2003

Vision-guided multiple planar object recognition and tracking.

Shahed. Shahir
University of Windsor

Vision Guided Multiple Planar Object Recognition and Tracking

By

Shahed Shahir

A Thesis
Submitted to the Faculty of Graduate Studies and Research through the Department of Electrical and Computer Engineering in Partial Fulfillment of the Requirements for the Degree of Master of Applied Science at the University of Windsor

Windsor, Ontario, Canada
2003

Your file Votre référence
ISBN: 0-612-84577-X
Our file Notre référence
ISBN: 0-612-84577-X

ABSTRACT

In order to minimize the cost for any improvement or changes in manufacturing products, a flexible manufacturing system is necessary. To do so, robots should be empowered with equipment like vision sensors, which can work as human eyes for robots. A tracking vision sensor consists of a CCD camera, data acquisition, preprocessing, recognition and tracking modules. In this thesis, a design methodology for invariant recognition-based object tracking is proposed, which can potentially be used for implementation of either vision sensors or tracking vision sensors. The performance of Invariant Recognition-based object tracking to a large extent depends on the object recognition process rather than object sensing and tracking. For that reason, in this thesis the major focus is mostly on the recognition procedure.

Fuzzy Associative Database (FAD) and Adaptive Fuzzy Associative Database (AFAD) are introduced as supervised networks to overcome the weaknesses of some computational intelligence approaches like fuzzy inference and neural networks for object recognition purposes. Despite fuzzy inference, which requires skilled human to determine the membership functions, AFAD and FAD guarantee the optimal performance through the automatically defined Bank of Fuzzy Associative Memory Matrix (BFAMM), and also in spite of neural networks that need large number of trainees before starting recognition, FAD and AFAD demand a minimum number of trainees. As time passes, the possibility of misclassification declines in AFAD; as for humans, AFAD can recall the objects viewed frequently better than the objects seen once.

FAD consists of a Fuzzy Database (FD) and a Fuzzy Search Engine (FSE). FD holds the trained object information, like human memory, and FSE, like the human brain, processes incoming information based on information existing in the memory database. The FD includes two tables. The FSE uses table one to construct a BFAMM in order to conduct search over table two. In fact, FSE establishes a correspondence between an object and one of the trained classes in table two of the FD. AFAD includes Learning Vector Quantization (LVQ) in addition to FD and FSE in FAD. LVQ is employed in order to update the FD accordingly. Finally, explanation

regarding the real time implementation of invariant recognition-based object tracking is presented.

DEDICATED TO

My wife, who is my sole mate and the reason for my endeavor,

My Mother who has given me life and inspiration,

My Father who is my life instructor and hero,

My sister who let me know the meaning of being patient.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Dr. Xiang Chen, for his suggestions, guidance, and support throughout the course of this research work.

I wish to thank my department reader Dr. Behnam Shahrrava and my external reader Dr. Xiao Jun Chen for their valuable advice toward the fulfillment of the thesis work.

I would also like to thank Dr. W. C. Miller, Dr. M. Ahmadi and Dr. M. Sid-Ahmed for their support and their warm and generous help for all events related to my graduate study and research.

I am indebted to my fellow graduate students for their friendship and their support. I would like to credit Mr Jun Yang for his outstanding work on implementaion of 2D vision rack, and I have also had the pleasure of working with him.

My thanks go to Mr. Frank Cicchello and Mr. Don Tersigni, the department technicians for their support throughout my research. I am also indebted to Mr. Ron Dumouchelle for editing my thesis manuscripts.

The author would like to acknowledge the financial support provided by the Natural Sciences and Engineering Research Council (NSERC) and Manufacturing and Material Ontario (MMO) throughout this research work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOTATIONS

In this thesis, standard notation have been mostly used to be clear and to keep matters simple.

Capital *italic* letter is used for vectors and matrices: A, W.

If A is a matrix or a vector, then A^T is its Transpose.

If W is a matrix or a vector, then w_{ij} is the element of the matrix or vector in the ith row and the jth column.

X represents the universe of discourse.

If x is belongs to a fuzzy set A and A is the subset of X, $\mu_A(x)$ represents the general form for membership function of the fuzzy set A.

TN(.) is a function represents the T-norm operator in general.

SN(.) is a function represents the S-norm operator in general.

If B is a ordinary set in Boolean logic, the power set of the set B is presented by 2^B.

If B is a fuzzy set, the fuzzy power set of the set B is presented by $F(2^B)$.

If a and b are subsets of X, $max(a,b)$ determines the largest value in the input arguments.

If a and b are subsets of X, $min(a,b)$ determines the smallest value in the input arguments.

In particular, I have tried to define practical conventions and use them consistently.

Additional notation will be introduced as we go through the chapters.

CHAPTER 1. INTRODUCTION

1.1. Introduction

In the last several years, progresses in different disciplines namely, photonics, hardware design, software design, VLSI, DSP and artificial intelligence have opened new challenges for researchers and engineers. Researchers can push the envelope of the state of the art in every aspect of human life. Researchers and engineers have accesses to more precise and powerful tools both practically and theoretically. These tools enable researchers to come up with novel methodologies for realizing and implementing more efficient and faster manufacturing systems than before.

Owing to the increasing costs of raw materials, workmanship, energy, and the growing competitive environment, manufacturers are forced to produce higher quality products with lower cost. As a result, manufacturers have to go through fundamental changes in their production lines in order to have flexible manufacturing systems. The greatest costs in manufacturing are in plant construction, production line design, and erection of essential industrial equipments and appliances for the specific product. Manufacturing products have to be updated in order to serve its customers better. Usually, any changes in production costs money for manufacturers. In order to minimize the cost for any improvement or changes in products, a flexible manufacturing system is necessary and vital. Flexible manufacturing requires intelligent production processes, which can guarantee the highest quality and efficiency. To meet the flexible manufacturing requirements, intelligent robots have to be employed. To some extent, robotic researches have been able to move towards flexible manufacturing. However, there is still long way to go to reach the point that robot can work as human in manufacturing production lines. In order to move forward, we need to empower the robots with equipment like vision sensors, which can work as human eyes. The implementation of robotic devices with intelligent sensory capability has been recognized as an essential ingredient towards the success of flexible manufacturing systems.

Robotic researchers continuously seek to improve productivity in manufacturing and automation. A need exists for robots that can work on conveyor belts without knowing the exact position of target pieces. In order to make a robotic manipulator function intelligently, feedback from the environment through sensors is essential. Vision-based close loop feedback can enhance the performance of a robotic manipulator in many applications such as in product processing, material handling, spray painting, spot welding, quality inspection and automatic assembly, which are common tasks in manufacturing plants. In the last several years more and more efforts have been put into designing a tracking vision sensor in order to implement close loop feedback for the cases in which traditional sensors cannot be employed. For instance, in the automotive door handle assembly line, the foam barrier [12] is pasted onto the escutcheon in order to prevent the intrusion of dust and other unwanted material; yet, robots cannot be employed to conduct this task because robots are mostly preprogrammed for a fixed environment. Robots are good at working with solid materials, but not with flexible ones. The flexible foam barriers are deformed under the external forces. Any external force makes the foam barriers squeezed, stretched, or rotated. The deformation makes robots malfunction when robots are working with flexible materials. To fulfill the task, touchless sensors, such as laser scanners, proximity sensors, photo-diodes or vision sensor, should be employed. Vision sensors are more economical than laser scanners because the price of a camera is twenty five times less than the price of laser scanner. In addition to a video camera, Vision sensor consists of different modules such as, data acquisition, preprocessing, segmentation, recognition, and tracking to handle the visual sensing tasks. Each module has its own constraints and limitations, which requires time and effort for better performance. In other words, more work needs to be done to solve vision-based feedback control for manufacturing robots. The speed of object recognition module determines to a large extent vision sensor's response time. In fact, an object recognition method has the most impact on the performance of a vision sensor. There are a number of researchers who are working on theory and implementation of tracking vision sensor explicitly or implicitly.

1.2. Literature Survey

The application of vision based object recognition and tracking for manufacturing automation has received a lot of attention in the past decade [2,8,19,33,40,41]. From mathematicians to scientists-- from engineers to physicians, all benefit from the research conducted on artificial vision, namely, vision sensor. Control researchers are no exception; they have a long history to research on this area for robot control applications. The first application of tracking vision sensors to manufacturing automation goes back early 1980's. An experiment for a touchless sensor was conducted for the first time in 1983 by means of a fixed video camera. In "Camera-in-hand System" [5], a fixed overhead video camera is located above a conveyor separated from a robot arm because the robot arm could not support the huge video camera. In addition, the field of view of the vision system is obstructed when the robot manipulator is located between the object and video camera. To overcome the problem, a fiber-optic cable is employed to transfer the robotic arm view to the stationary video camera [7]. The first digital camera was built in the early 1990's. With the invention of lightweight CCD cameras, robot arms are now able to carry the camera. Mounting a CCD camera on the robot arm makes the system more flexible. Then, many researches have been conducted in the field of vision based object recognition and tracking. Some major results about the application of vision sensor on object recognition and tracking are discussed as follows:

- Vision based robot control [2,8, 41]: Touch-less sensor (CCD camera) has been used to track objects and control robot.

- Robot control for recognition, tracking and grasping [18]: the vision processing and object recognition portion of the system are given top priority because of the computing times required. Invariant object recognition is conducted by means of Bayesian Classifier. If the object exists in the trained, the robot arm tracks and picks the object by means of the fuzzy logic controller through the six proximity sensors mounted on the robot arm for a precise result.

- Robot control for tracking and grasping [32]: Object position is determined by means of ADALINE network. The network is trained for 200 object positions. Any changes on the system requires the extensive training.

- Object tracking by means of fuzzy approach [6,18]: Fuzzy Associative Memory, Fuzzy Inference is applied to track moving objects. An object is tracked by determining distance and speed of the moving object with respect to the tracker.

- Object recognition by means of Backpropagation algorithm[17,39]: This approach employed an approximate steepest descent algorithm in order to classify the objects. The algorithm can also be used for the objects that are not linearly separable. This approach needs many patterns for training before it can work properly.

- Shape recognition by means of Directional Flow-Change [12,13,40]: this approach is appropriate for shape recognition but not object recognition because the approach is too sensitive to rotation and noise.

- Face recognition by means of Radial Base Function (RBF) [22,23,32]: RBF has a fast learning speed, and it can be used for universal clustering. RBF is used mostly as a hidden layer of a neural network in order to improve the accuracy and speed up the neural network performance for recognition purpose. The output layer in Radian Basis Function Network can be perseptron, ADALINE or backpropagation network. Training for the outer layer is necessary. To have an appropriate response, large number of training sampling is required.

- Pattern recognition by means of Fuzzy Inference Network(FIN) and Fuzzy Neural network(FNN)[14,15,47,48]: Fuzzy Inference network is employed for pattern classification. FIN is introduced as a four layer feedforward network. The number of neurons in input layer equal to the number of the pixels in an image. On the second layer of FNN and FIN, low-pass filter is applied on the pattern pixels. Applying low pass filter creates large amount of burden proportionally when the size of image increases. Therefore, the presented pattern recognition is not efficient for real time object classification.

- Object recognition by means of Template Matching [12,13, 19, 30]: In this approach correlation is applied to the coming object image and every single template image stored in memory. The best match has the maximum correlation result. This approach is really slow and cannot recognize rotated object.

- Application of invariant feature extraction for pattern Recognition [1,3,4,10,28,31,46,51]: to avoid excessive training of neural network for pattern recognition purposes, invariant feature extraction is recommended. Invariant feature extraction techniques are introduced for pattern recognition, such as, Fourier descriptors, regional descriptors and moment invariants.

- Moment Invariant verses Fourier Descriptors [3,4,47]: two techniques, Moment Invariants and Fourier Descriptors, have been compared with each other. It is shown that moment invariants are sensitive to the distribution of the mass in the image, while Fourier descriptors are particularly sensitive to perturbations in the object boundary. Fourier descriptors for a disk object with a tiny wedge missing and a full object will be very different; on the other hand, moment invariants representation for these two objects will be very similar.

1.3. Thesis Objective

The purpose of this research is to study multiple planar object recognition through existing 2D platform (single CCD camera). The first challenge is to devise a system where misplaced planar objects can be recognized from an arbitrary viewpoint. The second challenge is to categorize objects in distinct classes.

Explicitly, the primary goal is to come up with a novel methodology through which multiple objects can be recognized and tracked. Finally, vision based system is implemented to test the accuracy and efficiency of the novel methodology for object recognition and tracking.

Meanwhile, the methodology for vision based object recognition can potentially lead us to the design of a vision sensor, and the combination of object recognition and tracking will potentially lead us to the design of vision-based feedback control systems. Vision-based feedback control can make the robots more flexible than before and revolutionize to the next generation of robots.

1.4. Thesis Overview

This thesis is organized to cover some desired aspects of an expert system design and implementation for planar object recognition and tracking from basic theories to advance topics. The thesis organization is as follows: after the Introduction, Chapter 2 provides some preliminary theories required for the other chapters. In Chapter 3, the Fuzzy Associative Database and Adaptive Fuzzy Associative Database are presented. Design methodology for object recognition and tracking will be covered in Chapter 4.

Implementation and experimental results for real time object recognition and tracking are proposed in Chapter 5. And finally, the conclusion and the future work can be found in Chapter 6.

CHAPTER 2. PRELIMINARY THEORIES

In this chapter, some important preliminary theories are presented. In Section 2.1 relation between fuzziness and probability is briefly presented. In Section 2.2, the concept of fuzzy System, which is crucial for our discussion throughout this thesis, is presented. In Section 2.3, clustering theory is looked over briefly, and in the last section a Fuzzy Inference System is illustrated. In section 2.4, the fuzzy inference Network is proposed.

2.1. Fuzziness versus Probability

Randomness and fuzziness differ conceptually and theoretically [6]. Randomness and fuzziness also share many similarities. Both systems describe uncertainty with numbers in the unit interval [0, 1]. This ultimately means that both system describe uncertainty numerically. Both systems combine sets and propositions associatively, commutatively, and distributively. The key distinction concerns how the systems jointly treat a set A and its opposite A^c. Classical set theory demands

$A \cap A^c = \Phi$,

and probability theory conforms

$P(A \cap A^c) = P(\Phi) = 0$

So $A \cap A^c$ represents a probabilistically impossible event. But fuzziness begins when $A \cap A^c \neq \Phi$

2.2. Fuzzy System

Fuzziness is an alternative to randomness for describing uncertainty. A fuzzy set is defined as a point in unit hypercube I^n=[0,1]^n [6]. The distance between points with in the cube leads to measure the size and fuzziness of a fuzzy set. In other words, this is helpful to see how much one fuzzy set is a subset of another fuzzy set. Mapping between fuzzy cubes is another aspect, which led us to a definition for a fuzzy system. Fuzzy mapping provides a surprising and fruitful alternative to the prepositional and predicte-calculus reasoning techniques used in artificial-intelligence (AI) expert systems. It allows us to reason with sets instead of propositions. The fuzzy set framework is numerical and multidimensional. Fuzzy approach translates the structured knowledge into a flexible numerical framework and processes it in a manner that resembles neural network processing. The numerical framework also allows us to adaptively infer and modify fuzzy systems, perhaps with neural or statistical techniques directly from problem-domain sample data.

Between-cube theory is fuzzy-systems theory. A fuzzy set defines a point in cube. A fuzzy system defines a mapping between cubes.

Definition 2.1: *Fuzzy System*

A fuzzy system maps fuzzy sets to fuzzy sets. A fuzzy system, F is a transformation F: $I^n \rightarrow I^p$. The *n*-dimensional unit hypercube I^n houses all the fuzzy subsets of the domain space, or input universe of discourse, $X = \{x_1, x_2, ..., x_n\}$. I^p houses all the fuzzy subsets of the range space , or output universe of discourse, $Y = \{y_1, y_2, ..., y_p\}$. X and Y can also denote subsets of R^n and R^p.

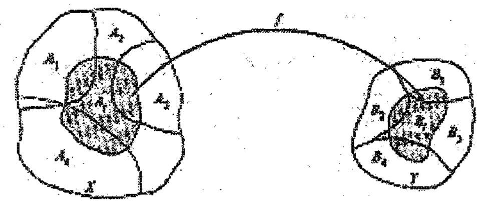


Figure 2.1: Fuzzy mapping

Figure 2.1 shows a fuzzy system. Here, F maps the input domain X to the output domain Y. In Figure 2.2, the other method for depicting a universe of discourse $X = \{A_1, A_2, A_3\}$ is shown.

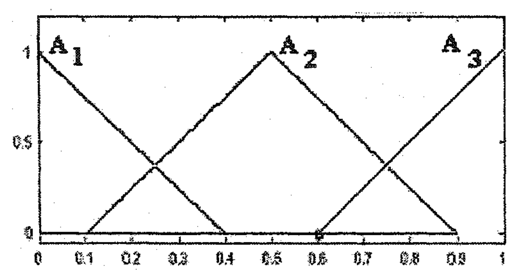


Figure 2.2: Universe of discourse X

2.2.1. Formulation of Fuzzy Membership Function

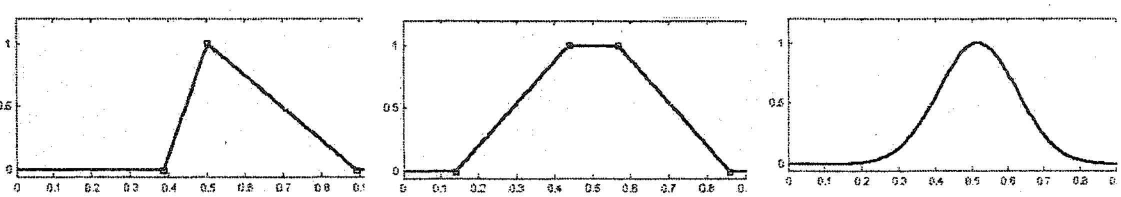
Zadeh in[29] has interpreted fuzzy sets as generalized indicator or membership functions(MF), mapping from domain X={x_1, ..., x_n} to range [0,1]. A fuzzy set is completely characterized by its Membership Function. Since most fuzzy sets have a universe of discourse X consisting of the real line R, it would be impractical to list all the pairs defining a membership function. A more convenient and concise way to define a membership function is to express it as a mathematical formula. Membership function can also be denoted as two-dimensional graphs $\mu_A(x)$ [49], with the domain X presented as one-dimensional axis and another axis is the range [0,1] in general. Let us define several classes of parameterized membership functions.

Definition 2.2: *Triangular Membership Function*

A triangular membership function is defined as follows

$$\mu_{\text{triangle}}(x;\ a,\ b,\ c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right), \tag{2.1}$$

where x is a self variable ($x \in X$), and a, b, c are three real parameters(with a<b<c) which determine the x coordinate of the three corners of the triangular membership function depicted in Figure 2.3(a).



(a) (b) (c)

Figure 2.3: Triangular, trapezoidal, and gaussian membership functions

Definition 2.3: Trapezoidal Membership Function

A trapezoidal membership function is defined as follows

$$\mu_{\text{trapezoidal}}(x;\ a,\ b,\ c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{d-x}{d-c}\right), 0\right), \tag{2.2}$$

where x is a self variable ($x \in X$), and a, b, c and d are four real parameters(with a<b<c<d) which determine the x coordinate of the four corners of the trapezoidal membership function depicted in Figure 2.3(b).

Definition 2.4: *Gaussian Membership Function*

A Gaussian membership function is specified by

$$\mu_{gaussian}(x; c, \sigma) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2}, \qquad (2.3)$$

where x is a self variable ($x \in X$), and c and σ are two real parameters for Gaussian membership function; c presents the membership function center and σ determines the width of the membership function depicted in Figure 2.3(c).

2.2.2. Fuzzy Intersection

The intersection of two fuzzy sets A and B is specified in general by a function TN:[0,1]x[0,1]=>[0,1], which aggregates two memberships as follows:

$$\mu_{A \cap B}(x) = TN(\mu_A(x), \mu_B(x)), \qquad (2.4)$$

where TN function is a **T-norm**[41]. This class of fuzzy intersection operator, which is usually referred to as T-norm operator, meets the following basic requirements.

