

# The Angular Integral of the Radon Transform (aniRT) as a Feature Vector in Categorization of Visual Objects

Andrew P. Papliński

Monash University, Australia  
Andrew.Paplinski@monash.edu

**Abstract.** The recently introduced angular integral of the Radon transform (aniRT) seems to be a good candidate as a feature vector used in categorization of visual objects in a rotation invariant fashion. We investigate application of aniRT in situations when the number of objects is significant, for example, Chinese characters. Typically, the aniRT feature vector spans the diagonal of the visual object. We show that a subset of the full aniRT vector delivers a good categorization results in a timely manner.

**Keywords:** Radon transform, categorization of visual objects, Chinese characters, Self-Organizing Maps, incremental learning.

## 1 Introduction

The paper is continuation of [20,22,21] where we considered categorization of visual objects based on the angular integral of the Radon transform [25] (aniRT). The principal problem in such categorization is the selection of the features that can represent a visual object in a way invariant to rotation, scaling, translation (RST), changes in illumination and the viewpoint. In this context, the main attraction of the Radon transform is its ability to deliver rotational invariance of the visual object. We will discuss this aspect in the section that follows. In this paper, we consider a simple case of one visual object in the image. This can be considered as a special case of more general invariant visual recognition based on a variety of local descriptors (e.g. [14,5,19]). Three groups of methods, namely, the Scale Invariant Feature Transform [16] (SIFT), PCA-SIFT [12] and the “Speeded Up Robust Features” (SURF) [2], seem to dominate the field. A comparison between the three methods is given in [11]. In [7] a rotation-invariant kernels are discussed in application to shape analysis. An interesting method (Ciratefi) that delivers the RST-invariance based on the template matching is presented in [13,1].

The prime application of the Radon transform has been in computer tomography. Relatively recently, Radon transform has been applied in a variety of image processing problems. Typically, Radon transform is used in conjunction with other transforms, wavelet and Fourier included. Magli et al. [17] and Warrick and

Delaney [26] seem to initiate the use of Radon transform in combination with wavelet transform. More recently, a similar combination of transforms has been used in rotation invariant texture analysis [10,29], and in shape representation [28]. Other approach to rotation invariant texture analysis uses Radon transform in combination with Fourier transform [27]. Chen and Kégl [3] consider feature extraction using combination of three transforms: Radon, wavelet and Fourier. In [15], texture classification is performed by using a feature descriptor based on Radon transform and an affine invariant transform. Miciak [18] describes a character recognition system based on Radon transform and Principal Component Analysis. Hejazi et al. [8] present discrete Radon transform in rotation invariant image analysis. Close to our considerations are object identification problems discussed by Hjouj and Kammler in [9].

To our knowledge, the application of the angular integral of the Radon transform (aniRT) in the rotational invariance categorization problem has originally been presented in [20,22,21]. In those papers we used the aniRT feature vector in conjunction with the Self-Organizing Maps to categorize a relatively small number, say 30, of visual objects. We used the full aniRT feature vector that has the number of components equal to the diagonal of the visual objects.

In this paper we aim at testing the aniRT feature vector when the number of visual objects is significant, say 20000. We use a black-and-white rendering of Chinese Characters as our test visual objects. Using all the components, say 100, of the aniRT vector would be very time-demanding in the visual categorization process.

## 2 An Angular Integral of Radon Transform (aniRT)

We consider first a real continuous function of two real variables  $x, y$  (or one complex variable  $z$ )  $f(z)$ , like an image, where  $z = x + jy$ . The Radon transform,  $Rf(\theta, s)$ , (see [25]) of such a function is an integral of  $f(z)$  over straight lines  $z(t) = e^{j\theta}(s + jt)$  (see Fig. 1).

$$Rf(\theta, s) = \int_{-\infty}^{+\infty} f(e^{j\theta}(s + jt)) dt \quad (1)$$

where  $\theta$  is the slope of the line (or ray),  $s$  is the distance of the line from the origin, and  $t \in (-\infty, +\infty)$  is the line parameter, such that for  $t = 0$  the line goes through the point  $z(0) = se^{j\theta}$  as indicated in Fig. 1. Descriptively, we say that each 2-D point of the Radon transform,  $Rf(\theta, s)$ , is calculated as a summation (an integral) of the values of the function  $f(z)$ , e.g. pixels of an image, along the ray  $z(t) = e^{j\theta}(s + jt)$ , which is a straight line located at the distance  $s$  from the origin (the image centre) and the slope  $\theta$ .

