工學碩士 學位論文

A Study on the Fingerprint Recognition Method Directional Feature Detection using Neural Networks

신경회로망을 이용한 방향성 특징추출 지문인식방법에 관한 연구

指導敎授 李 尙 培

2001年 2月

韓國海洋大學校 大學院

電子通信工學科

李　柱　尙

Contents

요약문

지문을 이용한 인식은 매우 오래전부터 이용되어 온 것으로 잘 알려져 있다. 지문은 본인만의 유일성과 불변성으로 오늘날 가장 널리 이용되는 신체 특징중의 하나이다. 그러므로 지문을 이용한 개인 식별은 개인의 인증 수단으로 가장 안전한 방법 중의 하나라고 할 수 있다.

본 논문에서는 신경회로망을 이용한 지문인식방법과 그레이-스케일 지문 영상으로 부터의 방향성 특징 벡터 추출방법에 대해 제안하였다. 방향성 특징 벡터는 이진화와 세선화 과정없이 그레이-스케일 영상으로부터 직접적으로 추출한다.

본 논문에서 제안한 특징 벡터 추출방법의 기본 아이디어는 융선 패턴의 지역 방위에 따라 그레이-스케일 영상의 융선을 따라가면서 융선의 방향성을 추출하는 것이다. 융선을 따라가는 시작점은 그레이-스케일 영상을 일정한 격자로 나누어서 격자안의 중심점으로 결정한다. 그 다음에 융선을 따라가면서 여러방향의 방향성 특징 벡터를 추출하고, 추출된 방향성 특징 벡터를 4방향성 특징 벡터로 라벨링한다.

실험은 4개의 지문에서 구성한 124개의 특징 패턴을 가지고 하였으며, 하나의 지문은 31개의 특징패턴으로 구성하였다. 그 결과 학습된 지문을 인식하는 능력이 매우 우수함을 보여주었다.

Abstract

Fingerprint-based identification is known to be used for a very long time. Owing to their uniqueness and immutability, fingerprints are today the most widely used biometric features. Therefore, recognition using fingerprints is one of the safest methods as a way of personal identification.

In this paper, a fingerprint identification method using neural networks and the direction feature vectors based on the directional image extracted from gray-scale fingerprint image without binarization and thinning is proposed.

The basic idea of the above mentioned method is to track the ridge lines on the gray-scale image, by “sailing” according to the local orientation of the ridge pattern. A set of starting points are determined by superimposing a grid on the gray-scale image. A labeling strategy is adopted to examine each ridge line only once and locate the intersections between ridge lines. After the direction feature vectors are consisted of vectors by four direction labeling. Matching method used in this paper is four direction feature vectors based matching.

The experiment are used total 124 feature patterns of four fingerprints, and One fingerprint image is consisted of 31 feature patterns. The results is presented excellent recognition capability of learned fingerprint images.

Chapter 1. Introduction

In this paper, a fingerprint identification method using neural networks[1] and the direction feature vectors based on the directional image extracted from gray-scale fingerprint image without binarization and thinning[2] is proposed. ; this choice is motivated by the following considerations: the first, a lot of information may be lost during the binarization process. the second, binarization and thinning are time-consuming. the third, the binarization techniques which were experimented proved to be unsatisfactory when applied to low-quality images.

The basic idea of the proposed method is to track the ridge lines on the gray-scale image, by “sailing” according to the local orientation of the ridge pattern. A set of starting points is determined by superimposing a grid on the gray-scale image; for each starting point, the algorithm keeps following the ridge lines until they terminate or intersect other ridge lines (direction detection). A labeling strategy is adopted to examine each ridge line only once and locate the intersections between ridge lines. After the direction feature vectors are consisted of vectors by four direction labeling. Matching method used in this paper is four direction feature vectors based matching. In this paper is proposed the use of Neural Networks(NN) in fingerprint matching.

In Section 2, Neural Networks(NN) are discussed. In section 3, discusses the direction feature vector detection, and four direction labeling and pattern detection. In section 4, discusses the result of fingerprint matching. Finally, in Section 5 some conclusions are drawn.

Chapter 2. Neural Networks

2.1 Introduction of Neural Networks

In search of better solutions for engineering and computing tasks, many avenues have been pursued. There has been a long history of interest in the biological sciences on the part of engineers, mathematicians, and physicists endeavoring to gain new ideas, inspirations, and designs. As the name implies, neural networks are computer models of the process and mechanisms that constitute biological nerve systems, to the extent that they are understood by researchers.

Namely, neural networks are systems that are deliberately constructed to make use of some organizational principles resembling those of the human brain. Neural networks are a promising new generation of information processing systems that demonstrate the ability to learn, recall, and generalize from training pattern of data [3 - 8].

In summary, neural networks are a parallel distributed information processing structure with the following characteristics:

1. Neural networks are neurally inspired mathematical models.
2. Neural networks consist of a large number of highly interco-

nnected processing elements.

3. Their connections(weight) hold the knowledge.
4. A processing element can dynamically respond to their input stimulus, and the response completely depends on their local information: that is, the input signals arrive at the processing element via impinging connections and connection weights.
5. Neural networks have the ability to learn, recall, and generalize from training data by assigning or adjusting the connection weights.
6. Their collective behavior demonstrates the computational power, and no single neuron carries specific information.

Neural networks is expected to be widely applied in vision, speech, decision-making, reasoning, and signal processors such as filters, detectors, and quality control systems. Also neural networks may offer solutions for cases in which a processing algorithm or analytical solutions are hard to find, hidden, or nonexistent. Such cases include modeling complex processes, extracting properties of large sets of data, and providing identification of plants that need to be controlled [7][10].

2.2 Biological Neuron and Artificial Neuron

Neural networks are inspired by modeling networks of real biological neurons in the brain. Hence, the processing elements in neural networks are also called artificial neurons, or simply neurons. A human brain consists of approximately 10^{11} neurons of many different types. A schematic diagram of a typical biological neuron is shown in Fig. 1 [5 - 8].

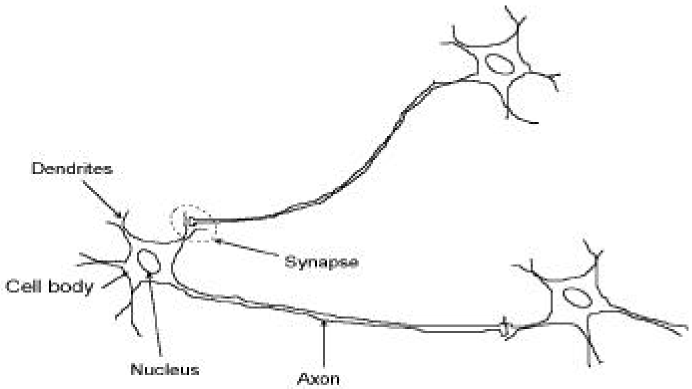


Fig. 1 Schematic diagram of a biological neuron

A typical neuron has three major parts: the cell body or soma, where the cell nucleus is located, the dendrites, and the axon. The signals reaching a synapse and received by dendrites are electric impulses. Such signal transmission involves a complex chemical process in which specific transmitter substances are released from the sending side of the junction. This raises or lowers the electric potential inside the body of the receiving cell. The receiving cell fires if its electric potential reaches a threshold, and a pulse or action potential of fixed strength and duration is sent out through the axon to the axonal arborization to synaptic junctions to other neurons. After firing, a neuron has to wait for a period of time called the refractory period before it can fire again. Synapses are excitatory if they let passing impulses cause the firing of the receiving neuron, or inhibitory if they let passing impulses hinder the firing of the neuron.

Fig. 2 shows a simple mathematical model of the above mentioned biological neuron proposed by McCulloch and Pitts[3],[5],[6].

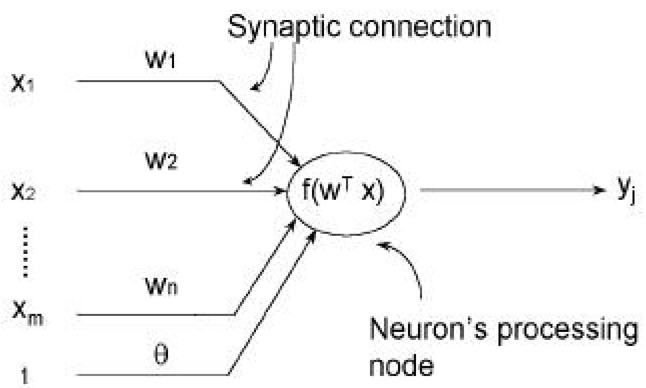


Fig. 2 Schematic diagram of a Mc Culloch and Pitts neuron

$$y_i = f\left(\sum_{j=1}^{m} w_{ij} x_j - \theta_i\right) \tag{2.1}$$

The weight w_{ij} represents the strength of the synapse connecting neuron j to neuron i. A positive weight corresponds to an excitatory synapse, and a negative weight corresponds to an inhibitory synapse. The neuron as a processing node performs the operation of summation of its weighted inputs, or the scalar product computation. Subsequently, it performs the nonlinear operation f(·) through its activation function. Some commonly used activation function show Fig. 3 such as unipolar sigmoid function, bipolar sigmoid function, linear function.

