OBJECT TRACKING IN VIDEO IMAGES BASED ON IMAGE SEGMENTATION AND PATTERN MATCHING

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

in

ELECTRONIC SYSTEMS AND COMMUNICATIONS

by

SANTHOSH KUMAR KADARLA

Department of Electrical Engineering
National Institute of Technology, Rourkela
2009

OBJECT TRACKING IN VIDEO IMAGES BASED ON IMAGE SEGMENTATION AND PATTERN MATCHING

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

in

ELECTRONIC SYSTEMS AND COMMUNICATION

by

SANTHOSH KUMAR KADARLA
ROLL NO: 207EE113

Under the Guidance of

PROF. DIPTI PATRA

Department of Electrical Engineering
National Institute of Technology, Rourkela
2009

National Institute Of Technology
Rourkela

CERTIFICATE

This is to certify that the thesis entitled, "**OBJECT TRACKING IN VIDEO IMAGES BASED ON IMAGE SEGMENTATION AND PATTERN MATCHING**" submitted by **Mr**. **SANTHOSH KUMAR KADARLA** in partial fulfillment of the requirements for the award of Master of Technology Degree in **ELECTRICAL ENGINEERING** with specialization in **"ELECTRONIC SYSTEMS AND COMMUNICATION"** at National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Date:

Prof. Dipti Patra
Department of Electrical Engineering
National Institute of Technology
Rourkela

ACKNOWLEDGEMENTS

This project is by far the most significant accomplishment in my life and it would be impossible without people who supported me and believed in me.

I would like to extend my gratitude and my sincere thanks to my honorable, esteemed supervisor ***Prof. Dipti Patra,*** *Department of Electrical Engineering. She is not only a great lecturer with deep vision but also most importantly a kind person. I sincerely thank for her exemplary guidance and encouragement. Her trust and support inspired me in the most important moments of making right decisions and I am glad to work under her supervision.*

I am very much thankful to our Head of the Department, ***Prof. Bidyadhar Subudhi****, for providing us with best facilities in the department and his timely suggestions and all my teachers* ***Dr.J.K.Sathapathy, Dr.P.K.Nanda****,* ***Dr.S.Das,*** *for providing a solid background for my studies. They have been great sources of inspiration to me and I thank them from the bottom of my heart.*

I would like to thank all my friends and especially my classmates for all the thoughtful and mind stimulating discussions we had, which prompted us to think beyond the obvious. I've enjoyed their companionship so much during my stay at NIT, Rourkela.

I would like to thank all those who made my stay in Rourkela an unforgettable and rewarding experience.

Last but not least I would like to thank my parents, who taught me the value of hard work by their own example. They rendered me enormous support being apart during the whole tenure of my stay in NIT Rourkela.

SANTHOSH KUMAR

CONTENTS

ABSTRACT

The moving object tracking in video pictures [1] has attracted a great deal of interest in computer vision. For ***object recognition***, ***navigation systems*** and ***surveillance systems*** [10], object tracking is an indispensable first-step. We propose a novel algorithm for object tracking in video pictures, based on ***image segmentation*** and ***pattern matching*** [1]. With the image segmentation, we can detect all objects in images no matter whether they are moving or not. Using image segmentation results of successive frames, we exploit pattern matching in a simple feature space for tracking of the objects. Consequently, the proposed algorithm can be applied to multiple moving and still objects even in the case of a moving camera. We describe the algorithm in detail and perform simulation experiments on object tracking which verify the tracking algorithm’s efficiency. VLSI implementation of the proposed algorithm is possible.

The conventional approach to object tracking is based on the difference between the current image and the background image. However, algorithms based on the difference image cannot simultaneously detect still objects. Furthermore, they cannot be applied to the case of a moving camera. Algorithms including the camera motion information have been proposed previously, but, they still contain problems in separating the information from the background.

The proposed algorithm, consisting of four stages i.e. ***image segmentation***, ***feature extraction*** as well as ***object tracking*** and ***motion vector determination*** [12]. Here Image Segmentation is done in 3 ways and the efficiency of the tracking is compared in these three ways, the segmentation techniques used are “***Fuzzy C means clustering using Particle Swarm Optimization*** [5],[6],[17]*”, ”**Otsu’s global thresholding*** [16]*”, ”**Histogram based thresholding by manual threshold selection**”*, after image segmentation the features of each object are taken and ***Pattern Matching*** [10],[11],[20] algorithm is run on consecutive frames of video sequence, so that the pattern of extracted features is matched in the next frame , the motion of the object from reference frame to present frame is calculated in both X and Y directions, the mask is moved in the image accordingly, hence the moving object in the video sequences will be tracked.

