Document Image Segmentation for Document Compression

Emma Jonasson 18376479

Supervisor: Dr. Peter Tischer
Contents

1 Introduction 1

2 Research Context 2

3 Research Plan and Methods 5
   3.1 Research Methods ............................................. 5
   3.2 Proposed Thesis Chapter headings .......................... 9
   3.3 Timetable ...................................................... 9
   3.4 Special Facilities Required .................................. 9

4 Relevance of the Project 10
1 Introduction

There has been a steady increase in document images surrounding the western world since the early days of facsimile machines, copiers and scanners. The need for being able to process these images has also increased. Firstly, an efficient way for storage, retrieval and transmission information within document images was needed. This often required an analogue to digital conversion. This in turn has brought about the desire to be able to process the images in digital form and more importantly process the content of these images.

To recognise the content in document images, or any images for that matter, has proved amazingly difficult. Entire research areas have attempted to find the solution to content recognition such as segmentation, content-based image retrieval and content classification. Segmentation tries to recognise the content by building a model of the image data. This model groups things within the image together by some common property, thus building up different segments. Segments are homogeneous with respect this common property. Also, segments tend to be spatially connected. Widely used common properties include intensity, colour and texture. The best model constructed through segmentation, by the definition of a 'best model', also best describes what the content of the image is.

This notion of a 'best model' can be quite useful for content-based image retrieval, CBIR. CBIR system attempts to find images with certain characteristics, such as 'find all images with a passport photo' or 'find all images with a moon'. By finding the optimal model of an image, labels can be put on the different objects within the image to describe what they are. This could then be extended on to implement a full CBIR system.

Also segmentation turns out to be quite useful for binary image compression. The smallest amount of information to describe the pixels in the image is needed if all pixels in a specific segment gets described as a unit. Therefore by knowing things are in the same segment, they can be encoded most compactly, thus yielding good compression. Therefore by finding an optimal model of the image data, the best compression rates can arise. This is something that the current international standard for bi-level images, JBIG 2, exploits by allowing for segmentation. It also allows for encoding of segments to be utilised across more than one page, which is where JBIG 2 gains a lot of compression.

However, JBIG 2 does neither incorporate a standard encoder nor a standard segmentation algorithm. In fact, how to carry out image segmentation is a much discussed topic within many research areas. Several algorithms have been proposed, most only functioning under very specific conditions but there is no widely recognised algorithm. This project will attempt to prove that a segmentation approach that achieves good compression can also be useful for other areas. It aspires to find this proof by investigating a novel approach to efficient segmentation of binary images of documents. The common property under consideration to segment document images with is 'information content', i.e. pixels within a particular segment contains similar information.
2 Research Context

The ever growing abundance of document images has brought about two interrelated topics concerning document processing. The first applications dealt with storage, retrieval and transmission of the image without changing the image data. For these applications to be efficient, it was found that compressing the image data saved both on storage space and transmission time.

When document images were first made popular in the 1970’s saving resources was very important because unlike today multi-gigabyte hard drives and fibre optic cables were non-existent. Greater compression was achieved if the images were binarised before being compressed as this discards information in the image. Because the image now only uses black and white colours it can be stored very compactly.

The second applications regarding document processing came about when it became apparent people wanted to alter, or process, the content in the image. One such application is content classification, which attempts to find the structure within the image. This classification process produces a model of the image data by grouping image data into different classes depending. Data within a certain class share some common characteristic.

It has been found that by differentiating text from pictures and other elements, and vice versa, segmentation can be useful for achieving better compression (Kia; 1997). The improvement comes from knowing the model and compressing each class differently. By using an encoding optimised for the type of data within a particular class, all data will be stored most compactly. For example, text is separated from pictures in behaviour, thus it should have a different encoding than pictures. Finding the best model of the data will also find the best compression, since the best model will minimise the length of the data (Baxter; 1996).

Clustering is a widely used method for classifying image data. It divides the image into clusters up by some homogeneity property, such as texture, intensity or colour. A cluster is simply a grouping of pixels in the image where those pixels share a common property. Segmentation is a special case of clustering where you insist on the pixels in the cluster also are spatially connected.