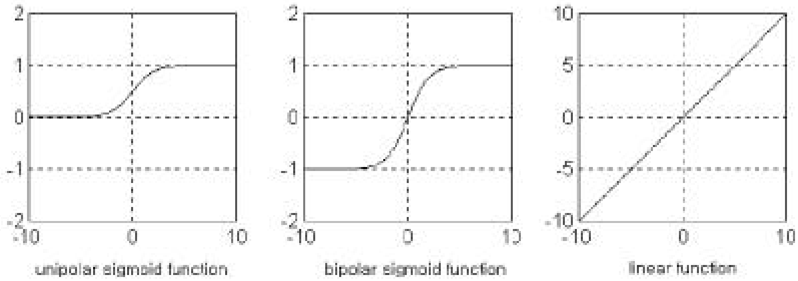


Fig. 3 Activation functions of a neuron

2.3 Learning and Structure of Multilayered Neural Networks

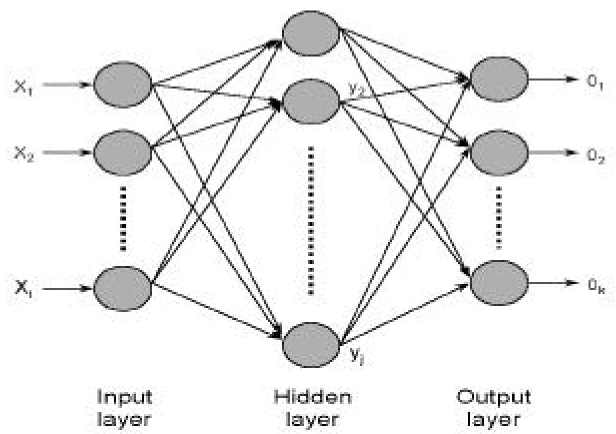


Fig. 4 Multilayered neural networks

Multilayered neural networks were used as basic structure for the applications discussed here. Fig. 4 shows multilayered neural networks[3],[5 - 8].

The back propagation training algorithm allows experiential acquisition of input/output mapping knowledge within multilayered neural networks. Fig. 5 illustrates the flowchart of the error back propagation training algorithm for a basic two layer network as in Fig. 4 [5 - 8].

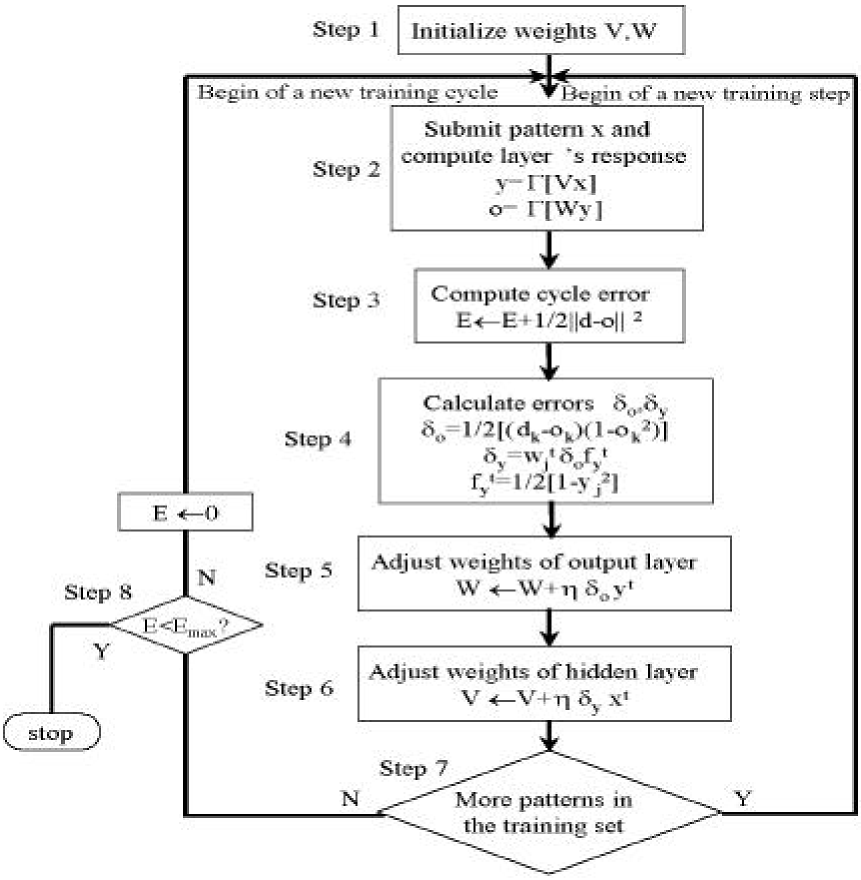


Fig. 5 Error back propagation training algorithm

Given are P training pairs, $\{x_1, d_1, x_2, d_2, \cdots, x_p, d_p\}$, where x_i is $(i \times 1)$, d_i is $(K \times 1)$, and $i = 1, 2, \cdots, P$. The operator Γ is a nonlinear diagonal operator with diagonal elements being identical activation functions. The learning begins with the feedforward recall phase(step 2). After a single pattern vector x is submitted at the input, the layers' responses y and o are computed in this phase. Then, the error signal computation phase(step 4) follows. Note that the error signal vector must be determined in the output layer first, and then it is propagated toward the network input nodes. The weights are subsequently adjusted within the matrix W, V in step 5, 6. Note that the cumulative cycle error of input to output mapping is computed in step 3 as a sum over all continuous output errors in the entire training set. The final error value for the entire training cycle is calculated after each completed pass through the training set $\{x_1, x_2, \cdots, x_p\}$. The learning procedure stops when the final error value below the upper bound, E_{max} is obtained as shown in step 8.

Also, weight adjustment use momentum method in this paper as shown Fig. 6. The purpose of the momentum method is to accelerate the convergence the error back propagation learning algorithm. This is usually done according to the formula (2.2).

$$\Delta w(t) = -\eta \frac{\partial E}{\partial w(t)} + \alpha \Delta w(t-1) \qquad (2.2)$$

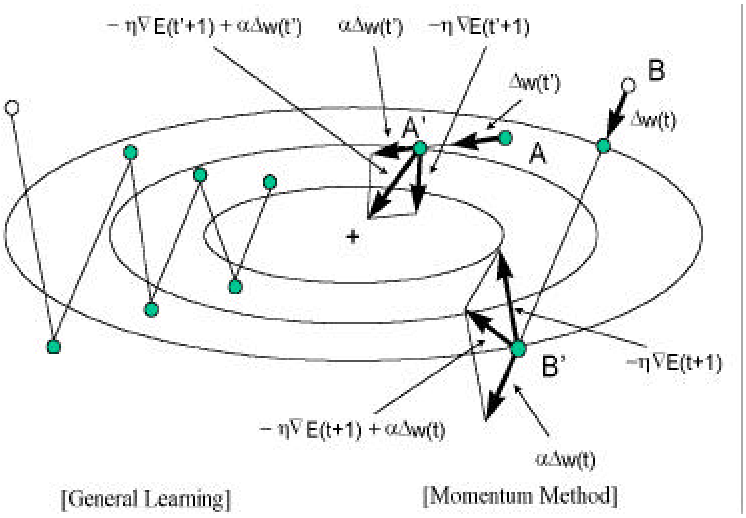


Fig. 6 Illustration of adding the momentum term in error back propagation training for a two-dimensional case

where the arguments t and t-1 are used to indicate the current and the most recent training step, respectively, and alpha is a user-selected positive momentum constant. Fig. 6 illustrates the momentum term heuristics and provides the justification for its use. Let us initiate the gradient descent procedure at point A'. The consecutive derivatives $\partial E/\partial w_1$ and $\partial E/\partial w_2$ at training points A', A'', ⋯ , are of the same sign. Obviously, combining the gradient components of several adjacent steps would result in convergence speed-up. After starting the gradient descent procedure at B', the two derivatives $\partial E/\partial w_1$ and $\partial E/\partial w_2$, initially negative at B', both alter their signs at B'', The figure indicates that the negative gradient does not provide an efficient direction of weight adjustment because the desired displacement from B'' should be more toward the minimum M, or move the weight vector along the valley rather than across it.

2.4 Multilayered Neural Networks used Experimental

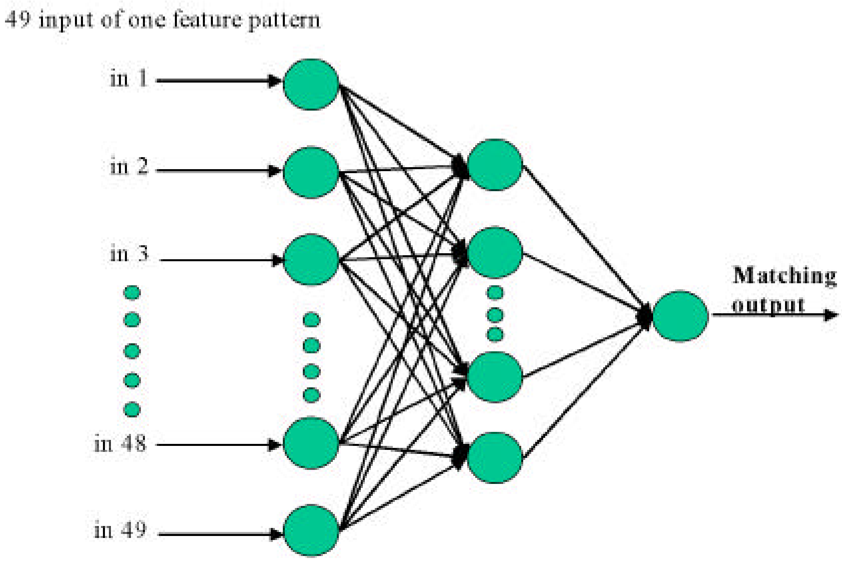


Fig. 25 Multilayer neural networks used for matching system

The proposed neural networks has the capability of excellent pattern identification. The number of input node neurons is fifty including bias, hidden node is fourteen, output node is one. The used algorithm is error back propagation algorithm in general multilayer neural networks. The proposed neural networks were learned until error become 0.01.