In the next step we consider the formation from the Radon transform,  $Rf(\theta, s)$ , a feature (or signature) function, say,  $h(s)$ , that can be used to categorize the function  $f(x, y)$  in general, and images in particular, in a rotation-invariant

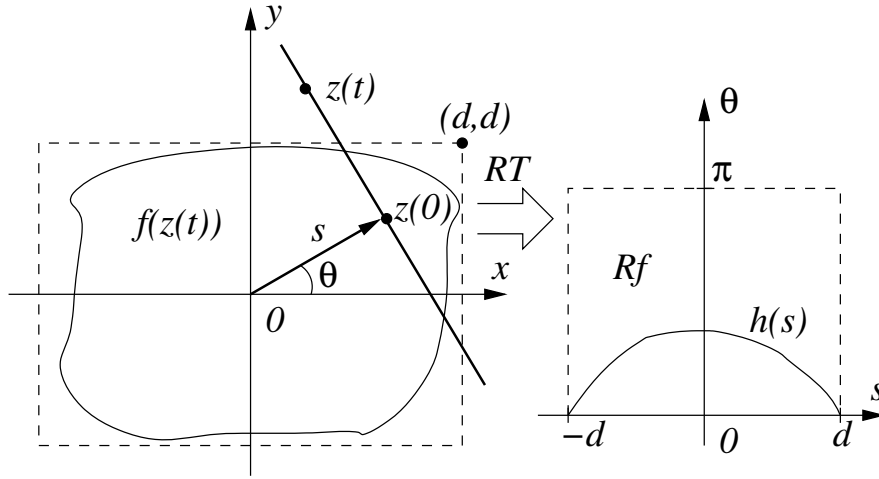


Fig. 1. The Radon Transform of a continuous function

fashion. This can be easily achieved by integrating the Radon transform wrt to the angular variable  $\theta$ :

$$h(s) = \int_0^\pi Rf(\theta, s) d\theta \tag{2}$$

With reference to the right-hand side of Fig. 1 we note that the integral is performed with respect to the angular variable. Such a feature (signature) function,  $h(s)$ , retains some characteristics of the original function  $f(x, y)$ , but the angular dependency is removed, hence, providing the rotational invariance. In a discrete case,  $h(s)$ , becomes a vector that spans the diagonal of the image,  $s \in [-d, +d]$ . We will be referring to this vector as the angular integral of the Radon Transform (aniRT).

Before we consider an effective way of calculating aniRT,  $h(s)$ , we recall a fundamental property of the Radon transform which says that the Radon transform at the point  $(\theta, s)$  of a function rotated by the angle  $\alpha$  is equal to the Radon transform of the original, un-rotated function calculated at the point  $(\theta + \alpha, s)$ :

$$Rf(\theta, s; \alpha) = \int_{-\infty}^{+\infty} f(e^{j(\theta+\alpha)}(s + jt)) dt = Rf(\theta + \alpha, s) \tag{3}$$

In practical terms it means that instead of rotating the rays  $z(t)$  we can rotate the image  $f(z)$  in order to calculate the Radon transform at the point  $(s, \theta)$ . Typically, rotation of an image for the purpose of visualization on a rectangular grid implies the need for interpolation. However, in order to calculate the Radon transform, or specifically the aniRT vector,  $h(s)$ , we just rotate the grid, that is, the coordinates of the pixels, without modifying the pixels values.

Let us consider some details of the computational procedure. Firstly, the rotation angles are quantized and the quantization step, that is, the smallest rotation angle  $\delta$ , should be at the order of (see Fig. 2)

$$\delta = \arctan \frac{2}{\rho} \tag{4}$$

where  $\rho = \max(r, c)$  and  $r$  and  $c$  are the number of rows and columns in the image, respectively. The rotation angles,  $\theta$ , for which the Radon transform is calculated are:

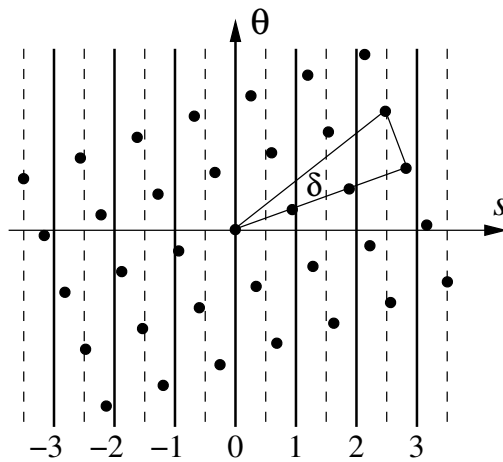


Fig. 2. Calculation of the aniRT vector

$$\theta_k = k \cdot \delta, \text{ for } k = 1, 2, \dots, \lfloor \frac{\pi}{\delta} \rfloor \tag{5}$$

