Number Recognition Using Integral Invariants on Spiral Architecture

Qiang Wu, Tom Hintz and Xiangjian He
Department of Computer Systems, University of Technology, Sydney
PO Box 123, Broadway 2007, Australia
{wuq, hintz, sean}@it.uts.edu.au

Abstract *In this paper, a number recognition algorithm on Spiral Architecture is developed and experimented. This algorithm uses affine integral invariants with invariant parameterization. It firstly separates the object edge into a few closed boundaries. Each boundary is segmented into a few sections according to their directions (clockwise or anticlockwise) around a common central. Finally, a feature vector rather than a single feature value is computed for each number. This proposed algorithm successfully computes the unique feature vectors for ten standard Arabic numerals. Every number can still be distinguished from others even experiencing an affine transformation.*

Keywords: Spiral Architecture, number recognition, affine integral invariants

1 Introduction

Number recognition and character recognition are playing more important roles in image processing field. In the international trade, as a result of the accelerated development of global industries, thousands of containers and trucks need to be registered every day at container terminals and depots. Normally, this registration will be done manually. However, this is not only prone to error but also slow to meet the increasing volume of containers and trucks. Hence, an automatic, fast and exact number recognition process is required.

The fundamental issue in pattern recognition is shape description and recognition. Shape analysis has been a field of intense study in image processing. Many methods have been seen in the past ten years such as a morphological function [1], a gradient propagation method [2] or a special weighted graph for shape similarity [3]. Many other techniques such as Fourier description, template matching, invariant moments or neural network [4] are also used for shape description and recognition.

Number recognition and character recognition are the fields to which shape analysis is applied. Traditionally, different approaches in this field can be classified into two categories: global analysis or structural analysis [5], which are used in conjunction with statistical classification methods or a syntactical classification approach. Tsang [6] proposed a novel algorithm which encodes the image using the neighborhood relations. In this way, information about the pattern structure is maintained under translation invariance, while the shape is examined taking into consideration the whole area instead of just the borders. The encoding results are fed into a back-propagation neural network to get the final recognition result. The correctness of this algorithm depends on how much noise is depressed by the pre-processing.

In fact, many existing algorithms used in general object recognition application can be extended to number recognition. The importance of invariants in object recognition and identification has been revealed in computer vision. Actually, the number can also be recognized and identified by such invariance. There are many kinds of invariants, which can be classified to two categories: global invariants and local invariants. Either type of invariants has advantages and disadvantages in computer vision application. Sato [7] summarized the properties of these existing invariants based on important requirements in computer vision application-noise sensitivity, correspondence, occlusion. In this paper, we choose integral invariants which take advantages of both differential invariants and algebraic invariants.

In order to improve the performance of recognition and speed up processing, an algorithm is achieved within Spiral Architecture in this paper. Spiral Architecture [8] is inspired from anatomical considerations of the primate's vision . From the research about the geometry of the cones on the primate's retina we can conclude that the cones' distribution has inherent organization and is featured by its potential powerful computation abilities. The cones with the shape of hexagons are arranged in a spiral clusters. This cluster consists of the organizational units of vision. Each unit is a set of seven-hexagon [9] as shown in Figure 1.

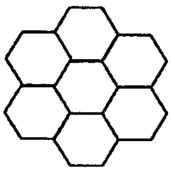

Figure 1. Seven-hexagon unit of vision

On Spiral Architecture, two algebraic operations have been defined, which are Spiral Addition and Spiral Multiplication [8]. These two operations can be used to define two transformations on Spiral address space respectively, which are translation of image and scaling rotation of image. During Spiral Multiplication, the original image is segmented into several parts. Each part is a near copy of the original image rotating in some degree. However, such image rotation will not affect image processing for object recognition if we choose affine integral invariant to represent the object's feature. In this paper, Spiral Multiplication will be used to make seven near copies of the original number firstly, so we can calculate seven feature vectors for the processed number. The final feature vector is the average of these seven vectors. In this way, the correctness of the final result is improved and the algorithm is more robust to noise. Moreover, these seven copies can be distributed to seven machines to be processed independently, so the processing time is greatly reduced.