Chapter 3. Fingerprint Recognition

3.1 Direction Feature Vector Detection

Let I be an a×b gray-scale image with gl gray levels, and gray(i,j) be the gray level of pixel (i,j) of I, i = 1,....a, j = 1,...b. Let z = S(i,j) be the discrete surface corresponding to the image I: S(i,j) = gray(i,j), i = 1,...a, j = 1,...b. By associating bright pixels with gray levels near to 0 and dark pixels with gray levels near to gl-1, the fingerprint ridge lines (appearing dark in I) correspond to surface ridges, and the spaces between the ridge lines (appearing bright in I) correspond to surface ravines(Fig. 7).

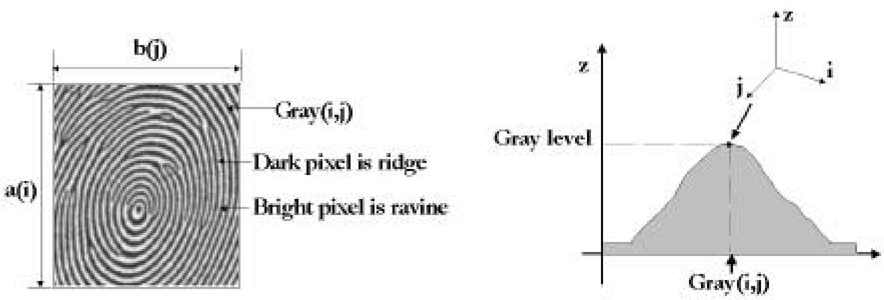


Fig. 7 a×b gray-scale fingerprint image

From a mathematical point of view, a ridge line is defined as a set of points which are local maxima along one direction. The ridge-line extraction algorithm attempts to locate, at each step, a local maximum relative to a section orthogonal to the ridge direction. By connecting the consecutive maxima, a polygonal approximation of the ridge line can be obtained

Let (i_s, j_s) be a local maximum of a ridge line of I, and φ_0 be the direction of the tangent to the ridge-line in (i_s, j_s); a pseudo-code version of the ridge-line following algorithm is ：

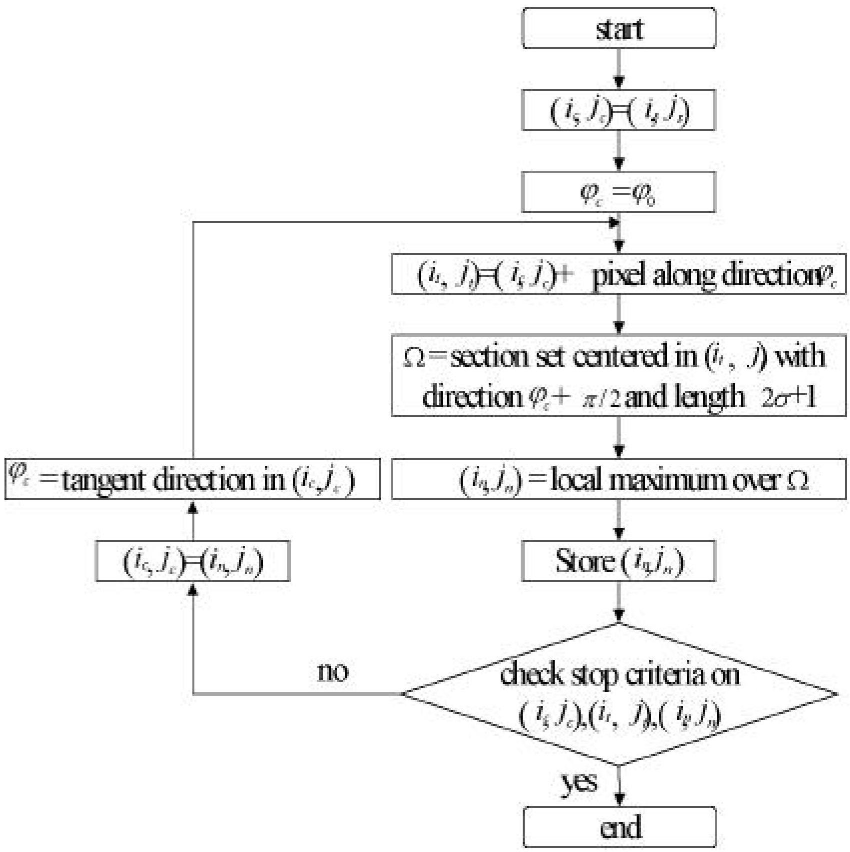


Fig. 8 Some ridge-line following algorithm

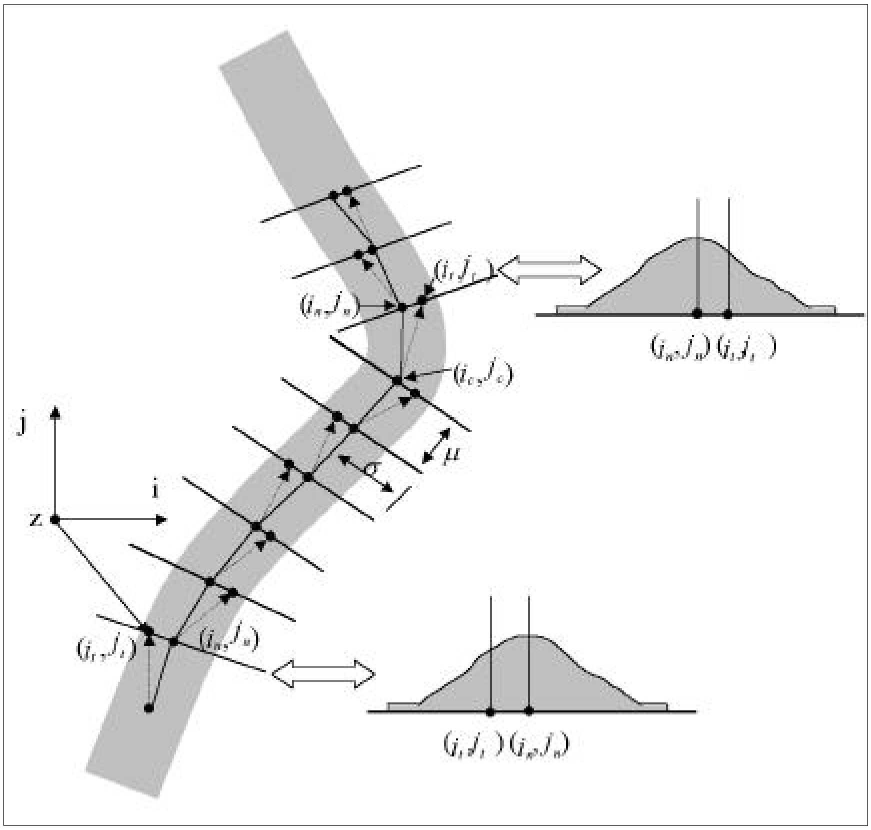


Fig. 9 Some ridge-line following steps; on the right, some sections are shown

The alogorithm runs until a stop criterion becomes true. At each step, it computes a point (i_t, j_t), moving μ pixels from (i_c, j_c) along direction φ_c. Then, it computes the section set Ω as the set of points belonging to the section segment lying on the ij-plane and having median point (i_t, j_t), direction orthogonal to φ_c and length $2\sigma+1$. A new point (i_n, j_n), belonging to the ridge line, is chosen among the local maxima of the set Ω. The point (i_n, j_n) becomes the current point (i_c, j_c) and a new direction φ_c is computed (Fig. 9). μ and σ are parameters whose optimal value can be determined according to the average thickness of the image ridge lines. The main algorithm steps, namely, sectioning and maximum determination, computation of the direction φ_c and testing of the stop criteria, are discussed in the following sub-sections.

3.2 Tangent Direction Computation

At each step, the algorithm computes a new point (i_t, j_t) by moving μ pixels from the current point (i_c, j_c) along direction φ_c. The direction φ_c represents the ridge line local direction and can be computed as the tangent to the ridge in the point (i_c, j_c).

Several methods for estimating image directional information have been proposed in the literature. The simplest approach is based on gradient computation. It is well known that the gradient phase angle denotes the direction of the intensity maximum change. Therefore, the direction φ_c of a hypothetical edge which crosses the region centered in pixel (i_c, j_c) is orthogonal to the gradient phase angle in (i_c, j_c). This method, although simple and efficient, suffers from the non-linearity due to the computation of the gradient phase angle.

Kawagoe and Tojo[11], in their work, use a different method. For each 2×2 pixel neighborhood, they make a straight comparison against four edge templates to extract a rough directional estimate, which is then arithmetically averaged over a larger region to obtain a more accurate estimate. Stock and Swonger[12], Mehtre, et al.[2], following similar approaches, evaluate the tangent direction on the basis of pixel alignments relative to a fixed number of reference directions.