Definition 2.5: *T-norm*

T-norm operator is a function with two input arguments which satisfies:

TN (0,0)=0, TN(a,1)=TN(1,a)=a (boundary)

$$
\begin{array}{lll}
TN(a, b) \le TN(a, d) \text{ if } a \le c \text{ and } b \le d & \text{(monotonicity)} & \\
TN(a, b) = TN(b, a) & \text{(commutativity)} & (2.5) \\
TN(a, TN(b, c)) = TN(TN(a, b), c) & \text{(associativity)} &
\end{array}
$$

where a, b and c are real numbers between interval zero and one. Three of the most frequently used T-norm operators are as follows:

$$
\begin{array}{lll}
\text{Minimum:} & TN_{min}(a, b)=\min(a, b)= a \cap b & \\
\text{Algebraic Product:} & TN_{ap}(a, b) = a*b & (2.6) \\
\text{Bounded product:} & TN_{bp}(a, b)=0 \cup (a+ b-1) &
\end{array}
$$

2.2.3. Fuzzy Union

The fuzzy union operator is specified in general by a function SN:[0,1]x[0,1]=>[0,1]:

$$\mu_{A \cup B}(x) = SN(\mu_A(x), \mu_B(x)), \tag{2.7}$$

This class of fuzzy union operators, which are often referred to as **S-norm** operator, satisfies the following basic requirements.

Definition 2.6: *S-norm*

A S-norm operator [6,26]

$$
\begin{array}{lll}
SN(1,1)=1,\ SN(a,0)=SN(0,a)=a & \text{(boundary)} & \\
SN(a, b) \le SN(a, d) \text{ if } a \le c \text{ and } b \le d & \text{(monotonicity)} & \\
SN(a, b) = SN(b, a) & \text{(commutativity)} & (2.8) \\
SN(a, SN(b, c)) = SN(SN(a, b), c) & \text{(associativity)} &
\end{array}
$$

where a, b and c are real numbers between interval zero and one.. Three of the most frequently used **S-norm** operators are as follows:

Maximum:	$SN_{max}(a, b)=max(a, b)= a \cup b$	
Algebraic sum:	$SN_{as}(a, b) = a+b$	(2.9)
Bounded sum:	$SN_{bs}(a, b)=1 \cap (a+ b)$	

2.2.4. Geometry of Fuzzy Sets: Sets and Points

In fuzzy set theory, the intersection operator is conducted by pairwise minimum (picking the smaller of the two elements); the union operator is conducted by pairwise maximum (picking the bigger of the two elements), and complementation is conducted by order reversal:

$$\mu_{A\cap B} = \min(\mu_A, \mu_B)$$
$$\mu_{A\cup B} = \max(\mu_A, \mu_B) \qquad (2.10)$$
$$\mu_{A^c} = 1 - \mu_A$$

Example 2.1: Two set A and B are defined as following:

$A = [\quad 1 \quad .8 \quad .4 \quad .5 \quad]$

$B = [\quad .9 \quad .4 \quad 0 \quad .7 \quad]$

if minimum and maximum applied as T-norm and S-norm respectively , intersection and union of A and B are as follows:

$A \cap B = [\quad .9 \quad .4 \quad 0 \quad .5 \quad]$

$A \cup B = [\quad 1 \quad .8 \quad .4 \quad .7 \quad]$

A complement is

$A^C = [\quad 0 \quad .2 \quad .6 \quad .5 \quad]$

Intersection and union of A and its complement:

$A \cap A^C = [\quad 0 \quad .2 \quad .4 \quad .5 \quad]$

$A \cup A^C = [\quad 1 \quad .8 \quad .6 \quad .5 \quad]$

Against Boolean logic, the overlap fit vector $A \cap A^C$ in this example does not yield the empty set, {}, and the under lap fit vector $A \cup A^C$ does not equal to the vector of all ones, the universe, as depicted in Table 2.1.

	Boolean	Fuzzy
$A \cup A^C$	X(universe)	$D\ (D \neq X)$
$A \cap A^C$	{}	$E\ (E \neq \{\})$

Table 2.1: Boolean verses fuzzy

Generally speaking, all the fuzzy subsets look like a cube; on the other hand, a fuzzy set looks like a point inside a cube. Before expanding the concept, let us look at an example in an ordinary set in a nonfuzzy domain. Consider the set of two elements, $X=\{x_1, x2\}$. The nonfuzzy power set 2^X contains four sets:

$$2^X = \{ \{\}, \{x_1\}, \{x_2\}, \{x_1,x_2\}\} \tag{2.11}$$

These four sets correspond respectively to the four bit vectors [0 0], [1 0], [0 1], and [1 1], shown in Figure 2.4. The 1s and 0s indicate the presence or absence of the ith element x_i in the subset.

Example 2.2: Now consider the fuzzy subsets of *X*. We can view the fuzzy subset *A*=(1/3,3/4) corresponding to the ordinary subset [0 1] in a nonfuzzy domain. The set *A* is illustrated as a point in Figure 2.4.

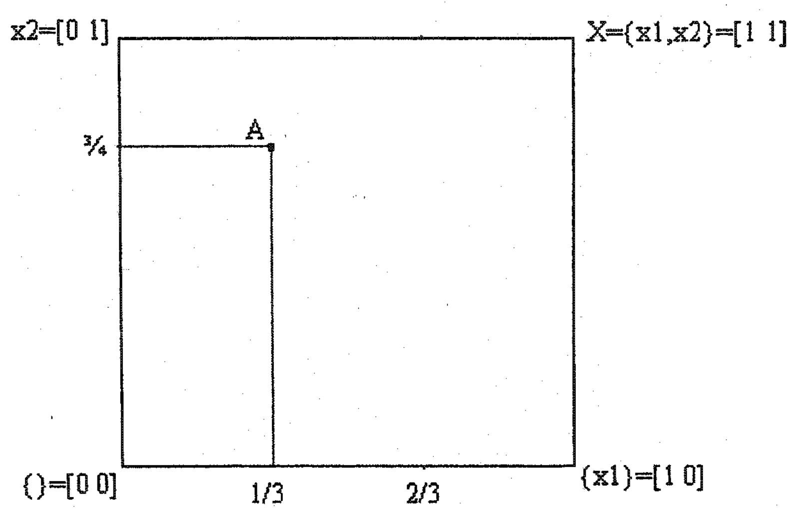


Figure 2.4: Fuzzy set A as a point(1/3 ,¾)

Consider again the two-dimensional fuzzy set *A* defined by the vector [1/3 ¾].

$$A = [\ 1/3 \quad 3/4\ ]$$
$$A^C = [\ 2/3 \quad 1/4\ ]$$
$$A \cup A^C = [\ 2/3 \quad 3/4\ ]$$
$$A \cap A^C = [\ 1/3 \quad 1/4\ ]$$

The sets contribute to form a unit square. Each of the mentioned points (sets) forms a vertex of the unit square, Figure 2.5.

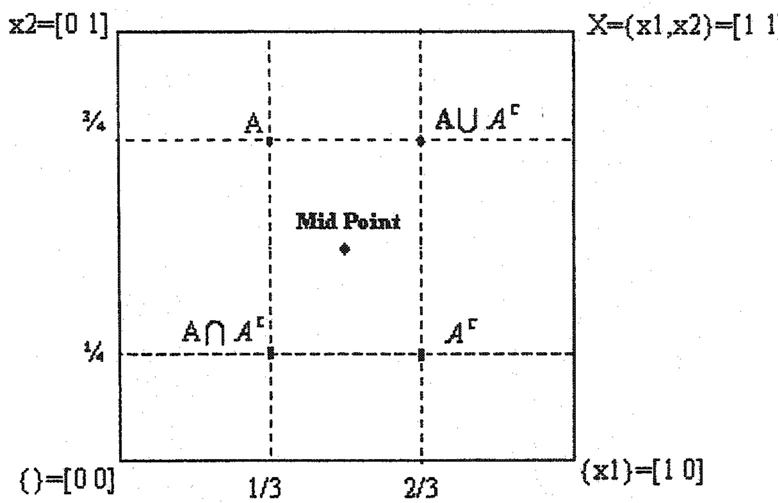


Figure 2.5: Fuzzy set A, A^c, $A \bigcap A^c$ and $A \bigcup A^c$

In Figure 2.5 the four fuzzy sets illustrate the subsets of the fuzzy set A (the sets A, A^C, $A \bigcup A^C$, and $A \bigcap A^C$). If set A moves toward the [0 1], the Boolean corners of the cube, it becomes less fuzzy; in other words, set A approaches an ordinary Boolean sets. Meanwhile, if set A approaches toward the mid point, the set becomes fuzzier.

2.2.5. Distance between Fuzzy Sets

Distance between two fuzzy sets is the norm of their difference vector. In general, "P Norm" is define as follow:

$$\left\|A,B\right\|^P = \sqrt[P]{(x_a - x_b)^P + \cdots + (y_a - y_b)^P} \tag{2.12}$$

which $1 \leq P < \infty$. In Euclidian space (p=2), distance between two sets A and B is defined by the following formula, Figure 2.6.

$$\|A,B\|^2 = \sqrt{(x_a - x_b)^2 + \cdots + (y_a - y_b)^2} \tag{2.13}$$

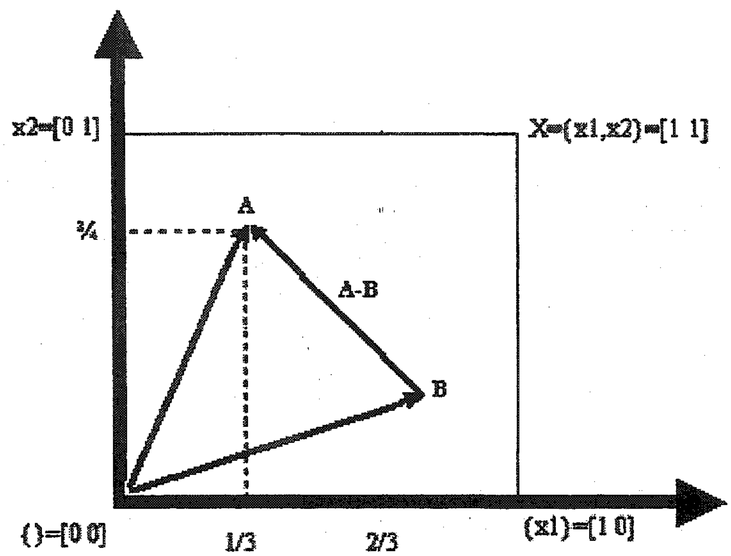


Figure 2.6: Distance between fuzzy sets *A* and B

If *B* is located at origin(*B*=[0 0]), then the distance between *A* and *B* represents the length of sets *A*.

$$\|A\|^2 = \sqrt{(x_a)^2 + (y_a)^2} \tag{2.14}$$

The distance defined for p=1 is called the Fuzzy Hamming Distance.

$$\|A\| = M(A) = |x_a| + |y_a| \tag{2.15}$$

Count of *A*, which is called the Fuzzy Hamming distance of set *A* from the origin, in *n*-dimensional domain is defined as follows:

$$M(A) = \sum_{i}^{n} \mu_A(x_i) \tag{2.16}$$

Example 2.3: if A is defined as follows

A =[1/3 ¾]

Count of A is

$M(A)$ = 13/12

In this example 13/12 depicts the Fuzzy Hamming distance of set A (see Figure 2.7).

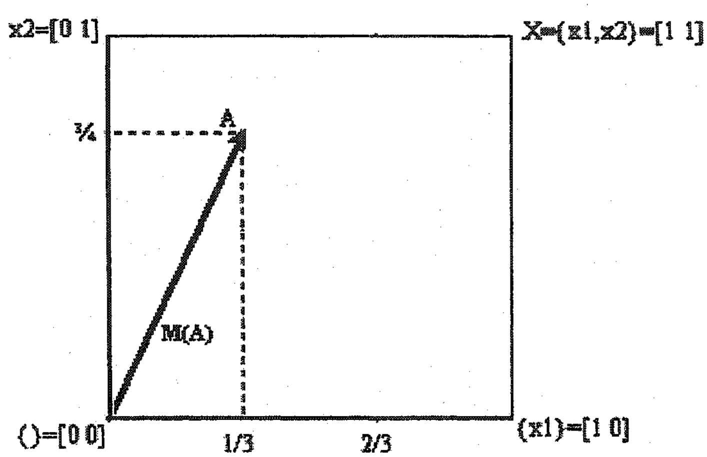


Figure 2.7: The Fuzzy Hamming distance of set A from the origin

2.2.6. Fuzzy Entropy Theorem

Fuzzy Entropy is defined as follow:

$$E(A) = \frac{M(A \cap A^C)}{M(A \cup A^C)} \tag{2.17}$$

If the fuzzy set A approaches the Boolean corner of the cube, at the same time, $A \cap A^C$ moves to the origin and $M(A \cap A^C)$ approaches the zero. While $M(A \cap A^C)$ approaches the zero, $E(A)$ also approaches the zero ($E(A)$=0). And vice versa, while the fuzzy set A move to the mid point, set A superimpose on its compliment A^C. At the same time, $A \cap A^C$ and $A \cup A^C$ will be superimposed on the top of each other. In this case $M(A \cap A^C)$ equals with $M(A \cup A^C)$. As a result $E(A)$ equals one($E(A)$=1).

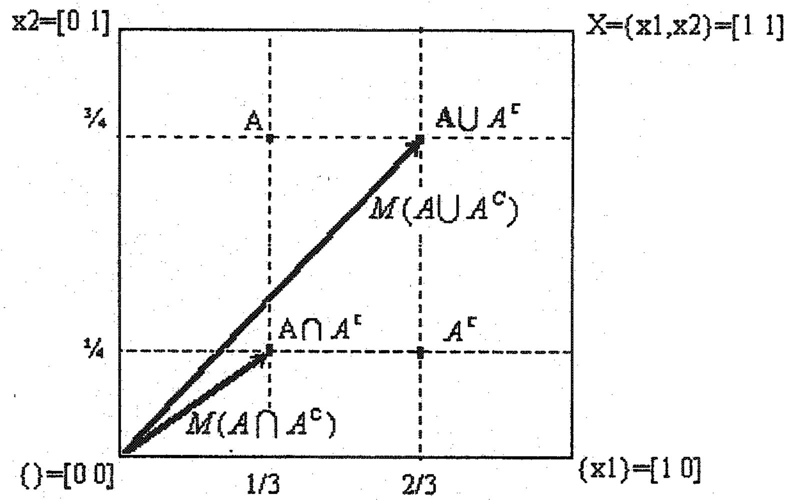


Figure 2.8: Count of $A \cup A^c$ and $A \cap A^c$

2.2.7. The Subset Hood Theorem

Sets contain subsets. In Boolean logic, A is a subset of B if and only if every element in A is an element of B. The power set, 2^B contains all of B's subsets. A is a subset of B if and only if A belongs to B's power set:

$$A \subset B \text{ if and only if } A \in 2^B \qquad (2.18)$$

Definition 2.7 : *dominated membership function relationship*

Here, a set A is a subset of fuzzy set B if and only if there is no element x that belongs to A but not to B, $\mu_A(x)$ =1 but $\mu_B(x)$=0, or the dominated membership can be express through the following formula:

$$A \subset B \text{ if and only if } \mu_A(x) \le \mu_B(x) \text{ for all } x \mid x \in X \tag{2.19}$$

Dominated membership function relationship can be also applied for the absolute fuzzy subset recognition. For example, if A=[0.3 0.7] and B=[0.4 0.9], A is a fuzzy subset of B ($A \subset B$), but B is not a subset of A. The preceding conclusion is a crisp conclusion based on Boolean subset definition, but it is not a fuzzy based conclusion because A is a subset of B; otherwise, it is not. It is crisp conclusion. The dominated membership function implies that $F(2^B)$ defines the hyper rectangle limited between the origin and $\mu_B(x)$. Figure 2.9 shows the fuzzy power set of the set B= (1/3 3/4). $F(2^B)$ has infinite subsets if it is not empty. In other words, as long as B is not located on origin, it has infinite subsets.

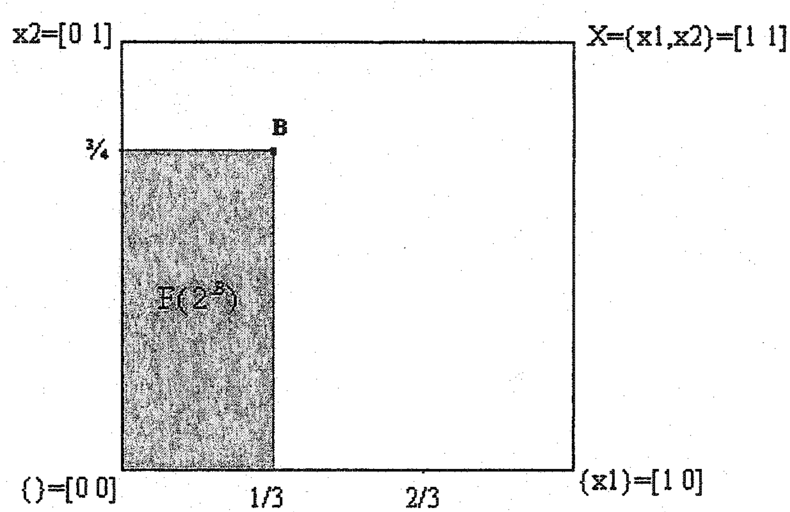


Figure 2.9: Fuzzy set B and its subsets

The size of $F(2^B)$ can be measured through volume, which is defined as follows.

Definition 2.8 : *Volume of a fuzzy set*

Volume of a fuzzy set with a finite-dimension is the product of the vector elements. Volume of the fuzzy set B is obtained through the following formula:

$$V(B)=\prod_{i=1}^{n} \mu_B(x_i) \qquad (2.20)$$

n is space dimension , in which B is defined.

Example 2.4: Here, B is defined as follows

B= [1/3 3/4],

So volume of the fuzzy set B in Euclidian space is

V(B)=3/12

The abstract fuzzy membership function can be any number in range [0,1]. The number is defined the degrees of subset hood. Consider again the dominated membership function relationship:

$$A \subset B \text{ if and only if } \mu_A(x) \le \mu_B(x) \text{ for all } x \mid x \in X \qquad (2.21)$$

Suppose element x_v violates the dominated membership function relationship: $\mu_A(x_v) > \mu_B(x_v)$. Then A is not a subset of B, at least not totally. Violation factor is defined below.

Definition 2.9 : *Violation factor*

Fuzzy set A is violated the dominated membership rule if A includes element x_v which violated the dominated membership; while $A \subset B$, violation is defined as the summation of distance between the violated points and fuzzy set B. Violation factor is determined by the following formula.

$$\text{Violation}=\sum_{x\in X}\max(0,\mu_A(x)-\mu_B(x)) \quad (2.22)$$

Definition 2.10 : *Superset hood factor*

Superset hood factor is the normalized violation factor, which show the percentage of the A's elements are not inside the B power set 2^B. Superset hood factor is defined as follows:

$$\text{Supersethood}(A, B)=\left(\sum_{x\in X}\max(0,\mu_A(x)-\mu_B(x))\right) / M(A) \quad (2.23)$$

In fact, a superset hood is the normalized Violation factor because $M(A)$ is the length of vector A. On the other hand, subset hood factor can be defined in the following.

Definition 2.11 : *Subset hood factor*

Subset hood factor is shown the percentage of the A's elements are inside the B power set, 2^B. In fact, subset hood factor is a compliment of superset hood factor, and determined by S function as shown in the following equations:

$$S(A,B)=1-\text{Supersethood}(A, B), \quad (2.24)$$

or

$$S(A,B)= 1 - \left(\sum_{x\in X}\max(0,\mu_A(x)-\mu_B(x))\right) / M(A) \quad (2.25)$$

For illustration, X may consist of one hundred values:

$$X=\{x_1,x_2,\ldots, x_{100}\} \quad (2.26)$$

The violation might occur in with the first element, namely, x_1, and other fifty elements are common between *C*, and *D*.

$$C=\{ x_1,x_2,\ldots, x_{50}\} \tag{2.27}$$

and

$$D=\{ x_2,x_3,x_{50} \} \tag{2.28}$$

Correspondingly, *D* is largely a subset of *C*. As a result, subset hood factor can be determined by having the violation factor.

