7	7	7	7	7	6
	2	18	9	9	8	7	7	7	7	7	6
	3	44	9	9	8	7	7	7	7	7	6
mouse	1	13	13	10	9	3	3	3	3	3	3
	2	16	16	12	9	9	9	9	6	6	6
	3	24	19	15	15	9	9	9	6	6	6
rose	1	18	10	10	7	7	7	7	6	6	5
	2	37	17	17	7	7	7	7	6	6	5
	3	37	30	25	15	15	15	7	6	6	5

Table 6.70: Number of clusters using stopping rule 3 with median cluster centres and Euclidean distance.

centre and distance type when 8 clusters were obtained, and all of these four results were virtually identical. When using absolute distance with either cluster centre type, and only looking for the first value of the percentage decrease which is below *perc*, the best segmentation was obtained for a larger range of the values for *perc* beginning with 0.05. The best results were obtained regardless of how many consecutive percentage decrease values were considered, except in the case of using mean cluster centres and absolute distance where only taking one percentage decrease value produced 8 clusters. A value greater than 0.10 would be required for *perc* in order for this result to be selected when two or three consecutive percentage decrease values are required to be below *perc*.

For the image *teapot*, the best overall result occurred when using mean cluster

image	no. consec.	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
balls	1	13	10	10	10	10	6	5	5	5	5
	2	17	12	10	10	10	6	5	5	5	5
	3	26	12	10	10	10	6	5	5	5	5
Lenna	1	11	7	7	7	6	6	6	5	5	5
	2	13	13	13	13	6	6	6	5	5	5
	3	13	13	13	13	6	6	6	5	5	5
molecule	1	23	19	15	13	8	8	8	8	8	8
	2	29	22	15	15	13	13	10	10	10	8
	3	32	22	18	18	13	13	13	13	13	8
teapot	1	18	16	11	9	9	8	7	7	6	6
	2	23	20	20	18	16	8	7	7	6	6
	3	28	20	20	18	16	16	7	7	6	6
ant	1	10	10	10	10	10	5	5	5	4	4
	2	10	10	10	10	10	10	10	10	4	4
	3	10	10	10	10	10	10	10	10	8	8
blond	1	8	7	5	5	5	3	3	3	3	3
	2	8	7	7	7	7	7	5	5	5	5
	3	8	7	7	7	7	7	5	5	5	5
jet	1	6	5	5	5	4	4	4	4	4	4
	2	16	5	5	5	4	4	4	4	4	4
	3	16	14	13	13	4	4	4	4	4	4
mandrill	1	14	11	11	9	6	6	6	6	5	5
	2	26	11	11	11	6	6	6	6	5	5
	3	32	26	11	11	11	9	9	9	5	5
peppers	1	14	12	9	8	8	5	5	5	5	5
	2	14	12	12	8	8	8	8	7	7	7
	3	49	12	12	11	8	8	8	7	7	7
mouse	1	22	11	11	9	9	9	9	3	3	3
	2	29	20	19	16	11	11	9	9	7	6
	3	43	20	19	16	11	11	9	9	7	6
rose	1	23	18	10	10	8	7	7	7	7	7
	2	27	20	20	18	10	7	7	7	7	7
	3	27	27	27	18	10	10	10	10	7	7

Table 6.71: Number of clusters using stopping rule 3 with median cluster centres and absolute distance.

centres and Euclidean distance, with the values ranging from 0.07-0.10 for *perc*, depending on the number of consecutive values required. Also, the segmentation results when using median cluster centres with either distance type were not bad, and were both very similar. These results were obtained regardless of how many consecutive percentage decrease values were required to be below *perc*, except in the case of using median cluster centres and Euclidean distance where only taking two or three consecutive values of the percentage decrease produced the best result. When using mean cluster centres and absolute distance, there was a big drop in the number of clusters for a certain value of *perc* which differs depending on how many consecutive values were required. This resulted in one segmentation where there were too many segments and another which did not contain enough regions to adequately represent

the image.

The image *ant* produced good results for each of the four approaches, for a broad range of values for *perc*. Each of these results contained the same number of clusters. The mean cluster centres and either distance and median cluster centres and absolute distance approaches produced the overall best results when only 5 clusters were produced, which were virtually identical from the visual point of view. The best results mostly occurred only when just one value of the percentage decrease was required to be below *perc*.

For the image *blond*, the best results occurred when using mean cluster centres with either distance type. However, a reasonable result also occurred when using median cluster centres and Euclidean distance. Each of these results were very similar, so no preference can be given to either of these three approaches. For this image, there tended to be a slightly bigger than normal decrease in the number of clusters obtained when using only one consecutive percentage decrease value, so for two particular consecutive values of *perc*, the segmented results changed from a result having slightly too many clusters to a result not containing enough clusters. A better number of clusters was obtained when three consecutive values of the percentage decrease were required to be below *perc*, and in the case of using mean cluster centres and absolute distance, taking only two consecutive percentage decrease values also led to a good result.

For the image *jet*, the best segmentation results occurred when the number of clusters was 5. This occurred when using all combinations of cluster centres and distance type, except when using mean cluster centres and absolute distance. All of the results producing 5 clusters were very similar, so no preference can be given to any of the three combinations of cluster centres and distance type which produced

them. For each of these results, the best segmentation was produced when requiring only one or two percentage decrease values to be below *perc*, and in the case when mean cluster centres and Euclidean distance was used, the best segmentation also occurred when three consecutive values for the percentage decrease were required to be below *perc*.

For the image *mandrill*, the best segmentation results occurred when 8 or 9 clusters were produced. When using mean cluster centres with either distance type, this only occurred when two or three consecutive values of the percentage decrease were required to be below *perc*. When using median cluster centres with either distance type, the best segmentation was obtained when either one or three consecutive values of the percentage decrease were required to be below *perc*. All of the results from the four approaches were very similar, so no preference can be given to any of the four approaches.

For the image *peppers*, the best results were achieved when 6 or 7 clusters were formed. These results were obtained with all of the different number of consecutive percentage decrease values, except when using median cluster centres and absolute distance where the result with 7 clusters was only produced when either 2 or 3 consecutive percentage decrease values were required to be below *perc*. Also of interest is the fact that when mean cluster centres and Euclidean distance was used, the correct number of clusters was obtained for a larger range of values for *perc*, ranging from 0.04-0.10, when only one consecutive percentage decrease value was required to be below *perc*. With all of the other results the same range of values for *perc* gave the correct segmentation no matter how many consecutive values were considered. All of the results from the four approaches were very similar, so no preference can be given to any of the four approaches.

The best results for the image *mouse* occurred when the number of clusters was 6. When using mean cluster centres with either distance type, this result was produced for the same range of values for *perc*, regardless of how many consecutive values of the percentage decrease were required to be below *perc*. When using median cluster centres with either distance type, the best segmentation could only be found when requiring two or three consecutive percentage decrease values to be below *perc*. All of the results with 6 clusters were very similar, so no preference can be given to any of the four combinations of cluster centre and distance type.

For the image *rose*, the best results occurred when the number of clusters was 6 or 7. These results were produced by taking one, two or three consecutive values to be below *perc*, except when using mean cluster centres and absolute distance, which only produced 6 clusters when just one value of the percentage decrease was required to be below *perc*. All of the results were very similar, so no preference can be given to any of the four approaches.

Figures 6.35-6.38 show the best segmented results for each of the images when using mean cluster centres and Euclidean distance, mean cluster centres and absolute distance, median cluster centres and Euclidean distance and median cluster centres and absolute distance, respectively. In Figures 6.35-6.38 the *perc* values producing these results are indicated together with the number of consecutive percentage decrease values, which are shown in brackets. Generally, a good segmentation was found when requiring only one percentage decrease value to be below *perc*. The main exception to this was for the image *blond* which required there to be three consecutive percentage decrease values below *perc* when Euclidean distance was used with either cluster centre type, and either two or three consecutive values to be below *perc* when mean cluster centres and absolute distance were used. Also of interest is the fact

that for *ant*, the best results were found when only one percentage decrease value was considered. When using only one percentage decrease value, the values in the range 0.05-0.07 gave the best segmentation results for most images. The main exception was for *balls* which required a much smaller value of 0.01 or 0.02 and *mandrill* required 0.04. The other exception was for the image *mouse* which required at least a value of 0.08 for *perc* when using mean cluster centres. Since the best results could be found when considering only one percentage decrease value, there is really no need to require a larger number of consecutive values to occur. All of the best results for each of the images were found when using mean cluster centres and Euclidean distance, except for *balls* which obtained its best result when using mean cluster centres and absolute distance or median cluster centres and Euclidean distance. In some cases, the results by each of the four combinations of cluster centre and distance type were all very similar, especially for the images *molecule*, *mandrill*, *peppers*, *mouse* and *rose*.

6.4.1.3 Results Using Context

Each of the results incorporating the context of the image are based on using only mean cluster centres and Euclidean distance and the use of the RGB colour space. Also, comparisons made to the results by the original RGB values refer to the results using mean cluster centres and Euclidean distance presented in section 6.4.1.2.

Tables 6.72-6.75 show the number of clusters obtained when using the *mean*, *median*, *weighted* and *regression* context values only, respectively, for various values of *perc*. Following is a detailed analysis of the results for each image. The results are summarized in section 6.4.2.

The image *balls* achieved the best segmentations when using either the *mean* or *weighted* context types when 11 clusters were produced. This was achieved when

image	no. consec.	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
balls	1	18	11	9	9	9	9	8	6	4	4
	2	18	18	11	11	11	11	8	8	8	4
	3	21	21	17	17	17	17	8	8	8	4
Lenna	1	30	13	12	11	8	8	6	6	5	5
	2	30	20	12	11	10	10	6	6	5	5
	3	52	23	23	11	10	10	6	6	5	5
molecule	1	18	18	15	15	12	9	9	9	8	8
	2	22	18	17	17	14	9	9	9	8	8
	3	22	18	17	17	17	9	9	9	8	8
teapot	1	19	14	14	9	9	8	8	5	5	5
	2	26	14	14	11	11	8	8	8	8	5
	3	26	21	19	11	11	11	8	8	8	8
ant	1	16	16	12	12	12	10	6	6	6	6
	2	29	29	12	12	12	12	10	10	8	8
	3	45	29	12	12	12	12	10	10	8	8
blond	1	20	11	11	9	4	4	4	4	4	4
	2	29	19	17	9	9	7	7	7	4	4
	3	46	19	17	9	9	7	7	7	4	4
jet	1	21	14	9	9	9	9	9	9	7	4
	2	32	20	14	14	9	9	9	9	7	7
	3	32	24	14	14	13	9	9	9	7	7
mandrill	1	20	18	13	12	8	7	7	5	5	5
	2	34	22	13	12	8	7	7	5	5	5
	3	34	22	16	12	8	7	7	5	5	5
peppers	1	18	13	13	7	7	7	7	7	7	7
	2	28	17	15	11	9	7	7	7	7	7
	3	32	20	15	11	9	7	7	7	7	7
mouse	1	17	16	13	12	10	10	8	7	7	6
	2	33	16	13	12	10	10	10	7	7	6
	3	36	16	13	12	10	10	10	7	7	6
rose	1	23	15	11	9	9	8	7	7	7	5
	2	41	28	14	14	14	8	7	7	7	7
	3	41	28	23	23	20	13	7	7	7	7

Table 6.72: Number of clusters using stopping rule 3 with *mean* context values only.

either one or two consecutive values of the percentage decrease were considered. When only using one percentage decrease value, only the value of 0.02 for *perc* gave a good segmentation for the *mean* and *weighted* contexts, however, for the *median* and *regression* context types, 0.03-0.04 for the *median* context and 0.02-0.05 for the *regression* context gave a reasonable level of segmentation and for both of these context types, this result was also obtained when using three consecutive values of the percentage decrease. Despite this, the *mean* and *weighted* context results were better as the pink ball appeared more smooth and connected in these results. In comparison to the results using the original RGB values, the context results showed more smoother and connected clusters which is a clear advantage as each ball should be classified as one segment.

image	no. consec.	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
balls	1	16	12	10	10	9	9	6	6	4	4
	2	16	12	12	10	9	9	8	8	4	4
	3	19	19	12	10	9	9	8	8	4	4
Lenna	1	14	13	12	9	6	6	6	6	5	5
	2	19	13	12	9	9	8	8	6	5	5
	3	29	25	12	12	12	8	8	6	5	5
molecule	1	17	14	13	13	11	11	8	8	8	8
	2	17	16	13	13	13	13	13	10	8	8
	3	17	16	16	16	13	13	13	13	8	8
teapot	1	16	12	12	12	8	8	8	5	5	5
	2	25	12	12	12	11	10	8	8	5	5
	3	25	18	18	18	11	10	8	8	8	8
ant	1	17	10	10	10	10	10	6	6	6	6
	2	17	14	14	12	12	12	12	12	9	9
	3	28	14	14	12	12	12	12	12	12	9
blond	1	13	13	8	8	4	4	4	4	4	4
	2	13	13	8	8	8	8	8	8	4	4
	3	36	26	8	8	8	8	8	8	4	4
jet	1	16	5	5	5	5	5	5	5	3	3
	2	16	16	11	10	10	10	10	8	8	8
	3	24	19	11	10	10	10	10	8	8	8
mandrill	1	17	14	9	9	9	7	6	6	5	5
	2	23	14	14	12	12	7	6	6	5	5
	3	34	17	14	12	12	7	6	6	5	5
peppers	1	13	11	11	10	7	7	7	7	7	7
	2	15	11	11	10	7	7	7	7	7	7
	3	33	11	11	10	10	7	7	7	7	7
mouse	1	19	13	10	9	9	9	8	7	7	6
	2	19	16	15	9	9	9	8	7	7	6
	3	22	16	15	9	9	9	8	7	7	6
rose	1	16	14	10	10	10	8	7	7	7	5
	2	28	24	23	19	16	12	7	7	7	7
	3	38	27	23	19	19	12	10	7	7	7

Table 6.73: Number of clusters using stopping rule 3 with *median* context values only.

The overall best segmentation for the image *Lenna* was produced when using the *mean* or *weighted* contexts, which both produced ten clusters only when two or three consecutive values of the percentage decrease were considered and when *perc* was set to 0.05-0.06. The *regression* context result was also quite good when ten clusters were produced, however, the segments were not quite as smooth in appearance as the *mean* and *weighted* context results. For the *median* context, a higher number of clusters was required to produce a good result. In comparison to the results using the original RGB values, the context results were more smoother, and looked a little blurry.

The image *molecule* produced good results for all four context types when *perc* was set to 0.09-0.10 regardless of how many consecutive values of the percentage decrease

image	no. consec.	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
balls	1	18	11	9	9	9	9	6	6	4	4
	2	18	18	11	11	11	11	8	8	8	4
	3	21	21	17	17	17	17	8	8	8	4
Lenna	1	32	13	12	11	8	8	6	6	5	5
	2	34	19	12	11	10	10	6	6	5	5
	3	34	23	22	11	10	10	6	6	5	5
molecule	1	20	19	15	15	10	9	9	9	8	8
	2	24	19	19	17	17	9	9	9	8	8
	3	24	19	19	17	17	9	9	9	8	8
teapot	1	20	14	14	9	9	8	5	5	5	5
	2	26	14	14	11	11	8	8	8	8	5
	3	26	20	18	11	11	11	8	8	8	8
ant	1	14	13	12	12	12	10	6	6	6	6
	2	25	13	12	12	12	12	10	10	8	8
	3	42	24	12	12	12	12	10	10	8	8
blond	1	30	19	11	9	4	4	4	4	4	4
	2	34	19	17	9	9	8	7	6	4	4
	3	34	19	17	9	9	8	7	6	4	4
jet	1	21	14	9	9	9	9	9	9	7	4
	2	29	20	14	14	13	13	11	9	7	7
	3	38	25	17	14	13	13	11	9	7	7
mandrill	1	20	19	13	12	8	7	7	5	5	5
	2	24	19	13	12	8	7	7	5	5	5
	3	33	22	16	12	8	7	7	5	5	5
peppers	1	16	13	13	7	7	7	7	7	7	7
	2	29	15	15	11	9	7	7	7	7	7
	3	29	21	21	11	9	7	7	7	7	7
mouse	1	26	15	13	12	10	10	8	7	7	6
	2	30	19	13	12	10	10	10	7	7	6
	3	35	19	13	12	10	10	10	7	7	6
rose	1	24	15	11	9	9	8	7	7	7	5
	2	33	23	14	14	14	8	7	7	7	7
	3	48	23	23	19	19	13	7	7	7	7

Table 6.74: Number of clusters using stopping rule 3 with *weighted* context values only.

were considered. Additionally, when using the *median* context and only one value of the percentage decrease, the values 0.07 and 0.08 for *perc* also produced a good result. Each of these results contained the same number of clusters, 8, and looked very similar, except that when using the *mean*, *weighted* and *regression* contexts, the edges of the molecule were a little fuzzy, as opposed to the best results by the *median* context and the original RGB values, which had much sharper edges in the segmented images. Apart from this, all of the results, including the best result when using only the original RGB values, looked very similar.

For the image *teapot*, the best results for the *mean* and *weighted* contexts occurred when using only one value of the percentage decrease when *perc* was set to 0.04 or 0.05. When using the *median* or *regression* contexts the best results occurred only

image	no. consec.	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
balls	1	22	11	11	11	11	10	4	4	4	4
	2	22	11	11	11	11	10	4	4	4	4
	3	31	19	11	11	11	10	10	10	10	4
Lenna	1	22	17	12	8	8	7	6	5	5	5
	2	32	17	12	11	10	7	6	5	5	5
	3	38	27	16	11	10	10	6	5	5	5
molecule	1	21	17	12	12	11	9	9	9	8	8
	2	21	21	15	15	11	9	9	9	8	8
	3	21	21	15	15	15	9	9	9	8	8
teapot	1	18	14	14	12	8	8	5	5	5	5
	2	22	14	14	12	10	10	8	8	5	5
	3	42	20	20	12	10	10	8	8	8	8
ant	1	16	13	12	11	11	6	6	6	6	6
	2	32	28	12	11	11	11	11	10	8	8
	3	43	28	18	11	11	11	11	10	8	8
blond	1	9	9	9	3	3	3	3	3	3	3
	2	40	20	15	15	11	8	8	6	5	5
	3	40	26	15	15	11	8	8	6	5	5
jet	1	19	14	6	6	6	6	6	3	3	3
	2	19	16	16	14	14	10	10	9	9	6
	3	30	25	22	14	14	10	10	9	9	6
mandrill	1	19	16	13	12	8	7	7	5	5	5
	2	30	18	15	12	8	7	7	5	5	5
	3	40	18	15	12	8	7	7	5	5	5
peppers	1	13	11	11	7	7	7	7	7	7	7
	2	25	11	11	11	11	7	7	7	7	7
	3	41	11	11	11	11	7	7	7	7	7
mouse	1	25	19	13	12	10	10	8	7	7	6
	2	29	23	18	12	10	10	10	7	7	6
	3	35	23	21	16	10	10	10	7	7	6
rose	1	22	15	12	11	6	6	6	6	6	6
	2	43	29	14	11	11	8	8	8	8	8
	3	49	29	25	18	17	17	17	8	8	8

Table 6.75: Number of clusters using stopping rule 3 with *regression* context values only.

when two or three consecutive values of the percentage decrease were considered when *perc* was 0.05 and also when *perc* was 0.06 for the *regression* context. The best overall results occurred when using the *mean* and *weighted* context types as they contained the smallest number of clusters and still represented the regions in the image adequately, however, the shadow of the teapot was grouped as one colour rather than showing the individual colours of the squares in the checkerboard. The *median* context result did not suffer as much from this phenomenon, however, the teapot itself contained more shades, as did the result by the *regression* context. In comparison to the best result using the original RGB values, the shadow region was handled better when using only the original RGB values, however, the reflection part contained some blue, which did not occur in the context results.

The image *ant* produced good results for each of the four context types when two or three consecutive values of the percentage decrease were considered when *perc* was set to 0.09-0.10. This was true in all cases except when the *median* context was used with three percentage decrease values, when only the value 0.10 for *perc* was good. The best overall results occurred when 8 clusters were obtained, which occurred when the *mean*, *weighted* and *regression* contexts were used, however, the *mean* and *weighted* context results were more smoother in appearance than the *regression* context result. The results obtained with only 6 clusters were also not too bad, except that part of the bottom of the mouse was black. These results occurred when using only one percentage decrease value when *perc* was 0.07-0.10, and also 0.06 when the *regression* context was used. In comparison to the results using the original RGB values, the context results had more smooth looking segments.

For the image *blond*, the best results occurred when the number of clusters was 8 or 9. These results were obtained regardless of how many percentage decrease values were considered for all context types except the *regression* context where only one percentage decrease value was needed when *perc* was 0.01-0.03. The values of *perc* which were good for the other context types were generally between 0.04 and 0.05, with a larger range obtaining good results for the *median* context, depending on how many consecutive values of the percentage decrease were considered. Each of the four results were very similar, except that when the *median* context was used, more of the lips were grouped together. These context results were much better than the results using only the original RGB values, as these results correctly identified the hair as a separate cluster in yellow, which did not occur when using only the original RGB values for an equivalent number of clusters.

For the image *jet*, the best segmentation results occurred when the number of

clusters was 9 or 10. In the case of the *mean* and *weighted* contexts, 9 clusters were obtained regardless of how many consecutive percentage decrease values were considered. In the case of the *median* and *regression* contexts, 10 clusters were obtained only when either two or three consecutive percentage decrease values were considered. The best overall result occurred when using the *regression* context, as the F-16 symbol was clearer. In this case the values 0.06 and 0.07 for *perc* were optimal. In comparison to the results using the original RGB values, the context results, especially for the *mean*, *median* and *weighted* contexts, were blurry, and generally more smoother looking, while the writing on the plane was much clearer when using only the original RGB values.

For the image *mandrill*, the optimal value for *perc* was 0.04 for each of the four context types, with the *median* context also achieving a good result for the values 0.05. In the case of the *median* context, the best result was only obtained when two or three consecutive values of the percentage decrease were considered, while for each of the other context types, the use of any number of percentage decrease values was good. The results were all very similar, except when using the *median* context, for which the right eye was partly classified together with the green fur segment. In comparison to the results using the original RGB values, the context results appeared a little blurred.

For the image *peppers*, the best results were achieved when 7 clusters were formed. This result was obtained with each of the four context types regardless of how many consecutive percentage decrease values were used. Mostly, the values 0.06-0.10 were good values for *perc*, and also in the case of using the *mean*, *weighted* and *regression* contexts with only one percentage decrease values, 0.04 and 0.05 were also good, and when using the *median* context type, with one or two percentage decrease values,

0.05 was also good. All of the results were very similar, however, the *median* context result was slightly better as the other results all missed a little part of the red pepper on the right, and grouped it with the background. In comparison to the results by the original RGB values, the context results had more smoother looking segments.

The best results for the image *mouse* occurred when the number of clusters was 7, which occurred when *perc* was set to 0.08 or 0.09. This occurred for each of the four context types, regardless of how many consecutive percentage decrease values were used. Each of these four results were very similar in appearance and in comparison to the results using only the original RGB values, the context results had more smoother segments.

For the image *rose*, the best results occurred when the number of clusters ranged between 9 and 11. In most cases this only occurred when just one percentage decrease value was considered. For the *mean* and *weighted* contexts, this occurred when *perc* was 0.04 or 0.05, for the *median* context the best result occurred when *perc* was 0.03-0.05 and for the *regression* context the best result occurred when *perc* was 0.04. Also, when three consecutive values of the percentage decrease values were used for the *median* context, a value of 0.07 for *perc* gave a good result, and when two consecutive values of the percentage decrease were used for the *regression* context, the values 0.04 and 0.05 for *perc* also produced a good result. Each of these results by the four context types were very similar from a visual point of view. In comparison to the results using the original RGB values, the context results showed more smoother clusters.

Figures 6.39-6.42 show the segmentation results when using only the *mean*, *median*, *weighted* or *regression* contexts for various values of *perc* which produced the best visual segmentation results, respectively. In Figures 6.39-6.42 the *perc* values

producing these results are indicated together with the number of consecutive percentage decrease values, which are shown in brackets. Generally, the context results tended to be slightly blurred and also contained more smooth looking segments. This is not surprising since when we apply the context operation to the original image, the resulting image is a slightly blurred version of the original since we are using the eight neighbours in calculating new colour values. Overall, the *mean* and *weighted* contexts performed well with good results occurring for most images. The *median* and *regression* contexts only produced good results for half of the images. It is very difficult to prefer one context type over the others as there was a lot a variation in the results. Generally, setting *perc* to be between 0.04 and 0.06 worked quite well for most images. The main exceptions to this occurred for the images *molecule*, *ant* and *mouse*, which required *perc* to be at least about 0.08 or 0.09.

Tables 6.76-6.79 show the number of clusters obtained when using the *mean*, *median*, *weighted* and *regression* context values together with the original RGB values, respectively, for various values for *perc*. Following is a detailed analysis of the results for each image. The results are summarized in section 6.4.2.

The image *balls* achieved the best segmentation results when the number of clusters was between 10 and 12. For the *mean* context, this occurred when one or two consecutive percentage decrease values were used and *perc* was 0.02-0.04. For the *median* context, this occurred when only one percentage decrease value was considered and when *perc* was 0.02-0.03. For the *weighted* context, the best result occurred only when three consecutive values of the percentage decrease were used and when *perc* was set to 0.05-0.06, and for the *regression* context, the best result occurred regardless of how many percentage decrease values were considered and when *perc* was 0.02-0.04, 0.03-0.04 and 0.04 for one, two or three percentage decrease values,

image	no. consec.	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
balls	1	18	10	10	10	9	7	5	5	4	4
	2	18	10	10	10	9	9	5	5	4	4
	3	18	17	13	13	9	9	5	5	4	4
Lenna	1	13	13	11	8	7	5	5	5	5	5
	2	13	13	13	10	7	7	5	5	5	5
	3	26	13	13	13	10	7	5	5	5	5
molecule	1	18	14	12	11	8	8	8	8	8	5
	2	18	18	18	11	11	10	10	10	10	10
	3	18	18	18	11	11	10	10	10	10	10
teapot	1	17	12	12	11	9	8	8	5	5	5
	2	17	15	15	11	9	8	8	5	5	5
	3	23	15	15	14	9	8	8	8	8	8
ant	1	16	12	12	11	11	6	6	6	6	6
	2	16	14	12	11	11	11	10	6	6	6
	3	16	14	12	11	11	11	10	9	9	6
blond	1	16	14	9	3	3	3	3	3	3	3
	2	20	16	9	9	7	7	6	5	5	5
	3	31	16	9	9	7	7	6	5	5	5
jet	1	18	8	8	8	4	4	4	4	4	4
	2	20	8	8	8	8	8	7	6	6	6
	3	26	16	12	11	11	8	7	6	6	6
mandrill	1	19	16	8	7	6	5	5	5	5	4
	2	21	16	10	7	6	5	5	5	5	4
	3	29	16	10	7	6	5	5	5	5	4
peppers	1	19	9	9	7	7	7	7	7	7	5
	2	19	9	9	7	7	7	7	7	7	5
	3	19	18	13	7	7	7	7	7	7	5
mouse	1	25	13	13	10	10	8	8	7	6	6
	2	25	13	13	10	10	10	8	7	6	6
	3	25	20	18	10	10	10	8	7	6	6
rose	1	18	14	9	9	6	6	6	6	6	5
	2	26	26	23	18	13	8	8	8	8	5
	3	31	31	23	21	16	16	8	8	8	8

Table 6.76: Number of clusters using stopping rule 3 with original and *mean* context values.

respectively. Each of these results were very similar from a visual point of view. In comparison to the results using only the original RGB values, the results using both the original and context values showed more smoother and connected clusters, which was mostly noticeable in the region containing the pink ball, and this is a clear advantage as each ball should be classified as one segment.

The best segmentation results for the image *Lenna* were produced when 10 clusters were formed. For each of the four context types, this occurred when two or three consecutive percentage decrease values were used. The optimal values for *perc* were 0.04 or 0.05, depending on how many percentage decrease values were considered. Each of the four results were very similar in appearance, and also looked similar to the best result using only the original RGB values, except that the results which

image	no. consec.	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
balls	1	13	10	10	9	4	4	4	4	4	4
	2	19	15	12	9	9	8	8	8	6	4
	3	19	15	12	12	12	8	8	8	6	4
Lenna	1	15	11	8	8	8	5	5	5	5	5
	2	21	13	13	10	10	5	5	5	5	5
	3	25	13	13	10	10	5	5	5	5	5
molecule	1	14	14	14	13	12	6	6	6	6	6
	2	14	14	14	13	12	12	11	9	9	9
	3	14	14	14	13	12	12	11	9	9	9
teapot	1	15	11	11	11	9	8	8	5	5	5
	2	15	14	14	14	9	8	8	5	5	5
	3	20	14	14	14	9	8	8	8	8	8
ant	1	14	14	10	10	10	6	6	6	6	6
	2	14	14	10	10	10	10	10	9	6	6
	3	25	14	13	13	13	10	10	9	9	6
blond	1	15	11	9	3	3	3	3	3	3	3
	2	17	14	14	8	8	8	6	5	5	5
	3	20	20	14	11	8	8	6	5	5	5
jet	1	11	9	8	8	8	4	4	4	4	4
	2	18	11	8	8	8	7	7	6	6	6
	3	18	11	11	11	11	7	7	6	6	6
mandrill	1	19	14	8	7	6	5	5	5	5	4
	2	28	14	8	7	6	5	5	5	5	4
	3	32	14	11	7	6	5	5	5	5	4
peppers	1	10	9	8	7	7	7	7	7	7	6
	2	10	9	8	7	7	7	7	7	7	6
	3	10	9	8	7	7	7	7	7	7	6
mouse	1	18	13	13	9	9	8	7	7	6	6
	2	18	13	13	11	11	8	7	7	6	6
	3	18	13	13	11	11	11	7	7	6	6
rose	1	21	9	9	7	7	6	6	6	6	5
	2	31	27	20	18	13	6	6	6	6	5
	3	31	27	20	18	16	16	9	9	9	5

Table 6.77: Number of clusters using stopping rule 3 with original and *median* context values.

use the context values as well contained more smoother looking clusters. While the results when using only the context values were blurry in appearance, the use of the original values as well overcame this effect.

The image *molecule* produced good results for all four context types. For the *mean*, *median* and *weighted* contexts, a good result only occurred when two or three percentage decrease values were used when *perc* was 0.06-0.10 for the *mean* and *weighted* contexts, and when *perc* was 0.08-0.10 for the *median* context. The *regression* context produced the overall best result when *perc* was 0.07-0.10, regardless of how many percentage decrease values were considered. This result contained the smallest number of clusters, which also happened to be the same number of clusters as the best result when only using the original RGB values.

image	no. consec.	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
balls	1	19	10	10	10	9	7	5	5	4	4
	2	21	18	14	14	9	9	5	5	4	4
	3	21	18	14	14	12	12	5	5	4	4
Lenna	1	12	12	11	8	7	5	5	5	5	5
	2	17	14	11	10	7	7	5	5	5	5
	3	23	22	14	10	10	7	5	5	5	5
molecule	1	15	13	12	8	8	8	8	8	8	5
	2	18	15	12	12	12	10	10	10	10	10
	3	18	18	18	18	18	10	10	10	10	10
teapot	1	18	12	12	11	9	8	8	5	5	5
	2	18	17	17	11	9	8	8	5	5	5
	3	24	17	17	17	9	8	8	8	8	8
ant	1	15	13	12	11	11	6	6	6	6	6
	2	15	13	12	11	11	11	10	6	6	6
	3	39	13	12	11	11	11	10	9	9	6
blond	1	16	14	9	3	3	3	3	3	3	3
	2	18	16	9	9	7	7	6	5	5	5
	3	32	16	9	9	7	7	6	5	5	5
jet	1	17	8	8	8	4	4	4	4	4	4
	2	17	8	8	8	8	8	7	6	6	6
	3	27	16	12	11	11	8	7	6	6	6
mandrill	1	19	15	8	7	6	6	5	5	5	4
	2	24	15	10	7	6	6	5	5	5	4
	3	27	15	10	7	6	6	5	5	5	4
peppers	1	19	9	9	7	7	7	7	7	7	5
	2	19	9	9	7	7	7	7	7	7	5
	3	19	18	17	7	7	7	7	7	7	5
mouse	1	23	13	13	10	10	8	8	7	6	6
	2	23	13	13	10	10	10	8	7	6	6
	3	33	20	18	10	10	10	8	7	6	6
rose	1	15	11	9	9	6	6	6	6	6	5
	2	27	27	14	14	13	8	8	8	8	5
	3	40	32	24	18	13	13	13	8	8	8

Table 6.78: Number of clusters using stopping rule 3 with original and *weighted* context values.

For the image *teapot*, the best results for the *mean* and *weighted* contexts occurred when using one or two values of the percentage decrease when *perc* was set to 0.04. When using the *median* context the best result occurred only when two or three consecutive values of the percentage decrease were considered when *perc* was 0.02-0.04 and when using the *regression* context, the best result occurred only when one percentage decrease value was used and *perc* was 0.02-0.04. The *mean*, *weighted* and *regression* context results were all very similar, however the *median* context result contained more clusters, which didn't really improve the result. The context results were similar to the results using only the original RGB values for an equivalent number of clusters, except that not all the checkerboard squares were visible in the shadow region in the context results.

image	no. consec.	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
balls	1	17	11	11	11	10	8	5	5	4	4
	2	21	19	11	11	10	10	5	5	4	4
	3	21	19	14	11	10	10	10	10	4	4
Lenna	1	11	11	11	8	7	5	5	5	5	5
	2	15	15	14	10	7	7	5	5	5	5
	3	15	15	14	10	10	7	5	5	5	5
molecule	1	14	13	13	13	13	12	8	8	8	8
	2	20	13	13	13	13	12	8	8	8	8
	3	20	16	13	13	13	12	8	8	8	8
teapot	1	15	11	11	11	9	8	8	5	5	5
	2	20	19	19	13	9	8	8	5	5	5
	3	20	19	19	13	9	8	8	8	8	8
ant	1	15	13	11	11	11	6	6	6	6	6
	2	15	13	11	11	11	11	10	9	6	6
	3	21	13	11	11	11	11	10	9	9	6
blond	1	18	4	4	3	3	3	3	3	3	3
	2	18	16	11	3	3	3	3	3	3	3
	3	18	16	14	10	8	8	7	6	6	6
jet	1	13	9	4	4	4	4	3	3	3	3
	2	25	16	13	13	11	9	3	3	3	3
	3	25	16	13	13	11	9	8	7	7	7
mandrill	1	19	13	9	7	7	6	5	5	5	4
	2	21	15	9	7	7	6	5	5	5	4
	3	26	15	9	7	7	6	5	5	5	4
peppers	1	15	10	10	7	7	7	7	7	7	5
	2	19	14	13	9	7	7	7	7	7	5
	3	35	14	13	9	7	7	7	7	7	5
mouse	1	22	13	12	10	10	8	8	7	6	6
	2	26	20	12	10	10	10	8	7	6	6
	3	33	20	17	10	10	10	8	7	6	6
rose	1	15	15	9	9	6	6	6	6	6	5
	2	35	25	14	14	13	13	8	8	8	5
	3	35	28	25	18	13	13	8	8	8	8

Table 6.79: Number of clusters using stopping rule 3 with original and *regression* context values.

The image *ant* produced good results for the *mean* and *weighted* contexts when three consecutive values of the percentage decrease were considered when *perc* was set to 0.08-0.09. For the *median* and *regression* contexts, using two percentage decrease values when *perc* was 0.08 and using three percentage decrease values when *perc* was 0.08-0.09 produced the best results. In all four cases, the number of clusters was 9. The results with 6 clusters were also reasonable, except that part of the bottom of the mouse was grouped together with the black region. All of the results were very similar from a visual point of view. In comparison to the results using the original RGB values, the context results had more smooth looking segments.

For the image *blond*, the best results occurred when the number of clusters was between 8 and 10. For the *mean* and *weighted* contexts, the best results occurred

when *perc* was 0.03 regardless of how many percentage decrease values were used, and also 0.04 when two or three consecutive percentage decrease values were used. For the *median* context, the best results occurred when two or three percentage decrease values were used and *perc* was 0.05-0.06, and also when *perc* was 0.04 when two percentage decrease values were used. For the *regression* context, the best result occurred when three consecutive percentage decrease values were used when *perc* was 0.04. Each of the four results were very similar. The results using both the original and context values were much better than the results using only the original RGB values, as these results correctly identified the hair as a separate cluster in yellow, which did not occur when using only the original RGB values for an equivalent number of clusters.

For the image *jet*, the best segmentation results occurred when the number of clusters was 8 or 9. In the case of the *mean* and *weighted* contexts, this occurred when *perc* was 0.02-0.04, 0.02-0.06 and 0.06 when using one, two or three consecutive percentage decrease values, respectively. For the *median* context, the best result occurred when *perc* was 0.03-0.05 for one or two percentage decrease values. For the *regression* context, the best result occurred when *perc* was 0.02 for one percentage decrease value and 0.06 for two or three percentage decrease values. The results for the *mean*, *weighted* and *regression* contexts were the best as the writing on the plane in the *median* context result was not as clear as the other results. In comparison to the results using the original RGB values, the context results contained more smooth looking clusters, which was especially noticeable in the tail and wing of the plane.

For the image *mandrill*, the optimal value for *perc* was 0.03 for each of the four context types. For the *mean* and *weighted* contexts, the best results occurred when two or three consecutive percentage decrease values were used, for the *median* context,

the best result occurred when only three consecutive percentage decrease values were used, and for the *regression* context the best result occurred regardless of how many percentage decrease values were used. The results were all very similar. In comparison to the results using the original RGB values, the fur in the context results appeared more smooth looking.

For the image *peppers*, the best results were achieved when 7 clusters were formed. This result was obtained with each of the four context types regardless of how many consecutive percentage decrease values were used. Mostly, the values 0.04-0.09 were good values for *perc*, however, in the case of using the *regression* context with two or three percentage decrease values, the best result was achieved when *perc* was 0.05-0.09. All of the results were very similar, and in comparison to the results by the original RGB values, the context results had more smoother looking segments.

The best results for the image *mouse* occurred when the number of clusters was 7 when *perc* was set to 0.08, and when the *median* context was used 7 clusters were also obtained when *perc* was 0.07. This occurred for each of the four context types, regardless of how many consecutive percentage decrease values were used. Each of these four results were very similar in appearance and in comparison to the results using only the original RGB values, the context results had more smoother segments, but otherwise looked very similar.

For the image *rose*, the best results occurred when the number of clusters ranged between 8 or 9. For the *mean* context this occurred when *perc* was 0.06-0.09 and 0.07-0.10 for two and three percentage decrease values, respectively. For the *median* context this occurred when *perc* was 0.02-0.03 and 0.07-0.09 for one and three percentage decrease values, respectively. For the *weighted* context this occurred when *perc* was 0.06-0.09 and 0.08-0.10 for two and three percentage decrease values, re-

spectively. For the *regression* context this occurred when *perc* was 0.07-0.09 for two percentage decrease values and 0.07-0.10 for three percentage decrease values. Each of these results were very similar from a visual point of view. In comparison to the results using the original RGB values, the context results showed more smoother looking segments.