The method used in this work, proposed by Donahue and Rokhlin[13], uses a gradient type operator to extract a directional estimate from each 2×2 pixel neighborhood, which is then averaged over a local window by least-squares minimization to control noise. In Appendix A, the basic steps of this method are described; more details can be found in[13]. This method allows for an unoriented direction to be computed. The computation of an oriented direction is subordinate to the choice of an orientation. For each step of the ridge line following, we choose the orientation in such a way that φ_c comes closest to the direction computed at the previous step. The technique used to compute the tangent directions, although rather efficient and robust, can become computationally expensive if the local windows used are large (if their side is 19 or more pixels) and the number of directions to be computed is very high. A more efficient implementation schema can be obtained by pre-computing the directional image over a discrete grid (Fig. 10) and then determining the direction φ_c through Lagrangian interpolation.

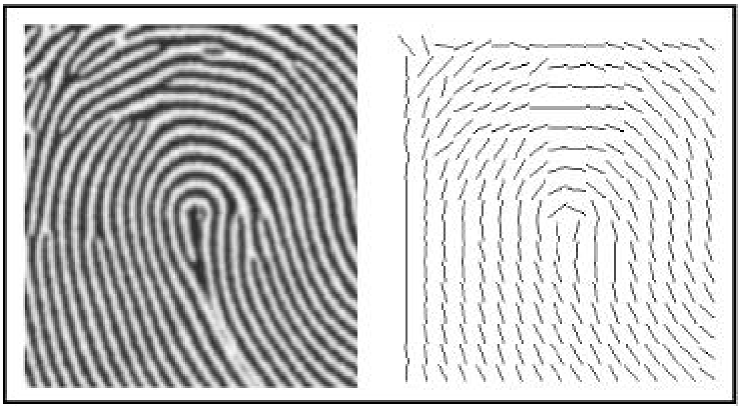


Fig. 10 A fingerprint and the corresponding directional image computed over a grid whose granularity is nine pixels.

3.3 Four Direction Labeling and Pattern Detection

I shall begin with four direction labeling. This algorithm steps, a various direction feature vectors of 360° are changed four direction labeling. In principle, each vector is computed simply by determining conditional ; using an angle value of the direction feature vector. Fig. 11 shows the coordinates which are four direction labeling. Labeling of coordinates, 0° = direct1, 45° = direct2, 90° = direct3, 135° = direct4. The direct1 is the direction feature vectors of 0° 22.4° or 157.5° 180°. The direct2 is the direction feature vectors of 22.5° 67.4°. The direct3 is the direction feature vectors of 67.5° 112.4°. The direct4 is the direction feature vectors of 112.5° 157.4°.

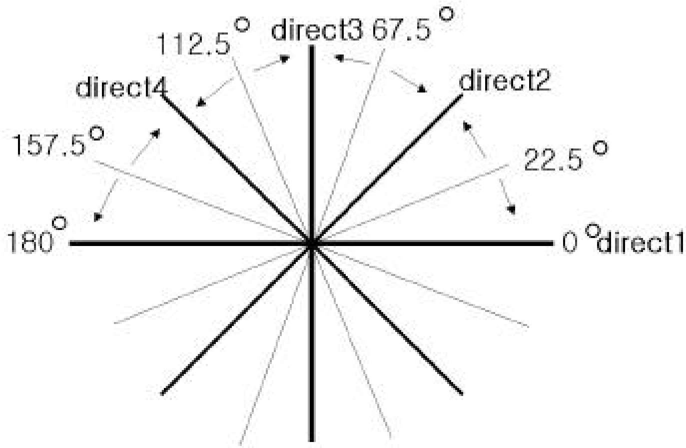


Fig. 11 Four direction labeling coordinates

The four direction labeling algorithm is:

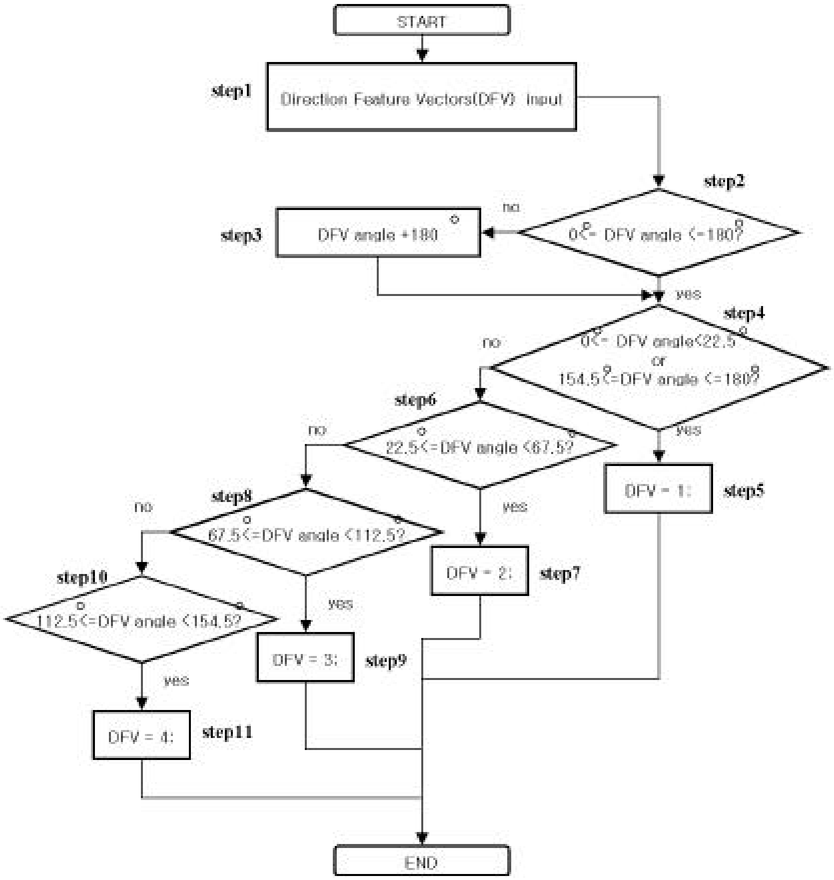


Fig. 12 Four direction labeling algorithm

The first step, it is input the direction feature vector. The step2, When it is 0° <= direction feature vector angle<=180°, the step4 progress. Otherwise, Add 180° to an angle of the direction feature vector(step3). The step4, When it is 0° <= direction feature vector angle< 22.5° or 154.5° <= direction feature vector angle<= 180° , the step5 progress, labeling (the direction feature vector = 1). Otherwise, the stpe6 progress. The step6, When it is 22.5 ° <= direction feature vector angle<= 67.5 ° , the step7 progress, labeling (the direction feature vector = 2). Otherwise, the stpe8 progress. The step8, When it is 67.5 ° <= direction feature vector angle<= 112.5 ° , the step9 progress, labeling (the direction feature vector = 3). Otherwise, the stpe10 progress. The step10, When it is 112.5 ° <= direction feature vector angle<= 154.5 ° , the step11 progress, labeling (the direction feature vector = 4). Fig. 13 shows four direction labeling image.

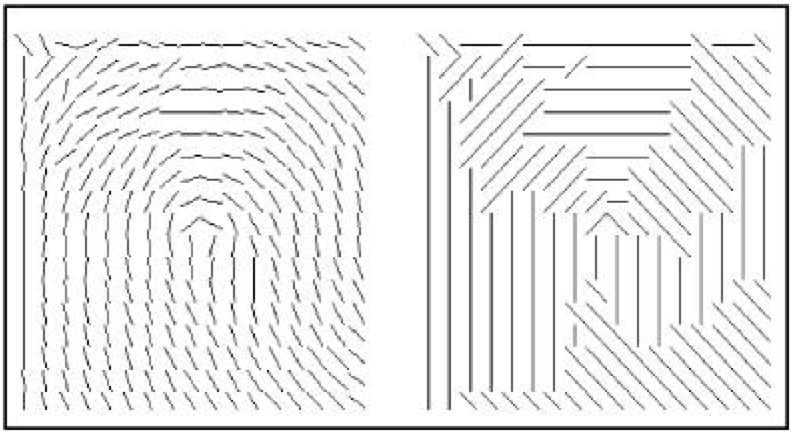


Fig. 13 Four direction labeling on a fingerprint.

In this explained, making fingerprint feature pattern of four direction labeling. A fingerprint image is divided on blocks the size of 15×15 pixels. At each block is labeling. Let 128×128 fingerprint image is consisted of 49 blocks. At each blocks, the direction vector is expressed label value(Fig. 14). All the blocks are consisted of label values. Therefore, a fingerprint image is built up of feature vector pattern using 49 direction label value.

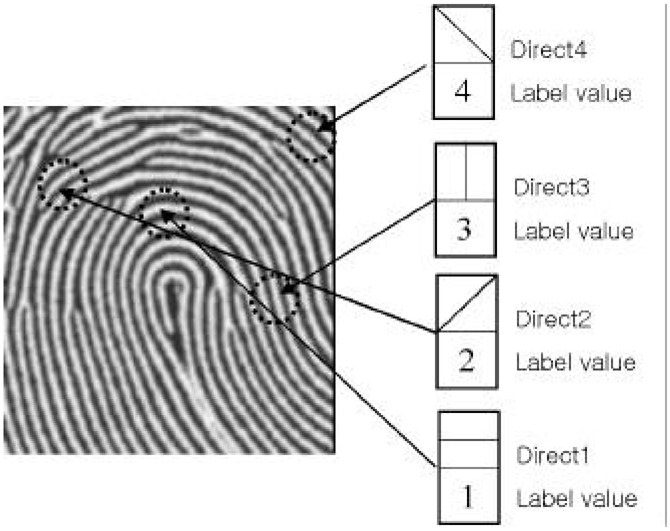


Fig. 14 A fingerprint image show label value.