Example 2.5: Consider a bit vectors *A*=[0.2 0 0.4 0.5] and *B*=[0.7 0.6 0.3 0.7]. Neither set is a proper subset of the other. *A* is almost a subset of *B* but not quite, since

$\mu_A(x_3)-\ \mu_B\ (x_3)=0.4-0.3=0.1>0$

then

Violation =0.1

And

$M(A)=0.2+0+0.4+0.5=1.1$

then

$S(A,B)=1-(0.1/1.1)=10/11=0.90$

Subset hood factor can be calculated $S(B,A)$:

$M(B)=0.7+0.6+0.3+0.7=2.3$

Violation=max(0,.7-.2)+ max(0,.6-0)+ max(0,.3-.4)+ max(0,.7-.5)

Violation==0.5+0.6+0.2=1.3

$S(B,A)=1.3/2.3=0.56$

As presented, A is 90 % subset of B ; however, B is 56% subset of A. So from the above example, the meaning of fuzzy subset can be conceived.

2.3. Clustering Analysis

The problem of unsupervised classification essentially reduces to partitioning the data in the feature space into clusters or sub groups. In clustering, the goal is to find subclasses. Clustering procedures describe data in terms of groups of data points that possess strong internal similarities. A cluster is described as connected regions of multidimensional space containing a high density of points, separated from other such regions by a region containing a relatively low density of points. A cluster can be defined as a set of entities that are alike, and entities from different clusters are not alike. Clustering technique is a formal study of algorithms and method for grouping or classifying objects. The problem of clustering is to find natural groupings or similarities among data samples, which is a complicated task. In order to group samples, it is necessary to define a similarity measure. One might suppose that two samples belong to the same cluster if the Euclidian distance between them is less than some threshold d_0. If d_0 is very large, all samples will be assigned to a single cluster. If d_0 is very small, each sample will form an isolated cluster. Figure 2.11 shows clusters of points in a two-dimensional space. Here, at a global level, there exist three clusters.

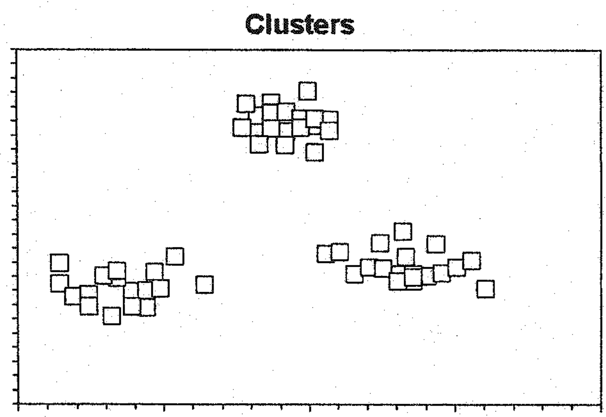


Figure 2.11: Feature space with three clusters

If N is the number of samples, and each sample characterizes by an n- dimensional vector. Each sample is to be placed into one of m cluster $(w_1, w_2, ..., w_m)$, where m may or may not be known. Let w_k denote the cluster to which sample i is assigned. Let the classification W be a vector made up of w_k and configuration P be a matrix made up p_i:

$$W=[w_1, w_2, ..., w_m]^T \tag{2.31}$$

$$P=[p_1^T, p_2^T, ..., p_n^T]^T \tag{2.32}$$

In general, the clustering is a function of W and P, and can be written as

$$J= J\,(W, P) \tag{2.33}$$

All the iterative algorithms, including Statistical approaches like K-means and Isodata, use some similarity measure. The most commonly used is the Euclidean distance in the feature space. Similarity measures such as the normalized correlation, and Hamming distance are mostly used. In all these algorithms, the input sample is assigned to a cluster on the basis that it is closer to samples in that cluster than the samples in other clusters. In classical clustering algorithms, a sample is assigned to one and only one cluster. However in practice, it is desirable to allow partial memberships, so that a sample can be assigned to more than one class, with the degree of belief specified to each class to which sample belongs. Fuzzy Associative network enables us to form partial clustering. In the section 3.1, Fuzzy Associative network will be introduced in order to implement partial clustering. Now let us open discussion regarding Fuzzy neuron and its construction, which lead us to a new definition for Fuzzy Associative network.

2.4. Fuzzy Inference Network

The architecture and learning rules of adaptive networks have been described in [15,16,24,26]. Structurally, the only limitation of network configuration is that it should be of feedforward type. In this section, Adaptive Network Fuzzy Inference System (ANFIS) is going to be proposed.

2.4.1. ANFIS Architecture

Any fuzzy " if-then" rules can be simulated by ANFIS. We assume the fuzzy inference system under consideration has two inputs x and y and one output f. Suppose that the rule base contains two fuzzy "if-then" rules of Takagei and Sugeno's type [47,48].

Rule one: if x is A_1 and y is B_1, then $f_1 = p_1 x + q_1 y + r_1$

Rule one: if x is A_2 and y is B_2, then $f_2 = p_2 x + q_2 y + r_1$

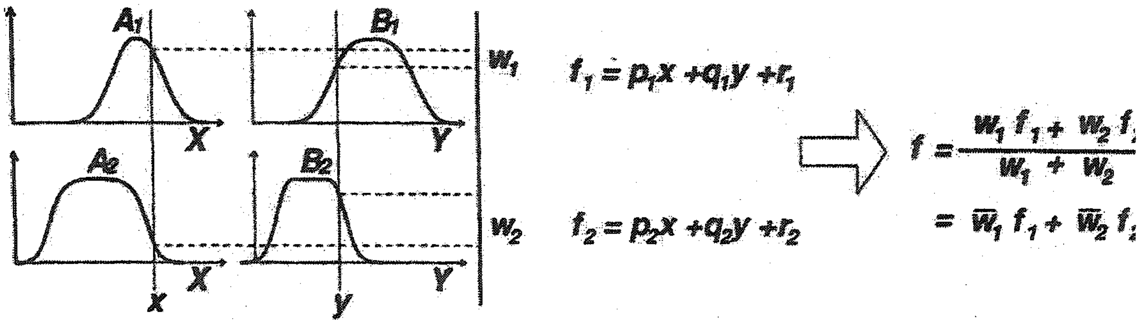


Figure 2.12: The Sugeno fuzzy model

Then the fuzzy reasoning is illustrated in Figure 2.12, and the equivalent ANFIS architecture is shown in Figure 2.13. The node functions are the same as described below.

Definition 2.12 : First Layer of ANFIS

Every node i in the first layer is a node function as follows:

$$O_i^1 = \mu_{A_i}(x) \qquad i=1,2. \tag{2.34}$$

Where x is the input to the node i, $x \in X$ and A_i is a membership function associated with this node function. In other words, O_i^1 specifies the degree to which the given x satisfies the quantifier A_i. $\mu_{A_i}(x)$ can be any membership functions defined in Section 2.2.1.

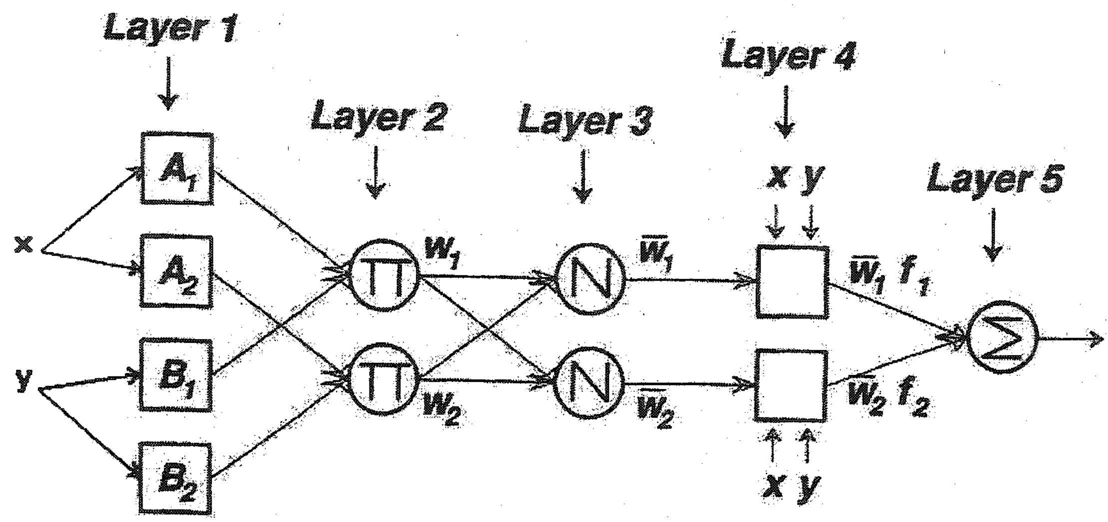


Figure 2.13: Sugeno fuzzy inference network

Definition 2.13: *Second Layer of ANFIS*

Every node in second layer aggregates by applying T-norm operator on the incoming signals.

$$w_i = TN(\mu_{A_i}(x), \mu_{B_i}(x)) \quad i=1,2. \tag{2.35}$$

w_i is the firing weights for the ith node in the second layer. Each node output represents the firing strength of a rule.

Definition 2.14: *Third Layer of ANFIS*

Every node in this layer is a circle node labeled *N*. the *i*th node calculate the normalized firing strengths for the *i*th note, as defined in the following formula.

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i=1,2. \tag{2.36}$$

Definition 2.15: *Forth Layer of ANFIS*

Every node ***i*** in this layer is a square node with a node function

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i) \quad i=1,2. \tag{2.37}$$

Where w_i is the third layer output, and $\{p_i, q_i, r_i\}$ are the set parameters.

Definition 2.16: *Fifth Layer of ANFIS*

The single node in this layer computes the overall output as the summation of all the incoming signals.

$$O_i^5 = \sum_{i=1}^{2} \overline{w}_i f_i \tag{2.38}$$

2.4.2. Hybrid Learning Algorithm

In Fuzzy Inference approach, membership functions are linguistic, so the human-determined membership functions are subject to the differences from person to person and from time to time; therefore the membership functions are rarely optimal in terms of reproducing desire output[1].

Neural networks, such as, ADALINE or Backpropagation can likely resemble the network with the same characteristics as fuzzy Inference system. In fact, system itself finds the optimum parameters for the system. But the price we have to pay is a long training period.

A hybrid Learning algorithm is recommended based on the trade-off between computation complexity and resulting performance [23]. In Hybrid Learning

1 In the next chapter, we will propose an approach, which enables systems to determine the membership function initially and to adjust the membership function also.

Algorithm, functional signals go forward till layer 4 and the consequent parameters are identified by one of the following methods.

- Gradient Descent Only
- Gradient Descent and one Pass of Least Squares Estimate (LSE)
- Gradient Descent and LSE
- LSE Only

CHAPTER 3. FUZZY ASSOCIATIVES

The most important characteristic of a desire object recognition algorithm is to be trained with the minimum number of objects. The ideal object recognition algorithm is to behave cognitively. Psychological research shows that humans can recall vocabularies better if the new vocabularies are looked over frequently. If you see a person one time through your life, the chance to recognize the person is still high; however, there is a possibility of not recognizing the person. As long as you meet the person frequently, you can certainly recall him. So, for the first step, we need an algorithm which can recognize an object with only one time training even though it is not guaranteed for one hundred percent recognition; in the next step, the algorithm recognition should be improved based on the number of trainees in each certain category.

There is no neural network based object recognition algorithm that can recall objects with a single trainee for each category. Algorithms, such as multi-layer neural networks require large number of objects in order to train the network. Even though FNN and FIN is introduce in [16,17,49] for pattern classification with a single training for each pattern category with a limited number of pixels, the network will be too slow for objects recognition because training is based on the pixel. Therefore, a number of pixels are inversely proportional to the network speed; it means, when a number of pixels increase, the network speed decreases.

To fulfill the task with a minimum number of trainees, Fuzzy Associative Network (FAN) is presented in Section 3.1. Fuzzy Associative network is a fundamental concept for the next two sections. In the second section, the novel concept of Fuzzy Associative Database (FAD) is presented. The advantage of FAD is to recognize the object with only one time training. In the last section, Adaptive Fuzzy Associative Database (AFAD), which has a cognitive characteristic similar to a human being, will be proposed; as time goes by, the AFAD can usually recall the category of the incoming object correctly.

3.1. Fuzzy Associative Network

The first two layers of ANFIN [24], can be used for global clustering. The first two layers of ANFIN can be employed in order to map n-dimensional feature space to ***m*** distinct clusters, as shown in Figure 3.1. If w_{ij}'s are defined by means of Gaussin function, the input layer weights are unit, and the output layer weights are defined by variables f_{ij}[2],

The network is called Radial Basis Function (RBF)[12,22,23,24,32].

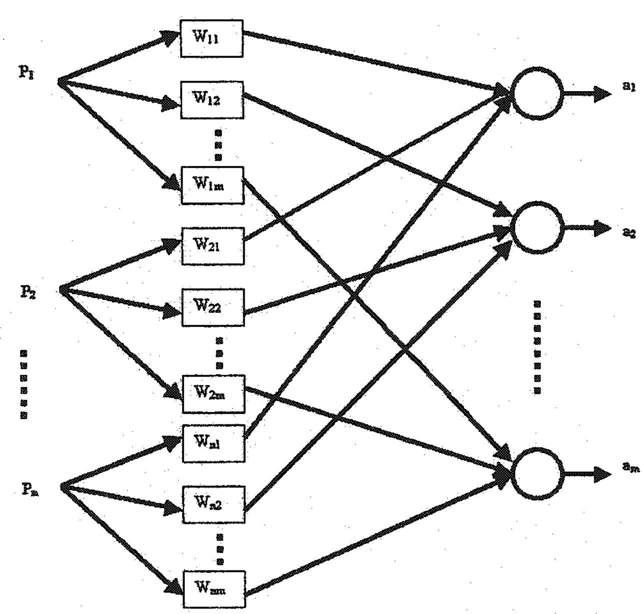


Figure 3.1: First two layers of ANFIN for global clustering

2 f_{ij} 's are determined by means of Linear Least Squared(LLS)method proposed for neural network training.

If we look more closely at Figure 3.1, w_{ij} works like weights rather than nodes. If w_{ij} 's are defined as a fuzzy sets, the network in Figure 3.1 can be presented in another way which we call a single layer Fuzzy Associative Network (FAN), as illustrated in Figure 3.2. The network name is derived from the fact that the network associates the input n-dimensional space to m-dimensional space, m distinct clusters, by means of a fuzzy approach. The FAN weights, w_{ij}'s are fuzzy sets, which can be characterized by its membership functions, typically, Triangular, Trapezoidal, and Gaussian membership function based on Definition 2.2-4 respectively.

A single layer of FAN works an unsupervised network. There is no explicit teacher, and the system forms the input patterns clusters. FAN consists of finite number of fuzzy neurons, which works in parallel.

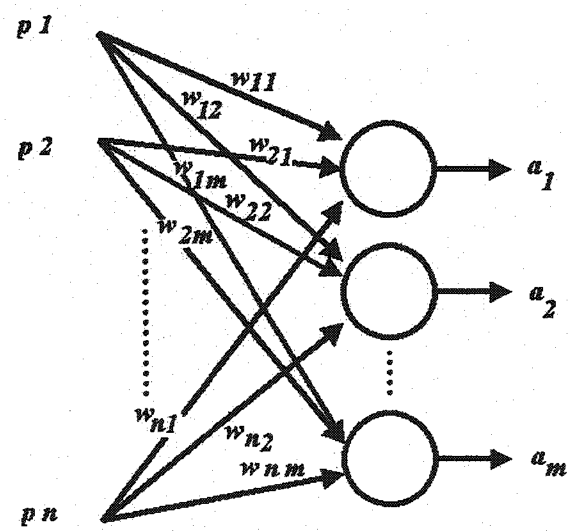


Figure 3.2: Single layer of Fuzzy Associative Network

Definition 3.1: A single layer of Fuzzy Associative network associates n-dimensional feature space into m-dimensional space based on the (3.1) equation. The network structure is shown in Figure 3.2.

$$a_j = \prod_{i=1}^{n} \mu_{w_{ij}}(p_i) \qquad j=1,2,\ \ldots,\ m \tag{3.1}$$

Where n denotes the number of features, m is the number of output and $\mu_{w_{ij}}$ is the weight of the fuzzy neuron connects the ith input feature to the jth output node. $\mu_{w_{ij}}$ can be any continuous and piecewise differentiable functions, such as, commonly used Gaussian, trapezoidal or triangular-shaped membership functions. In fact, other T-norm operators expressed in Example 2.1 can be used instead of algebraic product.

Lemma 3.1: the elements of the vector A show the extent of the similarity between the incoming pattern and each trained cluster correspondingly.

To make it simpler in the further discussion, fuzzy vectors are used to represent input and output and fuzzy matrix to represent weights.

$$P=[\, p_1, p_2, \ldots, p_n \,], \tag{3. 2}$$

where n is the number of dimension of feature space

$$A=[\, a_1, a_2, \ldots, a_m \,], \tag{3. 3}$$

where m is the number of defined cluster

$$W=[\, w_{11}, w_{12}, \ldots, w_{mn} \,]. \tag{3. 4}$$

Definition 3.2: The output of the network is determined base on the following formula. If the input of the FAN considered as Fuzzy vector,

$a_j = TN(TN(p_i, w_{ij}))$ $i=1,2, \ldots, n$ and $j=1,2, \ldots, m,$ (3.5)

where n denotes the number of features and m is the number of output

Lemma 3.2: The *condense network* representation is based on fuzzy vectors and fuzzy matrix in order to reduce the complexity of Fuzzy Associative network. The condensed form of the Fuzzy Associative network is depicted in Figure 3.3.

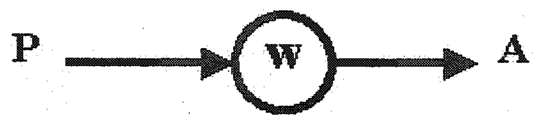


Figure 3.3: A single layer of FAM

Lemma 3.3: A single output FAN can be employed for single numerical clustering. The fuzziffied coordinates of the desired cluster center form the weights of the FAN.

Lemma 3.4: Multi-output *Fuzzy Associative network* can be employed for partial clustering if the fuzziffied coordinates of the cluster centers initialize the corresponding weights of the Fuzzy Associative network.

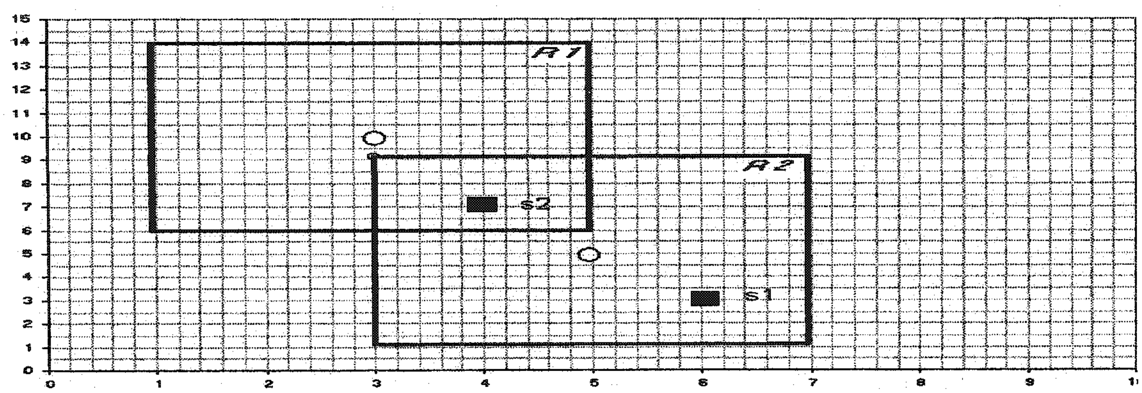


Figure 3.4: Two clusters in Euclidian feature space

Example 3.1: Let us look at the application of Fuzzy Associative network for unsupervised classification, in other words, partial clustering. The Fuzzy Associative network shown in Figure 3.5 has the capability to cluster universe of discourse similar to what is depicted in Figure 3.4. Initializing the weights is the first step in order to realize a Fuzzy Associative network. To do so, clusters' center should be determined[3], first. The cluster center coordinates in the two-dimensional Euclidean feature space are *R1* (3,10) and *R2* (5,5). Fuzziffied coordinates of the cluster forms the corresponding weight in the Fuzzy Associative network.