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Motivation:

The moving object tracking in video pictures has attracted a great deal of interest in computer vision. For ***object recognition, navigation systems*** and ***surveillance systems***, object tracking is an indispensable first step. Object tracking has significance in real time environment because it enables several important applications such as ***Security and surveillance*** [10] to recognize people, to provide better sense of security using visual information, In ***Medical therapy*** to improve the quality of life for physical therapy patients and disabled people, In Retail space instrumentation to analyze shopping behavior of customers to enhance building and environment design, Video abstraction to obtain ***automatic annotation of videos***, to generate object based summaries, Traffic management to analyze flow, to detect accidents, Video editing to eliminate cumbersome human operator interaction, to design futuristic video effects.

1.2 Literature review:

Tracking is a significant and difficult problem that arouses interest among computer vision researchers. The objective of tracking is to establish correspondence of objects and object parts between consecutive frames of video. It is a significant task in most of the surveillance applications since it provides cohesive temporal data about moving objects which are used both to enhance lower level processing such as motion segmentation and to enable higher level data extraction such as activity analysis and behavior recognition. Tracking has been a difficult task to apply in congested situations due to inaccurate segmentation of objects. Common problems of erroneous segmentation are long shadows, partial and full occlusion of objects with each other and with stationary items in the scene. Thus, dealing with shadows at motion detection level and coping with occlusions both at segmentation level and at tracking level is important for robust tracking. Tracking in video can be categorized according to the needs of the applications. It is used in or according to the methods used for its solution. Whole body tracking is generally adequate for outdoor video surveillance whereas objects' part tracking is necessary for some indoor surveillance and higher level behavior understanding applications.

There are two common approaches in tracking objects as a whole: one is based on correspondence matching and other one carries out explicit tracking by making use of position prediction or motion estimation. On the other hand, the methods that track parts of objects (generally humans) employ model-based schemes to locate and track body parts. Some example models are stick figure, Cardboard Model, 2D contour and 3D volumetric models combine motion estimation methods with correspondence matching to track objects. It is also able to track parts of people such as heads, hands, torso and feet by using the Cardboard Model which represents relative positions and sizes of body parts. It keeps appearance templates of individual objects to handle matching even in merge and split cases. Amer presents a non-linear voting based scheme for tracking objects as a whole. It integrates object features like size, shape, center of mass and motion by voting and decides final matching with object correspondence. This method can also detect object split and fusion and handle occlusions .The algorithm incorporates size and positions of objects for seeding and maintaining a set of Kalman filters for motion estimation. Also, Extended Kalman filters are used for trajectory prediction and occlusion handling in the work of Rosales and Sclaroff. As an example of model based body part tracking system, Pfinder makes use of a multi-class statistical model of color and shape to track head and hands of people in real-time.