Now, consider coordinates of all pixels,  $z_i = x_i + jy_i$ , for  $i = 1, 2, \dots, N$  where  $N$  is to total number of image pixels. We can pre-calculate all rotated image coordinates, that is,

$$\zeta_{ik} = z_i e^{j\theta_k} \tag{6}$$

With reference to Fig. 2 we notice that the Radon transform integral of eqn. (1) becomes a sum of pixel values  $f(\zeta_{ik})$  along the vertical direction. Such a sum gives one point of the Radon transform  $Rf(\theta, s)$  where  $s = \text{round}(\text{real}(\zeta))$ , is the rounded value of the real part of the rotated pixels coordinates. It is now obvious that to calculate the aniRT vector,  $h(s)$ , we sum pixels values,  $f(\zeta_{ik})$ , for all angles as in eqn. (2).

In Fig. 3 we present the aniRT vectors,  $h(s)$ , calculated for 10 randomly selected Chinese characters. The characters are rendered in the Microsoft JhengHei font of the size 28. The number of pixels is  $70 \times 70$ , hence the size (the number of components) of the aniRT vectors is 98. We observe two basic properties of the aniRT vectors:

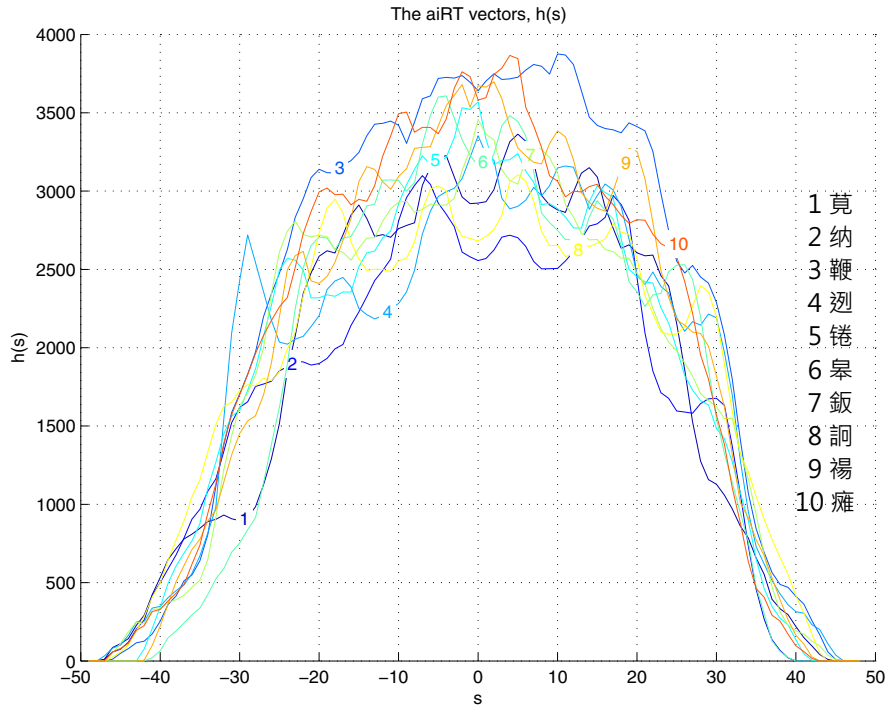


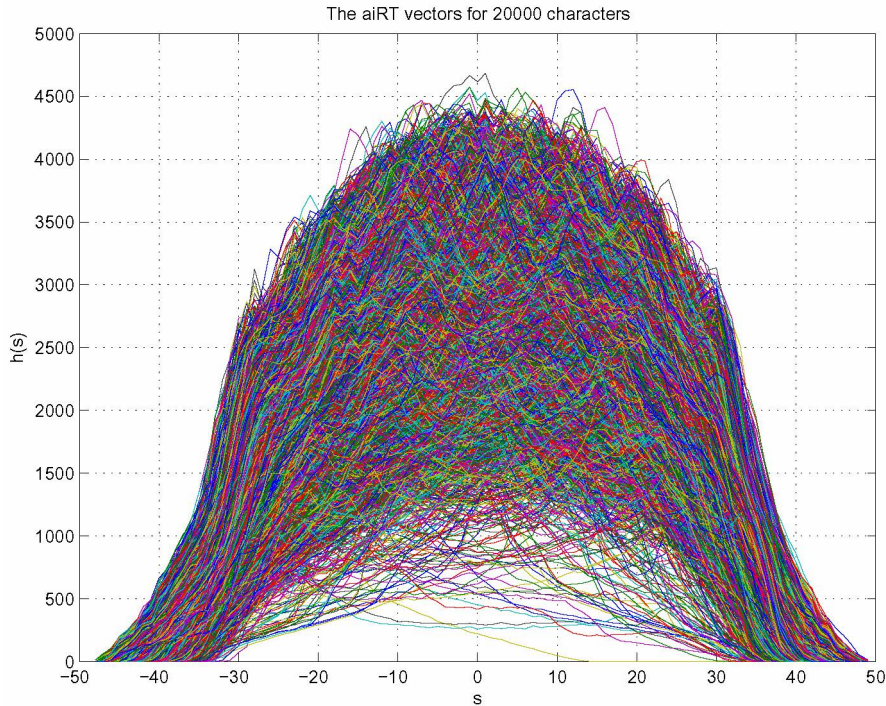
Fig. 3. The aniRT vectors,  $h(s)$ , for 10 randomly selected Chinese characters

- they are significantly different for different characters
- the main differentiation occurs in the central part of the vectors.

The obvious question now is how many components we need to differentiate a given set of images. Before we investigate this issue, in Fig. 4 we present the aniRT vectors calculated for 20,000 Chinese characters. We create 10 random permutations of the Chinese characters and investigate how many components of the aniRT vectors are needed to have a unique representation of the characters, The results