

การสกัดลักษณะของภาพที่ไม่แปรเปลี่ยนไปตามความเข้มสี การหมุนและการย่อขยาย โดยใช้โครงข่ายประสาท
ชนิดจัดกลุ่มเองที่อาศัยแนวทางไอเคนเวกเตอร์



นางสาวกิงกาญจน์ สุขคนาภิบาล

สถาบันวิทยบริการ

จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรดุษฎีบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย


ปีการศึกษา 2546

ISBN 974-17-4799-3

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

IMAGE FEATURE EXTRACTION INVARIANT TO COLOR INTENSITY, ROTATION AND SCALING
USING EIGENVECTOR-GUIDED SELF-ORGANIZING MAPPING NEURAL NETWORK

Miss Kingkarn Sookhanaphibarn



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in Computer Science

Department of Mathematics

Faculty of Science

Chulalongkorn University

Academic year 2003

ISBN 974-17-4799-3

Thesis Title IMAGE FEATURE EXTRACTION INVARIANT TO COLOR
INTENSITY, ROTATION, AND SCALING USING EIGENVECTOR-
GUIDED SELF-ORGANIZING MAPPING NEURAL NETWORK

By Ms. Kingkarn Sookhanaphibarn

Department Mathematics

Thesis Advisor Professor Chidchanok Lursinsap, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial
Fulfillment of the Requirements for the Doctor's Degree

..... Dean of the Faculty of Science
(Professor Piamsak Menasveta, Ph.D.)

THESIS COMMITTEE

..... Chairman
(Associate Professor Jack Asavanant, Ph.D.)

..... Thesis Advisor
(Professor Chidchanok Lursinsap, Ph.D.)

..... Member
(Assistant Professor Krung Sinapiromsaran, Ph.D.)

..... Member
(Associate Professor Kosin Chamnongthai, Ph.D.)

..... Member
(Assistant Professor Thitipong Tanprasert, Ph.D.)

..... Member
(Churarat Tanprasert, Ph.D.)

..... Member
(Siripun Sanguansintukul, Ph.D.)

กึ่งกาญจน์ สุขคนาภิบาล : การสกัดลักษณะเด่นของภาพที่ไม่แปรเปลี่ยนไปตามความเข้มของสี การหมุนและการย่อขยาย โดยใช้โครงข่ายประสาทชนิดจัดกลุ่มเองที่อาศัยแนวทางไอเกนเวกเตอร์. (IMAGE FEATURE EXTRACTION INVARIANT TO COLOR INTENSITY, ROTATION AND SCALING USING EIGENVECTOR-GUIDED SELF-ORGANIZING MAPPING NEURAL NETWORK) อ. ที่ปรึกษา : ศาสตราจารย์ ดร. ชิดชนก เหลือสินทรัพย์, 100 หน้า. ISBN 974-17-4799-3.

วิทยานิพนธ์ฉบับนี้เสนอขั้นตอนวิธีในการสกัดคุณลักษณะเด่นของภาพที่ไม่แปรเปลี่ยนไปตามความเข้มของสี การหมุนและการย่อขยายของเนื้อหาภาพโดยใช้โครงข่ายประสาทชนิดจัดกลุ่มเองที่อาศัยแนวทางไอเกนเวกเตอร์ (Eigenvector) การสกัดคุณลักษณะเด่นของภาพใช้หลักการวิเคราะห์แกนประกอบหลัก (PCA) และขั้นตอนการเรียนรู้แบบแข่งขัน แต่การวิเคราะห์แกนประกอบหลักไม่สามารถหาทิศการหมุนของภาพได้ถูกต้องถ้าการหมุนของภาพเกิน 180 องศา และวิธีการเรียนรู้แบบแข่งขันช่วยในการสกัดคุณลักษณะเด่นของภาพโดยอาศัยตำแหน่งของเวกเตอร์ถ่วงน้ำหนัก วิธีการที่นำเสนอสามารถประยุกต์ใช้กับภาพสีที่มีขนาดอย่างน้อย 256x256 พิกเซล การทดลองด้วยการใช้วิธีการที่นำเสนอ ได้ทดสอบกับภาพสีทั้งหมด 34 ภาพ บ้างมีรูปร่างที่เหมือนกันแต่ต่างกันที่ลายภาพสี บ้างมีส่วนของสีในภาพเหมือนกันแต่ต่างกันที่รูปร่าง ผลการทดลองแสดงให้เห็นถึงประสิทธิภาพในการสกัดคุณลักษณะเด่นของภาพที่ไม่แปรเปลี่ยนไปตามความเข้มของสี การหมุนและการย่อขยายได้เป็นอย่างดี

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา	คณิตศาสตร์	ลายมือชื่อนิสิต.....
สาขาวิชา	วิทยาการคอมพิวเตอร์	ลายมือชื่ออาจารย์ที่ปรึกษา.....
ปีการศึกษา	2546	

4273839623 : MAJOR COMPUTER SCIENCE

KEY WORD: FEATURE EXTRACTION / INVARIANCE / SELF-ORGANIZING MAP (SOM) / SCALING AND ROTATION / COLOR IMAGE.

KINGKARN SOOKHANAPHIBARN: IMAGE FEATURE EXTRACTION INVARIANT TO COLOR INTENSITY, ROTATION AND SCALING USING EIGENVECTOR-GUIDED SELF-ORGANIZING MAPPING NEURAL NETWORK, THESIS ADVISOR: PROFESSOR CHIDCHANOK LURSINSAP, Ph.D., 100 pp. ISBN 974-17-4799-3.

This paper proposes a new feature extraction algorithm to extract the invariant features of an image based on the concept of Principle Component Analysis (PCA) and competitive learning algorithm. PCA assists in guiding the rotation axis of the image but it cannot find the exact rotational direction of the image if the image is rotated through an angle greater than 180 degrees. Self-partitioning competitive learning can capture the texture of an image by using the location of weight vectors and it embeds the invariant ability of scaling. The proposed algorithm is developed to apply to colored-texture images of size at least 256x256 pixels. In addition to scaling and rotation invariance, color intensity change are addressed in the paper. The experiment data contains 34 color-textured images including images having the same shape but different colored texture and having the similar color distribution but different shape. Experimental results show that our new algorithm performs the discriminating capability of images having the same shape but different colored texture and also reserves the invariant ability.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Department **Mathematics**

Student's signature.....

Field of study **Computer Science**

Advisor's signature.....

Academic year **2003**

Acknowledgments

This dissertation is the result of five years of work whereby I have been accompanied and supported by many people. The first person I would like to thank is my supervisor Professor Chidchanok Lursinsap. His overt enthusiasm and integral view on research have made a deep impression on me. I owe him lots of gratitude for having shown me the way of research. Above all, he gave me an opportunity to pursue a Ph.D by employing me as his research assistant under the Thailand Research Fund. I would like to thank Dr. Kevin K. W. Wong who was my co-supervisor during eight months at Murdoch University, Perth. Thanks for monitoring my work and providing me with valuable comments on earlier versions of this dissertation.

Associate Professor Suchada Siripant is my beloved and admired person. She cheered me up when I was down and frustrated. I am deeply grateful her generosity. Supaporn Kamklad, Wiwat Sidhisoradej, Kodchakorn Na Nakornpanom, Anocha Rugchatjaroen, Nitass Sutaveephamochanon and Maytee Bamrungrajhirun are as close as if we were a family, and good friends to me. I am really glad that I have come to know all these people in my life.

I would like to thank Gp.Capt.Thanapant Raicharoen. I could not complete my experiment without his help on the synthesis some test data in the experiments. Special thanks to Wilsit Navawongs for looking after me as his sister during the stressful period of the qualifying examination. I would like to thank the committee for their checking and correction of the dissertation. The people have read draft versions of the dissertation: Ms. Mela Bryce, Miss Joyce Inma, Miss Inthira Trirangkura, Wg.Cdr.Sitthisak Saingoen. Lots of thanks to all.

In my opinion, doing a Ph.D is a sacred task and this was definitely one of the best decisions of my life. May I dedicate this work to my parents, Mr. Prayut and Mrs. Supatra Sookhanaphibarn. They made family purely meaning family. They have been dedicating their life to their daughters and son. I am grateful to my brother, Termetch, and sister, Nattakarn. They have always encouraged me, and fostered self-belief with their faith and love. Last, This dissertation would have been impossible without the fully financial support from TRF.

Table of Contents

Thai Abstract	iv
English Abstract	v
Acknowledgments	vi
List of Tables	ix
List of Figures	x
1 Introduction	1
1.1 Approaches of Invariant Recognition	1
1.2 Problem Formulation and Proposed Solutions	4
1.3 The Contributions of the Dissertation	4
1.4 Scope and Organization	7
2 Literature Reviews	13
2.1 Invariant Features in Pattern Recognition	13
2.2 Invariant Recognition Using Neural Networks (NN)	17
2.2.1 High-Order Neural Network (HONN)	17
2.2.2 Pulse-Coupled Neural Network (PCNN)	20
2.3 Invariant Recognition Based On Transformation Techniques	23
2.3.1 Fourier Transforms (FT)	23
2.4 Invariant Recognition based on Statistical Techniques	26
2.4.1 Moments	26
2.4.2 Co-occurrence Matrix (CM)	29
2.5 Invariant Feature by Using Fuzzy Color Histogram (FCM)	32
3 Rotational and Scaling Invariant Self-Organizing Mapping	35
3.1 Kohonen's Competitive Learning Concept	35
3.2 Rotational Direction	38
3.3 Self-Partitioning Competitive Learning	41

3.4	Feature Extractor for Color Images	45
3.4.1	Time Complexity of the Proposed Feature Extractor	50
3.4.2	Feature Extraction Based on The Location of Weight Vectors	51
4	Experimental Results	54
4.1	Test Data	54
4.2	Setting and Definitions	58
4.3	Results	59
4.3.1	Robustness of RSISOM Under Scaling	61
4.3.2	Robustness of RSISOM Under Color Intensity Changes	65
5	Conclusion	68
5.1	Invariance Capability	68
5.2	Distinguishability	69
5.3	Future Work	71
	REFERENCES	72
A	Publications	78
B	Mathematical Theory	79
C	Test Data	80
	Biography	86

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

List of Tables

1.1	A list of three formulated problems and the corresponding solutions.	5
1.2	Examples of invariant image recognition.	7
4.1	Parameter values used for all data sets.	58
4.2	The number of weight vectors used for data sets.	59
4.3	The results of feature classification of data set A are presented in classes of transformations, i.e., rotation, scaling, changing the intensity and combining all. The correctness of invariant classification is shown in the second column following the maximum, minimum and standard deviation (S.D.) of the distance between $\mathbf{f}_k^{[i]}$ and \mathbf{v}_{m_i} denoted by D_f in the third, fourth, fifth and sixth columns, respectively.	60
4.4	The average boundary distance ($\times 10^{-2}$) of each invariant class denoted by Mean of D_f	60
4.5	The distance measure ($\times 10^{-2}$) among each feature vectors of original images, airplane type \mathbf{A} , $\ \mathbf{v}_i - \mathbf{v}_j\ $	61
4.6	The distance measure ($\times 10^{-2}$) between each pair of feature vectors of original images (i,j) , $\ \mathbf{v}_i - \mathbf{v}_j\ $	62
4.7	The distance measure ($\times 10^{-2}$) between each pair of feature vectors of original images (i,j), $\ \mathbf{v}_i - \mathbf{v}_j\ $	62

List of Figures

1.1	For example, two different types of airplanes have the same colored texture. (a) Airplane type A-1 . (B) Airplane type F-1	2
1.2	For example, two images (a) and (b) with the similar color distribution appear different on the shape: squares and circles.	4
1.3	A system of automatic pattern recognition.	8
1.4	An example of an image having an object, <i>letter 'A'</i> , in the white background. (a) The prototype image. (b) The rotated ' <i>A</i> '. (c) The scaled ' <i>A</i> '.	9
1.5	An example of an image having two objects, <i>letter 'A'</i> and <i>letter 'I'</i> , in the white background. (a) The prototype image. (b) The rotated group which is formed by <i>letter 'A'</i> and <i>letter 'I'</i> . (c) The scaled group.	10
1.6	An example of a change in color intensity. (a) The prototype image. (b)-(d) Images decomposed from (a) into R, G and B planes, respectively; (e) The output image after adjusting the color intensity of R, G and B planes. (f)-(h) Images when $\alpha, \beta, \gamma = 0.59$ whereas $R_0 = 50, G_0 = 0$ and $B_0 = 100$	11
2.1	A mapping T maps the object f and its rotated object f' into the same point of the feature space.	14
2.2	An example of geometric transformations. (a) An original object. (b) The translated object. (c) The rotated object. (d) The scaled object.	15
2.3	A sample architecture of high-order neural network with four inputs, x_1, x_2, x_3, x_4	18
2.4	Two triangles with included angles (α, β, γ) . (a) Original triangle jkl . (b) A new triangle $j'k'l'$ by scaling and rotating triangle (a).	19
2.5	A computation diagram of pulse-coupled neural network according to Eq.(2.15)-(2.16).	20
2.6	A structure of pulse-coupled neural network with the input image of size $M \times N$ pixels.	22

2.7	A distant operation $d = (m, n)$ is defined that m pixels to the right and n pixels below corresponding to a solid block.	30
2.8	An example shows the co-occurrence matrix on a given image and the defined distant operation. (a) An image of size 16×16 pixels. (b) A distant operation $d = (1, 1)$, i.e., one pixels to the right and one pixel below. (c) A co-occurrence matrix after computing the image matrix (a) with the distant operation (b). .	31
3.1	A sample architecture of Kohonen's competitive network with three input and four output neurons which \mathbf{x}_k is a selected data vector at present.	36
3.2	An example shows the principle component axes of image "T" and its rotated image. Both images have the same direction of principle component axes. . . .	38
3.3	An example of the rotated image "T" illustrates the process of finding the proper direction of eigenvector of the image. (a) The eigenvector \mathbf{e} obtained directly from PCA. (b) The data vectors in C_1 and C_2 represented by dark area and gray area, respectively. (c) The proper direction of the image changed to the negative eigenvector.	41
3.4	The location of weight vectors after applying the Kohonen's competitive learning to the same data vectors in order to partition the data. (a)-(b) Performing the Kohonen's competitive learning in the first and second trials by using random initialization of weight vectors. (c)-(d) Performing the Kohonen's competitive learning in the third and fourth trials by using fixed initialization of weight vectors.	43
3.5	The location of initial weight vectors assigned by Eq. (3.15). The angle θ of \mathbf{w}_0 is equal to zero and the angle space of the adjacent weight vector, denoted by ϵ , is equal to $\frac{2\pi}{m}$ where m is the number of weight vectors.	44

3.6	Given an example of a color image “T”, there are two encoded vectors from a pixel. The first vector, $\mathbf{x}_k = [a_1 \ a_2]^T$, is the information of x -coordinate and y -coordinate at pixel k and the other, $\mathbf{i}_k = [b_1 \ b_2 \ b_3]^T$ is the information of intensities of red, green and blue colors at pixel k . The concatenation of both vectors is denoted by a new vector, $\mathbf{p}_k = [a_1 \ a_2 \ b_1 \ b_2 \ b_3]^T$ in a 5-dimensional space.	53
4.1	Dataset B consists of six airplane types, namely A-0 , A-1 , A-2 , A-3 , A-4 , and A-5 , which all are of the same model A	55
4.2	Dataset C consists of six airplane types, namely F-0 , F-1 , F-2 , F-3 , F-4 , and F-5 , which all are of the same model F	56
4.3	Dataset D has four images: (a)-(b) have similar superimposing structures and colors but different shapes, circles and squares. (c)-(d) consists of the same objects, namely circles, triangles, and squares.	57
4.4	The correctness percentage of both Zernike moments (ZM) and RSISOM when they test against the scaled data described in Section 4.1.	63
4.5	The accumulated number of images recognized correctly by both Zernike moments (ZM) and RSISOM with the parameter of scaling from 1.2 to 2.0 times of the original.	64
4.6	The correctness percentage of both fuzzy color histogram denoted by FCH and RSISOM when they test against the intensity-changed images described in Section 4.1.	66
4.7	The accumulated number of images recognized correctly by both fuzzy color histogram denoted by FCH and RSISOM with the parameter of intensity change from +5 to +45.	67
5.1	Pairs of airplane types with the similar color texture.	70

C.1	Original images m_i ; $1 \leq i \leq 20$ used in the experiment. The subscript i denotes the assigned class $[i]$	81
C.2	Original images m_i ; $21 \leq i \leq 34$ used in the experiment. The subscript i denotes the assigned class $[i]$	82
C.3	Transformed images: Rotating the original image m_{22} through 15-180 degrees.	83
C.4	Transformed images: Rotating the original image m_{22} through 195-345 degrees.	84
C.5	Transformed images: Scaling the original image m_{22} up to two times as large as the original size.	85



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

CHAPTER I

Introduction

Pattern recognition and classification is one of the most basic characteristics of human intelligence. It plays a key role in perception, as well as at the various levels of cognition. Nowadays, the advanced system of pattern recognition where the automatic recognition of an object in a scene regardless of its position, size and orientation has become increasingly necessary. To solve the problem of invariant recognition, the feature extraction is a key and an extremely significant step. This dissertation is aimed at developing a new algorithm to extract intrinsic features of an image and at considering computational complexity of the algorithm.

1.1 Approaches of Invariant Recognition

An image is basically characterized by a range of individual features such as color, texture and shape. The extracted feature is considered as a representative of the image. The representative should be as insensitive as possible for variations such as changes in size, rotation and translation as discussed in [1, 2, 3, 4, 5, 6, 7]. Several representative fashions are categorized into two classes. First, the representative is in forms of a boundary shape such as chain codes, polygonal approximations, signatures, skeletons and Fourier descriptor [2, 8, 9, 10, 11, 12, 13].

The other representatives of reflective properties, such as color and texture, are computed by a histogram of an image. Moments of the histogram which describe the properties of the histogram such as its variance, smoothness, skewness and flatness [10, 14] can be a representative of an image. However, using only the intensity histogram is not an efficient representative for classification. The representative based on the histogram lacks the image information re-

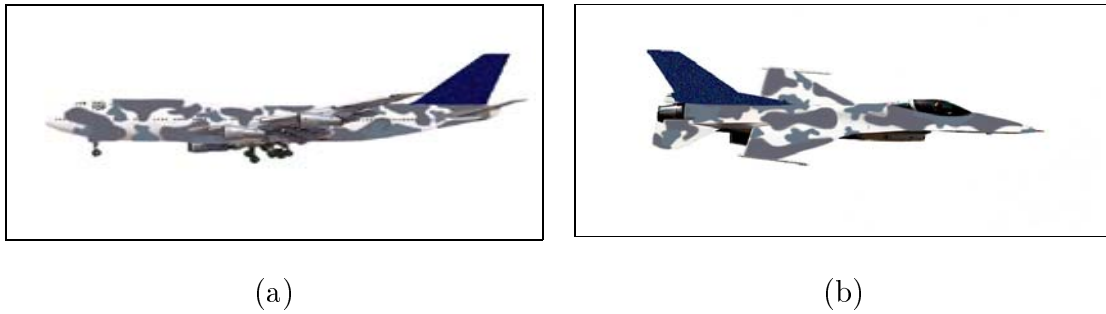


Figure 1.1: For example, two different types of airplanes have the same colored texture. (a) Airplane type **A-1**. (B) Airplane type **F-1**.

garding the relative position of pixels with respect to each other. A co-occurrence matrix is an alternative tool to consider not only the distribution of intensity but also the position of pixels. Its aim is to discriminate among images having different textures but it does not consider the shape and texture of the whole image. For example, the co-occurrence matrices of the two airplanes in Figure 1.1 are the same because of their texture. The technique is suitable for the problem of the texture classification in textile industry.

Several approaches on invariant recognition are realized by neural networks. These networks have a specific architecture designed for an individual task such as a recognition regardless of translation, rotation and scaling of an image. If a simple multilayer perceptron is used in the problem of invariant classification, it will be exhaustively trained over a large number of patterns containing the most possible transformations of them. For example, the technique of setting weight links in a neural network known as a shared weight neural network [15, 16, 17, 18, 19] has been proposed for a handwriting recognition since it has some invariant degree of local translation and rotation. Local translation and rotation are defined as some local parts of image which change in a shifting position and a rotating orientation, respectively. This technique is not applied to the invariant recognition of rotation and scaling of the whole image.

Third-order neural network [20, 21, 22, 23] is the other technique used for solving the problem of invariant recognition. Although it has successfully classified the images independently of

their sizes and rotational orientations, the growth rate of network complexity is in the order of n^3 where n is the size of image. Implementing the third-order neural network in the real applications is not feasible. Furthermore, several papers on image processing address the technique of a pulse-coupled neural network [24, 25, 26, 27, 28] to cope with the image segmentation with invariant property. The pulse-coupled neural network is a biological model inspired by cat's visual cortex. However, the model has been modified to be a feature extractor by gathering the produced pulses from all neurons. It is called "time signature". One disadvantage of the pulse-coupled neural network working as the feature extractor is how to measure the similarity and dissimilarity among the time signatures of the transformed images and the different images regardless of their orientation and sizes.

Moments have been utilized as extracted features to achieve invariant recognition of two-dimensional images. Zernike moment is the most widely accepted in many applications requiring the invariant properties as mentioned in [29, 30, 31]. Zernike moment declares only the rotation invariant feature. Subsequently, the normalization approach using a regular moment is applied to an image to obtain the scale and translation invariance prior to applying Zernike moment. The performance of Zernike moment is not robust to scaling of an image when it scales two times as large as the original.

Most techniques applied to the invariant recognition problems do not address the robustness of color intensity change. The technique of fuzzy color histogram [32] investigated for the image retrieval from image databases performs better tolerance to color intensity change than the conventional histogram does. This technique considers the color similarity of the color of each pixel color associated to all the histogram bins through fuzzy-set membership values. The fuzzy color histogram has also a weak point to discriminate two different images with the similar color distribution as shown in Figure 1.2.

Most of efficient image recognition methods do not consider the texture of color images. Some techniques use the concept of color distribution but they neither capture the colored texture nor consider the global pixel positions. In addition, the time complexity of some

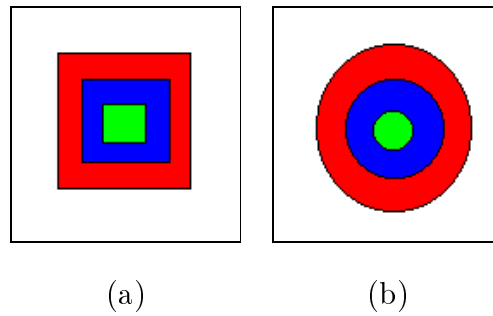


Figure 1.2: For example, two images (a) and (b) with the similar color distribution appear different on the shape: squares and circles.

techniques is not practical to apply to some real applications having images of size at least 256×256 pixels. In this dissertation, a model of the proposed technique is based on unsupervised learning neural network. The technique of competitive learning is applied to extract the feature of an image. To develop the new model, the classical competitive learning method is presented in Chapter 3 and then the formulation of the invariant recognition is discovered in Section 1.2.