The organization of this paper is as follows. Affine invariant parameterization and integral invariant representation are introduced in Section 2. Section 3 shows the procedure of number recognition within Spiral Architecture. This is followed by experiment results in Section 4. We conclude in Section 5.

2 Affine Invariant Parameterization and Integral Invariant Representation

Integral invariants under affine transformation are chosen to represent an object, which are less sensitive to noise than the classical invariants. In order to compute the affine invariant feature value at each point, it is necessary to parameterize the object contour with a correct parameter.

2.1 Invariant Parameterization

Consider a pre-extracted contour on the plane, $C \in \Re^2$, represented by $(x(t), y(t))$ with parameter t. There is a well-known parameter which is linearly transformed under an affine transformation [7], can be used for parameterising object contour. It is the area size, σ, defined as

$$\sigma(t_1) = \frac{1}{2} \int_{t_1-\Delta t}^{t_1+\Delta t} |x(t)y'(t) - y(t)x'(t)| dt \qquad (1)$$

at $(x(t_1), y(t_1))$. After the object undergoes a general affine transformation, its parameterization values will change. Hence, a normalization process is necessary to make these values same.

After analysis, we know that the affine transformation only make parameterization area size, $\sigma(t)$, time a determinant of a coefficient matrix, which is decided by the affine transformation [10]. So if we use ratio of each point's parameterization area size and the whole contour parameterization area size as represented on Equation 2 below, the multiplication constant can be removed

$$s(t_1) = \frac{\int_{t_1-\Delta t}^{t_1+\Delta t} |x(t)y'(t) - x'(t)y(t)| dt}{\int_C |x(t)y'(t) - x'(t)y(t)| dt} \qquad (2)$$

where $\int_C$ denotes the line integral along C, then s is invariant parameter under the general affine transformation. This new parameter, $s \in [0,1]$, will be used in this paper to parameterize the contour.

2.2 Integral Invariant Representation

Under the general affine transformation, we use integral invariants to represent the object in the form of

$$I(s_1) = \int_{s_1-\Delta s}^{s_1+\Delta s} F(s) ds \qquad (3)$$

where $I(s_1)$ is invariant feature value at point $(x(s_1), y(s_1))$, parameterized by s_1 defined in

Section 2.1. $F(s)$ is a differential invariant under general affine transformation. The function F is chosen such that the integral formula can be solved analytically and the resulting invariant has a simple form. For this purpose, let the contour C be represented by

$$\mathbf{V}(s)=\begin{pmatrix} x(s) \\ y(s) \end{pmatrix} \quad (4)$$

for $s \in [0,1]$. We define $F(s)$ in the form of

$$F(s)=\det[\mathbf{V}'(s),\mathbf{g}] \quad (5)$$

$\mathbf{V}'(s)$ denotes the derivative of $\mathbf{V}(s)$ with respect to s . $\mathbf{g}$ is a given constant for $\mathbf{g} \in \Re^2$. $\det[\mathbf{v}_1,\mathbf{v}_2]$ denotes the determinant of a matrix which consists of two column vector, $\mathbf{v}_1,\mathbf{v}_2 \in \Re^2$. They are substituted into Equation (4), we get

$$\begin{aligned} I(s_1) &= \int_{s_1-\Delta s}^{s_1+\Delta s} \det[\mathbf{V}'(s),\mathbf{g}]ds \\ &= \det[\mathbf{V}(s_1+\Delta s)-\mathbf{V}(s_1-\Delta s),\mathbf{g}] \end{aligned} \quad (6)$$

for $s_1 \in [0,1]$. For each fixed parameter s_1, we can define constant, $\mathbf{g}$, as