are presented in Table 1. A bit unexpectedly, we observe from the last row of the Table 1, the number of components to make the selection of 20,000 characters uniquely represented is just 3. The result seems to be perfectly correct if we consider the fact that the components of the aniRT vectors are coded by

Table 1. The minimum number of components of the aniRT vectors to differentiate specified number of characters

cmp	number of characters										
1	2	2	2	2	2	2	2	2	2	2	2
2	6	58	8	72	107	100	14	85	67	107	
3	1102	168	669	2137	836	892	951	1061	2106	825	



**Fig. 4.** The aniRT vectors,  $h(s)$ , for 20,000 Chinese characters

four 13-bit numbers (the maximum value is greater than  $4096 = 2^{12}$  — see Fig. 4), giving the total number of bits being 15, just enough to code 20,000 characters.

### 3 Example of Categorization of Chinese Characters with the Incremental Self-Organizing Map (iSOM)

Unlike our previous usage of aniRT vectors in categorization of visual objects ([20,22,21]) where the all components of the vector were utilized, here we demonstrate the result of categorization based on small number of components as discussed in the previous section. For this purpose we will create a self-organizing map for a random selection of 200 characters out of the total number of 20,000 characters as in Fig. 5. We will use the version of the incremental SOM as presented in [21]. Similar, but non-incremental SOMs for categorization are also presented in [23,6,4,24]. The main characteristic features distinguishing our SOMs from the most commonly used are as follows:

- Instead being located on a rectangular grid, our neuronal units are randomly distributed inside a unit circuit. The neuronal units are represented by the yellow dots in Fig. 5.