Chapter 4. Experimental Results

4.1 Experimental Environment and Method

In this section, the used data makes 124 feature patterns from four fingerprint images(whorl, arch, right loop, left loop). Each fingerprint images is presented as a 128×128 image with 256 gray levels. Fig. 15 shows four samples.

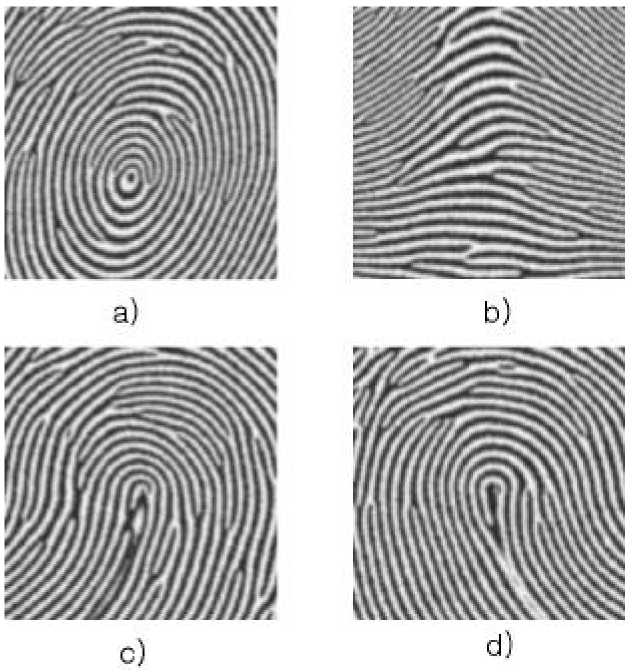


Fig. 15 Experimental fingerprint images.
[a) whorl, b) arch, c) left loop d) right loop]

The performance of experiment is executed which consist feature pattern of 49 label value in each fingerprint images, and a feature pattern is presented an angle of between -15 degrees and 15 degrees(Fig. 16, Fig. 17). Here, a feature pattern is consisted of 31 patterns(Fig. 17).

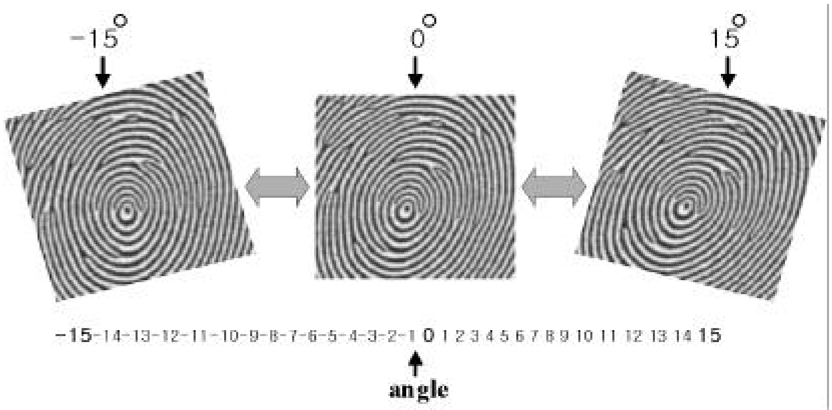


Fig. 16 A fingerprint images an angle of between -15 degrees and 15 degrees.

In experimental method is;

1. Classify learning data among 31 feature pattern ; this experiment is used learning data the feature pattern of an even angles(0°,2°,4°,6°,8°,10°,12°,14°,-2°,-4°,-6°, -8°,-10°,-12°,-14°).
2. Learning neural networks using 60 feature pattern ; each fingerprint images are presented 15 feature pattern of an even angles. Here, neural networks learning which is registered labeling each fingerprint images for numbers. In experiment, whorl registered to number1, arch registered to number2,

right loop registered to number3, left loop registered to number4.

3. Matching input the feature pattern of an odd angles(1°,3°, 5°,7°,9°,11°,13°,15°,-1°,-3°,-5°,-7°,-9°,-11°, -13°,-15°) not learning of each fingerprint images. ; matching result shows Fig. 26, 27.

Table 1. shows experimental environment.

Table 1.

Program running and Neural Networks learning	IBM PC Pentium II
Programing language	Borland C++ MFC 6.0
Fingerprint images purchase	VeriFinger v3.3

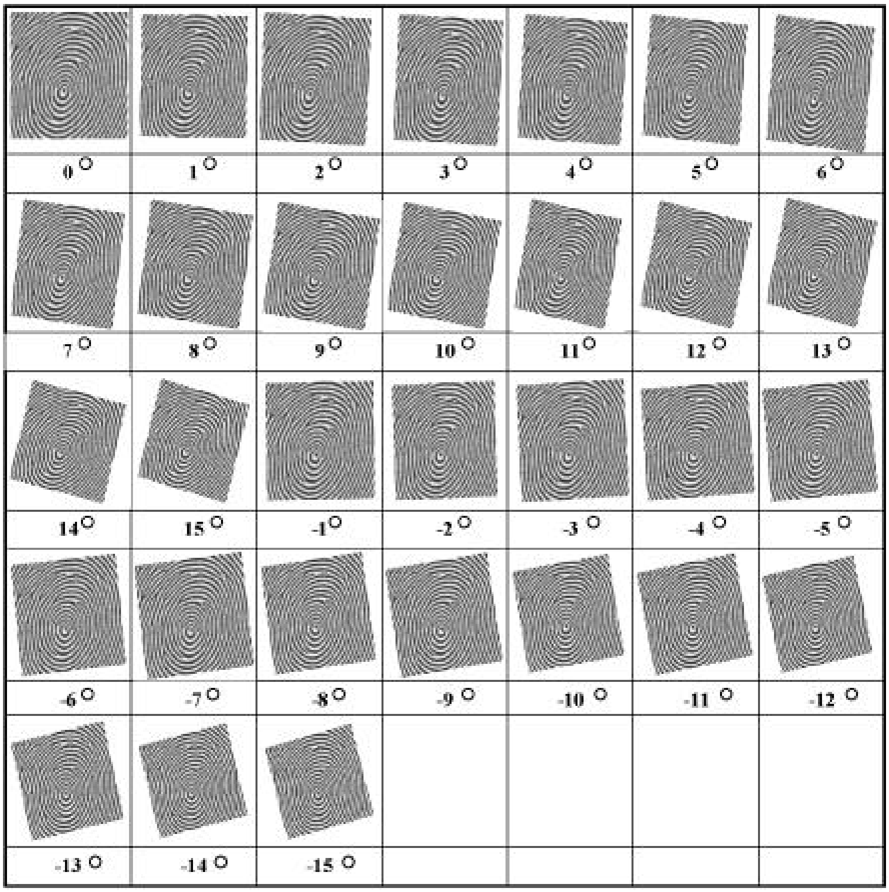


Fig. 17 Experimental 31 fingerprint images.

4.2 Experimental Results

In experimental, preference step1, four fingerprint images are detected as various direction feature vectors(Fig. 19), and step2, a various direction feature vectors are changed Four direction feature vectors(Fig. 20), and step3, the direction feature vectors are labeling, and step4, registered for matching system(neural networks) labeling each fingerprint images for number; in experimental, whorl registered to number1, arch registered to number2, right loop registered to number3, left loop registered to number4. Step5, Matching experimental using label feature patterns of each fingerprints. Fig 21, 22, 23, 24 shows label feature patterns.

Fig. 25 shows neural networks using matching experimental. Fig. 26 and Fig. 27 shows matching results. As shows experimental results is presented very good capability.

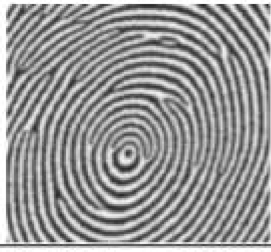
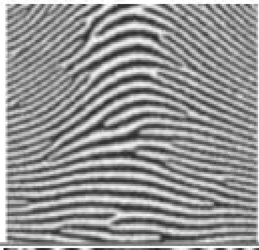
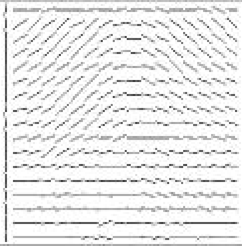
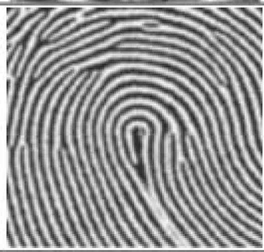
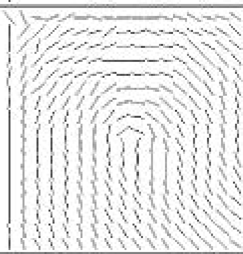
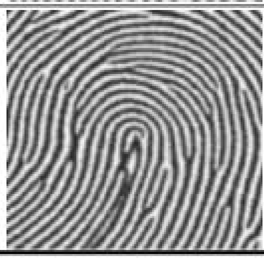
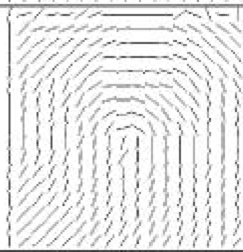
	Original	Various angle
Whorl		
Arch		
Right loop		
Left loop		