Figures 6.43-6.46 show the segmentation results for the *mean*, *median*, *weighted* and *regression* contexts, when used together with the original colour values, for various values of *perc* which produced the best visual segmentation results, respectively. In Figures 6.43-6.46 the *perc* values producing these results are indicated together with the number of consecutive percentage decrease values, which are shown in brackets. Generally, the results using both the original and context values tended to contain more smooth looking segments, which is an advantage over only using the original RGB values as we expect segments to be connected regions in the image. Overall, the *regression* context performed the best, producing good results for all of the images. The *mean*, *median* and *weighted* contexts also performed well with good results occurring for most images. For most of the cases, the results by the four context types were very similar, and no preference can be given to any one particular context type over the others. Generally, setting *perc* to be between 0.03 and 0.04 worked quite well for most images. The main exceptions to this occurred for the images *molecule*, *ant* and *mouse*, which required *perc* to be about 0.07 or 0.08.

6.4.1.4 Results Using Different Colour Spaces

All of the results in this section are based on only the use of the average variance in calculating the threshold and determining which cluster to split and also when only using mean cluster centres and Euclidean distance. The results using the RGB

colour space with mean cluster centres and Euclidean distance were presented in section 6.4.1.2.

Tables 6.80-6.82 show the number of clusters obtained for various values of *perc* when using the XYZ, KL and HSV colour spaces, respectively, where *perc* represents the threshold for the percentage decrease in intra-cluster distance. Following is a detailed analysis of the results for each image. The results are summarized in section 6.4.2.

image	no. consec.	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
balls	1	11	9	6	6	6	6	4	4	4	4
	2	11	11	9	9	9	9	9	6	6	6
	3	11	11	9	9	9	9	9	9	9	9
Lenna	1	8	8	7	6	6	6	6	6	5	5
	2	21	16	7	6	6	6	6	6	5	5
	3	21	16	10	6	6	6	6	6	5	5
molecule	1	20	19	16	14	13	7	7	7	7	7
	2	22	19	18	18	13	13	13	7	7	7
	3	22	22	18	18	18	16	13	13	13	13
teapot	1	23	12	12	9	9	8	4	4	4	4
	2	28	14	14	11	9	8	8	8	8	8
	3	28	28	28	17	9	8	8	8	8	8
ant	1	13	7	7	7	7	7	7	6	6	6
	2	41	16	15	15	12	12	12	6	6	6
	3	41	16	15	15	15	12	12	12	6	6
blond	1	8	7	4	4	4	3	3	3	3	3
	2	8	7	6	6	6	3	3	3	3	3
	3	8	7	6	6	6	6	6	6	6	6
jet	1	13	8	8	8	8	6	6	6	4	4
	2	15	13	8	8	8	8	8	8	8	8
	3	25	13	11	11	8	8	8	8	8	8
mandrill	1	21	14	13	9	9	6	6	6	6	6
	2	32	20	13	11	9	6	6	6	6	6
	3	83	30	13	11	9	6	6	6	6	6
peppers	1	11	9	8	7	5	5	5	5	5	5
	2	40	28	8	7	7	7	7	7	7	5
	3	48	37	16	7	7	7	7	7	7	5
mouse	1	24	14	11	10	10	8	8	8	7	6
	2	24	23	18	10	10	8	8	8	7	6
	3	29	23	18	10	10	8	8	8	7	6
rose	1	19	15	12	12	9	8	8	8	7	7
	2	41	24	23	18	12	8	8	8	7	7
	3	54	24	23	21	12	12	8	8	7	7

Table 6.80: Number of clusters using stopping rule 3 with the XYZ colour space.

The image *balls* achieved the best segmentations when using either the RGB or KL colour spaces. These two results also gave a similar number of clusters, 10 and 9, respectively. In the case of the KL colour space, the result was achieved regardless of how many consecutive percentage decrease values were required to be below *perc*.

image	no. consec.	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
balls	1	11	9	7	7	7	7	4	4	4	4
	2	13	9	9	9	9	9	6	6	4	4
	3	21	9	9	9	9	9	9	9	4	4
Lenna	1	8	8	8	7	7	6	5	5	5	5
	2	18	15	11	7	7	6	5	5	5	5
	3	18	15	11	11	11	6	5	5	5	5
molecule	1	20	18	18	14	14	12	8	8	8	8
	2	20	18	18	18	14	14	8	8	8	8
	3	20	18	18	18	14	14	14	14	8	8
teapot	1	17	10	10	10	6	6	6	3	3	3
	2	31	16	14	14	12	6	6	6	6	6
	3	31	19	14	14	12	12	10	10	10	10
ant	1	14	13	13	10	10	6	6	6	6	6
	2	14	13	13	12	12	12	12	9	6	6
	3	77	13	13	12	12	12	12	9	9	9
blond	1	10	10	4	4	3	3	3	3	3	3
	2	26	25	7	7	3	3	3	3	3	3
	3	26	25	7	7	7	7	7	6	6	6
jet	1	5	5	5	5	5	4	4	4	4	4
	2	11	5	5	5	5	4	4	4	4	4
	3	21	5	5	5	5	4	4	4	4	4
mandrill	1	20	14	11	7	7	6	5	5	5	5
	2	24	20	11	7	7	6	5	5	5	5
	3	24	20	11	11	10	6	5	5	5	5
peppers	1	11	10	9	8	7	7	7	7	7	6
	2	11	10	9	8	7	7	7	7	7	6
	3	38	10	9	8	7	7	7	7	7	6
mouse	1	15	14	10	10	9	9	8	6	6	6
	2	18	14	10	10	9	9	8	6	6	6
	3	18	17	14	13	9	9	8	6	6	6
rose	1	12	8	8	7	7	6	6	6	6	5
	2	43	21	20	7	7	6	6	6	6	5
	3	43	21	20	20	14	6	6	6	6	5

Table 6.81: Number of clusters using stopping rule 3 with the KL colour space.

The RGB colour space gave its best result when only taking one percentage decrease value with *perc* set to 0.01. The main difference with this result is that the pink ball was partly classified as belonging to the shadow segment, while with the KL colour space result, the pink ball was more smooth and connected. The XYZ colour space result containing 11 clusters was not too bad, and the HSV results tended to have noisy segments which were not as smooth looking as the other colour spaces. The best XYZ and HSV results occurred when *perc* was 0.01, regardless of how many consecutive percentage decrease values were considered. Also, a value of 0.02 produced the same number of clusters for the XYZ colour space when two or three consecutive percentage decrease values were required to be below *perc*.

The overall best segmentations for the image *Lenna* were produced when using the

image	no. consec.	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
balls	1	16	5	5	4	4	4	3	3	2	2
	2	24	8	8	4	4	4	3	3	2	2
	3	24	8	8	4	4	4	3	3	2	2
Lenna	1	7	6	6	6	6	4	4	3	3	3
	2	41	6	6	6	6	6	4	3	3	3
	3	49	22	10	10	10	10	4	3	3	3
molecule	1	7	5	5	5	5	5	5	4	4	4
	2	7	5	5	5	5	5	5	4	4	4
	3	10	5	5	5	5	5	5	4	4	4
teapot	1	16	13	12	11	9	6	6	6	6	4
	2	16	15	12	11	11	11	8	6	6	6
	3	24	15	15	11	11	11	11	6	6	6
ant	1	9	9	9	8	8	6	6	3	3	3
	2	11	11	11	8	8	8	8	8	8	8
	3	34	21	21	18	16	14	14	8	8	8
blond	1	5	3	3	3	3	3	3	3	3	3
	2	5	3	3	3	3	3	3	3	3	3
	3	5	3	3	3	3	3	3	3	3	3
jet	1	8	6	4	4	4	4	4	3	3	3
	2	18	14	13	11	10	10	10	3	3	3
	3	18	14	13	11	10	10	10	3	3	3
mandrill	1	20	13	7	7	7	7	5	4	4	3
	2	37	22	16	11	9	9	5	4	4	3
	3	66	22	22	11	9	9	5	4	4	3
peppers	1	9	8	8	8	7	4	4	4	4	4
	2	49	8	8	8	7	7	7	7	7	7
	3	49	13	8	8	7	7	7	7	7	7
mouse	1	5	5	5	4	4	4	4	4	4	4
	2	5	5	5	4	4	4	4	4	4	4
	3	21	13	13	4	4	4	4	4	4	4
rose	1	5	5	5	5	5	5	5	5	4	4
	2	51	26	17	17	17	16	14	14	4	4
	3	51	26	17	17	17	16	14	14	14	12

Table 6.82: Number of clusters using stopping rule 3 with the HSV colour space.

RGB and KL colour spaces when 6 clusters were obtained. The same value of 0.06 for *perc* produced 6 clusters for both the RGB and KL colour spaces, regardless of how many consecutive values of the percentage decrease were required to be below *perc*. The best result by the XYZ colour space was obtained when three consecutive values of the percentage decrease were required to be below *perc*, which was set to 0.03. The HSV colour space also required three consecutive values of the percentage decrease to obtain its best result when *perc* was between 0.03 and 0.06. However, this result was very noisy in appearance.

The image *molecule* produced good results for the RGB and KL colour spaces when 8 clusters were obtained, and both of these were virtually identical. When using the XYZ colour space, the best result which included all the different colours

in the molecule was only obtained when 13 clusters were achieved. The HSV colour space results were very poor, and even when 10 clusters were obtained, the result failed to distinguish the red part as a separate cluster.

For the image *teapot*, the best overall result occurred when using the KL colour space when 10 clusters were obtained. This was achieved when either one or three consecutive values of the percentage decrease were required to be below *perc*, with *perc* set to 0.02-0.04 and 0.07-0.10, respectively. The XYZ result with 9 clusters was poor as most of the reflection region was blue, and in the HSV result with 13 clusters, the shadow region was all one colour and failed to show the individual colours in the checker board.

For the image *ant*, the best results were produced by the XYZ and KL colour spaces which both contained 6 clusters. The RGB result with 5 clusters was also not bad, except that part of the bottom of the mouse was black. The HSV result with 8 clusters was very poor as it appeared quite noisy.

For the image *blond*, the best results occurred when either 6 or 7 clusters were obtained by the RGB, XYZ and KL colour spaces, which all produced very similar results. The XYZ colour space produced 7 clusters when *perc* was set to 0.02 regardless of how many percentage decrease values were used. The RGB colour space only produced 6 clusters when three consecutive percentage decrease values were required to be below *perc*, which was 0.08-0.10, and 7 clusters were obtained when two consecutive percentage decrease values were used when *perc* was 0.03-0.04. For the KL colour space, 7 clusters were obtained when either two or three consecutive values of the percentage decrease values were used, when *perc* was 0.03-0.04 and 0.03-0.07, respectively. The HSV colour space produced very poor results, with the maximum number of clusters being 5 when *perc* was 0.01, which was not enough clusters to

represent this image. A much larger number of clusters would be required than for any of the other colour spaces to produce a reasonable result for the HSV colour space.

For the image *jet*, the best segmentation results occurred for the RGB and KL colour spaces when the number of clusters were 5. This occurred regardless of how many consecutive values of the percentage decrease were used. The XYZ result with 8 clusters was also not bad, except that the tail of the plane was partly blue. For the HSV colour space, the result with 8 clusters was the best, however, this was very noisy in appearance.

For the image *mandrill*, the best segmentation results occurred when using the RGB or KL colour spaces, resulting in 8 and 11 clusters, respectively. In the case of the RGB colour space, this occurred when either two or three consecutive values of the percentage decrease were used when *perc* was 0.05-0.07, while for the KL colour space, this occurred regardless of how many percentage decrease values were used, with the values of *perc* being 0.03, and also 0.04 when three consecutive values were used. The XYZ and HSV results in general were poor as bits of red were included in the fur region.

For the image *peppers*, the best results were achieved when 6 or 8 clusters were formed. These results were obtained with the RGB and KL colour spaces, respectively. The XYZ and HSV results required many more clusters to obtain a reasonable result. For the RGB and KL results, the best results were achieved regardless of how many consecutive percentage decrease values were used. For the KL colour space, setting *perc* to 0.04 was the best and for RGB setting *perc* to 0.04-0.10 when only one percentage decrease value was used and 0.08-0.10 when two or three percentage decrease values were used was the best.

The best results for the image *mouse* occurred when 6 or 7 clusters were obtained. For the RGB and KL colour spaces, this occurred when *perc* was 0.08-0.10 for any number of percentage decrease values, while for the XYZ colour space, this occurred when *perc* was 0.09 for any number of percentage decrease values. Each of these results were very similar. The HSV results were in general very poor, as they were very noisy in appearance.

For the image *rose*, the best result occurred when using the RGB colour space when *perc* was 0.06 for any number of percentage decrease values, which resulted in 7 clusters. The KL result with 6 clusters was also good. This was produced when *perc* was 0.06-0.09 for any number of percentage decrease values. The XYZ colour space required a larger number of clusters to correctly classify all of the rose as separate from the background, while the HSV colour space result with 14 clusters was alright, however, the rose part didn't look that different from the RGB or KL results which had half the number of clusters. Most of the difference in fact can be seen in the background region, which isn't really adding anything useful to the result.

Figures 6.47-6.49 show the segmentation results for the XYZ, KL and HSV colour spaces for various values of *perc* which produced the best visual segmentation results, respectively. In Figures 6.47-6.49 the *perc* values producing these results are indicated together with the number of consecutive percentage decrease values, which are shown in brackets. As with the RGB colour space, a good segmentation can generally be found for any of the other colour spaces when requiring only one percentage decrease value to be below *perc*. Most of the images required *perc* to be between 0.01 and 0.05, however, there were some exceptions, such as *ant* and *mouse*. The HSV colour space performed very poorly, producing segmentation results which were noisy in appearance. The XYZ colour space was also not performing very well, while the

RGB and KL colour space produced very similar results. Therefore, there seems to be no real advantage in choosing a colour space other than RGB for colour image segmentation.

6.4.2 Summary of Results

Tables 6.83-6.86 show the values of *perc* which produced the best segmentation for each of the natural images for each combination of cluster centre and distance type, for each of the context types, when used on their own and together with the original colour values, and for each of the colour spaces, respectively. The result which is visually the best for each image is indicated in bold.

image	no. consec.	mean & Euc.	mean & abs.	med. & Euc.	med. & abs.
balls	1	0.01	**0.02**	**0.01**	0.01
	2	0.02-0.04	**0.02-0.04**	**0.02-0.05**	0.02
	3	0.02-0.04	**0.04-0.07**	**0.02-0.05**	0.02
Lenna	1	**0.06**	0.02-0.04	0.02	0.02-0.04
	2	**0.06**	0.03-0.04	0.02-0.05	0.01-0.04
	3	**0.06**	0.03-0.04	0.02-0.06	0.01-0.04
molecule	1	**0.07-0.10**	**0.05-0.10**	**0.08-0.10**	**0.05-0.10**
	2	**0.07-0.10**	0.07-0.10	**0.09-0.10**	**0.10**
	3	**0.09-0.10**	0.10	**0.09-0.10**	**0.10**
teapot	1	**0.07-0.08**	0.01	0.05-0.07	0.07-0.08
	2	**0.07-0.08**	0.02-0.03	0.08-0.09	0.07-0.08
	3	**0.07-0.10**	0.05	0.08-0.09	0.07-0.08
ant	1	**0.05-0.10**	**0.05-0.10**	0.06-0.10	**0.06-0.08**
	2	0.10	0.06-0.10	0.06-0.10	0.01-0.08
	3	0.10	0.06-0.10	0.01-0.10	0.09-0.10
blond	1	0.01	0.01-0.04	0.02	0.03-0.05
	2	0.03-0.04	**0.06-0.10**	0.02-0.04	0.07-0.10
	3	**0.08-0.10**	**0.06-0.10**	**0.10**	0.07-0.10
jet	1	**0.02-0.05**	0.01	**0.01-0.05**	**0.02-0.04**
	2	**0.02-0.10**	0.01-0.04	**0.01-0.05**	**0.02-0.04**
	3	**0.02-0.10**	0.03-0.10	0.03-0.05	0.03-0.04
mandrill	1	0.04	0.04	**0.04**	**0.04**
	2	**0.05-0.07**	**0.05-0.08**	0.03-0.06	0.02-0.04
	3	**0.05-0.07**	**0.06-0.08**	**0.07-0.08**	**0.06-0.08**
peppers	1	**0.04-0.10**	**0.07-0.10**	**0.10**	0.06-0.10
	2	**0.08-0.10**	**0.07-0.10**	**0.10**	**0.08-0.10**
	3	**0.08-0.10**	**0.07-0.10**	**0.10**	**0.08-0.10**
mouse	1	**0.08-0.10**	**0.09-0.10**	0.04	0.04-0.07
	2	**0.08-0.10**	**0.09-0.10**	**0.08-0.10**	**0.10**
	3	**0.08-0.10**	**0.09-0.10**	**0.08-0.10**	**0.10**
rose	1	**0.06**	**0.06-0.10**	**0.08-0.09**	**0.06-0.10**
	2	**0.06**	0.07-0.10	**0.08-0.09**	**0.06-0.10**
	3	**0.06**	0.07-0.10	**0.08-0.09**	**0.09-0.10**

Table 6.83: Values of *perc* giving best segmentation for all four combinations of cluster centre and distance type.

The results produced by some of the *perc* values shown in Tables 6.83-6.86 match

image	no. consec.	*mean*	*median*	*weighted*	*regression*
balls	1	**0.02**	0.03-0.04	**0.02**	0.02-0.05
	2	**0.03-0.06**	0.04	**0.03-0.06**	0.02-0.05
	3	0.03-0.06	0.04	0.03-0.06	0.03-0.05
Lenna	1	0.04	0.03	0.04	0.03
	2	**0.05-0.06**	0.03	**0.05-0.06**	**0.05**
	3	**0.05-0.06**	0.03-0.05	**0.05-0.06**	**0.05-0.06**
molecule	1	0.09-0.10	**0.07-0.10**	0.09-0.10	0.09-0.10
	2	0.09-0.10	**0.09-0.10**	0.09-0.10	0.09-0.10
	3	0.09-0.10	**0.09-0.10**	0.09-0.10	0.09-0.10
teapot	1	**0.04-0.05**	0.05-0.07	**0.04-0.05**	0.04
	2	0.04-0.05	0.05	0.04-0.05	0.05-0.06
	3	0.04-0.06	0.05	0.04-0.06	0.05-0.06
ant	1	0.07-0.10	0.07-0.10	0.07-0.10	0.06-0.10
	2	**0.09-0.10**	0.09-0.10	**0.09-0.10**	0.09-0.10
	3	**0.09-0.10**	0.10	**0.09-0.10**	0.09-0.10
blond	1	0.04	**0.03-0.04**	0.04	0.01-0.03
	2	0.04-0.05	**0.03-0.08**	0.04-0.05	0.05
	3	0.04-0.05	**0.03-0.08**	0.04-0.05	0.05
jet	1	0.03-0.08	0.01	0.03-0.08	0.02
	2	0.05-0.08	0.04-0.07	0.08	**0.06-0.07**
	3	0.06-0.08	0.04-0.07	0.08	**0.06-0.07**
mandrill	1	**0.04**	0.02	**0.04**	**0.04**
	2	**0.04**	0.04-0.05	**0.04**	**0.04**
	3	**0.04**	0.04-0.05	**0.04**	**0.04**
peppers	1	0.04-0.10	**0.05-0.10**	0.04-0.10	0.04-0.10
	2	0.06-0.10	**0.05-0.10**	0.06-0.10	0.06-0.10
	3	0.06-0.10	**0.06-0.10**	0.06-0.10	0.06-0.10
mouse	1	**0.08-0.09**	**0.08-0.09**	**0.08-0.09**	**0.08-0.09**
	2	**0.08-0.09**	**0.08-0.09**	**0.08-0.09**	**0.08-0.09**
	3	**0.08-0.09**	**0.08-0.09**	**0.08-0.09**	**0.08-0.09**
rose	1	**0.04-0.05**	**0.03-0.05**	**0.04-0.05**	**0.04**
	2	0.03-0.05	0.06	0.03-0.05	**0.04-0.05**
	3	0.06	**0.07**	0.06	0.08-0.10

Table 6.84: Values of *perc* giving best segmentation when using only context values.

with the results shown in Figures 6.35-6.49, where the particular combination of *perc* values and the number of consecutive percentage decrease values shown in the figures corresponds to the best results produced.

For stopping rule 3, a threshold of about 0.06 seems to give reasonable results for most of the natural images. If we take this value as the default value, the result is a method which is totally automated. However, the option of being able to modify this value should be available should the need arise to either increase or decrease the number of clusters. The methodology as described allows us to achieve this by requesting this value as an input parameter.

The *regression* context type is performing quite well for most of the images, especially when both the original and context values are taken into account to perform the

image	no. consec.	*mean*	*median*	*weighted*	*regression*
balls	1	**0.02-0.04**	**0.02-0.03**	0.02-0.04	**0.02-0.04**
	2	**0.02-0.04**	0.03	0.03-0.04	**0.03-0.04**
	3	0.03-0.04	0.03-0.05	**0.05-0.06**	**0.04**
Lenna	1	0.03	0.02	0.03	0.01-0.03
	2	**0.04**	**0.04-0.05**	**0.04**	**0.04**
	3	**0.05**	**0.04-0.05**	**0.04-0.05**	**0.04-0.05**
molecule	1	0.04	0.05	0.03	**0.07-0.10**
	2	0.06-0.10	0.08-0.10	0.06-0.10	**0.07-0.10**
	3	0.06-0.10	0.08-0.10	0.06-0.10	**0.07-0.10**
teapot	1	**0.04**	0.02-0.04	**0.04**	**0.02-0.04**
	2	**0.04**	0.02-0.04	**0.04**	0.05
	3	0.04	0.02-0.04	0.05	0.05
ant	1	0.06-0.10	0.06-0.10	0.06-0.10	0.06-0.10
	2	0.08-0.10	**0.08**	0.08-0.10	**0.08**
	3	**0.08-0.09**	**0.08-0.09**	**0.08-0.09**	**0.08-0.09**
blond	1	**0.03**	0.03	**0.03**	0.01
	2	**0.03-0.04**	**0.04-0.06**	**0.03-0.04**	0.03
	3	**0.03-0.04**	**0.05-0.06**	**0.03-0.04**	**0.04**
jet	1	**0.02-0.04**	0.03-0.05	**0.02-0.04**	**0.02**
	2	**0.02-0.06**	0.03-0.05	**0.02-0.06**	**0.06**
	3	**0.06**	0.02-0.05	**0.06**	**0.06**
mandrill	1	0.02	0.02	0.02	**0.03**
	2	**0.03**	0.02	**0.03**	**0.03**
	3	**0.03**	**0.03**	**0.03**	**0.03**
peppers	1	**0.04-0.09**	**0.04-0.09**	**0.04-0.09**	**0.04-0.09**
	2	**0.04-0.09**	**0.04-0.09**	**0.04-0.09**	**0.05-0.09**
	3	**0.04-0.09**	**0.04-0.09**	**0.04-0.09**	**0.05-0.09**
mouse	1	**0.08**	**0.07-0.08**	**0.08**	**0.08**
	2	**0.08**	**0.07-0.08**	**0.08**	**0.08**
	3	**0.08**	**0.07-0.08**	**0.08**	**0.08**
rose	1	0.03-0.04	**0.02-0.03**	0.03-0.04	0.03-0.04
	2	**0.06-0.09**	0.05	**0.06-0.09**	**0.07-0.09**
	3	**0.07-0.10**	**0.07-0.09**	**0.08-0.10**	**0.07-0.10**

Table 6.85: Values of *perc* giving best segmentation when using original and context values.

clustering. When using only the percentage decrease in intra-cluster distance, setting *perc* to 0.04-0.06 works well when using only the context values, and setting *perc* to 0.03-0.04 when using both the original and context values works quite well. When using the context values, especially when on their own, the segmentation results look smoother and the regions more connected, which is a clear advantage over the results using only the original RGB values. However, when only the context values are used, the segmented results look slightly blurred, which could be a problem when fine details, such as the writing on the plane in the image *jet*, need to be extracted. Another problem that can occur, especially when the background and foreground colours are similar, is that parts of the objects get grouped together with the background.

The results by different colour spaces indicate that the RGB colour space performs

image	no. consec.	RGB	XYZ	KL	HSV
balls	1	0.01	0.01	**0.02**	0.01
	2	0.02-0.04	0.01-0.02	**0.02-0.06**	0.01
	3	0.02-0.04	0.01-0.02	**0.02-0.08**	0.01
Lenna	1	**0.06**	0.01-0.02	**0.06**	0.01
	2	**0.06**	0.02	**0.06**	0.01
	3	**0.06**	0.03	**0.06**	0.03-0.06
molecule	1	**0.07-0.10**	0.05	**0.07-0.10**	0.01
	2	**0.07-0.10**	0.05-0.07	**0.07-0.10**	0.01
	3	**0.09-0.10**	0.07-0.10	**0.09-0.10**	0.01
teapot	1	0.07-0.08	0.04-0.05	**0.02-0.04**	0.02
	2	0.07-0.08	0.05	0.05	0.04-0.06
	3	0.07-0.10	0.05	**0.07-0.10**	0.04-0.07
ant	1	0.05-0.10	**0.08-0.10**	**0.06-0.10**	0.04-0.05
	2	0.10	**0.08-0.10**	**0.09-0.10**	0.04-0.10
	3	0.10	**0.09-0.10**	0.08-0.10	0.08-0.10
blond	1	0.01	**0.02**	0.01-0.02	0.01
	2	0.03-0.04	**0.02**	**0.03-0.04**	0.01
	3	**0.08-0.10**	**0.02**	**0.03-0.07**	0.01
jet	1	**0.02-0.05**	0.02-0.05	**0.01-0.05**	0.01
	2	**0.02-0.10**	0.03-0.10	**0.02-0.05**	0.05-0.07
	3	**0.02-0.10**	0.05-0.10	**0.02-0.05**	0.05-0.07
mandrill	1	0.04	0.03	**0.03**	0.01
	2	**0.05-0.07**	0.03	**0.03**	0.03
	3	**0.05-0.07**	0.03	**0.03-0.04**	0.02-0.03
peppers	1	**0.04-0.10**	0.01	**0.04**	0.01
	2	**0.08-0.10**	0.02	**0.04**	0.01
	3	**0.08-0.10**	0.03	**0.04**	0.02
mouse	1	**0.08-0.10**	**0.09**	**0.08-0.10**	0.01-0.03
	2	**0.08-0.10**	**0.09**	**0.08-0.10**	0.01-0.03
	3	**0.08-0.10**	**0.09**	**0.08-0.10**	0.02-0.03
rose	1	**0.06**	0.01	**0.06-0.09**	0.01-0.08
	2	**0.06**	0.04	**0.06-0.09**	0.07-0.08
	3	**0.06**	0.04	**0.06-0.09**	0.07-0.09

Table 6.86: Values of *perc* giving best segmentation for different colour spaces.

quite well and there is no real benefit in selecting a different colour space in which to perform the segmentation. The visual results indicated that the RGB and KL colour spaces perform very similarly, and the HSV colour space tends to result in the segmented images looking very noisy. The XYZ colour space could only obtain good results for *ant*, *blond* and *mouse*.

When using this stopping rule, the method is a very simple technique which can be easily extended to cope with any dimensionality of images, and for this reason, this method may also be used for multispectral images. These methods may also have wide ranging applications with respect to medical images by, for example, finding abnormalities, which should be grouped as a separate cluster.

6.5 Comparison of Different Merging and Splitting Methods

In this section, the results for the various approaches taken to clustering-based colour image segmentation, being merging and splitting methods, are compared. The comparison is not concerned with the criteria presented in chapter 5 and this chapter to determine the optimal number of clusters, but is more concerned with establishing whether the merging approach or splitting approach is more suitable for colour image segmentation in general. The results presented in chapter 5 and this chapter indicate the superiority of the mean cluster centres and Euclidean distance combination from a visual point of view, so the comparison of the results by the different methods is based only on the results obtained by using mean cluster centres and Euclidean distance.

In addition to the survey which was conducted to determine the optimal range for the number of clusters, results of which were shown in chapter 3, a second visual analysis survey was conducted with the same ten participants. This survey was aimed at investigating which method produced the best results and ultimately was aimed at trying to determine if there is one clustering method which produces superior results over all the others in relation to colour image segmentation. For this part of the survey, a number of results were shown for each image, with the comparison between methods being based on a comparable number of clusters. The survey was based on the results by the two merging methods, which are referred to as $merge_1$ and $merge_2$ and the splitting method involving the *K-means* step, which is referred to as *splitkm*. The results by these methods are also compared to a version of the splitting method which does not have a *K-means* step, but rather has only a single pass through the data to reassign pixels after the split procedure. This method is referred to as *split*.

Results by the *snob* method were also shown for comparison purposes. Table 6.87 shows the percentage of responses for each method for each of the natural images with all of the different numbers of clusters. The number of clusters in general were chosen to be in the range for the desired number of clusters, as indicated in Table 3.1 in chapter 3, and also match with the number of clusters indicated by the validity measure which is discussed in chapter 7. We also chose one segmentation result which had a smaller number of clusters and one segmentation result which had a larger number of clusters. In some cases the $merge_2$ method produced a slightly different number of clusters. Where this occurred, the number of clusters is indicated in that column. The number of clusters produced by *snob* was also different, and mostly very much larger than those shown in Table 6.87. Therefore, the column indicating the number of clusters does not apply to the *snob* results, of which there is only one for each image, and the actual number of clusters for these results are shown in appendix C. The *snob* method required the distribution of the data to be specified as a parameter, and using a continuous distribution resulted in a number of clusters which was almost equal to the number of colours in the image, which resulted in practically no clustering taking place, so all the results shown in appendix C are based on the use of a discrete distribution.

For *balls*, the *split* method without the *K-means* step performed well with the correct level of segmentation at 11 clusters indicating a similarity between the $merge_1$ method and the *split* method. For *Lenna* the *splitkm* method obtained the most responses for the correct level of segmentation, followed closely by the *split* method. Overall, both the split methods performed better than the merge methods for *Lenna*. The image *molecule* achieved its best result using the $merge_2$ method with 6 clusters, which 90% of the people surveyed indicated a preference for. For a higher number

image	clusters	$merge_1$	$merge_2$	*splitkm*	*split*	*snob*
balls	8	20%	10%		70%	
	11	40%	-	10%	50%	
	14	20%	10%	40%	30%	
Lenna	4		50%	30%	20%	
	8		-	60%	40%	
	11			60%	40%	
molecule	6	10%	90%			
	8	90%	-	10%		
	10	40%	-	50%	10%	
teapot	4	50%	-	50%		
	8	30%	20%	50%		
	12	50%	-	50%		
ant	4		-	90%	10%	
	7	20%	20% (6)	50%	10%	
	10	40%	20%	20%	20%	
blond	2		-	30%	70%	
	5		-	30%	70%	
	8			100%		
jet	3		90%	10%		
	6	10%	-	60%	30%	
	10	20%		40%	40%	
mandrill	5	40%	-	30%	30%	
	9	20%	(11)	30%	50%	
	13	10%	(15)	50%	40%	
peppers	3	50%	-	20%	30%	
	6	10%	-	20%	70%	
	9	20%		40%	40%	
mouse	3		-	90%	10%	
	6		-	30%	70%	
	9		10%	70%	20%	
rose	3	20%	-	30%	50%	
	6	40%	-	40%	20%	
	9	30%	-	50%	20%	

Table 6.87: Responses for survey of best method.

of clusters, the $merge_1$ method performed well. For *teapot* the responses were evenly spread between the, *splitkm* method and the merge methods. For *ant*, the *splitkm* method performed quite well for the correct level of segmentation, with the $merge_1$ method receiving the highest percentage of responses for the result with 10 clusters. For the image *blond*, the split methods received 100% of the responses for each of the results shown, with the method without the *K-means* step receiving a majority of the responses for 2 and 5 clusters and the method with the *K-means* step receiving 100% of the responses for the result with 8 clusters. The *splitkm* method is favoured for *jet* for the correct level of segmentation, however, for a higher number of clusters such as 10, the responses were even between the two split methods. For the image *mandrill*, the responses were evenly spread between the $merge_1$, *splitkm* and *split* methods for

the result with 5 clusters, however, the results were fairly evenly shared between the two split methods for the results with 9 and 13 clusters. For 6 clusters, the image *peppers* produced the most responses for the *split* method, while for 9 clusters the two split methods received the equal highest responses. For the image *mouse*, the two splitting methods outperformed the merge methods with the method with the *K-means* step being superior for the results with 3 and 9 clusters and the method without the *K-means* step being superior for the result with 6 clusters. For the image *rose*, the results were split between the $merge_1$ method and the *splitkm* method for the results with 6 clusters, while there was a clear preference for the *splitkm* method for the result with 9 clusters.

Of particular interest is the percentage of responses obtained for the split method without the *K-means* step. For many of the images, a high percentage of responses were obtained, however, results for some of the synthetic images using this method failed to obtain the correct segmentation. The images *molecule* and *teapot* did not receive many responses for the split method without the *K-means* step, which is not surprising since these images are more like synthetic images than natural images and the failing of this method is more pronounced for synthetic types of images.

No preference was given to any of the results by *snob*. In general, all of the other methods produced better segmentation results, even though a much smaller number of clusters were formed. Not only did the *snob* results have more clusters, the visual results were very poor, especially in the cases of *balls*, *molecule*, *teapot* and *rose*. Many of the results by *snob* tended to be noisy, especially for *balls*, *Lenna*, *jet*, *mandrill*, *peppers* and *rose*. Results were also obtained for *snob* when incorporating the context information and, for many of the images, great improvement in the segmented results were achieved, especially when using both the original and context values. Also, these

results did not suffer as much by the problem of being noisy. In any case, the results by the *snob* method still contained far too many segments, and the proposed methods could obtain better results with far fewer clusters. Overall, the split method with *K-means* performed very well, from a visual point of view, for a majority of the natural images.

Chapter 7

Cluster Validity for Determining the Number of Clusters

The criteria presented in the previous chapters make use of either the inter-cluster distance or the intra-cluster distance to determine the optimal number of clusters. The use of the inter-cluster distance enabled a clustering to be formed in which the clusters were well separated, and the use of the intra-cluster distance enabled a clustering which produced compact clusters to be formed. Ideally, what we want is both compact and well separated clusters, so both of these measures should be used in tandem to determine the number of clusters in an image. In this chapter, a simple validity measure, based on both the intra-cluster and inter-cluster distance measures, will be presented which allows the number of clusters to be determined automatically. The basic procedure involves producing all the segmented images for 2 up to K_{max} clusters, and calculating the validity measure to determine which is the best clustering by finding the minimum value for the measure. The validity measure is tested for the synthetic images for which the number of clusters is known, and is also implemented for the natural colour images. The results in this chapter will be based on the segmentation results by the splitting method presented in chapter 6.

7.1 Proposed Cluster Validity Measure

One of the limitations of validity measures, such as the Davies-Bouldin index [DB79] and Dunn's indexes [Dun74, BP98], when applied to colour image data is that there is a tendency to select the result with a small number of clusters, such as 2 or 3. Therefore, we propose a modification to the standard ratio between the intra-cluster and inter-cluster distances by incorporating a multiplier function which is a function of the number of clusters and has the form

$$V = y \times \frac{intra}{inter} \tag{7.1}$$

where y represents a function of the number of clusters.

7.1.1 Basic Validity Criterion

For the *basic* validity criterion, the function for y is taken as $y = 1$, which represents only the ratio between the intra-cluster and inter-cluster distances. This is the classical use of such measures, and only the intra-cluster and inter-cluster distances need to be defined.

A good indication of cluster compactness is given by the distances of all points from their respective cluster centre. If all of the distances are small, the clusters can be considered compact. Consequently, we take the average of all of these distances, defined as

$$intra = \frac{1}{N} \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} \| \mathbf{x} - \mathbf{z}_i \|^2 \tag{7.2}$$

where N is the number of pixels in the image, K is the number of clusters, and $\mathbf{z}_i$ is the cluster centre of cluster C_i. We obviously want to minimize this measure. This average measure is the typical formulation which is considered to represent the

variability of the clusters, and it is based on the Euclidean distance measure.

We also need to measure the inter-cluster distance, or the distance between clusters, which we want to be as big as possible. Typically, when considering the inter-cluster distance, the average measure is used to represent the average cluster separation. However, the average can be greatly affected by one cluster being very far away from all the other clusters, while the other clusters are all close to each other. So if we maximize the average inter-cluster distance, this will not necessarily ensure adequate cluster separation. Therefore, the proposed measure only considers the minimum inter-cluster distance, defined as

$$inter = \min(\| \mathbf{z}_i - \mathbf{z}_j \|^2), \quad i = 1, 2, \ldots, K\text{-}1, j = i\text{+}1, \ldots, K \tag{7.3}$$

Maximizing this measure will automatically ensure that all of the inter-cluster distances are sufficiently large. This measure is also based on Euclidean distance. Since we want to minimize the intra-cluster distance and maximize the inter-cluster distance, the overall validity measure in equation 7.1 must be minimized.

7.1.2 Modified Validity Criterion

The modification of the *basic* validity criterion involves only a different definition of the function represented by y. The main reason for the modification is due to the tendency of measures which only take the *basic* ratio between the intra-cluster and inter-cluster distances to select a small number of clusters, such as 2 or 3. We therefore want to incorporate a penalty function to allow results with small numbers of clusters to be penalized in favour of results with more clusters, resulting in more visually appealing results. Therefore, the function for y must be a function of the number of

clusters, K, where this function has the property that it is large for small values of K, and negligible thereafter. This function should also be continuously decreasing so that most of the modification occurs for the smallest numbers of clusters. One such function which has this property is the function for the Gaussian curve, given by

$$N(\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left[-\frac{(K-\mu)^2}{2\sigma^2}\right]} \tag{7.4}$$

where we use K instead of the typical variable x since this should be a function of the number of clusters, denoted by K. The mean, μ, and standard deviation, σ, need to be supplied for this function. In our case, we are only interested in the values from 2 upwards, and in most cases the number of clusters selected as the optimum result by using only the *basic* ratio is in the range 2-4, which will be shown in section 7.2.1.2, which does not match with the visual analysis survey discussed in chapter 3. We require the function to be continuously decreasing, and this means that the mean, μ, cannot be below 2, and since in most cases the *basic* ratio at 2 is the smallest, we want the maximum value of the Gaussian function to occur at 2, so we set μ to 2. Since the number of clusters in the range 2-4 are the most problematic, we must ensure that the validity measure for these values will be increased. The Gaussian curve has the property that 95% of the data fall between $\mu \pm 2\sigma$ and 99.7% of the data fall in the range $\mu \pm 3\sigma$. If we set σ to 1, this ensures that the multiplier will have an effect for 2, 3, and 4 and by the time we get to 5, which is three standard deviations from the mean, there should not be too much effect any more. This is ideal because the visual analysis survey, for which the results were shown in chapter 3, indicated that we are not interested in the number of clusters below 5. On its own the Gaussian function will not be adequate to change the values, because this

function has very small values. We therefore need some constant multiplier to increase the value suitably, and we also add 1 to the final result so that once the Gaussian multiplier has no effect, for larger values of K, then we are maintaining the original values of the ratio between the intra-cluster and inter-cluster distances. The function which we will use as the multiplier will be

$$y = c \times N(2, 1) + 1 \tag{7.5}$$

Figure 7.1 shows this function for different values of c.