1.2 Problem Formulation and Proposed Solutions

Many applications of Kohonen's competitive learning algorithm are aimed at classification, where the input data are assigned to individual classes. In particular, the input data are represented as the pixel coordinates of an image; therefore, the location of weight vectors can play a role of a feature representative of the image. The interesting problem is how to guarantee that all weight vector locations will not be altered due to the transformation, scaling, rotation and color intensity changes of the image. There are three main problems to be considered in this dissertation as shown in Table 1.1.

1.3 The Contributions of the Dissertation

Image recognition with the invariant capability is one of the significant and intriguing problems in the computer vision. Human does not even notice that one recognizes objects and patterns

Table 1.1: A list of three formulated problems and the corresponding solutions.

	Problems	Solutions
I	How can the weight vectors with respect to the rotated data vectors and the weight vectors with respect to the original data vectors be placed at the same location after the SOM learning is converged?	Apply the concept of Principle Component Analysis (PCA) to initialize the location of weight vectors. All weight vectors must be initialized at the same location with respect to the structural aspect of the data vectors. After the PCA step, the typical SOM learning can be proceeded.
II	How can the weight vectors with respect to the scaled data vectors and the weight vectors with respect to the original data vectors be placed at the same location after the SOM learning is converged?	Adopt the concept of selecting the data vectors to adjust each weight vector suggested by Clippingdale and Wilson [33]. The concept of Clippingdale and Wilson is used instead of adjusting a weight vectors of a winner neuron based on only one selected data vector.
III	Given two images whose textures are identical but their color intensities are different, how can the SOM algorithm be applied to these images to extract their invariant features?	Expand a dimension of data vectors in order to encode the whole information of an image with gray-leveled intensity or color intensity. Then, the problem of color intensity invariant is transformed to the problem of scaling invariant in the intensity dimension.

independently of changes in lighting conditions, shifts of the object, or changes in orientation and scale. The intensity of light, sound, odor and touch vary from place to place and from time to time. The stimulation of receptors and the presumed sensations are variable and changing extremely, unless they are experimentally controlled in a laboratory. The unanswered question of sense perception is how an observer, animal or human, can obtain constant perceptions in everyday life on the basis of these continually changing sensations.

Only a small subset of the problems outlined above is concerned in this dissertation, i.e., the invariant perception of two-dimensional patterns under shift, rotation and scaling in the plane. Considerably many approaches have been investigated to extract the features of an image which are independent on translation, rotation and scaling of an image. The approaches can be classified in groups of using the architecture of neural network [34, 35, 20, 16, 17, 36, 37, 38, 18, 19, 21, 22, 23, 39, 40], the integral transforms [41, 42, 43, 44, 45, 46, 47, 48, 49, 50] and the statistical methods [51, 21, 45, 52, 29, 53, 54, 48, 22, 55, 56, 50, 57, 58]. The tremendous number of researches emphasizes that invariant capability is a vital feature. Some classical and recent techniques are reviewed, and their strengths and weaknesses are identified in this dissertation. Researches in the invariant image recognition are contributed not only in the field of computer science but also expanded in breadth thorough many branches; scientific, agricultural, industrial, medical, environmental, educational, and military in [6, 7]. Table 1.2 shows the examples of invariant image recognition in the real applications.

To achieve the problem of image recognition with invariant property, it is necessary to extract the feature from an image which is not effected by the orientation or size of the image. The system commonly matches two images according to the extracted features. One of the difficult tasks is how to provide the most discrimination ability in the feature space and how to maintain the invariant ability aspects of rotation and scaling. It also is essential to regard the complexity of time and space of the approach in order to preserve its practical implementation in the real problems. In this dissertation, a new image recognition method is proposed and

¹Malignant melanoma is nowadays one of the leading cancers among many white-skinned populations around the world [61].

Table 1.2: Examples of invariant image recognition.

Branches	Applications
Scientific	DNA/Protein sequence analysis
Agricultural	Forecasting crop yields
Industrial	Machine perception to automate the process of sorting incoming material on a conveyor belt
Medical	Automated Melanoma ¹ Recognition [59]
Environmental	Automatic plankton image recognition [60]
Educational	Search engine on Internet
Military	Automatic airplane detection

the technique of feature extraction with the invariant properties is emphasized. The proposed technique is based on the concept of Kohonen's competitive learning and the knowledge of principle component analysis.

In particular, the contribution of this dissertation is to develop a new feature extraction algorithm based on the neural network concept of self-organizing mapping which has the following capabilities:

1. invariant to rotation, scaling and color intensity,
2. lower computational time and space than the current technique and the expected complexity in the order of $O(n)$,
3. applicable to a color image of size at least 256×256 pixels.

1.4 Scope and Organization

Basically, the system of automatic pattern recognition consists of the following three modules:

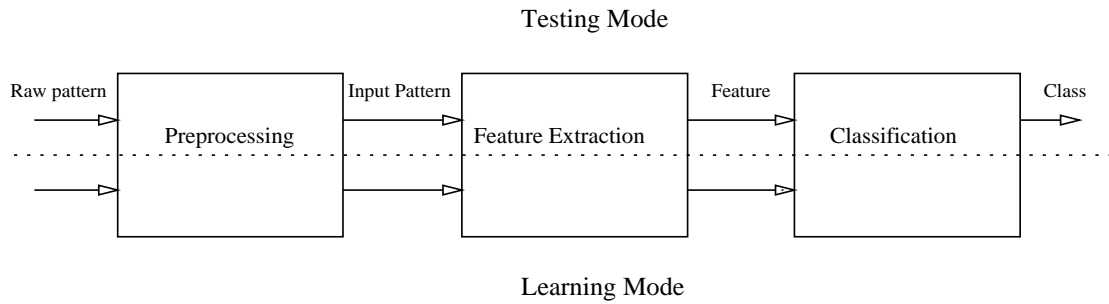


Figure 1.3: A system of automatic pattern recognition.

1. **Preprocessing:** The role of the module is to segment the pattern of interest from the background, to remove noise, to normalize the pattern and any other operation in order to define an input pattern. After the completion of preprocessing, the input patterns and the raw patterns¹ are in the Cartesian domain defined as a function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, $f(x, y) = z$, where z is the intensity of the image at coordinates (x, y) .
2. **Feature Extraction:** The module extracts the appropriate features for representing the input pattern invariant to shift, rotation and scaling.
3. **Classification:** Given the features of a pattern, there are two sorts of classification: supervised and unsupervised classification. The supervised classification requires the number of categories and the meaning of categories, defined by a system designer. The unsupervised classification can determine the number of classes and the meaning of categories by the similarity of pattern.

The system of automatic pattern recognition is illustrated in Figure 1.3. The system can classify the patterns after the learning mode. In the learning mode, a large number of patterns are proceeded and the classifier must learn to partition the feature space with respect to the desired classes. In the other, the learning classifier assigns the input pattern to one of the pattern classes according to the partitioning of learning mode.

¹A pattern acquired from any equipments such as a scanner is so-called a raw pattern.

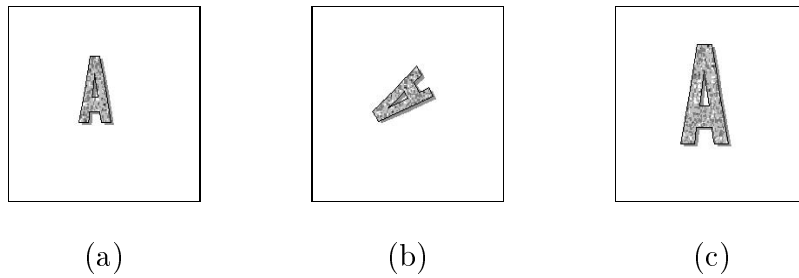


Figure 1.4: An example of an image having an object, *letter 'A'*, in the white background. (a) The prototype image. (b) The rotated 'A'. (c) The scaled 'A'.

In this dissertation, the pattern of interest, herein, is a two-dimension image with color intensity of size at least 256×256 pixels. Invariant recognition is constrained to rotation, scaling and color intensity. Constraints relevant to invariant rotation and scaling are defined as follows:

1. The considered image has only one object in the whole scene. The object is rotated and/or scaled without altering the background as shown in Figure 1.4.
2. In case of having many objects, it is more convenient to cluster them as a group and, then, consider them with the properties given in constraint 1 as shown in Figure 1.5.

In case of the invariant color intensity of an image, let $I(\mathbf{x})$ be the intensity of a gray-leveled image at \mathbf{x} where $0 \leq I(\mathbf{x}) \leq 255$ and $(R(\mathbf{x}), G(\mathbf{x}), B(\mathbf{x}))$ the intensity of a color image at \mathbf{x} presented in RGB format, where $0 \leq R(\mathbf{x}), G(\mathbf{x}), B(\mathbf{x}) \leq 255$. If the intensity of a gray-leveled image $I(\mathbf{x})$ is changed by an intensity ratio α , the new intensity, $I'(\mathbf{x})$, will become

$$I'(\mathbf{x}) = I_0 + \alpha I(\mathbf{x}) \quad (1.1)$$

$$\alpha = \frac{I_{max} - I_0}{255} \quad (1.2)$$

where I_0 and I_{max} are a constant, and $0 \leq I_0 \leq I_{max} \leq 255$. To change the intensity of a color image, let α , β , and γ be independently given intensity ratios for changing the intensities of $R(\mathbf{x})$, $G(\mathbf{x})$, and $B(\mathbf{x})$, respectively. The new intensity $(R'(\mathbf{x}), G'(\mathbf{x}), B'(\mathbf{x}))$ of an image is

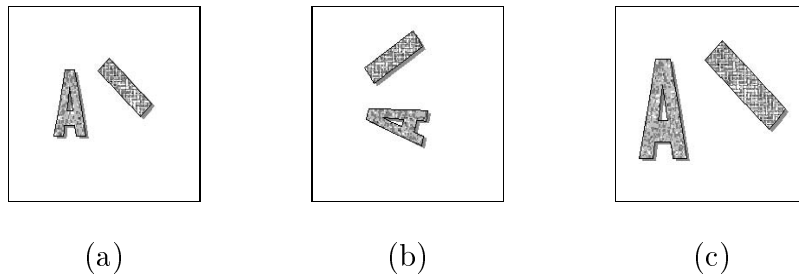


Figure 1.5: An example of an image having two objects, *letter 'A'* and *letter 'I'*, in the white background. (a) The prototype image. (b) The rotated group which is formed by *letter 'A'* and *letter 'I'*. (c) The scaled group.

computed by

$$(R'(\mathbf{x}), G'(\mathbf{x}), B'(\mathbf{x})) = (R_0 + \alpha R(\mathbf{x}), G_0 + \beta G(\mathbf{x}), B_0 + \gamma B(\mathbf{x})) \quad (1.3)$$

$$\alpha = \frac{R_{max} - R_0}{255} \quad (1.4)$$

$$\beta = \frac{G_{max} - G_0}{255} \quad (1.5)$$

$$\gamma = \frac{B_{max} - B_0}{255} \quad (1.6)$$

where $R_0, G_0, B_0, R_{max}, G_{max}$ and B_{max} are constants which $R_0 \leq R_{max}$, $G_0 \leq G_{max}$, and $B_0 \leq B_{max}$, and $R_0, G_0, B_0, R_{max}, G_{max}$ and B_{max} are between 0 and 255. An example is illustrated in Figure 1.6.

The dissertation is organized into six chapters. Chapter 2 reviews the previous techniques of invariant recognition such as neural network, transformation and statistics. Using neural network to recognize regardless of invariant geometric transformation is divided into subsections: neural network with shared weights, high-order neural network and pulse-coupled neural network. Transformation technique is the convolution of an image in the spatial domain with the kernel. Transforms are useful for invariant pattern recognition if it is possible to choose kernels so that the transformed images are invariant under some specified transformations. Fourier transform is the most broadly used in an image processing. In this dissertation,

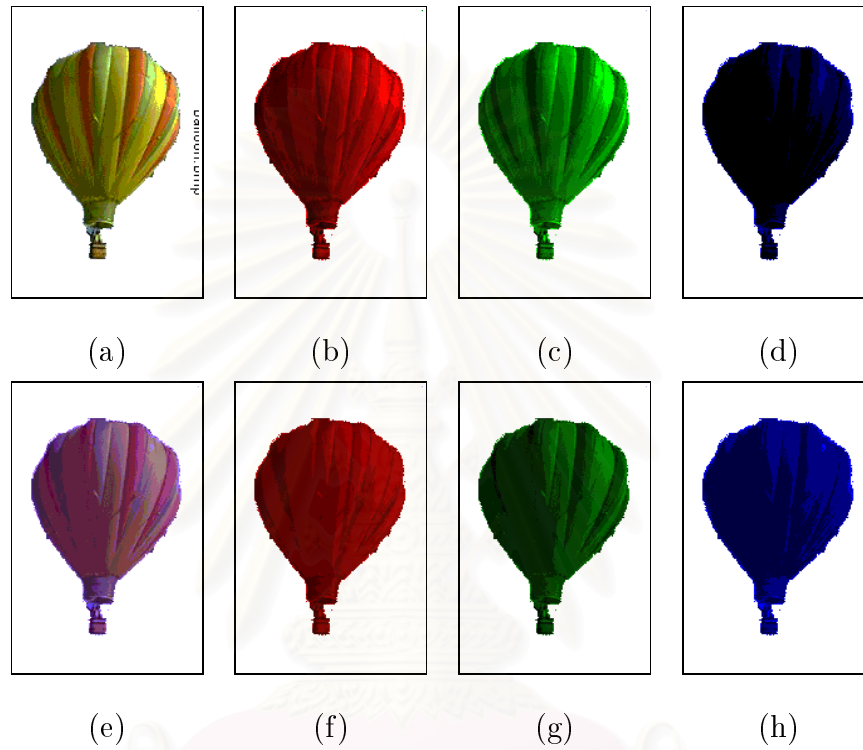


Figure 1.6: An example of a change in color intensity. (a) The prototype image. (b)-(d) Images decomposed from (a) into R, G and B planes, respectively; (e) The output image after adjusting the color intensity of R, G and B planes. (f)-(h) Images when $\alpha, \beta, \gamma = 0.59$ whereas $R_0 = 50$, $G_0 = 0$ and $B_0 = 100$.

Fourier transform is discussed as well as the other series of Fourier transform such as discrete Fourier transform, and Fourier-Mellin transform. Invariant recognition based on the statistical techniques such as moments and co-occurrence matrix is finally discussed. The discussion is focused on how each technique provides the invariant features and some derivations to prove those techniques. The strengths and weaknesses of each technique are also examined.

To understand the proposed algorithm, the concept of Kohonen's competitive learning is presented in the first section of Chapter 3. Then, the proposed algorithm, *Rotational and Scaling Invariant Self-Organizing Mapping Neural Network* algorithm, is described and its computational time is discussed in Chapter 3. For convenience, the proposed algorithm is shortly called RSISOM. RSISOM comprises two main algorithms, namely, *Rotational Direction* and *Self-Partitioning Competitive Learning*. *Rotational Direction* algorithm finds the correct direction of a pattern when it arbitrarily rotates through at most 360 degrees. Section 3.4 demonstrates how the proposed algorithms are applied to color images. The role of the other algorithm is to partition a pattern regardless of its scaling. The demonstration of applying RSISOM to color image is also demonstrated in Chapter 3.

Chapter 5 shows the experimental results performed by the proposed technique RSISOM on the synthesized image set. The test sets contain images with gray-leveled, color-texture and their size of up to 256×256 pixels. Furthermore, hierarchical RSISOM is tested against the color image set to compare the accuracy between the proposed methods with or without the hierarchy concept. Moreover, the comparisons with previous techniques are shown in Chapter 5. Finally, Chapter 6 concludes the dissertation and guides the future works.

CHAPTER II

Literature Reviews

In this chapter, a brief review of invariant pattern recognition is demonstrated in Section 2.1. The rest of chapter is about the existing methodologies of the approaches related to the problem of invariant pattern recognition. Their weaknesses are also indicated. There are both classical and modern techniques for solving the problem of invariant recognition applied to an image. Such techniques are classified in three main categories: special structures of neural networks, integral transforms into the frequency domain and statistical approaches. Most techniques in three categories do not address in the robustness of color intensity. In the last section, the techniques of fuzzy color histogram [32] is reviewed.

2.1 Invariant Features in Pattern Recognition

The aim of invariant pattern recognition is to identify an object independently of its rotational orientation and size, i.e., smaller or larger. The idea is to find a mapping \mathbf{T} that is capable of extracting the features of the object regardless of the rotational orientation or the size of the object are made different. The mapping \mathbf{T} necessarily maps all objects of an equivalent class under a group operation G into one point in the feature space as discussed by Burkhardt and Siggelkow [2]. Figure 2.1 illustrates the mapping \mathbf{T} can transform an object f and its rotated object into the same point of the feature space.

$$f \equiv_G f' \Rightarrow \mathbf{T}(f) = \mathbf{T}(f') \quad (2.1)$$

From the above equation, an object f will be equivalent to another object f' under a group operation G such as translation, rotation and scaling if they are mapped to the same point

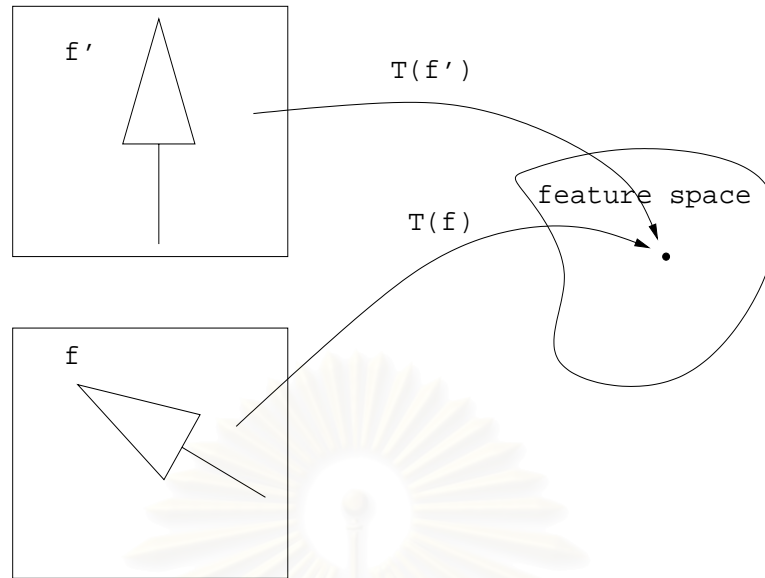


Figure 2.1: A mapping \mathbf{T} maps the object f and its rotated object f' into the same point of the feature space.

using the function \mathbf{T} in another space; the definition of the equivalent class is shown in the following equation:

$$f \equiv_G f' \iff \exists g \in G, f = g(f') \quad (2.2)$$

where g is an unary operation in group G .

The operation g is a geometric transformation such as translation, rotation and scaling. An example of transformation, rotation and scaling are shown in Figure 2.2. Let \mathbf{x} be the spatial coordinates (x_1, x_2) of object $f(\mathbf{x})$. For translation operation, if object $f(\mathbf{x})$ located at coordinate \mathbf{x} is shifted to a new coordinate \mathbf{x}' by using displacement (t_1, t_2) , the equation expressing the value of \mathbf{x}' is the following:

$$\mathbf{x}' = (x_1 + t_1, x_2 + t_2) \quad (2.3)$$

If object $f(\mathbf{x})$ is scaled by a scaling factor α , a new coordinate \mathbf{x}' of the scaled object will be

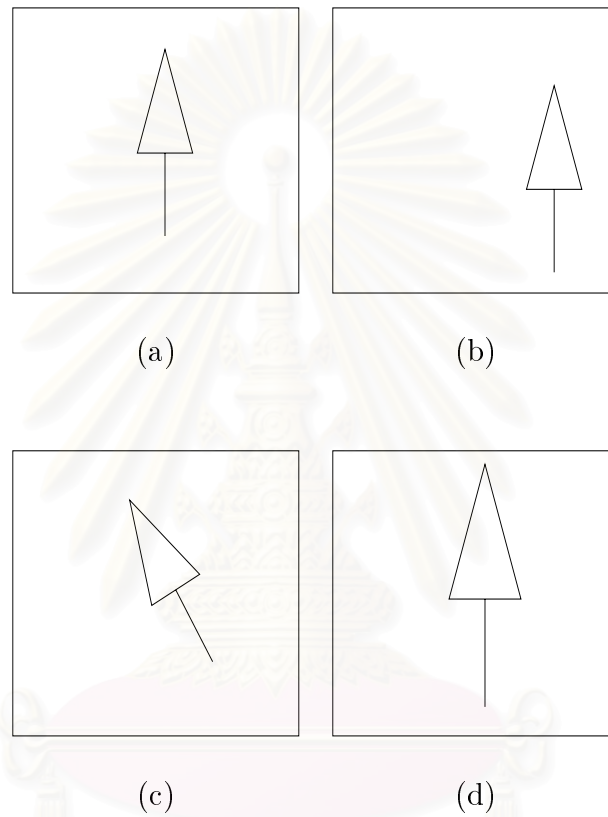


Figure 2.2: An example of geometric transformations. (a) An original object. (b) The translated object. (c) The rotated object. (d) The scaled object.

as follows:

$$\mathbf{x}' = (\alpha x_1, \alpha x_2) \quad (2.4)$$

In case of rotation, if object $f(\mathbf{x})$ is rotated through an angle θ , the obtained coordinates \mathbf{x}' will be achieved by using the equation below:

$$\mathbf{x}' = (x_1 \cos \theta + x_2 \sin \theta, -x_1 \sin \theta + x_2 \cos \theta) \quad (2.5)$$

However, a mapping \mathbf{T} that is invariant with respect to G is said to be *complete* if the following condition holds:

$$\mathbf{T}(f) = \mathbf{T}(f') \Rightarrow f \equiv_G f' \quad (2.6)$$

Let $I_{\mathbf{T}}(f)$ be a set of invariants of an object f with respect to a mapping \mathbf{T} is given by all elements that are mapped by \mathbf{T} into one point as defined below:

$$I_{\mathbf{T}}(f) = f_i | \mathbf{T}(f_i) = \mathbf{T}(f) \quad (2.7)$$

The set of objects within one equivalence class, that is, all images that can be generated from a prototype f by applying the group operation G such that

$$G(f) = f_i | f_i \equiv_G f \quad (2.8)$$

In general, the mapping \mathbf{T} yields the condition in Eq. 2.1 such that

$$G(f) \subseteq I_{\mathbf{T}}(f) \quad (2.9)$$

Ideally, such a mapping \mathbf{T} would identify f and f' as representatives of the same object. However, in many pattern recognition problems, the aim is to produce a system which classifies input patterns as belonging to a particular class, rather than to identify uniquely every single input pattern presented. In such cases, an unique representative for each and every possible input pattern can actually be a disadvantage. The invariant representative, therefore, retains enough information for distinct classes to be distinguishable in the invariant representative.