Within segmentation multiple approaches have been put forward. Most of these approaches tend to incorporate a couple of well defined techniques (Trepode; 2002):

- Region growing
- Edge following
- Hybrid method (aka combination of above)

The region growing method works by initially finding a few small regions in the image that are homogeneous in nature. It then expands these regions according to some common property (Marshall; 1997). The edge, or contour, following method distinguishes edges by finding boundaries between homogeneous regions. It then works along the edges and groups things within edges into a segment. By combining these two main techniques hybrid methods arise, which can lead to powerful segmentation algorithms.

Often pre-processing is carried out on an image before it goes through the segmentation process. Pre-processing procedures attempts to optimise the image for successful segmentation. Common pre-processing method include noise removal, contrast enhancement and even binarisation can be considered a form of pre-processing. Noise removal attempts to discard information that could lead to incorrect classification. For example there might have been specks of dust when the document image was scanned, or faxed, which might
disrupt the classification process so that text gets treated as a picture. Contrast enhancement almost has the opposite effect of noise removal. By enhancing the contrast certain image data becomes clearer visually, so that vital data does not get discarded. It is important to note that many methods are destructive to the original image. They do not allow for reconstruction of the original image.

As mentioned previously, segmentation can greatly improve compression. However early document image compression schemes, such as CCITT Group 3, and Group 4 did not take this into account (Kia; 1997). When the Joint Bi-Level Image Experts Group introduced the new international standard JBIG, it did not consider segmentation either. However it managed to improve compression performances by up to 50% compared to those of previous standards (Murray and vanRyper; 1996). JBIG has the benefit of adapting to the characteristics of the binary image, among other things. The way this adaption works is on a current context basis. The current context is the pixels surrounding the currently considered pixel. It is widely known that a pixel is likely to be similar to that of its neighbours, and that we can use the values of the neighbours to predict what the value of the current pixel (Kou; 1995). For example one context might be to use the western neighbour to predict the current pixel, or a combination of the western, northern and north western neighbours etc. Which pixels to consider when carrying out this prediction is determined by a template. By being able to adapt which neighbours to use as the current context, better encoding is possible. JBIG utilised a template containing 10 pixels, where the tenth pixel was adaptive and could be positioned in one of multiple surrounding positions.

JBIG 2 is the successor of JBIG and is the current standard to date. It does not only allow for lossless encoding, where no information content in the image is lost, but also it allows for lossy compression which discards some information. It extends on context encoding as well as allowing for model-based encoding, or segmentation if you prefer (Howard et al.; 1998). JBIG2 implements an up-to-16 pixels template, where four pixels are deemed adaptive. Because JBIG 2 allows for an encoding of a particular segment to be carried across multiple pages, it managed to achieve much better compression than most other compression schemes at the time it was introduced. For example when compressing a typical book of 512 pages, JBIG 2 yields a 13 times smaller lossless encoding than JBIG (Ono et al.; 2000).

Surprisingly, JBIG 2 does not come with a standard encoder (Tompkins; 2000). Rather it comes with a toolkit of mechanisms which can be utilised to implement an encoder(Howard et al.; 1998). This gives clients the freedom to be however specific or general in their implementation of JBIG 2, which to some might be an advantage. Others might claim there is no standard encoder because there is no standard segmentation algorithm.

A typical encoder would have to implement a segmentation algorithm. But once the image is segmented, JBIG 2 does incorporate tools for encoding the segments optimally. JBIG 2 also tends to informing the decoder what it needs to know to be able to decode the compressed image properly. It does this by sending along models explaining the structure of the image as well as models that describe what goes in the different parts of the structure. The models that describe the content of a segment indicates the type of segment, which page it belongs to and also the length of data(Howard et al.; 1998). JBIG 2 only allows for three types of segments (Tompkins and Kossentini; 1999):

- Textual regions
- Halftone (approximation of a picture, colour or grey scale, using only one colour, such as black on white) regions
- Generic regions (that is everything else)
However there is no stated proof that only allowing for three kinds of segments is going far enough in a compression sense. Should text be segmented further into headers, body text, etc.? Or will this yield an encoding that is too complex? There’s a trade off between a more detailed segmentation and overall compression performances.