$$\mathbf{g}=\mathbf{V}(s+\Delta s)-\mathbf{V}(s) \quad (7)$$

Then, Equation (6) changes to

$$I(s)=\det[\mathbf{V}(s+\Delta s)-\mathbf{V}(s),\mathbf{V}(s-\Delta s)-\mathbf{V}(s)] \quad (8)$$

for $s \in [0,1]$. Actually, it is the cross product of two vectors, $\mathbf{V}(s+\Delta s)-\mathbf{V}(s)$ and $\mathbf{V}(s-\Delta s)-\mathbf{V}(s)$ as shown in Figure 2.

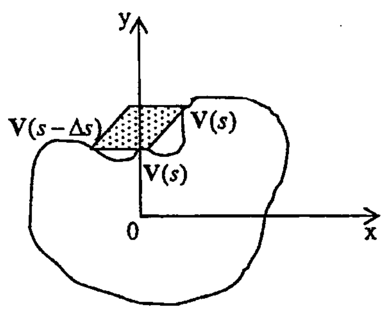

Figure 2. Cross Product of Two Vectors

In fact, $I(s)$ is only an affine relative invariant representation [10]. In order to protect this value from being affected by affine transformation, $I(s)$ can be represented as a ratio form

$$M(s)=\frac{I(s)}{I(v)} \quad (9)$$

where $|I(v)|=\max|I(s)|$ for $s \in [0,1]$. Equation (9) is defined as the affine integral invariant representation at the contour point for each parameter value s.

3 Number Recognition within Spiral Architecture

Spiral Architecture is a relatively new and powerful approach to general purpose machine vision system. It contains very useful geometric and algebraic properties. In this paper, we use it to separate the original image into seven near copies. After that, every copy will be processed independently to get the affine invariant feature vector. By doing this, we not only shorten the processing time via distributed processing but also create the multiple feature vectors to improve the performance of the final recognition scores.

3.1 Pre-processing

Spiral Architecture contains very useful geometric and algebraic properties, which can be interpreted in terms of the mathematical object, Euclidean ring. Two algebraic operations have been defined on SHM, Spiral Addition and Spiral Multiplication [8]. Among them, Spiral Multiplication can achieve uniform image segmentation. Each smaller part after separation is a near copy of the original image. That means each copy results in a unique sampling of the input image. Each sample is mutually exclusive and the collection of all such samples represents a partitioning of the input image. As each smaller part is the scaling down copy of the original image, each copy has less information. However, as none of the individual light intensities have been altered in any way, the scaled images in all still hold all of the information contained in the original. The whole process consists of process donw on individual near copies. But the processing time is greatly decreased if we put such processing into a distributed cluster system.

Gaussian Multi-Scale theory introduced by Linderberg [11] is applied here for edge-detection. According to this theory, image brightness function is parameterized. Image is blurred and the noise is removed when the parameter is positive. We can use this theory for edge detection to remove and suppress image noise, and then to simplify the processing tasks. In general, global Gaussian processing provides us the high precision but with the huge computation. On the other hand, local Gaussian processing certainly decreases computation with the comparatively low precision. So it is required to balance the computation and the precision when processing. Fortunately, because the original image has been separated into seven smaller parts before the Gaussian processing

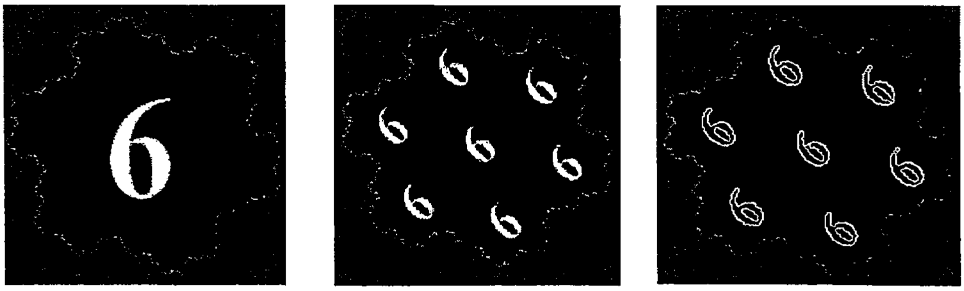

Figure 3. Image Pre-processing;
a) Original Image; b) Uniform Image Segmentation; c) Edge Map

and each part is a near copy of the original one as mentioned above, we can still achieve global processing but with shorter time.