2.2 Invariant Recognition Using Neural Networks (NN)

In recent literatures, several approaches to deal with invariant recognition are variations among neural networks. These networks have a specific architecture designed for an individual task in image recognition such as a recognition regardless of translation, rotation or scaling of an image. If a simple multilayer perceptron is used in the problem of invariant classification, it will be exhaustively trained over a large number of patterns containing the most possible transformations of them. High-Order Neural Network (HONN) has been widely used to solve the problem of invariant recognition as described in Subsection 2.2.1. The recent variation of neural network, so-called Pulse-Coupled Neural Network (PCNN), is reviewed in Subsection 2.2.2.

2.2.1 High-Order Neural Network (HONN)

A model of high-order neural network in [34, 35, 20, 36, 37, 38, 21, 22, 23, 39, 40] is proposed to manage and improve the invariant pattern recognition problem of the invariant capability of the first-order neural network.

In mathematical terms, the first-order neuron is defined by the following activation function:

$$y_i = \varphi\left(\sum_{j=1}^n w_{ij}x_j\right) \quad (2.10)$$

where x_j is the input of link j ; n is the number of x_j ; w_{ij} is the weight of link j of neuron i ; $\varphi(\cdot)$ is the activation function; and y_i is the output of the neuron i . Similarly, a high-order neuron consists of a set of different ordered activation terms. The order of a network is defined in terms of the maximum number of inputs multiplied by each weight in each activation term. For example, Figure 2.3 illustrates a sample architecture of a high-order neural network consisting of three activation terms, i.e. first-order, second-order and third-order activation terms. The general form of a high-order activation equation can be formulated as follows.

$$y_i = \varphi\left(\sum_{j=1}^n w_{ij}x_j + \sum_{j=1}^n \sum_{k=1}^n w_{ijk}x_jx_k + \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n w_{ijkl}x_jx_kx_l + \dots\right) \quad (2.11)$$

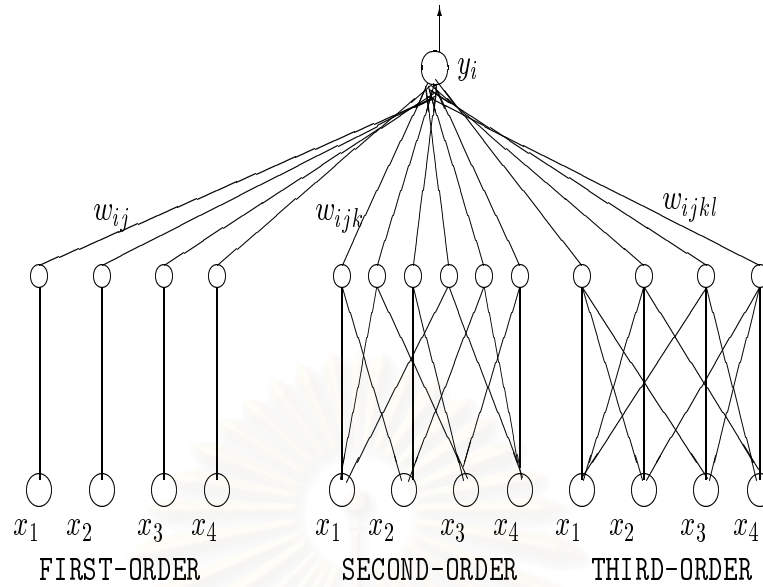


Figure 2.3: A sample architecture of high-order neural network with four inputs, x_1, x_2, x_3, x_4 .

In Eq. (2.11), each consequent term is called first-order term, second-order term, and so on, where the last term is a r^{th} -order term. To take full advantage of high-order neural network, the category of third-order and second-order neural networks have been investigated by Lee Giles and Maxwell [62] since 1987. Both a third-order and second-order neural networks are described later.

However, the technique of high-order neural network has a considerably inherent drawback concerning the number of combinatoric connections. Consequently, the implementation of a high-order neural network is not possible in the real applications because of the network complexity and computational time of order $O(n^r)$ where r is the highest order as proved in Appendix B.

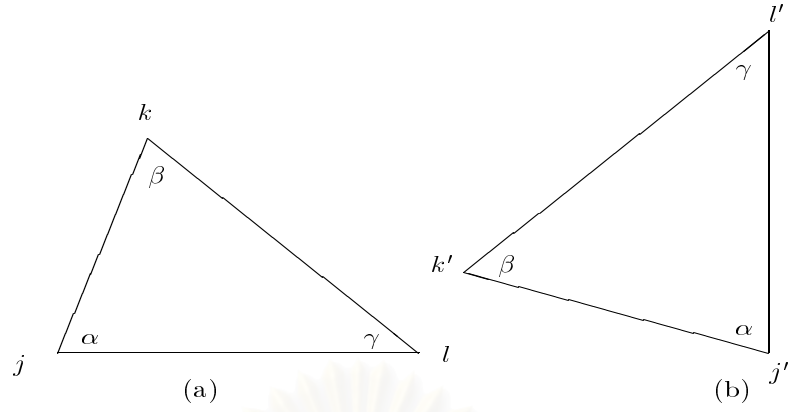


Figure 2.4: Two triangles with included angles (α, β, γ) . (a) Original triangle $jdkl$. (b) A new triangle $j'k'l'$ by scaling and rotating triangle (a).

Third-Order Neural Network (TONN)

To achieve the invariant classification with respect to translation, rotation and scaling, a third-order neural network is employed in [20, 21, 22, 23] which is defined in Eq.(2.12).

$$y_i = \varphi\left(\sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n w_{ijkl} x_j x_k x_l\right) \quad (2.12)$$

The application of third-order neural networks to invariant recognition for a bi-level image has been investigated by Delopoulos, Tirakis and Kollias [20], and Perantonis and Lisboa[21, 22]. Thus, x_j denotes the j^{th} element of a spatial coordinate of an image or equivalently the j^{th} input, w_{ijkl} is the weight that connects the product of x_j , x_k , and x_l to the output y_i . The variable n is the number of pixels in the image.

Reid [23] said that any three points within an object define a triangle with included angles (α, β, γ) . The fact of the triangle with the angles (α, β, γ) is invariant under object translation, scaling and rotation. Figure 2.4(a) shows an triangle $jdkl$ and then the triangle is moved into the other position, rotated and scaled up. The new triangle is $j'k'l'$ as shown in Figure 2.4(b). However, the set of three angles (α, β, γ) of both triangles $jdkl$ and $j'k'l'$ is the same.

Therefore, the weights in the third-order neural network to be invariant under translation,

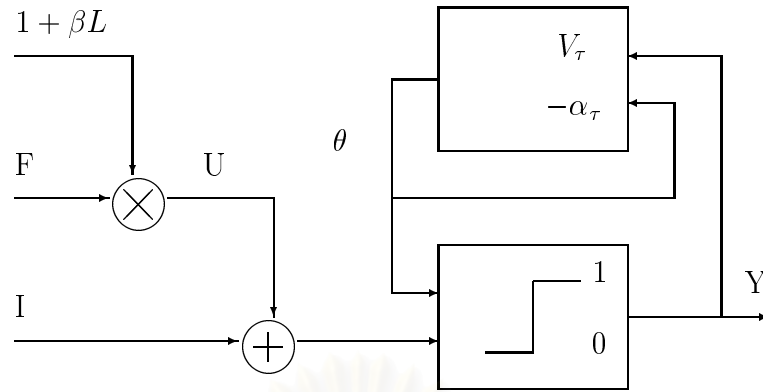


Figure 2.5: A computation diagram of pulse-coupled neural network according to Eq.(2.15)-(2.16).

rotation and scaling are defined as

$$w_{ijkl} = w_{ij'k'l'} \quad (2.13)$$

However, the third-order neural network is not feasible for the real applications because the high complexity of network and time are in the order of $O(n^3)$ where n is the number of pixels. For example, the image of size 128×128 requires the number of hidden nodes as much as 7.3×10^{11} .

2.2.2 Pulse-Coupled Neural Network (PCNN)

Several papers [24, 25, 26, 28, 27] in the image processing address the technique of pulse-coupled neural network to cope with the problems such as image segmentation, edge detection and image recognition. A PCNN is a biological model inspired from the cat's visual cortex. It is the extension of Eckhorn's model of the cat's visual cortex.

A neuron of PCNN is computed by the following equations and its computation diagram

is shown in Figure 2.5.

$$F_{ij}(n) = e^{-\alpha_F} F_{ij}(n-1) + S_{ij} + V_F e^{-\sigma_{kl}} m_{ijkl} Y_{kl}(n-1) \quad (2.14)$$

$$L_{ij}(n) = e^{-\alpha_L} L_{ij}(n-1) + V_L e^{-\sigma_{kl}} w_{ijkl} Y_{kl}(n-1) \quad (2.15)$$

$$U_{ij}(n) = F_{ij}(n)(1 + \beta L_{ij}(n)) + I \quad (2.16)$$

$$Y_{ij}(n) = \begin{cases} 1, & \text{if } U_{ij}(n) > \theta_{ij}(n) \\ 0, & \text{otherwise} \end{cases} \quad (2.17)$$

$$\theta_{ij}(n) = e^{-\alpha_\theta} \theta_{ij}(n-1) + V_\theta Y_{ij}(n) \quad (2.18)$$

$$(2.19)$$

where S is the input signal, F is the feed, L is the link, U is the internal activity, Y is the pulse output and θ is the dynamic threshold. The weights m_{ijkl} and w_{ijkl} are local interconnections, and β is the linking constant. V_F and V_L are the temporal response kernel. α_F , α_L , α_θ and σ are the decay time constant. V_F , V_L and V_θ are the amplitude gain. I is the inhibition term that is determined by the total activity of the network. The output values of all neurons are accumulated and fed back to each neuron.

The simple structure of a PCNN for a two-dimensional image is a network of neurons connected and arranged matching to the pixels of the image as shown in Figure 2.6. Let the size of an image be $M \times N$ pixels. Hence, the number of neurons is $M \times N$ as well. Suppose that the neurons are set to zero, so the input results in activation of all of the neurons at the first iteration. Normally, the threshold value is reduced with an exponential time. Only case of the threshold value increasing is after its neuron firing. But the neuron will fire when the threshold falls below the respective neuron's potential U . The pulses for each neuron are produced by repeating the process of neuron firing, threshold increasing and threshold slowly decaying, and neuron firing again after the threshold value being below the respective neuron's internal activity. However, there are supporters from neighboring neurons to fire simultaneously through interconnections. The firing neurons begin to communicate with their nearest neighbors, which in turn communicate with their neighbors.

The representative of an image is called a "time signature". The time signature is a

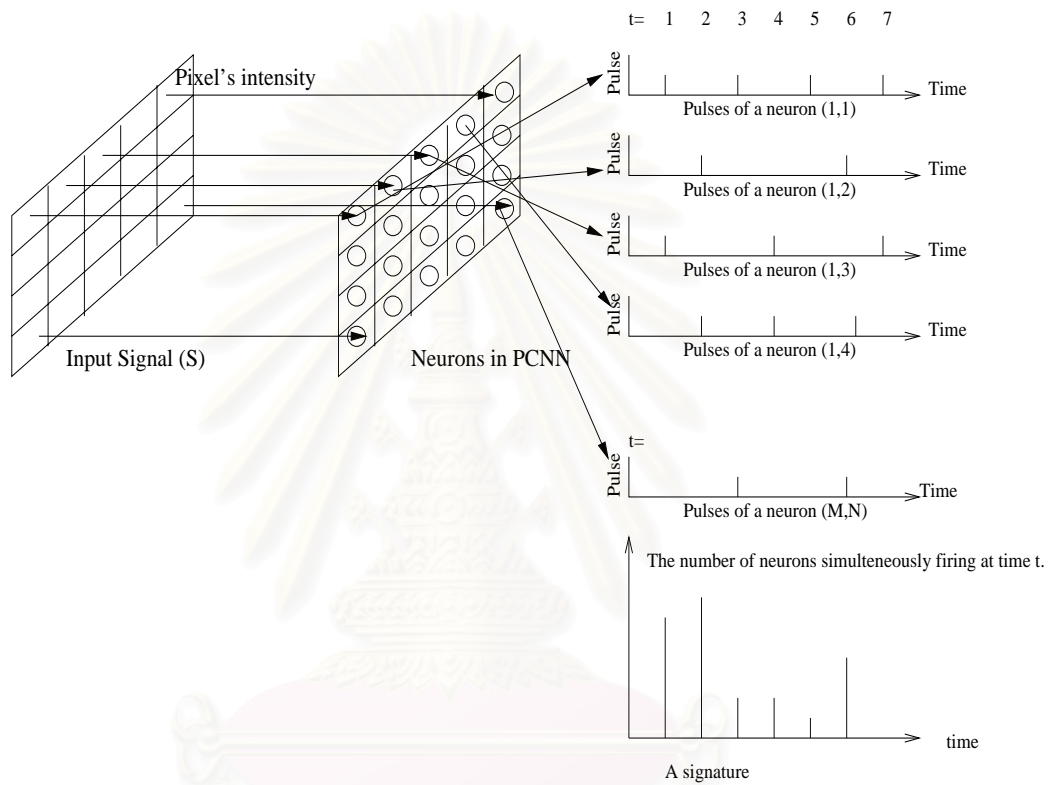


Figure 2.6: A structure of pulse-coupled neural network with the input image of size $M \times N$ pixels.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

histogram by time domain where the vertical axis is the amount of neurons firing at the time. However, the performance of a PCNN considerably depends upon the assigned parameters. The other disadvantage is how to measure the similarity of time signatures in which the images are transformed in either size or rotational orientation. It is necessary to apply the post-processing approach such as a multilayered neural network to recognize the time signatures.

2.3 Invariant Recognition Based On Transformation Techniques

2.3.1 Fourier Transforms (FT)

The application of Fourier transforms covers a wide range of image processing problems. Several researches have used it to extract the features of an image. The transformed image in the frequency space has the translation invariant property as discussed in [41, 43, 45, 46, 47, 63, 50]. But the other important properties, rotation and scaling, can be derived from Fourier-Mellin Transform applied in [42, 44, 48, 49]. Discrete Fourier Transform (DFT) to implement on a digital image is also explained in this section since this approach copes with a digital image in discrete domain. The following subsection describes each version of Fourier transform including its invariant properties.

Let us start with a 1-D Fourier transform equation. The Fourier transform of a continuous function $f(x)$ is defined in Eq.(2.20).

$$F(u) = \int_{-\infty}^{\infty} f(x)e^{-j2\pi ux} dx. \quad (2.20)$$

where $j = \sqrt{-1}$ and a variable u is the frequency variable. We denote the operation of taking the Fourier transform by the operator \mathfrak{F} so that we can write $F(u) = \mathfrak{F}f(x)$.

Now consider a 2-D transform equation. It can be extended to a function $f(x_1, x_2)$ as the following equation.

$$\mathfrak{F}f(x_1, x_2) = F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x_1, x_2)e^{-j2\pi(ux_1+vx_2)} dx_1 dx_2 \quad (2.21)$$

If $f(x)$ is transformed by the one parameter translation specified by t , the new $f'(x)$ is

$$f'(x) = f(x - t) \quad (2.22)$$

Taking the Fourier Transform of $f'(x)$, we obtain

$$\mathfrak{S}f'(x) = \int_{-\infty}^{\infty} f(x - t)e^{-j2\pi ux} dx \quad (2.23)$$

Then, we substitute x' with $x - t$.

$$\mathfrak{S}f'(x) = \int_{-\infty}^{\infty} f(x')e^{-j2\pi u(x+t)} dx' \quad (2.24)$$

$$= \int_{-\infty}^{\infty} e^{-j2\pi ux} e^{-j2\pi ut} dx' \quad (2.25)$$

$$= e^{-j2\pi ut} \int_{-\infty}^{\infty} e^{-j2\pi ux} dx' \quad (2.26)$$

$$= e^{-j2\pi ut} \mathfrak{S}f(x) \quad (2.27)$$

Hence, the Fourier transform provides the invariant properties under transformation since the translation in $f(x)$ does not affect the magnitude of its Fourier transform, as

$$|\mathfrak{S}f(x)| = |e^{-j2\pi ut} \mathfrak{S}f(x)| \quad (2.28)$$

In the same manner, the 2-D Fourier transform has the transformation property derived in [10] as the equation below.

$$f(x_1 - t_1, x_2 - t_2) \iff \mathfrak{S}f(x_1, x_2)e^{-j2\pi(ut_1+vt_2)} \quad (2.29)$$

The following Subsection describes the other versions of Fourier transform.

Discrete Fourier Transform (DFT)

The discrete Fourier transform is shown in Eq.(2.30).

$$\mathfrak{S}f(x_1, x_2) = F(u, v) = \frac{1}{MN} \sum_{x_1=0}^{M-1} \sum_{x_2=0}^{N-1} f(x_1, x_2) e^{-j2\pi(\frac{ux_1}{M} + \frac{vx_2}{N})} \quad (2.30)$$

where $u = 0, 1, 2, \dots, M - 1$ and $v = 0, 1, 2, \dots, N - 1$.

Fourier-Mellin Transform (FMT)

The Mellin transform of function $f(x)$ defined over the positive reals is the complex function $\phi(z)$, where

$$\phi(z) = \int_0^{\infty} x^{z-1} f(x) dx \quad (2.31)$$

Mellin transform of a function $f(x)$ is an integral function over the positive real numbers. Its equation is closely related to the Fourier transform if the parameter $x = e^{-jy}$ is replaced. The re-written equation becomes

$$\phi(z) = -j \int_0^{\infty} e^{-jy(z-1)} f(e^{-jy}) e^{-jy} dy \quad (2.32)$$

$$= -j \int_0^{\infty} e^{-jzy} f(e^{-jy}) dy \quad (2.33)$$

Hence, Mellin transform with the new parameter $y = \frac{-\ln(x)}{j}$ is similar to the Fourier transform where a coordinate x has been defined in logarithmic form. As described in Section 2.3.1, the Fourier transform transforms the function $f(x)$ in spatial domain into the function $\Im f(x)$ in frequency domain but its magnitude of $\Im f(x)$ is not changed even though the position of x is shifted. Thus, Mellin transform is invariant to translation.

Previous researches [42, 44, 48, 49] have discovered that Mellin transform is possible to obtain the other invariant properties such as rotation and scaling in the application of pattern recognition. Suppose the function $f(x_1, x_2) \in \Re \times \Re$ and then its coordinates (x_1, x_2) is represented in the polar coordinates (r, θ) where $r \in [0, \infty)$ and $\theta \in [0, 2\pi)$. Let \mathbf{v} be a vector from the origin to the pixel coordinate (x_1, x_2) , r the length of vector \mathbf{v} , and θ an angle between vector and x_1 -axis in counterclockwise direction. Such a rotation operation in a polar coordinate is

$$\theta' = \theta + \alpha \quad (2.34)$$

where θ' is the new angular coordinate after rotating through α degrees.

In case of scaling, a scaling operation in a polar coordinate is

$$r' = \lambda r \quad (2.35)$$

where λ is the new radius coordinate after scaling with a parameter λ . As mentioned above, $y = \frac{-\ln(r)}{j}$ is used so the re-write equation (2.35) is shown as follows.

$$\frac{-1}{j} \ln(r') = \frac{-1}{j} \ln(r\lambda) \quad (2.36)$$

$$= \frac{-1}{j} \ln(r) + \frac{-1}{j} \ln(\lambda) \quad (2.37)$$

$$y' = y + \beta \quad (2.38)$$

where $\beta = \frac{-\ln(\lambda)}{j}$. Obviously, Mellin transform has the properties invariant to translation, rotation and scaling.

2.4 Invariant Recognition based on Statistical Techniques

2.4.1 Moments

Various types of moments are examined to characterize the features of an image with the desirable properties in aspects of translation, rotation and scaling invariance in [51, 45, 52, 29, 53, 21, 54, 48, 22, 55, 56, 50, 57, 58]. Hu [52, 53, 54, 55, 56, 30] introduced a set of invariant moments based on regular moments¹. He generated the set of invariant moments by nonlinear combinations which has the properties of being invariant under translation, scaling and rotation.

The definition of regular moment can be viewed as the projection of $f(x_1, x_2)$ onto the monomial $x_1^p x_2^q$. Generally, the expression for the calculation of the moments is written in the forms of integration instead of summation. Assume that an image $f(x_1, x_2)$ is a continuous function. A given order $(p + q)$ is defined as

$$m_{p,q} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_1^p x_2^q f(x_1, x_2) dx_1 dx_2 \quad (2.39)$$

for $p, q = 0, 1, 2, \dots$

The central moment is used to be invariant under translation as follows.

$$\mu_{p,q} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x_1 - \bar{x}_1)^p (x_2 - \bar{x}_2)^q f(x_1, x_2) dx_1 dx_2 \quad (2.40)$$

¹Regular moments will be referred to as Geometric moments in [30].

for $p, q = 0, 1, 2, \dots$. The centroid of the image can be found by using

$$\bar{x}_1 = \frac{m_{10}}{m_{00}} \quad (2.41)$$

$$\bar{x}_2 = \frac{m_{01}}{m_{00}} \quad (2.42)$$

If the image is scaled with the scaling factor α , we can estimate the scaling factor by using the central moments μ_{20} and μ_{02} . Both the central moments μ_{20} and μ_{02} are the variance of the image as shown in Eq.(2.43) and (2.44).

$$\mu_{20} = m_{20} - \frac{m_{10}^2}{m_{00}} \quad (2.43)$$

$$\mu_{02} = m_{02} - \frac{m_{01}^2}{m_{00}} \quad (2.44)$$

Thus, Eq.(2.42) is re-defined as

$$\mu'_{p,q} = \frac{\mu_{p,q}}{\alpha^{p+q+2}} \quad (2.45)$$

Hu discovered the set of seven invariant moments of order up to three as the following.

$$\phi_1 = \eta_{20} + \eta_{02} \quad (2.46)$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (2.47)$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (2.48)$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (2.49)$$

$$\begin{aligned} \phi_5 = & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ & + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (2.50)$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \quad (2.51)$$

$$\begin{aligned} \phi_7 = & (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ & + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (2.52)$$

where $\eta_{p,q}$ is called *the normalized central moment* defined as

$$\eta_{p,q} = \frac{\mu'_{p,q}}{\mu_{00}^\gamma} \quad (2.53)$$

where $\gamma = \frac{p+q}{2} + 1$ and $p + q = 2, 3, 4, \dots$

The methods in Section 2.3 transforms an image into a space of the same dimensionality as the image space. A moment of a given order $p + q$ is scalar but the set of all its moments can be considered that the image is mapped into a discrete moment space, indexed by p and q . Consequently, moments can be considered as integral transforms.