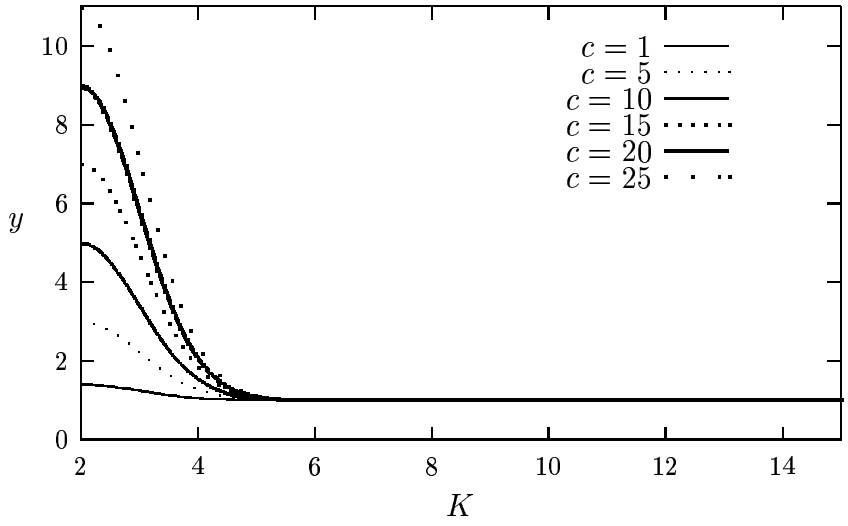


Figure 7.1: Gaussian curves represented by $y = c \times N(2, 1) + 1$ for different values of c.

We can see from Figure 7.1 that there is generally no effect on the multiplier function after about 5 to 6, as it reaches a constant value of 1, no matter what value of the constant, c, is used.

7.2 Results

In order to test the proposed measure, the basic procedure involves first producing the segmented images for 2 up to K_{max} clusters, where K_{max} is an upper limit on the

number of clusters. For testing purposes we use the *K-means* procedure as described in chapter 6. We then calculate the validity measure to determine which is the best clustering, and, therefore, what the optimal value of K is.

7.2.1 Basic Validity Criterion (y = 1)

7.2.1.1 Synthetic Images

This method was first implemented with synthetic images for which the ideal clustering was known beforehand. The six synthetic images shown in Figure 1.1 were used. For these images K_{max} was set to 25 as the ideal numbers of clusters are obviously well below this. All the results in this section are based on the use of the RGB colour space. Tables 7.1-7.4 show the *basic* validity measure for the synthetic images with the minimum values shown in bold, indicating the optimal number of clusters. Table 7.1 shows the *basic* validity measure values when using mean cluster centres and Euclidean distance, Table 7.2 shows the *basic* validity measure values when using mean cluster centres and absolute distance, Table 7.3 shows the *basic* validity measure when using median cluster centres and Euclidean distance and Table 7.4 shows the *basic* validity measure values when using median cluster centres and absolute distance. Figures 7.2-7.5 show the line diagrams corresponding to the values in Tables 7.1-7.4, respectively. It can be seen that the minimum values of the *basic* validity measure occurred at the correct number of clusters for each of the synthetic images. The Davies-Bouldin and Dunn's indexes were both implemented for these synthetic images which also resulted in their optimum values occurring at the correct number of clusters.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	**0.000392**	0.289912	0.153935	0.500452	0.394235	0.654147
3	0.680440	0.049309	0.120445	0.375647	0.450797	0.485759
4	0.498287	**0.000298**	0.132945	0.250706	0.330922	0.337446
5	0.409127	0.363747	0.095588	0.188158	0.257720	0.336405
6	0.922137	0.338716	**0.000312**	0.125706	0.143841	0.286343
7	0.855184	0.310998	0.377868	0.063244	0.150703	0.285098
8	0.782713	0.281818	0.367606	**0.000778**	0.083137	0.276478
9	0.578070	0.265686	0.354982	1.001002	0.053614	0.240530
10	0.548858	0.206981	0.342456	0.979171	**0.003205**	0.247285
11	0.522025	0.194218	0.319830	0.948082	0.577474	0.190489
12	0.442293	0.175007	0.296227	0.925747	0.557141	0.137444
13	0.535889	0.209963	0.288000	0.894988	0.540757	0.084637
14	0.545230	0.196616	0.280057	0.863515	0.515677	0.030784
15	0.401000	0.189058	0.272168	0.850674	0.503682	**0.005109**
16	0.391036	0.181745	0.220681	0.819233	0.478607	0.499132
17	0.380859	0.169670	0.204212	0.798438	0.742277	0.481920
18	0.403085	0.150898	0.197910	0.778188	0.717468	0.455533
19	0.447275	0.163999	0.184319	0.748777	0.677559	0.458063
20	0.456697	0.202190	0.176153	0.728933	0.668571	0.449974
21	0.378426	0.192370	0.164220	0.702496	0.656066	0.433102
22	0.340766	0.187323	0.178206	0.675108	0.641690	0.680397
23	0.342556	0.187171	0.175024	0.653813	0.624959	0.662615
24	0.347003	0.182439	0.171842	0.553785	0.613021	0.635521
25	0.346816	0.174063	0.165488	0.540072	0.601995	0.627609

Table 7.1: *Basic* validity measure for synthetic images using mean cluster centres and Euclidean distance.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	**0.000392**	0.289912	0.156799	0.500452	0.394235	0.654147
3	0.680440	0.049309	0.120445	0.375647	0.450797	0.485759
4	0.494554	**0.000298**	0.066875	0.250706	0.330922	0.337446
5	0.447120	0.363747	0.037860	0.188158	0.257720	0.336405
6	0.474217	0.338716	**0.000312**	0.125706	0.143841	0.286343
7	0.866605	0.310998	0.377868	0.063244	0.150703	0.285098
8	0.794245	0.282867	0.368442	**0.000778**	0.083137	0.239646
9	0.753329	0.267304	0.355792	1.001002	0.053614	0.280855
10	0.561017	0.257079	0.343241	0.979171	**0.003205**	0.399315
11	0.530698	0.241537	0.319148	0.948082	0.577474	0.308811
12	0.479169	0.186680	0.296309	0.925747	0.557141	0.131938
13	0.538808	0.213503	0.288601	0.894988	0.541163	0.081215
14	0.525024	0.193736	0.280751	0.863515	0.516083	0.030721
15	0.535140	0.184606	0.273110	0.850674	0.507097	**0.005109**
16	0.487515	0.177899	0.231451	0.819233	0.482022	0.500665
17	0.464648	0.165835	0.211276	0.799041	0.747751	0.483438
18	0.455752	0.161868	0.199089	0.779441	0.732320	0.456511
19	0.509262	0.159174	0.193911	0.750882	0.708245	0.458404
20	0.466383	0.151778	0.191200	0.731781	0.668335	0.452958
21	0.450806	0.169951	0.186796	0.705953	0.659130	0.436085
22	0.424536	0.180207	0.204380	0.679183	0.645121	0.685548
23	0.401117	0.171433	0.202743	0.658494	0.635390	0.675514
24	0.417869	0.166859	0.199019	0.551184	0.619191	0.648409
25	0.396345	0.173951	0.185944	0.541093	0.604020	0.631826

Table 7.2: *Basic* validity measure for synthetic images using mean cluster centres and absolute distance.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	**0.000392**	0.410278	0.140133	0.500550	0.390288	0.456813
3	0.809602	0.086341	0.133055	0.375574	0.331773	0.327728
4	0.571027	**0.000299**	72.651607	0.250618	0.252036	0.264259
5	0.414329	0.395646	0.048202	0.188122	0.199031	0.464241
6	0.478842	0.368849	**0.000312**	0.125650	0.143428	0.399549
7	0.930575	0.337146	0.421313	0.063197	0.126303	0.562651
8	0.874557	0.307293	0.407750	**0.000778**	0.090874	0.484492
9	0.817822	0.290773	0.393984	1.118297	0.081591	0.388077
10	0.644368	0.265818	0.380333	1.084219	**0.003208**	0.314997
11	0.618323	0.251759	0.353340	1.050182	0.738299	53.370647
12	0.466004	0.184505	0.327984	1.016483	0.481489	0.208785
13	0.611865	0.305982	0.319518	0.982832	0.686272	0.137275
14	0.592176	0.287256	0.310986	0.948769	0.654983	0.030619
15	0.832784	0.277152	0.294482	0.915856	0.639643	**0.005106**
16	0.816622	0.228415	0.251485	0.882465	0.608369	0.464146
17	0.799380	0.246610	0.245046	0.860476	0.920281	0.503806
18	0.783055	0.237648	0.238893	0.838635	0.889006	0.492916
19	0.731048	0.219698	0.219937	0.817868	0.840407	0.575871
20	0.715117	0.231349	0.213850	0.797081	0.826343	0.555048
21	0.698270	0.225166	0.168905	0.775066	0.808832	0.543625
22	0.685806	0.220294	0.226113	0.754367	0.779129	0.829500
23	0.633272	0.209882	0.215244	0.733064	0.758927	0.797019
24	0.621161	0.204631	0.211572	0.625556	0.745340	0.776396
25	0.609760	0.195453	0.204141	0.610508	0.731338	0.756128

Table 7.3: *Basic* validity measure for synthetic images using median cluster centres and Euclidean distance.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	**0.000392**	0.410278	0.140133	0.500550	0.390288	0.456813
3	0.809602	0.086341	0.148319	0.375574	0.331818	0.327728
4	0.572626	**0.000299**	0.097775	0.250618	0.252375	0.264158
5	0.426257	0.395646	0.049473	0.188122	0.197582	0.464244
6	0.484517	0.368849	**0.000312**	0.125650	0.143428	0.461176
7	0.943188	0.337146	0.421313	0.063197	0.126307	0.334626
8	0.882645	0.327311	0.407750	**0.000778**	0.090882	0.272894
9	0.826522	0.310867	0.393984	1.118297	0.081598	0.152421
10	0.796903	0.287875	0.380333	1.084219	**0.003208**	64.987084
11	0.760806	0.259596	0.353340	1.050182	0.738299	0.283708
12	0.639309	0.239080	0.328936	1.016483	0.457365	28.568707
13	0.760543	0.331317	0.320760	0.982832	0.686915	0.144484
14	0.599265	0.320317	0.331162	0.948769	0.655626	0.037036
15	0.585058	0.299946	0.314783	0.915856	0.642680	**0.005106**
16	0.642934	0.280561	0.306598	0.882465	0.611405	0.624951
17	0.599993	0.270595	0.299763	0.913577	0.925026	0.603532
18	0.589330	0.262676	0.293154	0.892760	0.895031	0.492472
19	0.576944	0.250712	0.273956	0.871557	0.846431	0.575333
20	0.541815	0.237455	0.246900	0.850975	0.830439	0.567172
21	0.528128	0.221326	0.241306	0.829977	0.813445	0.546349
22	0.517211	0.266767	0.239213	0.809365	0.783742	0.833755
23	0.506233	0.320327	0.234893	0.788322	0.772519	0.801275
24	0.495808	0.307711	0.323294	0.767590	0.761297	0.789480
25	0.460603	0.300317	0.305873	0.751027	0.742101	0.769920

Table 7.4: *Basic* validity measure for synthetic images using median cluster centres and absolute distance.

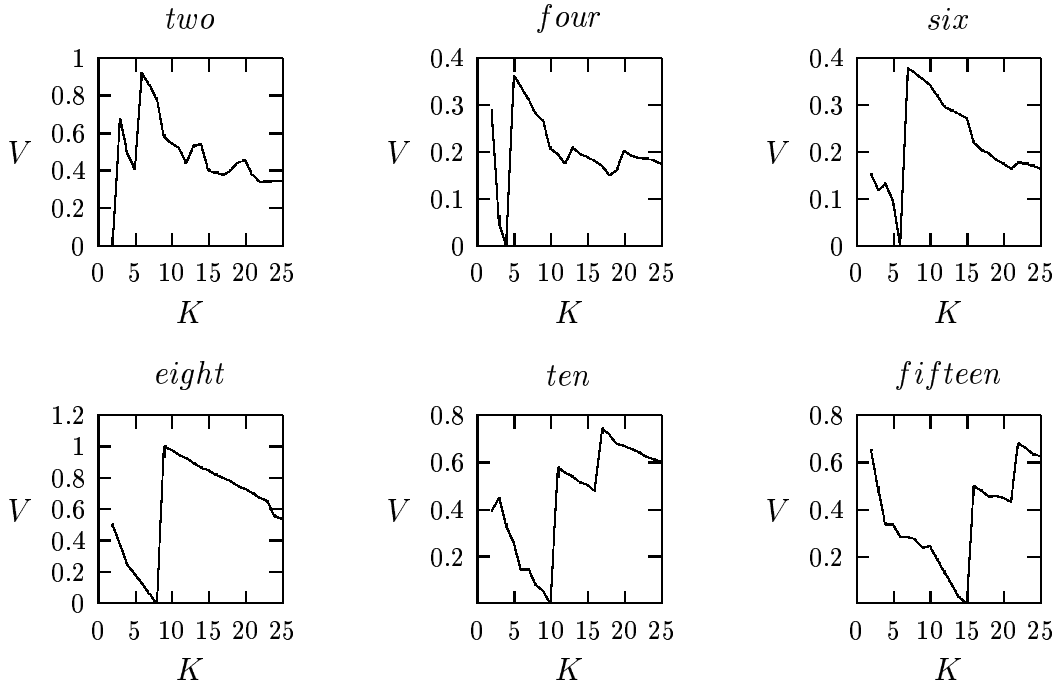


Figure 7.2: Line diagrams of *basic* validity measure for each of the synthetic images using mean cluster centres and Euclidean distance.

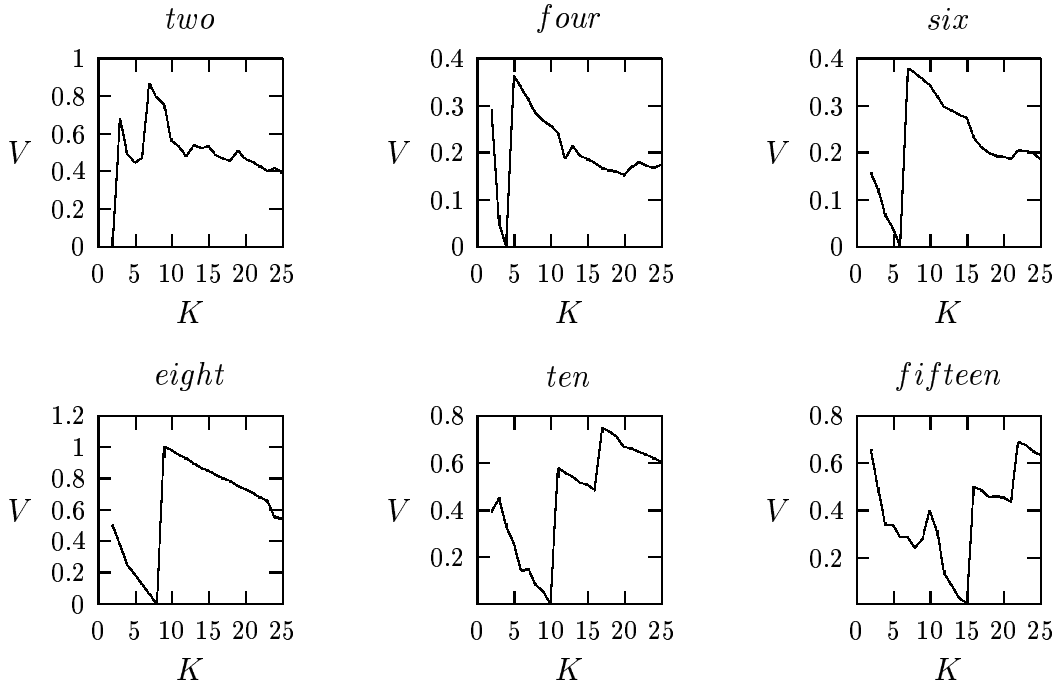


Figure 7.3: Line diagrams of *basic* validity measure for each of the synthetic images using mean cluster centres and absolute distance.

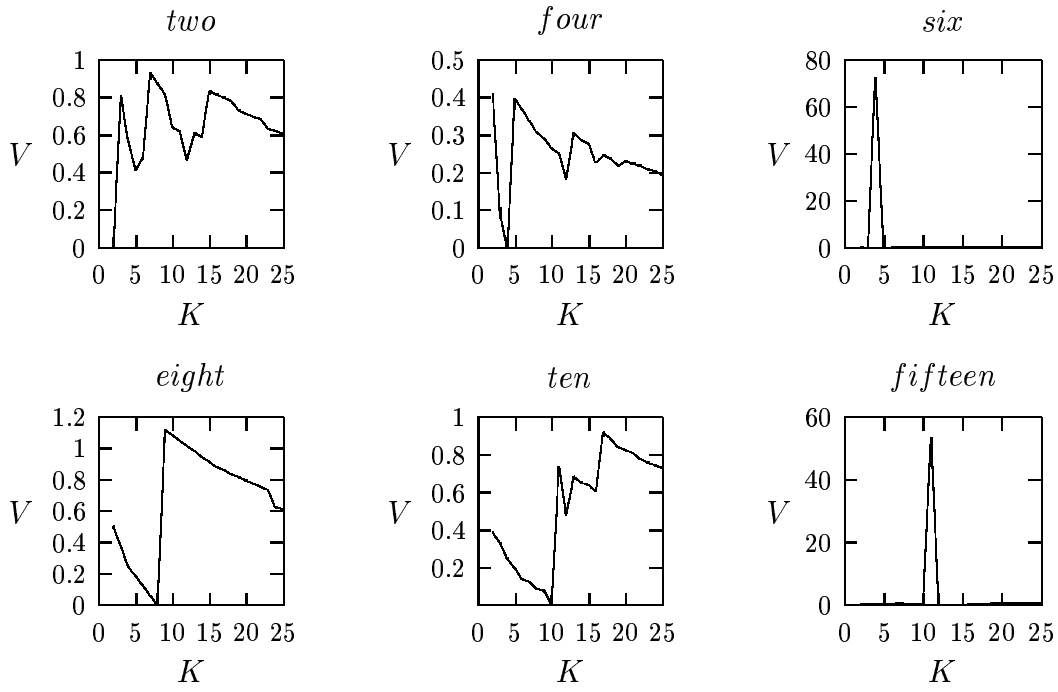


Figure 7.4: Line diagrams of *basic* validity measure for each of the synthetic images using median cluster centres and Euclidean distance.

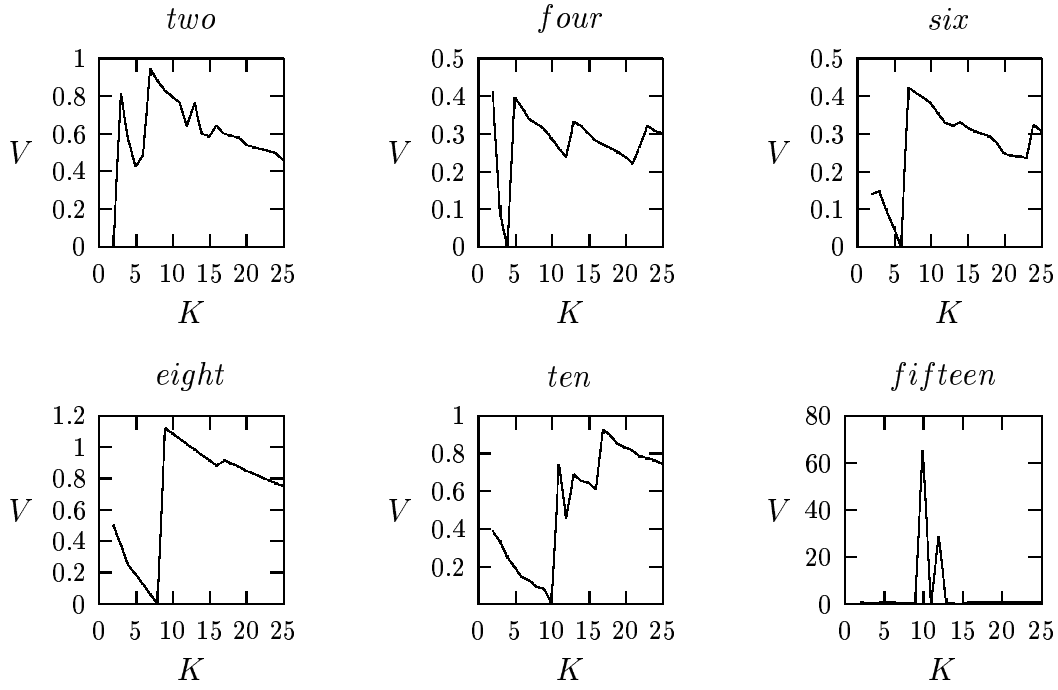


Figure 7.5: Line diagrams of *basic* validity measure for each of the synthetic images using median cluster centres and absolute distance.

Each of these results are based on using the average variance of the three colour components to determine which cluster has the maximum variance and should, therefore, be split. When using only the variance value corresponding to the component having maximum variance, the validity measure still produced its minimum value for the correct number of clusters.

From Tables 7.1-7.4, it is clear that the selected validity criterion works very well for the synthetic images, with the overall minimum values occurring at the correct number of clusters for each of the synthetic images.

7.2.1.2 Natural Images

The next step involved testing this method with real images. The original images are shown in Figure 1.2. All the results in this section are based on the use of the RGB colour space. Once again K_{max} was set to 25 because this is suitably large enough given that from a visual point of view we would not identify more than 25 colour regions in these images.

Table 7.5 shows the *basic* validity measure values obtained for each of the natural images for 2 clusters up to 25 clusters using mean cluster centres and Euclidean distance. Similarly, Table 7.6 shows the *basic* validity measure values obtained when using mean cluster centres and absolute distance, Table 7.7 shows the *basic* validity measure values obtained when using median cluster centres and Euclidean distance and Table 7.8 shows the *basic* validity measure values obtained when using median cluster centres and absolute distance. Figures 7.6-7.9 show the line diagrams corresponding to the values in Tables 7.5-7.8.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.0277	0.1393	0.0189	0.2804	0.1110	0.1671	0.0532	0.2873	0.1586	0.0519	0.1068
3	0.0771	0.1755	0.0606	0.0965	0.0732	0.2788	0.1092	0.2249	0.2465	0.1160	0.0960
4	0.3074	0.1907	0.0354	0.1422	0.1038	0.2417	0.0973	0.2181	0.2400	0.0690	0.1186
5	0.2660	0.2735	0.0453	0.1550	0.1119	0.2207	0.0883	0.3043	0.2512	0.0656	0.1870
6	0.2683	0.2639	0.0312	0.1318	0.1938	0.2570	0.0847	0.3177	0.2234	0.0576	0.1722
7	0.2593	0.2550	0.0200	0.1668	0.1178	0.3081	0.1573	0.3955	0.3726	0.0662	0.2086
8	0.2130	0.2494	0.0177	0.1225	0.2262	0.3284	0.2487	0.3301	0.3167	0.0627	0.2250
9	0.2098	0.3496	0.0331	0.2251	0.4566	0.3869	0.2966	0.3025	0.4446	0.1096	0.2911
10	0.1783	0.3551	0.0286	0.2095	0.3590	0.3818	0.2202	0.3781	0.3764	0.0982	0.2825
11	0.1731	0.3520	0.0232	0.1860	0.3023	0.3767	0.3075	0.3486	0.5022	0.1868	0.2080
12	0.2697	0.3127	0.0229	0.1649	0.2583	0.3717	0.3534	0.4006	0.4654	0.2463	0.1979
13	0.2612	0.3026	0.0231	0.1547	0.2931	0.3671	0.4429	0.5097	0.4443	0.2832	0.1760
14	0.2534	0.2880	0.0556	0.1474	0.2844	0.3349	0.3755	0.4905	0.4409	0.2638	0.2447
15	0.2447	0.4122	0.0514	0.1762	0.2790	0.3330	0.3578	0.5179	0.3875	0.2527	0.2280
16	0.2417	0.4645	0.0476	0.1727	0.2718	0.7249	0.3379	0.4428	0.3567	0.2437	0.2275
17	0.2354	0.4515	0.0445	0.1484	0.2261	0.7162	0.3094	0.4496	0.3491	0.2846	0.2953
18	0.2312	0.4485	0.0427	0.1458	0.2194	0.6315	0.2927	0.4395	0.3327	0.3507	0.2902
19	0.2209	0.4473	0.0366	0.2090	0.2721	0.5930	0.2826	0.4382	0.3048	0.3839	0.2732
20	0.3220	0.4468	0.0323	0.2207	0.2408	0.5801	0.2748	0.4046	0.4398	0.3828	0.2367
21	0.3191	0.4511	0.0312	0.2130	0.2255	0.5717	0.2701	0.5847	0.4028	0.3819	0.2053
22	0.3145	0.4317	0.0302	0.2975	0.2234	0.5677	0.2686	0.5555	0.3397	0.3778	0.1950
23	0.3121	0.4238	0.0298	0.2985	0.2675	0.5655	0.2561	0.5328	0.3675	0.3697	0.2359
24	0.3103	0.4204	0.0345	0.2922	0.2604	0.7231	0.2995	0.5059	0.3453	0.3655	0.2327
25	0.2814	0.3758	0.0400	0.2143	0.2685	0.6441	0.2945	0.4863	0.4220	0.3217	0.2230

Table 7.5: *Basic* validity measure for natural images using mean cluster centres and Euclidean distance.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.0268	0.1432	0.0189	0.2788	0.1110	0.1811	0.0532	0.2931	0.1593	0.0519	0.1082
3	0.0823	0.1687	0.0600	0.1021	0.0758	0.3387	0.1114	0.2076	0.2847	0.1160	0.1063
4	0.3012	0.1840	0.0449	0.1288	0.1255	0.2832	0.0969	0.2184	0.2431	0.0690	0.1457
5	0.3028	0.3714	0.0481	0.2961	0.0899	0.3062	0.1489	0.5422	0.2569	0.0703	0.3012
6	0.2686	0.4505	0.0495	0.5538	0.1931	0.4050	0.1380	0.3469	0.2524	0.0574	0.1683
7	0.2648	0.3893	0.0299	0.5161	0.1888	0.3326	0.1934	0.4385	0.2344	0.0663	0.1882
8	0.2057	0.3833	0.0177	0.5834	0.1897	0.3264	0.1856	0.3767	0.5696	0.0630	0.1459
9	0.2008	0.3580	0.0333	0.2846	0.3749	0.3248	0.1797	0.3342	0.6433	0.1094	0.1998
10	0.1877	0.3428	0.0286	0.2699	0.3611	0.3217	0.2022	0.5997	0.7716	0.1148	0.2146
11	0.2310	0.4773	0.0241	0.1966	0.3277	0.4470	0.2756	0.4563	0.7403	0.1504	0.1899
12	0.2225	0.4481	0.0389	0.1794	0.2809	0.4412	0.2725	0.3969	0.6792	0.1466	0.1909
13	0.2157	0.4169	0.0313	0.1948	0.2393	0.4392	0.3026	0.4215	0.6680	0.3119	0.1648
14	0.2627	0.4071	0.0242	0.1875	0.2192	0.7729	0.3652	0.4180	0.4991	0.2924	0.2146
15	0.2592	0.4099	0.0307	0.1765	0.2132	0.8635	0.3452	0.4392	0.4793	0.2785	0.2096
16	0.2407	0.4032	0.0476	0.1418	0.2096	0.8337	0.3224	0.4348	0.4644	0.3546	0.1871
17	0.2344	0.3915	0.0446	0.1392	0.2220	0.7123	0.3067	0.4328	0.4044	0.3448	0.2063
18	0.2313	0.4739	0.0409	0.1255	0.2158	0.6666	0.3048	0.6562	0.4699	0.4479	0.2455
19	0.2272	0.4420	0.0390	0.1213	0.2192	0.7843	0.2946	0.6144	0.4538	0.4417	0.2317
20	0.2243	0.4392	0.0383	0.1309	0.2174	0.7061	0.3000	0.5773	0.4371	0.4176	0.2485
21	0.2234	0.4381	0.0323	0.1290	0.2488	0.6834	0.2951	0.5506	0.4324	0.3324	0.2431
22	0.2140	0.4019	0.0320	0.1630	0.2209	0.6597	0.2882	0.5171	0.4307	0.3158	0.2263
23	0.2115	0.5299	0.0344	0.1615	0.2960	0.6282	0.4107	0.5141	0.4033	0.3285	0.3555
24	0.3802	0.4464	0.0402	0.2658	0.2776	0.6235	0.4063	0.5696	0.3306	0.3260	0.3430
25	0.3775	0.4312	0.0310	0.2706	0.2741	0.6125	0.4043	0.5840	0.3228	0.3240	0.3270

Table 7.6: *Basic* validity measure for natural images using mean cluster centres and absolute distance.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	2.0261	0.1467	0.0171	0.2144	0.0959	0.1519	0.0537	0.2446	0.1622	0.0530	0.0883
3	0.5157	0.2685	0.0601	0.0803	0.0818	0.3512	0.1858	0.3252	0.1751	0.1280	0.0887
4	0.3200	0.1661	0.0328	0.1967	0.0986	2.1179	0.1320	0.2634	0.3414	2.6136	0.1129
5	0.4270	0.2863	0.0644	0.1702	0.2104	1.9882	0.1074	0.3540	0.2742	0.0847	0.1785
6	0.3559	1.4252	0.0481	0.1394	0.1484	1.4440	0.4912	0.4713	0.2345	0.0564	0.1772
7	0.6352	1.2492	0.0295	0.1763	0.1290	1.3714	0.6432	0.4527	0.2568	0.0977	0.5364
8	0.6021	1.1032	0.0167	0.4044	0.5050	1.1548	0.4810	0.3792	0.4047	0.1403	0.5639
9	0.4944	0.9434	0.0326	0.4152	0.4087	1.1272	0.5613	0.3723	0.3821	0.1688	0.9622
10	0.3890	0.9140	0.0273	0.4731	0.4540	1.8703	0.5015	0.4842	0.6821	0.1557	0.8285
11	0.3772	0.8648	0.0229	0.4166	0.8343	2.7974	0.4466	0.5100	0.7875	0.2992	0.8063
12	0.3623	0.8547	0.0231	0.2906	0.8205	2.5074	0.4316	0.5051	0.6356	0.2712	0.5908
13	0.3497	0.8092	0.0254	0.2462	0.8053	2.4593	0.4233	0.4681	0.6211	0.2604	0.5189
14	0.3387	0.6827	0.0323	0.2778	0.6867	2.4194	0.5761	0.7077	0.6086	0.3069	0.4720
15	0.3314	0.6767	0.0340	0.2637	0.6787	2.3977	0.5075	0.4910	0.5164	0.4553	0.3935
16	0.3267	0.6751	0.0620	0.2251	0.6581	2.3609	0.5060	0.7042	0.4839	0.4403	0.3857
17	0.3248	0.6743	0.0572	0.2210	0.6431	1.9380	0.4835	0.8787	0.6274	0.4384	0.3542
18	0.3108	0.4701	0.0547	0.3911	0.9448	1.7431	0.4819	0.8603	0.5967	0.4374	0.3434
19	0.3077	0.7909	0.0469	0.3457	0.6217	1.6918	0.4489	0.8159	0.5797	0.4107	0.3395
20	0.3053	0.4601	0.0567	0.2925	0.5318	1.5845	0.4414	0.8048	1.0337	0.4029	0.4999
21	0.3036	0.7393	0.1416	0.2663	0.5264	1.5560	0.4389	0.7982	0.5576	1.2026	0.7733
22	0.2974	0.7194	0.4763	0.2557	0.5249	1.5417	0.4116	0.7877	0.5516	1.1780	3.0396
23	0.2925	0.6990	0.5521	0.2515	0.5034	1.5414	0.3958	0.8278	0.5205	1.1668	2.8531
24	0.2881	0.6914	0.6619	0.2371	0.4641	3.0575	0.3906	1.4344	0.4108	1.1324	2.7481
25	0.2845	0.6249	1.3239	0.2381	0.4601	3.0520	0.3887	1.3816	0.4072	1.1174	2.4329

Table 7.7: *Basic* validity measure for natural images using median cluster centres and Euclidean distance.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	2.0261	0.1475	0.0171	0.2913	0.0949	0.1617	0.0537	0.2528	0.1612	0.0530	0.0941
3	0.5157	0.2278	0.0671	0.0812	0.0882	0.3683	0.1296	0.3135	0.2666	0.1280	0.0903
4	0.3505	0.1666	0.0416	0.2853	0.1058	2.2659	0.1479	0.3204	0.3040	2.6136	0.1140
5	0.4652	0.3860	0.0365	0.1809	0.1275	2.1726	0.1194	0.3785	0.3306	0.0847	0.2237
6	0.5120	0.3397	0.0390	0.2371	0.2174	2.8946	0.4966	0.4122	0.6225	0.0565	0.3323
7	0.4484	0.4207	0.0276	0.1820	0.1396	2.6247	0.6500	0.4351	0.4696	0.0978	0.2432
8	0.4041	0.6128	0.0168	0.4520	0.1575	2.5650	0.4994	0.9178	0.4048	0.1407	0.5320
9	0.9596	0.5405	0.0327	0.4430	0.2712	2.5445	0.4484	0.7961	0.3870	0.1197	0.4313
10	0.8514	0.9773	0.0282	0.5169	0.2344	2.5319	0.4358	0.7483	0.4619	0.2027	0.4128
11	0.8280	0.8793	0.0237	0.4569	0.3962	2.5073	0.3276	0.6582	0.4179	0.1928	0.6250
12	0.7446	0.8720	0.0323	0.4421	0.7282	1.8590	0.3568	0.9015	0.5083	0.3767	0.4589
13	0.7234	0.8003	0.0258	0.3018	0.7201	1.7489	0.6193	0.8822	0.5575	0.3590	0.4889
14	0.7182	1.7564	0.0327	0.2618	0.7022	5.0128	0.5889	0.8267	0.6662	0.3460	0.4573
15	0.6944	1.7291	0.0437	0.2707	0.6874	4.9199	0.5766	0.8187	0.6570	0.4905	0.3568
16	0.6709	1.7010	0.0708	0.2738	0.5810	4.7855	0.5514	0.7826	0.6470	0.4305	0.3267
17	0.6463	1.6029	0.0754	0.2572	0.5743	4.0696	0.5416	0.7748	0.6254	0.4132	0.3100
18	0.6450	1.5625	0.0636	0.2271	0.5657	3.7593	0.5399	0.7514	0.5794	0.6139	0.2693
19	0.6347	1.5289	0.0933	0.3981	0.5375	3.6898	0.5381	0.7412	0.5639	0.5967	0.2574
20	0.6179	1.5176	0.1530	0.2904	0.5233	3.6596	0.5358	0.7296	1.1236	0.5838	0.2980
21	0.6109	1.5138	0.1203	0.2773	0.4879	2.8556	0.5324	0.7022	1.0570	0.5772	0.2910
22	0.6033	1.3707	0.1091	0.2667	0.4825	2.7595	0.4972	0.6401	0.9992	0.5714	0.2814
23	0.5556	1.3562	0.1384	0.2633	0.4810	2.6729	0.4820	0.8484	0.9609	1.1416	0.3062
24	0.5462	1.2515	0.2852	0.2588	0.4594	2.6452	0.4752	1.1839	0.8420	1.1316	0.3025
25	0.5382	1.2103	0.3240	0.2552	0.4553	2.6198	0.4673	1.1539	0.8331	0.5644	0.2960

Table 7.8: *Basic* validity measure for natural images using median cluster centres and absolute distance.

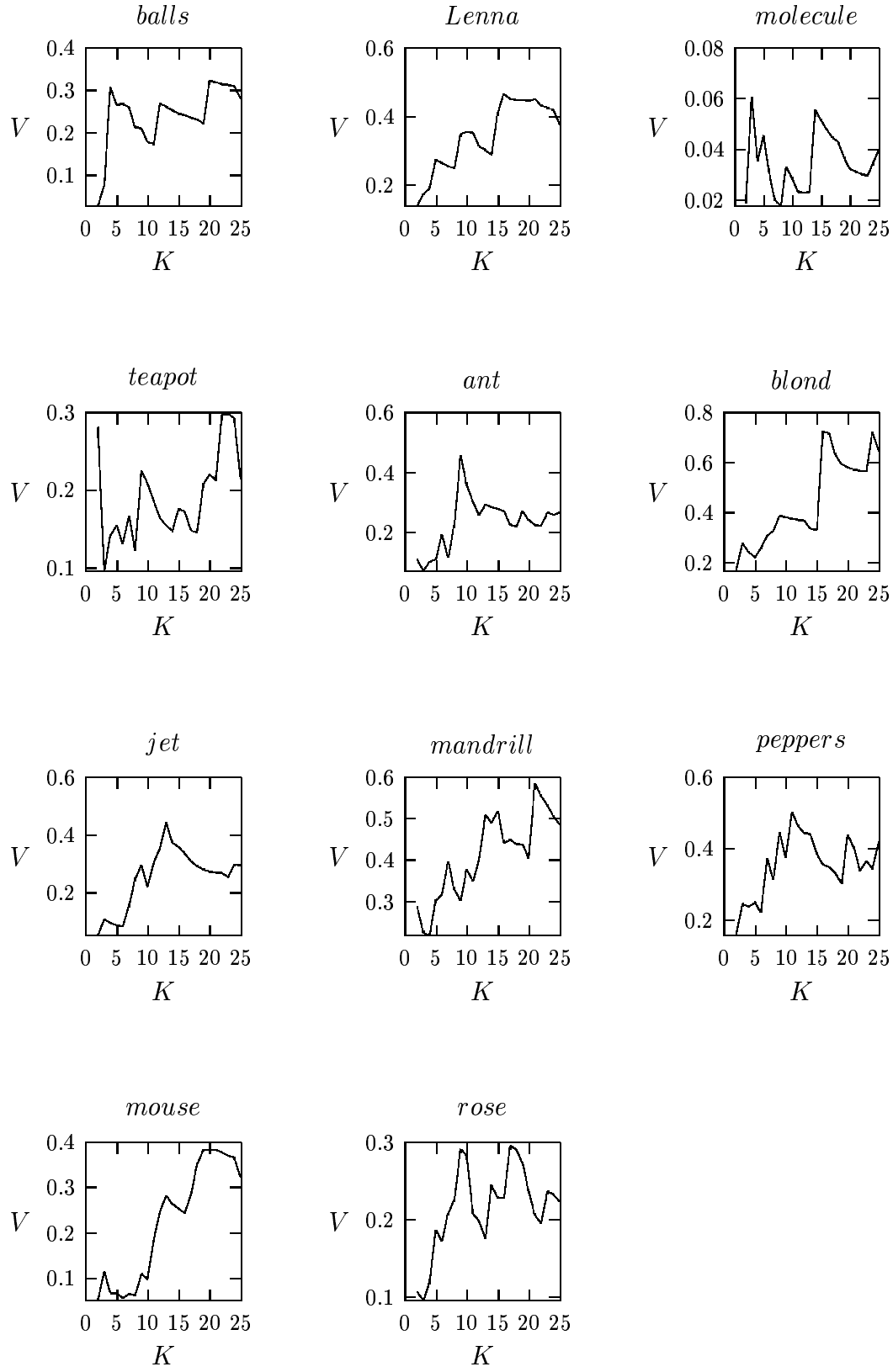


Figure 7.6: Line diagrams of *basic* validity measure for each of the natural images using mean cluster centres and Euclidean distance.