Whether affine integral invariants can successfully represent the object depends on the quality of image edge. But naturally, the edge map after Gaussian Edge-detection seldom forms a high quality image edge that is required for the object feature measurement due to the gaps left by noise and shading effects, so an edge-point-linking procedure is necessary. In our work, we use a three-step approach of edge-thinning, edge-point-linking and region mergence to improve the edge map quality. After this pre-processing, we get perfect single-pixel-wide connected boundaries. Figure 3c shows an after pre-processing edge map.

3.2 Boundary Segmentation

Normally, if we can segment the restored image edge after pre-processing (see section 3.1) into a few meaningful units before affine integral invariant representation and then measure each unit to get the feature value, the recognition performance can be improved, because many numbers have been distinguished from others according to such segmentation results. Two kinds of segmentations will be done in this sub-section.

In the ten Arabic numerals, some of them have more than one boundary such as number six (See Figure 3). Because these boundaries are separated, we had better process them respectively to get the individual affine integral invariant representation. In addition, in most case any two numbers can be distinguished from each other by the outmost boundary, which has the main features of the original number, so potentially the processing time can be shortened and the processing procedure can also be simplified.

Moreover, while walking on each boundary around the common central the direction often changes. Sometimes it is clockwise, but sometimes it is anticlockwise. Actually, this is also an important feature to distinguished one number from others. For this reason, each boundary will be segmented into many sections according to such walking direction. However, due to the noise and the errors introduced by digital processing, some very short sections will appear after segmentation. Since these short sections are not profitable to the final feature measurement but may damage the final results when they scatter along the main sections, it is very necessary to merge them into their neighboring sections. By doing this, we can avoid the interference brought by noise and errors without loosing the main features of the number. Finally, we define the longest clockwise section as the first section followed by other sections clockwise (see Figure 4).

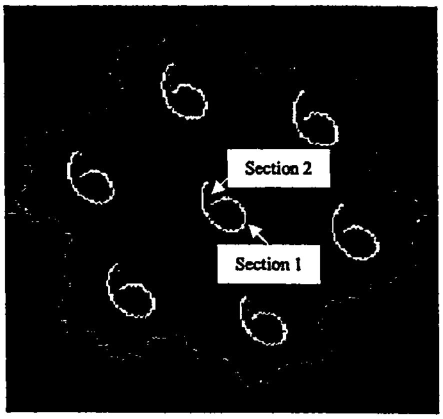

Figure 4. Boundary Segmentation

3.3 Computation of Parameter within Spiral Architecture

After boundary segmentation, each section will be parameterized independently. In this subsection, we show the methods for computation of parameterization area size, s, for each point within Spiral Architecture.

We choose the first point of the section along the walking direction on the original boundary as the starting point, which is the point with value 0 of the parameter s. Our definition of parameterization area is based on the assumption that the origin of Cartesian coordinate system is the centroid or the centre of the boundary section. Hence, it is important to find the centroid of boundary section. Let C be the boundary section, which is a set of contour points (hexagonal pixels). For any given hexagon $a \in C$, denote the Cartesian coordinates of a by (x,y), i.e., $a=(x,y)$. Suppose that the number of hexagons on C is N. Let

$$\bar{x}=\frac{1}{N}\sum_{a\in C}x \qquad \bar{y}=\frac{1}{N}\sum_{a\in C}y \tag{10}$$

Then the centroid or centre of the boundary section C is $(\bar{x},\bar{y})$.