Zernike Moment (ZM)

Among various types of moments, Zernike moment is the most potential method for extracting the invariant features of images. To process Zernike moment method, a spatial coordinate (x_1, x_2) of an image is normalized on a unit disk region as follows.

$$D = \{(x_1, x_2) \in R^2 | x_1^2 + x_2^2 \leq 1\} \quad (2.54)$$

The regions of interest can be normalized by scaling down their sizes until they fit into the unit disk. After the normalization, the centroid of the image should be located at the origin of the unit disk. In other words, a polar coordinate is replaced by a spatial coordinate as shown in Eq.(2.55).

$$D = \{(r, \theta) | 0 \leq r \leq 1 \text{ and } 0 \leq \theta \leq 2\pi\} \quad (2.55)$$

Then, the Zernike moment of order p with repetition q for the normalized image is defined in the form of polar coordinate in unit disk becomes the following:

$$A_{p,q} = \frac{(p+1)}{\pi} \int \int_D f(r, \theta) Z_{p,q}^*(r, \theta) r dr d\theta \quad (2.56)$$

where * denotes the complex conjugate. $Z_{p,q}$ is the polynomial formed as

$$Z_{p,q}(r, \theta) = R_{p,q}(r) e^{jq\theta} \quad (2.57)$$

where p is a positive integer or zero, q is a positive or negative integer, subject to the constraints $(p - |q|)$ is even and $|q|$ is less than or equal to p . Variable $R_{p,q}$ is a radial polynomial defined by

$$R_{p,q}(\rho) = \sum_{s=0}^{(p-|q|)/2} \frac{(-1)^s (p-s)! r^{p-2s}}{s! (\frac{p+|q|}{2} - s)! (\frac{p-|q|}{2} - s)!} \quad (2.58)$$

In accordance with the above-defined formula, Zernike moments are the projection of the image $f(x_1, x_2)$ onto the orthogonal basis functions except that the image $f(x_1, x_2)$ is outside the unit disk.

The Zernike moment of the image after rotation through an angle α denoted by $A'_{p,q}$ is calculated as shown in Eq.(2.59). In the same manner of the property of the Fourier transform, it is obvious that the magnitude of Zernike moment does not change when the image is rotated.

$$A'_{p,q} = A_{p,q} e^{-jq\alpha} \quad (2.59)$$

Therefore, the feature representative of an image invariant to rotation and scaling can be a vector containing the Zernike moments with different parameters either p or q . The vector of Zernike moments with the parameter p is defined as follows:

$$\mathbf{z}^p = \begin{cases} [A_{p,0} \ A_{p,2} \ \dots \ A_{p,2i}]^T \text{ and } i = 0, 1, \dots, \frac{p}{2} & \text{if } p \text{ is even} \\ [A_{p,1} \ A_{p,3} \ \dots \ A_{p,2i+1}]^T \text{ and } i = 0, 1, \dots, \frac{p-1}{2} & \text{if } p \text{ is odd} \end{cases} \quad (2.60)$$

Generally, most applications of Zernike moments focus on binary images in where the technique performs efficiently. However, the Zernike moments are very sensitive to intensity change when applied to gray-leveled images.

2.4.2 Co-occurrence Matrix (CM)

The aim of co-occurrence matrix is to measure the texture of an image. The gray-level histogram of an image cannot describe the texture information of an image. It does not provide the information regarding the relative position of pixels with respect to each other while the co-occurrence matrix is a tool which considers not only the distribution of intensities but also the positions of pixels with equal or nearly equal to intensity values.

Let S be a set of intensity values. An co-occurrence matrix is a two-dimensional array of size $|S| \times |S|$, $\mathbf{P}_d = [p_{ij}]$, in which each element p_{ij} is the number of times that points with intensity value z_i occur relative to points with intensity value z_j with the distant operator, where $0 \leq i, j \leq |S| - 1$. The distant operator denoted by $d = (m, n)$ is such m pixels to

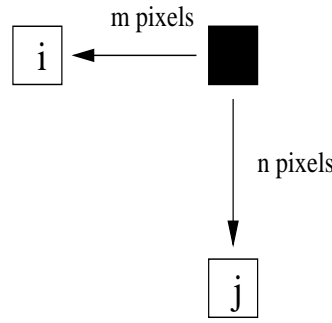


Figure 2.7: A distant operation $d = (m, n)$ is defined that m pixels to the right and n pixels below corresponding to a solid block.

the right and n pixels below; on the other hand, it can be written in a coordinate mapping as shown in Figure 2.8(a). For instance, suppose that an image contains a set of intensity values $S = 0, 1, 2$ as shown in Figure 2.8(b) and the distant operation $d = (1, 1)$ is assigned. The co-occurrence shown in Figure 2.8(c) describes that there are 16 pairs of pixels in the image which is performed by the distant operation and its size is 3×3 since there are only three gray levels.

To measure the similarity of any two co-occurrence matrices, the set of statistical parameters below are used.

1. Maximum probability, $Prop_{max}$, is a maximum element in the co-occurrence matrix calculated by

$$Prop_{max} = \max_{i,j} (p_{ij}) \quad (2.61)$$

2. Element difference moment of order k , DM_k , is computed by

$$DM_k = \sum_i^{|S|} \sum_j^{|S|} j^{|S|} (i-j)^k p_{ij} \quad (2.62)$$

3. Inverse element difference moment of order k , IDM_k is defined as

$$IDM_k = \sum_i^{|S|} \sum_j^{|S|} \frac{c_{ij}}{(i-j)^k}; \quad i \neq j \quad (2.63)$$

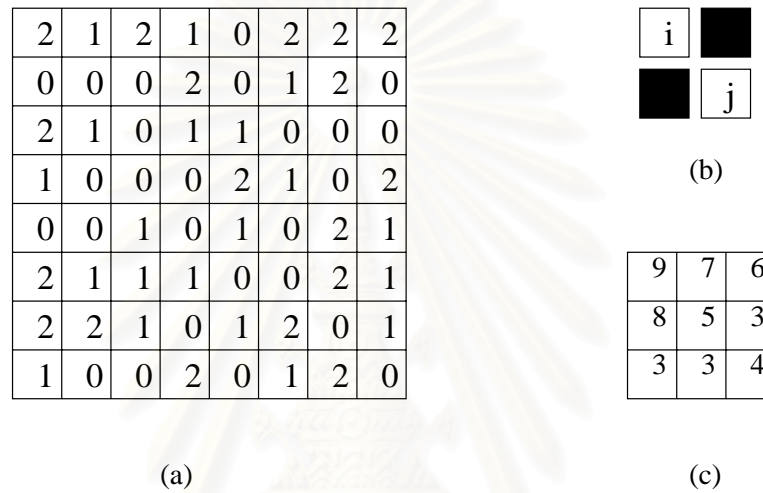


Figure 2.8: An example shows the co-occurrence matrix on a given image and the defined distant operation. (a) An image of size 16×16 pixels. (b) A distant operation $d = (1, 1)$, i.e., one pixels to the right and one pixel below. (c) A co-occurrence matrix after computing the image matrix (a) with the distant operation (b).

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4. Entropy, E , is defined as

$$E = - \sum_i^{|S|} \sum_j^{|S|} p_{ij} \log p_{ij} \quad (2.64)$$

5. Uniformity, U , is written by the following equation.

$$U = \sum_i^{|S|} \sum_j^{|S|} p_{ij}^2 \quad (2.65)$$

This technique emphasizes the texture feature but it does not consider the shape of an image.

2.5 Invariant Feature by Using Fuzzy Color Histogram (FCM)

According to the conventional color histogram, it assigns each pixel into one of the bins only. Unlike the conventional histogram, the fuzzy color histogram considers the color similarity information by spreading the total membership value of each pixel to all the histogram bins. Furthermore, the membership values of the fuzzy color histogram are computed by using fuzzy c-mean clustering algorithm.

The fuzzy c-mean algorithm is dominated by an objective function and a fuzzy c-partition of the set of data vectors as described in [64, 65, 66, 67]. Let $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ be a set of data vectors and $\mathbf{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_c\}$ a set of clustering center vectors where n and c are the number of data vectors and the number of clusters, respectively. A non-degenerate fuzzy c-partition of \mathbf{X} is conveniently represented by a matrix \mathbf{U} . The objective function of the fuzzy c-mean algorithm is minimized until the difference between the objective function at the previous time and at the present time is lower than a predefined minimum error. The objective function at time t is written as the following

$$obj(t) = \sum_{i=1}^c \sum_{k=1}^n (u_{ik}(t))^m \|x_k - v_i(t)\|^2 \quad (2.66)$$

where $\|\cdot\|$ is the Euclidean norm. The weighting exponent m , termed “amount of fuzziness”, is a constant number greater than 1. $u_{ik}(t)$ is an element of a matrix \mathbf{U} at time t , so-called “a fuzzy c-partition of X ”, defined as

$$u_{ik}(t) = \frac{\left(\frac{1}{\|\mathbf{x}_k - \mathbf{v}_i(t)\|^2}\right)^{\frac{1}{(m-1)}}}{\sum_{j=1}^c \left(\frac{1}{\|\mathbf{x}_k - \mathbf{v}_j(t)\|^2}\right)^{\frac{1}{(m-1)}}} \quad (2.67)$$

Note that, Eq.(2.67) must satisfy the following conditions:

1. $u_{ik}(t) \in [0, 1]$ for all i, k
2. $\sum_{i=1}^c u_{ik}(t) = 1$ for all k
3. $0 \leq \sum_{k=1}^n u_{ik}(t) \leq n$ for all i

The location of each clustering center vectors is directly moved into the substantial fuzzy-mean value of the group of data vector partitioning by the fuzzy c-partition in the following equation.

$$\mathbf{v}_i(t) = \frac{\sum_{k=1}^n u_{ik}(t) \mathbf{x}_k}{\sum_{k=1}^n u_{ik}(t)} \quad (2.68)$$

To begin with the fuzzy c-mean algorithm, the fuzzy c-partition \mathbf{U} is randomly initialized. It then follows with the three main steps: computing the clustering center vector V , updating the fuzzy c-partition \mathbf{U} , and calculating the objective function. They are repeated until no noticeable change in the objective function.

Given an image with n' colors, we need to classify the n' fine colors into n clusters in fuzzy c-mean (usually, n is much less than n'). The fuzzy clustering result of fuzzy c-mean is represented by matrix $\mathbf{U} = [u_{ik}]_{n \times n'}$ and u_{ik} is referred to as the grade of membership of color x_k with respect to cluster center v_i . Let $\mathbf{M} = [m_{ij}]_{n \times n'}$ be a membership matrix which each element m_{ij} is the membership value of the j th fine colors distributing to the i th coarse colors. Thus, the obtained matrix $\mathbf{U}_{n \times n'}$ can be viewed as the desired membership matrix $\mathbf{M}_{n \times n'}$ for computing fuzzy color histogram. The fuzzy color histogram of an image is expressed as follows

$$\mathbf{F}_{n \times 1} = \mathbf{M}_{n \times n'} \mathbf{H}_{n' \times 1} \quad (2.69)$$

where $\mathbf{H}_{n' \times 1}$ is a conventional color histogram with n' colors.

Therefore, the invariant feature is represented by the obtained histogram and the color bin of the histogram is based on the locations of clustering centers in the technique of fuzzy c-mean. As a result, the location of clustering center also is a feature representative of an image.



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

CHAPTER III

Rotational and Scaling Invariant Self-Organizing Mapping

3.1 Kohonen's Competitive Learning Concept

Among the neural network models, Self-Organizing Mapping (SOM), is categorized as an unsupervised learning. Either unsupervised or self-organized learning implies that there is no given target corresponding to its input data. Typically, a network of SOM is based on the competitive learning approach introduced by Kohonen [68]. A simple architecture of Kohonen's competitive learning network is a single layer of output neurons i and all inputs are fully connected to all output neurons in a feedforward fashion via synaptic weight connections \mathbf{w}_i as shown in Figure 3.1. Moreover, there are complete connections among the neurons known as lateral connections. The number of output neurons which must be specified prior to the learning process corresponds to the number of classes of the input data.

The output neurons in a network compete among themselves in order to be a winner during the learning period. The winning output neuron is called a *winner-takes-all neuron* or a *winner neuron*. Let $\mathbf{x}_k = [x_{k1} \dots x_{kl}]^T$ be a currently selected data vector and \mathbf{w}_i the weight vector of neuron i . Only one of the neurons is called a winner neuron i^* with respect to \mathbf{x}_k if its weight vector \mathbf{w}_{i^*} satisfies the following condition:

$$\mathbf{w}_{i^*} \cdot \mathbf{x}_k = \max_i(\mathbf{w}_i \cdot \mathbf{x}_k) \quad (3.1)$$

This condition can be viewed as the distance between the weight vector and the input vector if the length of the weight vector is normalized to one. Hence, the alternative condition for

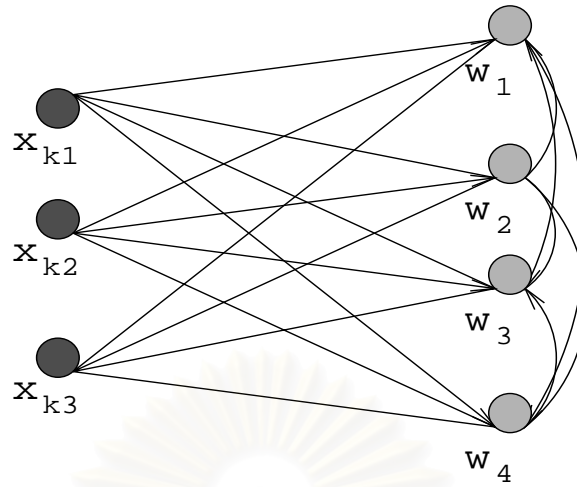


Figure 3.1: A sample architecture of Kohonen's competitive network with three input and four output neurons which \mathbf{x}_k is a selected data vector at present.

identifying a winning neuron is given by

$$\|\mathbf{w}_{i^*} - \mathbf{x}_k\| = \min_i \|\mathbf{w}_i - \mathbf{x}_k\| \quad (3.2)$$

After obtaining the winner neuron i^* , its weight vector \mathbf{w}_{i^*} is updated with respect to data vector \mathbf{x}_k by this simple learning rule

$$\mathbf{w}_{i^*}^{new} = \mathbf{w}_{i^*}^{old} + \Delta \mathbf{w}_{i^*} \quad (3.3)$$

$$\Delta \mathbf{w}_{i^*} = \eta(\mathbf{x}_k - \mathbf{w}_{i^*}) \quad (3.4)$$

where η is a learning rate.

Listing of notations used in Kohonen's SOM algorithm

T :	a constant number
l :	the dimension of data vector
n :	the number of output neurons in the network
m :	the number of data vectors in data set
$\mathbf{x}_k = [x_{k1} \dots x_{kl}]^T$:	data vector k
$\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$:	a set of data vectors
$\mathbf{w}_i = [w_{i1} \dots w_{il}]^T$:	weight vector of neuron i
$\mathbf{w}_i(t)$:	weight vector of neuron i being considered at time t
$\mathbf{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_n\}$:	a set of weight vectors
$\mathbf{C}_i(t)$:	data vectors in the cluster represented by \mathbf{w}_i at time t
$\mathbf{C}_i(t) = \{\mathbf{x}_k \mid \ \mathbf{x}_k - \mathbf{w}_i(t)\ \leq \ \mathbf{x}_k - \mathbf{w}_j(t)\ ; \forall i \neq j\}$	

Kohonen's SOM Algorithm

1. $t = 0$.
2. Initialize small random values for all weight vectors in a set \mathbf{W} .
3. **Do**
 - (a) $t = t+1$.
 - (b) Randomly select a data vector, \mathbf{x}_k , from a set \mathbf{X} .
 - (c) Determine the winner neuron i^* defined in Eq.(3.2).
 - (d) Update the weight vector \mathbf{w}_{i^*} of the winner neuron i^* by Eq.(3.3) and (3.4).

Until ($\forall i(\mathbf{C}_i(t) = \mathbf{C}_i(t-1))$) *OR* $t > T$)

From the Kohonen's competitive learning rule, it is obvious that the location of each \mathbf{w}_i depends on the selection sequence of each data vector \mathbf{x}_k and the initial location of each \mathbf{w}_i . To make the locations of all weight vectors invariant to the rotation and scaling of the data vectors,

all weight vectors must be initialized at the same locations with respect to the structural aspect of the data vectors. In addition, the selection sequence of the data vectors for directing the movement of the winner neurons to the same locations as those locations prior to either the rotation or scaling is also crucial.

3.2 Rotational Direction

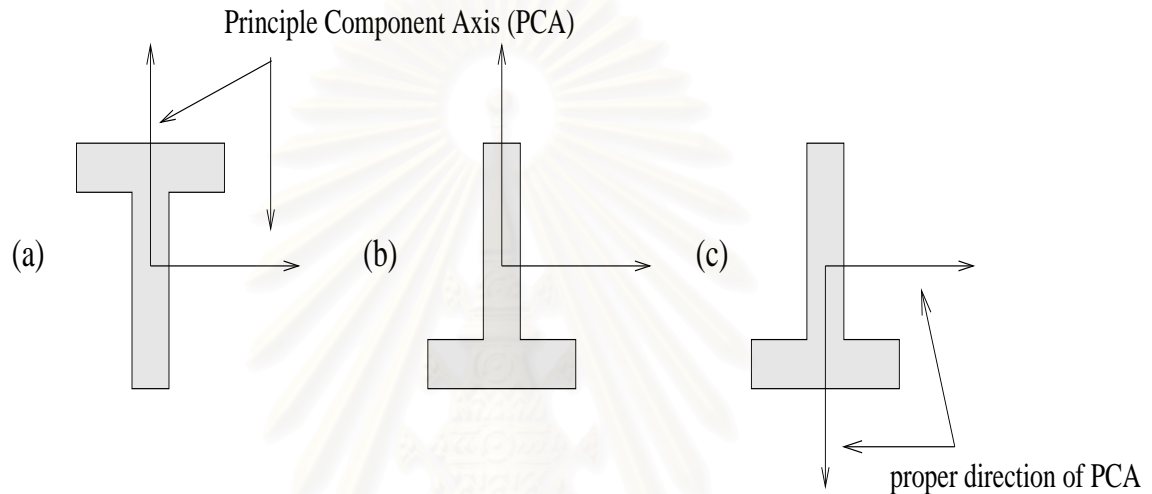


Figure 3.2: An example shows the principle component axes of image “T” and its rotated image. Both images have the same direction of principle component axes.

The initial location of each weight vector requires the information regarding the direction of the clustering aspect of the data vectors. This direction can be easily found by applying the concept of principle component analysis (PCA). Given a set of data in \mathbb{R}^2 , $\Gamma = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, suppose that unless Γ has a zero mean, its mean must be subtracted before proceeding. The principle component axis is computed from the covariance matrix as shown in the following equation:

$$\mathbf{Cov}(\Gamma) = \begin{bmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i y_i \\ \sum_{i=1}^n x_i y_i & \sum_{i=1}^n y_i^2 \end{bmatrix} \quad (3.5)$$

If a data vector with coordinate (x_i, y_i) is rotated through any angle θ , the new coordinate is defined as

$$(x'_i, y'_i) = (x_i \cos \theta + y_i \sin \theta, -x_i \sin \theta + y_i \cos \theta). \quad (3.6)$$

As a result, the covariance matrix of a set of rotated data through an angle θ , denoted by Γ_θ will become as shown below.

$$\mathbf{Cov}(\Gamma_\theta) = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \quad (3.7)$$

$$\text{where } a = \sum_{i=1}^n (x_i \cos \theta + y_i \sin \theta)^2 \quad (3.8)$$

$$b = \sum_{i=1}^n (x_i \cos \theta + y_i \sin \theta)(-x_i \sin \theta + y_i \cos \theta) \quad (3.9)$$

$$c = \sum_{i=1}^n (-x_i \sin \theta + y_i \cos \theta)^2 \quad (3.10)$$

Consequently, PCA cannot indicate the actual direction with respect to the structural aspect of the data vector cluster. It is noted that both original data vector and data vectors rotating through 180 degrees have the same covariance matrix. Hence, the principle component axis of both data vectors is the same. For example, the original and its image of letter "T" turned through 180 degrees shown in Figures 3.2 (a) and (b) have the same principle component directions. In this example, if we use the top of letter "T" as an indication for the correct direction of the principle component axis then the image in Figure 3.2 (b) should be changed as shown in Figure 3.2 (c).

To calculate the rotated angle, it is necessary to identify the proper direction. The concept of the density data vectors corresponding to both directions of PCA is used. Both directions of PCA are defined as the same direction and the opposite direction of the obtained eigenvector. The proposed algorithm to find the correct direction is called *Rotational Direction Algorithm*. Let \mathbf{e}_p be the eigenvector corresponding to the eigenvalue p of a set of data vector, Γ . Let C_1 be a set of data vectors in which its inner product with the eigenvector is not less than the

arbitrary parameter τ as shown in Eq.(3.11).

$$C_1 = \{\mathbf{x}_i | \mathbf{e}_p \cdot \mathbf{x}_i \geq \tau\} \quad (3.11)$$

and let C_2 be a set of data in which its inner product with the negative eigenvector is less than the negative value of the parameter τ as shown in Eq.(3.12).

$$C_2 = \{\mathbf{x}_i | -\mathbf{e}_p \cdot \mathbf{x}_i < -\tau\}. \quad (3.12)$$

The desired direction depends on the greater number of data vectors between the sets C_1 or C_2 . Herein, the desired direction is changed to the opposite direction of the obtained eigenvector if $\|C_1\| \geq \|C_2\|$. Otherwise, the desired direction is the same direction of the obtained eigenvector. In addition, the parameter τ keeps increasing until the size of C_1 is not equal to the size of C_2 . For example, Figure 3.3 (a) shows the eigenvector \mathbf{e} with the maximum eigenvalue of the rotated image “T” after applying PCA. Then, the sets C_1 and C_2 are calculated by Eq.(3.11) and (3.12), respectively. As shown in Figure 3.3 (b), the data vectors in C_1 are represented by a dark-colored area in the same direction of \mathbf{e} while the data vectors in C_2 are represented by a gray-colored area in the opposite direction of \mathbf{e} , denoted by $-\mathbf{e}$. Since $\|C_1\| < \|C_2\|$, the desired direction is changed to the direction of $-\mathbf{e}$ as shown in Figure 3.3 (c).