Fig. 19 Four fingerprint images are detected various direction feature vectors

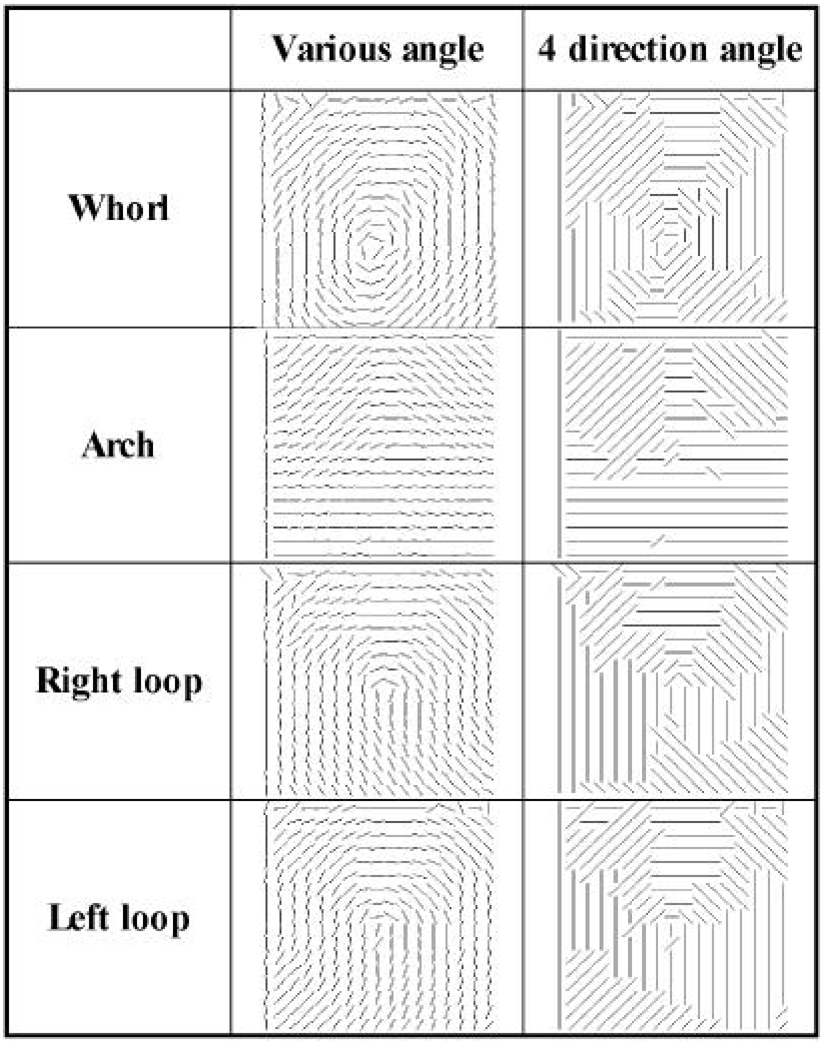


Fig. 20 A various direction feature vectors are changed four direction feature vectors