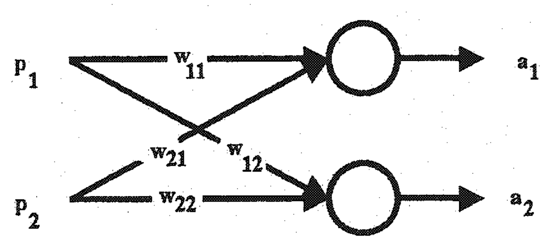


Figure 3.5: A single layer of FAN with two inputs and two outputs

$$w_{11}^T = [0 \quad 0.5 \quad 1 \quad 0.5 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0],$$

3 In the next section implicitly and in next chapter explicitly, we will discus how to determine the center of clusters.

$$w_{12}{}^{T} = [0 \quad 0 \quad 0 \quad 0.5 \quad 1 \quad 0.5 \quad 0 \quad 0 \quad 0 \quad 0],$$

$$w_{21}{}^{T} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0.25 \quad 0.5 \quad 0.75 \quad 1 \quad 0.75 \quad 0.5 \quad 0.25 \quad 0 \quad 0],$$

$$w_{22}{}^{T} = [0 \quad 0.25 \quad 0.5 \quad 0.75 \quad 1 \quad 0.75 \quad 0.5 \quad 0.25 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0].$$

Determining the weights, fuzzy matrix, the multi-input fuzzy neuron with two inputs can be constructed. If a pattern with feature similar with *S1 (6,3)* appears in the network input, the fuzziffied features are as follows.

$$p_1{}^{T} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$p_2{}^{T} = [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

Based on (3.5)equation if we use Algebraic product and minimum for T-norm, the output of the network can be determined.

$$a_1 = \min((p_1{}^{T} \times w_{11}),(p_2{}^{T} \times w_{21})),$$

$$a_1 = 0,$$

And

$$a_2 = \min((p_1{}^{T} \times w_{12}),(p_2{}^{T} \times w_{22})),$$

$$a_2 = 0.5,$$

so the pattern with the feature (5,3) *S1* is classified as a member of *R2*, not *R1*. Now let us look at the pattern with the feature (4,7) *S2*.

$$p_1{}^{T} = [0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$p_2{}^{T} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

the outputs of the network is

$$a_1 = \min((p_1{}^{T} \times w_{11}),(p_2{}^{T} \times w_{21})),$$

$a_1 = 0.125,$

And

$a_2 = \min((p_1{}^T \times w_{12}),(p_2{}^T \times w_{22})),$

$a_2 = 0.25,$

As it is expected, the pattern with properties similar to *S2* can be classified in both declared clusters *R1* and *R2*. In FAN, the out puts of the FAN show the extent of similarity between input and each defined clusters. As a result, the pattern with properties similar to *S2* is closer to the samples in cluster *R2,* than the samples in cluster *R1*. A single layer of FAN can distinguish the similarity of an input to the finite number of clusters defined for the network. FAN can improve the speed and performance of systems. In fact, FAN minimizes the training time and maximizes the speed of systems. There is no need to train the network as in ADALINE network or Back propagation network. Because of parallel processing, FAN can also speed up the performance.

3.2. Competitive Fuzzy Associative Network

In order to determine final decision for clustering of an input pattern, it is necessary to add a competitive layer after a single layer of Fuzzy Associative network as depicted in Figure 3.6. The network with the two layers is called Competitive Fuzzy Associative Network (CFAN). The first and second layer of CFAN are Fuzzy Associative network and Competitive network respectively. There exist other competitive networks for clustering, such as, CFNN and CFIS [16,17,50]. CFAN performs faster than CFNN and CFIS because CFAN has two layers less than CFNN and CFIS. CFNN and CFIS are a pixel wise pattern recognition approach; however, CFAN is an approach based on invariant values. CFAN is more efficient than CFNN and CFIS for object recognition, especially when the image size is large or rotation is involved.

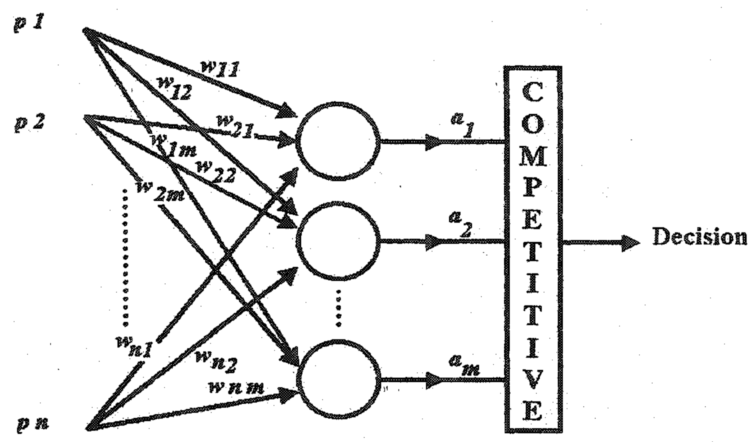


Figure 3.6: Competitive Fuzzy Associative Network

Figure 3.7 shows the condensed form of a Competitive Fuzzy Associative network. Let us look at competitive layer and see how it works. As explained in section 3.1, the outputs of a single layer FAN is a fuzzy vector A whose elements show the extent of the similarity between the incoming object and the defined clusters. A competitive layer works by finding the index i^* of the neuron with the largest net input and setting its output to 1 and all the other outputs are set to 0.

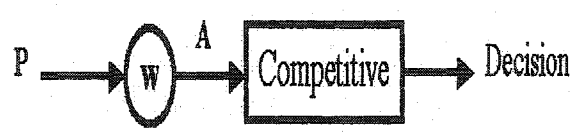


Figure 3.7: Condensed form of a CFAN

Example 3.2: Let us go back to Example 3.1 and use CFA network to classify pattern with property similar to *S2*; the result is

A^T=[0.125 0.5]

or

a_1=0.125,

a_2=0.25

then

max=0.25=> i^*=2

As a result, *S2* is classified as second class (*R2*).

3.3. Fuzzy Associative Database

Fuzzy Associative Database (FAD) is a reformulated CFAN based on a database concept[44].

Meanwhile, FAD can work as a supervised and unsupervised network. Unsupervised FAD works exactly as CFAN for clustering purposes; on the other hand, supervised FAD can work similar to Backpropagation. In fact, supervised FAD can be applied to classify even separated categories, like XOR problem.

Fuzzy Associative Database and CFNN are clustering network [33]; additionally, Fuzzy Associative Database is recommended for object recognition.

FAD is a feasible approach because of the advancement in the field of computer software and hardware especially in the field of Database. FAD consists of a Fuzzy Database (FD) and a fuzzy search Engine (FSE). FD holds the specified pattern information, as in human memory, and FSE performs as a brain, which processes incoming information based on information exists in memory, database. In the next two subsections we look at an FD and FSE in more detail.

3.3.1. Fuzzy Database

Fuzzy Database includes two tables, Figure 3.8. Table-one holds the coordinate of the specified clusters. The index field of the table-two shows the cluster number, and second field in table two specify the class in which the cluster belongs to.

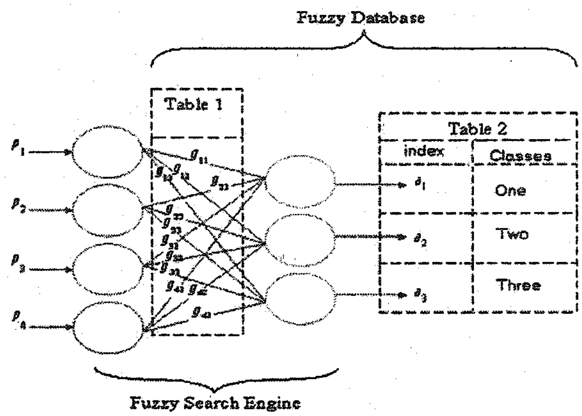


Figure 3.8: Fuzzy Associative Database construction

3.3.2. Fuzzy Search Engine

Fuzzy Search Engine performs a search on the database to find the closest class with similar properties with the input pattern. In order to conduct fuzzy searching for classification, a Bank of Fuzzy Associative Memory Matrix (BFAMM) [43,44] is first built

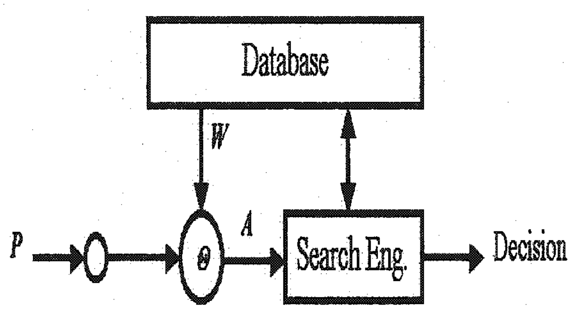


Figure 3.9: Condensed form of FAD construction

based on the information held on table-one of the FD. BFAMM associates incoming objects to one of the clusters recorded on table-two of the FD. BFAMM can be constructed through the following expression:

$$BFAMM = \begin{bmatrix} w_{11} & w_{21} & w_{31} & \cdots & w_{m1} \\ w_{12} & w_{22} & w_{32} & \cdots & w_{m2} \\ w_{13} & w_{23} & w_{33} & \cdots & w_{m3} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{1n} & w_{2n} & w_{3n} & \cdots & w_{mn} \end{bmatrix} \qquad (3.6)$$

where $w_{ij}(i=1,2,...,m$ *and* $j=1,2,...,n;$ n denotes the number of features and m is the number of output*)* are the fuzzified center of the ***i-th*** cluster. Next, based on Fuzzy Associative network formula, (3.5), the extent of similarity of the input pattern with the defined clusters can be determined.

Definition 3.3: Let $\boldsymbol{P}$ be a n×1 fuzzy vector and $\boldsymbol{W}$ be a n×m fuzzy matrix. Let $\boldsymbol{p_j}$, $j=1,2,...,n$ be the fuzzy entries of $\boldsymbol{P}$ and $\boldsymbol{w_{ij}}$, $i=1,2,...,m;$ $j=1,2,...,n$ be the fuzzy entries of $\boldsymbol{W}$. Then the operation Θ returns a new $1 \times m$ fuzzy vector $\boldsymbol{A=P^T \Theta W}$ with the fuzzy entry $\mathbf{a_i}$ of $\boldsymbol{A}$ given by

$$a_i = \prod_{j=1}^{n} \min(p_i, w_{ij}), \quad i=1,2,3,\ ...,\ m. \qquad (3.7)$$

Lemma 3.5: The elements of the vector A show the extent of the similarity between the incoming pattern and the specified cluster in the second table of the fuzzy database. The index $\boldsymbol{i^*}$ of the closest initialized cluster in the second table to the incoming object is the one which maximizes $\boldsymbol{a_i}$, that is

$$a_{i^*} = \max_i (a_i) \quad i=1,2,\ldots,m \qquad (3.8)$$

On the same record with the winner cluster, the second field in Table-two presents the class the incoming object belongs to.

Fuzzy Associative Database requires less time for training because only one trainee is required for each class in comparison with Backpropagation. The first trainee in each category has a critical rule in performance of the network. As long as, the trainee's feature vector approaches to the center of corresponding cluster, the performance of FAD on the pattern approaches to the ideal recognition. However, if the trainee's feature vector deviates from the center of corresponding cluster, FAD get more likely involved in misclassification. Adaptive Fuzzy Associative Database is an improved algorithm for FAD. Adaptive Fuzzy Associative Database can adjust the weight automatically in order to decrease the chance of misclassification.

3.4. Adaptive Fuzzy Associative Database

Finding the cluster centers is not an easy task. Even though it is possible for us to find cluster centers, there is no guarantee that the defined centers are located in the ideal positions. Competitive Fuzzy Associative network and Fuzzy Associative Database can misclassify the incoming pattern if the feature vector deviates from the initialized cluster center. To solve misclassification problem, Adaptive Fuzzy Associative Database (AFAD) [43] is introduced. Adaptive Fuzzy Associative Database is an adaptive feed forward network designed to minimize possibility of misclassification. Construction of AFAD is illustrated in Figure 3.10 and 3.11.

Weight initialization and pattern classification in AFAD are similar to FAD; meanwhile, AFAD updates the weights according to the following steps:

1. Determine the extent of similarity of the incoming object with the defined clusters (A), Lemma 3.1.
2. Update the records, whose a_i is none zero, by means of improved Kohonen rule

$w_i(N+1)=w_i(N)+\alpha(t)a_i(N)(p-w_i(N))$ (3.9)

where

N is the number of iterations

t is the number of trainees in a specific category

p is the the network input

α is learning rate (3.10)

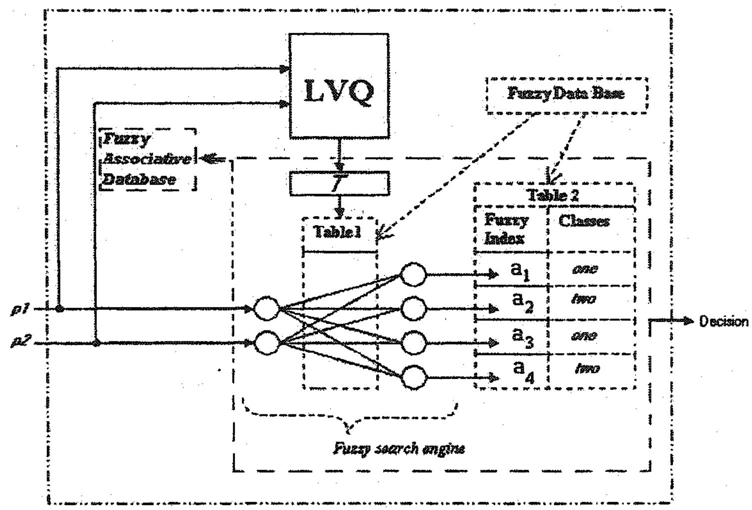


Figure 3.10: Fuzzy Associative Database construction

In the next chapter, Fuzzy Associative Database and Adaptive Fuzzy Associative Database are going to present for real time object recognition.

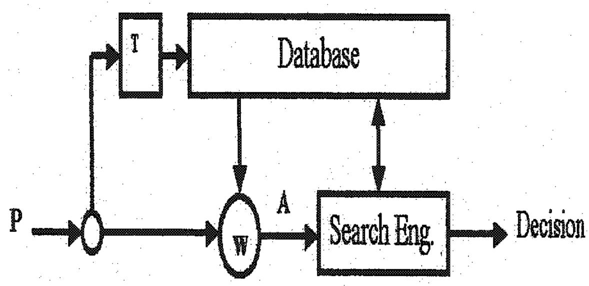


Figure 3.11: Condensed form of FAD construction

CHAPTER 4. OBJECT RECOGNITION AND TRACKING

In this chapter, design methodology for real time object recognition and object tracking is presented. The design methodology can be used for flexible manufacturing automation, especially the design of tracking vision sensor. The vision sensor can potentially work for robots as human eyes. For instance, if a robot is supposed to drill the center of a work piece, first the robot should check to see if the desired piece is underneath the drill bit. If the piece is under the drill, the robot has to move the drill towards the center. To do so, the robot has to behave intelligently by recognizing the work piece. After the robot recognizes the right piece under the drill, the robot has to determine the deviation of the current position of the drill bit from the center of the piece. When deviation is determined, the robot will move the drill towards the center of the work piece. The explained task, for the drilling robot can be divided into two distinct categories, object recognition and object tracking. In Chapter 4.1, object recognition is proposed in detail, and object tracking, in Chapter 4.2 is presented.

4.1. Object Recognition

The object recognition task is computationally complex and expensive. As a result, performance of the recognition-based tracking algorithm relies on the efficiency of the recognition module. It is necessary to mention that object recognition differs from image recognition. In the case of object movement, the appearance of an object is changed from the viewpoint of a stationary viewer, but the appearance of an image particularly remains the same, regardless of the image position because of the properties of 2D images.

In this section object recognition methodology is presented. In the first subsection, system constraints set by hardware are explained. In Section 4.1.2, scene reconstruction from a serialized signal is presented. Invariant values and fuzzifiers are demonstrated in Section 4.1.3 and Section 4.1.4 respectively. In Section 4.1.5 and Section 4.16, Input fuzzy vector and BFAMM structures are illustrated. The composition operator is introduced in Section 4.1.7, and finally in 4.1.8 object recognition is presented.

4.1.1. Capturing an Image

Charge Coupled Devices (CCD Everfocuse EX100) transforms the light reflected from the object to the video signal, and image grabber devices (Data Translation DT3120CCD) change the video signal to the serialized signal. The captured single frame maximum resolution is 500 (H) by 580 (V) pixels gray scale with speed of 30 frames per second. In order to manipulate the serialized signal, a software package is required.

4.1.2. Preprocessing

After, the scene is reconstructed from the serialized signal, a third order Low-pass filter is applied to the reconstructed image in order to minimize the effect of noise on the system performance. Since pixel-wise computation is computationally expensive, invariant-object recognition is employed. To do so, feature extraction has to be conducted.

A gray level image has to transform to two levels to ease the feature extraction. For that reason, a threshold is applied over the gray level image. For example, a gray level image is defined as $f(x,y)$ and the threshold is defined similar to Figure 4.1.

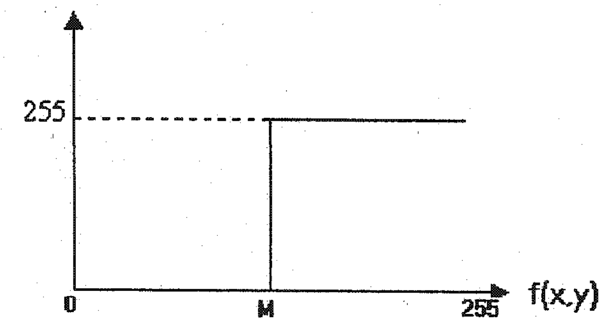


Figure 4.1: Threshold for two level image

In this technique, the *f(x,y)* values blew *M* are compressed to black (0), the *f(x,y)* values above *M* are stretched to white (255). Working on binary images guarantees better and faster performance for the vision-based systems. Logical operations can be substituted for arithmetic operations. Logical operations work much faster than mathematical operations. An Image can be transferred from two level to binary, simply by applying scalar (1/255). In this case, all the white and black pixels change to 1 and 0 respectively.

4.1.3. Invariant Values

Pre-processing transforms a gray scale image to a binary image. The binary image works as an input for the feature extraction module as shown in Figure 4.2.

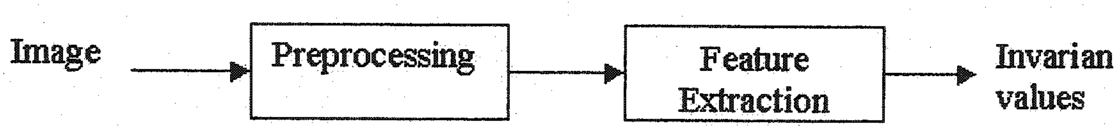


Figure 4.2: Invariant value block diagram

Invariant values characterize an object to be recognized by measurements. In fact, the values are very similar for the objects in the same category, and distinct for the objects in different categories.

Generally, it is common practice to use various approaches for extracting a reliable feature vector. Some of the well-known techniques for invariant feature extraction are furrier descriptors, regional descriptors and moment invariants[4]. One regional descriptor and one moment invariant are employed in order to trade off between complexity and efficiency for object recognition purposes. Usage of furrier descriptors is limited because the appearance of an object is varies from position to position because of misplacement and rotation in the project, and furrier descriptors are particularly sensitive to perturbations in the object boundary.

Used for recognition purpose in this thesis, a regional descriptor and moment invariant, namely, a compactness factor and a second moment invariant are as follows, respectively:

$$C = \frac{\left(\sum_{x=1}^{h}\sum_{y=1}^{w} f_{boundary}(x,y)\right)^2}{\sum_{x=1}^{h}\sum_{y=1}^{w} f(x,y)} \tag{4.1}$$

$$M = \mu_{20} + \mu_{02} \tag{4.2}$$

where ,

$$\mu_{20} = \frac{\sum_{x=1}^{h}\sum_{y=1}^{w} (x-\bar{x})^2 . f(x,y)}{\sum_{x=1}^{h}\sum_{y=1}^{w} f(x,y)} \tag{4.3}$$

$$\mu_{02} = \frac{\sum_{x=1}^{h}\sum_{y=1}^{w} (y-\bar{y})^2 . f(x,y)}{\sum_{x=1}^{h}\sum_{y=1}^{w} f(x,y)} \tag{4.4}$$

[4] For more information refer to Appendix A

And h, w, $f(x,y)$ and $f_{boundary}(x,y)$ are height, width, gray level of the pixel at (x,y), gray level of the pixel on image boundary respectively.