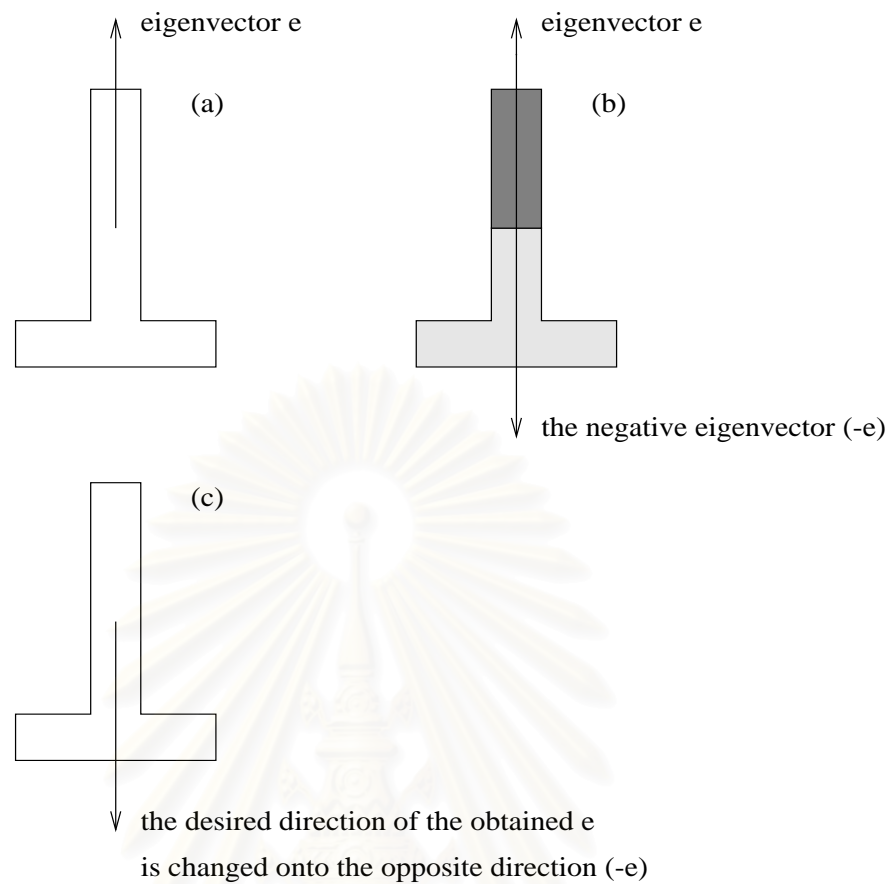


Figure 3.3: An example of the rotated image “T” illustrates the process of finding the proper direction of eigenvector of the image. (a) The eigenvector e obtained directly from PCA. (b) The data vectors in C_1 and C_2 represented by dark area and gray area, respectively. (c) The proper direction of the image changed to the negative eigenvector.

3.3 Self-Partitioning Competitive Learning

The principal goal of the Kohonen's competitive learning in Section 3.1 is to partition the data vectors into clusters where the number of clusters is defined by a user. The obtained result after learning is the center of the clusters represented by the location of weight vectors but the algorithm does not consider the location of the obtained weight vectors. For instance, Figures 3.4 (a)-(d) show the location of weight vectors after performing four trials of the Kohonen's competitive learning on the same data vectors. It is obvious that the obtained weight vectors are different in location. Figure 3.4 (a)-(b) show the location of weight vectors after performing the Kohonen's competitive learning in the first and second trials by using random initialization of weight vectors while Figure 3.4 (c)-(d) show the location of weight vectors after performing the Kohonen's competitive learning in the third and fourth trials by using fixed initialization of weight vectors. Due to these experiments, the final location of each \mathbf{w}_i depends on how initial weight vectors and selected data vectors are shown at random in the updating procedure.

The data vectors are represented as the pixel coordinates of an image. The interested problem of a scaled image is how the weight vectors with respect to the scaled data vectors and the original data vectors are placed at the same location after learning. An algorithm, named "Self-Partitioning Competitive Learning" based on the concept of Voronoi diagram is introduced to find the data partition and to guide the movements of the weight vectors of a considered data and its scaled data to the same locations.

Given any weight vector \mathbf{w}_i and any data vector \mathbf{x}_k , $H(\mathbf{w}_i)$ be a polyhedral region of data vectors closer to the weight vector \mathbf{w}_i than to any other weight vectors.

$$H(\mathbf{w}_i) = \{\mathbf{x}_k \mid \|\mathbf{x}_k - \mathbf{w}_i\| < \|\mathbf{x}_k - \mathbf{w}_j\|; \forall j \neq i\} \quad (3.13)$$

where $\|\cdot\|$ is the Euclidean norm. The mean vector of a region is defined as the summation of all data vectors of its region divided by the number of belonging data vectors as follows:

$$\mu(H(\mathbf{w}_i)) = \frac{1}{|H(\mathbf{w}_i)|} \sum_{\forall k, \mathbf{x}_k \in H(\mathbf{w}_i)} \mathbf{x}_k \quad (3.14)$$

where $|H(\mathbf{w}_i)|$ is the number of data vectors in a region $H(\mathbf{w}_i)$. Similar to a Voronoi diagram,

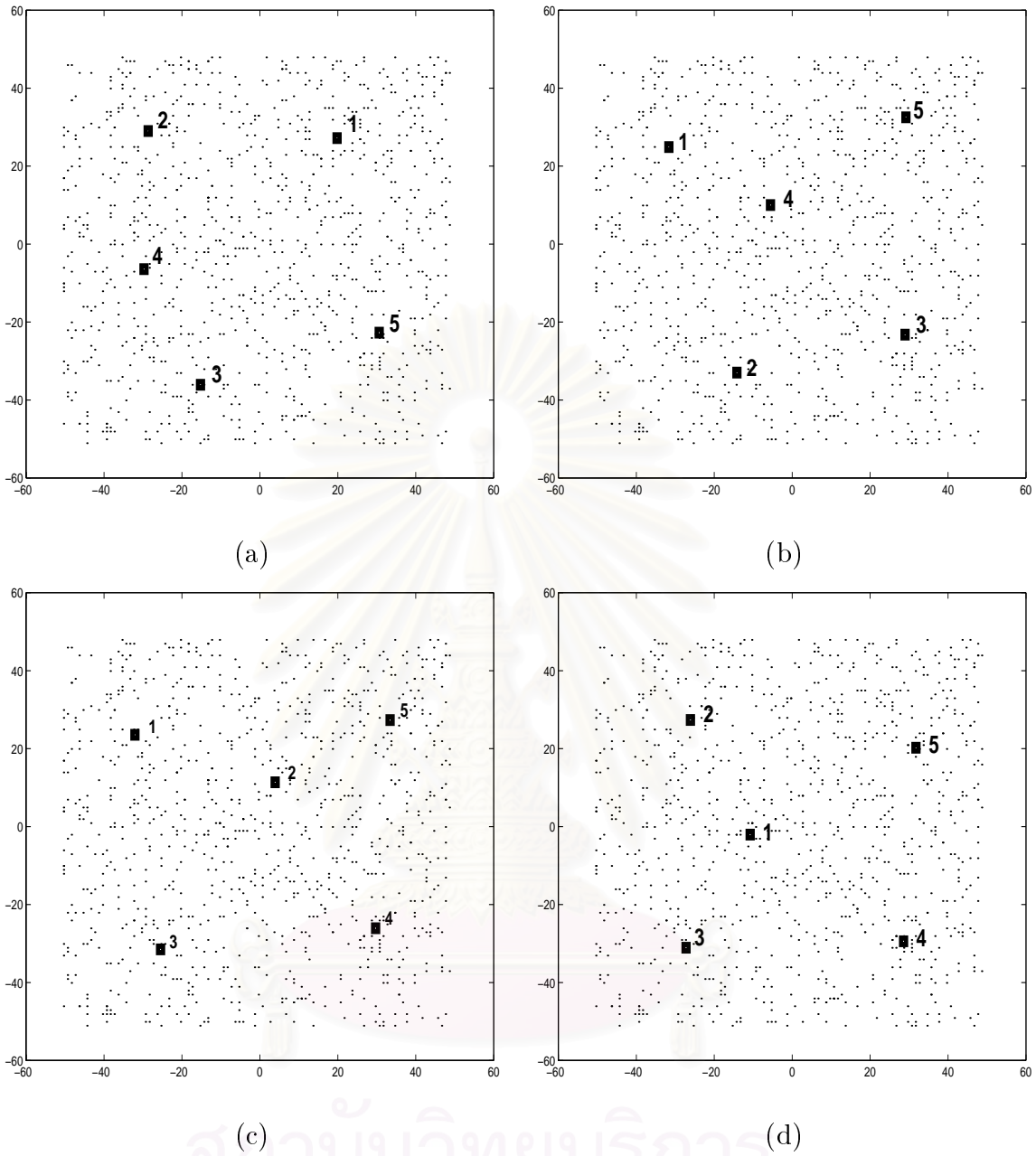


Figure 3.4: The location of weight vectors after applying the Kohonen's competitive learning to the same data vectors in order to partition the data. (a)-(b) Performing the Kohonen's competitive learning in the first and second trials by using random initialization of weight vectors. (c)-(d) Performing the Kohonen's competitive learning in the third and fourth trials by using fixed initialization of weight vectors.

the whole data are partitioned into regions corresponding to the weight vectors and each weight vector is moved closer to the mean vector of region.

Beginning with the initialization of weight vectors, we can put the weight vectors on the curve of the circular function defined in terms of $\cos \theta$ and $\sin \theta$ for the x -coordinate and y -coordinate, respectively, as follows:

$$\mathbf{w}_i = [r \cos \theta_i \quad r \sin \theta_i]^T \quad (3.15)$$

r is a constant and less than the variance of the set of data vectors. θ_i is equal to $\frac{2\pi}{m}i$ where m is the number of weight vectors. As shown in Figure 3.5, \mathbf{w}_0 is assigned by θ equal to zero. The angle of the following weight vectors $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m$ is increased by ϵ degree.

In the learning process, the regions are updated corresponding to the location of weight vectors until there is no significant difference between the data vectors of the regions at the time t and $t-1$. Thus, the self-partitioning competitive learning algorithm is written as follows. Let $\mathbf{w}_i(t)$ and $H(\mathbf{w}_i)(t)$ be the weight vector i and the polyhedral region of \mathbf{w}_i at time t .

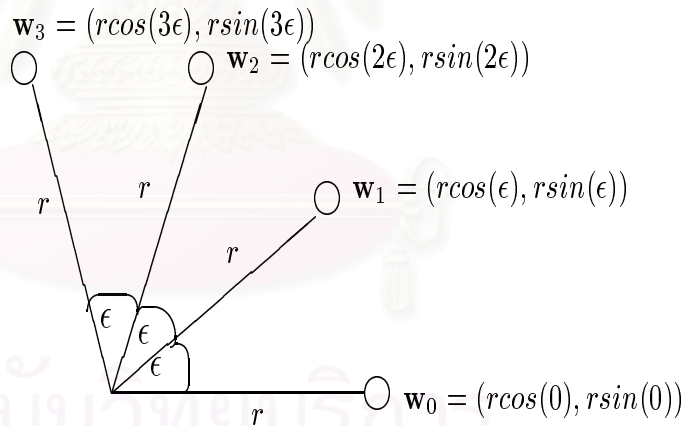


Figure 3.5: The location of initial weight vectors assigned by Eq. (3.15). The angle θ of \mathbf{w}_0 is equal to zero and the angle space of the adjacent weight vector, denoted by ϵ , is equal to $\frac{2\pi}{m}$ where m is the number of weight vectors.

Self-Partitioning Competitive Learning

1. $t=0$.
2. Initialize the weight vector \mathbf{w}_i by assigning the function:

$$\mathbf{w}_i(t) = [r \cos \theta_i \ r \sin \theta_i]^T$$

where $\theta_i = \epsilon i$ and m is the number of weight vectors.

3. **do**
4. $t = t + 1$.
5. Update each \mathbf{w}_i as follows:

$$\mathbf{w}_i(t) = \begin{cases} \mu(H(\mathbf{w}_i(t-1))) & \text{if } |H(\mathbf{w}_i(t-1))| > 0 \\ \mathbf{w}_i(t-1) & \text{if } |H(\mathbf{w}_i(t-1))| = 0 \end{cases}$$

6. **until** $(\forall i(H(\mathbf{w}_i(t)) = H(\mathbf{w}_i(t-1))) \text{ or } t > T)$
7. Obtain all \mathbf{w}_i .

3.4 Feature Extractor for Color Images

In this paper, we also consider an image with color formats. Therefore, the dimension of the data vectors is expanded in order to encode the information of an image with color intensity. Figure 3.6 gives an example of encoding pixel k with the vectors \mathbf{x}_k and \mathbf{i}_k as follows.

Each pixel of an image is considered as a data vector, which is called coordinate data vector. Let $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ be a set of coordinate data vectors such that the mean of \mathbf{X} is

$$\mu(\mathbf{X}) = \mathbf{0} \quad (3.16)$$

However, if \mathbf{X} has a nonzero mean, its mean is subtracted from every data vector to obtain a new \mathbf{x}_k , denoted by \mathbf{x}_k^{new} ,

$$\mathbf{x}_k^{new} = \mathbf{x}_k - \mu(\mathbf{X}) \quad (3.17)$$

Definition 3.1. *The normalized coordinate data vector, \mathbf{x}'_k , is defined as*

$$\mathbf{x}'_k = \frac{\alpha_1}{2\vartheta} \mathbf{x}_k \quad (3.18)$$

where α_1 is a constant, and ϑ is the maximum value of the projection of coordinate data vectors on the principle component axis with the maximum eigenvalue defined as follows.

$$\vartheta = \max\left\{\max_{\forall k}(\mathbf{x}_k \cdot \mathbf{e}), \max_{\forall k}(\mathbf{x}_k \cdot (-\mathbf{e}))\right\} \quad (3.19)$$

where \mathbf{e} is the principle component axis of \mathbf{C} with the maximum eigenvalue after applying PCA.

The intensity data vector, $\mathbf{i}_k = [r_k \ g_k \ b_k]^T$, at pixel k of an image is a vector whose elements are the intensities of red (r_k), green (g_k) and blue (b_k) colors, respectively.

Definition 3.2. The normalized intensity data vector, \mathbf{i}'_k , is defined as

$$\mathbf{i}'_k = \alpha_2 \begin{bmatrix} \frac{1}{\max_{\forall k}(\mathbf{i}_k \cdot [1 \ 0 \ 0]^T) - \min_{\forall k}(\mathbf{i}_k \cdot [1 \ 0 \ 0]^T)} & 0 & 0 \\ \frac{1}{\max_{\forall k}(\mathbf{i}_k \cdot [0 \ 1 \ 0]^T) - \min_{\forall k}(\mathbf{i}_k \cdot [0 \ 1 \ 0]^T)} & 0 & 0 \\ \frac{1}{\max_{\forall k}(\mathbf{i}_k \cdot [0 \ 0 \ 1]^T) - \min_{\forall k}(\mathbf{i}_k \cdot [0 \ 0 \ 1]^T)} & 0 & 0 \end{bmatrix} \cdot \mathbf{i}_k \quad (3.20)$$

where α_2 is a constant.

Feature extraction algorithm combines the techniques of the rotational direction and the self-partitioning competitive learning described in subsection 3.2 and 3.3, respectively. Basically, only the coordinate data vectors are changed when the image is rotated. Finding the eigenvector and applying the rotational direction algorithm can be achieved by considering only the information of the coordinate data vectors. During the self-partitioning competitive learning, the coordinates of each pixel and its intensity data vector are simultaneously considered as a vector in a five dimensional space.

Each region is obtained by applying the partitioning process in Eq.(3.13) and then calculating the mean of the partitioned region by Eq.(3.14). Finally, the weight vectors are moved to the mean of their partitioned regions. Let $\mathbf{p}_k = [\mathbf{x}_k; \mathbf{i}_k]^T$ be a data vector derived from the concatenation between the coordinate data vector and the intensity data vector, and $\mathbf{w}_i = [\mathbf{w}_i^C; \mathbf{w}_i^I]^T$ be a weight vector derived from the concatenation between the coordinate weight vector \mathbf{w}_i^C and the intensity weight vector \mathbf{w}_i^I . To apply the algorithm to color images,

we change Eq.(3.14) to the following equation:

$$\mathbf{w}_i^C(t) = \begin{cases} (\sum_{\forall k, \mathbf{p}_k \in H(\mathbf{w}_i)(t-1)} \mathbf{x}_k) / |H(\mathbf{w}_i)(t-1)| & \text{if } |H(\mathbf{w}_i)(t-1)| > 0 \\ \mathbf{w}_i^C(t-1) & \text{Otherwise.} \end{cases} \quad (3.21)$$

$$\mathbf{w}_i^I(t) = \begin{bmatrix} \min\{\mathbf{mode}_{\forall k, \mathbf{p}_k \in H(\mathbf{w}_i)}(r_k)\} \\ \min\{\mathbf{mode}_{\forall k, \mathbf{p}_k \in H(\mathbf{w}_i)}(g_k)\} \\ \min\{\mathbf{mode}_{\forall k, \mathbf{p}_k \in H(\mathbf{w}_i)}(b_k)\} \end{bmatrix} \quad (3.22)$$

where $\mathbf{w}_i^C(t)$ is the coordinate weight vector i considered at time t and $\mathbf{w}_i^I(t)$ is the intensity weight vector i considered at time t . The symbols $\mathbf{mode}_{\forall k, \mathbf{p}_k \in H(\mathbf{w}_i)}(r_k)$, $\mathbf{mode}_{\forall k, \mathbf{p}_k \in H(\mathbf{w}_i)}(g_k)$ and $\mathbf{mode}_{\forall k, \mathbf{p}_k \in H(\mathbf{w}_i)}(b_k)$ indicate the most frequent value of red (r_k), green (g_k), and blue (b_k) colors, respectively, for all k where $\mathbf{p}_k \in H(\mathbf{w}_i)$.

In general, the scaling procedure is a method in which an operator must introduce some pixels to extend the size of image. The intensity values of the new pixels are defined by the interpolating technique such as neighboring estimation, bilinear interpolation and cubic interpolation. Suppose that the technique of neighboring estimation is used to introduce the new pixels for scaling operation. Consequently, we consider the coordinate weight vector separately and the intensity weight vector in the procedure of moving weight vectors from the location at time $t-1$ to the new location at time t as shown in Eq.(3.21) and Eq.(3.22).

The algorithm of feature extraction applied to color images is developed by using *Rotational Direction* algorithm to guide the correction direction in the process of initializing weight vectors and *Self-Partitioning Competitive Clearing* algorithm to capture the color texture of the image with the scaling and intensity invariant properties. This new algorithm is named *Rotational and Scaling Invariant Self-Organizing Mapping* algorithm.

List of notations used in RSISOM algorithm

r :	a constant number used for initializing weight vectors
T :	a constant number
m :	the number of weight vectors
n :	the number of data vectors in data set
$\mathbf{p}_k = [\mathbf{x}_k; \mathbf{i}_k]^T$:	data vector k derived from the concatenation between the coordinate data vector and the intensity data vector
$\mathbf{Cov}(\mathbf{X})$:	covariance matrix of \mathbf{X}
$\mathbf{w}_i^C = [y_{i1} \ y_{i2}]^T$:	coordinate weight vector i
$\mathbf{w}_i^I = [z_{i1} \ z_{i2} \ z_{i3}]^T$:	intensity weight vector of neuron i
$\mathbf{w}_i = [\mathbf{w}_i^C; \mathbf{w}_i^I]^T$:	weight vector i derived from the concatenation between the coordinate weight vector and the intensity weight vector
$\mathbf{w}_i(t)$:	weight vector i considered at time t
$\mathbf{w}_i^C(t)$:	coordinate weight vector i considered at time t
$\mathbf{w}_i^I(t)$:	intensity weight vector i considered at time t
\mathbf{e} :	principle component axis with the maximum eigenvalue after applying PCA; $\ \mathbf{e}\ = 1$
\mathbf{e}' :	unit vector $[1 \ 0]^T$.

RSISOM algorithm

1. Compute $\mathbf{Cov}(\mathbf{X}) = \mathbf{X}\mathbf{X}^T$.
2. Compute the eigenvector \mathbf{e} using the maximum eigenvalue.
3. Determine the proper direction of \mathbf{e} using *Rotational Direction Algorithm*.
4. Compute \mathbf{p}_k^{new} by concatenating the normalized coordinate data vector \mathbf{x}'_k and the normalized intensity data vector \mathbf{i}'_k as defined by Eq. 3.18 and Eq. 3.20
5. Initialize all $\mathbf{w}_i(t)$ when $t = 0$ using *Initializing Weight Algorithm*.
6. Determine each region $H(\mathbf{w}_i)(t)$ when $t = 0$ as follows

$$H(\mathbf{w}_i)(t) = \{\mathbf{p}_k \mid \|\mathbf{p}_k - \mathbf{w}_i(t)\| < \|\mathbf{p}_k - \mathbf{w}_j(t)\|; \forall j \neq i\}.$$

7. $t = t + 1$.

8. Update each $\mathbf{w}_i(t)$ as follows:

$$8.1. \quad \mathbf{w}_i^C(t) = \begin{cases} (\sum_{\forall k, \mathbf{p}_k \in H(\mathbf{w}_i)(t-1)} \mathbf{x}_k) / \|H(\mathbf{w}_i)(t-1)\| & \text{if } \|H(\mathbf{w}_i)(t-1)\| > 0 \\ \mathbf{w}_i^C(t-1) & \text{Otherwise.} \end{cases}$$

$$8.2. \quad \mathbf{w}_i^I(t) = \begin{bmatrix} \min\{\mathbf{mode}_{\forall k, \mathbf{p}_k \in H(\mathbf{w}_i)}(r_k)\} \\ \min\{\mathbf{mode}_{\forall k, \mathbf{p}_k \in H(\mathbf{w}_i)}(g_k)\} \\ \min\{\mathbf{mode}_{\forall k, \mathbf{p}_k \in H(\mathbf{w}_i)}(b_k)\} \end{bmatrix}$$

$$8.3. \quad \mathbf{w}_i(t) = [\mathbf{w}_i^C; \mathbf{w}_i^I].$$

8.4. Determine each region $H(\mathbf{w}_i)(t)$ as follows:

$$H(\mathbf{w}_i)(t) = \{\mathbf{p}_k \mid \|\mathbf{p}_k - \mathbf{w}_i(t)\| < \|\mathbf{p}_k - \mathbf{w}_j(t)\|; \forall j \neq i\}.$$

9. **if** $(\forall i(H(\mathbf{w}_i)(t) = H(\mathbf{w}_i)(t-1))$ **or** $t > T)$ **then** obtain all \mathbf{w}_i **else** go to step 6.