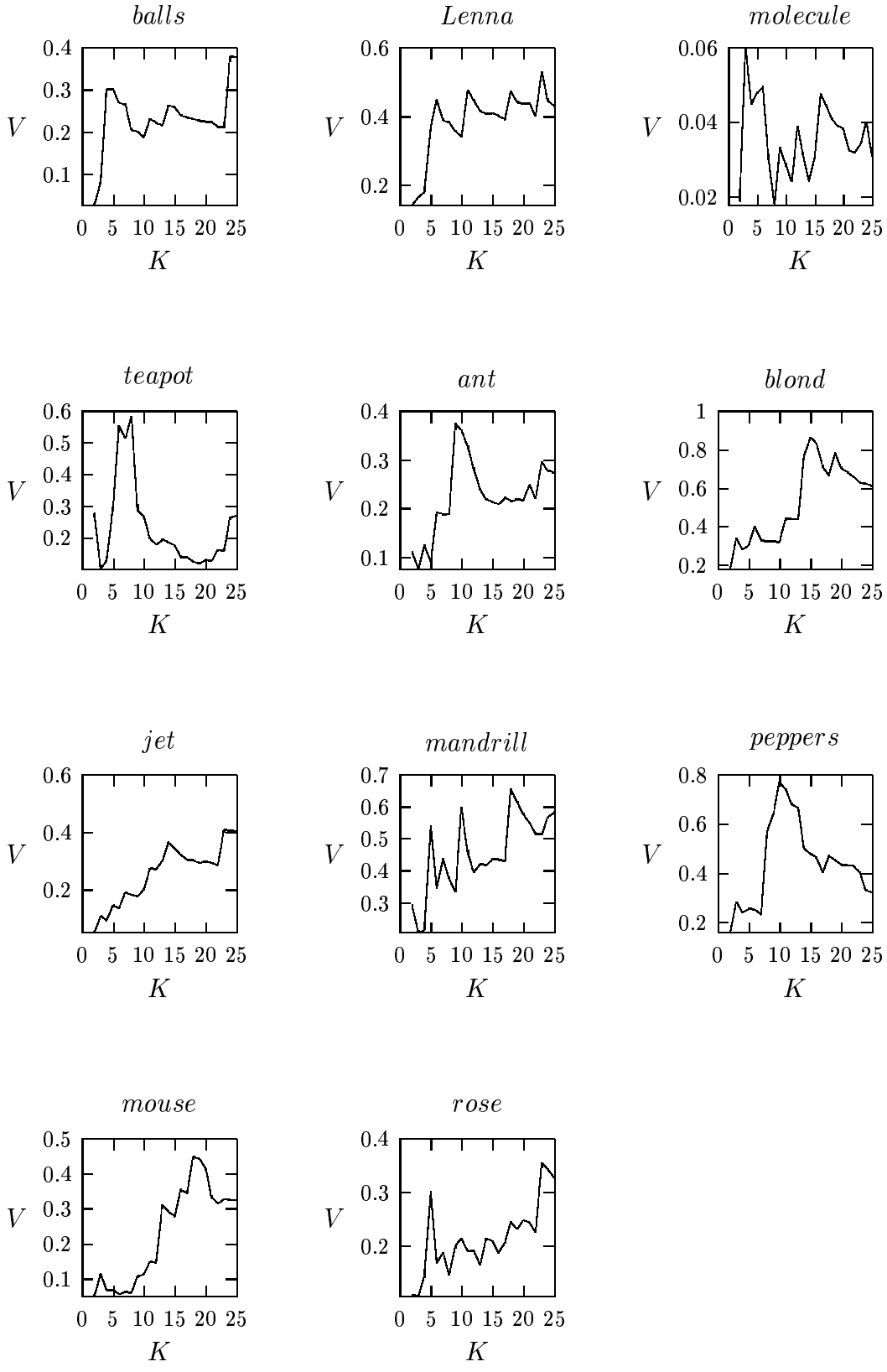


Figure 7.7: Line diagrams of *basic* validity measure for each of the natural images using mean cluster centres and absolute distance.

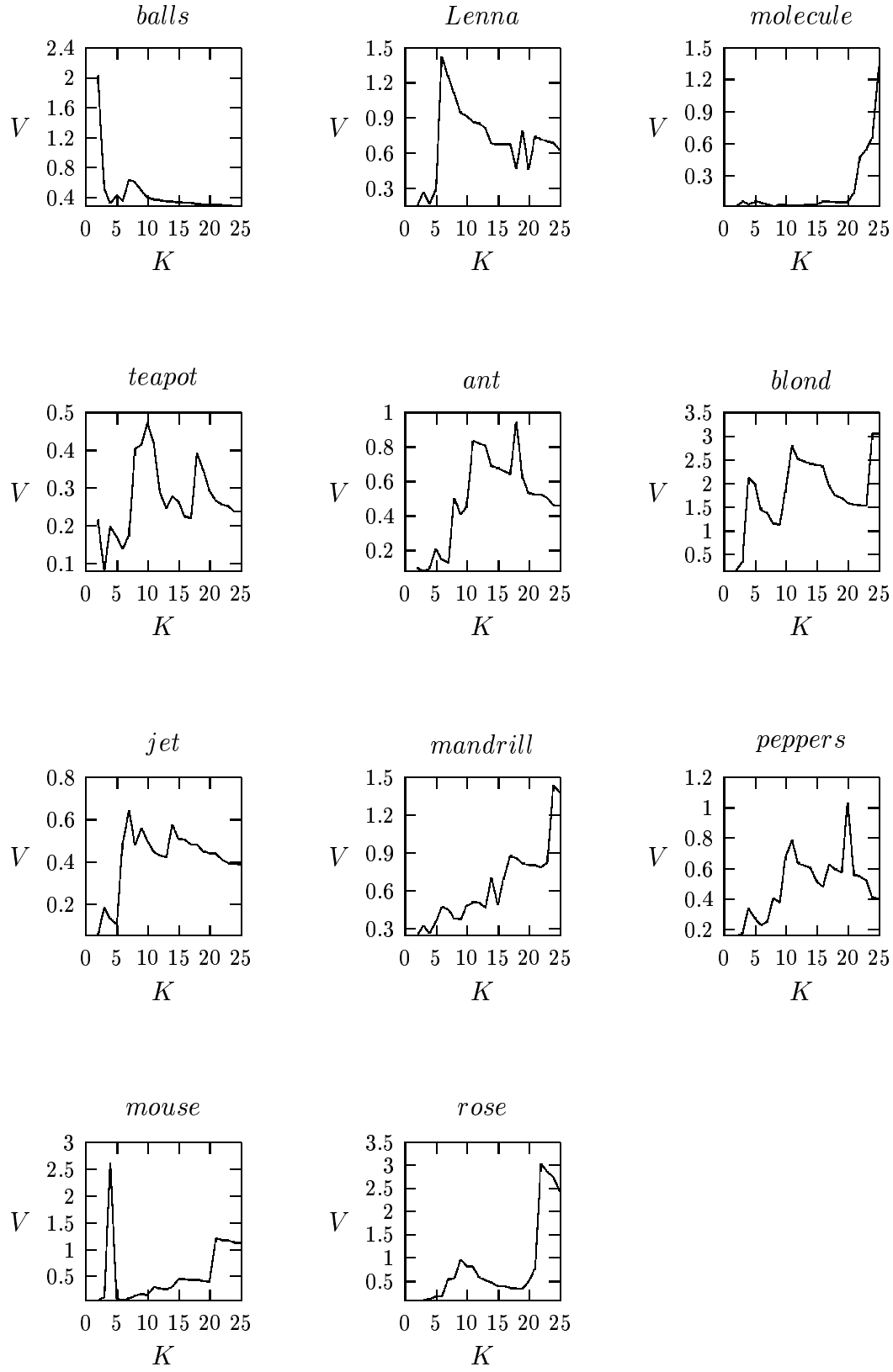


Figure 7.8: Line diagrams of *basic* validity measure for each of the natural images using median cluster centres and Euclidean distance.

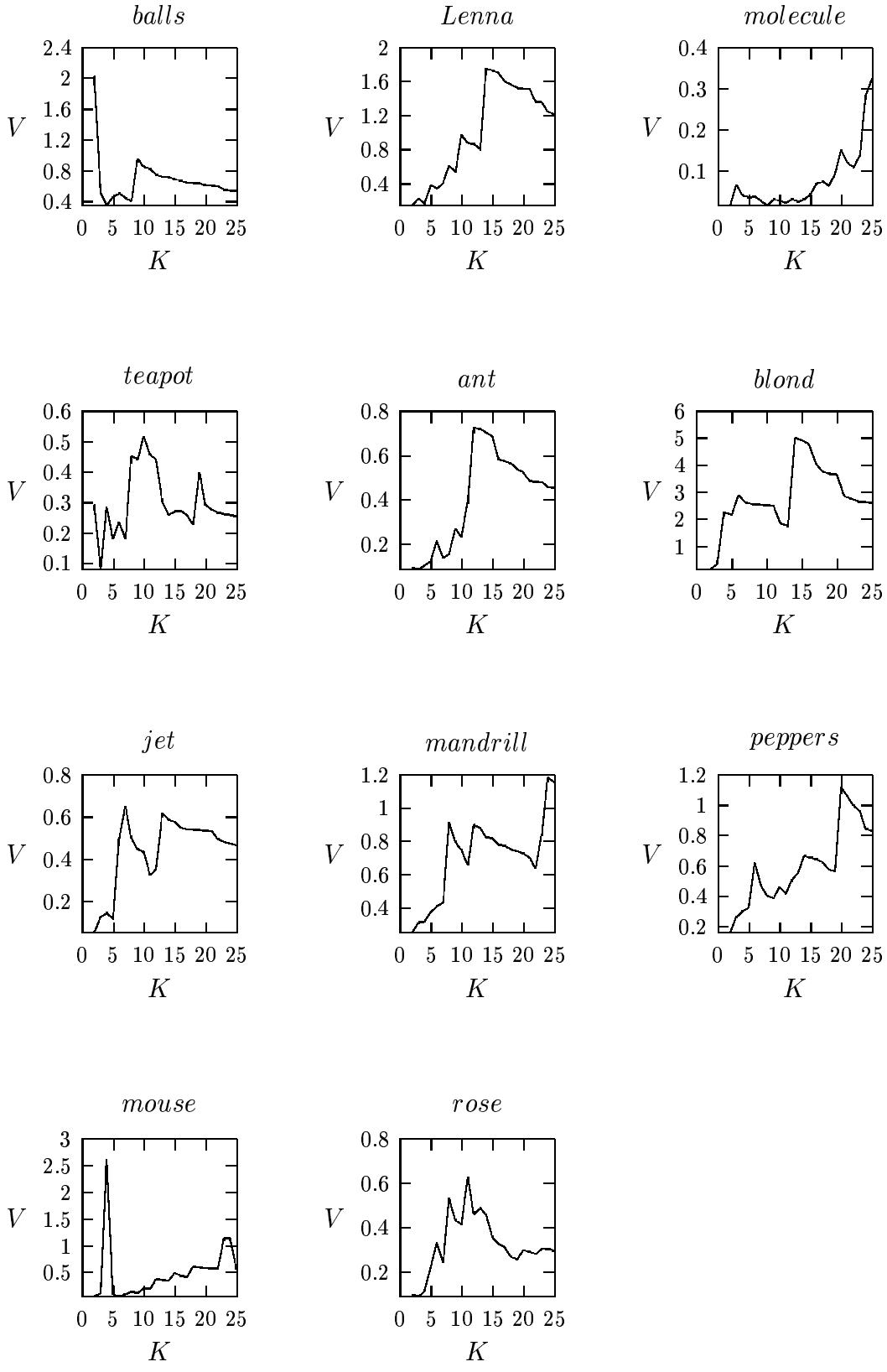


Figure 7.9: Line diagrams of *basic* validity measure for each of the natural images using median cluster centres and absolute distance.

Table 7.9 shows the number of clusters obtained by taking the minimum value of the *basic* validity measure. The first thing we notice from Table 7.9 is that there is a tendency for the minimum value of the validity measure to occur for small numbers of clusters in the range 2-4. This is due to a large inter-cluster distance value occurring when the number of clusters is this low, resulting in the validity measure being very small. The only exceptions occurred for the image *molecule* for which the minimum value for the validity measure was produced for 8 clusters, and for *balls*, when using median cluster centres with Euclidean distance, the resulting number of clusters was rather high at 25.

image	mean & Euc.	mean & abs.	median & Euc.	median & abs.
balls	2	2	25	4
Lenna	2	2	2	2
molecule	8	8	8	8
teapot	3	3	3	3
ant	3	3	3	3
blond	2	2	2	2
jet	2	2	2	2
mandrill	4	3	2	2
peppers	2	2	2	2
mouse	2	2	2	2
rose	3	3	2	3

Table 7.9: Number of clusters using *basic* criterion.

The segmented results using the *basic* validity measure to determine the number of clusters are shown in Figures 7.10-7.13. The most common failing with these results is that part of the objects are classified together with the background. For *balls*, when using mean cluster centres or median cluster centres with absolute distance, the white and yellow balls are completely classified as part of the background. For *Lenna* and *blond*, there is loss of facial features as part of their faces are combined with the background. For the image *teapot*, parts of the checkerboard are combined with the background, while for *ant*, most of the ant is combined with the green toy mouse. For *jet*, the region of the plane which is red is combined with the blue parts of the plane and the mountains. For *mandrill* the blue nose area is combined with

parts of the fur. For *peppers*, the red peppers are combined with the background. For *mouse*, half of the mouse, including the legs, are combined with the background, and for *rose*, the details of the different layers of petals are lost. All of these problems are caused by an inadequate number of clusters to represent the regions in the images. Similar results were obtained for the Davies-Bouldin and Dunn's indexes, for which the numbers of clusters selected are shown in appendix D. It is clear from this that the ratio of the intra-cluster and inter-cluster distance on its own is not good enough to enable the correct level of segmentation to be found.

7.2.2 Modified Validity Criterion ($y = c \times N(2,1) + 1$)

7.2.2.1 Synthetic Images

We first examine what effect incorporating the multiplier function has on selecting the correct number of clusters for the synthetic images. All the results in this section are based on the use of the RGB colour space. Line diagrams of the validity measure values incorporating the multiplier function with c set to 10, 15, 20 and 25 are shown in Figure 7.14. These results are based on using mean cluster centres and Euclidean distance. The actual *modified* validity measure values are shown in appendix E only for 2-7 clusters as there is no change in the values after this point.

We can see from Figure 7.14 that the correct number of clusters can still be obtained for the synthetic images when the multiplier function is incorporated into the validity measure, regardless of the value which is selected for the constant value, c. The Davies-Bouldin and Dunn's indexes were also modified by incorporating the multiplier, however, in the case of Dunn's indexes, the function in equation 7.5 is used as a divisor since the ratio is taken as the inter-cluster distance over the intra-cluster distance. For the synthetic images, the modified form of the Davies-Bouldin

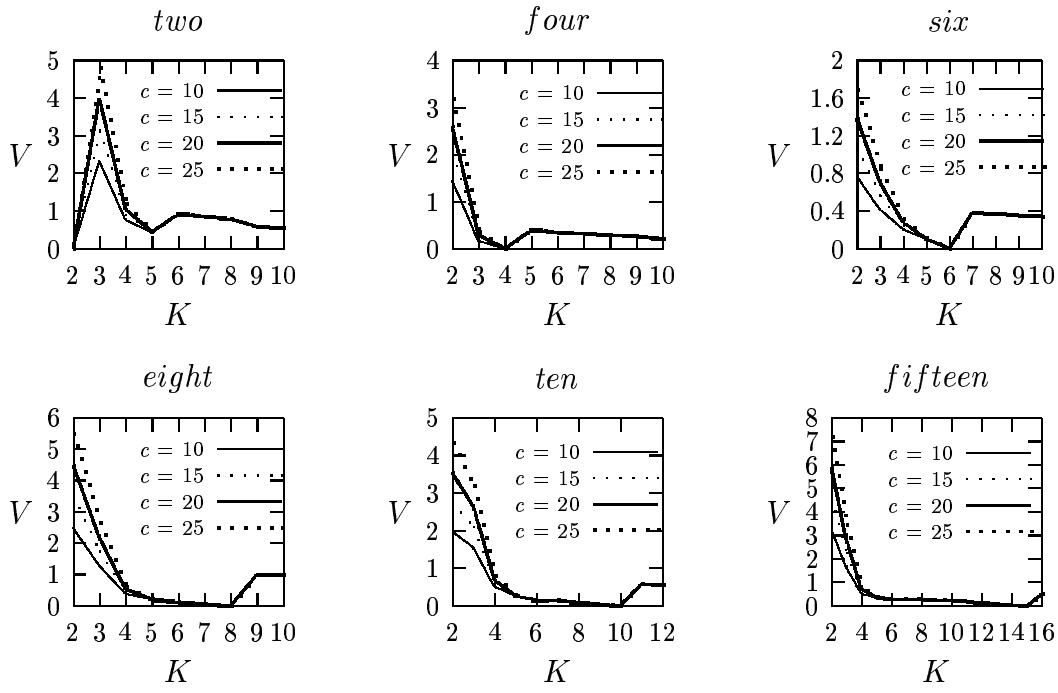


Figure 7.14: Line diagrams of *modified* validity measure for each of the synthetic images using mean cluster centres and Euclidean distance.

index and all derivations of Dunn's index were able to identify the correct number of clusters.

7.2.2.2 Natural Images

We now turn our attention to the natural colour images and see what effect the incorporation of the multiplier function has on the selection of the number of clusters. All the results in this section are based on the use of the RGB colour space. Line diagrams of the *modified* validity measure values incorporating the multiplier function with c set to 10, 15, 20 and 25 are shown in Figures 7.15-7.18, where Figure 7.15 shows the diagrams when mean cluster centres and Euclidean distance are used, and Figures 7.16, 7.17 and 7.18 show the line diagrams when mean cluster centres and absolute distance, median cluster centres and Euclidean distance and median cluster centres and absolute distance were used, respectively. The actual *modified* validity

measure values are shown in appendix E only for 2-7 clusters as there is no change in the values after this point. Figures 7.15-7.18 show that the validity measure is increased dramatically for 2 and 3 clusters compared to the *basic* ratio without the multiplier function. Tables 7.10-7.13 show the number of clusters using the *modified* measure which incorporates the multiplier function with c taken as 10, 15, 20 and 25.

image	$c = 10$	$c = 15$	$c = 20$	$c = 25$
balls	2	11	11	11
Lenna	8	8	8	8
molecule	8	8	8	8
teapot	8	8	8	8
ant	5	7	7	7
blond	5	5	5	5
jet	6	6	6	6
mandrill	9	9	9	9
peppers	6	6	6	6
mouse	6	6	6	6
rose	6	6	6	6

Table 7.10: Number of clusters using *modified* measure for mean cluster centres and Euclidean distance.

image	$c = 10$	$c = 15$	$c = 20$	$c = 25$
balls	2	2	10	10
Lenna	4	4	10	10
molecule	8	8	8	8
teapot	19	19	19	19
ant	5	5	5	5
blond	5	10	10	10
jet	6	6	6	6
mandrill	9	9	9	9
peppers	7	7	7	7
mouse	6	6	6	6
rose	8	8	8	8

Table 7.11: Number of clusters using *modified* measure for mean cluster centres and absolute distance.

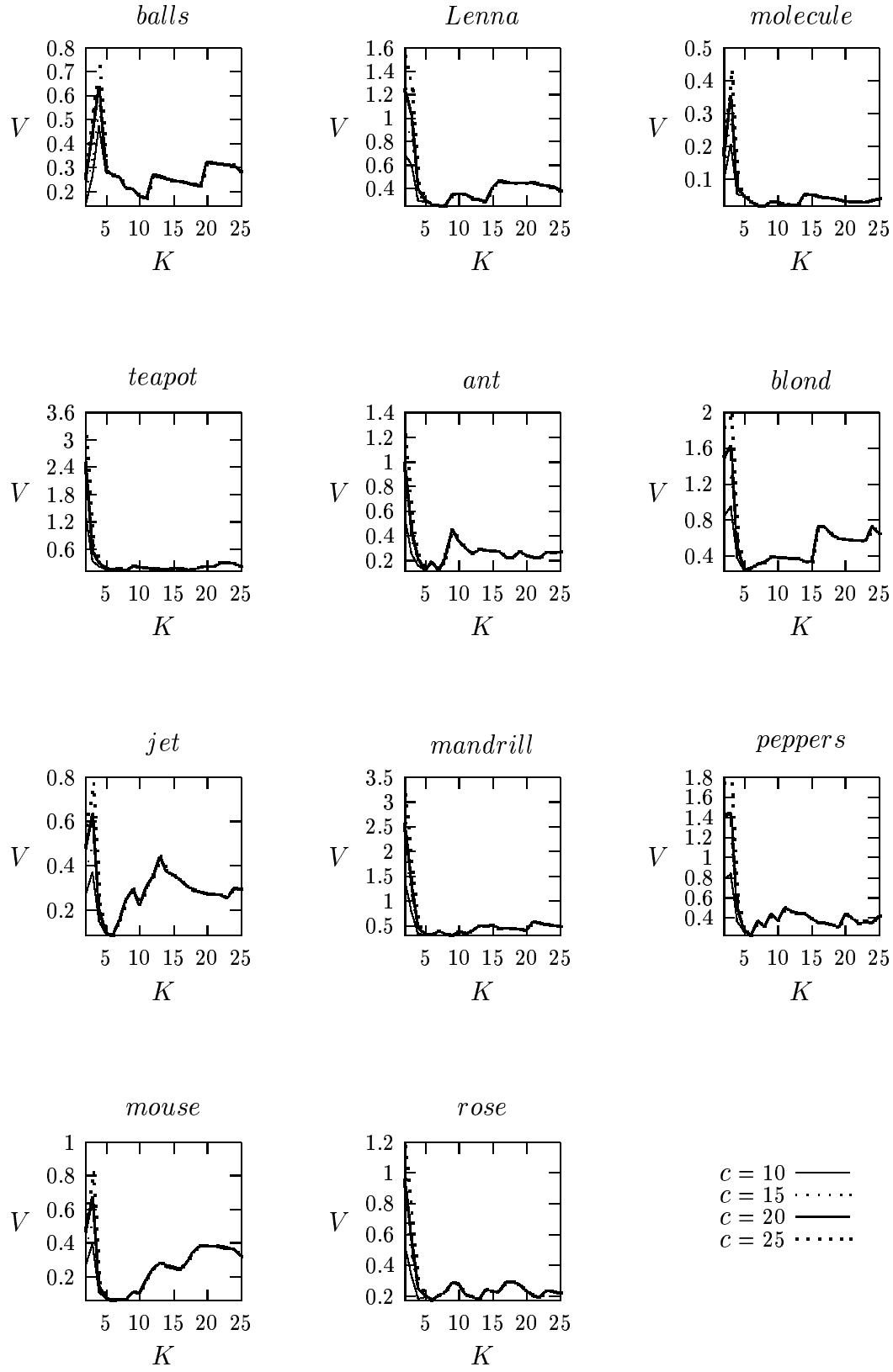


Figure 7.15: Line diagrams of *modified* validity measure for each of the natural images using mean cluster centres and Euclidean distance.

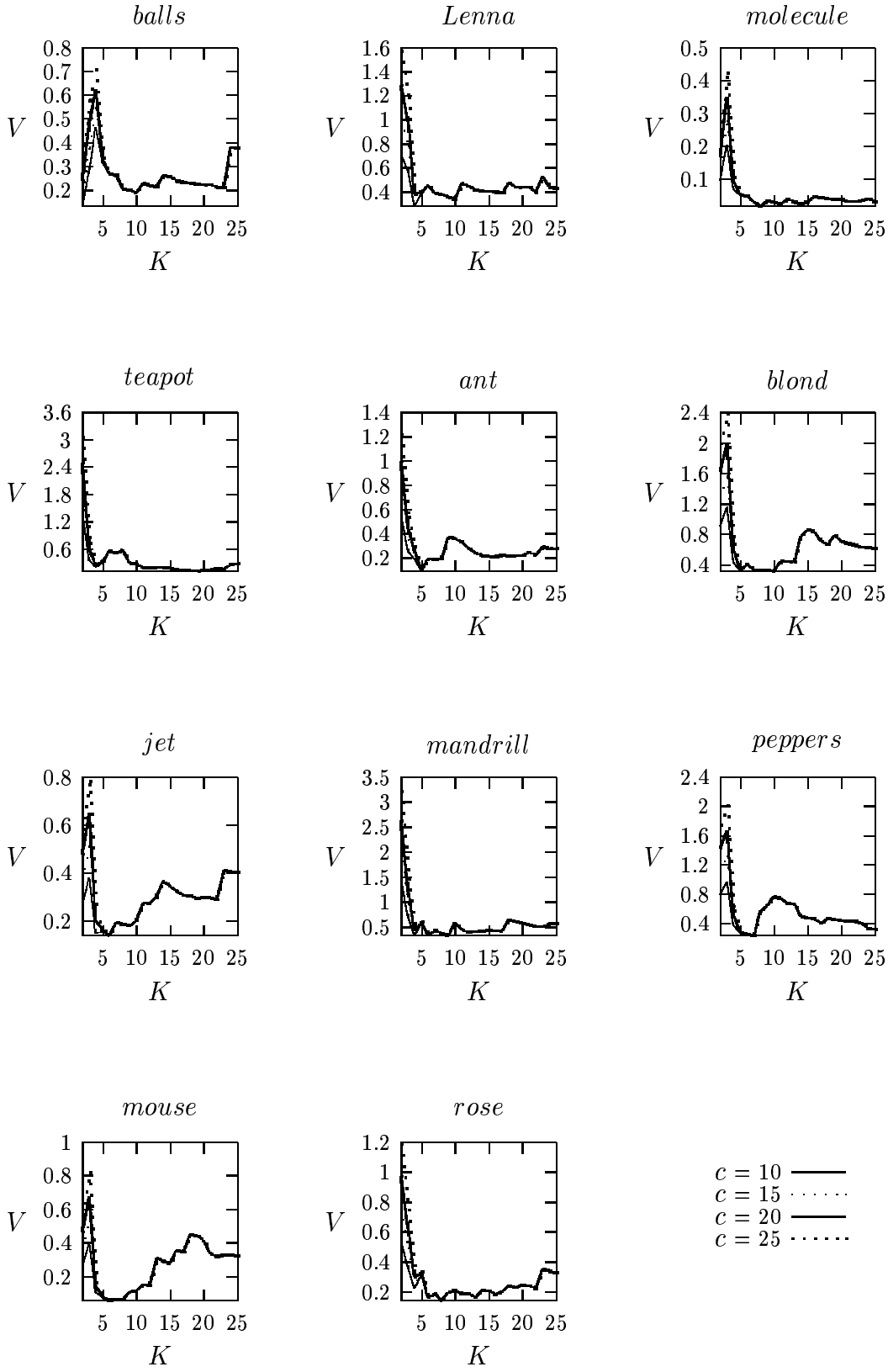


Figure 7.16: Line diagrams of *modified* validity measure for each of the natural images using mean cluster centres and absolute distance.

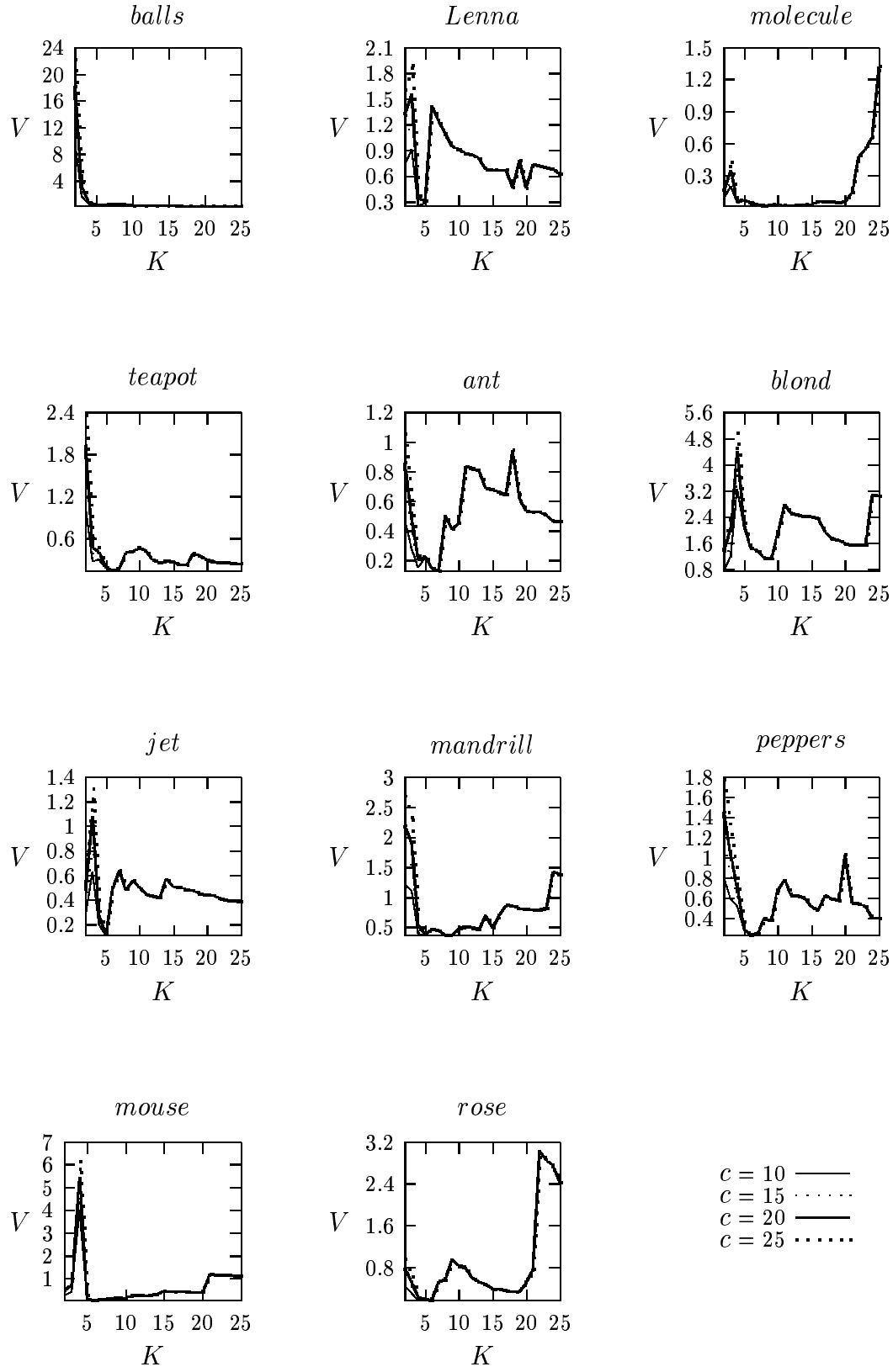


Figure 7.17: Line diagrams of *modified* validity measure for each of the natural images using median cluster centres and Euclidean distance.

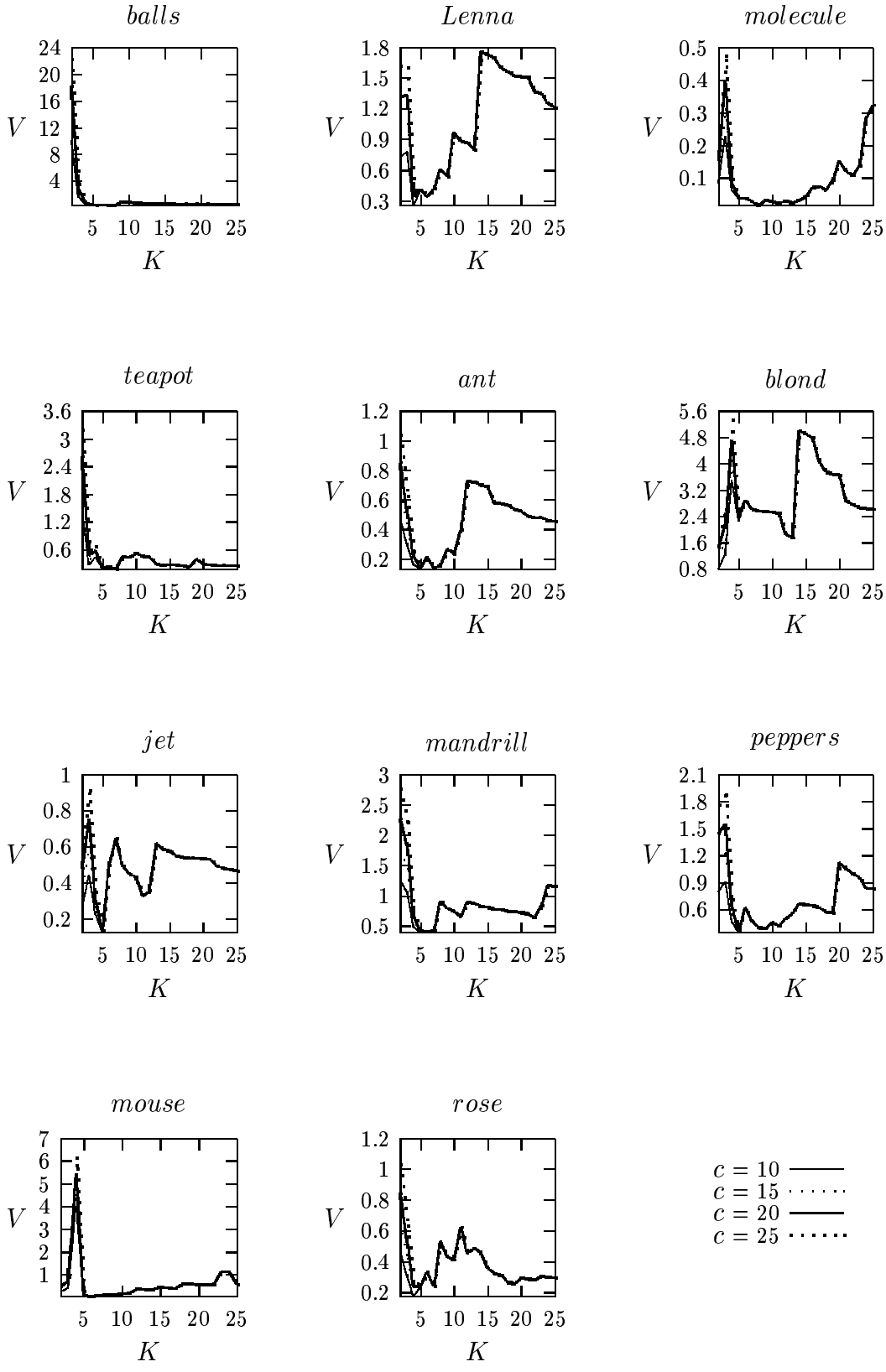


Figure 7.18: Line diagrams of *modified* validity measure for each of the natural images using median cluster centres and absolute distance.

image	$c = 10$	$c = 15$	$c = 20$	$c = 25$
balls	25	25	25	25
Lenna	4	4	5	5
molecule	8	8	8	8
teapot	6	6	6	6
ant	7	7	7	7
blond	2	2	9	9
jet	5	5	5	5
mandrill	5	9	9	9
peppers	6	6	6	6
mouse	6	6	6	6
rose	4	6	6	6

Table 7.12: Number of clusters using *modified* measure for median cluster centres and Euclidean distance.

image	$c = 10$	$c = 15$	$c = 20$	$c = 25$
balls	8	8	8	8
Lenna	4	4	6	6
molecule	8	8	8	8
teapot	7	7	7	7
ant	5	5	5	7
blond	2	2	2	13
jet	5	5	5	5
mandrill	5	5	5	6
peppers	5	5	5	5
mouse	6	6	6	6
rose	4	4	4	7

Table 7.13: Number of clusters using *modified* measure for median cluster centres and absolute distance.

For each combination of cluster centre and distance type, the number of clusters is generally the same when c is set to 20 and above. The only exceptions occurred for the images *ant*, *blond*, *mandrill* and *rose* when median cluster centres were used with absolute distance. In many cases, there is no change in the number of clusters even when 15 is used as the value for c. Given that in most cases increasing the value of c beyond 15 or 20 has no effect on the number of clusters selected, we can say that the proposed validity measure is fairly robust with respect to the constant used in the multiplier function. It only needs to be large enough to allow all of the images to skip over the results with small numbers of clusters.

Figures 7.19-7.22 show the segmented results for each of the natural images with the optimal number of clusters produced when incorporating the multiplier function

with c set to 25. The use of the *modified* measure overcomes the problems encountered with the *basic* validity criterion, as this allows the clustering results with a small number of clusters to be skipped. Each of these segmentation results show a vast improvement over the results shown in Figures 7.10-7.13. The results in Figures 7.19-7.22 mostly have an adequate number of clusters to represent the regions in the images. For the image *balls*, using mean cluster centres with either distance type produced an acceptable segmentation, while using median cluster centres with Euclidean distance resulted in the number of clusters being too high. The result when using median cluster centres and absolute distance produced too few clusters as the yellow ball was classified as belonging to the background. For the image *Lenna*, the number of clusters produced when using median cluster centres was too small, while using mean cluster centres resulted in good segmentations. For *teapot*, all of the results are good except when using mean cluster centres and absolute distance, which resulted in too many clusters. All of the results for *ant* and *blond* are good. For the image *mandrill*, the number of clusters produced when using median cluster centres and absolute distance was too small, while all the other combinations of cluster centre and distance type resulted in good segmentations. All of the results for *jet*, *peppers*, *mouse* and *rose* are good. Overall, the mean cluster centres and Euclidean distance approach performed the best giving good results for all of the natural images.

The Davies-Bouldin and Dunn's indexes also have a tendency to select a small number of clusters. Therefore, the multiplier/divisor function can be applied to them as well. The number of clusters produced by the Davies-Bouldin and Dunn's indexes for both the *basic* ratio and by using the multiplier/divisor function with c set to 25 are shown in appendix D. These results are based on using mean cluster centres and Euclidean distance. By using the *modified* form of the Davies-Bouldin index, a

reasonable number of clusters was selected for only five of the images, with the result for *balls* producing too few clusters to adequately represent the regions in the image, and for *Lenna*, *teapot*, *blond*, *mandrill* and *peppers*, the number of clusters selected was too high. For *peppers*, even the use of the *basic* form of this index resulted in 22 clusters which is extremely large for this image. For the eighteen versions of Dunn's index, the three involving the interset distance function δ_1 performed the worst, as the number of clusters was much too big, even by the *basic* measure, so the *modified* rule was not applied to these indexes. The best results by using the *modified* measure were achieved for the measures D_{21}, D_{31}, D_{41} and D_{51} which produced an adequate number of clusters for most images, except for *molecule* for which the number of clusters was too small, for *teapot*, for which the number of clusters was too big for D_{21} and D_{41}, and for *peppers*, for which the number of clusters was too big for D_{31}, D_{41} and D_{51}. The measures D_{22}, D_{23}, D_{42}, D_{43} and D_{61} also did reasonably well in finding a suitable number of clusters for many of the natural images.

The proposed validity measure works more consistently for the natural images than either the Davies-Bouldin index or Dunn's indexes. This is true especially when the results are based on the use of mean cluster centres and Euclidean distance as in this case the number of clusters produced by the proposed measure resulted in good segmentations for each of the natural images as opposed to the other measures for which a good segmentation could not be found for each of the images. The use of the multiplier function allows results containing a small number of clusters to be abandoned in favour of results containing more clusters which are also more visually appealing.