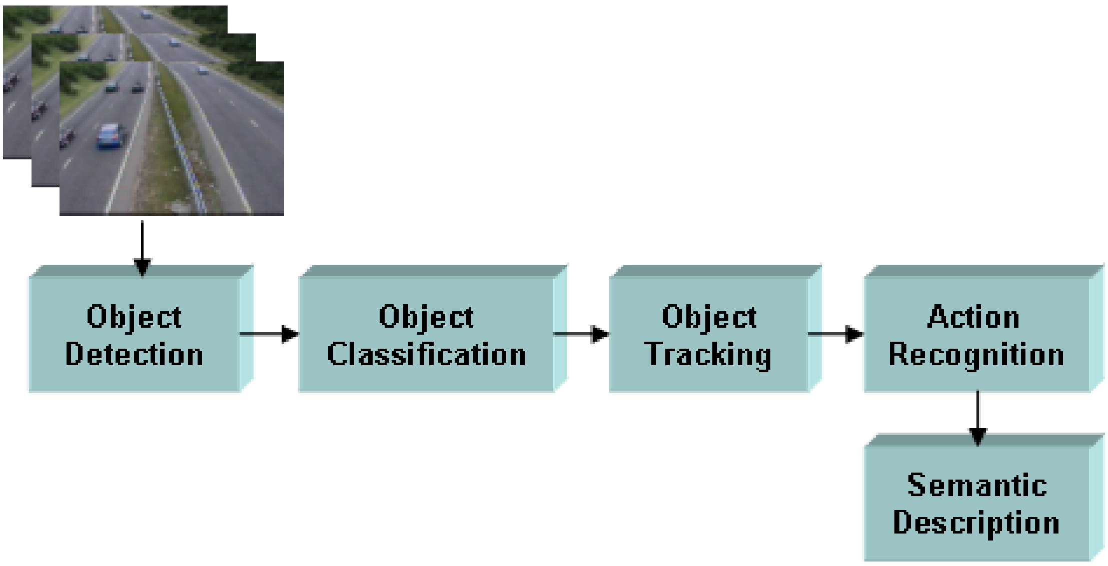


Fig.1. A generic framework for smart video processing algorithms

Moving object tracking [Fig.1] is the process of locating a moving object in time using a camera. An algorithm analyses the video frames and outputs the location of moving targets within the video frame. The main difficulty in video tracking is to associate target locations in consecutive video frames, especially when the objects are moving fast relative to the frame rate. Here, video tracking systems usually employ a motion model which describes how the image of the target might change for different possible motions of the object to track.

Examples of simple motion models are:

- To track planar objects, the motion model is a 2D transformation (affine transformation or homograph) of an image of the object.
- when the target is a rigid 3D object, the motion model defines its aspect depending on its 3D position and orientation
- For video compression, key frames are divided into macro blocks. The motion model is a disruption of a key frame, where each macro block is translated by a motion vector given by the motion parameters
- The image of deformable objects can be covered with a mesh, the motion of the object is defined by the position of the nodes of the mesh.

The role of the tracking algorithm [Fig.1] is to analyze the video frames in order to estimate the motion parameters. These parameters characterize the location of the target.

Conventional approaches:

There are two major components of a visual tracking system; Target Representation and Localization and Filtering and Data Association. Target Representation and Localization is mostly a bottom-up process. Typically the computational complexity for these algorithms is low. The following are some common Target Representation and Localization algorithms:

- **Blob tracking**: Segmentation of object interior (blob detection, block-based correlation)
- **Kernel-based tracking** (Mean-shift tracking): An iterative localization procedure based on the maximization of a similarity measure (Bhattacharyya coefficient).
- **Contour tracking**: Detection of object boundary.

Filtering and Data Association is mostly a top-down process, which involves incorporating prior information about the scene or object, dealing with object dynamics, and evaluation of different hypotheses. The computational complexity for these algorithms is usually much higher. The following are some common Filtering and Data Association algorithms:

- **Kalman filter**: An optimal recursive Bayesian filter for linear functions subjected to Gaussian noise.
- **Particle filter** [9]: Useful for sampling the underlying state-space distribution of non-linear and non-Gaussian processes.

Object tracking is an important task within the field of computer vision. The proliferation of high-powered computers and the increasing need for automated video analysis has generated a great deal of interest in object tracking algorithms. In general it is used mostly in surveillance systems. There are three key steps in video surveillance: detection of interesting moving objects [12], tracking of such objects from frame to frame, and analysis of object tracks to recognize their behavior. In this report we present a feature based object tracker which uses a pan-tilt (PT) camera to keep track of the target. Generally, the task is to keep the target at the center of the grabbed image. As the target starts moving in the real world, its position in the grabbed image is reported in subsequent frames through a feature based tracking algorithm, based on the work of Lukas, kanade and Tomasi. The image position error is processed by a proportional-integral controller and the camera is re-positioned accordingly to place the target in the pre-specified image region.

1.3 The video-object segmentation problem:

The task of video-object segmentation is to identify and separate the important objects in a video scene from the scene background. Clearly, to approach this problem, it is necessary to define what is exactly meant with important objects and how the correct object masks should look like. However, in practice, it turns out that even an unambiguous definition of video objects is a fundamental problem. In the following, the involved definition problems are addressed and grouped into physical problems, being a consequence of the image formation, and semantic problems. The physical problems are as follows

• Reflections: The problem of handling reflections is actually similar to object shadows. However, reflections are more difficult, because the appearance of the reflected images depends on the physical properties of the reflecting surface and because the reflection is not necessarily attached to the object.

• Occlusions: The object shape can also change because of occlusions. It depends on the application whether the masks of occluded objects should be extended to their original shape.

• Translucent objects: Objects can appear partially translucent since they are made of translucent materials, or because thin structures like hair or cloth appear translucent. Moreover, pixels along object boundaries are always a mixture of foreground color and background color. To model the translucency, the segmentation algorithm has to compute an alpha-channel mask which identifies the translucency factor for each pixel instead of only computing a binary object mask. Accurate alpha-channel information cannot be obtained from a single image, but algorithms using heuristic approaches have been proposed.