An alternative for binary image compression is the propriety compression approach known as DjVu (LeCun et al.; 1998). DjVu does not only operate on binary images, but also on grey scale and full colour images. It manages to combine small file sizes with often visually lossless images. DjVu also utilises segmentation in a slightly different way to JBIG 2. It divides an image into a background image and foreground image, as well as mask image. The mask image is then used to state whether to use foreground or background at a particular pixel location. One reason DjVu performs well on all kinds of images is that it incorporates different encoders depending on the type of data. An interesting fact is that DjVu uses a JBIG 2 variation known simply as JB2 to compress bi-level parts of the image.

Segmentation is not simply beneficial to compression. Because of its content recognition behaviour it can also contribute to research areas such as content-based image retrieval (CBIR) and optical character recognition (OCR). CBIR has become very popular lately because of increased interest in processing of image content. CBIR systems allow us to enter an image or a string into a search engine and retrieve images or documents corresponding to the input. So by finding the optimal way of storing the content of the images, there exists also an optimal way of retrieving it, making searching efficient (Gudivada and Raghavan; 1995). Segmentation as stated before finds the optimal way by creating a model of the data, making it perfect to OCR is concerned with finding text within an image. Segmenting the image into text and not text does just that. By classifying the text further, each character can be retrieved and grouped into its own segment. This results in each character being segmented from both the background and the other characters (Le; 1997; Wikipedia; 2005).

All previously mentioned topics are concerned with recognising the content within an image. Still there is no single widely used algorithm for segmentation that is employed across all these areas. One of the reasons for this is perhaps that there are difficulties comparing segmentation algorithms objectively to find a best algorithm. It may depend on particular circumstances that a best algorithm for segmentation purposes is found. Another major reason is that many proposed solutions to image segmentation are very specific in nature. Some claim to be time efficient, others space efficient, or yielding a better results, or more robust and so on. Some are specific to OCR segmentation where segmentation has to be much more accurate than solutions proposed for JBIG 2.

Some have been optimised to work with JBIG 2 encoders, such as the time efficient algorithm proposed by Tompkins and Kossentini which operates by rate change in the image (Tompkins and Kossentini; 1999). Tompkins and Kossentini’s approach might be time efficient but is obviously constrained to work only with JBIG 2 since it only detects text and not text without classifying further. It does not appear to utilise the full three types of segments JBIG 2 caters for, which obviously would yield a quicker algorithm. But this would not be beneficial to implement for someone that also needs to detect halftones from other regions.

Overall there does not appear to be any solutions which will work well on a variety of uses.
3 Research Plan and Methods

3.1 Research Methods

This project hopes to develop a novel approach for efficient segmentation of a binary image to both enhance compression rates and find an optimal model for classifying the document content. Being able to model the data in an ideal way is an important part of computing, as this is fundamental to many computing areas. The proposed way for accomplishing this new segmentation approach involves several sub-problems, which can be seen in Figure 1.

![Block diagram of sub-problems](image)

Figure 1: Block diagram of sub-problems

The whitening transform can be thought of as a pre-processing step. It attempts to optimise the image for segmentation by roughly capturing the information content of the image. It operates on a context template basis, where different templates need to be investigated to find the optimal template. By using the current context, this process attempts to predict the value of the current pixel. If this prediction is correct, white is output, else black is output. Depending on the type of image and its content, the whitened image will have an improved compressibility if a low order Markov model is used. A template that does not incorporate any neighbouring pictures, but works on a global context of the pixels that have been processed so far uses a zeroth order Markov model. A template that incorporates the western neighbouring pixel e.g., uses a first order Markov model, and so
on. The original image can actually be reconstructed from the whitened image as long as the same context is used. An example of an extract from a document image with its corresponding whitened image can be seen in Figures 2 and 3. The document image is one of the eight CCITT test images used to train Group 3 and Group 4 algorithms on.