7.2.3 Results Using Context

All of the results obtained when using context are based only on the mean cluster centres and Euclidean distance approach and the use of the RGB colour space. Table 7.14 shows the number of clusters obtained when using only the context values, for each of the four context types, by both the *basic* criterion and the *modified* criterion using the multiplier function with c set to 25.

image	*mean*		*median*		*weighted*		*regression*	
	basic	*mod.*	*basic*	*mod.*	*basic*	*mod.*	*basic*	*mod.*
balls	2	7	2	7	2	7	2	6
Lenna	2	7	2	9	2	7	2	7
molecule	2	8	8	8	2	8	2	8
teapot	3	9	3	9	3	9	3	9
ant	3	5	3	5	5	5	3	5
blond	2	5	2	5	2	5	2	5
jet	2	5	2	5	2	5	2	5
mandrill	3	11	3	6	3	11	3	6
peppers	2	6	2	6	2	6	2	6
mouse	6	6	2	6	6	6	2	6
rose	3	6	3	6	3	6	3	6

Table 7.14: Number of clusters using context values only.

Once again, we can see that the *basic* criterion performs poorly giving an inadequate number of clusters. Figures 7.23-7.26 show the segmented results for the *modified* ratio with c set to 25 when using only the context values for each of the *mean*, *median*, *weighted* and *regression* contexts, respectively. The *mean* and *weighted* context results are very similar for each of the images. For *balls*, *Lenna*, *ant*, *blond*, *peppers*, *mouse* and *rose*, the *median* context results were also similar to the *mean* and *weighted* context results, and for *teapot*, *ant*, *blond*, *peppers* and *mouse*, the *regression* context results were also very similar to the *mean* and *weighted* context results. The *median* and *regression* context results are also similar for the images *molecule*, *ant*, *blond*, *mandrill*, *peppers* and *mouse*. For *ant*, *blond*, *peppers* and *mouse*, all of the four results are very similar. Visually, the results for *balls* do not contain enough clusters, especially the *regression* context result. The *median* context result for *Lenna* was

the best as the hat had less pixels allocated to the background region compared to the other results. The best result for *molecule* occurred when using the *median* and *regression* contexts. For *teapot* all of the results were good, especially the *median* context result which was able to distinguish the individual squares in the shadow region of the board. For *jet*, the best result occurred when using the *regression* context as the F-16 symbol is clearer, while for *mandrill*, all of the results are good, especially the *mean* and *weighted* context results. The best result for *rose* was produced by the *regression* context. The *median* and *regression* contexts performed the best producing good results for eight and seven out of the eleven images, respectively. The *mean* and *weighted* contexts only produced good results for six out of the eleven images.

Table 7.15 shows the number of clusters obtained when using both the original and the context values, for each of the four context types, by both the *basic* criterion and the *modified* criterion using the multiplier function with c set to 25.

image	*mean*		*median*		*weighted*		*regression*	
	basic	*mod.*	*basic*	*mod.*	*basic*	*mod.*	*basic*	*mod.*
balls	2	8	2	7	2	8	2	7
Lenna	2	7	2	7	2	7	2	7
molecule	2	7	9	9	2	7	2	8
teapot	3	6	3	8	3	6	3	6
ant	3	5	3	5	3	5	3	5
blond	2	5	2	5	2	5	2	5
jet	2	5	2	5	2	5	2	5
mandrill	3	6	3	6	3	6	3	6
peppers	2	6	2	6	2	6	2	6
mouse	2	6	2	6	2	6	2	6
rose	3	6	3	6	3	6	3	6

Table 7.15: Number of clusters using original and context values.

Once again, we can see that the *basic* criterion performs poorly giving an inadequate number of clusters. Figures 7.27-7.30 show the segmented results for the *modified* ratio with c set to 25 when using both the original and the context values for each of the *mean*, *median*, *weighted* and *regression* contexts, respectively. In many cases each of the four context types gave very similar results, which occurred for the

images *ant*, *blond*, *mandrill*, *peppers*, *mouse* and *rose*. For each of these images an identical number of clusters was also found, and the visual results were quite good. For *balls*, the *mean* and *weighted* contexts produced similar results and the *median* and *regression* contexts produced similar results. However, these results, especially by the *median* and *regression* contexts, do not contain enough clusters. For *Lenna*, the best result was obtained by the *median* context which resulted in the feathers being treated as a separate cluster. The remaining three context types produced very similar results. For *molecule*, the *mean* and *weighted* contexts produced very similar results although the green part contains a black stripe in the segmentation result. The *median* and *regression* context results are quite good and are both very similar. For *teapot*, the results by the *mean*, *weighted* and *regression* contexts are very similar, however, the overall best result was achieved when using the *median* context which resulted in more clusters and also resulted in the reflection region being correctly identified. For *jet* the best results were obtained by the *mean*, *median* and *weighted* contexts as the writing on the plane is clearer. Here the *regression* context result was bad as the blue in the plane was grouped together with the blue in the mountains. Each of the context types performed very similarly, with perhaps a slight bias towards the *median* context. In comparison to the results by using only the original colour values, the context results tend to have more smooth looking segments.

7.2.4 Results Using Different Colour Spaces

Each of the results obtained when using different colour spaces are based only on the mean cluster centres and Euclidean distance approach. The results using the RGB colour space with mean cluster centres and Euclidean distance were presented in sections 7.2.1.2 and 7.2.2.2. Table 7.16 shows the number of clusters obtained for

the XYZ, KL and HSV colour spaces by both the *basic* ratio and by the *modified* ratio using the multiplier function with c set to 25.

image	XYZ		KL		HSV	
	basic	*mod.*	*basic*	*mod.*	*basic*	*mod.*
balls	2	8	2	12	2	5
Lenna	2	19	2	11	2	5
molecule	2	7	8	8	3	5
teapot	3	25	2	7	14	14
ant	3	6	3	6	8	8
blond	2	5	2	5	5	6
jet	2	5	2	5	3	8
mandrill	2	6	3	6	18	18
peppers	4	12	2	6	2	13
mouse	2	7	2	6	3	9
rose	2	6	3	6	8	8

Table 7.16: Number of clusters using different colour spaces.

Once again, we can see that the *basic* ratio performs poorly giving an inadequate number of clusters. The main exceptions to this occurred when using the KL colour space, where for *molecule* the minimum for the *basic* ratio occurred for 8 clusters, which was an identical result as that produced by the RGB colour space. Also, the image *teapot* resulted in 14 clusters, the image *mandrill* resulted in 18 clusters and the images *ant* and *rose* resulted in 8 clusters for the *basic* ratio using the HSV colour space. Figures 7.31-7.33 show the segmented results for the *modified* ratio for each of the XYZ, KL and HSV colour spaces, respectively, with c set to 25. In general, the HSV colour space performed poorly as either the number of clusters were too small, especially in the case of *balls* and *molecule*, or the results look very noisy, especially in the case of *Lenna*, *ant*, *peppers* and *mouse*. Overall, the results by the KL colour space were the best, which produced very similar results to the RGB colour space. While the results by both the XYZ and KL colour spaces produced the same number of clusters for the *modified* method for the images *jet* and *mandrill*, the segmented results are much superior by the KL colour space, as the red part of the *jet* is included as a separate cluster, while the segmented results for *mandrill* using the KL colour

space did a better job of separating the red nose and eye regions from the fur, while the XYZ result combined these with parts of the fur.

7.3 Summary

By incorporating the validity measure based on the intra-cluster and inter-cluster distance measures, the number of clusters present in an image can be determined automatically. The validity measure proposed here works well with synthetic images, producing a minimum value for the expected number of clusters. Although there is a tendency to select smaller numbers of clusters for natural images, this is due to the inter-cluster distance being much greater and greatly affecting the validity measure. We overcome this by incorporating a multiplier function based on the Gaussian function. Use of this *modified* measure allows the results containing a small number of clusters to be penalized in favour of results containing more clusters which are also more visually appealing. The constant multiplier value c was required to be at least 15 to achieve a suitable level of segmentation for most images, although a value of 20 or 25 ensures that all of the images obtain a suitable level of segmentation. The *modified* measure still allows the optimal number of clusters to be selected for the synthetic images. The Davies-Bouldin index and Dunn's indexes could not detect the correct number of clusters for all the natural images, even in their *modified* form, however, the proposed validity measure performed more consistently for all of the natural images, producing good segmentation results.

The context results indicate that the *mean* and *weighted* contexts produce very similar results, and the *median* and *regression* contexts also produce very similar results. These results tend to have more smooth looking segments than the results which only use the original colour values. In the case of the image *balls*, the context

results do not contain enough clusters to adequately represent the different coloured balls.

The colour space results indicate that the RGB and KL colour spaces produce very similar results, and both of these results are the best, especially when mean cluster centres and Euclidean distance are used.

The proposed validity measure is not restricted to colour image data. It can be easily used on any type of data on which clustering is applied. Although, different values for the constant c used in equation 7.5 may be required in these instances, especially in cases where the dimensionality of the data is different, for example, multispectral images containing more than 3 bands, and grey scale images, which have only one dimension.

Chapter 8

Conclusions

8.1 Summary of Contributions

The main focus of this thesis has been on developing methods which automatically determine the number of clusters in a colour image. The intra-cluster and inter-cluster distances are measures which can be used to determine the quality of a clustering, where the intra-cluster distance measures cluster compactness and the inter-cluster distance measures cluster separation. An investigation of these measures revealed a statistically significant correlation between these measures. This finding led to the approaches of developing criteria based on either one or the other of these two measures to determine if a good clustering has been found. All together three main approaches were investigated.

The first approach involved the development of methods based on having a good cluster separation, which is measured by the inter-cluster distance. Two such methods were developed, both of which incorporated a merging process, where the first method allowed only two clusters to be merged during an iteration, while the second method allowed more than two clusters to be merged during an iteration. In both of these

methods thresholds were needed to determine when the optimal clustering has been found. For the first method the threshold was calculated as some *weight* multiplied by the average inter-cluster distance, while for the second method the threshold was calculated as the maximum of the minimum inter-cluster distances for each cluster from the first iteration. An alternative was proposed for the second method to allow outlier colours to be treated as separate clusters. In this case the threshold was calculated as the mean of the minimum inter-cluster distance plus four standard deviations.

The second approach involved developing methods which aimed to achieve good cluster compactness, which is measured by the intra-cluster distance. A splitting method was developed which aimed to achieve cluster compactness, and three stopping rules were proposed to determine when the optimal clustering was obtained. The first stopping rule was based on some *weight* multiplied by the overall variance of the image, the second stopping rule was based on both the normalized intra-cluster distance and the percentage decrease in intra-cluster distance being below some given thresholds, and the third stopping rule was based on only the percentage decrease in intra-cluster distance being below a threshold.

The third approach involved the development of a validity measure based on both the intra-cluster and the inter-cluster distances. The proposed validity measure was based on the ratio between the average intra-cluster distance and the minimum inter-cluster distance. The minimum inter-cluster distance was used because the average could be greatly affected by one cluster being very far away from the rest and optimizing this will not necessarily lead to adequate cluster separation between all clusters. A further modification to the measure was made by incorporating a multiplier function which is a function of the number of clusters. This function was based on the Gaus-

sian function with a mean of 2 and a standard deviation of 1, which was multiplied by some constant and 1 was added to it.

To aid in the analysis of the results, a visual analysis survey was conducted, where a number of results were shown with varying numbers of clusters and participants were asked to select the best segmentation result. The results of this survey gave a benchmark for the number of clusters which could be used to determine the accuracy of the proposed methods. A second survey was also conducted to determine the applicability of the general methods to the natural colour images, where the two merging methods and the splitting method were compared. Results by a version of the splitting program without the *K-means* step was also compared. The responses for many of the images were fairly evenly spread between the merging and splitting methods, however, for the images *Lenna*, *blond* and *mouse* there was a definite favouritism for the splitting approach, while the image *molecule* generally obtained better results for the merging methods, with its best result occurring for merging method 2, when using the *maximum* method to determine the threshold.

Other clustering considerations were also taken into account such as the distance measure used to determine similarity, where the Euclidean and absolute distances were investigated, and the method of cluster centre representation, where mean and median were considered. Also, the context of the image was taken into account by using a function of the eight neighbouring pixel values. Four methods of doing this were proposed including *mean*, *median*, *weighted* and *regression*. Two methods of using the context information were proposed, the first being to perform the clustering on the three context components, and the second being to use the three original colour components together with the three context components, leading to a six dimensional clustering problem. Also, the clustering was done in a number of different colour

spaces, including RGB, XYZ, KL and HSV, to determine which colour space produced the best segmentation results.

8.1.1 Summary of Results by Merging Methods

For merging method 1 the final result relied on a parameter value, *weight*. Based on the results for all of the images, the optimal value for the *weight* is somewhere in the range 0.3-0.5. For the second merging method the *maximum* threshold produced good results for some images, especially for *molecule* which obtained its best result by this method. However, when calculating the threshold by this method, some images were reduced to very small numbers of clusters. Application of the *mean* method to calculate the threshold successfully allowed the images which failed by the *maximum* method to obtain a much improved segmentation result.

8.1.2 Summary of Results by Splitting Methods

For the splitting method, the use of stopping rule 1 involved a parameter value, *weight*. Based on the results for all the images, the optimal value for the *weight* is somewhere in the range 0.05-0.2. The image *blond* required a much larger *weight* than this, however, this is mostly due to the fact that this image as a whole has a very small variability, so an acceptable threshold was calculated by using a larger *weight* value.

The second stopping rule involved two parameters, $perc_1$ and $perc_2$, for the normalized intra-cluster distance and percentage decrease in intra-cluster distance, respectively. Based on the results for all of the images, a value of 0.4-0.5 for $perc_1$ and 0.05 for $perc_2$ gave good segmentations in most cases. However, there were some exceptions. For *blond* and *jet* larger values of $perc_1$ were required. The image *mouse*

obtained a better segmentation when a value of 0.1 was used for $perc_2$, mainly due to this image requiring a small number of clusters for which a percentage decrease in intra-cluster distance below 0.05 does not generally occur.

The third stopping rule involved one parameter, *perc*, as well as also considering up to three consecutive percentage decrease values, which were required to be below *perc*. In most cases a good segmentation was found when only one percentage decrease value was required to be below the threshold, *perc*, and in this case the optimal value for *perc* was in the range 0.05-0.07. The image *balls* required a smaller value while for *mouse* a value of at least 0.08 was required. Given that good segmentations could be found when only considering one percentage decrease value, there is really no need to require more than one value to be below the threshold, however, the image *blond* failed to obtain a good segmentation using only one percentage decrease value as the number of clusters produced was too low. This image often required three consecutive percentage decrease values to be below *perc* in order to obtain a good segmentation. For all of the stopping rules proposed for the splitting method, the image *blond* always gave contrary results by requiring different values for the parameters, which was not really surprising given that it is very different from the other images in terms of its overall variability being much smaller.

8.1.3 Summary of Results using Proposed Validity Measure

Two versions of the validity measure were implemented, the *basic* criterion, where only the ratio between the average intra-cluster and minimum inter-cluster distances were considered, and the *modified* criterion which incorporated the multiplier function. For the *basic* measure there was a tendency to select a small number of clusters, such as 2 or 3. This problem also occurred for the Davies-Bouldin index and Dunn's indexes.

The *modified* criterion greatly improved the results. Generally setting the constant in the multiplier function to 15 achieved suitable results, and there was no change in the number of clusters obtained even if the constant was set to 25, so we used 25 to ensure that we definitely obtain a suitable result. The *modified* measure was correctly able to identify the number of clusters in the synthetic images as well as each of the natural images, for which the results matched well with the survey which was conducted to establish the optimal range for the number of clusters. When the Davies-Bouldin and Dunn's indexes were also modified by employing the Gaussian function, the results by these indexes did not achieve a good level of segmentation for all of the images. On the whole, therefore, the proposed measure outperformed these.

8.1.4 Findings on other Clustering Considerations

Of the four combinations of cluster centre and distance type investigated, visual results indicated a preference for the mean cluster centres and Euclidean distance combination for each of the proposed methods, and while there were exceptions to this, in a majority of the cases, the mean cluster centres and Euclidean distance approach gave the best segmentation results.

The results using the context tended to have more connected regions in the segmented images, and for the image *blond*, the hair could correctly be identified as a separate segment from the face and hands. This is a clear advantage when compared to the results not incorporating the context information. In terms of performance, the various contexts are fairly even when using only the context components, however, when using the original colour components together with the context components, the *regression* context outperformed the other context types for many of the natural

images.

Four colour spaces were used to perform the clustering, including RGB, XYZ, KL and HSV. Interestingly, even though the Karhunen-Loève transformation was applied to the original RGB values to produce the KL colour space, and which decorrelated the components, the results obtained for both of these colour spaces were very similar from a visual point of view. The results by the XYZ colour space were in general poor, and good results could only be obtained for the images *ant*, *blond* and *mouse*. The HSV results were also quite poor, with many results appearing noisy, although for the image *balls*, the use of this colour space did a good job of identifying the different colours of the balls. In general, there seems to be no benefit in selecting a colour space other than RGB in which to perform the clustering.

8.2 Further Research

While merging method 2 is totally automatic, with no user input required, there are two methods used to determine the threshold value, and for a given image we would only want to use one of them, depending on whether there are any outlier colours in the image. Further research is required to establish a criterion to determine automatically whether there exists any outlier colours so we can then use the appropriate threshold.

A further issue arises for merging method 2 based on the problem of some images producing too many clusters. A possible further avenue of research could involve incorporating the number of pixels having a certain colour into the calculation of the inter-cluster distance. A colour that occurs more often is more likely to belong to a separate cluster, so the inter-cluster distance calculation can be weighted in some way in order to make colours which occur more frequently shift further away from

the rest of the colours in terms of the inter-cluster distance measure. This may then result in the threshold which is selected to be larger, thus reducing the final number of clusters produced.

One disadvantage of the use of the context, especially when using it together with the original values, is that there was a tendency for two clusters to be produced in place of one, where the two clusters have the same centres for the original colour components but differ only in the context centres. Another problem which can occur when using only the context components is that the results look slightly blurred, especially in the case of the *mean*, *median* and *weighted* contexts, and the pixels along the boundary of an object can be assigned to a separate cluster. This effect can be observed for the image *molecule* for which the results show an outline for each portion of the molecule in a different colour. This is mainly due to the calculation of the context along the boundaries of the objects, which can result in the context values being vastly different from the original values since some of the neighbouring pixels have a very different colour value as it comes from a different segment of the image. A possible improvement, which would involve further research, is to only consider neighbouring pixels which are more likely to belong to the same region as the current pixel, so some criterion would be needed to help decide this.

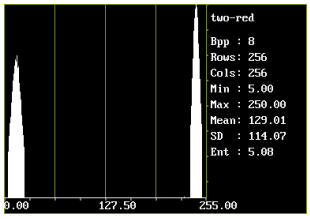
In this thesis results were obtained for two different distance measures and four different colour spaces. Further research could involve the investigation of other distance measures, such as Mahalanobis distance which takes into account the covariances in the data, and also the application of the methods to other colour spaces, of which there are many. Also, incorporating the context when using a colour space other than RGB would be useful to determine whether the results can be improved for the colour spaces which did not perform very well.

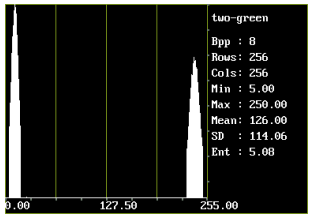
The application of the proposed validity measure, as well as other similar measures, can be problematic when comparing results from two different methods. Consequently, a better validity criterion for comparing results by two different methods is still required. The MML criterion may be useful for this purpose. It is a worry that the clustering algorithm *snob*, which is based on the MML principle, did not produce satisfactory segmentation results. However, since the MML criterion provides a means of carrying out model selection, and if the different clusterings or image segmentations are considered to be the models, a MML criterion should be able to objectively determine which is the best. A further research project could involve the investigation of exactly how to make use of the MML criterion to develop an objective evaluation criterion for image segmentation results.

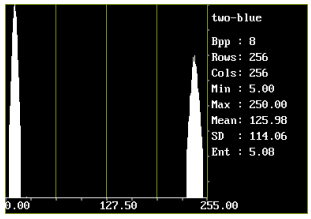
Many clustering methods have been presented in this thesis, and all of them were applied only on colour image data. Further areas of research would involve applying the proposed methods to other types of data to see if the choice of parameter values are affected, and to also apply the methods to data of different dimensionality, including one, which would apply to grey scale images.

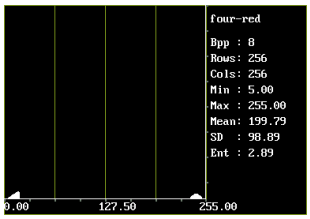
Appendix A

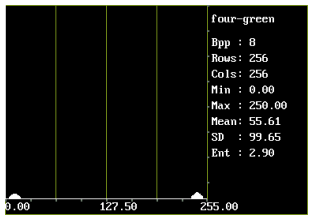
Image Histograms

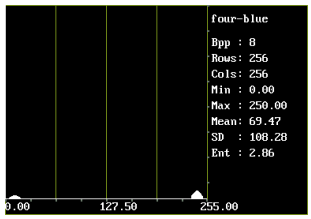


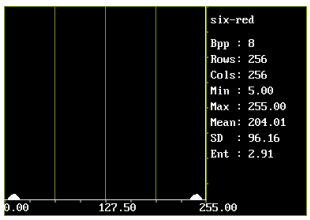


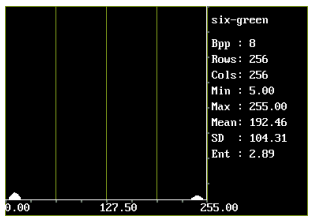


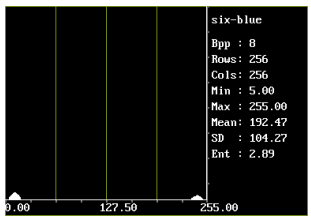


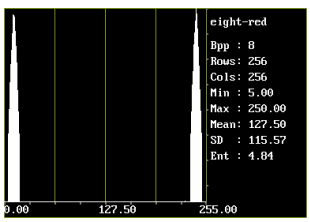


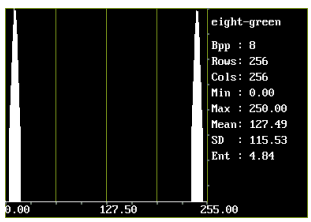


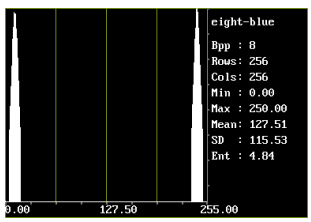


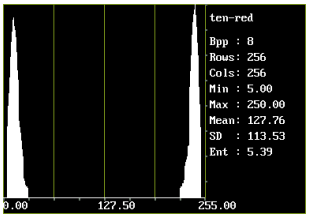


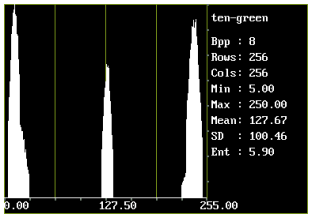


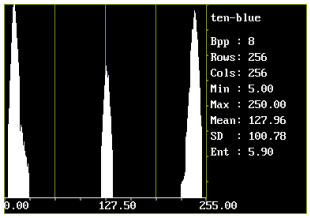


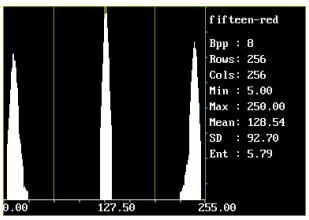


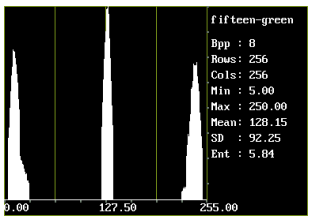


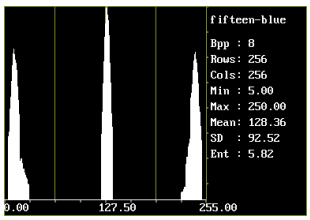

ten-red
Bpp : 8
Rows: 256
Cols: 256
Min : 5.00
Max : 250.00
Mean: 127.76
SD : 113.53
Ent : 5.39
0.00
127.50
255.00

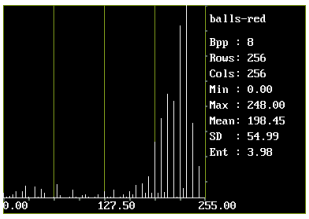

ten-green
Bpp : 8
Rows: 256
Cols: 256
Min : 5.00
Max : 250.00
Mean: 127.67
SD : 100.46
Ent : 5.90
0.00
127.50
255.00

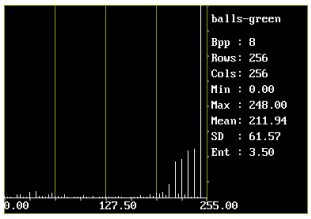

ten-blue
Bpp : 8
Rows: 256
Cols: 256
Min : 5.00
Max : 250.00
Mean: 127.96
SD : 100.78
Ent : 5.90
0.00
127.50
255.00

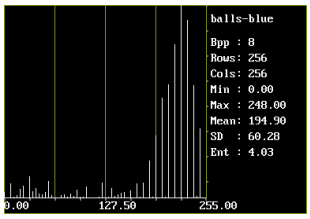

fifteen-red
Bpp : 8
Rows: 256
Cols: 256
Min : 5.00
Max : 250.00
Mean: 128.54
SD : 92.70
Ent : 5.79
0.00
127.50
255.00

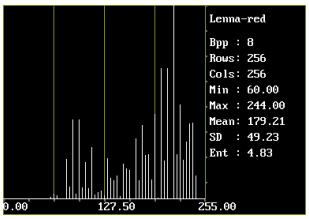

fifteen-green
Bpp : 8
Rows: 256
Cols: 256
Min : 5.00
Max : 250.00
Mean: 128.15
SD : 92.25
Ent : 5.84
0.00
127.50
255.00

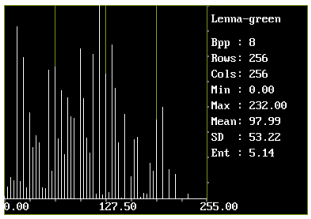

fifteen-blue
Bpp : 8
Rows: 256
Cols: 256
Min : 5.00
Max : 250.00
Mean: 128.36
SD : 92.52
Ent : 5.82
0.00
127.50
255.00

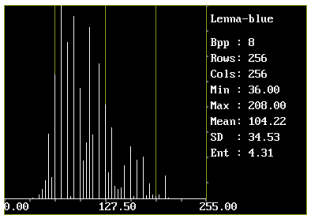

balls-red
Bpp : 8
Rows: 256
Cols: 256
Min : 0.00
Max : 248.00
Mean: 198.45
SD : 54.99
Ent : 3.98
0.00
127.50
255.00

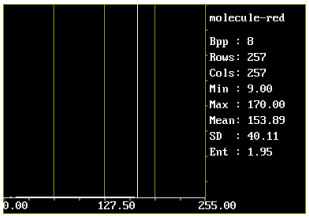

balls-green
Bpp : 8
Rows: 256
Cols: 256
Min : 0.00
Max : 248.00
Mean: 211.94
SD : 61.57
Ent : 3.50
0.00
127.50
255.00

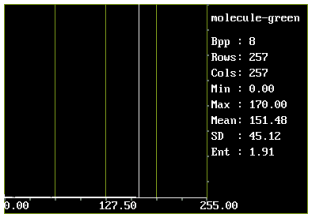

balls-blue
Bpp : 8
Rows: 256
Cols: 256
Min : 0.00
Max : 248.00
Mean: 194.90
SD : 60.28
Ent : 4.03
0.00
127.50
255.00

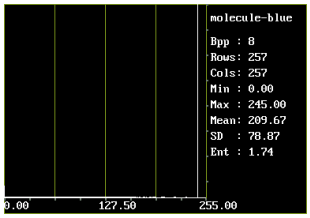

Lenna-red
Bpp : 8
Rows: 256
Cols: 256
Min : 60.00
Max : 244.00
Mean: 179.21
SD : 49.23
Ent : 4.83
0.00
127.50
255.00

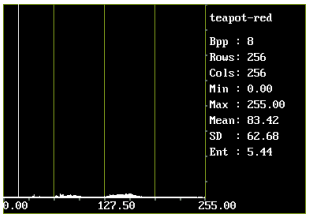

Lenna-green
Bpp : 8
Rows: 256
Cols: 256
Min : 0.00
Max : 232.00
Mean: 97.99
SD : 53.22
Ent : 5.14
0.00
127.50
255.00

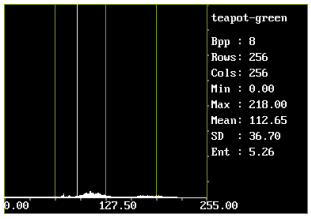

Lenna-blue
Bpp : 8
Rows: 256
Cols: 256
Min : 36.00
Max : 208.00
Mean: 104.22
SD : 34.53
Ent : 4.31
0.00
127.50
255.00

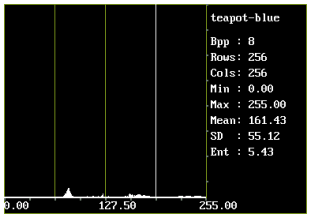

molecule-red
Bpp : 8
Rows: 257
Cols: 257
Min : 9.00
Max : 170.00
Mean: 153.89
SD : 40.11
Ent : 1.95
0.00
127.50
255.00

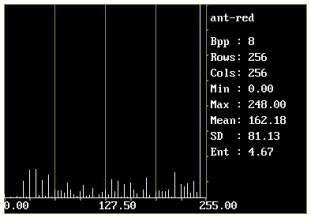

molecule-green
Bpp : 8
Rows: 257
Cols: 257
Min : 0.00
Max : 170.00
Mean: 151.48
SD : 45.12
Ent : 1.91
0.00
127.50
255.00

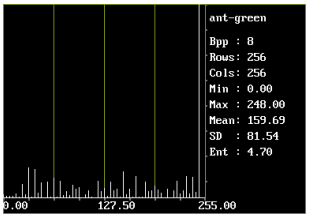

molecule-blue
Bpp : 8
Rows: 257
Cols: 257
Min : 0.00
Max : 245.00
Mean: 209.67
SD : 78.87
Ent : 1.74
0.00
127.50
255.00

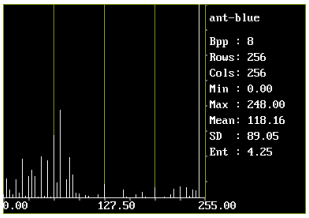

teapot-red
Bpp : 8
Rows: 256
Cols: 256
Min : 0.00
Max : 255.00
Mean: 83.42
SD : 62.68
Ent : 5.44
0.00
127.50
255.00

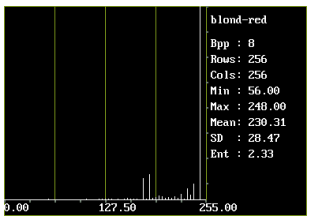

teapot-green
Bpp : 8
Rows: 256
Cols: 256
Min : 0.00
Max : 218.00
Mean: 112.65
SD : 36.70
Ent : 5.26
0.00
127.50
255.00

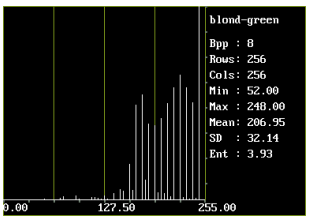

teapot-blue
Bpp : 8
Rows: 256
Cols: 256
Min : 0.00
Max : 255.00
Mean: 161.43
SD : 55.12
Ent : 5.43
0.00
127.50
255.00

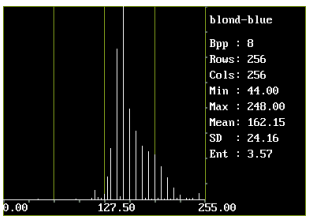

ant-red
Bpp : 8
Rows: 256
Cols: 256
Min : 0.00
Max : 248.00
Mean: 162.18
SD : 81.13
Ent : 4.67
0.00
127.50
255.00

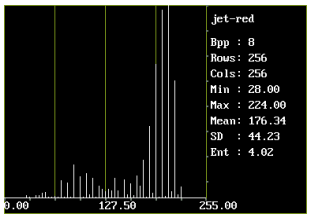

ant-green
Bpp : 8
Rows: 256
Cols: 256
Min : 0.00
Max : 248.00
Mean: 159.69
SD : 81.54
Ent : 4.70
0.00
127.50
255.00

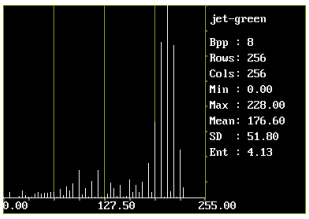

ant-blue
Bpp : 8
Rows: 256
Cols: 256
Min : 0.00
Max : 248.00
Mean: 118.16
SD : 89.05
Ent : 4.25
0.00
127.50
255.00

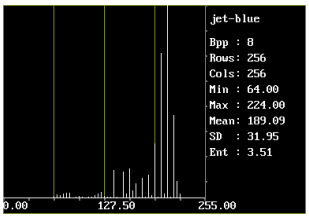

blond-red
Bpp : 8
Rows: 256
Cols: 256
Min : 56.00
Max : 248.00
Mean: 230.31
SD : 28.47
Ent : 2.33
0.00
127.50
255.00

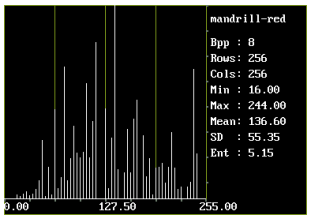

blond-green
Bpp : 8
Rows: 256
Cols: 256
Min : 52.00
Max : 248.00
Mean: 206.95
SD : 32.14
Ent : 3.93
0.00
127.50
255.00

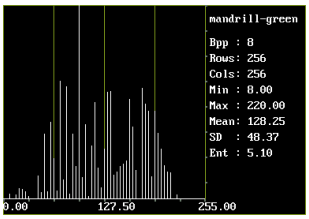

blond-blue
Bpp : 8
Rows: 256
Cols: 256
Min : 44.00
Max : 248.00
Mean: 162.15
SD : 24.16
Ent : 3.57
0.00
127.50
255.00

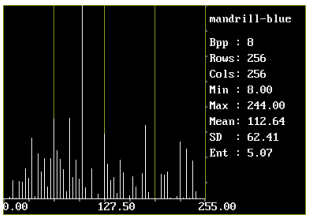

jet-red
Bpp : 8
Rows: 256
Cols: 256
Min : 28.00
Max : 224.00
Mean: 176.34
SD : 44.23
Ent : 4.02
0.00
127.50
255.00

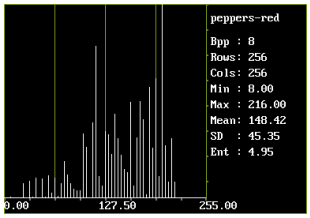

jet-green
Bpp : 8
Rows: 256
Cols: 256
Min : 0.00
Max : 228.00
Mean: 176.60
SD : 51.80
Ent : 4.13
0.00
127.50
255.00

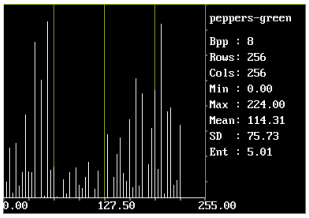

jet-blue
Bpp : 8
Rows: 256
Cols: 256
Min : 64.00
Max : 224.00
Mean: 189.09
SD : 31.95
Ent : 3.51
0.00
127.50
255.00

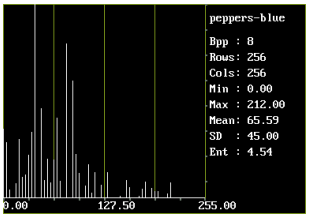

mandrill-red
Bpp : 8
Rows: 256
Cols: 256
Min : 16.00
Max : 244.00
Mean: 136.60
SD : 55.35
Ent : 5.15
0.00
127.50
255.00

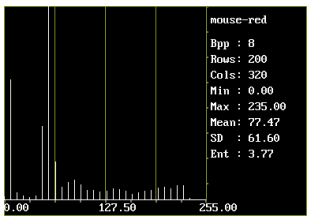

mandrill-green
Bpp : 8
Rows: 256
Cols: 256
Min : 8.00
Max : 220.00
Mean: 128.25
SD : 48.37
Ent : 5.10
0.00
127.50
255.00

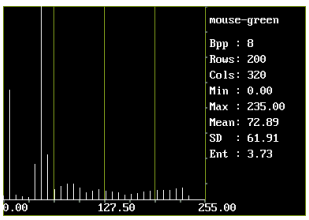

mandrill-blue
Bpp : 8
Rows: 256
Cols: 256
Min : 8.00
Max : 244.00
Mean: 112.64
SD : 62.41
Ent : 5.07
0.00
127.50
255.00

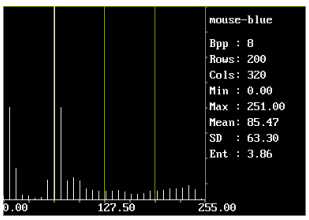

peppers-red
Bpp : 8
Rows: 256
Cols: 256
Min : 8.00
Max : 216.00
Mean: 148.42
SD : 45.35
Ent : 4.95
0.00
127.50
255.00

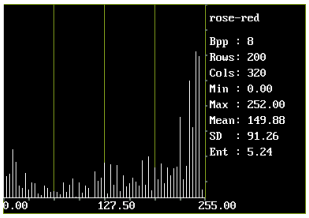

peppers-green
Bpp : 8
Rows: 256
Cols: 256
Min : 0.00
Max : 224.00
Mean: 114.31
SD : 75.73
Ent : 5.01
0.00
127.50
255.00

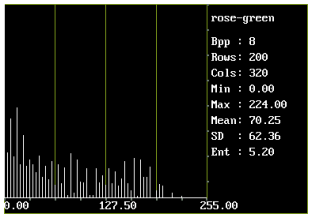

peppers-blue
Bpp : 8
Rows: 256
Cols: 256
Min : 0.00
Max : 212.00
Mean: 65.59
SD : 45.00
Ent : 4.54
0.00
127.50
255.00

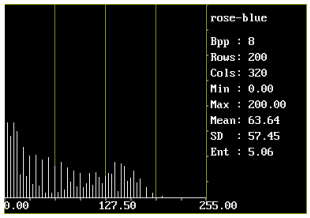

mouse-red
Bpp : 8
Rows: 200
Cols: 320
Min : 0.00
Max : 235.00
Mean: 77.47
SD : 61.60
Ent : 3.77
0.00
127.50
255.00


mouse-green
Bpp : 8
Rows: 200
Cols: 320
Min : 0.00
Max : 235.00
Mean: 72.89
SD : 61.91
Ent : 3.73
0.00
127.50
255.00


mouse-blue
Bpp : 8
Rows: 200
Cols: 320
Min : 0.00
Max : 251.00
Mean: 85.47
SD : 63.30
Ent : 3.86
0.00
127.50
255.00


rose-red
Bpp : 8
Rows: 200
Cols: 320
Min : 0.00
Max : 252.00
Mean: 149.88
SD : 91.26
Ent : 5.24
0.00
127.50
255.00


rose-green
Bpp : 8
Rows: 200
Cols: 320
Min : 0.00
Max : 224.00
Mean: 70.25
SD : 62.36
Ent : 5.20
0.00
127.50
255.00


rose-blue
Bpp : 8
Rows: 200
Cols: 320
Min : 0.00
Max : 200.00
Mean: 63.64
SD : 57.45
Ent : 5.06
0.00
127.50
255.00

Appendix B

Number of Clusters for Splitting Method Using Stopping Rule 1

This appendix shows the number of clusters obtained for some of the images for the splitting method when using stopping rule 1 with various *weight* values as indicated in the tables below. For each image, the first row indicates the *weight* value, w, and the second row indicates the number of clusters, K, obtained for the given *weight* value.