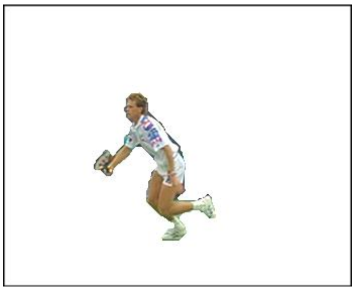


Fig.2.a. Input image Fig.2.b.Background image Fig.2.c.Segmented foreground

Apart from the physical problems, there are semantic definition problems, like the following.

• Objects of interest (foreground objects): The first and obvious question of video segmentation is what parts of an image constitute the foreground objects [Fig.2c]. This issue is already surprisingly difficult, since the intuitive human understanding of foreground objects is strongly depending on the scene context. Mostly, human intuition expects that this should be the main acting objects. For example, in sports broadcast, the players are usually considered foreground and the audience is considered background, even if the audience is moving. This distinction is on a very high semantic level, since it assumes knowledge about the meaning of the scene. Note that the object definition can also vary with the application. A surveillance system in a sports stadium will be interested in other objects than a system for automatic analysis of the sports game.

• Small background movements: When taking a more detailed view on the last point, it can be observed that the distinction between foreground [Fig.2c] and background [Fig.2b] is in fact gradual. The question is to what extent a background should change such that it is considered part of the foreground. For example, trees may occur in the background with leaves moving slightly in the wind, or there may be a clock on a wall at the back of the room.

• Object-status change: Objects can also change their classification over time. For example, most people would consider a car that drives along a street as an important object. But how to define the object status when the car stops and parks at the side of the street? Alternatively, the opposite case may occur that a car that was parked for a long time suddenly drives away.

Note that it is practically impossible to separate all objects, including the static ones, into independent objects, since this would imply that all future actions would have to be predicted.

• Multi-body objects: Objects may be separated into several nonconnected regions in the image. One reason for this can be that an occluding object cuts the foreground object into pieces. Another complex example are objects that are really composed of several parts but still belonging together like flocking birds.

• Hierarchical objects: Additional to multi-body objects, there can also exist a hierarchical relationship between objects. One example is a car object that contains a driver object. When considering all of these problems simultaneously, it can only be concluded that a general-purpose segmentation of video objects is virtually impossible, since the definition of the expected output from the algorithm depends largely on the scene context and the application that we have in mind. However, despite all the mentioned problems, it is still possible to design algorithms that cover a multitude of specific applications and that work well in many practical cases.

1.4 Proposed approach [1]:

- **Image segmentation and Pattern matching**: The frames extracted from the video are segmented first, features of each object in the segmented image are extracted, pattern matching is done on the consecutive frames having the desired features in the hand, the motion vectors are calculated and mask is moved accordingly.

Block diagram:

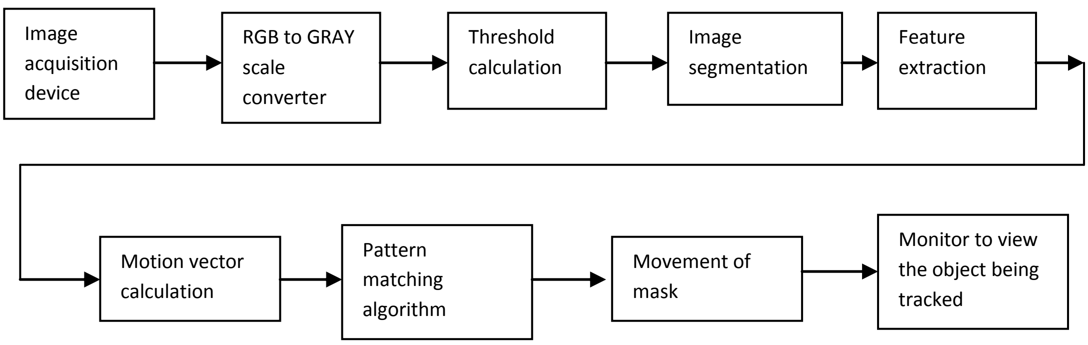


Fig.3. Block diagram of proposed method of object tracking

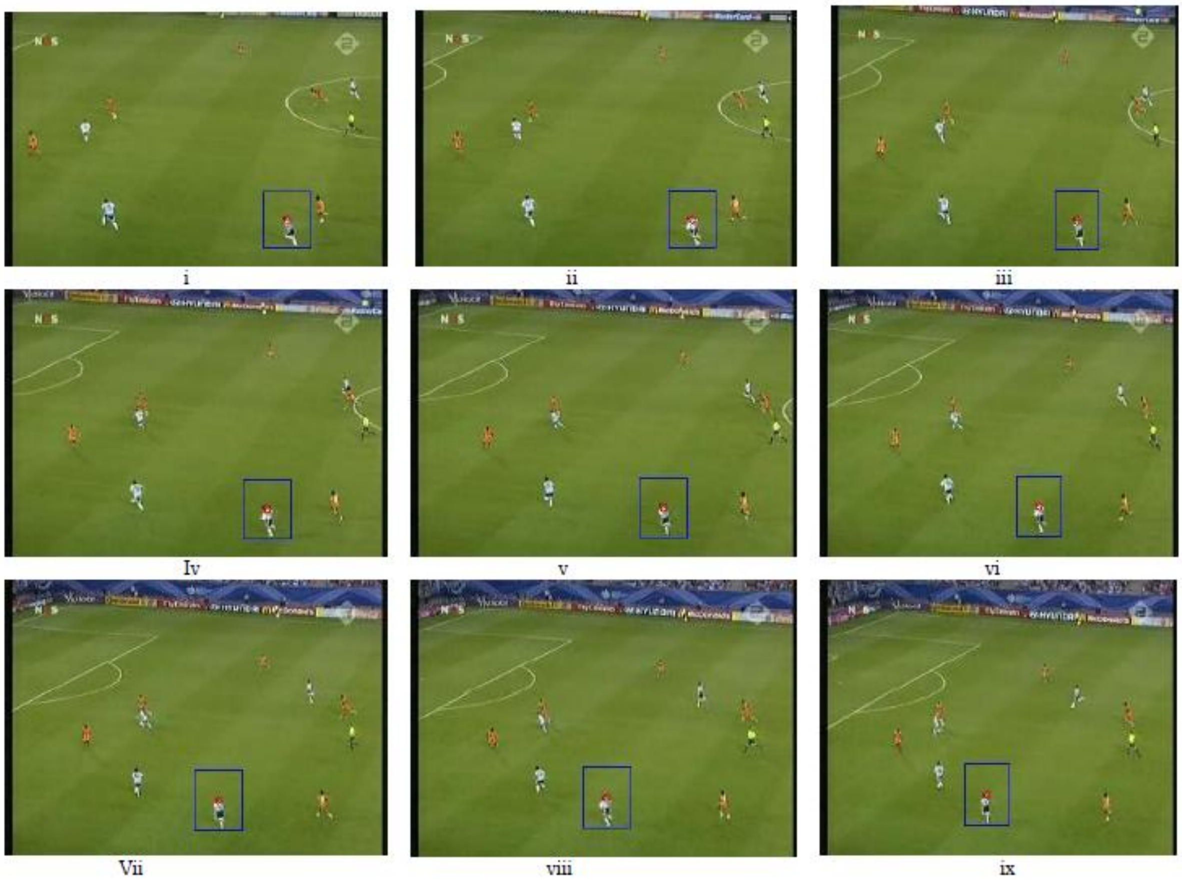


Fig.3. Illustration of moving object tracking in video sequences.

1.5 Organization of thesis:

In chapter 1, it is all discussed about the applications of Object tracking, where it is used in real time environment [8], significance of object tracking in various fields like computer vision, robotics, artificial intelligent systems, surveillance systems, navigation systems, traffic monitoring systems in heavy crowded urban areas and it is also discussed about the conventional approaches for object tracking and proposed algorithm also, different steps [Fig.3] in object tracking is also shown in this chapter.

In chapter 2, it is mainly concentrated on Image Segmentation using Otsu’s global thresholding [14], [15] in partitioned windows, basic concepts in segmentation of an image and several types of image segmentation techniques invented so far are briefly explained. The formulation of Otsu’s threshold calculation using mean, variances of 2 set of classes is explained, the important aspects regarding the threshold calculation is also discussed.

Image segmentation using Fuzzy C means clustering [4] is discussed in chapter 3, the clustering analysis, applications of clustering, advantages of FCM over other segmentation techniques, Problem formulation of FCM clustering technique, a detailed description of FCM algorithm is given here.

In chapter 4, it is discussed about the evolutionary computing techniques like particle swarm optimization [19], notion of particle swarm optimization, and different aspects of PSO invented by well known scientists. Description of particle swarm optimization algorithm, velocity, position updation rules. Here the proposed approach, i.e. how the particle swarm optimization technique can be incorporated to the FCM is illustrated.