Rotational Direction Algorithm

1. Set threshold $\tau_1 = 0$, $\epsilon_1 = 0.2\frac{\pi}{2}$, and dummy $\mathbf{u} = \mathbf{e}$
2. **do**
3. $\mathbf{u} = [u_1 \cos(\tau_1) + u_2 \sin(\tau_1) \quad -u_1 \sin(\tau_1) + u_2 \cos(\tau_1)]^T$
4. Compute maximum count $M = \max_{\forall i}(\mathbf{u} \cdot \mathbf{x}_i)$
5. Set threshold $\tau_2 = 0$, $\epsilon_2 = 1$, $count1 = 0$, and $count2 = 0$
6. **do**
7. **for** each coordinate data vector \mathbf{x}_i **do**
8. **if** $\mathbf{e} \cdot \mathbf{x}_i \geq \tau_2$ **then** $count1 = count1 + 1$
9. **if** $\mathbf{e} \cdot \mathbf{x}_i < -\tau_2$ **then** $count2 = count2 + 1$
10. **end**
11. **if** $count1 \neq count2$ **then** go to step 17
12. $\tau_2 = \tau_2 + \epsilon_2$
13. **until** $\tau_2 \geq M$

14. $\tau_1 = \tau_1 + \epsilon_1$
15. **until** $\tau_1 \geq \frac{\pi}{2}$
16. **if** $count1 \geq count2$ **then** set $\mathbf{e} = \mathbf{e}$
17. **if** $count1 < count2$ **then** set $\mathbf{e} = -\mathbf{e}$

Initializing Weight Algorithm

1. Compute the angle θ between \mathbf{e} and \mathbf{e}'
2. $\theta = \cos^{-1}(\mathbf{e} \cdot \mathbf{e}')$
3. $\epsilon = \frac{2\pi}{m}$
4. **for** $1 \leq i \leq m$ **do**
5. $\mathbf{w}_i = (r \cos \theta, r \sin \theta, 0, 0, 0)$
6. $\theta = \theta + \epsilon$
7. **end**

3.4.1 Time Complexity of the Proposed Feature Extractor

Theorem 3.1. Rotational and Scaling Invariant Self-Organizing Mapping (RSISOM) *algorithm extracts the invariant feature of an image in $O(n)$ time where n is the number of data vectors.*

Proof. The algorithm begins with computing the covariance matrix, $\mathbf{Cov}(\mathbf{X})$, in which size of matrix \mathbf{X} is $2 \times n$. Let f_{pq} be an element of the covariance matrix \mathbf{X} , and a_{ij} and b_{ij} be an element of the matrix \mathbf{X} and an element of the matrix \mathbf{X}^T , respectively. The matrix multiplication between \mathbf{X} and \mathbf{X}^T is written as follows:

$$f_{pq} = \sum_{i=1}^n a_{pj} b_{iq} \quad (3.23)$$

Therefore, the arithmetic complexity time to compute the covariance matrix $\mathbf{Cov}(\mathbf{X})$ of size equal to 2×2 is $O(4n)$. Then, the maximum eigenvalue and the eigenvector, then, are computed

in a constant time. To determine the proper direction of the eigenvector \mathbf{e} , *Rotational Direction Algorithm* is applied. Its time complexity is, obviously, $O(\frac{M}{\epsilon_2}n)$ where $\frac{M}{\epsilon_2}$ is much less than n . Let m be the number of weight vectors. Then, the location of weight vectors is assigned by *Initializing Weight Algorithm*. Its complexity time is $O(m)$.

Finally, the learning step is to update the weight vectors either until there is no significant difference between the region of the data vectors at the present time, $H(\mathbf{w}_i)(t)$, and the region of the data vectors at the previous time, $H(\mathbf{w}_i)(t-1)$, or until the time exceeds the constant number, T . For each iteration, the coordinate weight vector moves closer to the arithmetic mean of its region of coordinate data vectors and the intensity weight vector is set to the mode value of its region of intensity data vectors. Finding the elements of each region, $H(\mathbf{w}_i)$, requires n times. Therefore, the total run time is $O(mn)$ for m regions. To compute the function **mode** for each region, it takes $O(n)$ in the worst case. Therefore, the total run time is $O(nm)$ for m regions.

Thus, *RSISOM algorithm* requires $O(4n) + O(m) + O(n) + O(m) + O(mn) + O(mn)$ equal to $O(mn)$ and it is equals to $O(n)$ if m is much less than n . \square

3.4.2 Feature Extraction Based on The Location of Weight Vectors

After locating the weight vectors, $\mathbf{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_m\}$ where m is the number of weight vectors, by *RSISOM algorithm*, the feature vector of an image, \mathbf{f} , is defined as follows:

$$\mathbf{f} = [\mathbf{a}_1; \dots; \mathbf{a}_i; \dots; \mathbf{a}_m]^T \quad (3.24)$$

Symbol “;” means the concatenation. Element \mathbf{a}_i is the concatenation between the coordinate weight vector, \mathbf{w}_i^C , after rotating through an angle, θ , and the intensity weight vector, \mathbf{w}_i^I and it is explicitly computed by

$$\mathbf{a}_i = \left[\begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \mathbf{w}_i^C; \mathbf{w}_i^I \right]^T \quad (3.25)$$

where $\theta = \cos^{-1}(\mathbf{e} \cdot \mathbf{e}')$.

The obtained feature vector of an image is appropriate to proceed through the classification invariant to rotation, scaling and intensity changing as shown in Chapter 4.



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

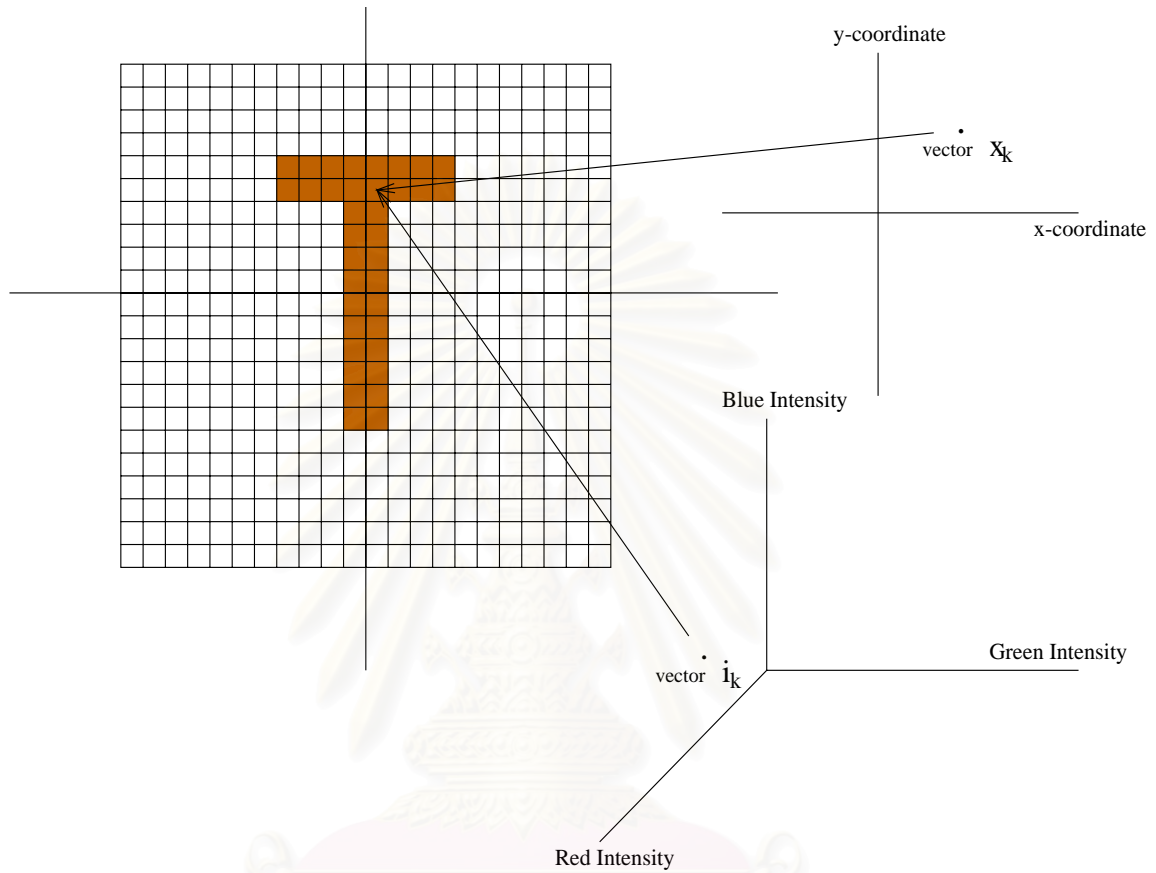


Figure 3.6: Given an example of a color image “T”, there are two encoded vectors from a pixel. The first vector, $\mathbf{x}_k = [a_1 \ a_2]^T$, is the information of x -coordinate and y -coordinate at pixel k and the other, $\mathbf{i}_k = [b_1 \ b_2 \ b_3]^T$ is the information of intensities of red, green and blue colors at pixel k . The concatenation of both vectors is denoted by a new vector, $\mathbf{p}_k = [a_1 \ a_2 \ b_1 \ b_2 \ b_3]^T$ in a 5-dimensional space.

CHAPTER IV

Experimental Results

In this chapter, the proposed algorithm, RSISOM, is tested and evaluated on a number of artificial and real data sets. The experiments are designed to evaluate the success on the distinguishability and robustness of rotation, scaling and intensity change. The performance is measured by Euclidean distance among the feature vectors of images.

4.1 Test Data

There are four data sets of which all tested images are constrained by the following properties:

1. Color intensity of each pixel is in RGB format.
2. Size of images is at least 256×256 pixels.
3. The background of each image is white.

Firstly, Dataset A consists of 34 different images obtained from clip-arts on the Internet [see Appendix C]. Next, Dataset B and data set C are the synthesis images of airplane model **A** and **F**, respectively. Figure 4.1 (a)-(f) illustrate six airplane types, namely **A-0**, **A-1**, **A-2**, **A-3**, **A-4**, and **A-5**. Likewise, the six airplane types in Figure 4.2 (a)-(f) are **F-0**, **F-1**, **F-2**, **F-3**, **F-4**, and **F-5**. Lastly, Dataset D is the synthesis data test of four images; Figure 4.3 (a)-(b) have similar superimposing structures and colors. They only differ in shapes, circles and squares. Figure 4.3 (c)-(d) consists of the same objects, namely circles, triangles, and squares. The objects in both figures are located at different locations. Herein, the different images mean that each image is different in shape, colored texture, or both of them.



Figure 4.1: Dataset **B** consists of six airplane types, namely **A-0**, **A-1**, **A-2**, **A-3**, **A-4**, and **A-5**, which all are of the same model **A**.



Figure 4.2: Dataset **C** consists of six airplane types, namely **F-0**, **F-1**, **F-2**, **F-3**, **F-4**, and **F-5**, which all are of the same model **F**.

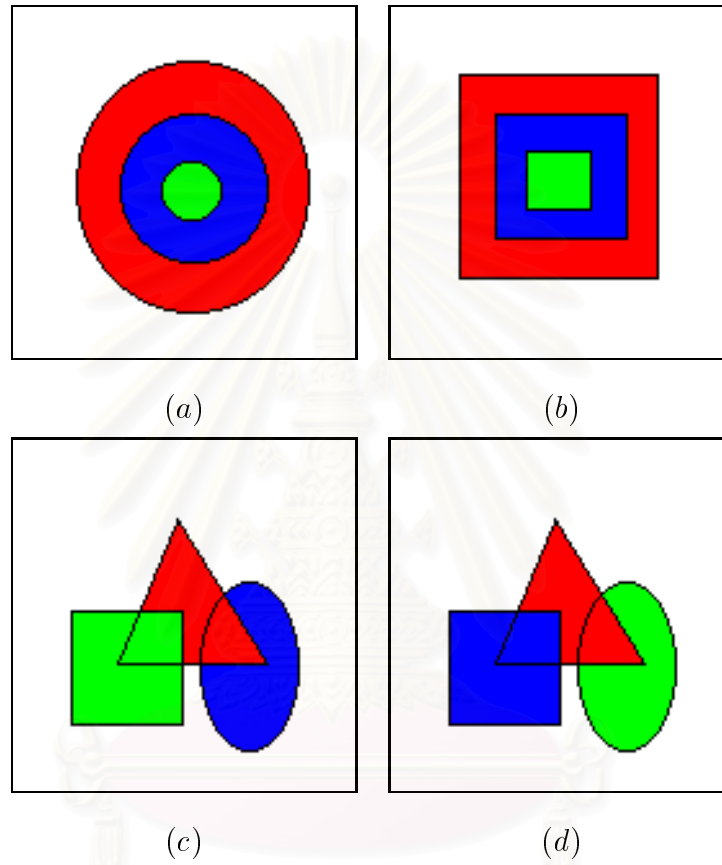


Figure 4.3: Dataset **D** has four images: (a)-(b) have similar superimposing structures and colors but different shapes, circles and squares. (c)-(d) consists of the same objects, namely circles, triangles, and squares.

Table 4.1: Parameter values used for all data sets.

Parameter	Value
T : maximum number of iterations	200
α_1 : the constant used for normalization of coordinate data vectors	100
α_2 : the constant used for normalization of intensity data vectors	20/255

To prove that the features extracted by RSISOM is invariant to rotation, scaling and color intensity changes, a computer can create 83 transformed images [see Appendix C] from each original image as follows:

1. 24 rotated images. They are produced by rotating the original image through 0 – 360 degrees.
2. 5 scaled images. They are produced by scaling the original image up to 2 times as large as the original size.
3. 10 intensity-changed images. They are produced by increasing or decreasing the intensity of R, G and B planes up to 20% of the original intensity.
4. 44 mixed-transformation images. They are produced by combining all above transformations; for example, the original image is rotated through 210 degrees, then scaled up 1.6 times of the original size, and, last, is increased the intensity 10% of the original intensity.

4.2 Setting and Definitions

Applying RSISOM to data sets, the parameter values are set as shown in Table 4.1. The number of weight vectors used for each data set is set with respect to the intrinsic information of images of data set as shown in Table 4.2.

Table 4.2: The number of weight vectors used for data sets.

Parameter	The number of weight vectors used for data sets			
	A: Cliparts	B: Model A	C: Model F	D: Geometry
m	12	20	20	10

Let \mathbf{v}_i be a feature vector of an original image i and $\mathbf{f}_j^{[i]}$ be a feature vector of the image j trasformed from the original image i . Conveniently, the class of the original image i is represented by $[i]$. The classification measurement of the feature vector $\mathbf{f}_k^{[j]}$ belonging to the class $[i^*]$ is given as the following:

$$i^* = \arg \min_{\forall i} (\|\mathbf{f}_k^{[j]} - \mathbf{v}_i\|) \quad (4.1)$$

The misclassification of feature vector $\mathbf{f}_k^{[j]}$ occurs when the obtained class $[i^*]$ is not in the same class as $\mathbf{f}_k^{[j]}$. In the other hand, the transformed image $\mathbf{f}_k^{[j]}$ is misclassified if $i^* \neq j$.

4.3 Results

We test our proposed feature extraction algorithm against data set A described in Section 4.1. The features of all original images and their transformed versions are extracted and classified by Eq.(4.1). The accuracy of the classification is shown in the first column in Table 4.3. The other columns in Table 4.3 show the maximum, minimum and standard deviation (S.D.) of the distance between $\mathbf{f}_k^{[i]}$ and \mathbf{v}_{m_i} denoted by D_f . The average of each column is computed at the last row in Table 4.3.

To demonstrate the power of RSISOM discrimination among images with the same shape but different color texture, RSISOM is tested against data set B in Section 4.1. The accuracy of the invariant classification of data set B is 95.59 %. Table 4.4 shows the average boundary distance of each invariant airplane type. Table 4.5 shows the distance among each feature vectors of original images of model A.

Table 4.3: The results of feature classification of data set A are presented in classes of transformations, i.e., rotation, scaling, changing the intensity and combining all. The correctness of invariant classification is shown in the second column following the maximum, minimum and standard deviation (S.D.) of the distance between $\mathbf{f}_k^{[i]}$ and \mathbf{v}_{m_i} denoted by D_f in the third, fourth, fifth and sixth columns, respectively.

	Correctness(%)	Max D_f	Min D_f	Mean of D_f	S.D. of D_f
Rotation	100.000	0.000	0.000	0.000	0.000
Scaling	87.65	0.374	0.022	0.155	0.133
Intensity	88.89	0.377	0.028	0.122	0.136
Combination	93.58	0.170	0.130	0.175	0.017
Mean	94.51	0.230	0.045	0.113	0.072

Table 4.4: The average boundary distance ($\times 10^{-2}$) of each invariant class denoted by Mean of D_f .

Airplane type	Average boundary distance (Mean of D_f)
A-0	16.0030
A-1	22.1201
A-2	18.3161
A-3	14.9190
A-4	20.2280
A-5	11.2144

Table 4.5: The distance measure ($\times 10^{-2}$) among each feature vectors of original images, airplane type \mathbf{A} , $\|\mathbf{v}_i - \mathbf{v}_j\|$.

$\mathbf{v}_{(A1)}$	57.3894				
$\mathbf{v}_{(A2)}$	50.4136	55.5353			
$\mathbf{v}_{(A3)}$	56.5283	57.7535	55.8296		
$\mathbf{v}_{(A4)}$	44.8099	53.5162	52.4130	59.6914	
$\mathbf{v}_{(A5)}$	55.0028	60.6946	46.5255	59.8578	54.2431
	$\mathbf{V}_{(A0)}$	$\mathbf{V}_{(A1)}$	$\mathbf{V}_{(A2)}$	$\mathbf{V}_{(A3)}$	$\mathbf{V}_{(A4)}$

In additional, RSISOM can discriminates effectively among images with the same color texture but different shape. The distances between the feature vectors of original airplane of model \mathbf{A} and of model \mathbf{B} , which are extracted by RSISOM, are shown in Table 4.6. RSISOM also performs efficiently to differentiate between a pair of images in data set D as shown in Table 4.7.

4.3.1 Robustness of RSISOM Under Scaling

The test data in Section 4.1 is used to compare the robustness of scaling of RSISOM with Zernike moment. It contains 34 sets of scaled images but the colors of those images are changed to gray-leveled intensity. The feature vector applied by Zernike moments, \mathbf{z}^p , in this comparison uses the parameter p equal to 12, namely,

$$\mathbf{z}^{12} = [A_{12,0} \ A_{12,2} \ A_{12,4} \ \dots \ A_{12,12}]^T \quad (4.2)$$

Similar to the measure in Section 4.2, the classification accuracy is measured by Euclidean distance. Fig. 4.3.1 shows that RSISOM is robust to scaling but Zernike moment is more sensitive when the scaling factor is close to two. The scaling factors used in the other experiments [21, 22] are less than 1.3 which our ranges from 1.2 to 2.

Table 4.6: The distance measure ($\times 10^{-2}$) between each pair of feature vectors of original images (i,j) , $\|\mathbf{v}_i - \mathbf{v}_j\|$.

	Distance between each pair of feature vectors of original images
(A1, F1)	46.1211
(A2, F2)	68.1502
(A3, F3)	57.9190
(A4, F4)	65.1002
(A5, F5)	73.3345

Table 4.7: The distance measure ($\times 10^{-2}$) between each pair of feature vectors of original images (i,j), $\|\mathbf{v}_i - \mathbf{v}_j\|$.

	Distance between each pair of feature vectors of original images
(Fig. 4.3(a), Fig 4.3(b))	22.8911
(Fig. 4.3(c), Fig 4.3(d))	35.4245

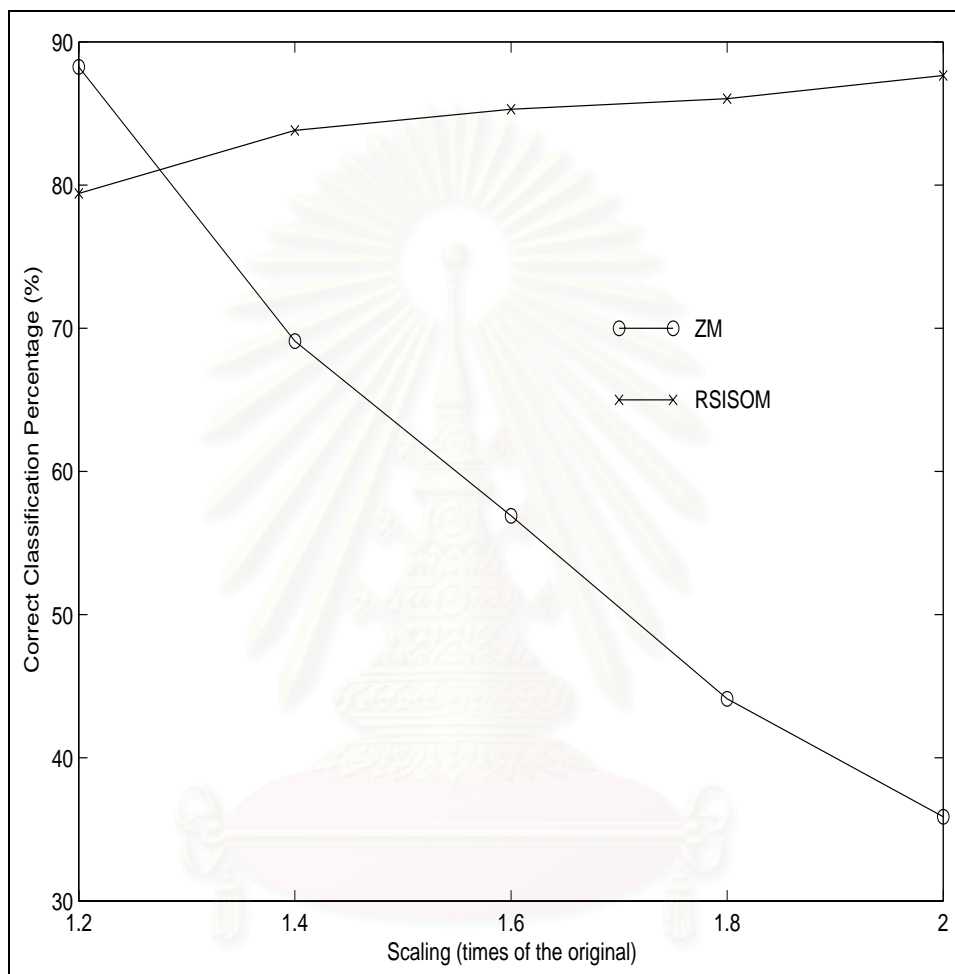


Figure 4.4: The correctness percentage of both Zernike moments (ZM) and RSISOM when they test against the scaled data described in Section 4.1.