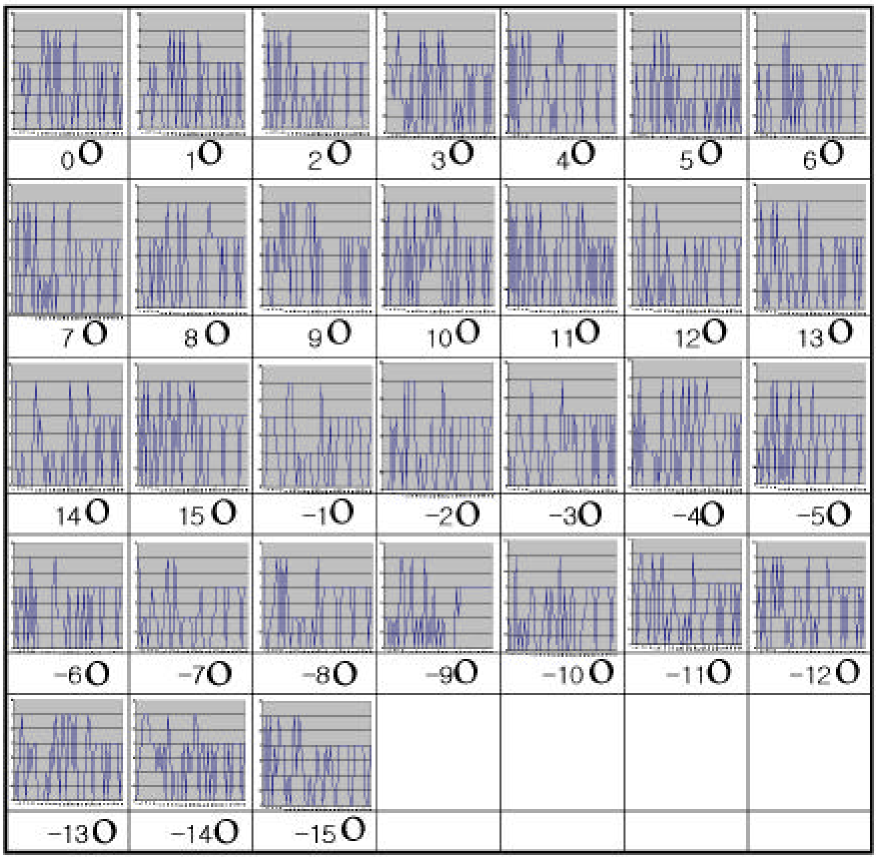


Fig. 21 Feature patterns of whorl

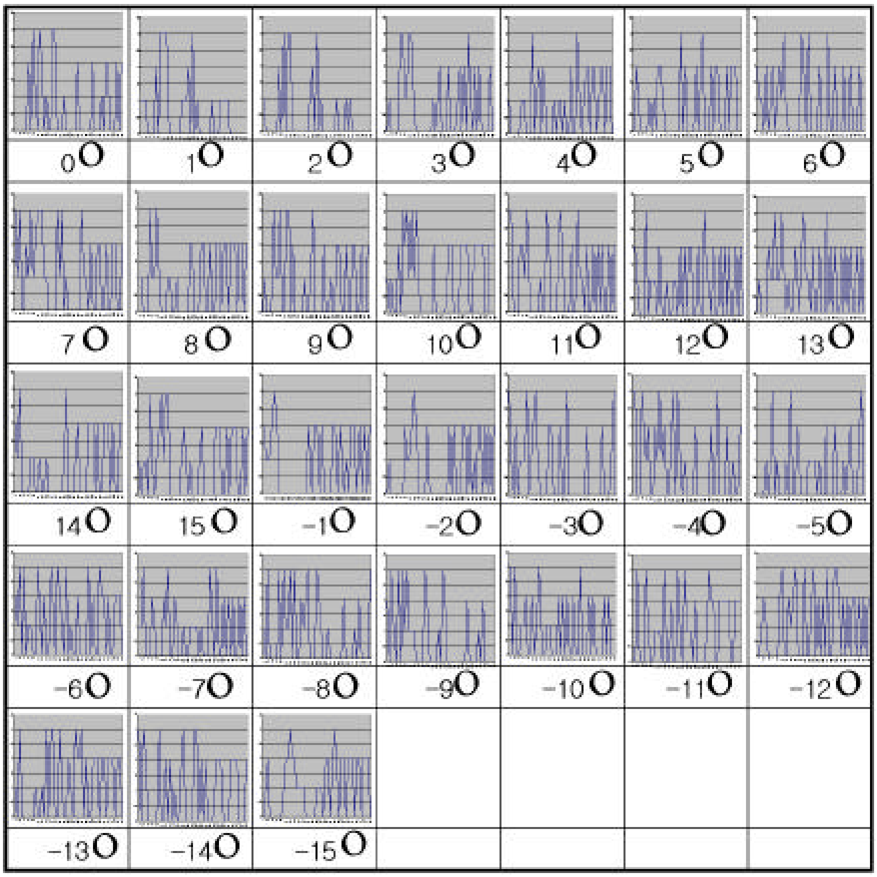


Fig. 22　Feature patterns of arch

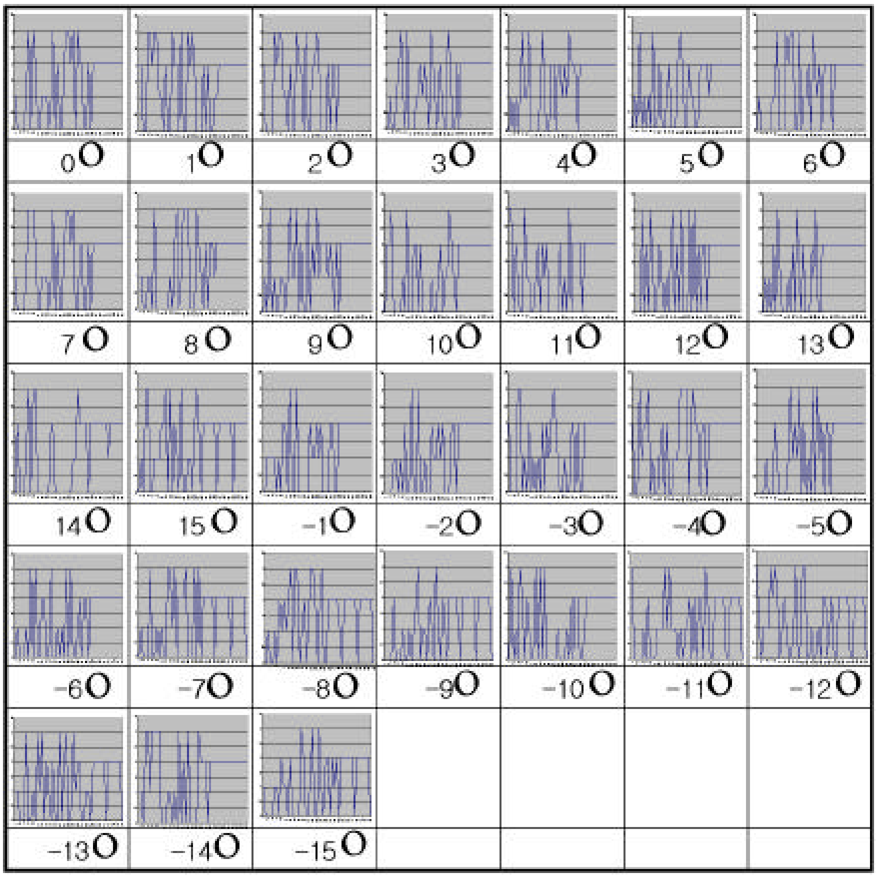


Fig. 23 Feature patterns of right loop

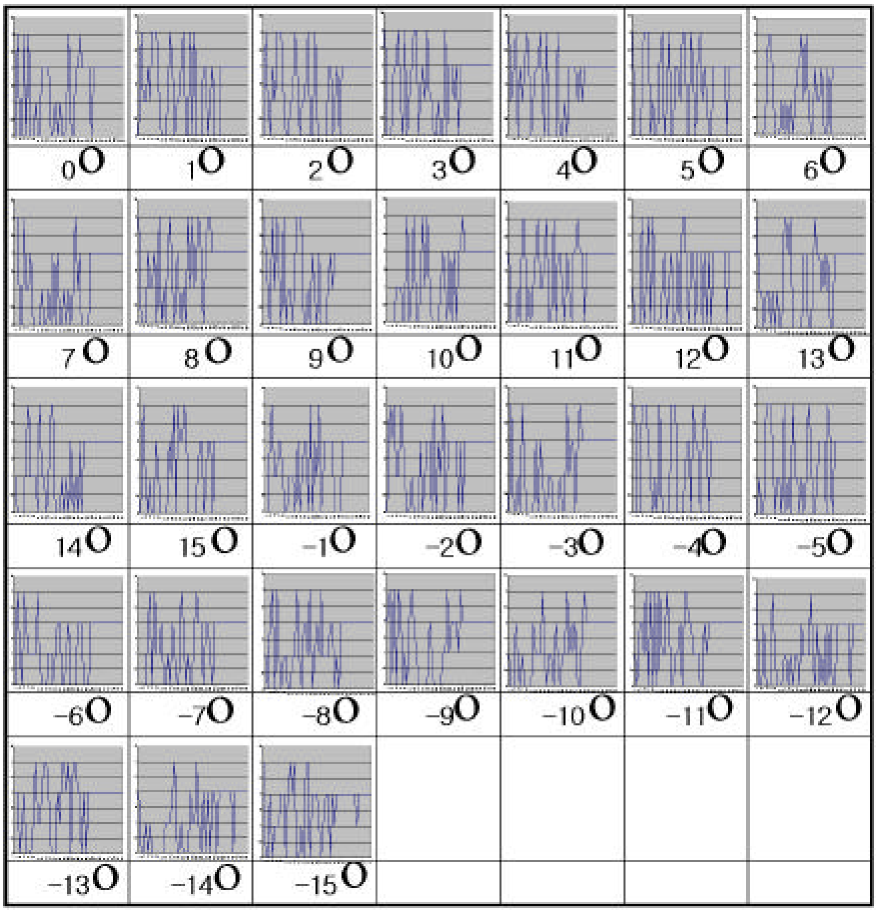
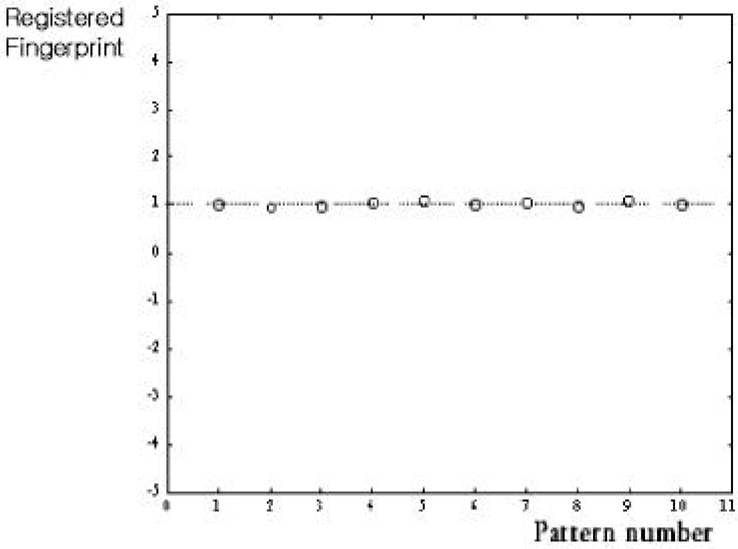


Fig. 24 Feature patterns of left loop

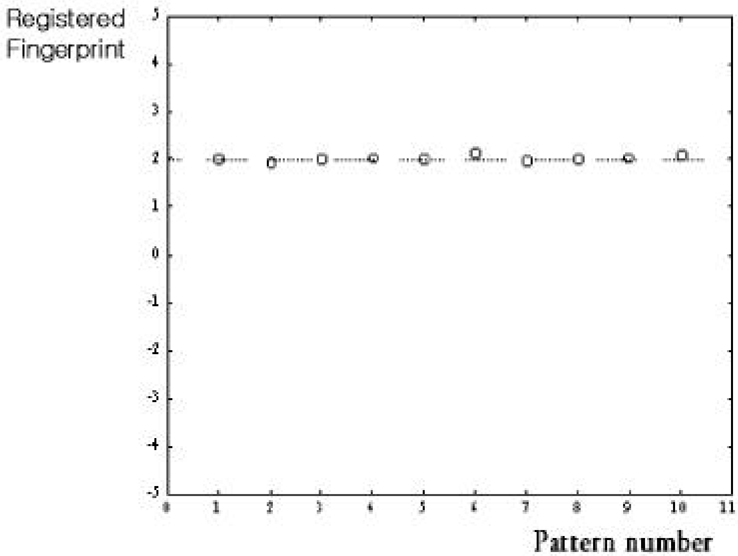
In experiment are used total 124 feature patterns of four fingerprints. One fingerprint image is consisted of 31 feature patterns. Each fingerprint images are learned fifteen patterns of an even angles, and matching ten patterns of an odd angles. The results shows Fig. 26 and 27. Fig. 26-(a) shows which is Registered labeling whorl image for number1, and Fig. 26-(b) shows which is Registered labeling arch image for number2, and Fig. 27-(c) shows which is Registered labeling right loop image for number3, and Fig 27-(d) shows which is registered labeling left loop image for number4. In experimental results shows very excellent identification capability irrespective of angle transformation. Input patterns using in experiment shows table 2.

Table 2.

Fingerprint \ Input number	1	2	3	4	5	6	7	8	9	10
Whorl(number 1)	3°	5°	7°	-3°	-9°	11°	15°	-15°	-7°	13°
Arch(number 2)	-9°	-13°	5°	3°	7°	-15°	-11°	-3°	-1°	9°
Right loop(number 3)	1°	9°	7°	11°	3°	-3°	13°	-13°	15°	-9°
Left loop(number 4)	-15°	-7°	-9°	-3°	-5°	13°	5°	7°	-11°	3°