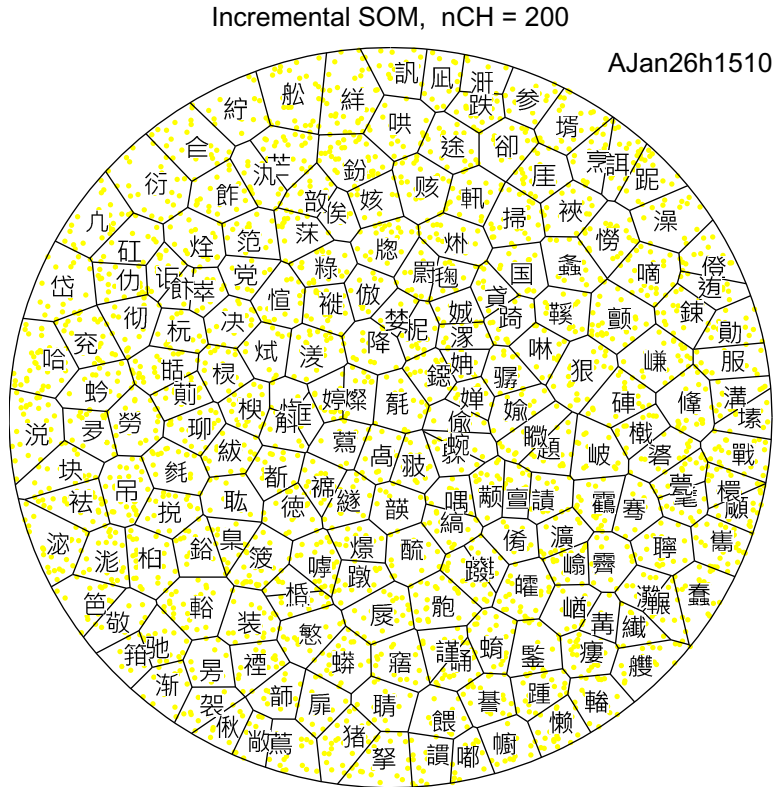


Fig. 5. iSOM for 200 Chinese characters

- A fix number (stochastically), 16 in the example, of neuronal units per character is maintained. This is to simulate a redundance observed in our brain to represents mental objects.
- All vectors are normalised to be unity length and located on the respective hyperspheres.
- We use the “dot-product” version of the Kohonen learning law.
- Although not explicitly shown here, the learning is done in an incremental fashion, adding one character at a time as in [21].

With reference to Fig. 5 it needs to be said that the topological grouping of Chinese characters is not an “intelligent” one, e.g. characters with the same radicals being grouped together. Instead the grouping is based on the similarity of the angular integral of the Radon Transform, or rather the four central components of the aniRT. In general, the objective of the example is to demonstrate that such a small number of components is enough to spread apart 200 characters in the SOM. As discussed in [22] the aniRT vectors ensure the rotational invariance of the representation of visual objects.

## 4 Concluding Remarks

We have investigated some properties of the recently introduced feature vector for images, aniRT, which is calculated as an angular integral of the Radon Transform of images. We have demonstrated a novel method of calculating the aniRT vector based on rotation of the image grid. Such a method gives a significant time advantage when an aniRT vector is calculated for a large number of images as in our example with Chinese characters. Subsequently, we have shown that to differentiate 20,000 characters only four components of the aniRT vectors are required. Finally, we have demonstrated the formation of a SOM consisting of 200 characters using 4 components of the aniRT feature vector.

## References

1. Araújo, S.A., Kim, H.Y.: Color-Ciratefi: A color-based RST-invariant template matching algorithm. In: Proc. 17th Int. Conf. Systems, Signals and Image Proc., pp. 1–4. Rio de Janeiro (2010)
2. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
3. Chen, G., Kégl, B.: Feature extraction using Radon, wavelet and Fourier transform. In: Proc. IEEE Int. Conf. Syst. Man and Cybernetics, pp. 1020–1025 (2007)
4. Chou, S., Papliński, A.P., Gustafsson, L.: Speaker-dependent bimodal integration of Chinese phonemes and letters using multimodal self-organizing networks. In: Proc. Int. Joint Conf. Neural Networks, Orlando, Florida, pp. 248–253 (August 2007)
5. Fehr, J.: Local rotation invariant patch descriptors for 3D vector fields. In: Proc. Int. Conf. Pat. Rec., Istanbul, pp. 1381–1384 (2010)
6. Gustafsson, L., Papliński, A.P.: Bimodal integration of phonemes and letters: an application of multimodal self-organizing networks. In: Proc. Int. Joint Conf. Neural Networks, Vancouver, Canada, pp. 704–710 (July 2006)
7. Hamsici, O.C., Martinez, A.M.: Rotation invariant kernels and their application to shape analysis. IEEE Tran. PAMI 31(11), 1985–1999 (2009)
8. Hejazi, M., Shevlyakov, G., Ho, Y.S.: Modified discrete Radon transforms and their application to rotation-invariant image analysis. In: Proc. IEEE Workshop Mult. Sig. Proc., pp. 429–434 (2006)
9. Hjouj, F., Kammler, D.W.: Identification of reflected, scaled, translated, and rotated objects from their Radon projections. IEEE Trans. Img. Proc. 17(3), 301–310 (2008)
10. Jafari-Khouzani, K., Soltanian-Zadeh, H.: Rotation-invariant multiresolution texture analysis using Radon and wavelet transforms. IEEE Trans. Img. Proc. 14(6), 783–795 (2005)
11. Juan, L., Gwun, O.: A comparison of SIFT, PCA-SIFT and SURF. Int. J. Img. Proc. 3(4), 143–152 (2009)
12. Ke, Y., Sukthankar, R.: PCA-SIFT: A more distinctive representation for local image descriptors. In: Proc. Conf. Comp. Vis. Pat. Rec., Washington, pp. 511–517 (2004)