In chapter 5, pattern recognition [20] technique is described in detail. It's applications in various fields, basic concepts hidden in pattern recognition like clustering techniques [7], cluster analysis, feature extraction, feature matching, flow chart of a typical pattern recognition system are given. How a pattern recognition system can be used for Image processing applications, is also discussed. Motion vector determination using feature points in an image and Pattern recognition algorithm for moving object tracking are also described.

All Simulation results are explained in this chapter 6, Object tracked images in two video sequences are shown, here the two video sequences i) ball video sequence (320 X 240) ii) Tennis video sequence are taken (352 X 220), we have done the object tracking in 3 ways

1. Histogram based segmentation and pattern matching
2. Otsu's global thresholding and pattern matching
3. Fuzzy C means clustering using Particle swarm optimization and pattern matching

The three techniques are repeated for both the video sequences and the results are compared in both subjective quality as well as objective quality.

Chapter 2

Image Segmentation

2.1 Introduction:

In designing automated systems for the interpretation or manipulation of image data, system developers often need to perform software imaging operations, called segmentation [Fig.5b,6b], that extract information about the structure of objects and to separate and discern various parameters of interest within the data. Measurements or attributes of these objects, known as features, can then be calculated and used for defect inspection, quality control, or clinical qualitative analysis. Accordingly, common vision processes deal with the identification of discrete objects within an image. Such processes transform single-pixel representations of the image data into geometric descriptors representing groups of pixel elements. These descriptors, known as objects, take the form of points, lines, regions, polygons, or other unique representations.

Segmentation techniques are divided into two basic categories: edge-based and region-based. Edge-based segmentation is primarily used to look for image discontinuities. The technique is generally applied where changes of gray-level intensity occur in the image. The assumption is that changes occur in the data at the boundary between objects of interest. The output of edge-segmentation schemes can be x and y gradient two images are used to represent the edges found, one in the x direction and one in the y direction .

1. Gradient strength and direction
2. Binary edge map
3. Edge representation.

In contrast, region-based segmentation is used to look for similarities between adjacent pixels. That is, pixels that possess similar attributes are grouped into unique regions. The assumption is made that each region represents one object of interest. Using gray-level intensity is the most common means of assigning similarity, but many other possibilities exist, such as variance, color, and multispectral features.

Most commercial vision systems use region-based segmentation schemes based on pixel-intensity values. These segmentation techniques assume that the objects of interest possess uniform shading and that a significant and constant gray-level change occurs between the objects of interest and the background. However, in many vision applications, these assumptions have proven erroneous. Therefore, these techniques are considered fragile and commonly require controlled conditions or human supervision.

Effects of uneven sample illumination, shadowing, partial occlusion, clutter, noise, and subtle object-to-background changes can all contribute to errors in basic segmentation processes. They generally result in false segmentations of the background, partial segmentations of the objects of interest, clumping of objects, or inadequate segmentations. Errors in the segmentation of the data can also result in the calculation of erroneous features. Therefore, it is essential that the segmentation method chosen support the final processing goals of the vision system.

Thresholding [15] is perhaps the most common segmentation technique and is the most basic region-segmentation technique. The technique separates pixels into background and foreground (object of interest) classes based upon their similarities in gray-level intensity. To implement this technique, a threshold (T) value is chosen. Every pixel in the image is then compared to the T value. Each pixel is given a region label of "0" (background) if the pixel value is less than or equal to T or "1" (foreground) if greater than T. This form of region segmentation results in a binary image, in which each region is either white (1) or black (0). Many variations exist within the general concept of segmentation by thresholding, which will be discussed in a future column.

Image segmentation is one of the fundamental problems in image processing and computer vision. Segmentation is also one of the first steps in many image analysis tasks. Image understanding systems such as face or object recognition often assume that the objects of interests are well segmented. Different visual cues, such as color, texture, and motion, help in achieving segmentation. Segmentation is also goal dependent, subjective, and hence ill-posed in a general set up. However, it is desirable to consider generic criteria that can be applied to a large variety of images and can be adapted for specific applications.