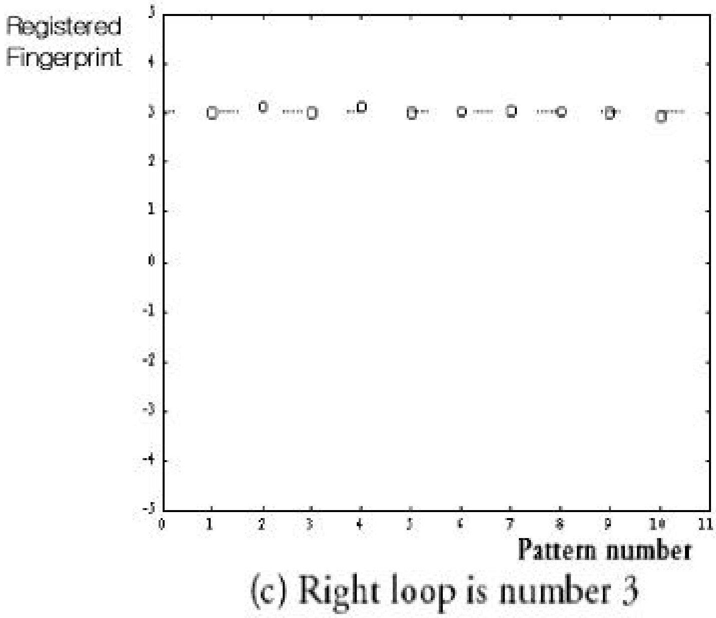


(a) Whorl is number 1



(b) Arch is number 2

Fig. 26 Results for feature pattern matching



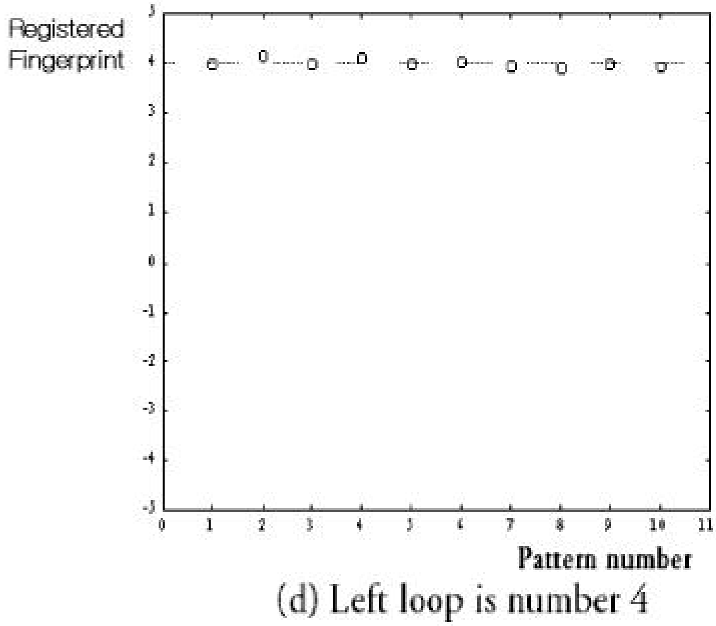


Fig. 27 Results for feature pattern matching

Chapter 5. Conclusion

In this paper we have presented approach to automatic the direction feature vectors detection, which detects the ridge line directly in gray scale images.

In spite of a greater conceptual complexity, we have shown that our technique has less computational complexity than the complexity of the techniques which require binarization and thinning. And a various direction feature vectors are changed four direction feature vectors. In this paper used matching method is four direction feature vectors based matching.

This four direction feature vectors consist feature patterns in fingerprint images. This feature patterns were used for identification of individuals inputed multilayer Neural Networks(NN) which has capability of excellent pattern identification.

In experimental results is presented very good capability. In the future work, in order to reduce error rate mistaken identification, have to continue research, and apply actual automatic systems for fingerprint comparison.

References

[1] Baldi, P. and Chauvin, Y. “Neural Networks for Fingerprint Recognition," Neural Computation, 5, pp. 402-418, 1993.

[2] B. M. Mehtre, N.N. Murthy, S. Kapoor, and B. Chatterjee, “Segmentation of Fingerprint Images Using the Directional Image," Pattern Recognition, vol. 20, No. 4, pp. 429-435, 1987.

[3] Demetri Psaltis et al., "A Multilayered Neural Network Controller", IEEE Control System Magazine, Vol. 8, pp. 17-21, April 1988.

[4] K. Hornik, M. Stinchcombe, H. White, "Multilayer feedforward networks are universal approximators", Dept. Economics, University of California, San Diego, CA, Discussion Pap., June 1988.

[5] J. M. Zurada, *Introduction to Artificial Neural Systems*, West Publishing Company, 1992.

[6] Simon Haykin, *Neural Networks*, Macmillan Company, 1994

[7] S. Omatu, M. Khalid, R. Yusof, *Neuro-Control and its Applications*, springer, 1995.

[8] J. S. R. Jang, C. T. Sun, E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice Hall, 1997.

[9] K. S. Narendra, K. Parthasarathy, "Neural Networks and Dynamical Systems. Part I: A Gradient Approach to Hopfield Networks", Center Syst. Sci., Dept. Electrical Eng., Yale University, New Haver, CT, Tech. rep. 8820, October 1988.

[10] L. Ljung and J. Sjoberg, "A System Identification Perspective on Neural Nets", in Neural Networks for Signal

Processing II, Proceedings of the 1992 IEEE-SP Workshop, pp. 423-435, 1992.
[11] M. Kawagoe and A. Tojo, “Fingerprint Pattern Classification," Pattern Recognition vol. 17, no. 3, pp. 295-303, 1984.
[12] R.M. Stock and C. W. Swonger, “Development and Evaluation of a Reader of Fingerprint Minutiae," Cornell Aeronautical Laboratory, Technical Report CAL No. XM-2478-X-1:13-17, 1969.
[13] Donahue, M. J. and Rokhlin, S. I. “On the Use of Level Curves in Image Analysis," Image Understanding, Vol. 57, No. 2, pp. 185-203, 1993.
[14] Akhan, M.B. and Emiroglu, I. “A Flexible Matching Algorithm for Fingerprint Identification System," Proc. The Tenth International Symposium on Computer and Information Sciences, Izmir, Turkey, pp. 806-815, 1995.
[15] Clarke, R. “Human Identification in Information Systems: Management Challenges and Public Policy Issues," Info. Technol. People, 7(4), pp.6-37, 1994.
[16] Mehtre, B.M. “Fingerprint Image Analysis for Automatic Identification," Machine Vision and Applications, Vol. 6, No. 2-3, 1993.
[17] O'Gorman, L. and Nickerson, J. V. “An approach to fingerprint filter design," Pattern Recognition, Vol. 22, No. 1, pp. 29-38, 1989.
[18] **김관형**, **김태영**, **이상배**, “**퍼지-뉴럴 알고리즘을 이용한 효과적인 용접제어시스템에 관한 연구**”, 1997**년도 한국퍼지 및 지능시스템학회 논문집**, Vol. 7, pp. 189-193.
[19] **이재현**, **강성인**, **이상배**, “**신경회로망을 이용한 동적 시스템의 상태 공간 인식 모델에 관한 연구**”, 1997**년 한국퍼지 및 지능시스템 학회 논문집**, Vol 7, pp.115-120.

[20] **강성인**, **김관형**, **탁한호**, **이상배**, ,“**신경회로망을 이용한 용접 갭 검출과 실시간 모니터링에 관한 연구**” 1998**년도 한국퍼지 및 지능 시스템학회 논문집** Vol. 8, No. 2, pp. 539-544.

[21] **김태영**, **김관형**, **김일**, **이상배** “**신경회로망의 패턴분류를 이용한 용접 갭 검출에 관한 연구**”, 1998**년도 한국퍼지 및 지능 시스템학회 논문집** Vol. 8, No. 2, pp. 527-531.

감사의 글

2년여의 시간동안 여전히 함께 하셨던 임마누엘의 하나님을 찬양합니다.

논문을 완성하기까지 세심한 배려와 관심으로 지도해주셨던 이상배 지도교수님께 진심으로 감사드리고, 바쁘신 와중에도 성심껏 심사해주신 홍창희 교수님, 양규식 교수님께 감사드립니다.

멀리서 지켜봐 주신 경상대학교의 박연식 교수님, 이상욱 교수님, 김청 교수님, 이종룡 교수님, 전성근 교수님, 성길영 교수님, 이우재 교수님께도 감사의 말씀을 올립니다.

물심양면으로 지원해 주신 김일, 탁한호 선생님을 비롯한 졸업하신 여러 선배님들께도 감사드립니다. 시원한 바닷바람과 탁트인 해안풍경을 바라보며 2년동안 동고동락했던 선배님들과 내가 힘들고 지칠 때마다 함께 힘이 되어 주었던 사랑하는 동기들인 불타는 고구마 창우형과 날뛰는 망아지 동민이형, 그리고 박사같은 석사인 빈대 석민이형에게도 깊은 우정과 감사를 드립니다. 언제나 나에 대한 깊은 신뢰와 변함없는 우정을 주는 사랑하는 친구들과 철한이형, 순웅이에게도 감사를 드립니다. 같은 주제를 가지고 함께 공감하며 나누었던 영상처리연구실의 이동기씨에게도 감사드리고, 많은 시간들을 함께 지내 주었던 미라씨에게도 감사를 드립니다.

나를 신앙으로 지도해 주신 손기현 목사님과 관심과 격려로 도움을 주셨던 여러 집사님들과 사랑과 믿음으로 함께한 청학교회 청년회 지체들에게 감사를 드립니다.

마지막으로, 언제나 자식의 행복과 발전을 위해 어려움을 감추시고 평생 사랑으로 헌신하신 세상에서 내가 가장 사랑하는 아버님, 어머님의 크신 은혜와 사랑에 머리 숙여 깊이 감사드리고, 사랑하는 형님 두분과 형수님 두분, 그리고 우리 이쁜이 조카들을 비롯한 모든 가족들에게 이 작은 결실을 바칩니다.

대학원에서의 2년이라는 세월은
지금까지 살아왔던 어떤 시간보다도
가치있고 힘들었던 시간이었다.

푸른 바다를 바라보며 살아었기에,
마음의 가슴앓이를 하지 않았던 것 같다.

누구보다도 소중한 동기들과 아웅다웅하면서
때론 서로의 다툼으로 마음도 상했었지만,
지금은 아름다운 추억으로 간직되어 진다.

'찰나'라는 말처럼 너무나도 빠르게
2년이 지나간 것 같다.

내 인생에서 가장 소중한 가치를
깨닫게 해준 2년의 시간이
내게는 너무나도 소중한 경험이었다.