Let $a_1=(x_1,y_1)$ and $a_2=(x_2,y_2)$ be two conjoined points along the walking direction on the original boundary. Then the area size at point a_1 is defined by

$$E(a_1)=\frac{1}{2}\left|[(x_2,y_2)-(\bar{x},\bar{y})]\times[(x_1,y_1)-(\bar{x}-\bar{y})]\right| \tag{11}$$

i.e., one half of the absolute value of the cross product of the vectors $(x_2-\bar{x},y_2-\bar{y})$ and $(x_1-\bar{x},y_1-\bar{y})$ denoted by $E(a)$. Suppose that there are N points on the boundary section, the parameterization area size, s, of each point is defined by

$$s_0=0 \qquad s_i=\frac{\sum_{j=1}^{i}E(a_j)}{\sum_{j=1}^{N-1}E(a_j)} \tag{12}$$

where $i=1,2,\cdots,N-1$.

3.4 Affine Integral Invariant Feature Value Calculation and Feature Matching

Now we can calculate the affine integral invariant feature value of each point on every boundary section according to Equation (8) and Equation (9) using the parameter, s, computed in subsection 3.3.

In theory, any parameter value $s\in[0,1]$ corresponds to a point on the boundary section, but due to digital processing we cannot get the correct previous point $V(s-\Delta s)$ and the correct successive point $V(s+\Delta s)$ for each reference point $V(s)$. So interpolation is used here to estimate the potential point position. In the experiment, it is found that in order to decrease the interpolation errors the reference points adopt the practically existing points on the boundary section. We only estimate its previous point and its successive point. To some shorter section, we still need to decrease Δs properly.

In this work, we use the average $M(s)$ as the invariant feature value for each boundary section and denote it by $\overline{M}$, i.e.,

$$\overline{M}=\frac{1}{N}\sum\{M(s)\mid V(s)\in C\} \tag{13}$$

At last, we create the feature value vector for each boundary. The dimension of the vector is determined by the number of sections. The first component value is the longest section's affine invariant feature value defined by Equation (13) followed by other section's affine invariant feature values along the walking direction on the original boundary. So finally we have seven feature vectors of seven near copies because of distributed processing as mentioned in subsection 3.1. We use the average of these seven feature vectors, denoted by

$$\overline{\overline{M}}=\sum_{i=1}^{7}\overline{M}_i \tag{14}$$

as the final feature vector to recognize the numeral. To compare the feature value vector of two boundaries with the same fixed Δs value. The difference between the two feature vectors should be small if the two boundaries represent the same numbers under the affine transformation.

One important application of the recognition of an unknown numeral is to recognize the numeral in an image as one of ten model Arabic numerals. This process requires the representation of the numeral be matched with one of those in the database. An numeral in an image is considered to belong to a specific class if the degree of dissimilarity between the numeral and models of