Figure 2: Part of original image

Figure 3: Part of whitened image

The whitening transform appears to produce a skeleton image of the original image data. This skeleton appears to be suitable for segmentation of the original image as it outlines the edges of the image content. What the whitening transform actually attempts to do is quickly identify regions with similar information content.

After the whitening transform, a conversion from binary to grey scale is to be imple-
mented. This is a trivial process that blocks $n \times n$ pixels together to yield a single pixel in the new low resolution grey scale image. Different block sizes need to be investigated to yield the best results. The grey level is given by adding up the black pixels in the block. An alternative to this would be to subtract the number of black pixels in the block from the maximum grey level value. The maximum grey level value in this project will initially be 255, as the ‘blocked’ image will have 8 bits per pixel. The number of bits per pixel is subject to change. This blocking process will create a blurry and pixelated version of the whitened image. The blocked image will be much smaller because one single pixel is replacing $n \times n$ pixels in the whitened image. The hope is that the new grey scale image will produce a histogram which has many different peaks signifying the different kinds of information content. Each peak should correspond to a specific type of element in the image such as text, halftones, line art and so forth. We anticipate text and halftones, for example, will have different patterns in the original image as well as the whitened image. The different patterns occur because text has a different information content to that of halftones. Consequently this should produce different intensity values in the histogram of the blocked image. By examining the histogram and its peaks there is hope that the peaks can act as thresholds for clustering, and thus segmenting, the image data.

Examples of blocked versions of the whitened extract found in Figure 4 can be seen in Figures 5 and 6.

Figure 4: Extract from whitened image

![Extract from whitened image](image)

Figure 5: Extract from blocked image, $n = 8$

![Extract from blocked image, $n = 8$](image)

Figure 6: Extract from blocked image, $n = 16$

![Extract from blocked image, $n = 16$](image)
The whitening transform and the blocking process are both simple and time efficient procedures. When combined, they remove some noise in the image and, to a certain extent, also remove useless information. Also there will be a reduction by in the number of pixels. This should speed up the segmentation process.

Once the criterion for segmentation has been established, the original binary image needs to be segmented and encoded. The desire is to implement already existing segmentation programs, as these are likely to have been tested for both correctness and robustness. However there may not be any freely distributed segmentation programs that are suitable for this project. If this is the case, the main techniques within segmentation will have to be research and implemented.

The model for the segmentation will also be required to be encoded so that the encoder can forward this information to the decoder along with the segmented image data. It would be desirable to implement a JBIG 2 encoder to compare results with the current standard. However there appears to be no public domain for the toolkit of mechanisms JBIG 2 provides. This toolkit is unfortunately too complex to encode in a honours project as this would be an entire project in itself.

It might be possible to approximate the results by implementing a JBIG-like encoder and a segment map. A segment map is simply a way of letting us know which segment we are currently in. So a pixel would not only be determined by the current context template but also the current segment. This should technically yield better results than if no segmentation was carried out.

To test that the developed segmentation approach actually improves compression, comparisons between using segmentation and not using segmentation will be done. Likewise to show that the whitening coder optimises the image for segmentation, analysis between using and not using the whitening transform can be done. Obviously if the whitening transform does not greatly influence the segmentation, this approach will have to be reconsidered and other approaches might have to be found. Testing for an optimal segmentation is somewhat trickier. Obviously it can be compared to results of other segmentation approaches, but there is no guarantee there are any public domain segmentation software to compare against.

Another way of evaluating the segmentation is by using the human eyes. Humans are exceptionally good at finding structure within a document at a glance. If the segmentation developed in the project is close to what a human perceives as an optimal segmentation, it could be concluded to be successful. Unfortunately human beings are terribly subjective which makes it difficult to state one single optimal segmentation. While some consider all text to be in one segment, others might want to divide it further. This is where some limitations will have to be considered regarding how accurate the segmentation must be. Also the optimal structure according to a human might not be optimal for classification and compression. All these things must be evaluated in great detail so that all aspects are analyzed and the best overall segmentation is found. This evaluation should bear indication on whether the segmentation has been successful or not.