4.1.4. Fuzzifier

The inputs, weights, and outputs for FAN are all fuzzy sets (see Section 3.1), so before any operation the crisp values should be transferred to the fuzzy domain. To do so, the values have to be fuzziffied[5] by means of the following equation.

$$f(x) = \max\left(\min\left(\frac{x-(1-k)a}{2ka}, \frac{(1+k)a-x}{2ka}\right), 0\right) \qquad (4.5)$$

x is a self-variable ($x \in X$)

a is an invariant value

k is a support factor coefficient that is determined by the desired precision for object recognition. For instance, for an object with a thickness less than 5 mm, k is 0.025.

f is a fuzzy set corresponding with the crisp value.

4.1.5. Input Fuzzy Vector

The fuzziffied invariant values form an input fuzzy vector (2×1) , namely, **P**. Input fuzzy vector is shown in equation (4.6) and(4.7).

$$\mathbf{P}=\{\forall\, p_j \,/\, p_j = \text{fuzziffied invariant value } \& \ j=1,2\} \qquad (4.6)$$

[5] The different membership functions, such as triangular, Trapezoidal and Gaussian are explained in Section 2.1.1.

$$P = \begin{pmatrix} \text{Fuzziffied invariant value 1} \\ \\ \text{Fuzziffied invariant value 2} \end{pmatrix} \qquad (4.7)$$

4.1.6. Bank of Fuzzy Associative Memory Matrix

The records of Table 1 form the columns of the BFAMM is a fuzzy matrix, (see Section 3.3). BFAMM is constructed based on the records of the Table-one of FD. If the feature space is two-dimensional and the number of distinct clusters is set to four, then BFAMM is expressed based on Equations (3.6) as follows.

BFAMM=$\{\forall b_{ij} / b_{ij}$ is a fuzzy set & $i=1,2$ & $j=1,2,3,4\}$ (4.8)

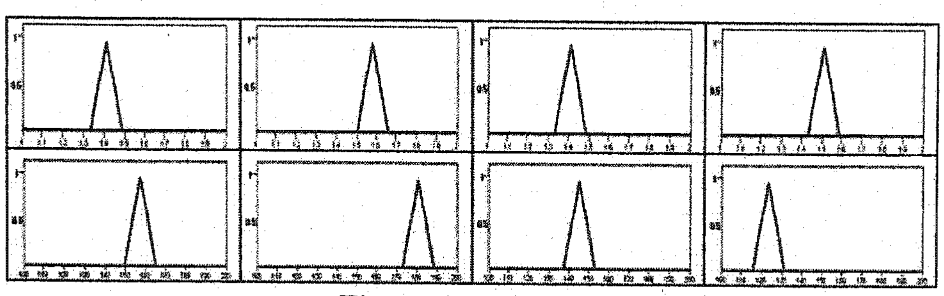


Figure 4.3:BFAMM

4.1.7. Composition

According to equation (3.5), Composition operation is defined in the following if minimum and algebraic product are used as T-norm.

$$A = P^T \Theta BFAM \qquad (4.10)$$

$$a_j = \prod_{i=1}^{2} \min(p_j b_{ij}), j = 1,2,3,4 \qquad (4.11)$$

4.1.8. Recognition

If composition applies on the transpose of input fuzzy vector and BFAMM, the resulting fuzzy vector (Lemma 3.1) shows the extent of the similarity of an incoming object with each defined category, (see equation 3.8). The class of incoming object is determined based on Table two of FD, Figure 4.3. Here, j_* is the index number of the cluster to which an incoming object belongs.

$$d_{j_*} = \max_j(d_j), \quad j=1,2,3$$

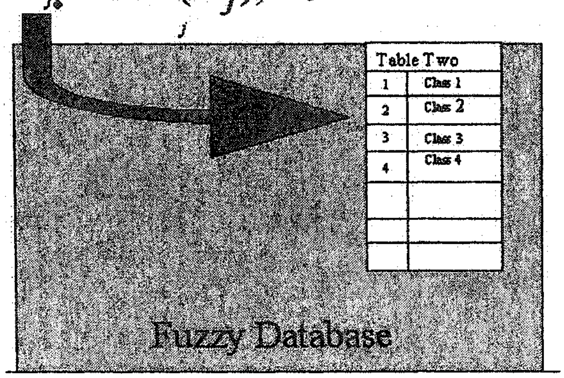


Figure 4.3: Clustering schema

4.2. Object Tracking

After the desired object is recognized, it is necessary to find the deviation from the desired object in order to track the object. The method of tracking is called Recognition-based tracking. The flow chart for the object tracking is depicted in Figure 4.4.

The position of the tracked object is determined by the center of the gravity [8,19,27,33,41] through the (4.12) and (4.13) equations.

$$\bar{x} = \frac{\sum_{x=1}^{h}\sum_{y=1}^{w} x.f(x,y)}{\sum_{x=1}^{h}\sum_{y=1}^{w} f(x,y)} \tag{4.12}$$

$$\bar{y} = \frac{\sum_{x=1}^{h}\sum_{y=1}^{w} y.f(x,y)}{\sum_{x=1}^{h}\sum_{y=1}^{w} f(x,y)} \tag{4.13}$$

Where ***h*** is the height of image

w is the width of image

f(x,y) is the gray level for a pixel located in x and y position

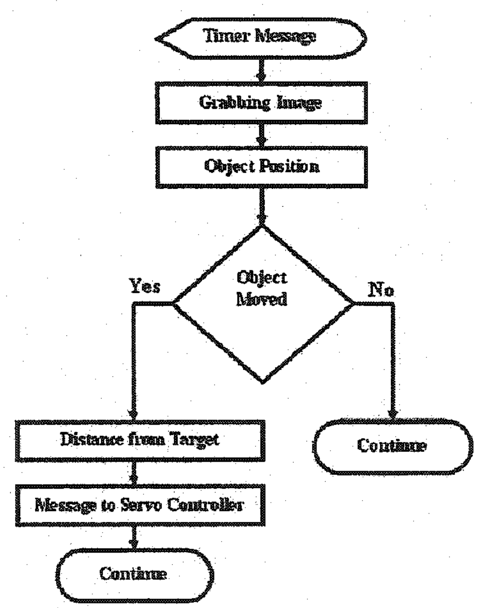


Figure 4.4: Object tracking flow chart

After determining the object position, the distance from the object can be calculated based on (4.14) equation.

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{4.14}$$

where x_2 and y_2 are the coordinates for the object position and x_1 and y_1 are the tracker coordinates. By having a robot distance from target, the robot can be guided to the right position.

CHAPTER 5. IMPLEMENTATION AND EXPERIMENT

It is necessary to experience the performance of the aforementioned algorithms online to discover their advantages and disadvantages. To do so, the platform, designed for pure object tracking in [27], is altered for recognition-based object tracking. In this chapter, the implementation and experiments on recognition-based object tracking are presented respectively in Section 5.1 and Section 5.2.

5.1. Implementation

Implementation consists of three major procedures, namely, sensing, recognition, and tracking. Figure 5.1 shows the flow of implementation for recognition-based object tracking, and the complete implementation block diagram is depicted in Figure 5.2. In this section, sensing, recognition and tracking modules are presented in Section 5.1.1, 5.1.2 and 5.1.3, respectively.

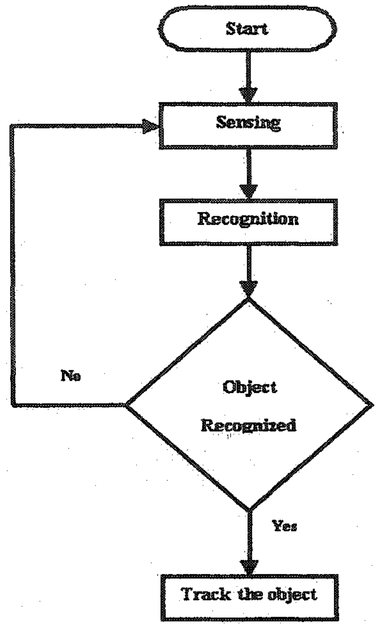


Figure 5.1: Recognition-based object tracking implementation flow chart

5.1.1. Video Card Library

The Microsoft Vision SDK package is mostly used for image capturing and image manipulation. Microsoft Vision SDK is a commercial package. This means that Microsoft has built the package for plug and play usage in Windows operating system; as a result, Vision SDK creates a huge overhead for any software that uses the package, and more overhead for software means slowing down the software. The most important aspect for a real time software designer is software speed. Since the object recognition algorithm has to be tested on line, software speed is critical. To side step the overhead problem, access to the Vision SDK source code is necessary, but because of the Microsoft monopoly, access to the code is impossible.

As a result, development of a self-contained package for image capturing and image manipulation is necessary.

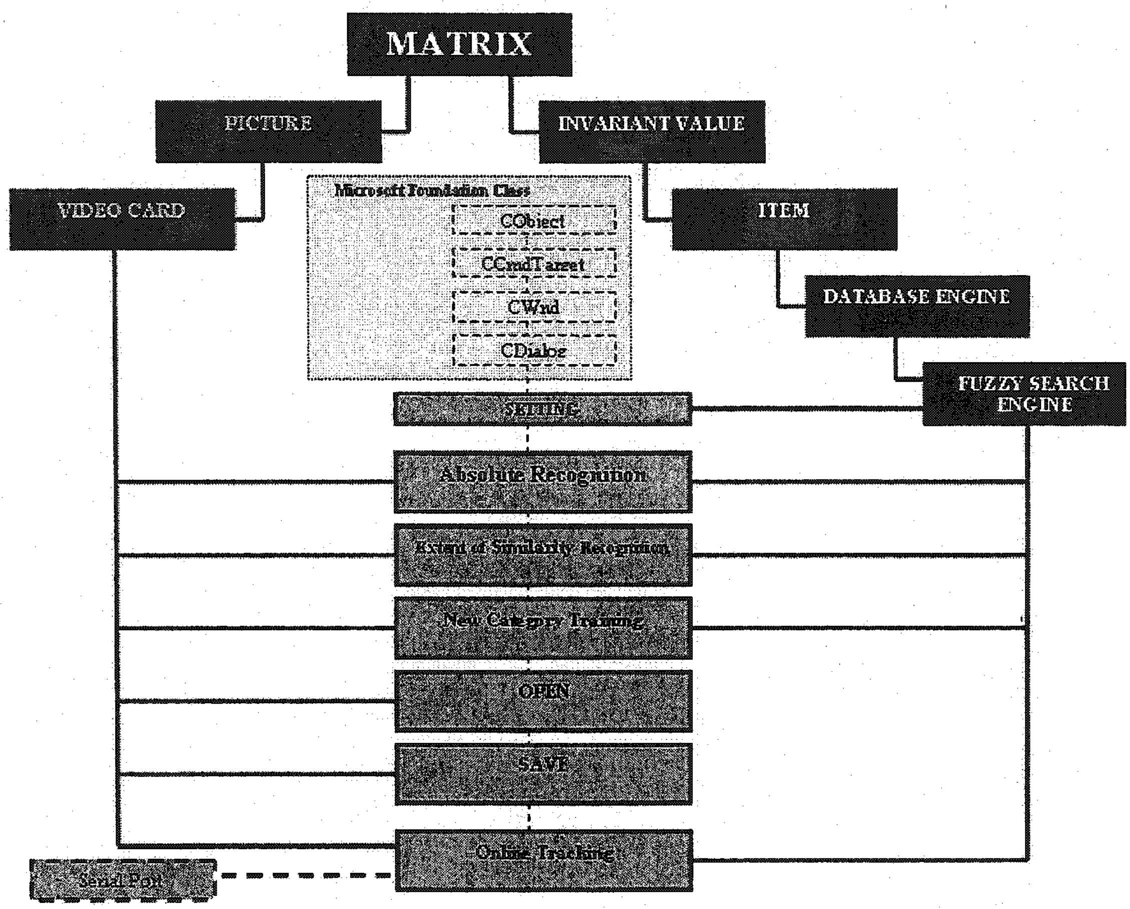


Figure 5.2: Recognition-based object tracking implementation block diagram[6]

Video Card library[7] is developed for this purpose to guarantee the highest speed for recognition and tracking. Video-Card library consists of three classes, namely, Matrix, Picture and VideoCard. VideoCard class is a derived class from Picture class, and Matrix is a super class for Picture objects. The structure of Video-Card Library is depicted in Figure 5.3.

[6] The solid rectangles classes have been implemented specially for recognition-based object tracking by the author, S. Shahir. The software manual in more detail can be found in the appendix B.

[7] Video-Card library is implemented by the author, S. Shahir

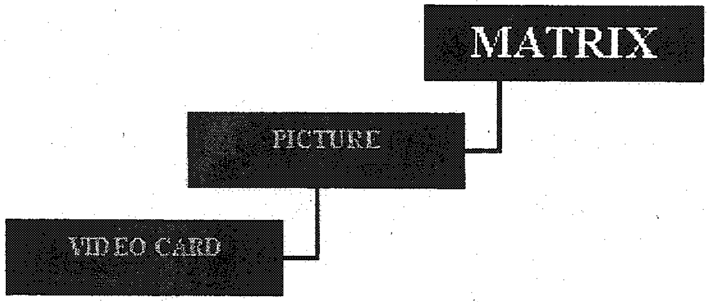


Figure 5.3: Structure of Video Card library

The advantage and disadvantage of Video Card library verses Microsoft Vision SDK are illustrated in Table 5.1.

	Video Card Library	Microsoft Vision SDK
Operation platform	Unix, Linux, Windows 9x, 2000 and XP	Windows 98
Over head	No	Yes
Speed	High	Slow
Specific hardware required	Yes	No
Access to source code	Yes	No

Table 5.1: Video Card library versus Microsoft Vision SDK

5.1.2. Object Recognition

AFAD (Section 3.4) is employed for recognizing the objects. Various modules contribute to object recognition implementation, such as, Invariant Value, Database Engine and Fuzzy Search Engine classes. The Hierarchical object recognition Library is illustrated in Figure 5.4. The object recognition block diagram is depicted in Figure 5.5.

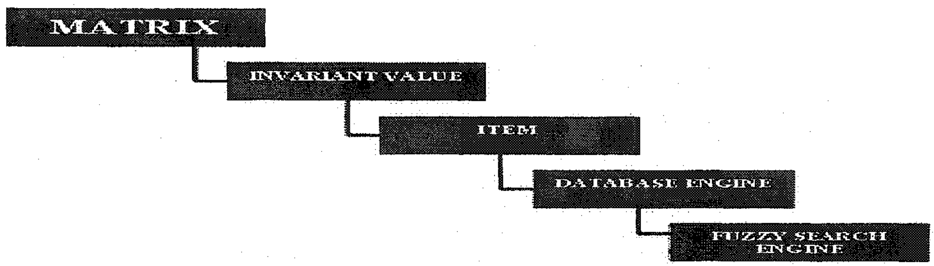


Figure 5.4: Object recognition implementation hierarchical library chart

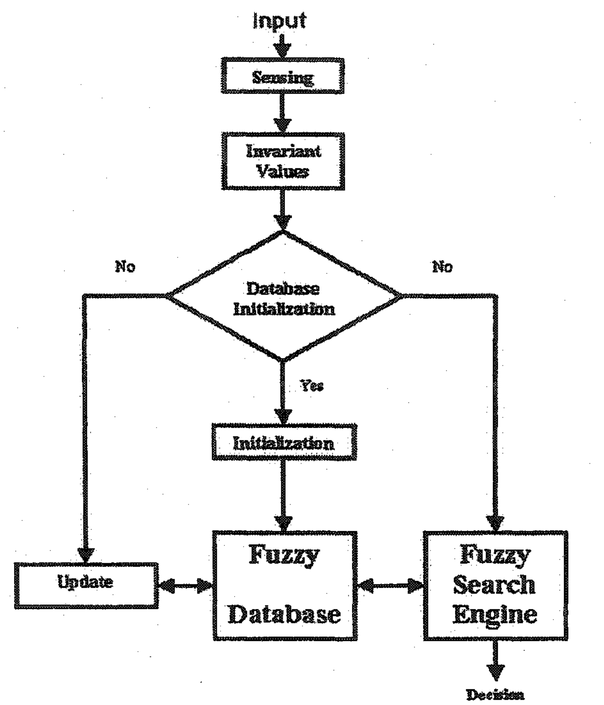


Figure 5.5: Object recognition block diagram

5.1.3. Object Tracking

Online-Tracking class is implemented for online tracking purpose. Online-Tracking[8] class is derived from Serial Port class[9]. The tracking block diagram is illustrated in Figure 5.5. The platform (see Figure 5.6) is enabled to track objects by means of the implemented classes if the object is recognized as the desired object.

[8] Online-Tracking class is implemented by the author, S. Shahir

[9] Serial Port class is implemented by J. Yang [27]

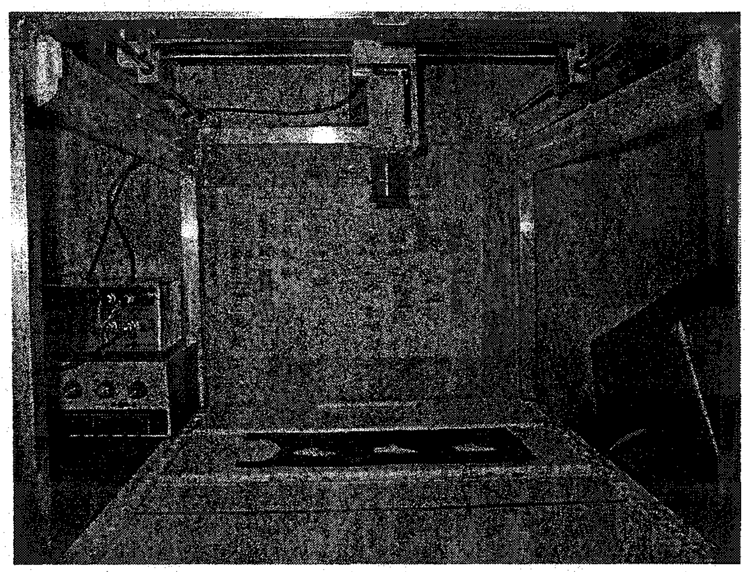


Figure 5.6: Vision based tracking platform

5.2. Experiment

In order to test the accuracy of the theory and implementation, the system is trained with eight different objects. The objects are planar and the thickness of the objects is almost 4 millimeters. The objects are depicted in Figure 5.7, below:

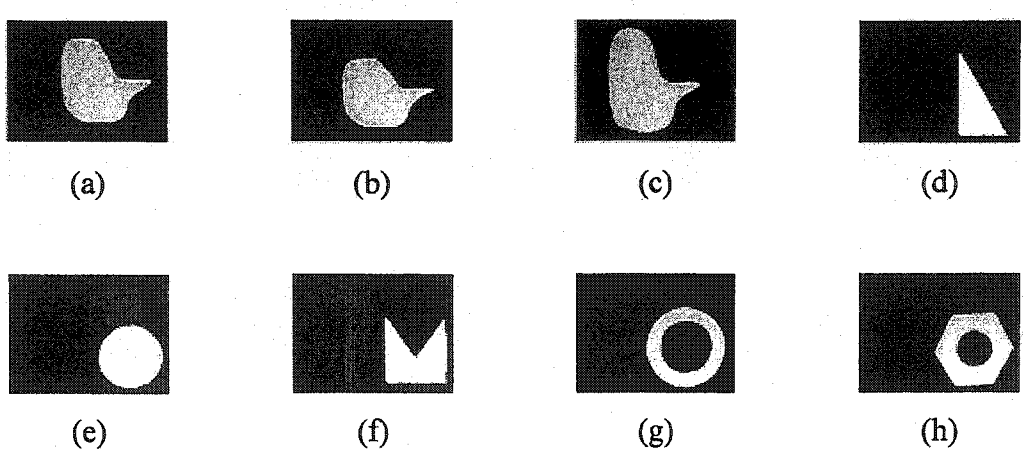


Figure 5.7: Objects used as trainee for FAD

The trained system is tested with the objects in different positions and at different angles. All the objects were recognized successfully. The system is designed for two types of recognition. Extent of similarity and absolute recognition are developed based on FAN and FAD approach, respectively. The extent of similarity recognition discriminates the objects, which are similar to the incoming object, Figure 5.9, and absolute recognition classify the closest trained category in the FD[10] with the viewed object, Figure 5.10.

The system can be trained with Finite number of categories. If overlapped clusters exist the possibility of misclassification arises depending on how severe the clusters overlap. Let us look at a case in which a new category's cluster overlaps on one of the existing categories. As an example, the system is trained with the object shown in Figure 5.8.