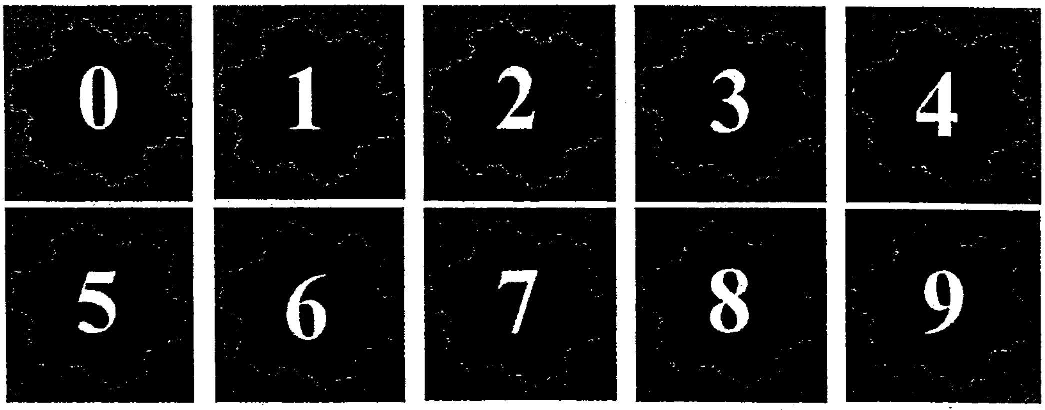

Figure 5. Ten Standard Arabic Numerals on Spiral Architecture

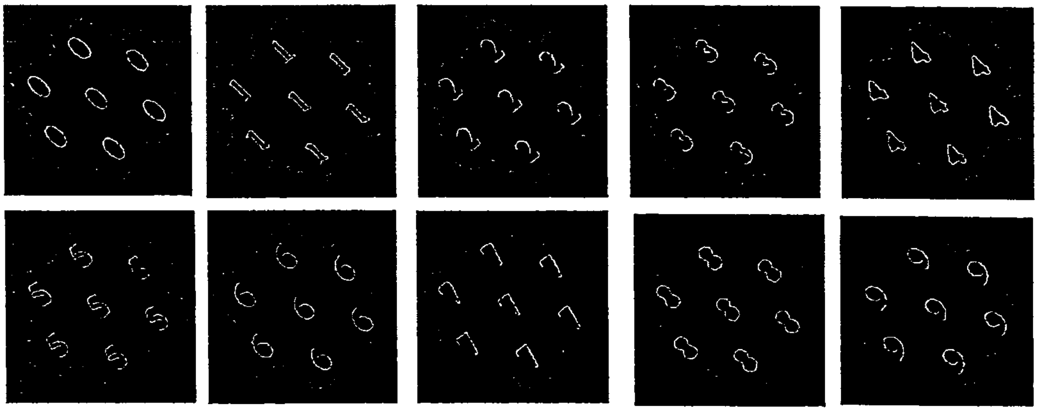

Figure 6. Outmost Boundaries with Different Sections Including the Centre Point of Each Section

Table 1. Feature Vectors of Ten Standard Arabic Numerals

Zero	One	Two	Three	Four	Five	Six	Seven	Eight	Nine
$[-0.02099]$	$\begin{bmatrix}0.012481\\0.000717\end{bmatrix}$	$\begin{bmatrix}-0.00423\\0.019323\\0.004147\\0.001618\end{bmatrix}$	$\begin{bmatrix}0.001312\\0.002607\\0.010798\\0.011079\end{bmatrix}$	$[-0.0159]$	$\begin{bmatrix}-0.01007\\0.009813\\0.00004\\0.006291\end{bmatrix}$	$\begin{bmatrix}-0.02047\\0.008493\end{bmatrix}$	$\begin{bmatrix}-0.00034\\0.004905\end{bmatrix}$	$[0.00191]$	$\begin{bmatrix}-0.02047\\0.008493\end{bmatrix}$

that class is the smallest in comparison to the others.

4 Experiment

We use this algorithm to process ten standard Arabic numerals (See Figure 5) to calculate their affine invariant feature vectors. After boundary segmentation, their outmost boundaries with different sections including the centre point of each section can be seen in Figure 6. In the computation of affine integral invariant representation, let $\Delta s = 0.01$ and linear interpolation is used to estimate the previous points and the successive point for each reference point. The values of $\overline{\overline{M}}$ of ten numerals are listed in Table 1. From Table 1 it is evident that each numeral is distinguished from others clearly except between numeral six and numeral nine, since numeral six actually is the rotation of numeral nine. We will discuss it in Section 5 and propose the methods we are researching to resolve this problem.