Monash Image Library (MIL) will be heavily relied upon to provide the basic image processing functions needed for parts of this project, such as retrieving and storing pixels, and finding the histogram of the image (School of Computer Science and Software Engineering; 2000). Since MIL is implemented in the programming language C it is reasonable to assume the entire project will be coded in C and, for some instances also C++. Also it will be designed to run under Linux.
3.2 Proposed Thesis Chapter headings

1 Introduction
   a) Purpose of research
   b) Objectives of research
2 Document image processing
3 Segmentation pre-processing
   a) "Whitening" transformation of binary image
   b) Binary to grey scale blocking conversion
4 Segmentation of grey scale images
5 Segmentation encoding
6 Results
7 Conclusion and Future Work
8 Bibliography
9 Appendix A sample data
10 Appendix B programs

3.3 Timetable

<table>
<thead>
<tr>
<th>Date</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>18/04/05</td>
<td>Research Proposal Draft</td>
</tr>
<tr>
<td>27/04/05</td>
<td>Research Proposal</td>
</tr>
<tr>
<td>28/04/05</td>
<td>Whitening transform encoder/decoder</td>
</tr>
<tr>
<td>03/05/05</td>
<td>Binary to grey scale image converter and Analysis</td>
</tr>
<tr>
<td>04/05/05</td>
<td>Commence Literature Review</td>
</tr>
<tr>
<td>10/05/05</td>
<td>Thesis Chapter 3</td>
</tr>
<tr>
<td>24/05/05</td>
<td>Thesis Chapter 2</td>
</tr>
<tr>
<td>25/05/05</td>
<td>Research Methods Assignment</td>
</tr>
<tr>
<td>26/05/05</td>
<td>Presentation Preparation</td>
</tr>
<tr>
<td>02/06/05</td>
<td>Interim Presentation</td>
</tr>
<tr>
<td>10/06/05</td>
<td>Literature Review Draft</td>
</tr>
<tr>
<td>06/06/05</td>
<td>Segmentation program</td>
</tr>
<tr>
<td></td>
<td>Thesis Chapter 4 and 5</td>
</tr>
<tr>
<td></td>
<td>JBIG2 baseline program/JBIG1 modified program</td>
</tr>
<tr>
<td>18/07/05</td>
<td>Putting it all together for analysis</td>
</tr>
<tr>
<td>27/07/05</td>
<td>Literature Review</td>
</tr>
<tr>
<td>17/10/05</td>
<td>Presentation Preparation</td>
</tr>
<tr>
<td>24/10/05</td>
<td>Final Presentation</td>
</tr>
<tr>
<td>Mid10/05</td>
<td>Final Report Draft</td>
</tr>
<tr>
<td>01/11/05</td>
<td>Final Report</td>
</tr>
<tr>
<td>01/11/05</td>
<td>Research Log Book</td>
</tr>
<tr>
<td>10/11/05</td>
<td>Project Web Site</td>
</tr>
</tbody>
</table>

Table 1: Timetable for project

See Table 1.

3.4 Special Facilities Required

The facilities offered for Honours students at Monash University, Clayton Campus, are sufficient to do the research.
4 Relevance of the Project

One of the most vital areas within computing is that of modeling data. This project aspires to do this that by retrieving the structure from the document image and classifying the content within. Project attempts to explore the idea that the ’best’ homogeneity property for segmentation should be ‘information content’. If an optimal modeling of the information in the document can be found, it will not only be frolicsome in compression terms, but could also be beneficial to many other areas such as content classification, content-based image retrieval, to name a couple.
References


URL: http://www.ldtechnologies.com/techhtm.htm#preprocessing%20product

URL: http://www.djvuzone.org/djvu/techpapers/lecun-98c/lecun-98c.ps.gz

URL: http://www.cs.cf.ac.uk/Dave/Vision_lecture/node33.html


URL: http://www.ime.usp.br/~walter/vision_c/ISVMregr.htm

URL: http://en.wikipedia.org/wiki/Optical_character_recognition