13. Kim, H.Y., de Araújo, S.A.: Grayscale template-matching invariant to rotation, scale, translation, brightness and contrast. In: Mery, D., Rueda, L. (eds.) PSIVT 2007. LNCS, vol. 4872, pp. 100–113. Springer, Heidelberg (2007)
14. Kingsbury, N.: Rotation-invariant local feature matching with complex wavelets. In: Proc. 14th Europ. Sig. Proc. Conf., Florence, Italy, pp. 1–5 (2006)
15. Liu, G., Lin, Z., Yu, Y.: Radon representation-based feature descriptor for texture classification. *IEEE Trans. Img. Proc.* 18(5), 921–928 (2009)
16. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comp. Vision* 60(2), 91–110 (2004)
17. Magli, E., Presti, L.L., Olmo, G.: A pattern detection and compression algorithm based on the joint wavelet and Radon transform. In: Proc. IEEE 13th Int. Conf. Dig. Sig. Proc., pp. 559–562 (1997)
18. Miciak, M.: Character recognition using Radon transformation and principal component analysis in postal applications. In: Proc. Int. Multiconf. Comp. Sci. Info. Tech., pp. 495–500 (2008)
19. Nelson, J.D.B., Kingsbury, N.G.: Enhanced shift and scale tolerance for rotation invariant polar matching with dual-tree wavelets. *IEEE Trans. Img. Proc.* 20(3), 814–821 (2011)
20. Papliński, A.P.: Rotation invariant categorization of visual objects using Radon transform and self-organizing modules. In: Wong, K.W., Mendis, B.S.U., Bouzerdoum, A. (eds.) ICONIP 2010, Part II. LNCS, vol. 6444, pp. 360–366. Springer, Heidelberg (2010)
21. Papliński, A.P.: Incremental self-organizing map (iSOM) in categorization of visual objects. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) ICONIP 2012, Part II. LNCS, vol. 7664, pp. 125–132. Springer, Heidelberg (2012)
22. Papliński, A.P.: Rotation invariant categorization of colour images using Radon transform. In: Proc. WCCI-IJCNN, pp. 1408–1413. IEEE (2012)
23. Papliński, A.P., Gustafsson, L.: Multimodal feedforward self-organizing maps. In: Hao, Y., Liu, J., Wang, Y.-P., Cheung, Y.-M., Yin, H., Jiao, L., Ma, J., Jiao, Y.-C. (eds.) CIS 2005. LNCS (LNAI), vol. 3801, pp. 81–88. Springer, Heidelberg (2005)
24. Papliński, A.P., Gustafsson, L., Mount, W.M.: A recurrent multimodal network for binding written words and sensory-based semantics into concepts. In: Lu, B.-L., Zhang, L., Kwok, J. (eds.) ICONIP 2011, Part I. LNCS, vol. 7062, pp. 413–422. Springer, Heidelberg (2011)
25. Radon, J.: Radon transform, [http://en.wikipedia.org/wiki/Radon\\_transform](http://en.wikipedia.org/wiki/Radon_transform)
26. Warrick, A., Delaney, P.A.: Detection of linear features using a localized Radon transform with a wavelet filter. In: Proc. ICASSP, pp. 2769–2772 (1997)
27. Xiao, S.S., Wu, Y.X.: Rotation-invariant texture analysis using Radon and Fourier transforms. *J. Phys.: Conf. Ser.* 48, 1459–1464 (2007)
28. Yao, W., Pun, C.M.: Invariant shape representation by Radon and wavelet transforms for complex inner shapes. In: Proc. IEEE Int. Conf. Inform. Autom., pp. 1144–1149 (2009)
29. Yu, G., Cao, W., Li, Z.: Rotation and scale invariant for texture analysis based on Radon transform and wavelet transform. In: Proc. 3rd ICPCA, pp. 704–708 (2008)