In order to verify the correctness of the algorithm under affine transformation, the additional two numerals, two and three, which are affine transformation results of standard number two and numeral three (See Figure 7). Table 2 lists their feature vectors. It shows that after affine transformation the feature vector still approach their corresponding model vectors.

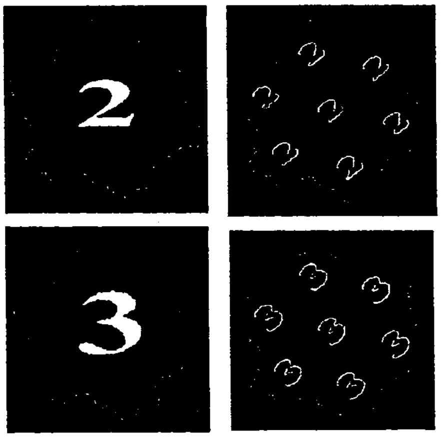

Figure 7. Affine Transformation of Numeral Two and Numeral Three

Table 2. Feature Vectors of '2' and '3' After Affine Transformation

Two	Three
$\begin{bmatrix} -0.00522 \\ 0.011382 \\ 0.002675 \\ 0.010711 \end{bmatrix}$	$\begin{bmatrix} 0.001483 \\ 0.001343 \\ 0.010316 \\ 0.00659 \end{bmatrix}$

5 Conclusion

This paper presents the number recognition using affine integral invariant representation within Spiral Architecture. Based on affine integral invariant theory combined with Spiral Architecture theory, we get the feature vector rather than a single feature value for each Arabic numeral, which is average of multiple feature vectors. The experiment results show that such algorithm can calculate the affine invariant feature value of the Arabic numerals, and then distinguish one numeral from others.

During the work, we find since numeral six is the rotation of numeral nine vice versa, they cannot be distinguished from each other. Currently, we use the relationship of two centre points of two sections on their outmost boundaries (See Figure 6) to distinguish them, but surely this simple method does not work when the numerals rotate in a large degree. Another potential method we are considering is to investigate the space relation between numeral six, numeral nine and other numerals when they are put with other numerals together.

Reference

[1] D. Sinha and C.R.Giardina. "Discrete black and white object recognition via morphological functions". *IEEE Trans. PAMI* 12 (3), 275—293, 1990

[2] J. Ben-Arie, K.R. Rao and Z. Wang. "Robust shape description and recognition by gradient propagation". In *Proc. IEEE ICIP 95 (3)*, Washington, DC, 368-371, 1995

[3] K.Y. Kupeev and H.J. Wolfson. "A new method of estimating shape similarity". *Pattern Recognition Letter* 17 (8), 873—887, 1996.

[4] S.W. Lee. "Off-line recognition of totally unconstrained handwritten numerals using multiple cluster neural network". *IEEE Trans. PAMI* 18(6), 648—652, 1996

[5] C.Y. Suen, C. Nadal, R. Legault, T.A. Mai and L. Lam. "Computer recognition of unconstrained handwritten numerals", *Proc. IEEE* 80 (7), 1162—1180, 1992.

[6] I.J. Tsang, I.R. Tsang and D. Van Dyck. "Image coding using neighbourhood relations", *Pattern Recognition Letters* 20, 1279—1286, 1999

[7] J. Sato and R. Clpolla, "Affine integral invariants for extracting symmetry axes", *Image and Vision Computing* 15, 627—635, 1997

[8] P. Sheridan. "Spiral Architecture for machine vision". *PhD thesis*, University of Technology, Sydney, Australia, 1996

[9] P. Sheridan, T. Hintz, and D. Alexander. "Pseudo-invariant image transformations on a hexagonal lattice". *Image and Vision Computing* 18 (11): 907-917, 2000

[10] X. He. "2D-object recognition with Spiral Architecture". *PhD thesis*, University of Technology, Sydney, Australia, 1999

[11] T. Lindeberg, "Scale-Space theory in computer vision". *Kluwer Academic Publishers*, London, 1994