B.1 Results Using Original RGB values

image	*weight* values and corresponding number of clusters								
balls	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	11	10	10	10	10	9	8	8
Lenna	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	11	8	7	5	4	4	4	4
blond	w	0.41	0.42	0.43	0.44	0.46	0.47	0.48	0.49
	K	10	10	10	8	3	3	3	3
jet	w	0.16	0.17	0.18	0.19	0.21	0.22	0.23	0.24
	K	11	10	10	9	7	7	7	6
mandrill	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	15	11	10	10	8	7	7	6

Table B.1: Number of clusters using mean cluster centres, Euclidean distance and average variance.

image	*weight* values and corresponding number of clusters								
balls	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	10	9	9	8	8	8	8	8
Lenna	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	12	12	5	5	4	4	4	4
blond	w	0.36	0.37	0.38	0.39	0.41	0.42	0.43	0.44
	K	11	11	11	10	10	3	3	3
jet	w	0.16	0.17	0.18	0.19	0.26	0.27	0.28	0.29
	K	11	9	9	9	8	8	7	6
mandrill	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	20	17	16	10	8	8	7	6

Table B.2: Number of clusters using mean cluster centres, absolute distance and average variance.

image	*weight* values and corresponding number of clusters								
balls	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	12	12	12	10	10	10	10	9
Lenna	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	13	9	5	5	4	4	4	4
blond	w	0.36	0.37	0.38	0.39	0.41	0.42	0.43	0.44
	K	16	16	16	16	3	3	2	2
jet	w	0.16	0.17	0.18	0.19	0.21	0.22	0.23	0.24
	K	13	9	7	7	6	6	6	6
mandrill	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	18	16	11	10	9	9	8	8

Table B.3: Number of clusters using median cluster centres, Euclidean distance and average variance.

image	*weight* values and corresponding number of clusters								
balls	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	11	10	10	10	10	10	8	8
Lenna	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	11	10	10	5	4	4	4	4
blond	w	0.31	0.32	0.33	0.34	0.36	0.37	0.38	0.39
	K	11	11	11	11	3	3	3	3
jet	w	0.16	0.17	0.18	0.19	0.21	0.22	0.23	0.24
	K	12	10	7	6	5	5	5	5
mandrill	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	21	16	12	11	10	9	9	9

Table B.4: Number of clusters using median cluster centres, absolute distance and average variance.

image	*weight* values and corresponding number of clusters								
balls	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	17	15	12	9	9	8	8	8
Lenna	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	14	12	10	9	5	4	4	4
blond	w	0.26	0.27	0.28	0.29	0.36	0.37	0.38	0.39
	K	12	12	12	11	11	11	11	11
jet	w	0.16	0.17	0.18	0.19	0.21	0.22	0.23	0.24
	K	12	12	9	9	6	5	5	5
mandrill	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	20	13	13	12	9	9	8	8

Table B.5: Number of clusters using mean cluster centres, Euclidean distance and maximum variance.

image	*weight* values and corresponding number of clusters								
balls	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	18	15	14	13	10	10	10	9
Lenna	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	17	17	16	13	5	5	5	4
blond	w	0.41	0.42	0.43	0.44	0.46	0.47	0.48	0.49
	K	12	12	12	4	3	3	3	3
jet	w	0.16	0.17	0.18	0.19	0.26	0.27	0.28	0.29
	K	15	13	13	9	9	9	9	7
mandrill	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	15	14	14	12	11	10	8	8

Table B.6: Number of clusters using mean cluster centres, absolute distance and maximum variance.

image	*weight* values and corresponding number of clusters								
balls	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	14	12	12	10	8	8	8	8
Lenna	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	17	17	17	5	5	5	4	4
blond	w	0.41	0.42	0.43	0.44	0.46	0.47	0.48	0.49
	K	11	11	11	11	3	3	3	3
jet	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	17	16	14	14	11	11	10	10
mandrill	w	0.16	0.17	0.18	0.19	0.21	0.22	0.23	0.24
	K	12	12	12	8	8	7	7	7

Table B.7: Number of clusters using median cluster centres, Euclidean distance and maximum variance.

image	*weight* values and corresponding number of clusters								
balls	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	18	17	13	12	11	11	10	9
Lenna	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	20	18	18	17	5	4	4	4
blond	w	0.36	0.37	0.38	0.39	0.41	0.42	0.43	0.44
	K	15	15	4	4	4	4	4	3
jet	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	16	16	14	14	8	8	5	5
mandrill	w	0.16	0.17	0.18	0.19	0.21	0.22	0.23	0.24
	K	17	15	9	9	9	8	8	7

Table B.8: Number of clusters using median cluster centres, absolute distance and maximum variance.

B.2 Results Using Context

image	*weight* values and corresponding number of clusters								
balls	w	0.06	0.07	0.08	0.09	0.11	0.12	0.13	0.14
	K	15	15	11	10	10	9	8	7
Lenna	w	0.06	0.07	0.08	0.09	0.11	0.12	0.13	0.14
	K	27	25	18	14	12	11	8	8
blond	w	0.21	0.22	0.23	0.24	0.26	0.27	0.28	0.29
	K	16	15	14	12	12	11	11	10
jet	w	0.16	0.17	0.18	0.19	0.21	0.22	0.23	0.24
	K	13	10	10	10	9	8	6	6
mandrill	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	28	13	11	9	9	9	9	9

Table B.9: Number of clusters using *mean* context values only with average variance.

image	*weight* values and corresponding number of clusters								
balls	w	0.06	0.07	0.08	0.09	0.11	0.12	0.13	0.14
	K	18	14	13	13	10	10	9	9
Lenna	w	0.06	0.07	0.08	0.09	0.11	0.12	0.13	0.14
	K	27	20	20	20	8	8	8	5
blond	w	0.21	0.22	0.23	0.24	0.26	0.27	0.28	0.29
	K	11	9	9	9	4	4	4	4
jet	w	0.16	0.17	0.18	0.19	0.21	0.22	0.23	0.24
	K	11	9	7	6	6	5	5	5
mandrill	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	24	14	11	9	9	8	7	6

Table B.10: Number of clusters using *median* context values only with average variance.

image	*weight* values and corresponding number of clusters								
balls	w	0.06	0.07	0.08	0.09	0.11	0.12	0.13	0.14
	K	16	14	11	10	10	9	8	7
Lenna	w	0.06	0.07	0.08	0.09	0.11	0.12	0.13	0.14
	K	28	21	20	14	12	11	8	8
blond	w	0.26	0.27	0.28	0.29	0.31	0.32	0.33	0.34
	K	13	13	11	10	4	4	4	4
jet	w	0.16	0.17	0.18	0.19	0.21	0.22	0.23	0.24
	K	12	10	10	10	9	8	6	6
mandrill	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	26	13	12	9	9	9	9	9

Table B.11: Number of clusters using *weighted* context values only with average variance.

image	*weight* values and corresponding number of clusters								
balls	w	0.06	0.07	0.08	0.09	0.11	0.12	0.13	0.14
	K	22	21	16	12	12	11	10	9
Lenna	w	0.06	0.07	0.08	0.09	0.11	0.12	0.13	0.14
	K	33	26	24	17	13	11	8	7
blond	w	0.36	0.37	0.38	0.39	0.41	0.42	0.43	0.44
	K	10	10	10	9	8	3	3	3
jet	w	0.16	0.17	0.18	0.19	0.21	0.22	0.23	0.24
	K	17	17	15	13	11	11	9	9
mandrill	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	27	13	12	9	9	9	9	9

Table B.12: Number of clusters using *regression* context values only with average variance.

image	*weight* values and corresponding number of clusters								
balls	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	15	12	12	11	9	9	8	8
Lenna	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	28	28	28	23	18	4	4	4
blond	w	0.41	0.42	0.43	0.44	0.46	0.47	0.48	0.49
	K	19	19	3	3	3	3	3	3
jet	w	0.26	0.27	0.28	0.29	0.31	0.32	0.33	0.34
	K	21	19	18	17	6	5	5	5
mandrill	w	0.16	0.17	0.18	0.19	0.21	0.22	0.23	0.24
	K	24	24	14	13	10	10	8	8

Table B.13: Number of clusters using original and *mean* context values with average variance.

image	*weight* values and corresponding number of clusters								
balls	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	17	16	13	12	10	9	9	8
Lenna	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	45	34	28	27	18	4	4	4
blond	w	0.36	0.37	0.38	0.39	0.41	0.42	0.43	0.44
	K	17	17	17	17	3	3	3	3
jet	w	0.21	0.22	0.23	0.24	0.26	0.27	0.28	0.29
	K	38	36	33	13	13	11	7	5
mandrill	w	0.16	0.17	0.18	0.19	0.21	0.22	0.23	0.24
	K	40	20	20	14	12	11	9	9

Table B.14: Number of clusters using original and *median* context values with average variance.

image	*weight* values and corresponding number of clusters								
balls	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	17	13	13	10	9	9	8	8
Lenna	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	28	26	26	24	20	4	4	4
blond	w	0.41	0.42	0.43	0.44	0.46	0.47	0.48	0.49
	K	19	19	3	3	3	3	3	3
jet	w	0.26	0.27	0.28	0.29	0.31	0.32	0.33	0.34
	K	17	17	17	15	6	5	5	5
mandrill	w	0.16	0.17	0.18	0.19	0.21	0.22	0.23	0.24
	K	24	24	14	13	10	10	8	8

Table B.15: Number of clusters using original and *weighted* context values with average variance.

image	*weight* values and corresponding number of clusters								
balls	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	15	15	13	13	10	9	9	8
Lenna	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	28	28	28	27	17	4	4	4
blond	w	0.41	0.42	0.43	0.44	0.46	0.47	0.48	0.49
	K	20	20	20	17	3	3	3	3
jet	w	0.21	0.22	0.23	0.24	0.26	0.27	0.28	0.29
	K	29	28	27	27	18	18	13	6
mandrill	w	0.16	0.17	0.18	0.19	0.21	0.22	0.23	0.24
	K	30	12	12	11	10	9	8	8

Table B.16: Number of clusters using original and *regression* context values with average variance.

B.3 Results Using Different Colour Spaces

image	*weight* values and corresponding number of clusters								
balls	w	0.06	0.07	0.08	0.09	0.11	0.12	0.13	0.14
	K	14	13	11	9	8	8	8	8
Lenna	w	0.06	0.07	0.08	0.09	0.11	0.12	0.13	0.14
	K	13	12	9	6	5	5	5	4
blond	w	0.16	0.17	0.18	0.19	0.36	0.37	0.38	0.39
	K	15	15	15	15	10	10	3	3
jet	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	8	8	7	7	7	7	6	5
mandrill	w	0.06	0.07	0.08	0.09	0.11	0.12	0.13	0.14
	K	22	19	15	12	9	8	8	7

Table B.17: Number of clusters using the XYZ colour space with average variance.

image	*weight* values and corresponding number of clusters								
balls	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	10	9	9	9	9	8	8	7
Lenna	w	0.06	0.07	0.08	0.09	0.11	0.12	0.13	0.14
	K	18	15	12	11	10	10	8	5
blond	w	0.41	0.42	0.43	0.44	0.46	0.47	0.48	0.49
	K	10	10	10	8	3	3	3	3
jet	w	0.16	0.17	0.18	0.19	0.21	0.22	0.23	0.24
	K	10	10	9	9	8	8	8	6
mandrill	w	0.11	0.12	0.13	0.14	0.16	0.17	0.18	0.19
	K	15	14	13	13	10	7	7	7

Table B.18: Number of clusters using the KL colour space with average variance.

image	*weight* values and corresponding number of clusters								
balls	w	0.36	0.37	0.38	0.39	0.41	0.42	0.43	0.44
	K	13	13	13	13	13	12	12	11
Lenna	w	0.06	0.07	0.08	0.09	0.11	0.12	0.13	0.14
	K	9	9	7	7	3	3	3	3
blond	w	0.36	0.37	0.38	0.39	0.41	0.42	0.43	0.44
	K	12	12	12	12	9	9	9	9
jet	w	0.16	0.17	0.18	0.19	0.21	0.22	0.23	0.24
	K	9	8	8	8	8	8	8	6
mandrill	w	0.16	0.17	0.18	0.19	0.21	0.22	0.23	0.24
	K	16	14	14	14	11	11	11	10

Table B.19: Number of clusters using the HSV colour space with average variance.

Appendix C

Results by snob

Tables C.1-C.3 show the number of clusters produced by the *snob* program. On the whole the number of clusters are very large, and the results themselves are quite bad from the visual point of view. We can generally achieve better results by the other methods presented in chapters 5-7 in which the number of clusters is much smaller than those indicated by *snob*. The segmented images produced by *snob* are shown in Figures C.1-C.9.

image	No. of Clusters
balls	20
Lenna	31
molecule	27
teapot	65
ant	41
blond	11
jet	22
mandrill	42
peppers	39
mouse	18
rose	40

Table C.1: Number of clusters using *snob*.

image	*mean*	*median*	*weighted*	*regression*
balls	15	17	15	14
Lenna	15	18	16	14
molecule	29	29	28	20
teapot	67	71	67	42
ant	20	18	20	16
blond	10	6	10	9
jet	14	12	13	12
mandrill	11	13	10	11
peppers	10	17	11	11
mouse	24	20	24	20
rose	22	26	22	19

Table C.2: Number of clusters using *snob* with context values only.

image	*mean*	*median*	*weighted*	*regression*
balls	20	23	22	28
Lenna	30	29	27	27
molecule	71	43	63	63
teapot	92	118	98	88
ant	31	30	29	36
blond	20	20	17	20
jet	23	25	28	23
mandrill	37	34	37	36
peppers	29	31	32	34
mouse	22	20	24	22
rose	38	36	34	32

Table C.3: Number of clusters using *snob* with original and context values.

Appendix D

Results for Davies-Bouldin and Dunn's indexes

For each of the results shown in this appendix, the value of 25 for c is used in equation 7.5 when using the *modified* validity measure.

image	D_{11}	D_{12}	D_{13}
	basic	*basic*	*basic*
balls	25	25	24-25
Lenna	22-25	22-25	22,24
molecule	15-25	19	19
teapot	25	24-25	24-25
ant	18-25	24-25	22-25
blond	25	25	25
jet	19	24	25
mandrill	16-17,19-23	23	23
peppers	25	25	25
mouse	25	24	24
rose	25	25	25

Table D.1: Number of clusters by Dunn's indexes using δ_1.

image	D_{21}		D_{22}		D_{23}	
	basic	*modified*	*basic*	*modified*	*basic*	*modified*
balls	2	11	2	14	2	10-11
Lenna	2	8	2	5	2	5
molecule	2	5	13	13	13	13
teapot	4	17-18	3	6	3	6
ant	2	7	2	7	4	7
blond	2	5	2	5	2	5
jet	2	6	2	6	2	5
mandrill	3	8	5	6	5	6
peppers	2	6	2	6	2	6
mouse	3	6	3	6	3	6
rose	2	6	2	14	2	14

Table D.2: Number of clusters by Dunn's indexes using δ_2.

image	D_{31}		D_{32}		D_{33}	
	basic	*modified*	*basic*	*modified*	*basic*	*modified*
balls	2	11	2	22,24	2	24
Lenna	2	8	2	8	2	8
molecule	2	7	2	13	2	13
teapot	4	8	3	6	3	6
ant	2	7	2	7	2	7
blond	5	9-11	2	13	2	15
jet	2	6	2	6	2	6
mandrill	3	12	4	6	4	6
peppers	2	21	2	6	2	6
mouse	2	6	2	8	2	8
rose	3	5	2	14	2	14

Table D.3: Number of clusters by Dunn's indexes using δ_3.

image	D_{41}		D_{42}		D_{43}	
	basic	*modified*	*basic*	*modified*	*basic*	*modified*
balls	2	11	2	10-11	2	10-11
Lenna	2	8	2	8	2	8
molecule	2	7	2	8	2	8
teapot	4	18	3	6	3	6
ant	2	7	2	7	3	7
blond	5	5	2	13	2	15
jet	2	6	2	6	2	6
mandrill	3	9	4	6	4	6
peppers	2	22	2	6	2	6
mouse	2	6	2	8	2	8
rose	3	6	2	13	2	13

Table D.4: Number of clusters by Dunn's indexes using δ_4.

image	D_{51}		D_{52}		D_{53}	
	basic	*modified*	*basic*	*modified*	*basic*	*modified*
balls	2	11	2	22,24	2	24
Lenna	2	8	2	8	2	8
molecule	2	7	2	13	2	13
teapot	4	8	3	6	3	6
ant	2	7	2	7	2	7
blond	5	9-11	2	13	2	15
jet	2	6	2	6	2	6
mandrill	3	12	4	6	4	6
peppers	2	21	2	6	2	6
mouse	2	6	2	8	2	8
rose	3	5	2	13	2	13

Table D.5: Number of clusters by Dunn's indexes using δ_5.

image	D_{61}		D_{62}		D_{63}	
	basic	*modified*	*basic*	*modified*	*basic*	*modified*
balls	2	5	2	22,24	2	24
Lenna	2	8	2	7	2	7
molecule	2	5	13	13	13	13
teapot	4	8	3	6	3	6
ant	2	8	2	6	2	6
blond	2	5	2	5	2	5
jet	2	6	2	6	2	11
mandrill	3	8	4	11	4	11
peppers	4	24	4	6	4	8
mouse	3	6	3	6	3	6
rose	3	5	2	14	3	14

Table D.6: Number of clusters by Dunn's indexes using δ_6.

image	*basic*	*modified*
balls	2	6
Lenna	2	20
molecule	2	8
teapot	3	19
ant	3	7
blond	5	25
jet	2	7
mandrill	4	19
peppers	22	22
mouse	4	6
rose	3	10

Table D.7: Number of clusters using Davies-Bouldin index.

Appendix E

Results for Proposed Validity Measure

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	0.001958	1.446495	0.768046	2.496967	1.967003	3.263815
3	2.326906	0.168621	0.411888	1.284602	1.541595	1.661153
4	0.767318	0.000460	0.204723	0.386065	0.509590	0.519636
5	0.427259	0.379867	0.099824	0.196497	0.269142	0.351314
6	0.923371	0.339169	0.000313	0.125874	0.144033	0.286726
7	0.855196	0.311002	0.377874	0.063245	0.150705	0.285102

Table E.1: *Modified* validity measure for synthetic images using mean cluster centres and Euclidean distance with $c = 10$.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	0.001958	1.446495	0.782339	2.496967	1.967003	3.263815
3	2.326906	0.168621	0.411888	1.284602	1.541595	1.661153
4	0.761568	0.000460	0.102982	0.386065	0.509590	0.519636
5	0.466936	0.379867	0.039537	0.196497	0.269142	0.351314
6	0.474852	0.339169	0.000313	0.125874	0.144033	0.286726
7	0.866618	0.311002	0.377874	0.063245	0.150705	0.285102

Table E.2: *Modified* validity measure for synthetic images using mean cluster centres and absolute distance with $c = 10$.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	0.001957	2.047050	0.699183	2.497454	1.947312	2.279233
3	2.768601	0.295261	0.455007	1.284353	1.134566	1.120733
4	0.879330	0.000461	111.876912	0.385929	0.388112	0.406934
5	0.432692	0.413180	0.050339	0.196459	0.207852	0.484815
6	0.479482	0.369343	0.000312	0.125818	0.143620	0.400084
7	0.930589	0.337151	0.421319	0.063198	0.126305	0.562660

Table E.3: *Modified* validity measure for synthetic images using median cluster centres and Euclidean distance with $c = 10$.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	0.001957	2.047050	0.699184	2.497454	1.947312	2.279233
3	2.768601	0.295261	0.507207	1.284353	1.134721	1.120733
4	0.881793	0.000461	0.150565	0.385929	0.388635	0.406779
5	0.445148	0.413180	0.051666	0.196459	0.206339	0.484818
6	0.485166	0.369343	0.000312	0.125818	0.143620	0.461793
7	0.943202	0.337151	0.421319	0.063198	0.126309	0.334631

Table E.4: *Modified* validity measure for synthetic images using median cluster centres and absolute distance with $c = 10$.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	0.002741	2.024786	1.075101	3.495224	2.753387	4.568649
3	3.150139	0.228278	0.557609	1.739080	2.086994	2.248849
4	0.901833	0.000540	0.240612	0.453744	0.598924	0.610732
5	0.436325	0.387928	0.101943	0.200666	0.274853	0.358768
6	0.923988	0.339396	0.000313	0.125959	0.144129	0.286918
7	0.855203	0.311005	0.377877	0.063246	0.150706	0.285104

Table E.5: *Modified* validity measure for synthetic images using mean cluster centres and Euclidean distance with $c = 15$.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	0.002741	2.024786	1.095108	3.495224	2.753387	4.568649
3	3.150139	0.228278	0.557609	1.739080	2.086994	2.248849
4	0.895075	0.000540	0.121035	0.453744	0.598924	0.610732
5	0.476844	0.387928	0.040376	0.200666	0.274853	0.358768
6	0.475169	0.339396	0.000313	0.125959	0.144129	0.286918
7	0.866624	0.311005	0.377877	0.063246	0.150706	0.285104

Table E.6: *Modified* validity measure for synthetic images using mean cluster centres and absolute distance with $c = 15$.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	0.002740	2.865435	0.978708	3.495905	2.725824	3.190443
3	3.748101	0.399721	0.615984	1.738742	1.535963	1.517236
4	1.033481	0.000541	131.489565	0.453584	0.456151	0.478272
5	0.441873	0.421948	0.051407	0.200627	0.212262	0.495103
6	0.479803	0.369589	0.000313	0.125902	0.143716	0.400351
7	0.930596	0.337154	0.421322	0.063198	0.126306	0.562664

Table E.7: *Modified* validity measure for synthetic images using median cluster centres and Euclidean distance with $c = 15$.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	0.002740	2.865435	0.978710	3.495905	2.725824	3.190443
3	3.748101	0.399721	0.686651	1.738742	1.536172	1.517236
4	1.036376	0.000541	0.176960	0.453584	0.456765	0.478090
5	0.454594	0.421948	0.052762	0.200627	0.210717	0.495105
6	0.485490	0.369589	0.000313	0.125902	0.143716	0.462101
7	0.943209	0.337154	0.421322	0.063198	0.126310	0.334634

Table E.8: *Modified* validity measure for synthetic images using median cluster centres and absolute distance with $c = 15$.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	0.003524	2.603078	1.382157	4.493482	3.539771	5.873483
3	3.973372	0.287934	0.703330	2.193557	2.632392	2.836546
4	1.036348	0.000621	0.276501	0.521424	0.688258	0.701827
5	0.445391	0.395988	0.104061	0.204836	0.280564	0.366222
6	0.924605	0.339623	0.000313	0.126043	0.144226	0.287109
7	0.855209	0.311007	0.377879	0.063246	0.150707	0.285106

Table E.9: *Modified* validity measure for synthetic images using mean cluster centres and Euclidean distance with $c = 20$.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	0.003524	2.603078	1.407878	4.493482	3.539771	5.873483
3	3.973372	0.287934	0.703330	2.193557	2.632392	2.836546
4	1.028582	0.000621	0.139089	0.521424	0.688258	0.701827
5	0.486752	0.395988	0.041215	0.204836	0.280564	0.366222
6	0.475487	0.339623	0.000313	0.126043	0.144226	0.287109
7	0.866631	0.311007	0.377879	0.063246	0.150707	0.285106

Table E.10: *Modified* validity measure for synthetic images using mean cluster centres and absolute distance with $c = 20$.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	0.003523	3.683821	1.258233	4.494357	3.504336	4.101652
3	4.727600	0.504181	0.776960	2.193132	1.937360	1.913739
4	1.187633	0.000622	151.102217	0.521240	0.524189	0.549610
5	0.451054	0.430715	0.052475	0.204796	0.216672	0.505390
6	0.480123	0.369836	0.000313	0.125986	0.143812	0.400619
7	0.930603	0.337156	0.421325	0.063199	0.126307	0.562668

Table E.11: *Modified* validity measure for synthetic images using median cluster centres and Euclidean distance with $c = 20$.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	0.003523	3.683821	1.258235	4.494357	3.504336	4.101652
3	4.727600	0.504181	0.866096	2.193132	1.937623	1.913739
4	1.190960	0.000622	0.203355	0.521240	0.524895	0.549400
5	0.464039	0.430715	0.053858	0.204796	0.215095	0.505393
6	0.485814	0.369836	0.000313	0.125986	0.143812	0.462410
7	0.943216	0.337156	0.421325	0.063199	0.126311	0.334636

Table E.12: *Modified* validity measure for synthetic images using median cluster centres and absolute distance with $c = 20$.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	0.004307	3.181369	1.689212	5.491739	4.326155	7.178317
3	4.796605	0.347590	0.849051	2.648035	3.177791	3.424243
4	1.170863	0.000701	0.312391	0.589103	0.777593	0.792922
5	0.454457	0.404049	0.106179	0.209005	0.286274	0.373677
6	0.925222	0.339849	0.000313	0.126127	0.144322	0.287301
7	0.855215	0.311009	0.377882	0.063247	0.150709	0.285108

Table E.13: *Modified* validity measure for synthetic images using mean cluster centres and Euclidean distance with $c = 25$.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	0.004307	3.181369	1.720648	5.491739	4.326155	7.178317
3	4.796605	0.347590	0.849051	2.648035	3.177791	3.424243
4	1.162090	0.000701	0.157142	0.589103	0.777593	0.792922
5	0.496659	0.404049	0.042054	0.209005	0.286274	0.373677
6	0.475804	0.339849	0.000313	0.126127	0.144322	0.287301
7	0.866637	0.311009	0.377882	0.063247	0.150709	0.285108

Table E.14: *Modified* validity measure for synthetic images using mean cluster centres and absolute distance with $c = 25$.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	0.004305	4.502207	1.537758	5.492809	4.282848	5.012862
3	5.707100	0.608641	0.937937	2.647521	2.338757	2.310241
4	1.341784	0.000703	170.714870	0.588895	0.592227	0.620948
5	0.460235	0.439482	0.053543	0.208965	0.221083	0.515677
6	0.480444	0.370083	0.000313	0.126070	0.143908	0.400886
7	0.930610	0.337159	0.421329	0.063199	0.126308	0.562672

Table E.15: *Modified* validity measure for synthetic images using median cluster centres and Euclidean distance with $c = 25$.

K	*two*	*four*	*six*	*eight*	*ten*	*fifteen*
2	0.004305	4.502207	1.537761	5.492809	4.282848	5.012862
3	5.707100	0.608641	1.045540	2.647521	2.339075	2.310241
4	1.345543	0.000703	0.229750	0.588895	0.593025	0.620711
5	0.473485	0.439482	0.054954	0.208965	0.219474	0.515680
6	0.486138	0.370083	0.000313	0.126070	0.143908	0.462719
7	0.943223	0.337159	0.421329	0.063199	0.126312	0.334639

Table E.16: *Modified* validity measure for synthetic images using median cluster centres and absolute distance with $c = 25$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.1384	0.6948	0.0945	1.3990	0.5541	0.8336	0.2657	1.4335	0.7915	0.2589	0.5327
3	0.2635	0.6003	0.2071	0.3300	0.2503	0.9535	0.3734	0.7692	0.8429	0.3968	0.3282
4	0.4734	0.2936	0.0545	0.2190	0.1599	0.3722	0.1498	0.3358	0.3695	0.1062	0.1827
5	0.2778	0.2856	0.0474	0.1618	0.1168	0.2304	0.0923	0.3178	0.2624	0.0685	0.1953
6	0.2686	0.2643	0.0313	0.1320	0.1941	0.2573	0.0848	0.3181	0.2237	0.0576	0.1725
7	0.2593	0.2550	0.0200	0.1668	0.1178	0.3081	0.1573	0.3955	0.3726	0.0662	0.2086

Table E.17: *Modified* validity measure for natural images using mean cluster centres and Euclidean distance with $c = 10$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.1338	0.7144	0.0941	1.3908	0.5541	0.9038	0.2654	1.4626	0.7948	0.2589	0.5397
3	0.2815	0.5768	0.2051	0.3490	0.2593	1.1582	0.3809	0.7100	0.9735	0.3968	0.3637
4	0.4638	0.2834	0.0692	0.1983	0.1933	0.4361	0.1491	0.3362	0.3743	0.1062	0.2244
5	0.3162	0.3879	0.0502	0.3092	0.0939	0.3198	0.1555	0.5662	0.2682	0.0734	0.3146
6	0.2690	0.4511	0.0496	0.5545	0.1933	0.4056	0.1382	0.3473	0.2528	0.0575	0.1685
7	0.2648	0.3893	0.0299	0.5161	0.1888	0.3326	0.1934	0.4385	0.2344	0.0663	0.1882

Table E.18: *Modified* validity measure for natural images using mean cluster centres and absolute distance with $c = 10$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	10.1092	0.7318	0.0851	1.0699	0.4787	0.7578	0.2677	1.2205	0.8092	0.2645	0.4403
3	1.7634	0.9180	0.2055	0.2745	0.2799	1.2011	0.6353	1.1122	0.5989	0.4377	0.3033
4	0.4927	0.2558	0.0505	0.3030	0.1518	3.2613	0.2033	0.4056	0.5258	4.0247	0.1738
5	0.4459	0.2989	0.0672	0.1777	0.2197	2.0763	0.1122	0.3697	0.2863	0.0884	0.1864
6	0.3563	1.4271	0.0482	0.1396	0.1486	1.4459	0.4918	0.4719	0.2349	0.0565	0.1774
7	0.6352	1.2492	0.0295	0.1763	0.1290	1.3714	0.6432	0.4528	0.2568	0.0977	0.5364

Table E.19: *Modified* validity measure for natural images using median cluster centres and Euclidean distance with $c = 10$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	10.1092	0.7360	0.0852	1.4536	0.4737	0.8070	0.2681	1.2611	0.8042	0.2645	0.4695
3	1.7634	0.7792	0.2294	0.2777	0.3018	1.2595	0.4430	1.0721	0.9116	0.4377	0.3087
4	0.5398	0.2566	0.0641	0.4393	0.1629	3.4893	0.2277	0.4934	0.4681	4.0247	0.1756
5	0.4858	0.4031	0.0381	0.1889	0.1331	2.2688	0.1247	0.3953	0.3452	0.0885	0.2336
6	0.5127	0.3402	0.0390	0.2374	0.2177	2.8985	0.4972	0.4128	0.6234	0.0566	0.3327
7	0.4484	0.4207	0.0276	0.1820	0.1396	2.6247	0.6500	0.4351	0.4696	0.0978	0.2432

Table E.20: *Modified* validity measure for natural images using median cluster centres and absolute distance with $c = 10$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.1938	0.9726	0.1323	1.9583	0.7756	1.1669	0.3719	2.0066	1.1079	0.3624	0.7457
3	0.3567	0.8127	0.2804	0.4467	0.3389	1.2909	0.5056	1.0413	1.1412	0.5372	0.4443
4	0.5563	0.3451	0.0641	0.2574	0.1879	0.4374	0.1761	0.3947	0.4343	0.1248	0.2147
5	0.2837	0.2917	0.0484	0.1653	0.1193	0.2353	0.0942	0.3246	0.2679	0.0700	0.1994
6	0.2688	0.2645	0.0313	0.1321	0.1942	0.2575	0.0849	0.3183	0.2238	0.0577	0.1726
7	0.2593	0.2550	0.0200	0.1668	0.1178	0.3081	0.1573	0.3956	0.3726	0.0662	0.2086

Table E.21: *Modified* validity measure for natural images using mean cluster centres and Euclidean distance with $c = 15$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.1873	1.0000	0.1318	1.9468	0.7756	1.2651	0.3715	2.0474	1.1126	0.3624	0.7555
3	0.3810	0.7809	0.2776	0.4725	0.3511	1.5680	0.5157	0.9612	1.3179	0.5372	0.4923
4	0.5452	0.3330	0.0813	0.2331	0.2272	0.5125	0.1753	0.3952	0.4399	0.1248	0.2637
5	0.3230	0.3961	0.0513	0.3157	0.0959	0.3266	0.1588	0.5782	0.2739	0.0749	0.3213
6	0.2692	0.4514	0.0496	0.5549	0.1935	0.4059	0.1383	0.3476	0.2529	0.0575	0.1686
7	0.2648	0.3893	0.0299	0.5162	0.1889	0.3326	0.1934	0.4385	0.2344	0.0663	0.1882

Table E.22: *Modified* validity measure for natural images using mean cluster centres and absolute distance with $c = 15$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	14.1508	1.0244	0.1192	1.4977	0.6701	1.0607	0.3748	1.7085	1.1326	0.3702	0.6164
3	2.3873	1.2428	0.2783	0.3715	0.3789	1.6261	0.8600	1.5057	0.8108	0.5925	0.4107
4	0.5791	0.3006	0.0594	0.3561	0.1784	3.8331	0.2390	0.4767	0.6179	4.7303	0.2043
5	0.4554	0.3053	0.0687	0.1815	0.2244	2.1204	0.1146	0.3775	0.2924	0.0903	0.1903
6	0.3566	1.4280	0.0482	0.1397	0.1487	1.4469	0.4921	0.4723	0.2350	0.0566	0.1776
7	0.6352	1.2492	0.0295	0.1763	0.1290	1.3714	0.6432	0.4528	0.2568	0.0977	0.5364

Table E.23: *Modified* validity measure for natural images using median cluster centres and Euclidean distance with $c = 15$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	14.1508	1.0302	0.1193	2.0348	0.6631	1.1296	0.3753	1.7653	1.1256	0.3702	0.6572
3	2.3873	1.0548	0.3106	0.3759	0.4085	1.7051	0.5998	1.4514	1.2341	0.5926	0.4179
4	0.6344	0.3016	0.0754	0.5163	0.1914	4.1010	0.2676	0.5799	0.5502	4.7303	0.2063
5	0.4961	0.4117	0.0389	0.1929	0.1360	2.3170	0.1274	0.4037	0.3525	0.0903	0.2386
6	0.5130	0.3404	0.0390	0.2376	0.2178	2.9005	0.4976	0.4131	0.6238	0.0566	0.3329
7	0.4484	0.4207	0.0276	0.1820	0.1396	2.6247	0.6500	0.4351	0.4696	0.0978	0.2432

Table E.24: *Modified* validity measure for natural images using median cluster centres and absolute distance with $c = 15$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.2491	1.2503	0.1701	2.5176	0.9971	1.5002	0.4781	2.5797	1.4243	0.4659	0.9586
3	0.4499	1.0251	0.3536	0.5635	0.4275	1.6282	0.6377	1.3134	1.4394	0.6776	0.5603
4	0.6393	0.3966	0.0736	0.2958	0.2159	0.5027	0.2024	0.4536	0.4991	0.1434	0.2468
5	0.2896	0.2977	0.0494	0.1687	0.1218	0.2402	0.0962	0.3313	0.2735	0.0714	0.2036
6	0.2690	0.2646	0.0313	0.1322	0.1943	0.2577	0.0849	0.3186	0.2240	0.0577	0.1727
7	0.2593	0.2550	0.0200	0.1668	0.1178	0.3081	0.1573	0.3956	0.3726	0.0662	0.2086

Table E.25: *Modified* validity measure for natural images using mean cluster centres and Euclidean distance with $c = 20$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.2408	1.2856	0.1694	2.5029	0.9971	1.6264	0.4776	2.6321	1.4303	0.4659	0.9713
3	0.4806	0.9850	0.3502	0.5960	0.4428	1.9777	0.6504	1.2124	1.6624	0.6776	0.6210
4	0.6265	0.3827	0.0934	0.2679	0.2611	0.5890	0.2014	0.4541	0.5055	0.1434	0.3031
5	0.3297	0.4043	0.0523	0.3223	0.0979	0.3333	0.1621	0.5902	0.2796	0.0765	0.3279
6	0.2694	0.4517	0.0496	0.5553	0.1936	0.4061	0.1384	0.3478	0.2531	0.0576	0.1688
7	0.2648	0.3893	0.0299	0.5162	0.1889	0.3326	0.1934	0.4385	0.2344	0.0663	0.1882

Table E.26: *Modified* validity measure for natural images using mean cluster centres and absolute distance with $c = 20$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	18.1923	1.3170	0.1532	1.9254	0.8615	1.3637	0.4818	2.1964	1.4561	0.4760	0.7924
3	3.0112	1.5676	0.3510	0.4686	0.4779	2.0510	1.0848	1.8992	1.0226	0.7474	0.5180
4	0.6655	0.3455	0.0682	0.4092	0.2050	4.4048	0.2746	0.5478	0.7101	5.4358	0.2347
5	0.4648	0.3116	0.0701	0.1853	0.2290	2.1644	0.1169	0.3853	0.2985	0.0922	0.1943
6	0.3568	1.4290	0.0482	0.1398	0.1488	1.4478	0.4925	0.4726	0.2352	0.0566	0.1777
7	0.6353	1.2493	0.0295	0.1763	0.1290	1.3714	0.6432	0.4528	0.2568	0.0977	0.5364

Table E.27: *Modified* validity measure for natural images using median cluster centres and Euclidean distance with $c = 20$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	18.1923	1.3244	0.1534	2.6159	0.8525	1.4522	0.4825	2.2695	1.4471	0.4760	0.8449
3	3.0112	1.3305	0.3918	0.4741	0.5153	2.1507	0.7565	1.8307	1.5566	0.7475	0.5271
4	0.7290	0.3465	0.0866	0.5933	0.2200	4.7127	0.3075	0.6664	0.6323	5.4358	0.2371
5	0.5064	0.4202	0.0397	0.1969	0.1388	2.3651	0.1300	0.4121	0.3599	0.0922	0.2435
6	0.5133	0.3406	0.0391	0.2378	0.2180	2.9024	0.4979	0.4134	0.6242	0.0567	0.3331
7	0.4484	0.4207	0.0276	0.1820	0.1396	2.6247	0.6500	0.4351	0.4696	0.0978	0.2432

Table E.28: *Modified* validity measure for natural images using median cluster centres and absolute distance with $c = 20$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.3044	1.5281	0.2079	3.0769	1.2186	1.8335	0.5843	3.1528	1.7407	0.5694	1.1716
3	0.5431	1.2374	0.4269	0.6802	0.5160	1.9656	0.7698	1.5855	1.7376	0.8180	0.6764
4	0.7223	0.4481	0.0832	0.3342	0.2440	0.5679	0.2286	0.5124	0.5639	0.1621	0.2788
5	0.2955	0.3038	0.0504	0.1721	0.1243	0.2451	0.0981	0.3380	0.2791	0.0729	0.2077
6	0.2692	0.2648	0.0313	0.1322	0.1945	0.2579	0.0850	0.3188	0.2241	0.0577	0.1728
7	0.2593	0.2550	0.0200	0.1668	0.1178	0.3081	0.1573	0.3956	0.3726	0.0662	0.2086