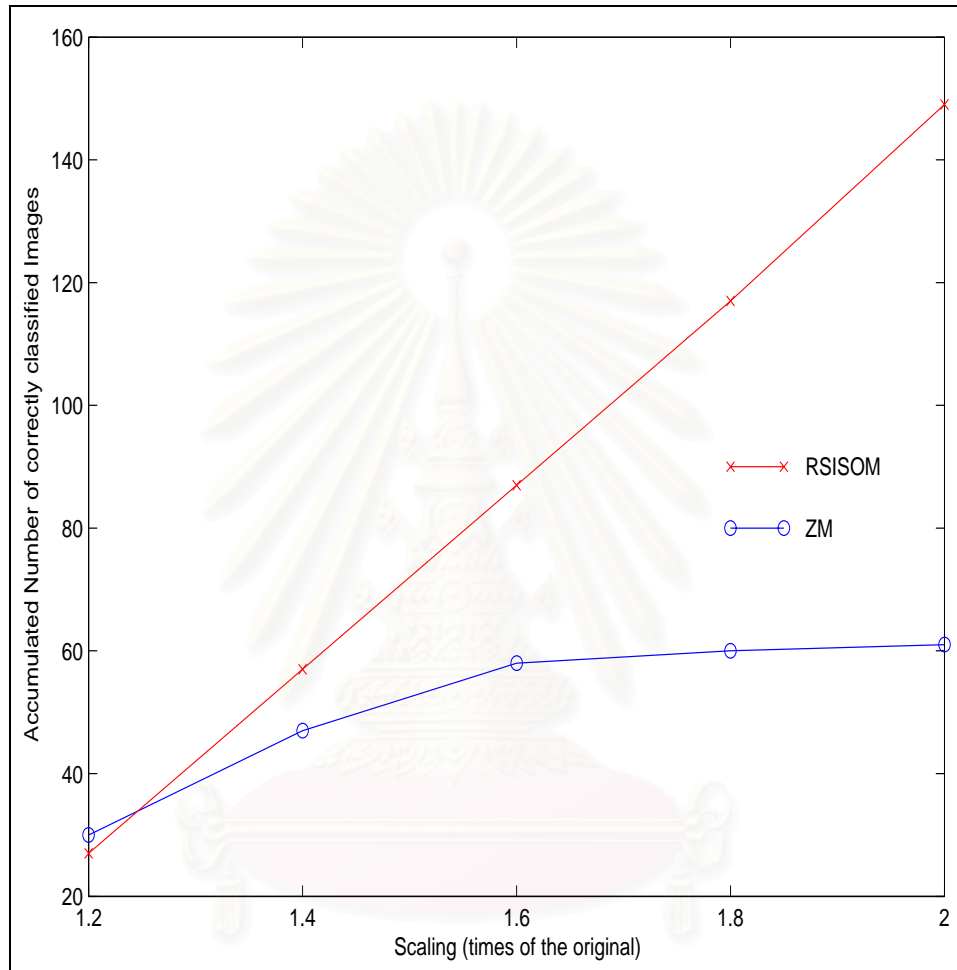


Figure 4.5: The accumulated number of images recognized correctly by both Zernike moments (ZM) and RSISOM with the parameter of scaling from 1.2 to 2.0 times of the original.

4.3.2 Robustness of RSISOM Under Color Intensity Changes

The robustness of RSISOM with respect to color intensity change are compared with fuzzy color histogram. In fuzzy color histogram, the fuzzy c-mean algorithm is applied to compute the membership grades. Following the parameters suggested in [32], the value of m and the number of clusters are set to 1.9 and 30, respectively. Fig 4.6 shows that our approach consistently outperforms fuzzy color histogram when the intensity is increased from 5 to 45, i.e., the intensity changes are equal to +5, +10, +15, + 20, +25, +30, +35, +40 and +45 while the changes are set between 5 to 25 in [32]. The extracted features of RSISOM are deployed against five dimensional data composed pixels' coordinate and color intensity but which of fuzzy color histogram excludes information of pixels' coordinate. As a result, RSISOM is more robust than fuzzy color histogram in color intensity change.



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

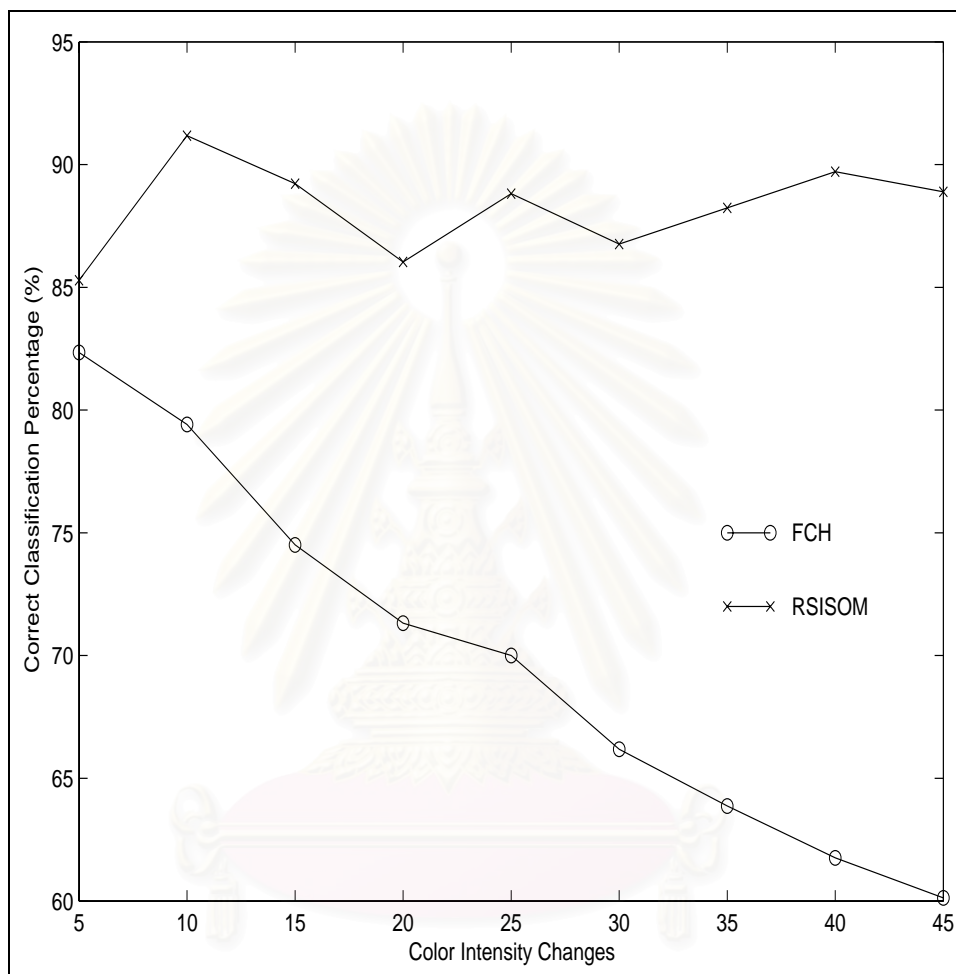


Figure 4.6: The correctness percentage of both fuzzy color histogram denoted by FCH and RSISOM when they test against the intensity-changed images described in Section 4.1.

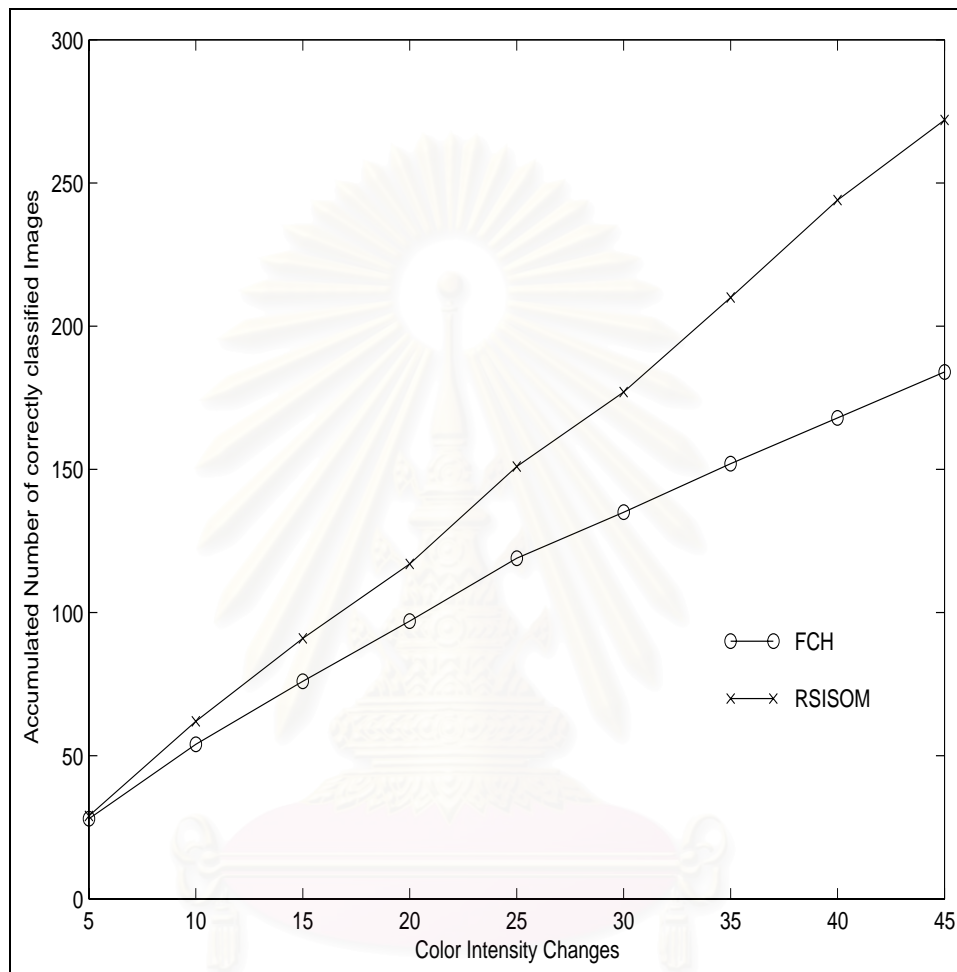


Figure 4.7: The accumulated number of images recognized correctly by both fuzzy color histogram denoted by FCH and RSISOM with the parameter of intensity change from +5 to +45.

CHAPTER V

Conclusion

A new algorithm based on the concept of self-organizing mapping and principle component analysis is introduced to solve the problem of invariant pattern recognition. The experiments indicate that the proposed algorithm extracts successfully the feature of an image with the capability to capture the color-texture of an image while it preserves the invariant capability against rotation, scaling and color intensity. The experimental results performed on the test data in Chapter 4 show that their extracted features can be successfully discriminated among others. The averaged percentage of correct classification of data set A and of data set B is 94.51 % and 95.59 % as indicated in Section 4.3.

5.1 Invariance Capability

Feature extraction is a mapping function addressed in Section 2.1. The perfect mapping \mathbf{T} would map the image f and its transformed image f' into one point in the invariance space. Most of mapping \mathbf{T} are not perfect so all feature vectors of all the images produced by rotating, scaling and changing their color intensity are grouped within some boundary. Therefore, the achievement of mapping T is evaluated by the Euclidean distance between the feature vectors of the original image and of its transformed image, called boundary distance. The considered transformations in this dissertation are rotation, scaling and color intensity change.

Table 4.3 indicates the efficiency of the proposed algorithm that maps all images of an equivalent class into the feature space. The equivalence class is defined as transformations of images under rotation, scaling and changing color intensity in this dissertation. The values of

the maximum, minimum and standard deviation of the distance between the feature vectors of a transformed image and its original image indicate the performance of the algorithm extracting the feature.

Rotation Invariance

Invariance under rotation is completely achieved by the proposed algorithm, *Rotational Direction* algorithm demonstrated in 3 as indicated in Table 4.3. The classification performance is 100 % and the boundary distance is equal to zero.

Scaling and Color Intensity Invariance

The scaling operation on an image not only extends the original location of coordinate data but also introduces new data by interpolating the existing data. Therefore, the scaled image might be distorted if the scaling factor is not a whole number. Likewise, the degree of changing the color intensity probably produces the new image much different to the original image. As a result, the proposed algorithm, *Self-Partitioning Competitive* algorithm, transformations such as scaling and increasing intensity level of an image are not perfect. However, RSISOM is much more invariant to scaling than Zernike moment when the scaling factor greater than 1.3 as illustrated in Figure 4.3.1-4.3.1. According to robustness of changing intensity level, RSISOM also outperforms fuzzy color histogram.

5.2 Distinguishability

Regarding with the technique of color metric such as fuzzy color histogram, it does not consider the actual texture of an image, In Chapter 4, the synthetic images of data set D as shown in Figure 4.3 are distinguishable by using RSISOM as shown in Table 4.7.

Nevertheless, the remarkable performance of RSISOM is able to distinguish among airplane type in data set B as shown in Table 4.5. Six airplanes in data set B have the same shape of model **A** but different color texture. Co-occurrence technique cannot differentiate two airplanes



(a) A-1



(b) F-1



(c) A-2



(c) F-2



(d) A-3



(e) F-3



(d) A-4



(e) F-4



(d) A-5



(e) F-5

Figure 5.1: Pairs of airplane types with the similar color texture.

having the same color texture but different models, for example, pairs of airplane types with the similar color texture as shown in Figure 5.1 while RSISOM can differentiate between two different type but similar texture such as pair of **A-1** and **F-1**, **A-2** and **F-2**, **A-3** and **F-3**, **A-4** and **F-4**, **A-5** and **F-5**. The result of RSISOM is shown in Table 4.6.

5.3 Future Work

One possible future improvement on RSISOM is to integrate the concept of Hierarchy. Hierarchical Rotational and Scaling Invariant Self-Organizing Map (Hierarchical RSISOM) will be investigated further as a modified RSISOM algorithm with hierarchical architecture. RSISOM has the complexity in the order of $O(n)$, where n is the number of data vectors, as proved in Theorem 3.1. Generally, a search algorithm with $O(\log n)$ complexity can be obtained if the neurons are arranged in a tree. A hierarchical SOM is a tree-structured neural network composed of independent SOMs that is capable of representing hierarchical relations between the input data. Barbalho and Batista [69, 70] present the applications of hierarchical SOM for image compression and handwritten digit recognition and their experimental results show better performance of hierarchical SOM, mainly regarding to the processing time.

To improve the classification accuracy, the structure of hierarchical RSISOM is appropriately designed for extracting the feature of a color image. The important information of a color image comprises pixel coordinates and R, G and B intensity values. Thus, the structure of the hierarchy consists of two layers. The first layer has only one RSISOM performing on the coordinate data vectors. The second layer has several independent scaling invariant self-organizing maps (SISOM) of which each computes on the intensity data vector. As a result, the hierarchical RSISOM possibly outperforms RSISOM without hierarchy concept.

REFERENCES

- [1] E. Barnard and D. Casasent, "Invariance and neural nets," *IEEE Transactions on Neural Networks*, vol. 2, pp. 498–508, 1991.
- [2] H. Burkhardt and S. Siggelkow, *Nonlinear Model-Based Image Video Processing and Analysis*. John Wiley and Sons, 2000.
- [3] C. Chen, G. Gagaudakis, and P. Ronsin, "Content-based image visualization," in *Proceedings of IEEE International Conference on Information Visualization*, 2000, pp. 13–18.
- [4] E. Demir, L. Akarun, and E. Alpaydin, "Two-stage approach for pose invariant face recognition," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2000, pp. 2342–2344.
- [5] M. Egmont-Petersen, D. de Ridder, and H. Handels, "Image processing with neural networks—a review," *Pattern Recognition, Elsevier Science*, vol. 35(10), pp. 2279–2301, 2002.
- [6] A. Jain, R. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 4–37, 2000.
- [7] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, 2000.
- [8] Y. Chung and M. Wong, "Handwritten character recognition by Fourier descriptors and neural networks," in *Proceedings of IEEE Region 10 Annual Conference on Speech and Image Technologies for Computing and Telecommunications (TENCON'97)*, vol. 1, 1997, pp. 391–394.
- [9] X. Dai and S. Khorram, "A feature-based image registration algorithm using improved chain-code representation combined with invariant moments," *Proceedings of IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, pp. 2351–2362, 1999.
- [10] R. Gonzalez and R. Woods, *Digital Image Processing*. Massachusetts: Addison Wesley, 1993.
- [11] B. Keal and A. Krzyzak, "Piecewise linear skeletonization using principle curves," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 59–74, 2002.
- [12] J. Pettier and J. Camillerapp, "Script representation by a generalized skeleton," in *Proceedings of IEEE International Conference on Document Analysis and Recognition*, 1993, pp. 850–853.
- [13] Y. Xirouhakis and S. Kollias, "Affine-invariant curve normalization for shape-based retrieval," *IEEE International Conference on Pattern Recognition*, vol. 1, pp. 1015–1018, 2000.
- [14] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York: Springer, 2001.
- [15] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press, 1995.
- [16] D. De Ridder, A. Höekstra, and R. Duin, "Feature extraction in shared weights neural networks," in *Proceedings of The 2nd annual conference of the Advanced School for Computing and Imaging*, 1996, pp. 289–294.

- [17] K. Fukushima and N. Wake, "Handwritten alphanumeric character recognition by the neocognitron," *IEEE Transaction on Neural Networks*, vol. 2, pp. 355–365, 1991.
- [18] S. Lawrence, C. Giles, A. Tsoi, and A. Back, "Face recognition: a convolutional neural-network approach," *IEEE Transaction on Neural Networks*, vol. 8, pp. 98–113, 1997.
- [19] Y. Le Cun, O. Matan, B. Boser, J. S. Denker, D. Henderson, R. Howard, W. Hubbard, L. Jacket, and H. Baird, "Handwritten zip code recognition with multilayer networks," in *Proceedings of The 10th International Conference on Pattern Recognition*, vol. 2, 1990, pp. 35–40.
- [20] A. Delopoulos, A. Tirakis, and S. Kollias, "Invariant image classification using triple-correlation-based neural networks," *IEEE Transactions on Neural Networks*, vol. 5, 1994.
- [21] P. Lisboa and S. Perantonis, "Invariant pattern recognition using third-order networks and Zernike moments," in *Proceedings of IEEE International Joint Conference on Neural Networks*, vol. 2, 1991, pp. 1421–1425.
- [22] S. Perantonis and P. Lisboa, "Translation, rotation, and scale invariant pattern recognition by high-order neural networks and moment classifiers," *IEEE Transactions on Neural Networks*, vol. 3, pp. 241–251, 1992.
- [23] M. Reid, L. Spirkovska, and E. Ochoa, "Rapid training of higher-order neural networks for invariant pattern recognition," in *Proceedings of IEEE International Joint Conference on Neural Networks*, vol. 1, 1989, pp. 689–692.
- [24] A. Gollamudi, P. Calvin, G. Yuen, M. Bodruzzaman, and M. Malkani, "Pulse coupled neural network based image classification," in *IEEE Proceedings of the Thirtieth South-eastern Symposium on System Theory*, 1998, pp. 402–406.
- [25] J. Johnson, "Time signatures of images," *IEEE International Conference on Neural Network*, vol. 2, pp. 1279–1284, 1994.
- [26] J. Karvonen and M. Simila, "Classification of sea ice types from scansar radarsat images using pulse-coupled neural networks," in *Proceedings of IEEE Geoscience and Remote Sensing Symposium*, vol. 5, 1998, pp. 2505–2508.
- [27] H. Ranganath, G. Kuntimad, and J. Johnson, "Pulse coupled neural networks for image processing," in *Proceedings of IEEE Proceedings on Visualize the Future, Southeast-con'95*, 1995, pp. 37–43.
- [28] H. Rughooputh and S. Rughooputh, "A pulse-coupled-multilayer perceptron hybrid neural network for condition monitoring," in *Proceedings of IEEE Aficon 1999*, vol. 2, 1999, pp. 749–752.
- [29] A. Khotanzad and Y. Hong, "Invariant image recognition by Zernike moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, 1990.
- [30] C. Teh and R. Chin, "On image analysis by the methods of moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, 1988.
- [31] A. Wallin and O. Kubler, "Complete sets of Complex Zernike Moment invariants and the role of the pseudoinvariants," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 1106–1110, 1995.
- [32] J. Han and K. Ma, "Fuzzy color histogram and its use in color image retrieval," *IEEE Transactions on Image Processing*, vol. 11, pp. 944–952, 2002.

- [33] S. Clippingdale and R. Wilson, "Self-similar neural networks based on a Kohonen learning rule," *Neural Networks*, vol. 9, pp. 747–763, 1996.
- [34] L. Bongkyu, C. Yookun, and C. Seongwon, "New invariant pattern recognition system based on preprocessing and reduced second-order neural network," in *Proceedings of IEEE International Conference on Neural Networks*, 1995, pp. 2099–2102.
- [35] L. Bon-Kyu, K. Dong-Kyu, C. Yoo-Kun, L. Heong-Ho, and H. Hee-Yeung, "Reduced shift invariant second order neural networks using principal component analysis and pixel combinations," in *Proceedings of IEEE World Congress on Computational Intelligence*, 1994, pp. 4283–4287.
- [36] L. Heung-Ho, K. Hee-Young, and H.-Y. H., "Scale and rotation invariant pattern recognition using complex-log mapping and translation invariant neural network," in *Proceedings of IEEE World Congress on Computational Intelligence*, 1994, pp. 4306–4308.
- [37] S. Kroner, "Adaptive averaging in higher order neural network for invariant pattern recognition," in *Proceedings of IEEE International Conference on Neural Networks*, 1995, pp. 2438–2444.
- [38] H. Y. Kwon, B. C. Kim, D. S. Cho, and H. Y. Hwang, "Scale and rotation invariant pattern recognition using complex-log mapping and augmented second order neural network," *IEEE Electronics Letters*, vol. 29, pp. 620–621, 1993.
- [39] J. Wu and C. Jyh-Yeong, "Invariant pattern recognition using higher-order neural network," in *Proceedings of IEEE International Conference on Neural Network*, 1993, pp. 1273–1276.
- [40] H. Zhengquan and M. Y. Siyal, "Recognition of transformed english letters with modified high-order neural networks," *IEEE Electronics Letters*, vol. 34, pp. 2415–2416, 1998.
- [41] A. Calway, "Image analysis using a generalised wavelet transform," in *Proceedings of IEE Colloquium on Applications of Wavelet Transforms in Image Processing*, 1993, pp. 8/1–8/4.
- [42] Q. Chen, M. Defrise, and F. Deconinck, "Symmetric phase-only matched filtering of Fourier-Mellin Transforms of image registration and recognition," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 1156–1168, 1994.
- [43] F. C. C. De Castro, J. N. Amoral, and P. R. G. Franco, "Invariant pattern recognition of 2D images using neural networks and frequency-domain representation," in *Proceedings of IEEE International Conference on Neural Networks*, 1997, pp. 1644–1649.
- [44] N. Gotze, S. Drue, and G. Hartmann, "Invariant object recognition with discriminant features based on local Fast-Fourier Mellin Transform," in *Proceedings of IEEE International Conference on Pattern Recognition*, 2000, pp. 948–951.
- [45] S. Greenberg, H. Guterman, and S. R. Rotman, "Rotation and shift invariant image classifier using NN," in *Proceedings of IEEE Convention on Electrical and Electronics Engineers in Israel*, 1996, pp. 212–215.
- [46] P. R. Hill, C. N. Canagaraja, and D. R. Bull, "Rotationally invariant texture based features," in *Proceedings of IEEE International Conference on Image Processing*, 2001, pp. 141–144.
- [47] D. P. Mital, G. W. Leng, and S. K. Gupta, "An automatic rotation invariant recognition technique for colour objects and patterns," in *Proceedings of IEEE International Conference on Industrial Electronics, Control and Instrumentation*, 1993, pp. 1366–1370.