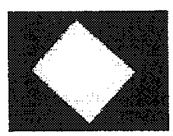


Figure 5.8: The object added to database

By looking at Table 5.2, we notice that the new trained object cluster is partially overlapped with the 6^{th} trained object; as a result, there is a possibility of misclassification. There are four ways to avoid the possibility of misclassification. First, by deleting the old trained class which has conflicted with the new one. Secondly, by changing the invariant values to eliminate the conflict among the trained objects. Thirdly, by changing the feature space dimension. Finally, by applying a smarter approach for object recognition. The first solution is not wise because there is a possibility that both object categories are needed. The second remedy is not practical because finding two invariant values for a finite number of objects is not always possible. The third approach is almost the same as what we have done so far,

[10] refer to Section 3.3.1

but the feature vector can be changed from two-dimension to three or more if it is necessary. In our experiment, we have applied AFAD as another recognition approach to solve the misclassification in the case of cluster overlapping.

Table 5.2 shows the trained category's properties. The compactness factor and moment factor have the best performance for rounded objects.

Table 5.3 shows that FAN and FAD and AFAD has the same efficiency while the approaches are applied for cluster without overlapping. Table 5.4 shows that AFAD reacts better than FAN and FAD when the clusters are overlapped. In Table 5.5, FAN, FAD, AFAD, and Backpropagation are compared with each other based on their performance classifying overlapped objects without and with overlapping and training speed. As a result, AFAD and Backpropagation are similar with respect to the case of classification; however, AFAD requires less time for training.

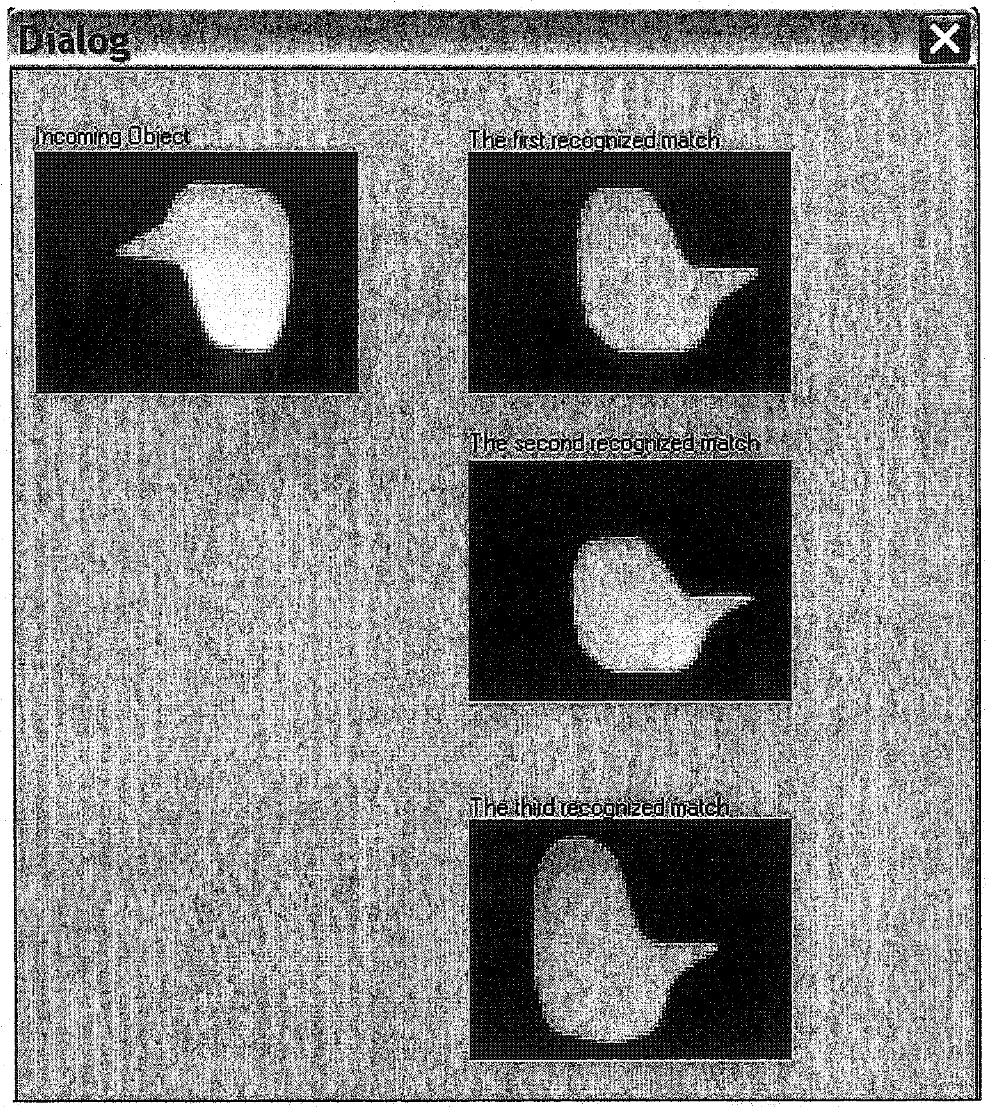


Figure 5.9: Extent of similarity recognition dialog box

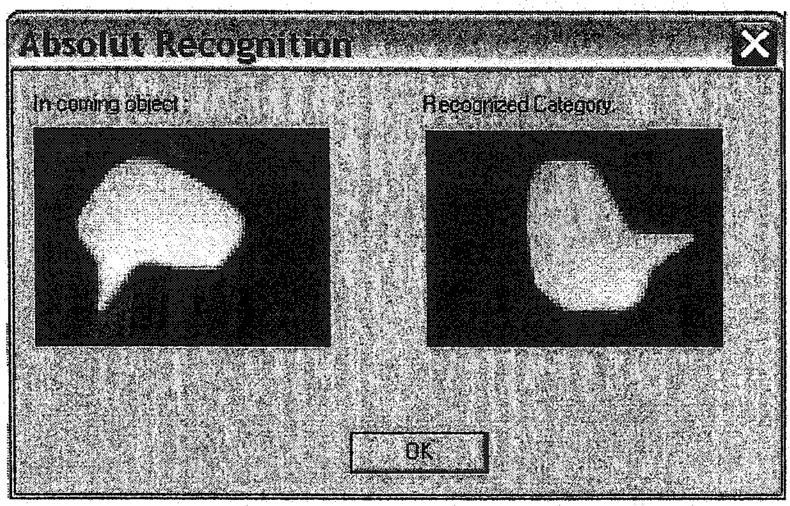


Figure 5.10: Absolute recognition dialog box

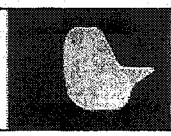
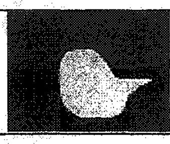
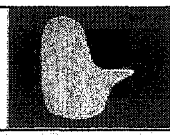
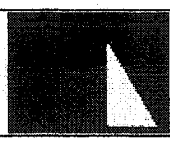
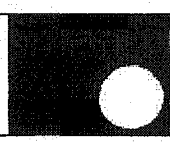
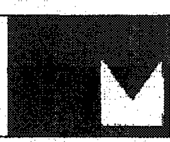
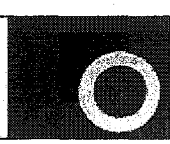
	Planar Object	Image Size Pixel (Height × Width)	Invariant Values	Max	Min	Mean	Variance
1		240×320					
			Com	28.74	22.63	26.19	2.44
			Mom	220	200	210.88	27.32
2		240×320					
			Com	28.34	24.49	26.51	1.15
			Mom	172	168	169.65	1.23
3		240×320					
			Com	31.41	25.42	28.28	2.54
			Mom	277	269	273.21	7.17
4		240×320					
			Com	41.8	29.11	34.50	19.36
			Mom	143	125	135	26.05
5		240×320					
			Com	19.87	18.49	19.20	0.098
			Mom	130	118	123.84	13.28
6		240×320					
			Com	47.48	34.39	39.93	24.13
			Mom	164	147	154.77	23.53
7		240×320					
			Com	94.55	88.71	92.41	1.71
			Mom	287	267	277.53	33.52

Table 5.2

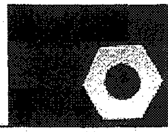
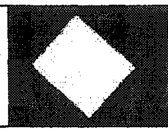
8		240×320					
			Com	59.82	55.32	56.85	0.78
			Mom	214	199	206.82	22.46
9		240×320					
			Com	29.41	15.29	22.04	37.43
			Mom	241	220	231.48	29.57

Table 5.2: Trained object invariant values properties (25 samples for each category)

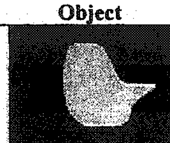
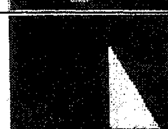
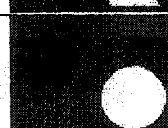
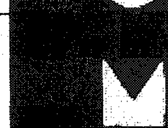
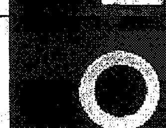
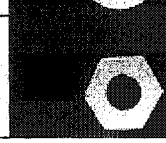
	Planar Object	FAN	FAD	AFAD
1		100%	100%	100%
2		100%	100%	100%
3		100%	100%	100%
4		100%	100%	100%
5		100%	100%	100%
6		100%	100%	100%
7		100%	100%	100%
8		100%	100%	100%

Table 5.3: Recognition of objects without clustering overlapping

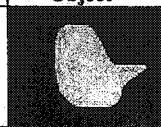
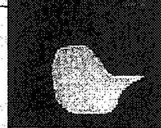
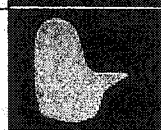
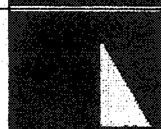
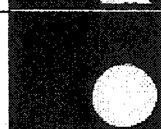
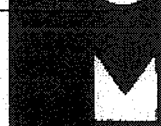
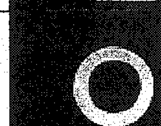
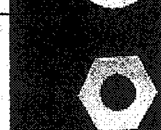
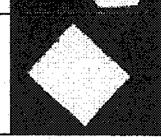
	Planar Object	FAN	FAD	AFAD
1		100%	100%	100%
2		100%	100%	100%
3		100%	100%	100%
4		60%	60%	100%
5		100%	100%	100%
6		100%	100%	100%
7		100%	100%	100%
8		100%	100%	100%
9		50%	50%	100%

Table 5.4: Recognition of objects with cluster overlapping

	Recognition Approach	Minimum Overlapping	Severe Overlapping	Training Speed
1	FAN	100%	60%	Very Fast
2	FAD	100%	60%	Very Fast
3	AFAD	100%	98%	Very Fast
4	Backpropagation	100%	98%	Slow

Table 5.5: Comparison between different object recognition approaches

CHAPTER 6. CONCLUSION AND FUTURE WORK

In this chapter, summary of the conducted research is presented, Section 6.1, and then the work recommended for further research is proposed, Section 6.2.

6.1. Conclusion

Throughout this thesis, vision guided multiple planar object recognition and tracking are presented. In fact, the presented online object recognition theories, FAD and AFAD, can potentially be applied to design of a vision sensor, and combination of online object recognition and object tracking can also end with a tracking vision sensor.

In short, FAD and AFAD are introduced for object recognition purpose. The methods guarantee high speed for recognition. FAD and AFAD are introduced as a supervised network. FAD training time are minimum in comparison with other object recognition methods. For that reason the method is suitable to use in a production environment and easy to use with less skilled workers in production lines.

The performance of FAD and AFAD is similar to a human being. By viewing an object only one time, it can recognize the object. As time going by, the possibility of misclassification declines in AFAD. FAD and AFAD are introduced as a feasible approach for multiple objects recognition in this thesis. The approaches are applicable for recognition of any object as long as there exist the appropriate invariant values.

The implemented software, object recognition and object tracking can be used for real time manufacturing purposes, such as inspecting production quality, differentiating between various products, finding the part to be picked up by robot, guiding the robot towards the desire object.

6.2. Future work

Generally, future work for either object recognition or object tracking relies on the specified applications. The application of object recognition and tracking is numerous, and it can be used in many disciplines. Meanwhile, future work implicitly can be focused on three main streams; namely, testing the FAD for different application, improving the current platform and designing stand-alone vision sensors. The research on each of the main streams itself can be divided into the sub research groups explicitly as follows:

a. Future work related with FAD
   - Apply AFAD for character recognition
   - Employ AFAD for human face recognition
   - Test the efficiency of AFAD for recognition on moving objects

b. Improving the current platform
   - Use another camera in order to measure the distance from object.
   - Exercise 3D object recognition by means of two CCD cameras
   - Work on recognition of Occluded Object.
   - Enhance the system by using the color CCD.
   - Improve the micro-code implemented for controlling the servomotors, in a way that embedded software can communicate with the micro-code.
   - Use more powerful servomotor for tracking purpose.

- Substitute a video chip instead of CCD camera

c. Stand alone vision sensor

- Design of a vision sensor by means of Field-programmable gate array (FPGA)
- Implement a tracking vision sensor by means of Field-programmable gate array (FPGA)
- Realize a specific object recognition integrated circuit (ASIC)
- Design a specific object recognition and tracking integrated circuit (ASIC)

APPENDIX A

Regional Descriptors

Various approaches are applied as follows to describe image region[36]:

Area: The area of a region is defined as the number of pixels in the region.

$$A=\sum_{x=1}^{h}\sum_{y=1}^{w} f(x,y) \tag{a.1}$$

where h, w and $f(x,y)$ are height, width and gray level of the pixel at (x,y) respectively.

Perimeter: the perimeter of a region is the length of its boundary.

$$P=\sum_{x=1}^{h}\sum_{y=1}^{w} f_{boundary}(x,y) \tag{a.2}$$

where h, w and $f_{boundary}(x,y)$ are height, width and gray level of the pixel on image boundary respectively.

Compactness: Compactness of a region is defined as follows. Compactness is dimension less quantity and is minimal for a disk-shaped region.

$$C=\frac{\left(\sum_{x=1}^{h}\sum_{y=1}^{w} f_{boundary}(x,y)\right)^2}{\sum_{x=1}^{h}\sum_{y=1}^{w} f(x,y)} \tag{a.3}$$

where h, w, $f(x,y)$ and $f_{boundary}(x,y)$ are height, width, gray level of the pixel at (x,y), gray level of the pixel on image boundary respectively.

Central Moments

Moments with respect to origin μ_{pq} are defined[28] as follows

$$\mu_{pq} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}(x-\bar{x})^p (y-\bar{y})^q f(x,y)d(x-\bar{x})d(y-\bar{y}) \tag{a.4}$$

The following seven functions of central moments are invariant to rotation and scale differences.

$$M_1 = \mu_{20} + \mu_{02} \tag{a.5}$$

$$M2 = (\mu_{20} + \mu_{02})^2 + 4\mu_{11} \tag{a.6}$$

$$M_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \tag{a.7}$$

$$M_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2 \tag{a.8}$$

$$\begin{aligned} M_5 = {} & (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{21} + \mu_{03})^2 - 3(\mu_{21} + \mu_{03})^2] \\ & + (3\mu_{21} + \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{12} + \mu_{03})^2] \end{aligned} \tag{a.9}$$

$$M_6 = (\mu_{20} + \mu_{02})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] \tag{a.10}$$

$$\begin{aligned} M_7 = {} & (3\mu_{12} + \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] \\ & - (\mu_{30} + 3\mu_{12})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{12} + \mu_{03})^2] \end{aligned} \tag{a.11}$$

APPENDIX B

Software Manual

Before running the program, make sure the vision platform is installed properly as explained in [27].
After installing the hardware and drivers for DT3120, you can run the Vision Guided Multiple Planar Object Recognition and Tracking software.
To start the platform working, click "Video On" on "Device" from the menu bar, Figure b.1.

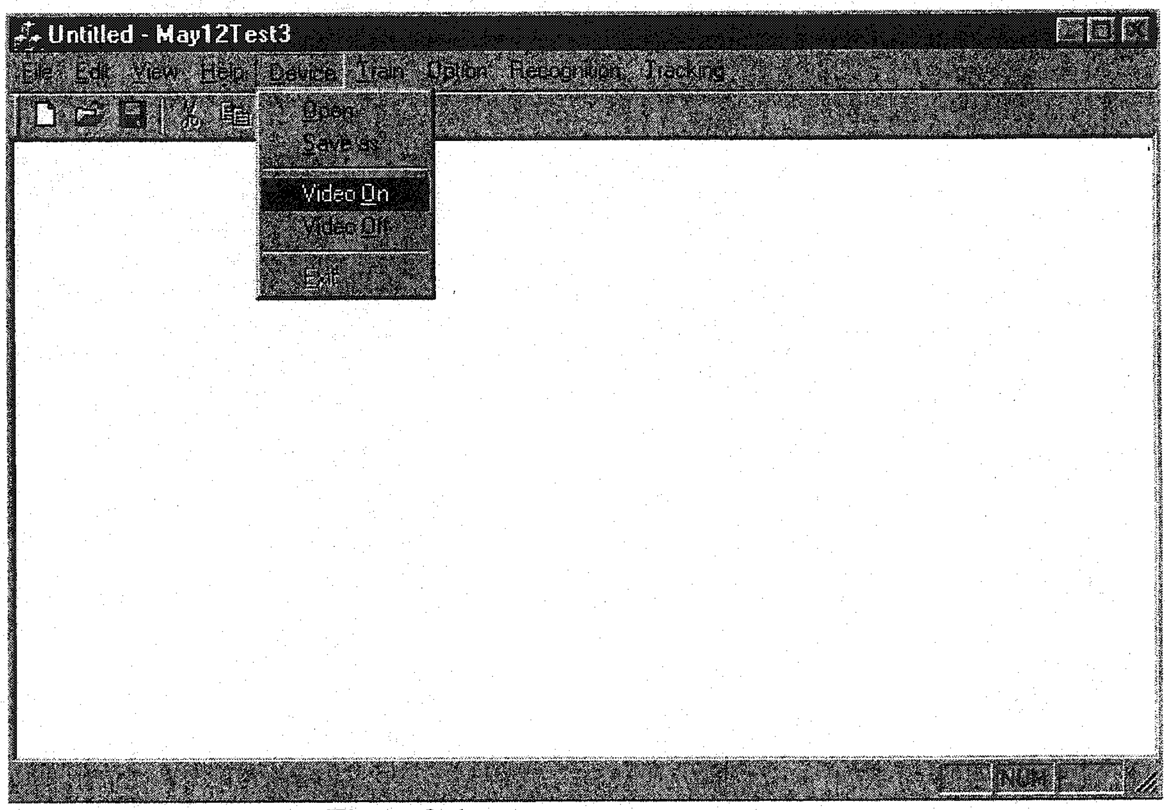


Figure b.1: Start asynchronous capturing

The platform starts working, Figure b.2. Any single object or product can be trained through "Train", "FAD", "New Category" and "Unsafe" from the menu bar Figure b.3. When you click on the "Unsafe", a training dialog box will bop up Figure b.4. If the captured object is the desired one, click train button on the dialog box; otherwise, click cancel button, Figure b.5. At any time, you can ask the package to recognize the object. In this software, the author has developed two ways to recognize objects Absolute recognition and Extent of similarity recognition. To perform Absolute recognition, while the system is working, click on "Absolute recognition" on "Recognition" from the menu bar. A dialog box will pop up a show the closest object in the database with the incoming object, Figure b.5. To perform Extent of similarity, click on "Extent of similarity" on "Recognition" from the menu bar Figure b.6.