Table E.29: *Modified* validity measure for natural images using mean cluster centres and Euclidean distance with $c = 25$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.2943	1.5712	0.2070	3.0589	1.2186	1.9877	0.5837	3.2169	1.7481	0.5694	1.1870
3	0.5802	1.1890	0.4227	0.7195	0.5345	2.3875	0.7852	1.4636	2.0068	0.8180	0.7497
4	0.7078	0.4324	0.1055	0.3027	0.2949	0.6654	0.2276	0.5131	0.5712	0.1621	0.3424
5	0.3364	0.4126	0.0534	0.3289	0.0999	0.3401	0.1654	0.6022	0.2853	0.0780	0.3346
6	0.2695	0.4520	0.0497	0.5557	0.1937	0.4064	0.1385	0.3480	0.2533	0.0576	0.1689
7	0.2648	0.3893	0.0299	0.5162	0.1889	0.3326	0.1934	0.4385	0.2344	0.0663	0.1882

Table E.30: *Modified* validity measure for natural images using mean cluster centres and absolute distance with $c = 25$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	22.2339	1.6096	0.1873	2.3531	1.0529	1.6666	0.5889	2.6844	1.7796	0.5817	0.9685
3	3.6351	1.8924	0.4237	0.5657	0.5769	2.4760	1.3095	2.2927	1.2345	0.9022	0.6253
4	0.7518	0.3903	0.0771	0.4623	0.2316	4.9765	0.3103	0.6189	0.8023	6.1414	0.2652
5	0.4743	0.3180	0.0715	0.1890	0.2337	2.2085	0.1193	0.3932	0.3046	0.0941	0.1982
6	0.3571	1.4299	0.0483	0.1399	0.1489	1.4488	0.4928	0.4729	0.2353	0.0566	0.1778
7	0.6353	1.2493	0.0295	0.1763	0.1290	1.3714	0.6432	0.4528	0.2568	0.0977	0.5364

Table E.31: *Modified* validity measure for natural images using median cluster centres and Euclidean distance with $c = 25$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	22.2339	1.6187	0.1874	3.1971	1.0419	1.7749	0.5896	2.7736	1.7686	0.5817	1.0326
3	3.6351	1.6061	0.4730	0.5724	0.6221	2.5963	0.9133	2.2099	1.8791	0.9023	0.6363
4	0.8236	0.3915	0.0978	0.6703	0.2485	5.3243	0.3475	0.7529	0.7143	6.1414	0.2679
5	0.5167	0.4288	0.0405	0.2009	0.1416	2.4133	0.1327	0.4204	0.3672	0.0941	0.2485
6	0.5137	0.3409	0.0391	0.2379	0.2181	2.9043	0.4982	0.4136	0.6246	0.0567	0.3334
7	0.4484	0.4207	0.0276	0.1820	0.1396	2.6248	0.6501	0.4351	0.4696	0.0978	0.2432

Table E.32: *Modified* validity measure for natural images using median cluster centres and absolute distance with $c = 25$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.0307	0.1379	0.0212	0.2828	0.1132	0.1476	0.0579	0.3234	0.1577	0.0514	0.1055
3	0.0665	0.1689	0.0558	0.0959	0.0773	0.2357	0.0984	0.1837	0.2592	0.1145	0.0936
4	0.2907	0.1737	0.0398	0.1619	0.0989	0.2894	0.1371	0.2284	0.2308	0.0690	0.1137
5	0.2447	0.2655	0.0357	0.1413	0.0782	0.2488	0.2069	0.2936	0.2245	0.0691	0.1668
6	0.2112	0.2276	0.0389	0.1998	0.0984	0.3871	0.2973	0.2914	0.2152	0.0501	0.1591
7	0.1759	0.2163	0.0257	0.1745	0.1187	0.3994	0.3390	0.3272	0.2597	0.0524	0.1867
8	0.2780	0.3479	0.0235	0.1380	0.2332	0.3282	0.3760	0.3126	0.3561	0.0671	0.1962
9	0.2263	0.3578	0.0348	0.1223	0.2416	0.3256	0.4783	0.2915	0.3067	0.0966	0.3456
10	0.2099	0.3380	0.0381	0.1374	0.2180	0.3538	0.4515	0.3713	0.2178	0.2806	0.4557
11	0.2622	0.3089	0.0577	0.2093	0.2127	0.3317	0.4445	0.2841	0.3883	0.2676	0.4197
12	0.2549	0.3684	0.0513	0.2238	0.2066	0.3331	0.4801	0.3686	0.3505	0.3054	0.4717
13	0.2359	0.3767	0.0441	0.1989	0.2412	0.4153	0.5278	0.3557	0.4001	0.3055	0.4153
14	0.4888	0.4415	0.0886	0.2809	0.1983	0.3726	0.5729	0.3385	0.4234	0.3858	0.3670
15	0.4546	0.4010	0.0905	0.2983	0.2064	0.3560	0.5754	0.3394	0.3617	0.3832	0.4616
16	0.4396	0.4026	0.0881	0.3001	0.2265	0.3273	0.5676	0.3177	0.3137	0.4029	0.4906
17	0.4579	0.4218	0.0731	0.2622	0.2224	0.3428	0.5440	0.3266	0.4269	0.3951	0.4172
18	0.4285	0.5362	0.0749	0.3835	0.2098	0.3473	0.5698	0.3392	0.4724	0.3876	0.3780
19	0.4124	0.5031	0.0736	0.2923	0.2181	0.3743	0.5417	0.3807	0.4650	0.3840	0.4111
20	0.3985	0.4073	0.0717	0.2846	0.2163	0.3850	0.5308	0.3786	0.4746	0.2648	0.3664
21	0.5843	0.3997	0.0699	0.2744	0.2437	0.3809	0.5147	0.3924	0.3851	0.3829	0.3427
22	0.5665	0.4103	0.0680	0.3152	0.2443	0.3719	0.5003	0.4179	0.4047	0.3980	0.3197
23	0.5524	0.4291	0.0676	0.3053	0.2647	0.3867	0.5319	0.4108	0.4027	0.4318	0.2895
24	0.5454	0.3881	0.0666	0.3002	0.2157	0.5719	0.7342	0.4087	0.3784	0.4156	0.2862
25	0.5408	0.3815	0.0659	0.2945	0.2790	0.5691	0.7310	0.4096	0.3808	0.9188	0.4183

Table E.33: *Basic* validity measure for natural images using *mean* context values only.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.0272	0.1331	0.0191	0.2807	0.1101	0.1453	0.0507	0.2945	0.1509	0.0516	0.1054
3	0.0736	0.1639	0.0630	0.0953	0.0722	0.2257	0.0971	0.1993	0.2377	0.1176	0.0931
4	0.2879	0.1714	0.0353	0.1492	0.0970	0.2648	0.0788	0.2100	0.2225	0.0668	0.1129
5	0.3282	0.2644	0.0451	0.1487	0.0798	0.2238	0.0908	0.2936	0.2211	0.0725	0.1736
6	0.2407	0.2607	0.0318	0.1320	0.0916	0.3380	0.1953	0.2619	0.2057	0.0537	0.1568
7	0.1876	0.2314	0.0208	0.1701	0.1072	0.3582	0.2284	0.2757	0.2330	0.0553	0.1895
8	0.2971	0.2428	0.0177	0.1266	0.1258	0.3190	0.3110	0.3388	0.4183	0.0710	0.1964
9	0.2736	0.2171	0.0343	0.1144	0.2883	0.3457	0.3112	0.3102	0.3511	0.0984	0.3379
10	0.2789	0.2400	0.0284	0.2276	0.2436	0.3178	0.3993	0.4009	0.4378	0.1188	0.2864
11	0.2922	0.4283	0.0240	0.1966	0.2519	0.4589	0.3840	0.3371	0.3660	0.1197	0.3202
12	0.2952	0.3932	0.0292	0.2019	0.2054	0.3217	0.3588	0.3469	0.3442	0.1691	0.2753
13	0.2860	0.3761	0.0236	0.2112	0.1929	0.3305	0.3327	0.3024	0.3177	0.1556	0.3904
14	0.2754	0.4801	0.0386	0.2145	0.1912	0.3304	0.3155	0.4392	0.3114	0.2811	0.3409
15	0.2563	0.4762	0.0351	0.1978	0.1903	0.3287	0.3021	0.3667	0.4162	0.2556	0.3448
16	0.2427	0.4243	0.0358	0.1782	0.2105	0.3295	0.2863	0.3566	0.4119	0.2436	0.2971
17	0.2371	0.4176	0.0342	0.1773	0.2507	0.3943	0.2826	0.5268	0.4099	0.2374	0.2917
18	0.2311	0.3953	0.0393	0.2523	0.2441	0.3545	0.2782	0.5889	0.3549	0.2431	0.2925
19	0.3551	0.3791	0.0396	0.2480	0.2424	0.3174	0.3661	0.5707	0.3821	0.2728	0.2551
20	0.3479	0.3725	0.0428	0.2995	0.2191	0.3042	0.3609	0.5670	0.4551	0.2809	0.2395
21	0.3424	0.3706	0.0419	0.2961	0.2174	0.4158	0.3727	0.4471	0.4435	0.2765	0.2239
22	0.3355	0.5596	0.0416	0.2253	0.3379	0.4225	0.4728	0.4084	0.3371	0.4537	0.2780
23	0.3316	0.5157	0.0413	0.2580	0.3114	0.3505	0.4772	0.5127	0.6351	0.4428	0.2670
24	0.3301	0.5934	0.0410	0.2615	0.3425	0.3462	0.4470	0.5042	0.6342	0.4419	0.2461
25	0.3242	0.5234	0.0409	0.2446	0.3406	0.3938	0.4394	0.5015	0.4883	0.4425	0.2426

Table E.34: *Basic* validity measure for natural images using *median* context values only.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.0291	0.1379	0.0213	0.2827	0.1128	0.1480	0.0577	0.3224	0.1577	0.0510	0.1055
3	0.0725	0.1685	0.0570	0.0963	0.0770	0.2377	0.0992	0.1836	0.2595	0.1138	0.0938
4	0.3023	0.1745	0.0395	0.1621	0.0997	0.2899	0.1398	0.2270	0.2312	0.0683	0.1136
5	0.2581	0.2649	0.0355	0.1417	0.0764	0.2493	0.1975	0.2944	0.2247	0.0699	0.1675
6	0.2242	0.2274	0.0384	0.2002	0.0973	0.3903	0.2979	0.2952	0.2153	0.0507	0.1597
7	0.1754	0.2172	0.0259	0.1737	0.1185	0.3924	0.3434	0.3247	0.2593	0.0511	0.1856
8	0.2781	0.3247	0.0234	0.1363	0.2429	0.3290	0.3792	0.3146	0.3544	0.0681	0.1963
9	0.2289	0.3612	0.0386	0.1242	0.2445	0.3305	0.4868	0.2906	0.3116	0.0968	0.3427
10	0.2110	0.3430	0.0378	0.1418	0.2218	0.3729	0.4497	0.3666	0.2177	0.3069	0.4604
11	0.2618	0.3092	0.0527	0.2085	0.2155	0.3317	0.5445	0.2822	0.3931	0.2936	0.4255
12	0.2533	0.3748	0.0473	0.2236	0.1985	0.3342	0.4950	0.3722	0.3709	0.3301	0.4754
13	0.2342	0.3693	0.0462	0.2040	0.2420	0.4240	0.5290	0.3588	0.4036	0.3462	0.4190
14	0.4913	0.4379	0.0353	0.2807	0.2187	0.3713	0.6075	0.3415	0.4339	0.3673	0.3696
15	0.4573	0.4024	0.0900	0.2987	0.2212	0.3555	0.5821	0.3408	0.3695	0.4420	0.4650
16	0.4421	0.4094	0.0892	0.2986	0.2095	0.3242	0.5722	0.3172	0.3505	0.4298	0.4869
17	0.4604	0.4295	0.0777	0.2581	0.2253	0.3414	0.5857	0.3434	0.3454	0.2982	0.4144
18	0.4309	0.4795	0.0730	0.1979	0.2142	0.3463	0.5734	0.3925	0.3825	0.4809	0.3755
19	0.4154	0.3419	0.0830	0.2910	0.2117	0.3658	0.5478	0.3609	0.4194	0.5098	0.3369
20	0.4013	0.4881	0.0794	0.2804	0.2103	0.3848	0.5426	0.3800	0.4311	0.4847	0.3156
21	0.3778	0.4824	0.0790	0.2718	0.2415	0.3723	0.5155	0.3941	0.5011	0.5360	0.3521
22	0.3653	0.4802	0.0766	0.2613	0.2427	0.3775	0.5007	0.4024	0.4841	0.5215	0.3269
23	0.3575	0.4216	0.0755	0.3120	0.2700	0.3886	0.4832	0.4225	0.4706	0.4942	0.2969
24	0.3529	0.4000	0.0715	0.3069	0.2154	0.3822	0.4968	0.4110	0.4672	0.4782	0.4410
25	0.5253	0.3782	0.0711	0.2943	0.2695	0.5786	0.7038	0.3942	0.4541	0.4595	0.4365

Table E.35: *Basic* validity measure for natural images using *weighted* context values only.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.0283	0.1375	0.0201	0.2818	0.1115	0.1535	0.0570	0.3275	0.1582	0.0513	0.1064
3	0.0741	0.1675	0.0753	0.0971	0.0753	0.2589	0.0989	0.1871	0.2491	0.1155	0.0949
4	0.3023	0.1761	0.0377	0.1542	0.0990	0.2267	0.1023	0.2245	0.2344	0.0692	0.1136
5	0.2354	0.2716	0.0514	0.1469	0.0816	0.2511	0.1243	0.2979	0.2320	0.0711	0.1758
6	0.1993	0.2326	0.0380	0.2016	0.0958	0.3809	0.3697	0.2890	0.2218	0.0521	0.1304
7	0.2037	0.2073	0.0245	0.1738	0.1278	0.4483	0.3566	0.3223	0.2692	0.0534	0.2163
8	0.3113	0.3005	0.0222	0.1367	0.2432	0.3862	0.3499	0.3211	0.3877	0.0695	0.1882
9	0.2743	0.3652	0.0317	0.1228	0.2768	0.3325	0.4613	0.2981	0.3157	0.0994	0.3511
10	0.6699	0.3578	0.0353	0.2342	0.2539	0.3273	0.5460	0.3499	0.5258	0.3093	0.3190
11	0.6580	0.3389	0.0311	0.2039	0.2194	0.3123	0.5488	0.2962	0.4570	0.3107	0.2737
12	0.6188	0.3500	0.0330	0.2325	0.2210	0.3545	0.5631	0.3234	0.4385	0.3377	0.3979
13	0.5962	0.3646	0.0445	0.1974	0.2520	0.3625	0.5190	0.3845	0.3975	0.3243	0.4238
14	0.5667	0.4228	0.0838	0.2966	0.2144	0.3982	0.5840	0.3820	0.3888	0.3332	0.3759
15	0.5182	0.4203	0.0769	0.2913	0.2743	0.3475	0.5718	0.3503	0.3934	0.2458	0.4721
16	0.5258	0.4541	0.0739	0.3100	0.2605	0.3675	0.5771	0.3702	0.3955	0.5313	0.4886
17	0.5083	0.4543	0.0734	0.2639	0.2562	0.3466	0.5556	0.4527	0.3480	0.5504	0.4181
18	0.6990	0.3759	0.0719	0.3700	0.2500	0.4166	0.5405	0.3676	0.4015	0.5249	0.3744
19	0.8045	0.6053	0.0574	0.3451	0.2456	0.3477	0.6096	0.4303	0.3585	0.4953	0.3442
20	0.7941	0.5558	0.0562	0.2603	0.3199	0.3930	0.5974	0.4461	0.4100	0.4806	0.4072
21	0.7583	0.3318	0.0529	0.3103	0.3923	0.4090	0.5898	0.4250	0.4047	1.0259	0.3808
22	0.7516	0.4784	0.0561	0.3097	0.3545	0.4084	0.7009	0.4711	0.4088	1.1171	0.3453
23	0.7368	0.4679	0.0558	0.3049	0.3580	0.5734	0.7038	0.4364	0.4273	1.0568	0.3513
24	0.7275	0.4260	0.0994	0.3064	0.3429	0.5553	0.7443	0.4355	0.4085	1.0136	0.3235
25	0.7289	0.4075	0.0816	0.2876	0.3317	0.5408	0.8941	0.3967	0.4030	0.9731	0.2992

Table E.36: *Basic* validity measure for natural images using *regression* context values only.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.0291	0.1440	0.0208	0.2866	0.1130	0.1637	0.0606	0.3547	0.1628	0.0515	0.1070
3	0.0832	0.1818	0.0915	0.0995	0.0783	0.2810	0.1123	0.2283	0.2583	0.1169	0.0961
4	0.3230	0.2066	0.0434	0.1588	0.1066	0.2492	0.1695	0.2728	0.2552	0.0707	0.1167
5	0.3877	0.3318	0.0559	0.1588	0.0825	0.3094	0.1477	0.3805	0.2614	0.0736	0.1818
6	0.2663	0.2861	0.0488	0.1428	0.1104	0.4473	0.5100	0.3739	0.2526	0.0547	0.1304
7	0.2390	0.2560	0.0298	0.1840	0.1512	0.5466	0.5337	0.4551	0.3093	0.0573	0.2396
8	0.2062	0.5478	0.0548	0.1459	0.3692	0.4574	0.7071	0.5574	0.4691	0.0780	0.1987
9	0.3337	0.5069	0.0504	0.2782	0.3250	0.5015	0.7118	0.5538	0.4522	0.1084	0.3988
10	0.4129	0.5282	0.0350	0.2425	0.2854	0.5502	0.6719	0.4963	0.4411	0.3442	0.4480
11	0.3847	0.5858	0.0360	0.2281	0.2556	0.5479	0.8104	0.6036	0.4825	0.3489	0.3924
12	0.3761	0.5630	0.0656	0.2319	0.2971	0.5320	0.8270	0.6046	0.3656	0.3787	0.5367
13	0.8445	0.4885	0.1527	0.2558	0.2772	0.5700	0.8090	0.6041	0.4966	0.3886	0.4892
14	0.7979	0.4759	0.1838	0.1979	0.3355	0.5265	0.7755	0.5291	0.4585	0.3910	0.4363
15	0.7670	0.4683	0.1876	0.3045	0.3568	0.5306	0.7399	0.4978	0.5908	0.6422	0.5020
16	0.7533	0.5643	0.1649	0.3065	0.3366	0.6145	0.9988	0.5405	0.5893	0.3612	0.5034
17	0.8101	0.5395	0.1562	0.3164	0.3258	0.5595	0.9845	0.5111	0.5386	0.3449	0.5058
18	0.8013	0.5266	0.1364	0.3130	0.3156	0.5240	0.9014	0.5871	0.4620	0.6746	0.4597
19	0.7973	0.5195	0.1349	0.3124	0.3130	0.5262	0.8635	0.5494	0.4900	0.6442	0.4659
20	0.7911	0.5441	0.1337	0.3061	0.3525	0.6036	0.8882	0.5490	0.4889	0.8385	0.4323
21	0.7631	0.5289	0.1329	0.3037	0.3981	0.5959	0.8595	0.5284	0.5131	0.8165	0.4017
22	0.7427	0.5154	0.1172	0.3015	0.4884	0.5697	0.8483	0.5213	0.5018	0.7812	0.4000
23	0.7235	0.5119	0.1202	0.4185	0.4909	0.5192	0.8680	0.5389	0.4994	0.7667	0.5415
24	0.7140	0.5860	0.1129	0.4181	0.4859	0.6011	1.0433	0.5293	0.5053	0.7450	0.5364
25	0.7085	0.5796	0.1122	0.4305	0.4480	0.5878	1.0313	0.5466	0.5019	0.7311	0.6318

Table E.37: *Basic* validity measure for natural images using original and *mean* context values.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.0276	0.1418	0.0208	0.2841	0.1114	0.1626	0.0576	0.3543	0.1606	0.0516	0.1067
3	0.0769	0.1802	0.0853	0.0980	0.0744	0.2781	0.1120	0.2394	0.2353	0.1183	0.0956
4	0.3150	0.1990	0.0388	0.1525	0.1037	0.2464	0.1647	0.2762	0.2507	0.0699	0.1152
5	0.3030	0.3223	0.0516	0.1584	0.0875	0.3030	0.1401	0.3954	0.2560	0.0750	0.1817
6	0.2656	0.2814	0.0454	0.1391	0.1009	0.4326	0.5289	0.4102	0.2497	0.0553	0.1304
7	0.2161	0.2765	0.0388	0.1796	0.1413	0.5235	0.5373	0.4746	0.2933	0.0581	0.2682
8	0.3707	0.3220	0.0216	0.1381	0.4478	0.4390	0.5991	0.5698	0.4599	0.0776	0.3601
9	0.3402	0.3969	0.0197	0.2648	0.3650	0.5337	0.6101	0.5396	0.4693	0.1094	0.3797
10	0.3815	0.5044	0.0282	0.2378	0.3127	0.4994	0.5867	0.6763	0.4595	0.1236	0.4240
11	0.3678	0.5883	0.0326	0.2555	0.3017	0.4617	0.7482	0.6345	0.4445	0.3256	0.3774
12	0.8162	0.5771	0.0282	0.2654	0.2860	0.4179	0.7242	0.6293	0.4294	0.3406	0.5154
13	0.7593	0.5487	0.0363	0.2401	0.2783	0.5405	0.6905	0.6106	0.4245	0.3322	0.4563
14	0.7424	0.5285	0.0343	0.2263	0.2633	0.5359	0.6567	0.5480	0.5342	0.3373	0.4050
15	0.7101	0.6197	0.0333	0.2208	0.2614	0.5618	0.6685	0.5399	0.4322	0.3476	0.4369
16	0.6797	0.6120	0.0350	0.2176	0.3200	0.5576	0.6402	0.5501	0.5356	0.3657	0.3812
17	0.6648	0.5746	0.0348	0.2146	0.3245	0.5722	0.6251	0.5501	0.5326	0.3557	0.4466
18	0.6468	0.5633	0.0348	0.3099	0.3515	0.5663	0.5956	0.6793	0.5374	0.3671	0.4038
19	0.6690	0.5307	0.0347	0.2821	0.3493	0.5658	0.5888	0.7046	0.5166	0.3635	0.4182
20	0.6572	0.5107	0.0347	0.2776	0.3532	0.5130	0.5828	0.6855	0.4514	0.4894	0.3859
21	0.6457	0.5208	0.0355	0.2749	0.3509	0.5042	0.5797	0.6649	0.4496	0.4847	0.3642
22	0.6427	0.5145	0.0505	0.2722	0.3498	0.4941	0.5676	0.7591	0.6640	0.4829	0.3675
23	0.6244	0.5113	0.0955	0.2712	0.3335	0.4716	0.8193	0.7758	0.8882	0.6365	0.3480
24	0.6118	0.5125	0.1062	0.2700	0.3321	0.4603	0.8186	0.7360	0.9700	0.6241	0.5722
25	0.6024	0.5565	0.1061	0.2694	0.5566	0.4875	0.7950	0.6954	0.9610	0.9735	0.5238

Table E.38: *Basic* validity measure for natural images using original and *median* context values.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.0290	0.1438	0.0206	0.2864	0.1128	0.1634	0.0604	0.3540	0.1626	0.0515	0.1069
3	0.0811	0.1815	0.0918	0.0995	0.0779	0.2811	0.1117	0.2280	0.2573	0.1169	0.0961
4	0.3211	0.2052	0.0434	0.1584	0.1060	0.2486	0.1641	0.2712	0.2541	0.0706	0.1161
5	0.3881	0.3295	0.0556	0.1585	0.0861	0.3083	0.1413	0.3784	0.2600	0.0737	0.1813
6	0.2630	0.2851	0.0488	0.1425	0.1091	0.4452	0.4995	0.3717	0.2510	0.0545	0.1303
7	0.2367	0.2532	0.0312	0.1830	0.1509	0.5200	0.5253	0.4527	0.3059	0.0574	0.2387
8	0.2045	0.5418	0.0560	0.1445	0.3724	0.4569	0.6915	0.5584	0.4639	0.0778	0.1981
9	0.3306	0.5031	0.0517	0.2772	0.3250	0.4994	0.6933	0.5064	0.4497	0.1079	0.3961
10	0.4059	0.5056	0.0358	0.2424	0.2817	0.5511	0.6748	0.4829	0.4352	0.3433	0.4469
11	0.3814	0.5846	0.0350	0.2271	0.2520	0.5401	0.8039	0.6007	0.4754	0.3492	0.4912
12	0.8650	0.5598	0.1585	0.2288	0.2896	0.5304	0.8206	0.6051	0.3628	0.3793	0.5380
13	0.8289	0.5418	0.1836	0.2512	0.3549	0.5696	0.8002	0.6017	0.4280	0.3853	0.4872
14	0.9066	0.4828	0.1883	0.1931	0.3661	0.5213	0.7854	0.5312	0.4529	0.3876	0.4350
15	0.9133	0.4663	0.1656	0.3037	0.3494	0.5234	0.7589	0.5319	0.5863	0.6325	0.5721
16	0.8551	0.5724	0.1632	0.3002	0.3452	0.6185	0.9765	0.5423	0.5859	0.3571	0.5730
17	0.8205	0.5537	0.1550	0.2512	0.3276	0.6107	0.9770	0.5057	0.5318	0.3396	0.5012
18	1.1947	0.5405	0.1348	0.2540	0.3096	0.5090	0.9488	0.5803	0.4550	0.6603	0.4548
19	1.2467	0.5331	0.1335	0.2497	0.3071	0.5106	0.8830	0.5434	0.4799	0.6286	0.4218
20	1.2102	0.5403	0.1324	0.2472	0.3472	0.5041	0.8663	0.5469	0.4751	0.8309	0.4278
21	1.1681	0.5332	0.1319	0.2465	0.3429	0.5888	0.8412	0.5631	0.5047	0.8082	0.4051
22	1.1389	0.5110	0.1312	0.2444	0.3887	0.5738	0.8300	0.5541	0.4952	0.7732	0.3742
23	1.1092	0.5054	0.1303	0.2420	0.4874	0.6183	0.8632	0.5531	0.5026	0.7511	0.3708
24	1.0872	0.5013	0.1225	0.3324	0.4882	0.6037	1.0161	0.7072	0.5092	0.7307	0.5451
25	1.0761	0.4996	0.1143	0.3310	0.4500	0.5993	0.9866	0.6646	0.7174	0.7527	0.6026

Table E.39: *Basic* validity measure for natural images using original and *weighted* context values.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.0280	0.1429	0.0199	0.2852	0.1121	0.1653	0.0576	0.3546	0.1622	0.0514	0.1069
3	0.0777	0.1800	0.0668	0.0994	0.0765	0.2875	0.1121	0.2284	0.2430	0.1173	0.0960
4	0.3207	0.1984	0.0392	0.1547	0.1034	0.2517	0.1152	0.2626	0.2508	0.0701	0.1145
5	0.3892	0.3256	0.0529	0.1589	0.0875	0.2313	0.1129	0.3648	0.2559	0.0733	0.1834
6	0.2584	0.2799	0.0438	0.1409	0.1054	0.2819	0.1356	0.3638	0.2482	0.0553	0.1275
7	0.2268	0.2448	0.0266	0.1886	0.1515	0.4253	0.4534	0.4438	0.2971	0.0567	0.2222
8	0.3592	0.4127	0.0264	0.1445	0.4380	0.4578	0.4704	0.5421	0.4889	0.0780	0.1763
9	0.3230	0.4898	0.0342	0.2708	0.3724	0.4673	0.6325	0.5197	0.4001	0.1078	0.3785
10	0.7808	0.5188	0.0346	0.2362	0.3247	0.4237	0.6206	0.4785	0.5783	0.3389	0.4331
11	0.7807	0.5706	0.1529	0.2191	0.2891	0.5110	0.7274	0.5401	0.5729	0.3512	0.3742
12	0.7481	0.5499	0.1356	0.2148	0.2722	0.5206	0.6689	0.4651	0.5769	0.3585	0.5162
13	0.7022	0.5501	0.1299	0.2446	0.2837	0.5405	0.8268	0.5953	0.5353	0.3945	0.4584
14	0.7233	0.4605	0.1817	0.2475	0.2932	0.5097	0.8011	0.5284	0.4672	0.3991	0.4064
15	0.6987	0.4376	0.2035	0.2290	0.3624	0.5071	0.8732	0.5616	0.5178	0.3207	0.5503
16	0.6739	0.4446	0.1893	0.2268	0.3776	0.5237	0.8437	0.4963	0.5082	0.3104	0.5496
17	0.9549	0.4350	0.1837	0.3296	0.3714	0.5552	0.8372	0.5550	0.5064	0.6615	0.4773
18	0.9344	0.4495	0.1766	0.3277	0.3423	0.5737	0.8531	0.5443	0.4409	0.6374	0.4307
19	1.1122	0.4346	0.1715	0.2945	0.3740	0.5680	0.8478	0.7546	0.4421	0.8185	0.3978
20	1.1036	0.5317	0.1675	0.2932	0.3736	0.5562	0.8484	0.7262	0.4709	0.7755	0.3989
21	1.0646	0.5238	0.1661	0.2911	0.5486	0.5446	0.8469	0.7093	0.4607	0.7530	0.3765
22	1.0388	0.4966	0.1663	0.2937	0.5366	0.5398	0.9644	0.6985	0.6141	0.7209	0.3441
23	1.0189	0.4908	0.1656	0.2978	0.6246	0.5388	0.9391	0.7589	0.7767	0.7027	0.3239
24	0.9952	0.4856	0.1609	0.2950	0.6061	0.5454	0.9234	0.7365	0.7945	0.6856	0.3515
25	0.9644	0.4981	0.1361	0.2942	0.5758	0.5957	0.9228	0.7257	0.8403	0.6655	0.3298

Table E.40: *Basic* validity measure for natural images using original and *regression* context values.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.0262	0.1279	0.0144	0.3181	0.1053	0.1558	0.0466	0.1602	0.1219	0.0515	0.0734
3	0.0568	0.1308	0.0515	0.1216	0.0706	0.2280	0.0790	0.2356	0.2207	0.1190	0.0883
4	0.2509	0.1303	0.0278	0.1396	0.1040	0.1843	0.0990	0.3368	0.1180	0.0664	0.1251
5	0.2468	0.2292	0.0240	0.2320	0.1334	0.1713	0.1449	0.2862	0.2225	0.0678	0.1381
6	0.1913	0.2900	0.0260	0.1761	0.1203	0.2439	0.1832	0.2616	0.2439	0.0548	0.1340
7	0.1857	0.3034	0.0240	0.2162	0.1231	0.2334	0.1701	0.2778	0.2798	0.0536	0.2565
8	0.1596	0.2774	0.0294	0.2051	0.2595	0.4005	0.2605	0.2967	0.4571	0.0698	0.2153
9	0.2207	0.2749	0.0252	0.1966	0.2151	0.3984	0.2486	0.3528	0.3360	0.0719	0.2195
10	0.2051	0.2772	0.0325	0.1905	0.2817	0.3936	0.2384	0.4031	0.3238	0.1090	0.2519
11	0.2046	0.4100	0.0296	0.1702	0.2337	0.3952	0.2330	0.3753	0.2408	0.1631	0.2204
12	0.2016	0.3848	0.0243	0.2863	0.1910	0.3799	0.2242	0.4170	0.2388	0.1536	0.2087
13	0.1964	0.3936	0.0509	0.2793	0.2453	0.3712	0.2121	0.4385	0.2883	0.1439	0.3377
14	0.1924	0.3351	0.0461	0.2336	0.2949	0.3648	0.3255	0.4139	0.2574	0.2161	0.2140
15	0.3592	0.3164	0.0432	0.2266	0.2578	0.3632	0.3176	0.4675	0.3490	0.2093	0.2632
16	0.3579	0.2745	0.0389	0.2189	0.3056	0.4456	0.3122	0.4424	0.2683	0.2557	0.2557
17	0.3236	0.2732	0.0467	0.1711	0.2979	0.4076	0.3455	0.4102	0.2660	0.4132	0.2223
18	0.3173	0.2638	0.0344	0.1613	0.2878	0.3323	0.3167	0.4261	0.3260	0.3631	0.1925
19	0.3154	0.2522	0.0325	0.1595	0.2798	0.2762	0.3136	0.7625	0.4039	0.3406	0.2301
20	0.3072	0.3546	0.0316	0.1450	0.2497	0.5996	0.2993	0.6769	0.3968	0.3294	0.2270
21	0.3668	0.3382	0.0313	0.1413	0.2327	0.5355	0.3250	0.6467	0.3869	0.3121	0.2085
22	0.3611	0.6425	0.0263	0.1806	0.2949	0.5578	0.3960	0.6373	0.3504	0.2999	0.1919
23	0.2210	0.6478	0.0260	0.1803	0.4090	0.5547	0.3847	0.6081	0.4683	0.2726	0.1796
24	0.2129	0.6431	0.0290	0.1782	0.4023	0.5128	0.3757	0.5942	0.4468	0.2624	0.1931
25	0.2455	0.5729	0.0313	0.1255	0.6520	0.5363	0.3093	0.5708	0.4397	0.2601	0.2753

Table E.41: *Basic* validity measure for natural images using XYZ colour space.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.0277	0.1393	0.0189	0.0660	0.1110	0.1671	0.0532	0.2959	0.1587	0.0519	0.1068
3	0.0771	0.1755	0.0607	0.3002	0.0732	0.2786	0.1071	0.2049	0.2465	0.1160	0.0960
4	0.3074	0.1906	0.0403	0.3000	0.1038	0.2415	0.1080	0.2180	0.2409	0.0690	0.1186
5	0.2525	0.2734	0.0304	0.1283	0.1463	0.2199	0.0886	0.2857	0.2512	0.0656	0.1870
6	0.2683	0.2638	0.0314	0.0943	0.0966	0.2568	0.2645	0.3086	0.2234	0.0575	0.1396
7	0.2216	0.2588	0.0200	0.0869	0.1145	0.3078	0.1235	0.4172	0.2842	0.0653	0.2644
8	0.2143	0.2620	0.0177	0.2681	0.3287	0.3280	0.1441	0.4104	0.3016	0.0664	0.3326
9	0.2317	0.2601	0.0331	0.2159	0.2537	0.3663	0.2008	0.4262	0.3196	0.1095	0.3394
10	0.2234	0.2475	0.0286	0.1432	0.2193	0.3658	0.3153	0.3586	0.3684	0.0982	0.2562
11	0.2107	0.2213	0.0232	0.1441	0.3023	0.3605	0.3056	0.3246	0.3401	0.1868	0.2877
12	0.2051	0.3160	0.0229	0.1375	0.2583	0.3682	0.3021	0.3751	0.3270	0.2107	0.2032
13	0.2575	0.3014	0.0231	0.1616	0.2609	0.3664	0.3261	0.4327	0.3217	0.2770	0.1988
14	0.2728	0.2848	0.0556	0.3259	0.2531	0.3342	0.2983	0.3279	0.3332	0.2580	0.1603
15	0.2503	0.4131	0.0514	0.3197	0.2471	0.3741	0.2993	0.4985	0.3283	0.2471	0.1912
16	0.2454	0.4068	0.0476	0.3128	0.2443	0.3354	0.2840	0.4729	0.3206	0.3229	0.1996
17	0.2424	0.3954	0.0445	0.2960	0.2349	0.6444	0.2700	0.5136	0.4265	0.3078	0.2095
18	0.2274	0.3843	0.0403	0.2964	0.2258	0.6366	0.2617	0.4373	0.3831	0.2837	0.2453
19	0.2234	0.3817	0.0392	0.2189	0.5147	0.5854	0.2604	0.4184	0.3817	0.3967	0.2266
20	0.2114	0.3807	0.0380	0.2156	0.2358	0.5817	0.2517	0.3893	0.5397	0.3790	0.2311
21	0.2829	0.3803	0.0376	0.2137	0.2246	0.5495	0.3136	0.6212	0.5742	0.3781	0.2802
22	0.2793	0.4160	0.0530	0.2071	0.2192	0.4597	0.3117	0.6063	0.5995	0.3772	0.2697
23	0.2774	0.3596	0.0549	0.2035	0.2172	0.7577	0.3102	0.6006	0.5954	0.3714	0.3622
24	0.2761	0.3436	0.0673	0.3889	0.2191	0.7597	0.3030	0.5619	0.5875	0.3273	0.3438
25	0.2692	0.4357	0.0646	0.2815	0.4325	0.7075	0.3006	0.5557	0.4717	0.4831	0.3054

Table E.42: *Basic* validity measure for natural images using KL colour space.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.0908	0.0416	0.0725	0.2950	0.3337	0.2767	0.2100	0.5004	0.1111	0.1048	0.2211
3	0.0950	0.1691	0.0148	0.1554	0.2231	0.1653	0.1388	0.3076	0.2882	0.0944	0.2147
4	0.0981	0.3593	0.0155	0.2775	0.4221	0.1554	0.1780	0.3693	0.2343	0.2007	0.1878
5	0.1016	0.2265	0.0169	0.2721	0.3302	0.1486	0.2313	0.4971	0.2352	0.1587	0.1589
6	0.1360	0.4700	0.0397	0.1596	0.2338	0.1557	0.3420	0.4030	0.3030	0.1557	0.1535
7	0.1204	0.4564	0.0385	0.1286	0.2482	0.3831	0.4719	0.3920	0.2614	0.1541	0.1330
8	0.4851	0.4519	0.0469	0.1132	0.1842	0.3134	0.2452	0.4134	0.2768	0.1478	0.1277
9	0.4644	0.2837	0.0547	0.1405	0.2418	0.3042	0.3460	0.2987	0.3070	0.1233	0.1699
10	0.4548	0.3182	0.0260	0.1276	0.3306	0.1998	0.3579	0.2665	0.3056	0.2016	0.1746
11	0.4386	0.3106	0.0384	0.1233	0.3355	0.4796	0.3528	0.3678	0.2879	0.1853	0.2140
12	0.5642	0.3061	0.0375	0.1113	0.4316	0.4780	0.3374	0.3515	0.2756	0.2081	0.1671
13	0.5359	0.2881	0.0373	0.1058	0.4316	0.4733	0.3063	0.3263	0.2150	0.1972	0.1465
14	0.4777	0.3048	0.0496	0.0995	0.3304	0.4703	0.2912	0.3184	0.3051	0.1979	0.1678
15	0.5058	0.2912	0.0493	0.1262	0.3115	0.5887	0.2792	0.2972	0.3572	0.2748	0.1638
16	0.4838	0.3548	0.0325	0.1200	0.2757	0.5705	0.2704	0.2611	0.4757	0.3325	0.1742
17	0.4774	0.3440	0.0323	0.1173	0.2684	0.5110	0.2635	0.2586	0.4559	0.3208	0.1612
18	0.6259	0.4580	0.0319	0.1158	0.2424	0.5064	0.2575	0.2508	0.6632	0.3197	0.2234
19	0.6196	0.4125	0.0505	0.1916	0.2367	0.4361	0.2550	0.3808	0.5818	0.2978	0.2184
20	0.6012	0.5647	0.0488	0.1850	0.2277	0.4432	0.2530	0.4198	0.5627	0.2884	0.2084
21	0.6381	0.4773	0.0486	0.1820	0.2179	0.3778	0.2490	0.4125	0.5128	0.2675	0.1893
22	0.5244	0.4406	0.0481	0.1790	0.2139	0.3317	0.2452	0.3844	0.4819	0.2613	0.1866
23	0.4801	0.4246	0.0469	0.2322	0.2079	0.3304	0.2953	0.3808	0.4610	0.2558	0.2115
24	0.4660	0.4158	0.0456	0.2246	0.2039	0.3299	0.4035	0.3708	0.4483	0.2546	0.1818
25	0.4549	0.4068	0.0455	0.2223	0.2371	0.3300	0.5348	0.4358	0.4286	0.2541	0.1758