- [48] P. K. Patra, M. Nayak, S. K. Nayak, and N. K. Gobbak, "Probabilistic neural network for pattern classification," in *Proceedings of IEEE International Joint Conference on Neural Networks*, 2002, pp. 1200–1205.
- [49] S. P. Raman and U. B. Desai, "2-D object recognition using Fourier Mellin Transform and a MLP network," in *Proceedings of IEEE International Conference on Neural Networks*, 1995, pp. 2154–2156.
- [50] D. Sim, H. Kim, and D. Oh, "Translation, scale and rotation invariant texture descriptor for texture-based image retrieval," in *Proceedings of IEEE International Conference on Image Processing*, 2000, pp. 742–745.
- [51] E. Bigorgne, C. Achard, and J. Devars, "An invariant local vector for content-based image retrieval," in *Proceedings of IEEE International Conference on Pattern Recognition*, 2000, pp. 1019–1022.
- [52] H. Kim and K. Nam, "Object recognition of one-DOF tools by a back-propagation neural net," *IEEE Transactions on Neural Networks*, vol. 6, pp. 484–487, 1995.
- [53] S. S. Kumar and A. Guez, "A neural network approach to target recognition," in *Proceedings of IEEE International Joint Conference on Neural Networks*, 1989, p. 573.
- [54] S. Paschalakis and P. Lee, "Pattern recognition in grey level images using moment based invariant features," in *Proceedings of IEEE International Conference on Image Processing and Its Applications*, 1999, pp. 245–249.
- [55] K. K. Rao and R. Krishnan, "Shape feature extraction from object corners," in *Proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation*, 1994, pp. 160–165.
- [56] G. I. Salarna and A. L. Abbott, "Moment invariants and quantization effects," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1998, pp. 157–163.
- [57] L. Wang and G. Healey, "Illumination and geometry invariant recognition of texture in color images," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1996, pp. 419–424.
- [58] M. Zhenjiang and Y. Baozong, "A NN image understanding system for maps and animals recognition," in *Proceedings of IEEE Region 10 Conference on Computer, Communication, Control and Power Engineering*, 1993, pp. 902–905.
- [59] H. Ganster, A. Pinz, R. Rohrer, and E. Wildling, "Automated melanoma recognition," *IEEE Transactions on Medical Imaging*, vol. 20, pp. 233–239, 2001.
- [60] X. Tang, W. K. Stewart, and L. Vincent, "Automatic plankton image recognition," *Artificial Intelligence Review*, vol. 12, pp. 177–199, 1998.
- [61] A. Green, N. Martin, G. McKenzie, J. Pfitzner, F. Quintarelli, B. W. Thomas, M. O'Rourke, and N. Knight, "Computer image analysis of pigmented skin lesions," *Melanoma Res.*, vol. 1, pp. 231–236, 1991.
- [62] C. L. Giles and M. T., "Learning invariance, and generalization in high order neural networks," *Applied Optics*, vol. 26(23), pp. 4972–4972, 1987.
- [63] B. S. Reddy and B. N. Chatterji, "An FFT-based technique for translation, rotation and scale-invariant image registration," *IEEE Transactions on Image Processing*, vol. 5, pp. 1266–1271, 1996.

- [64] A. Baraldi and P. Blonda, "A survey of fuzzy clustering algorithms for pattern recognition—part ii," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 29, pp. 786–801, 1999.
- [65] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.
- [66] A. Chong, T. Gedeon, and L. Koczy, "A hybrid approach for solving the cluster validity problem," in *Proceedings of The 14th International Conference on Digital Signal Processing*, 2002.
- [67] S. Chuai-Aree, C. Lursinsap, P. Sophasathit, and S. Siripant, "Fuzzy c-mean: A statistical feature classification of text and image segmentation method," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 9, pp. 661–671, 2001.
- [68] T. Kohonen, "The self-organizing map," in *Proceedings of The IEEE*, vol. 78, 1990, pp. 1464–1480.
- [69] J. M. Barbalho, A. D. D. Neto, J. A. F. Costa, and M. L. A. Netto, "Hierarchical SOM applied to image compression," in *Proceedings of International Joint Conference on Neural Networks*, 2001, pp. 442–447.
- [70] L. B. Batista, H. M. Gomes, and R. F. Herbster, "Application of growing hierarchical self-organizing map in handwritten digit recognition," in *Symposium on Computer Graphics and Image Processing*, 2003.



APPENDICES

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Appendix I

Publications

- [I] K. Sookhanaphibarn, K.W. Wong and C. Lursinsap, "Application of Hierarchical Self-Organizing Mapping to Invariant Recognition of Color-Texture Images", in *Proceeding of the 9th International Conference on Neural Information Processing (ICONIP' 02)*, November 2002.
- [II] Kingkarn Sookhanaphibarn and Chidchanok Lursinsap, "Rotation and Scaling Invariant Self-Organizing Mapping", in *Proceeding of the 10th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2001)*, December 2001, Vol. 1, pp. 203-206.
- [III] K. Sookhanaphibarn and C. Lursinsap, "Feature Extraction with Color Texture-Sensitive, Rotational and Scaling Invariant Capability using Eigenvector-Guided Self-Organizing Mapping", in *Proceeding of the 8th International Conference on Neural Information Processing (ICONIP' 01)*, November 2001.
- [IV] Kingkarn Sookhanaphibarn and Chidchanok Lursinsap, "A New Feature Extractor Invariant to Intensity, Rotation and Scaling of Color Images", submitted in *IEEE Transactions on System, Man and Cybernetics Part B: Cybernetics*.

Appendix II

Mathematical Theory

Theorem B.1. *The time complexity of high-order neural network is $O(n^r)$, where r is the highest order.*

Proof. Let n be the number of inputs. The time and network complexities are equivalent to the number of hidden nodes in the network. Consequently, the proof below is shown the growing number of hidden nodes in the network following the order term of the high-order neural network. The number of hidden nodes is equal to all possible combinations of inputs where the combinations comprise ${}^n C_1$, i.e., n -choose-1, for the first term, ${}^n C_2$ for the second term, ${}^n C_3$ for the third term and so on. As shown in the above equations, the network and time complexities of the high-order neural network are $O(n) + O(n^2) + O(n^3) + \dots + O(n^r) = O(n^r)$. \square

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Appendix III

Test Data

The total of 34 different color images of Set A used in Chapter 4 are shown in Figure C.1-C.2. These images are download from the Internet. To show the invariant capability, each of 34 sets can produce 83 transformed images. For example, the original image m_{22} produces the transformed images as shown in Figure C.3-C.5.



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

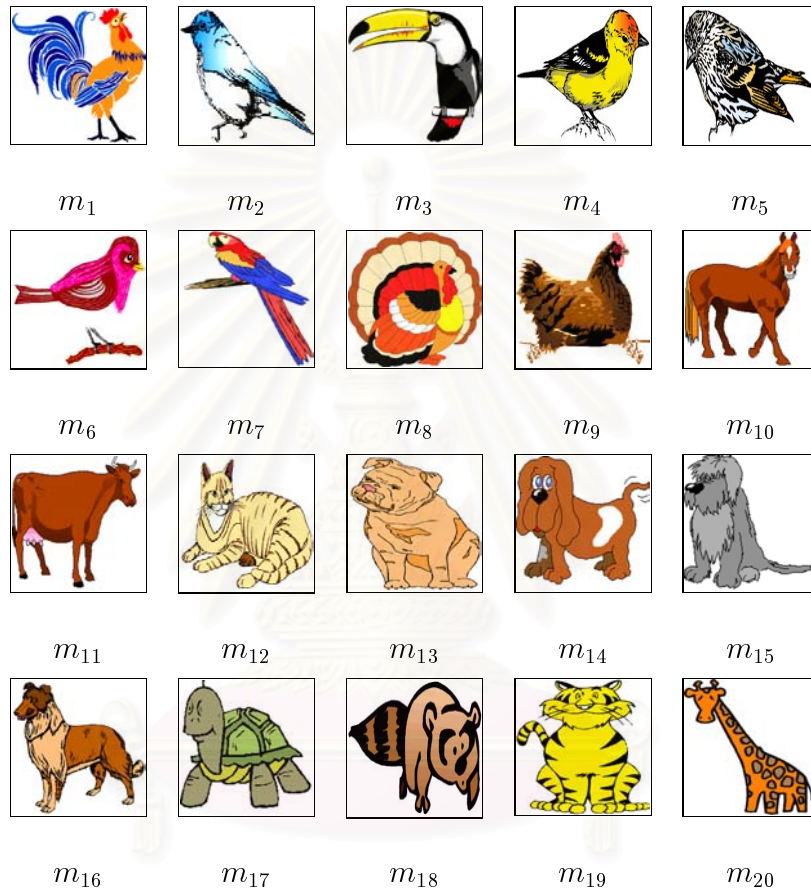


Figure C.1: Original images $m_i; 1 \leq i \leq 20$ used in the experiment. The subscript i denotes the assigned class $[i]$.

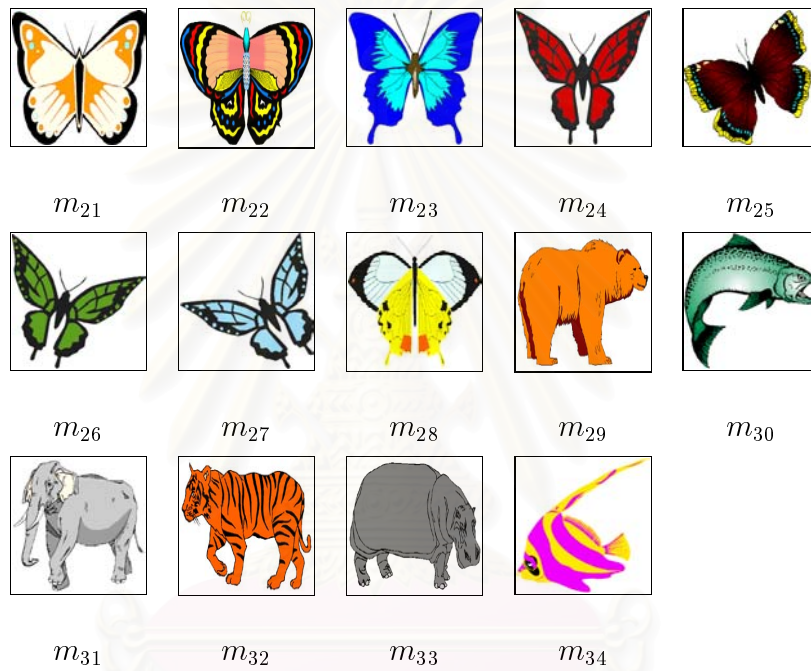


Figure C.2: Original images m_i ; $21 \leq i \leq 34$ used in the experiment. The subscript i denotes the assigned class $[i]$.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

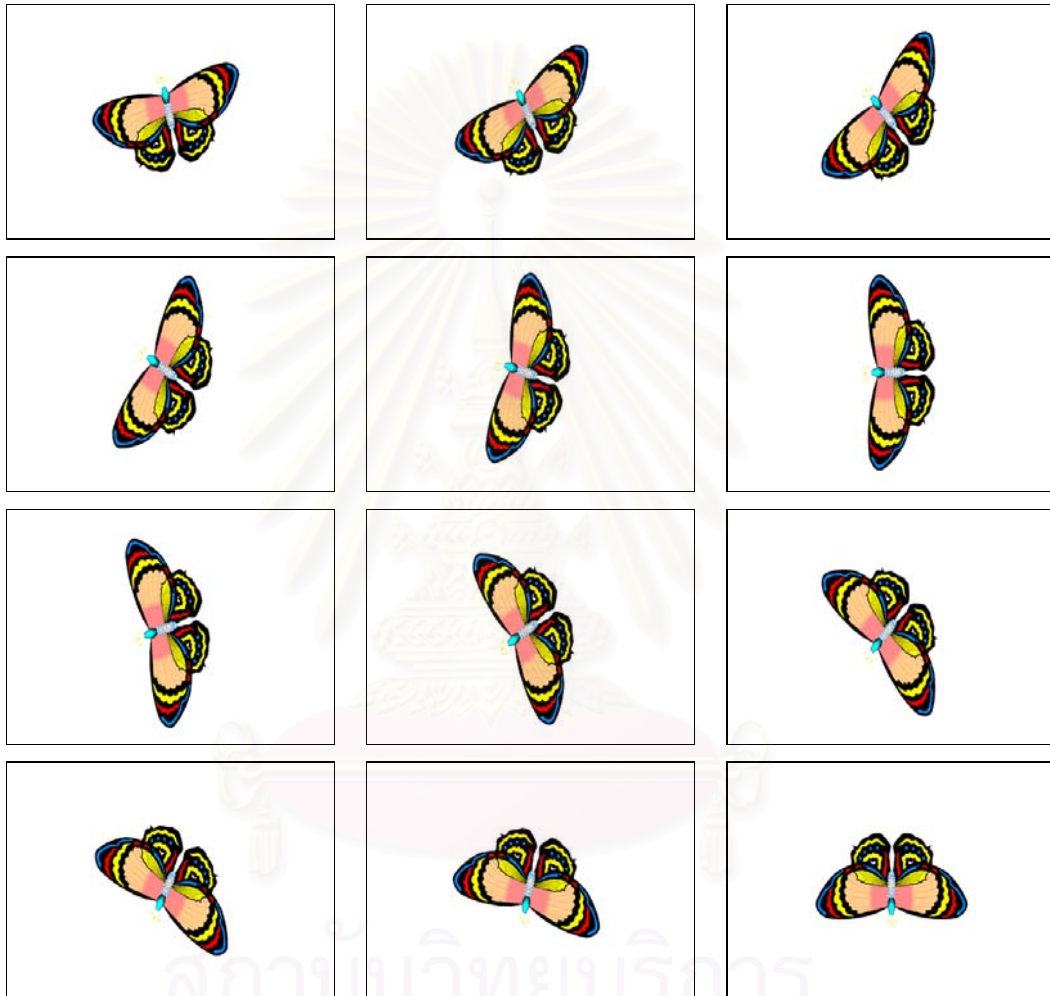


Figure C.3: Transformed images: Rotating the original image m_{22} through 15-180 degrees.

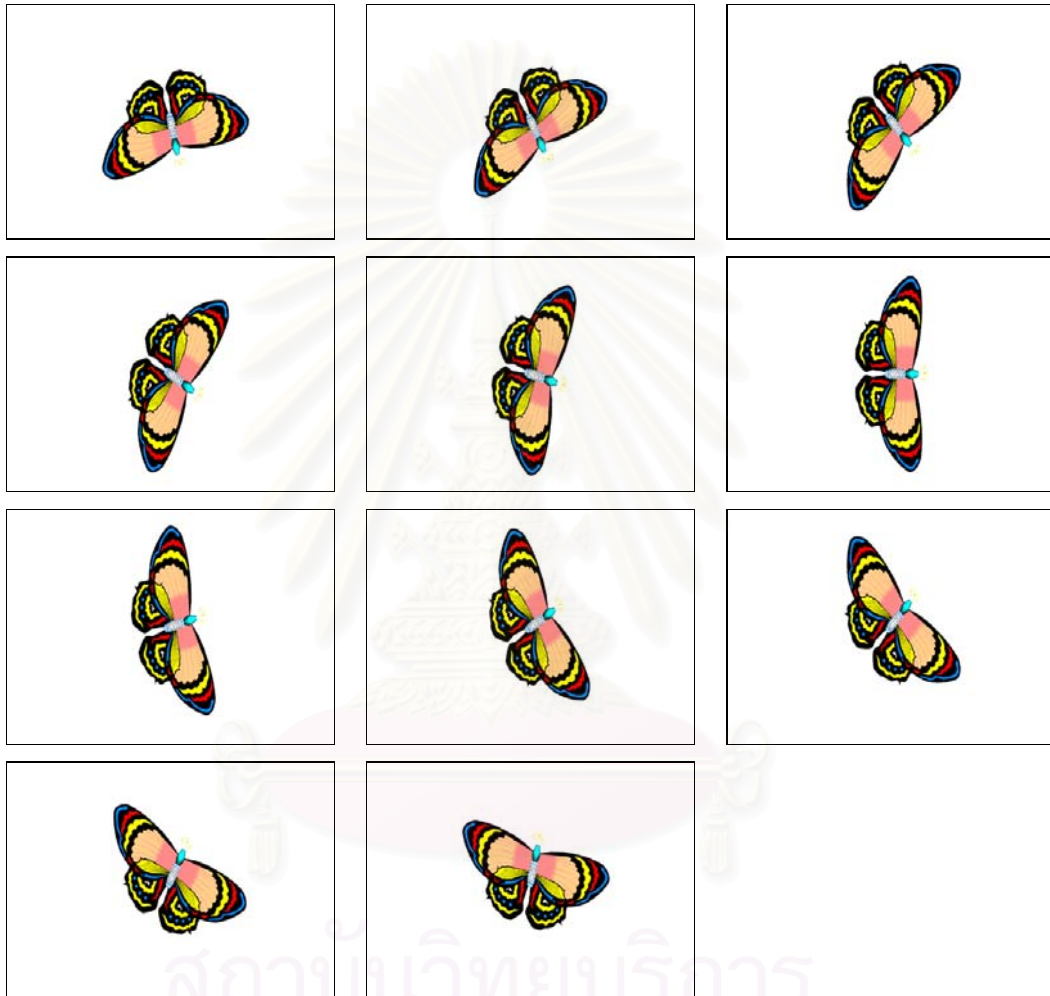


Figure C.4: Transformed images: Rotating the original image m_{22} through 195-345 degrees.

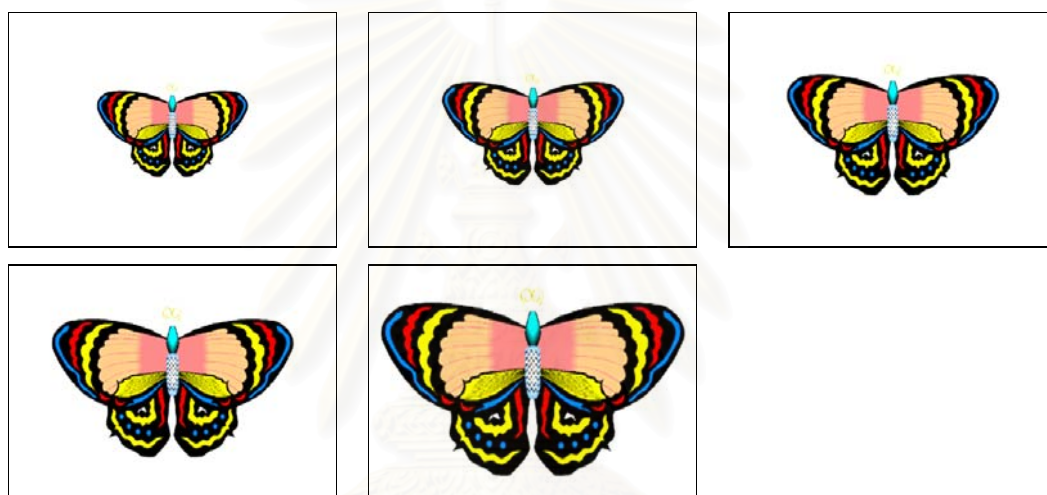


Figure C.5: Transformed images: Scaling the original image m_{22} up to two times as large as the original size.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Biography

Miss Kingkarn Sookhanaphibarn

PERSONAL DETAILS:

Date of Birth: April 29, 1978
Place of Birth: Bangkok, Thailand

EDUCATION:

Jun'99-May'04 Ph.D., Program in Computer Science, Department of Mathematics, Chulalongkorn University, Thailand.
Apr'02-Dec'02 Ph.D. Visiting student, School of Information Technology, Murdoch University, Western Australia.
Jun'97-May'99 M.Sc. Program in Computational Science, Department of Mathematics, Chulalongkorn University, Thailand.
Jun'93-Mar'97 B.Sc. Program in Computer Science, Department of Mathematics, Chulalongkorn University, Thailand.

HONOR AWARD:

1999 Distinguished Graduate, M.Sc. in Computational Science, Department of Mathematics, Chulalongkorn University, Thailand.

SCHOLARSHIPS:

Jun'03-Mar'04 Scholarship of Dhurakijpundit University for Ph.D. Program.
Jun'99-Dec'02 Scholarship of the Royal Golden Jubilee (RGJ) Ph.D. Program.
Jun'98-May'99 The Shell Centennial Education Fund for Thesis of MS.c. in Computational Science, titled *Application of Genetic Algorithm to High-Level Circuit Synthesis*.

PUBLICATIONS:

Jan'04 T. Raicharoen, K. Sookhanaphibarn, C. Lursinsap, F.C. Lin and V. Sayakanit, "*Emergency Flood Warning*", Proceedings of Workshop on Flood Prevention and Control on the Yangtze River.
July'99 K. Sookhanaphibarn and C. Lursinsap, "*Genetic Self-Recovery Micro-Rollback Synthesis*", in Proceedings of Congress on Evolutionary Computation, Vol. 2, pp. 1194-1199.
Dec'98 K. Sookhanaphibarn, T. Kitcharoensup, J. Wuttichaiworagul and C. Lursinsap, "*Recognition of Naked Bodies by Supervised Neural Network*", in Proceedings of the National Computer Science and Engineering Conference, Thailand.
Mar'98 K. Sookhanaphibarn and C. Lursinsap, "*Applying Genetic Algorithm to High-Level Synthesis*", in Proceedings of the Second Annual National Symposium on Computational Science and Engineering, Thailand, pp. 225 -232.