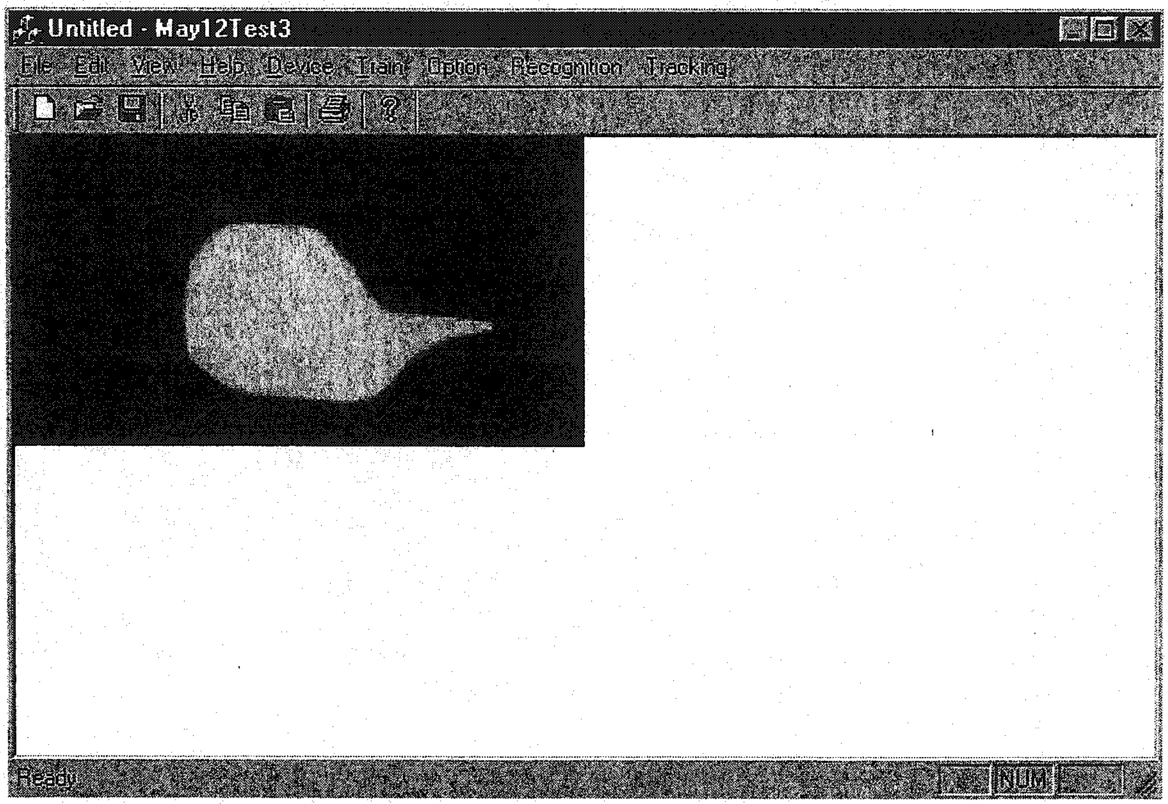


Figure b.2: Normal view of a single planar object

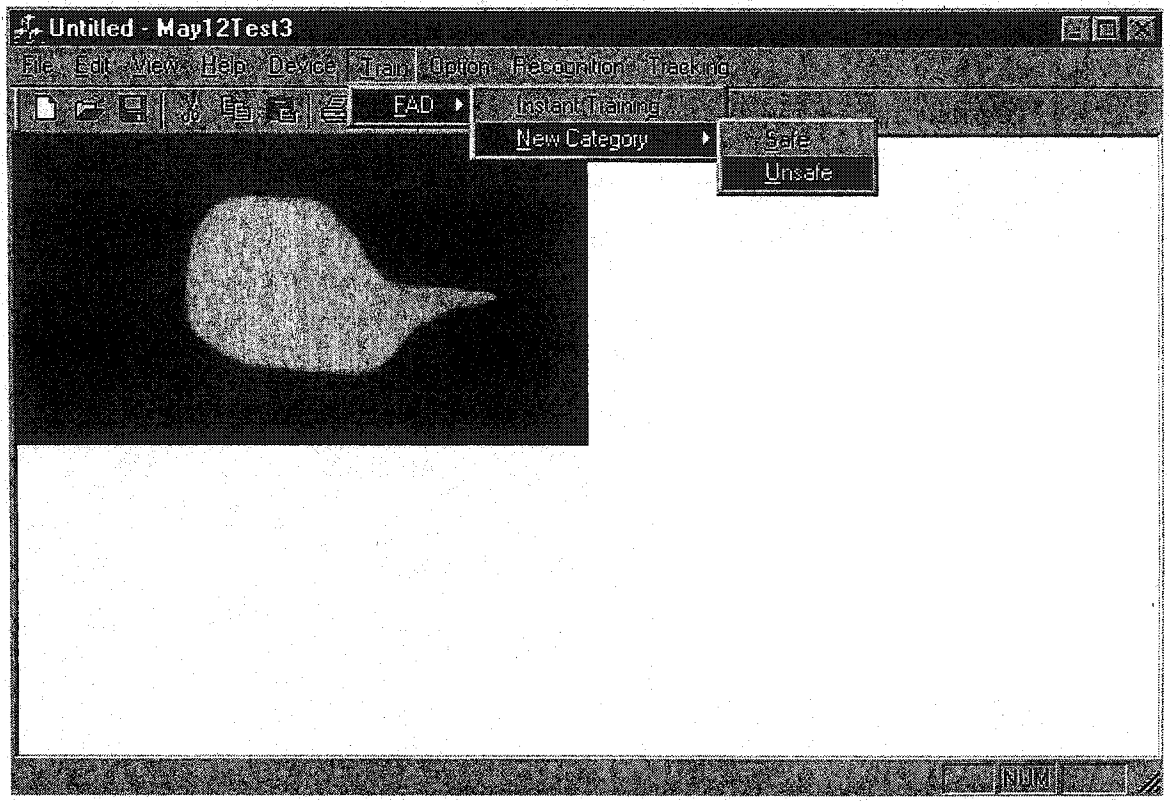


Figure b.3: Training a single object

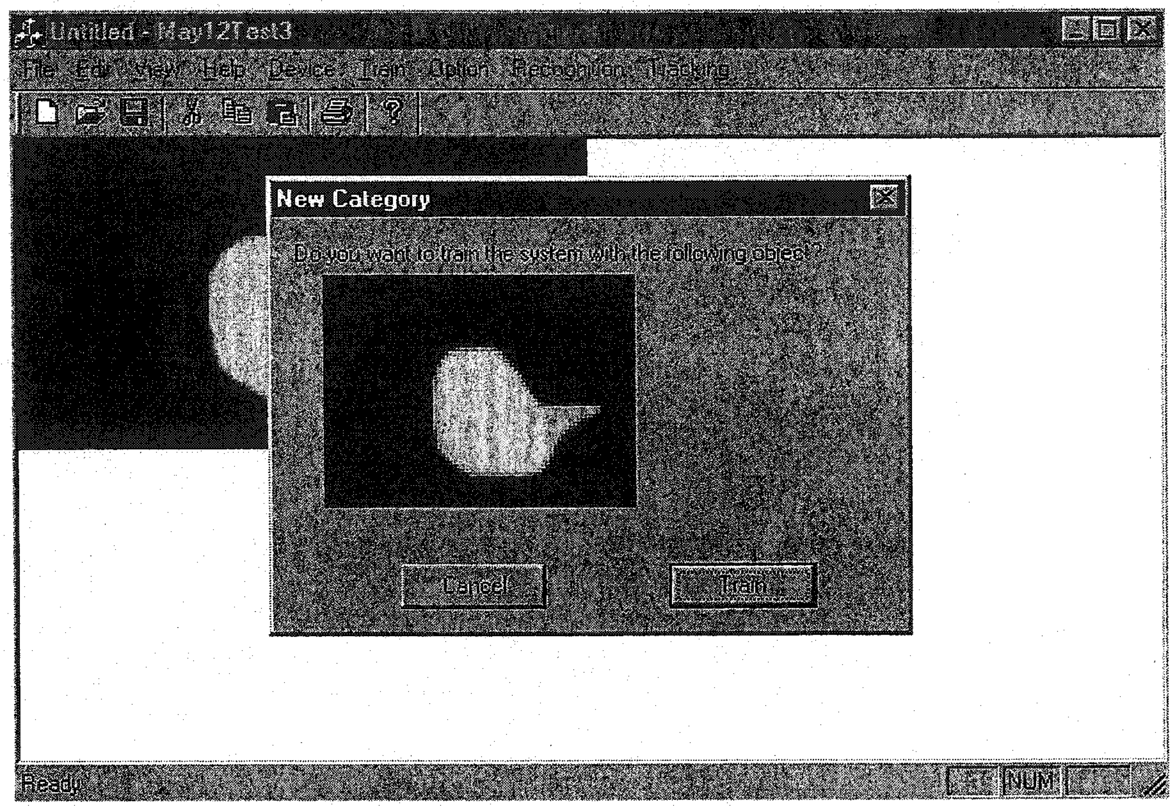


Figure b.4: New category dialog box

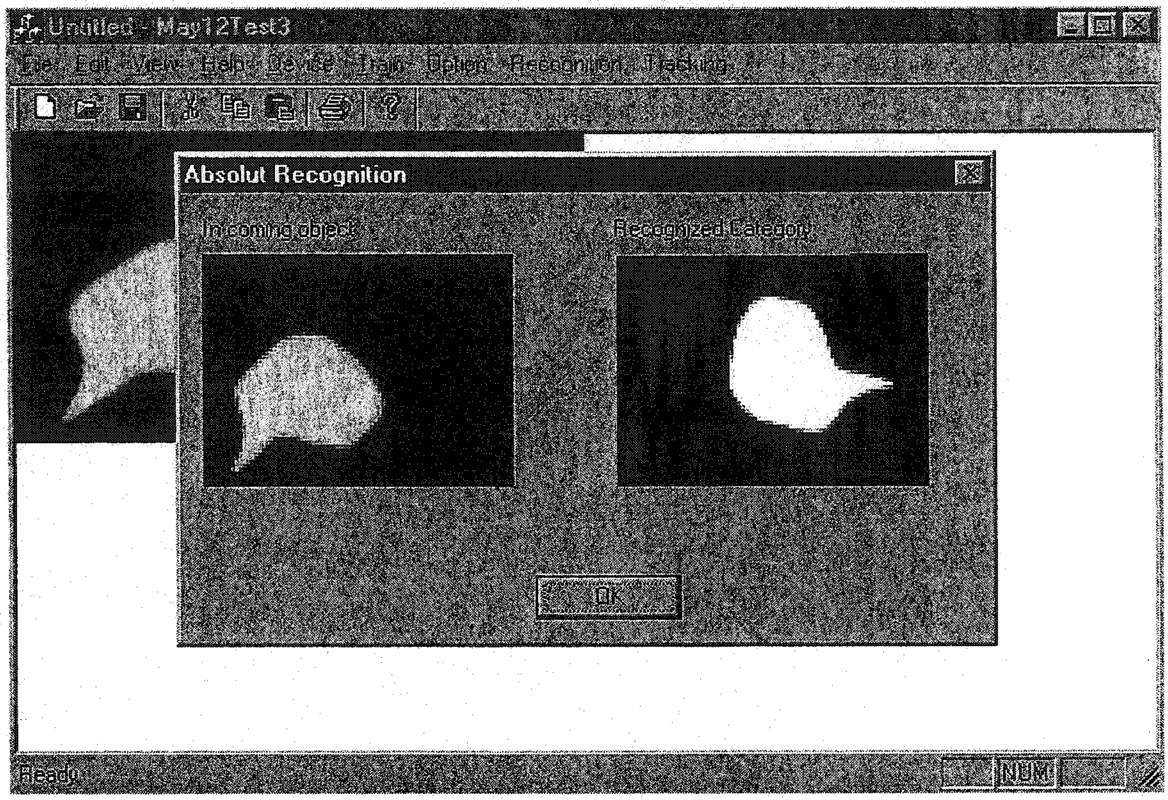


Figure b.5: Absolute recognition dialog box

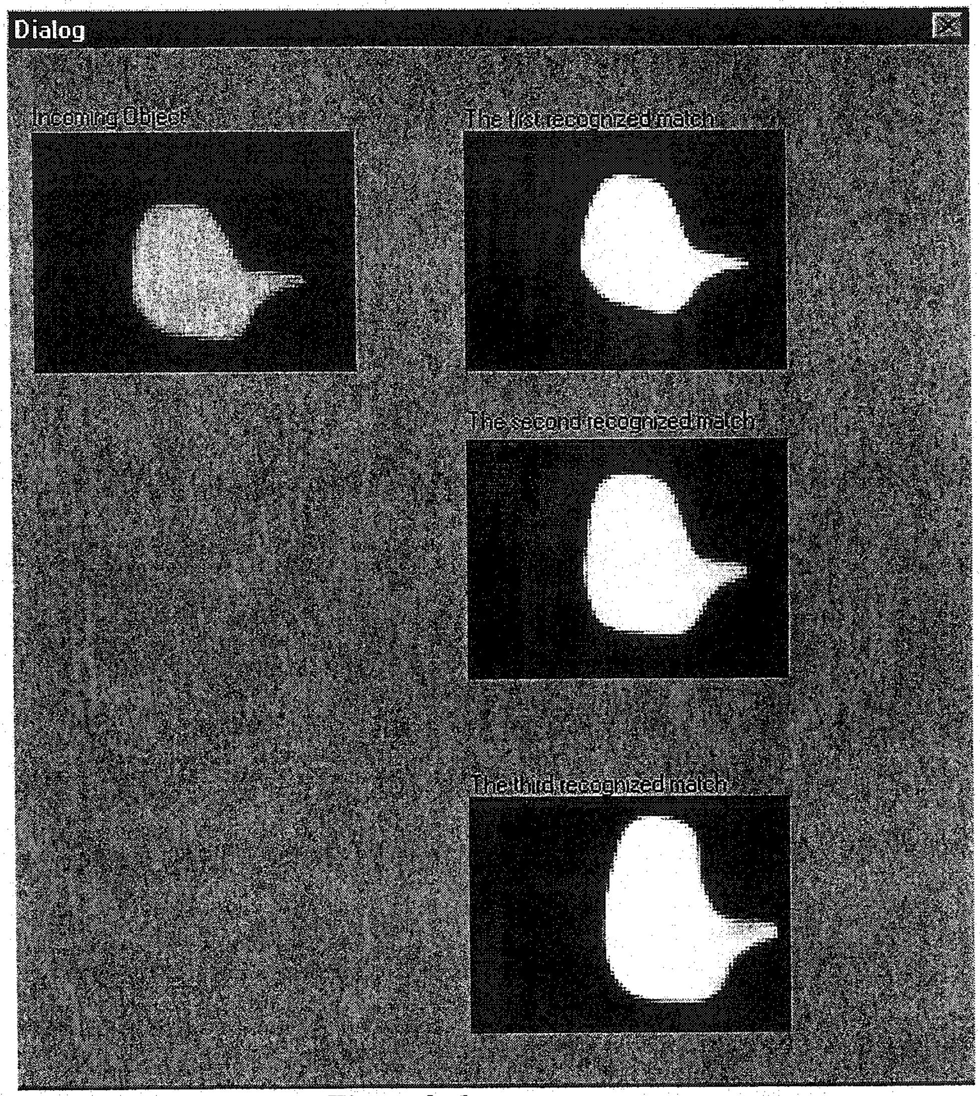


Figure b.6: Extent of similarity dialog box

To track a desire object, first the object has to be specified by training the object on "train object" option on "Tracking" from the menu bar, Figure b.7. Then to start tracking click on "Online Tracking On", on "Tracking" from the menu bar, Figure b.7. Thus, the tracking dialog box pops up, Figure b.8. From now on, when ever the object move, the tracker will determine the distance from target and shows the distance in x and y axis direction in the dialog box, Figure b.8; at the same time this program guide the tracker toward the target. The process continues till the tracker reaches the target, Figure b.9. The system can locate the position of target with the tolerance of 0.5-millimeter approximation. To close the tracking dialog box at the end, you have to click "Online Tracking Off" on "Tracking" from the menu bar, Figure b.10. To close the system first stop image capturing by clicking on the "Video

Off" on "Device" on the menu bar, Figure b.11, and then close the software by clicking on "Exit" on "File" from the menu bar.