Table E.43: *Basic* validity measure for natural images using HSV colour space.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.3374	1.5132	0.2331	3.1032	1.2418	1.6200	0.6351	3.5484	1.7309	0.5641	1.1582
3	0.4685	1.1903	0.3930	0.6763	0.5446	1.6614	0.6936	1.2947	1.8273	0.8071	0.6601
4	0.6831	0.4083	0.0935	0.3804	0.2325	0.6801	0.3221	0.5368	0.5423	0.1622	0.2672
5	0.2718	0.2950	0.0397	0.1569	0.0868	0.2763	0.2298	0.3261	0.2494	0.0768	0.1853
6	0.2119	0.2283	0.0391	0.2005	0.0987	0.3884	0.2983	0.2924	0.2159	0.0502	0.1596
7	0.1759	0.2163	0.0257	0.1745	0.1187	0.3994	0.3390	0.3272	0.2597	0.0524	0.1867

Table E.44: *Modified* validity measure for natural images using *mean* context values only with $c = 25$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.2989	1.4610	0.2101	3.0805	1.2078	1.5946	0.5560	3.2320	1.6559	0.5666	1.1570
3	0.5188	1.1555	0.4438	0.6715	0.5091	1.5910	0.6845	1.4048	1.6756	0.8288	0.6560
4	0.6765	0.4027	0.0830	0.3506	0.2279	0.6222	0.1852	0.4935	0.5227	0.1569	0.2652
5	0.3646	0.2936	0.0502	0.1651	0.0886	0.2486	0.1009	0.3261	0.2456	0.0805	0.1929
6	0.2415	0.2616	0.0319	0.1324	0.0919	0.3391	0.1959	0.2628	0.2064	0.0539	0.1573
7	0.1876	0.2314	0.0208	0.1701	0.1072	0.3582	0.2285	0.2757	0.2330	0.0553	0.1895

Table E.45: *Modified* validity measure for natural images using *median* context values only with $c = 25$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.3197	1.5132	0.2337	3.1025	1.2383	1.6243	0.6329	3.5378	1.7307	0.5597	1.1581
3	0.5113	1.1878	0.4018	0.6785	0.5427	1.6757	0.6995	1.2943	1.8294	0.8024	0.6610
4	0.7104	0.4101	0.0928	0.3809	0.2343	0.6811	0.3286	0.5334	0.5433	0.1604	0.2670
5	0.2867	0.2943	0.0394	0.1574	0.0848	0.2769	0.2194	0.3270	0.2496	0.0776	0.1860
6	0.2250	0.2281	0.0385	0.2009	0.0977	0.3916	0.2989	0.2961	0.2161	0.0509	0.1602
7	0.1755	0.2172	0.0259	0.1737	0.1185	0.3925	0.3434	0.3247	0.2593	0.0511	0.1857

Table E.46: *Modified* validity measure for natural images using *weighted* context values only with $c = 25$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.3107	1.5088	0.2210	3.0922	1.2241	1.6841	0.6259	3.5937	1.7361	0.5634	1.1671
3	0.5226	1.1808	0.5307	0.6846	0.5308	1.8248	0.6968	1.3187	1.7558	0.8141	0.6687
4	0.7103	0.4138	0.0885	0.3623	0.2327	0.5327	0.2405	0.5276	0.5507	0.1626	0.2669
5	0.2615	0.3017	0.0571	0.1632	0.0906	0.2789	0.1380	0.3309	0.2577	0.0790	0.1953
6	0.2000	0.2334	0.0382	0.2023	0.0961	0.3822	0.3709	0.2900	0.2226	0.0523	0.1308
7	0.2037	0.2073	0.0245	0.1738	0.1278	0.4483	0.3567	0.3223	0.2692	0.0534	0.2163

Table E.47: *Modified* validity measure for natural images using *regression* context values only with $c = 25$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.3195	1.5797	0.2279	3.1446	1.2401	1.7967	0.6647	3.8924	1.7868	0.5647	1.1737
3	0.5867	1.2815	0.6451	0.7014	0.5522	1.9808	0.7913	1.6093	1.8211	0.8238	0.6777
4	0.7589	0.4856	0.1020	0.3731	0.2505	0.5855	0.3982	0.6411	0.5997	0.1662	0.2741
5	0.4307	0.3686	0.0621	0.1764	0.0916	0.3437	0.1640	0.4226	0.2903	0.0818	0.2020
6	0.2672	0.2871	0.0490	0.1433	0.1107	0.4488	0.5117	0.3751	0.2535	0.0549	0.1309
7	0.2390	0.2560	0.0298	0.1840	0.1512	0.5466	0.5337	0.4551	0.3094	0.0573	0.2396

Table E.48: *Modified* validity measure for natural images using original and *mean* context values with $c = 25$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.3030	1.5565	0.2282	3.1180	1.2227	1.7846	0.6321	3.8881	1.7628	0.5663	1.1711
3	0.5423	1.2706	0.6010	0.6911	0.5243	1.9601	0.7895	1.6877	1.6589	0.8342	0.6740
4	0.7401	0.4675	0.0911	0.3584	0.2437	0.5790	0.3871	0.6490	0.5890	0.1642	0.2706
5	0.3366	0.3580	0.0573	0.1759	0.0971	0.3366	0.1556	0.4392	0.2844	0.0833	0.2018
6	0.2664	0.2823	0.0456	0.1396	0.1013	0.4340	0.5307	0.4116	0.2505	0.0555	0.1309
7	0.2161	0.2765	0.0388	0.1796	0.1413	0.5236	0.5374	0.4746	0.2933	0.0581	0.2682

Table E.49: *Modified* validity measure for natural images using original and *median* context values with $c = 25$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.3178	1.5775	0.2262	3.1432	1.2380	1.7934	0.6631	3.8843	1.7840	0.5646	1.1730
3	0.5720	1.2793	0.6471	0.7011	0.5494	1.9816	0.7871	1.6072	1.8139	0.8242	0.6775
4	0.7546	0.4822	0.1020	0.3721	0.2491	0.5841	0.3856	0.6373	0.5971	0.1660	0.2727
5	0.4311	0.3661	0.0618	0.1761	0.0957	0.3425	0.1570	0.4203	0.2888	0.0819	0.2014
6	0.2639	0.2861	0.0490	0.1430	0.1095	0.4467	0.5011	0.3729	0.2518	0.0547	0.1307
7	0.2367	0.2532	0.0312	0.1830	0.1509	0.5200	0.5253	0.4528	0.3059	0.0574	0.2387

Table E.50: *Modified* validity measure for natural images using original and *weighted* context values with $c = 25$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.3074	1.5680	0.2180	3.1301	1.2304	1.8144	0.6320	3.8918	1.7797	0.5640	1.1729
3	0.5476	1.2688	0.4706	0.7008	0.5395	2.0269	0.7899	1.6103	1.7127	0.8270	0.6767
4	0.7535	0.4663	0.0922	0.3635	0.2429	0.5914	0.2708	0.6170	0.5893	0.1647	0.2691
5	0.4323	0.3617	0.0588	0.1765	0.0972	0.2569	0.1254	0.4052	0.2842	0.0814	0.2038
6	0.2593	0.2808	0.0440	0.1414	0.1057	0.2829	0.1361	0.3650	0.2490	0.0555	0.1280
7	0.2268	0.2448	0.0266	0.1886	0.1516	0.4253	0.4534	0.4438	0.2971	0.0567	0.2223

Table E.51: *Modified* validity measure for natural images using original and *regression* context values with $c = 25$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.2879	1.4034	0.1579	3.4912	1.1551	1.7092	0.5118	1.7580	1.3381	0.5650	0.8060
3	0.4006	0.9222	0.3631	0.8574	0.4977	1.6074	0.5572	1.6607	1.5554	0.8391	0.6225
4	0.5895	0.3063	0.0654	0.3281	0.2445	0.4331	0.2325	0.7915	0.2774	0.1561	0.2940
5	0.2741	0.2546	0.0267	0.2577	0.1481	0.1902	0.1609	0.3180	0.2472	0.0753	0.1534
6	0.1920	0.2910	0.0261	0.1766	0.1207	0.2447	0.1839	0.2624	0.2448	0.0550	0.1344
7	0.1857	0.3034	0.0240	0.2162	0.1231	0.2334	0.1701	0.2778	0.2798	0.0536	0.2565

Table E.52: *Modified* validity measure for natural images using XYZ colour space with $c = 25$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.3044	1.5281	0.2079	0.7243	1.2186	1.8337	0.5843	3.2473	1.7415	0.5694	1.1716
3	0.5431	1.2371	0.4280	2.1158	0.5160	1.9641	0.7551	1.4441	1.7374	0.8180	0.6765
4	0.7223	0.4480	0.0948	0.7050	0.2440	0.5675	0.2538	0.5122	0.5660	0.1621	0.2788
5	0.2804	0.3036	0.0337	0.1425	0.1625	0.2443	0.0984	0.3174	0.2790	0.0729	0.2077
6	0.2692	0.2647	0.0315	0.0946	0.0969	0.2577	0.2654	0.3097	0.2241	0.0577	0.1401
7	0.2216	0.2588	0.0200	0.0869	0.1145	0.3078	0.1235	0.4172	0.2842	0.0653	0.2644

Table E.53: *Modified* validity measure for natural images using KL colour space with $c = 25$.

K	*balls*	*Lenna*	*molecule*	*teapot*	*ant*	*blond*	*jet*	*mandrill*	*peppers*	*mouse*	*rose*
2	0.9968	0.4562	0.7957	3.2377	3.6621	3.0360	2.3044	5.4913	1.2195	1.1506	2.4258
3	0.6695	1.1921	0.1045	1.0958	1.5726	1.1650	0.9787	2.1683	2.0319	0.6656	1.5134
4	0.2305	0.8443	0.0363	0.6520	0.9918	0.3652	0.4183	0.8678	0.5506	0.4717	0.4413
5	0.1128	0.2516	0.0187	0.3023	0.3667	0.1650	0.2569	0.5522	0.2613	0.1763	0.1766
6	0.1365	0.4716	0.0398	0.1601	0.2346	0.1562	0.3432	0.4043	0.3040	0.1562	0.1540
7	0.1204	0.4565	0.0385	0.1286	0.2482	0.3831	0.4719	0.3920	0.2614	0.1541	0.1330

Table E.54: *Modified* validity measure for natural images using HSV colour space with $c = 25$.

Appendix F

Publications

[1] R.H. Turi and S. Ray, Determination of the Number of clusters in Colour Image Segmentation, *IEEE Trans. Syst., Man, Cybern.* (Communicated).

[2] R.H. Turi and S. Ray, An Application of Clustering in Colour Image Segmentation, *Proc. Sixth International Conference of Control, Automation, Robotics and Vision (ICARCV2000), Singapore*, Paper No. 349, Session TA1, 6 pages, December 5-8, 2000.

[3] S. Ray and R.H. Turi, Determination of Number of Clusters in K-means Clustering and Application in Colour Image Segmentation, *Proc. 4th International Conference on Advances in Pattern Recognition and Digital Techniques*, Calcutta, India, pp. 137-143, December 27-29, 1999.

[4] R.H. Turi and S. Ray, K-means clustering for colour image segmentation with automatic detection of K, *Proc. IASTED International Conference, Signal and Image Processing (SIP'98), Las Vegas, Nevada, USA*, pp. 345-349, October 28-31, 1998.

[5] R.H. Turi and S. Ray, Clustering-based colour image segmentation using inter-

cluster distance, *Proc. Australian Pattern Recognition Society (Victorian Branch) Student Conference on Image Analysis, Depts Physics & Computer Science, Monash University, Melbourne, Australia*, pp. 37-48, October 3, 1997.

[6] R.H. Turi and S. Ray, A new approach to clustering-based colour image segmentation, *Proc. IASTED International Conference, Signal and Image Processing (SIP-96), Orlando, Florida, USA*, pp. 345-349, November 11-14, 1996.

[7] R.H. Turi and S. Ray, Clustering-based colour image segmentation, *Proc. Australian Pattern Recognition Society Student Conference, Victoria University of Technology, Melbourne, Australia*, pp. 78-88, October 4, 1996. (published in 1997).

[8] S. Ray, R.H. Turi and P.E. Tischer, Clustering-based colour image segmentation: An evaluation study, *Proc. DICTA-95 conference, Brisbane, Australia*, pp. 86-92, December 4-6, 1995.

Bibliography

[AM96] A.J. Abrantes and J.S. Marques, A Class of Constrained Clustering Algorithms for Object Boundary Extraction, *IEEE Trans. Image Processing*, vol. 5, no. 11, pp. 1507-1521, 1996.

[AK88] M. Amadasun and R.A. King, Low-level segmentation of multispectral images via agglomerative clustering of uniform neighbourhoods, *Pattern Recognition*, vol. 21, no. 3, pp. 261-268, 1988.

[And73] M.R. Anderberg, *Cluster Analysis for Applications*, New York: Academic Press, 1973.

[And70] H.C. Andrews, *Computer Techniques in Image Processing*, New York: Academic Press, 1970.

[AHDS96] P. Arabie, L.J. Hubert and G. De Soete (Eds.), *Clustering and Classification*, Singapore: World Scientific, 1996.

[BJ81] E. Backer and A.K. Jain, A clustering performance measure based on fuzzy set decomposition, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 3, no. 1, pp. 66-75, 1981.

[BJLS98] S.J. Baek, B.K. Jeon, D. Lee and K.-M. Sung, Fast clustering algorithm for vector quantisation, *Electronics Letters*, vol. 34, no. 2, pp. 151-152, 1998.

[BA98] P. Bajcsy and N. Ahuja, Location- and Density-Based Hierarchical Clustering Using Similarity Analysis, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 9, pp. 1011-1015, 1998.

[BH67] G.H. Ball and D.J. Hall, A clustering technique for summarizing multivariate data, *Behavioral Science*, vol. 12, pp. 153-155, 1967.

[BB99a] A. Baraldi and P. Blonda, A Survey of Fuzzy Clustering Algorithms for Pattern Recognition - Part I, *IEEE Trans. Syst., Man, Cybern.*, vol.29, no. 6, pp. 778-785, 1999.

[BB99b] A. Baraldi and P. Blonda, A Survey of Fuzzy Clustering Algorithms for Pattern Recognition - Part II, *IEEE Trans. Syst., Man, Cybern.*, vol. 29, no. 6, pp. 786-801, 1999.

[BG99] M. Barni and R. Gualtieri, A new possibilistic clustering algorithm for line detection in real world imagery, *Pattern Recognition*, vol. 32, no. 11, pp. 1897-1909, 1999.

[BCM96] M. Barni, V. Cappellini and A. Mecocci, Comments on "A Possibilistic Approach to Clustering", *IEEE Trans. Fuzzy Systems*, vol. 4, no. 3, pp. 393-396, 1996.

[BHBC96] A.M. Bensaid, L.O. Hall, J.C. Bezdek and L.P. Clarke, Partially Supervised Clustering for Image Segmentation, *Pattern Recognition*, vol. 29, no. 5, pp. 859-871, 1996.

[Ber94] J.M. Bernardo, Optimizing Prediction with Hierarchical Models: Bayesian Clustering, In P.R. Freeman and A.F.M. Smith (Eds.), *Aspects of Uncertainty: A Tribute to D.V. Lindley*, Chichester: Wiley, pp. 67-76, 1994.

[Bez74] J.C. Bezdek, Cluster Validity with Fuzzy Sets, *J. Cybern.*, vol. 3, no. 3, pp. 58-73, 1974.

[BP98] J.C. Bezdek and N.R. Pal, Some new indexes of cluster validity, *IEEE Trans. Syst., Man, Cybern.*, vol. 28, no. 3, pp. 301-315, 1998.

[BP87] B. Bhanu and B.A. Parvin, Segmentation of natural scenes, *Pattern Recognition*, vol. 20, no. 5, pp. 487-496, 1987.

[BW70] D.M. Boulton and C.S. Wallace, A program for numerical classification, *The Computer Journal*, vol. 13, no. 1, pp. 63-69, 1970.

[BW73] D.M. Boulton and C.S. Wallace, An information measure for hierarchic classification, *The Computer Journal*, vol. 16, no. 3, pp. 254-261, 1973.

[BPM93] B. Braathen, W. Pieczynski and P. Masson, Global and local methods of unsupervised Bayesian segmentation of images, *Machine Graphics Vis.*, vol. 2, pp. 39-52, 1993.

[CPH97] H. Caillol, W. Pieczynski and A. Hillion, Estimation of Fuzzy Gaussian Mixture and Unsupervised Statistical Image Segmentation, *IEEE Trans. Image Processing*, vol. 6, no. 3, pp. 425-440, 1997.

[Cel90] M. Celenk, A Color Clustering Technique for Image Segmentation, *Comput. Vision Graphics Image Process.*, vol. 52, pp. 145-170, 1990.

[CLP98] C.W. Chen, J. Luo and K.J. Parker, Image Segmentation via Adaptive K-mean Clustering and Knowledge-Based Morphological Operations with Biomedical Applications, *IEEE Trans. Image Processing*, vol. 7, no. 12, pp. 1673-1683, 1998.

[Che95] Y. Cheng, Mean Shift, Mode Seeking, and Clustering, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 8, pp. 790-799, 1995.

[CY95] Z. Chi and H. Yan, Image segmentation using fuzzy rules derived from K-means clusters, *Journal of Electronic Imaging*, vol. 4, no. 2, pp. 199-206, 1995.

[CA79] G.B. Coleman and H.C. Andrews, Image segmentation by clustering, *Proc. IEEE*, vol. 67, no. 5, pp. 773-785, 1979.

[CB99] C. Colombo and A.D. Bimbo, Color-induced image representation and retrieval, *Pattern Recognition*, vol. 32, no. 10, pp. 1685-1695, 1999.

[CF97] C. Connolly and T. Fliess, A Study of Efficiency and Accuracy in the Transformation from RGB to CIELAB Color Space, *IEEE Trans. Image Processing*, vol. 6, no. 7, pp. 1046-1048, 1997.

[CM88] M.C. Cooper and G.W. Milligan, The effect of measurement error on determining the number of clusters in cluster analysis, In W. Gaul and M. Schader (Eds.), *Data, expert knowledge and decisions*, Berlin: Springer-Verlag, pp. 319-328, 1988.

[DK97] R.N. Davé and R. Krishnapuram, Robust Clustering Methods: A Unified View, *IEEE Trans. Fuzzy Systems*, vol. 5, no. 2, pp. 270-293, 1997.

[DB79] D.L. Davies and D.W. Bouldin, A cluster separation measure, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 1, no. 2, pp. 224-227, 1979.

[DMP97] Y. Delignon, A. Marzouki and W. Pieczynski, Estimation of Generalized Mixtures and Its Application in Image Segmentation, *IEEE Trans. Image Processing*, vol. 6, no. 10, pp. 1364-1375, 1997.

[DLR77] A.P. Dempster, N.M. Laird and D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm (with discussions), *Journal of the Royal Statistical Society B*, vol. 39, pp. 1-38, 1977.

[DK82] P.A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Hemel Hempstead: Prentice Hall, 1982.

[DH73] R.O. Duda and P.E. Hart, *Pattern classification and scene analysis*, New York: Wiley, 1973.

[Dun74] J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, *J. Cybern.*, vol. 3, no. 3, pp. 32-57, 1974.

[ECS+94] F. Ercal, A. Chawla, W.V. Stoecker, H.-C. Lee and R.H. Moss, Neural Network Diagnosis of Malignant Melanoma From Color Images, *IEEE Trans. Biomed. Eng.*, vol. 41, no. 9, pp. 837-845, 1994.

[Eve74] B.S. Everitt, *Cluster Analysis*, London: Heinemann Educational Books, 1974.

[Fis36] R.A. Fisher, The Use of Multiple Measurements in Taxonomic Problems, *Annals of Eugenics*, vol. 7, pp. 179-188, 1936.

[FV84] J.D. Foley and A. Van Dam, *Fundamentals of Interactive Computer Graphics*, Reading, Massachusetts: Addison-Wesley, 1984.

[For65] E.W. Forgy, Cluster analysis of multivariate data: efficiency vs. interpretability of classifications, abstract, *Biometrics*, vol. 21, pp. 768-769, 1965.

[FK97] H. Frigui and R. Krishnapuram, Clustering by competitive agglomeration, *Pattern Recognition*, vol. 30, no. 7, pp. 1109-1119, 1997.

[FK99] H. Frigui and R. Krishnapuram, A Robust Competitive Clustering Algorithm With Applications in Computer Vision, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 5, pp. 450-465, 1999.

[FM81] K.S. Fu and J.K. Mui, A survey on image segmentation, *Pattern Recognition*, vol. 13, no. 1, pp. 3-16, 1981.

[FCE00] C.-S. Fuh, S.-W. Cho and K. Essig, Hierarchical Color Image Region Segmentation for Content-Based Image Retrieval System, *IEEE Trans. Image Processing*, vol. 9, no. 1, pp. 156-162, 2000.

[Fuk90] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd edition, San Diego: Academic Press, 1990.

[FD97] J.-K. Fwu and P.M. Djurić, EM Algorithm for Image Segmentation Initialized by a Tree Structure Scheme, *IEEE Trans. Image Processing*, vol. 6, no. 2, pp. 349-352, 1997.

[GS99] T. Gevers and A.W.M. Smeulders, Color-based object recognition, *Pattern Recognition*, vol. 32, no. 3, pp. 453-464, 1999.

[GW92] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Massachusetts: Addison-Wesley, 1992.

[GGD71] A.M. Goon, M.K. Gupta and B. Dasgupta, *Fundamentals of Statistics Volume One*, Calcutta: World Press Private, 1971.

[GR95] K.C. Gowda and T.V. Ravi, Divisive clustering of symbolic objects using the concepts of both similarity and dissimilarity, *Pattern Recognition*, vol. 28, no. 8, pp. 1277-1282, 1995.

[Gow67] J.C. Gower, A comparison of some methods of cluster analysis, *Biometrics*, vol. 23, pp. 623-628, 1967.

[HS85] R.M. Haralick and L.G. Shapiro, Survey image segmentation techniques, *Comput. Vision Graphics Image Process.*, vol. 29, no. 1, pp. 100-132, 1985.

[Hea92] G. Healey, Segmenting Images Using Normalized Color, *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 1, pp. 64-73, 1992.

[HB97] T. Hofmann and J.M. Buhmann, Pairwise Data Clustering by Deterministic Annealing, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 1, pp. 1-14, 1997.

[HPB98] T. Hofmann, J. Puzicha and J.M. Buhmann, Unsupervised Texture Segmentation in a Deterministic Annealing Framework, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 8, pp. 803-818, 1998.

[HP74] S.L. Horowitz and T. Pavlidis, Picture segmentation by directed split and merge procedure, *Proc. 2nd Int. Joint Conf. Pattern Recognition*, pp. 424-433, 1974.

[HP78] S.L. Horowitz and T. Pavlidis, A Graph-Theoretic Approach to Picture Processing, *Comput. Graphics Image Process.*, vol. 7, pp. 282-291, 1978.

[Hun87] R.W.G. Hunt, *The reproduction of colour in photography, printing and television*, 4th edition, Tolworth, England: Fountain Press, 1987.

[Jai86] A.K. Jain, Cluster Analysis, In T.Y. Young and K.S. Fu (Eds.), *Handbook of pattern recognition and image processing*, Orlando: Academic Press, pp. 33-57, 1986.

[Jai89] A.K. Jain, *Fundamentals of Digital Image Processing*, Englewood Cliffs, NJ: Prentice Hall, 1989.

[JD88] A.K. Jain and R.C. Dubes, *Algorithms for clustering data*, New Jersey: Prentice Hall, 1988.

[JMF99] A.K. Jain, M.N. Murty and P.J. Flynn, Data Clustering: A Review, *ACM Computing Surveys*, vol. 31, no. 3, pp. 264-323, 1999.

[Jan91] Y. Jang, Identification of interreflection in color images using a physics-based reflection model, *Conference on Computer Vision and Pattern Recognition*, pp. 632-637, 1991.

[Joh67] S.C. Johnson, Hierarchical clustering schemes, *Psychometrika*, vol. 32, pp. 241-254, 1967.

[KMW96] M.S. Kankanhalli, B.M. Mehtre and J.K. Wu, Cluster-based Color Matching for Image Retrieval, *Pattern Recognition*, vol. 29, no. 4, pp. 701-708, 1996.

[KDNS96] T. Kanungo, B. Dom, W. Niblack and D. Steele, A Fast Algorithm for MDL-Based Multi-Band Image Segmentation, In J.L.C. Sanz (Ed.), *Im-*

age technology: advances in image processing, multimedia and machine vision, Berlin; New York: Springer-Verlag, pp. 147-186, 1996.

[KSW85] J.N. Kapur, P.K. Sahoo, and A.K.C. Wong, A new method for gray-level picture thresholding using the entropy of the histogram, *Comput. Vision Graphics Image Process.*, vol. 29, pp. 273-285, 1985.

[KFN98] T. Kaukoranta, P. Fränti and O. Nevalainen, A new iterative algorithm for VQ codebook generation, *International Conference on Image Processing*, pp. 589-593, 1998.

[Kin67] B. King, Step-wise clustering procedures, *J. Am. Stat. Assoc.*, vol. 62, pp. 86-101, 1967.

[Kow95] F. Kowalewski, A gradient procedure for determining clusters of relatively high point density, *Pattern Recognition*, vol. 28, no. 12, pp. 1973-1984, 1995.

[KK93] R. Krishnapuram and J.M. Keller, A Possibilistic Approach to Clustering, *IEEE Trans. Fuzzy Systems*, vol. 1, no. 2, pp. 98-110, 1993.

[KK96] R. Krishnapuram and J.M. Keller, The Possibilistic C-Means Algorithm: Insights and Recommendations, *IEEE Trans. Fuzzy Systems*, vol. 4, no. 3, pp. 385-393, 1996.

[Kun99] S. Kundu, Gravitational clustering: a new approach based on the spatial distribution of the points, *Pattern Recognition*, vol. 32, no. 7, pp. 1149-1160, 1999.

[Kur91] T. Kurita, An efficient agglomerative clustering algorithm using a heap, *Pattern Recognition*, vol. 24, no. 3, pp. 205-209, 1991.

[KC97] S.H. Kwok and A.G. Constantinides, A Fast Recursive Shortest Spanning Tree for Image Segmentation and Edge Detection, *IEEE Trans. Image Processing*, vol. 6, no. 2, pp. 328-332, 1997.

[LC99] P. Lambert and T. Carron, Symbolic fusion of luminance-hue-chroma features for region segmentation, *Pattern Recognition*, vol. 32, no. 11, pp. 1857-1872, 1999.

[LW66] G.N. Lance and W.T. Williams, Computer programs for hierarchical polythetic classification, *Comp. J.*, vol. 9, pp. 60-64, 1966.

[LMZ98] D.A. Langan, J.W. Modestino and J. Zhang, Cluster Validation for Unsupervised Stochastic Model-Based Image Segmentation, *IEEE Trans. Image Processing*, vol. 7, no. 2, pp. 180-195, 1998.

[LLL00] A.W.C. Liew, S.H. Leung and W.H. Lau, Fuzzy image clustering incorporating spatial continuity, *IEE Proc. Vision, Image and Signal Process.*, vol. 147, no. 2, pp. 185-192, 2000.

[LL90] Y.W. Lim and S.U. Lee, On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques, *Pattern Recognition*, vol. 23, no. 9, pp. 935-952, 1990.

[LY94] J. Liu and Y.-H. Yang, Multiresolution color image segmentation, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 7, pp. 689-700, 1994.

[Mac67] J. MacQueen, Some methods for classification and analysis of multivariate observations, *Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281-297, 1967.

[MMR96] D. Maio, D. Maltoni and S. Rizzi, Dynamic Clustering of Maps in Autonomous Agents, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 11, pp. 1080-1091, 1996.

[MK97] G.J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, New York: Wiley, 1997.

[Mil85] G.W. Milligan, An Algorithm for Generating Artificial Test Clusters, *Psychometrika*, vol. 50, no. 1, pp. 123-127, 1985.

[Mil96] G.W. Milligan, Clustering validation: Results and implications for applied analyses, In P. Arabie, L.J. Hubert and G. De Soete (Eds.), *Clustering and Classification*, Singapore: World Scientific, pp. 341-375, 1996.

[MC85] G.W. Milligan and M.C. Cooper, An examination of procedures for determining the number of clusters in a data set, *Psychometrika*, vol. 50, pp. 159-179, 1985.

[MB97] A. Moghaddamzadeh and N. Bourbakis, A Fuzzy Region Growing Approach for Segmentation of Color Images, *Pattern Recognition*, vol. 30, no. 6, pp. 867-881, 1997.

[NS96] L. Najman and M. Schmitt, Geodesic Saliency of Watershed Contours and Hierarchical Segmentation, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 12, pp. 1163-1173, 1996.

[Nev86] R. Nevatia, Image segmentation, In T.Y. Young and K.S. Fu (Eds.), *Handbook of pattern recognition and image processing*, Orlando: Academic Press, pp. 215-231, 1986.

[OPR78] R. Ohlander, K. Price and D.R. Reddy, Picture segmentation using a recursive region splitting method, *Comput. Graphics Image Process.*, vol. 8, pp. 313-333, 1978.

[OKS80] Y.-I. Ohta, T. Kanade and T. Sakai, Color information for region segmentation, *Comput. Graphics Image Process.*, vol. 13, pp. 222-241, 1980.

[OH94] J.J. Oliver and D. Hand, Introduction to minimum encoding inference, Technical Report No. 94/205, *Department of Computer Science, Monash University, Australia*, 1994.

[PB95] N.R. Pal and J.C. Bezdek, On cluster validity for the fuzzy c-means model, *IEEE Trans. Fuzzy Systems*, vol. 3, no. 3, pp. 370-379, 1995.

[PB97] N.R. Pal and J. Biswas, Cluster validation using graph theoretic concepts, *Pattern Recognition*, vol. 30, no. 6, pp. 847-857, 1997.

[PP89] N.R. Pal and S.K. Pal, Object-background segmentation using new definitions of entropy, *IEE Proceedings, pt. E*, vol. 136, no. 4, pp. 284-295, 1989.

[PP93] N.R. Pal and S.K. Pal, A review on image segmentation techniques, *Pattern Recognition*, vol. 26, no. 9, pp. 1277-1294, 1993.

[PMJ96] J.S. Pan, F.R. McInnes and M.A. Jack, Fast Clustering Algorithms for Vector Quantization, *Pattern Recognition*, vol. 29, no. 3, pp. 511-518, 1996.

[Pap92] T.N. Pappas, An Adaptive Clustering Algorithm for Image Segmentation, *IEEE Trans. Signal Process.*, vol. 40, no. 4, pp. 901-914, 1992.

[PYL98] S.H. Park, I.D. Yun and S.U. Lee, Color image segmentation based on 3-D clustering: morphological approach, *Pattern Recognition*, vol. 31, no. 8, pp. 1061-1076, 1998.

[Pat91] J.D. Patrick, SNOB: A program for discriminating between classes, Technical Report No. 91/151, *Department of Computer Science, Monash University, Australia*, 1991.

[PF99] E.J. Pauwels and G. Frederix, Finding Salient Regions in Images, *Computer Vision and Image Understanding*, vol. 75, nos. 1/2, pp. 73-85, 1999.

[PC99] S.-C. Pei and C.-M. Cheng, Color Image Processing by Using Binary Quaternion-Moment-Preserving Thresholding Technique, *IEEE Trans. Image Processing*, vol. 8, no. 5, pp. 614-628, 1999.

[PM82] T. Peli and D. Malah, A study of edge detection algorithms, *Comput. Graphics Image Process.*, vol. 20, no. 1, pp. 1-21, 1982.

[PG87] A. Pérez and R.C. Gonzalez, An iterative thresholding algorithm for image segmentation, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, no. 6, pp. 742-751, 1987.

[PFM93] N. Phamdo, N. Farvardin and T. Moriya, A unified approach to tree-structure and multistage vector quantization for noisy channels, *IEEE Trans. Inform. Theory*, vol. 39, pp. 835-850, 1993.

[Pie92] W. Pieczynski, Statistical image segmentation, *Machine Graphics Vis.*, vol. 1, pp. 261-268, 1992.

[PP97] K. Popat and R.W. Picard, Cluster-Based Probability Model and Its Application to Image and Texture Processing, *IEEE Trans. Image Processing*, vol. 6, no. 2, pp. 268-284, 1997.

[PC96] R. Porter and N. Canagarajah, A Robust Automatic Clustering Scheme for Image Segmentation Using Wavelets, *IEEE Trans. Image Processing*, vol. 5, no. 4, pp. 662-665, 1996.

[PS87] F. Preparata and M. Shamos, *Computational Geometry: An Introduction*, New York: Springer-Verlag, 1987.

[Pun80] T. Pun, A new method for grey-level picture thresholding using the entropy of the histogram, *Signal Processing*, vol. 2, pp. 223-237, 1980.

[Rao73] C.R. Rao, *Linear statistical inference and its applications*, 2nd edition, New Delhi: Wiley Eastern, 1973.

[Ray93] S. Ray, Entropy-based image segmentation, *Proc. Third International Conference on Advances in Pattern Recognition and Digital Techniques*, Calcutta, India, pp. 698-715, Dec. 28-31, 1993.

[RT94] S. Ray and R.H. Turi, Entropic thresholding of a colour image, *Proc. APRS workshop on electronic colour imaging and applications, Canberra, Australia*, pp. 149-155, 5-7 December, 1994.

[RD79] A. Rosenfeld and L.S. Davis, Image Segmentation and Image Models, *Proc. IEEE*, vol. 67, no. 5, pp. 764-772, 1979.

[Rub81] R.Y. Rubinstein, *Simulation and the Monte Carlo Method*, New York: Wiley, 1981.

[SSWC88] P.K. Sahoo, S. Soltani, A.K.C. Wong and Y.C. Chen, A survey of thresholding techniques, *Comput. Vision Graphics Image Process.*, vol. 41, no. 2, pp. 233-260, 1988.

[Sch97] P. Scheunders, A Genetic c-Means Clustering Algorithm Applied To Color Image Quantization, *Pattern Recognition*, vol. 30, no. 6, pp. 859-866, 1997.

[SPK97] L. Shafarenko, M. Petrou and J. Kittler, Automatic Watershed Segmentation of Randomly Textured Color Images, *IEEE Trans. Image Processing*, vol. 6, no. 11, pp. 1530-1544, 1997.

[SPK98] L. Shafarenko, M. Petrou and J. Kittler, Histogram-based Segmentation in a Perceptually Uniform Color Space, *IEEE Trans. Image Processing*, vol. 7, no. 9, pp. 1354-1358, 1998.

[ST97] G. Sharma and H.J. Trussell, Digital Color Imaging, *IEEE Trans. Image Processing*, vol. 6, no. 7, pp. 901-932, 1997.

[SSN98] X. Shen, M. Spann and P. Nacken, Segmentation of 2D and 3D images through a hierarchical clustering based on region modelling, *Pattern Recognition*, vol. 31, no. 9, pp. 1295-1309, 1998.

[Sne57] P.H.A. Sneath, The Application of computers to taxonomy, *J. Gen. Microbiol.*, vol. 17, pp. 201-226, 1957.

[SS73] P.H.A. Sneath and R.R. Sokal, *Numerical taxonomy: the principles and practice of numerical classification*, San Francisco: W.H. Freeman, 1973.

[TA97] M. Tabb and N. Ahuja, Multiscale Image Segmentation by Integrated Edge and Region Detection, *IEEE Trans. Image Processing*, vol. 6, no. 5, pp. 642-655, 1997.

[TRT95] P.E. Tischer, S. Ray and R.H. Turi, Minimal message length thresholding, *Proc. DICTA-95 conference, Brisbane, Australia*, pp. 395-400, 1995.

[Tou79] J.T. Tou, DYNOC - A dynamic optimal cluster-seeking technique, *International Journal of Computer and Information Sciences*, vol. 8, no. 6, pp. 541-547, 1979.

[TG74] J.T. Tou and R.C. Gonzalez, *Pattern Recognition Principles*, Massachusetts: Addison-Wesley, 1974.

[TPV94] P.E. Trahanias, I. Pitas, and A.N. Venetsanopoulos, Color image processing, *Control and Dynamic Systems*, vol. 67, pp. 45-90, 1994.

[TB97] A. Tremeau and N. Borel, A region growing and merging algorithm to color segmentation, *Pattern Recognition*, vol. 30, no. 7, pp. 1191-1203, 1997.

[TB86] M.M. Trivedi and J.C. Bezdek, Low-level segmentation of aerial images with fuzzy clustering, *IEEE Trans. Syst., Man, Cybern.*, vol. 16, no. 4, pp. 589-598, 1986.

[VS91] L. Vincent and P. Soille, Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 6, pp. 583-598, 1991.

[Wal84] C.S. Wallace, An improved program for classification, Technical Report No. 47, *Department of Computer Science, Monash University, Australia*, 1984.

[WB68] C.S. Wallace and D.M. Boulton, An information measure for classification, *The Computer Journal*, vol. 11, pp. 185-194, 1968.

[WD94] C.S. Wallace and D.L. Dowe, Intrinsic classification by MML - the Snob program, *Proc. 7th Australian Joint Conference on Artificial Intelligence*, UNE, Armidale, NSW, Australia, pp. 37-44, November 1994.

[War63] J.H. Ward, Hierarchical grouping to optimize an objective function, *J. Am. Statist. Ass.*, vol. 58, pp. 236-244, 1963.

[WS89] A.K.C. Wong and P.K. Sahoo, A gray-level threshold selection method based on maximum entropy principle, *IEEE Trans. Syst., Man, Cybern.*, vol. 19, no. 4, pp. 866-871, 1989.

[WYC94] J. Wu, H. Yan and A.N. Chalmers, Color image segmentation using fuzzy clustering and supervised learning, *Journal of Electronic Imaging*, vol. 3, no. 4, pp. 397-403, 1994.

[Xia97] Z. Xiang, Color image quantization by minimizing the maximum inter-cluster distance, *ACM Trans. Graphics*, vol. 16, no. 3, pp. 260-276, 1997.

[XB91] X.L. Xie and G.A. Beni, Validity measure for fuzzy clustering, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 3, no. 8, pp. 841-846, 1991.

[Zad65] L.A. Zadeh, Fuzzy Sets, *Information and Control*, vol. 8, pp. 338-353, 1965.

[ZLE99] N. Zahid, M. Limouri and A. Essaid, A new cluster-validity for fuzzy clustering, *Pattern Recognition*, vol. 32, no. 7, pp. 1089-1097, 1999.

[Zha96] Y.J. Zhang, A survey on evaluation methods for image segmentation, *Pattern Recognition*, vol. 29, no. 8, pp. 1335-1346, 1996.

[ZL98] D. Zugaj and V. Lattuati, A new approach of color images segmentation based on fusing region and edge segmentations outputs, *Pattern Recognition*, vol. 31, no. 2, pp. 105-113, 1998.