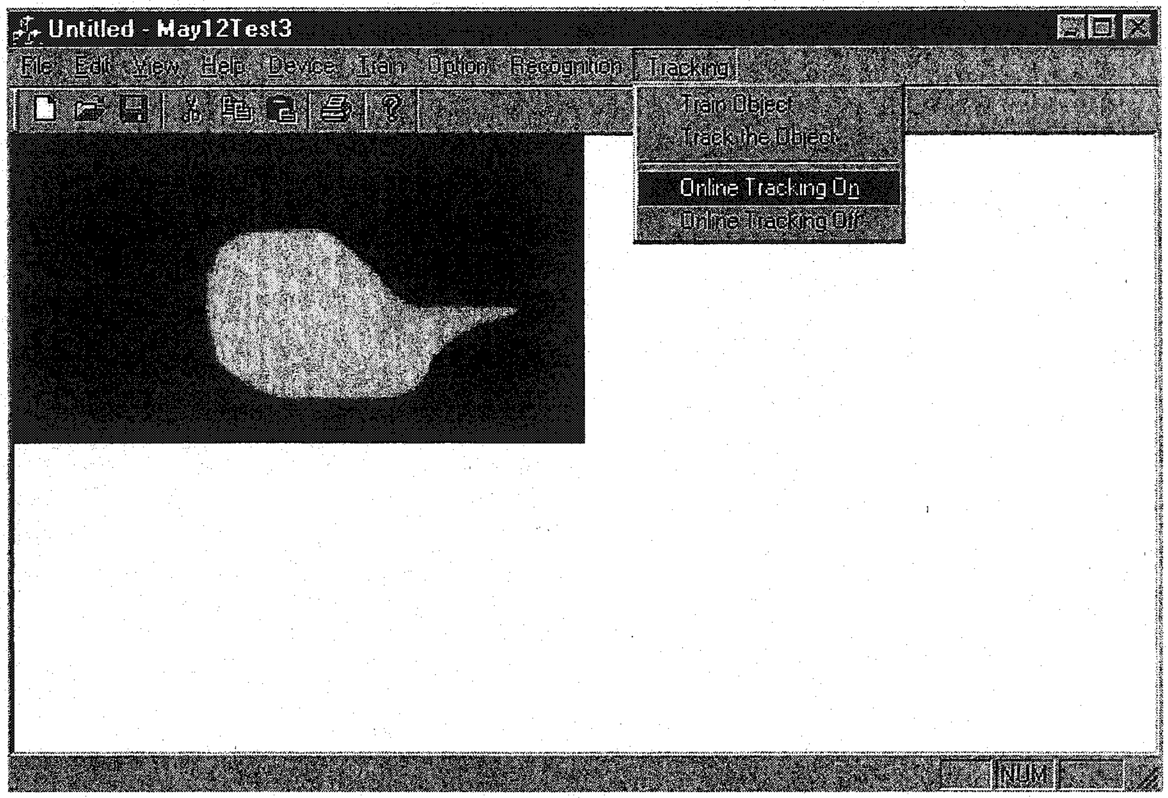


Figure b.7: Start online tracking

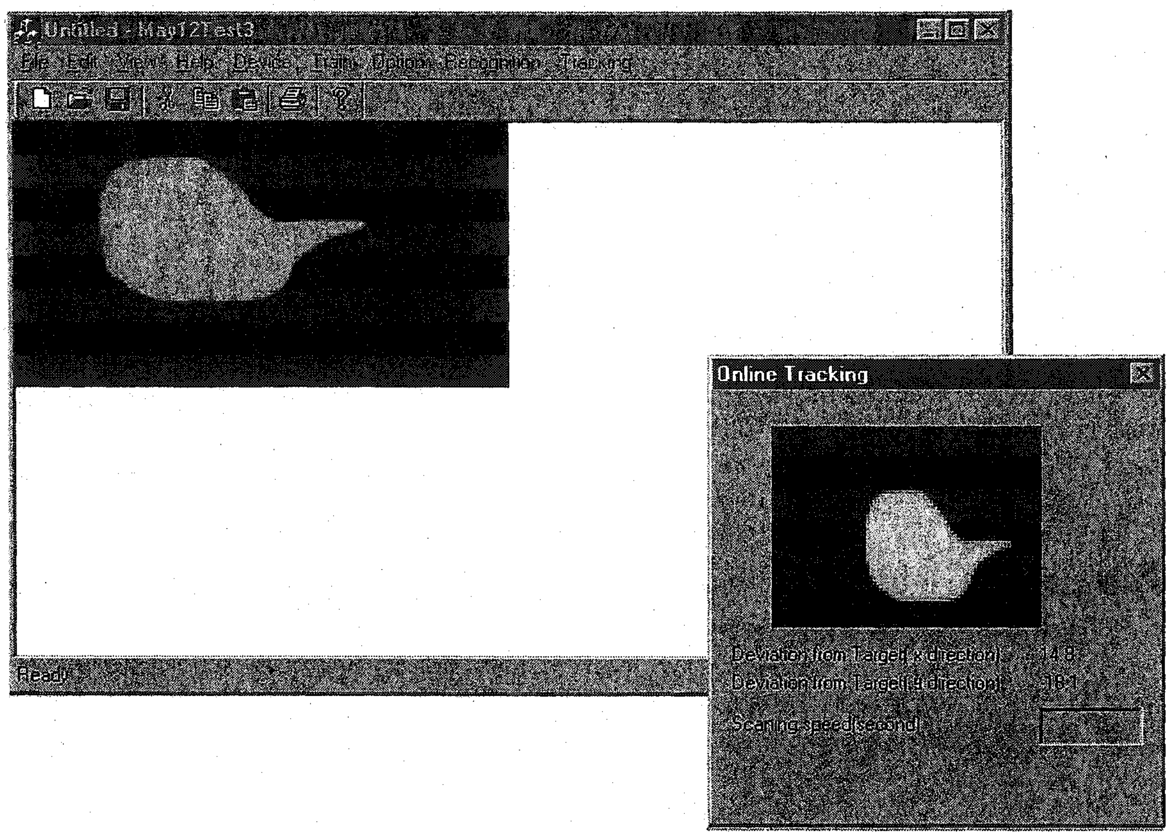


Figure b.8: Tracking dialog box

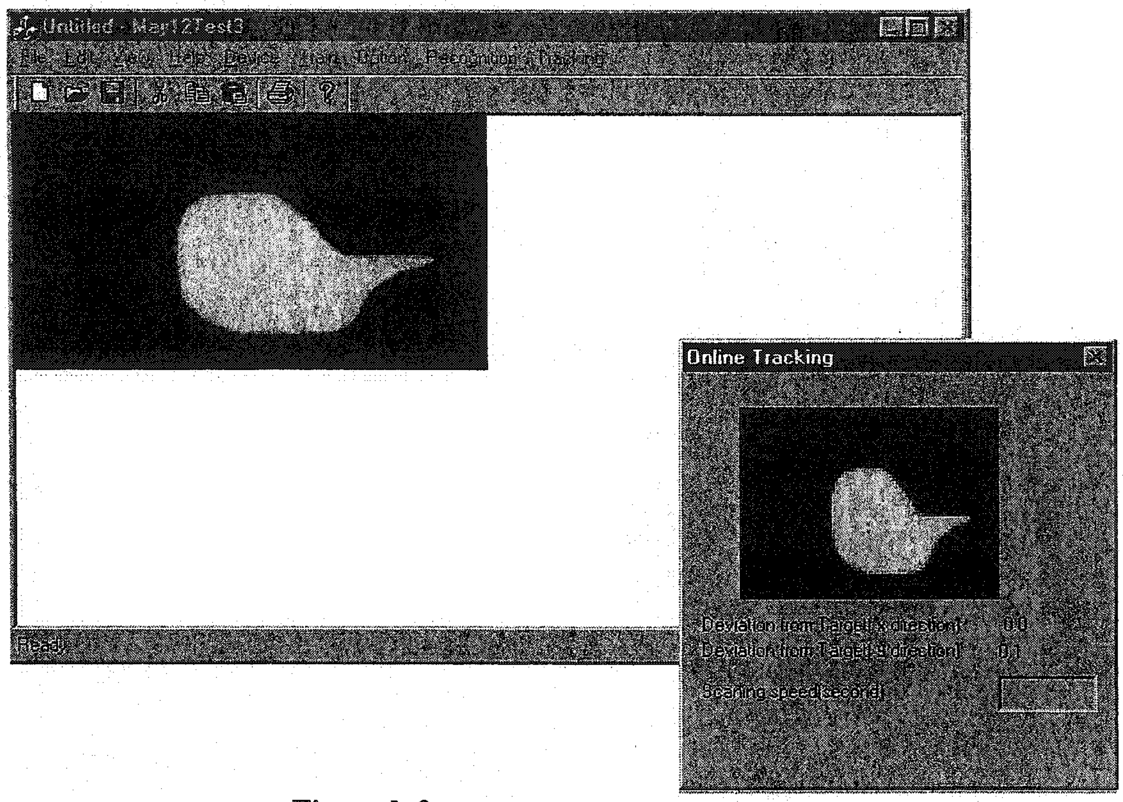


Figure b.9: Superimposed the tracker and target

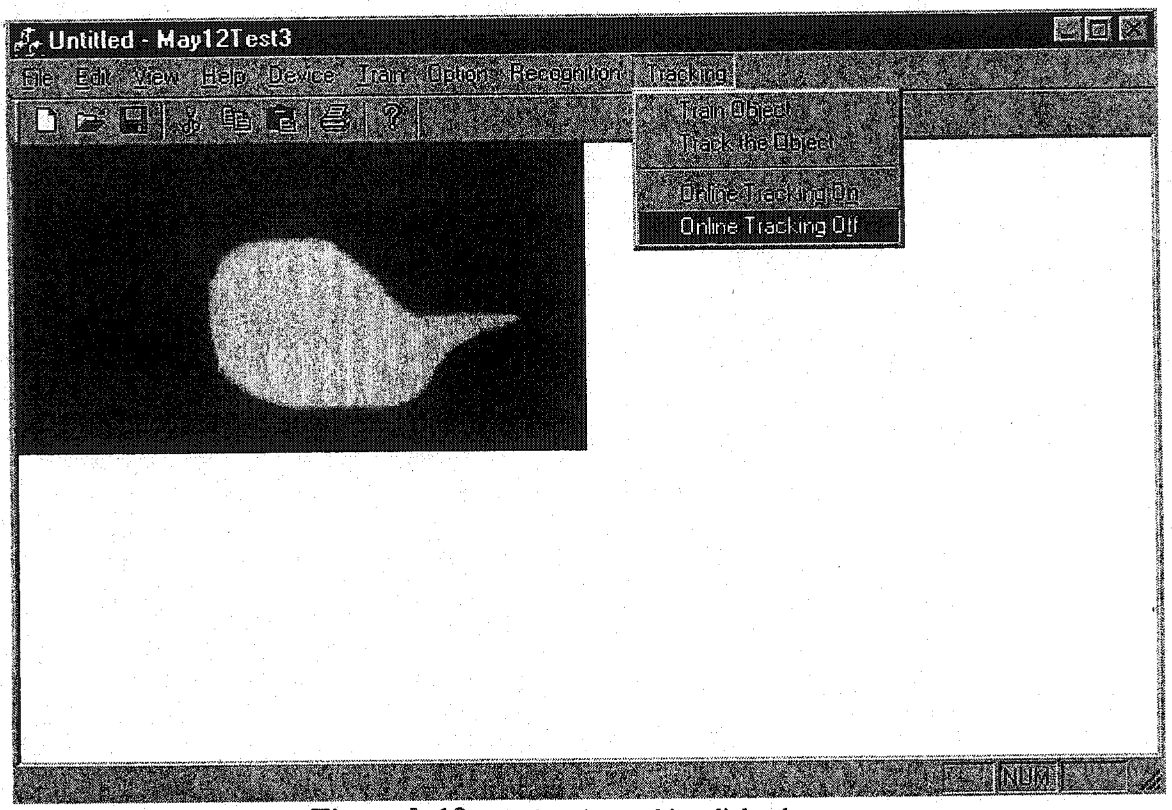


Figure b.10: Closing the tracking dialog box

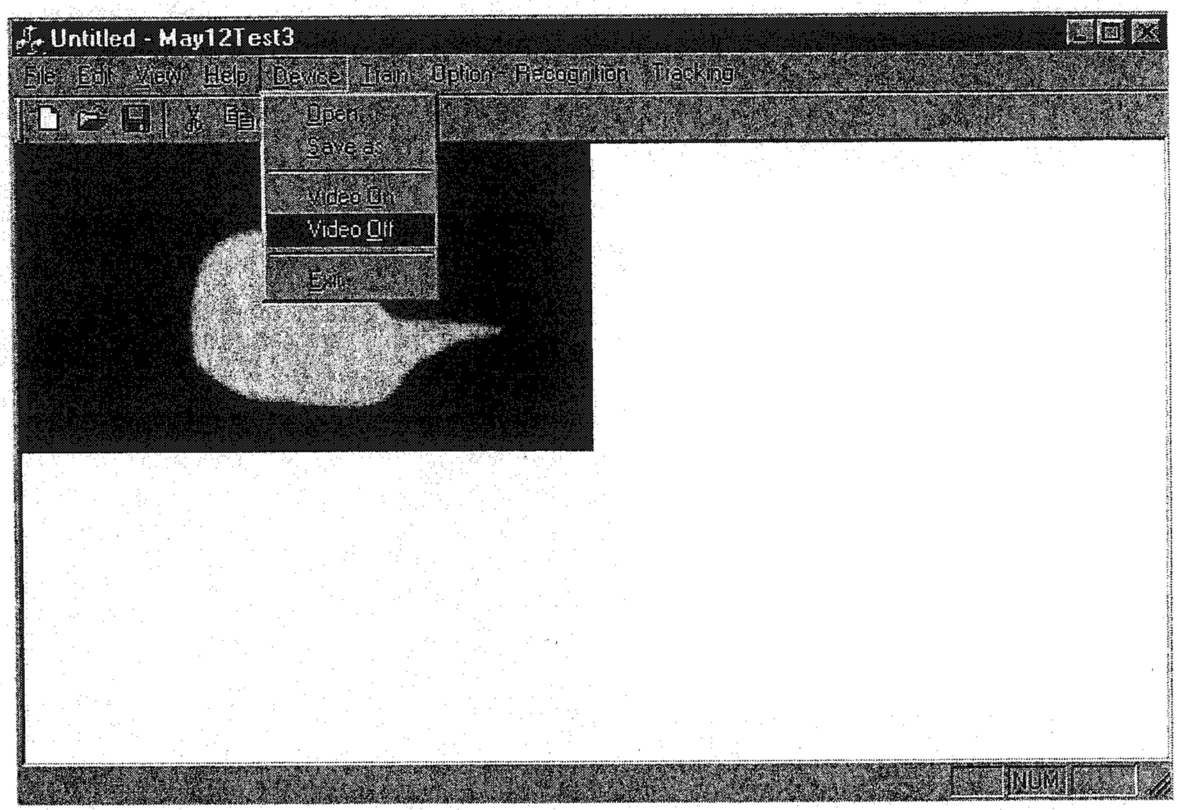


Figure b.11: Stop Asynchronous Capturing

BIBLIOGRAPHY

[1] A. D. Kulkarini, Al. C. Yap and P. Byars (1990). "Neural Networks for Invariant Object Recognition ", *IEEE Proceeding of Applied Computing*, pp.28-32.

[2] A. G. Bors and I. Pitas (1998). "Optical Flow Estimation and Moving Object Segmentation Based on Median Radial Basis Function Network", *IEEE Transaction on Image Processing*, Vol. 7, pp.693-702.

[3] A. p. Reeves (1981). "The General Theory of Moments for Shape Analysis and the Parallel Implementation of Moment Operation" , Purdu Univesity Tech. Rep. TR-EE 81-37.

[4] A. P. Reeves, R. J. Prokop, S. E. Andrews and F. P. Kuhl (1988). "Three-Dimensional Shape Analysis Using Moments and Fourier Descriptors", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 6, pp. 937-943.

[5] A. Pugh and E. Tal (1991). "A Research in Sensor Guide Assembly", *Computing and Control Engineering Journal,* Vol. 2, No. 1, pp: 13-21.

[6] B. Kosko (1992). *Neural Network and Fuzzy System*, New Jersey: prentice hall.

[7] C. Loughlin (1983). "Eye-in-hand Robot Vision Scores over Fixed Camera", *Sensor Review*, Vol.3, January.

[8] G. Bianco and A. Zelinsky (2000). "Mobile Robot Guidance while Operating with Visual Strategies ", *IEEE International Conference on robotics and Automation*, pp.3778-3783.

[9] G. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum, New York, 1981.

[10] G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty, and Information*, Prentice Hall, Englewood Cliffs, NJ, 1988.

[11] G. Neil and K. M. Curtis (1996). "Scale and Rotationally Invariant Object Recognition Using Fractal Transformation", *IEEE International Conference on Acoustic, Speech, and Signal Processing,* Vol.6, pp.3458-3461.

[12] H. C. Andersen, A. Lotfi, L. C. Wstphal, and J. S. Roger Jang (1998). "Coments on: Functional Equivalence Between Radial Basis Function Networks and Fuzzy Inference Systems", *IEEE Transactions on Neural Networks*, Vol. 9, No. 6, pp.1529-1531.

[13] H. Gao (2001). *"Algorithem Design For Smart Vision Sensors* ", Master Theses, University of Windsor

[14] H. Gao and Xiang Chen (2001). "Theoretic Design of a Smart Vision Sensor", *Proc. Canadian Conference on Electrical and Computer Engineering*, Vol, 2, pp.1223-1228.

[15] H. Gao, Xiang Chen and Zhang Ren (2002). "Algorithm Design for a position tracking sensor based on patter Recognition", *IECON02, Industrial Electronics Society, IEEE.*

[16] H. K. Kwan and L. Y. Cai (2000). "Supervised Fuzzy Inference Network for Invariant Pattern Recognition", *IEEE Midwest Symposium on Circuits and Systems*, pp.850-854.

[17] H. K. Kwan and Y. Cai (1993). " A Fuzzy Neural Network and its Application to Pattern Recogntion", *IEEE Transaction on Fuzzy Systems*, No.3, Vol.2, pp.185-193.

[18] I. Guyon, P. Alberecht, Y. l. Cun, Jdenker, and W. Hubbard (1991). "Design of a Neural Network Character Recognition for a Touch Terminal", *Pattern Recognition*, Vol. 24, No.2, pp.105-117.

[19] I. Konukseven, B. Kaftanoglu and T. Balkan (1997). "Multi sensor Controlled Robotic Tracking and Automatic Pick and Place", *IEEE, International Conference on Intelligent Robots and Systems*, Vol.3, pp. 1356-1362.

[20] J. Blais, V. Fischer and Y. Moalem (1998). "Correlation of Digital Image Matrices To Production ADC Matching Performance", *IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, pp. 86-92.

[21] J. Denzeler, R. Beb, J. Horengger, H. Niemann and D. Paulus (1994). "Learning, Tracking and Recognition of 3D Object", *IEEE International Conference on Intelligent Robots and Systems*, Vol. 1, pp. 89-96.

[22] J. Haddadnia, K. Faez and P. Moallem (2001). "Neural Network Based Faced Recognition with Moment Invariant", *IEEE, International Conference on Image Processing,* Vol. 1, pp.1018-1021.

[23] J. Haddadnia, M. Ahmadi and K, Faez (2002). "A Hybrid Learning RBF Neural Network for Human Face Recognition with Pseudo Zernike Moment Invariant", *IEEE, International Joint Conference on Neural Networks*, pp.11-16.

[24] J. S. Roger Jang (1993). "ANFIS: Adaptive-Network-Based Fuzzy Inference System", *IEEE Transaction on Systems, Man, and Cybernetics*, Vol. 23, No. 3, pp665-685.

[25] J. S. Roger Jang (1993). "Functional Equivalence Between Radial Basis Function Network and Fuzzy Inference Systems", *IEEE Transaction on Neural Networks*, Vol.4, No. 1, pp.156-159.

[26] J. S. Roger Jang (1997). *Neuro-Fuzzy and Soft Computing*, Prentice Hall, Inc.

[27] J. Yang (2002). *Design of Vision Tracking System*, Master Theses, University of Windsor.

[28] K. ArBter,E. Snyder and H. Burkhardt (1990). "Application of Affine-invariant Fourier Descriptors to Recognition of 3-D objects", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 6.

[29] L. A. Zadeh, "Fuzzy Sets", Infromation and Control, Vol. 8, 338-353, 1965.

[30] L. Austin, A. Turner, M. Turner, and K. Lees (1999). "Chemical Structure Matching Using Correlation Matrix Memories", *Proceedings IEE Conference on Artificial Neural Networks*, pp. 619-624.

[31] M. Hu (1962). " Visual pattern Recognition by Moment Invariant", *IEEE Transaction on Information Theory*, Vol. 8, Issue.2.

[32] M. Joo S. Wu, J. Lu and H. Lye (2002). " Face Recognition with Radial Basis Function Neural Networks", *IEEE Transaction on Neural Networks*, Vol. 13, No. 3, pp.697-710.

[33] N. Afzulpurkar, S. Plermkamon and M. Dange (2002). "Adaptive Linear Robot Control for Tracking and Grasping a Dynamic Object", *IEEE, International Conference on Industrial Technology*, Vol. 1, pp. 578-583.

[34] N. R. Pal,G. k. Mandal and E. V. Kumar (1999). "Comments on : a Fuzzy Neural Network and its Application to Pattern Recognition", *IEEE Transaction on Fuzzy System*, Vol. 7, No. 4, pp.479-480.

[35] P. Hecht-Nielsen (1990). *Neurocomputing*, Addison-Wesley Pub. Co.

[36] P. J. Pacini and B. Kosko (1992). "Adaptive Fuzzy Systems for Target Tracking", *IEE Intelligent Systems Engineering*, Vol.1, Issue: 1, pp. 3-21.

[37] P. W. Huang, S. K. Dai, and P.l. Line (1999). "Planar Shape Recognition By Directional Flow-change Method", *Pattern Recognition Letters*, Vol. 20, No.2, pp.163-170.

[38] P. Zhu and P. M. Chirlian (1995). "On Critical Point Detection of Digital Shapes", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 17 No. 8 pp. 737-748

[39] R. C. Gonzalez (2002). *Digital Image Processing*, Prentice-Hall.

[40] R. C. Luo, R. E. Mullen and D. E. Wessell (1988). "An Adaptive Robotic Tracking System", *IEEE International conference on Robotics and Automation*, Vol. 1, pp. 568-573.

[41] R. Koker, C. Oz and A. Ferikoglu (2001). "Development of a Vision Based Object Classification System for an Industrial Robotic Manipulator", *IEEE International Conference on Circuits and Systems*, Vol. 3, pp. 1281-1284.

[42] S. Fukami, M. Mizumoto, and k. Tanaka (1992). "Some Consideration on Fuzzy Conditional Inference", CRC Press, Inc., FL.

[43] S. Shahir and X. Chen (2003). "Adaptive Fuzzy Associative Memory for Online Quality Control", *IEEE SSST*, pp. 357-361.

[44] S. Shahir, X. Chen and M. Ahmadi (2003). "Fuzzy Associative Database for Multi Planar Object Recognition", *IEEE International Symposium on Circuits and System*, Vol. 5, pp. 805-808.

[45] T. H. Park and B. H. Lee (1997). "Dynamic Tracing Line: Feasible Tracking Region of a Robot in Conveyor Systems", *IEEE, Transaction on Man and Cybernetics*, Vol.27, No. 6,pp.1022-1030.

[46] T. H. Reiss (1992). "Features Invariant to Linear Transformations in 2D and 3D", *IEEE International Conference on Signal Analysis*, pp. 493-496.

[47] T. P. Wallace and P. Wintz (1980). "An Efficient, Three-dimensional Aircraft Recognition Algorithm Using Normalized Fourier Descriptors", *Computer Graphics Image processing*, Vol. 3, pp. 99-126.

[48] T. Takagi and M. Sugeno (1983). " Derivation of Fuzzy Control Rules from Human Operator's control actions", *Conference on Fuzzy information knowledge Representation and Decision Analysis*, pp. 55-60.

[49] T. Takagi and M. Sugeno (1985). "Fuzzy Identification o Systems and its Applications to Modeling and control", *IEEE Transaction on Systems, Man, and Cybernetic Vol. 15, pp.* 116-132.

[50] Y. Cai and H. K. Kwan (1995). "A Fuzzy Inference Network for Classification", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 5, pp. 3499-3502.

[51] Y. Li. (1991). "Application of Moment Invariants to Neurocomputing for pattern Recognition", *Electronics Letters*, Vol. 27, No. 7.

VITA AUCTORIS

Shahed Shahir, was born December 23, 1968 in historical city of Esfahan, Iran. He received his Bachelor of Science Degree in Electrical Engineering from K. N. T. University of Technology, Tehran in 1994. He has fulfilled two years of national service, working as an engineering consultant, and privately as a contractor with several companies. He has earned a computer-programming diploma from College of Advance Technology Seneca, Toronto in 2001. Shahed enrolled at the University of Windsor in September 2001 , and he graduated with a Master of Science degree in Electrical and Computer Engineering in Sep 2003. He has also been accepted at the Ph. D. program in Electrical and Computer Engineering at the University of Windsor.