Fig.5a. Input image

Fig.6b.Segmented image

Fig.6a. Input image

Fig.6b.Segmented image

2.2 Various types of Image Segmentation Techniques:

Clustering methods: The K-means algorithm is an iterative technique that is used to partition an image into *K* clusters. The basic algorithm is:

1. Pick *K* cluster centers, either randomly or based on some heuristic
2. Assign each pixel in the image to the cluster that minimizes the variance between the pixel and the cluster center
3. Re-compute the cluster centers by averaging all of the pixels in the cluster
4. Repeat steps 2 and 3 until convergence is attained (i.e. no pixels change clusters)

In this case, variance is the squared or absolute difference between a pixel and a cluster center. The difference is typically based on pixel color, intensity, texture, and location, or a weighted combination of these factors. *K* can be selected manually, randomly, or by a heuristic. This algorithm is guaranteed to converge, but it may not return the optimal solution. The quality of the solution depends on the initial set of clusters and the value of *K*.

In statistics and machine learning, the k-means algorithm is clustering algorithm to partition n objects into k clusters, where k < n. It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data. The model requires that the object attributes correspond to elements of a vector space. The objective it tries to achieve is to minimize total intra-cluster variance, or, the squared error function. The k-means clustering was invented in 1956. The most common form of the algorithm uses an iterative refinement heuristic known as Lloyd's algorithm. Lloyd's algorithm starts by partitioning the input points into k initial sets, either at random or using some heuristic data. It then calculates the mean point, or centroid, of each set. It constructs a new partition by associating each point with the closest centroid. Then the centroids are recalculated for the new clusters, and algorithm repeated by alternate application of these two steps until convergence, which is obtained when the points no longer switch clusters (or) alternatively centroids are no longer changed.

Histogram-based methods:

Histogram-based [14] methods are very efficient when compared to other image segmentation methods because they typically require only one pass through the pixels. In this technique, a histogram is computed from all of the pixels in the image, and the peaks and valleys in the histogram are used to locate the clusters in the image. Color or intensity can be used as the measure. A refinement of this technique is to recursively apply the histogram-seeking method to clusters in the image in order to divide them into smaller clusters. This is repeated with smaller and smaller clusters until no more clusters are formed. One disadvantage of the histogram-seeking method is that it may be difficult to identify significant peaks and valleys in the image. In this technique of image classification distance metric and integrated region matching are familiar.

Region growing methods:

The first region growing method was the seeded region growing method [Fig.7]. This method takes a set of seeds as input along with the image. The seeds mark each of the objects to be segmented. The regions are iteratively grown by comparing all unallocated neighbouring pixels to the regions. The difference between a pixel's intensity value and the region's mean, δ, is used as a measure of similarity. The pixel with the smallest difference measured this way is allocated to the respective region. This process continues until all pixels are allocated to a region. Seeded region growing requires seeds as additional input. The segmentation results are dependent on the choice of seeds. Noise in the image can cause the seeds to be poorly placed. Unseeded region growing is a modified algorithm that doesn't require explicit seeds.

Flow chart of Region growing algorithm:

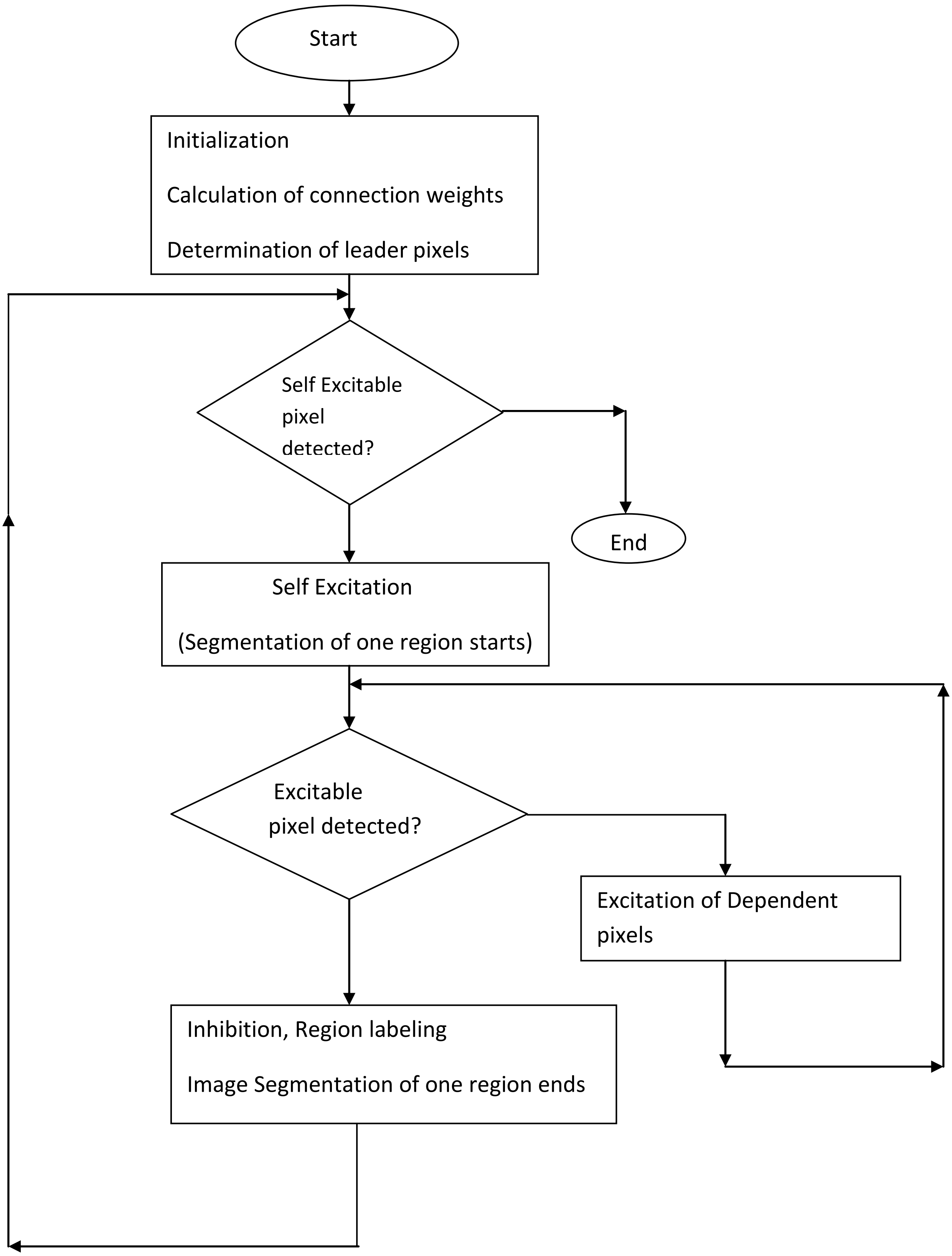


Fig.7.Flow chart of seeded region growing algorithm

It starts off with a single region A_1, the pixel chosen here does not significantly influence final segmentation. At each iteration, it considers the neighbouring pixels in the same way as seeded region growing. It differs from seeded region growing in that if the minimum δ is less than a predefined threshold T then it is added to the respective region A_j. If not, then the pixel is considered significantly different from all current regions A_i and a new region A_{n+1} is created with this pixel.

Region growing is one of the simplest region-based image segmentation [Fig.7] methods and it can also be classified as one of the pixel-based image segmentations because it involves the selection of initial seed points. This approach to segmentation examines the neighboring pixels of the initial "seed points" and determines if the pixel should be added to the seed point or not. The process is iterated as same as data clustering.

Edge detection methods:

Edge detection is a well-developed field on its own within image processing. Region boundaries and edges are closely related, since there is often a sharp adjustment in intensity at the region boundaries. Edge detection techniques have therefore been used as the base of another segmentation technique. The edges identified by edge detection are often disconnected. To segment an object from an image however, one needs closed region boundaries. Discontinuities are bridged if the distance between the two edges is within some predetermined threshold. One such method is the edge linking method, proposed by Pathegama and Gol.

Watershed transformation:

The watershed transformation considers the gradient magnitude of an image as a topographic surface. Pixels having the highest gradient magnitude intensities (GMIs) correspond to watershed lines, which represent the region boundaries. Water placed on any pixel enclosed by a common watershed line flows downhill to a common local intensity minimum (LMI). Pixels draining to a common minimum form a catch basin, which represents a segment.

Graph partitioning methods:

Graphs can effectively be used for image segmentation. Usually a pixel or a group of pixels are vertices and edges define the similarity among the neighborhood pixels. Some popular algorithms of this category are random walker, minimum mean cut, minimum spanning treebased algorithm, normalized cut, etc. The *normalized cuts* method was first proposed by Shi and Malik in 1997.In this method, the image being segmented is modeled as a weighted, undirected graph. Each pixel is a node in the graph, and an edge is formed between every pair of pixels. The weight of an edge is a measure of the similarity between the pixels. The image is partitioned into disjoint sets (segments) by removing the edges connecting the segments. The optimal partitioning of the graph is the one that minimizes the weights of the edges that were removed (the *cut*). Shi's algorithm seeks to minimize the normalized cut, which is the ratio of the cut to all of the edges in the set.

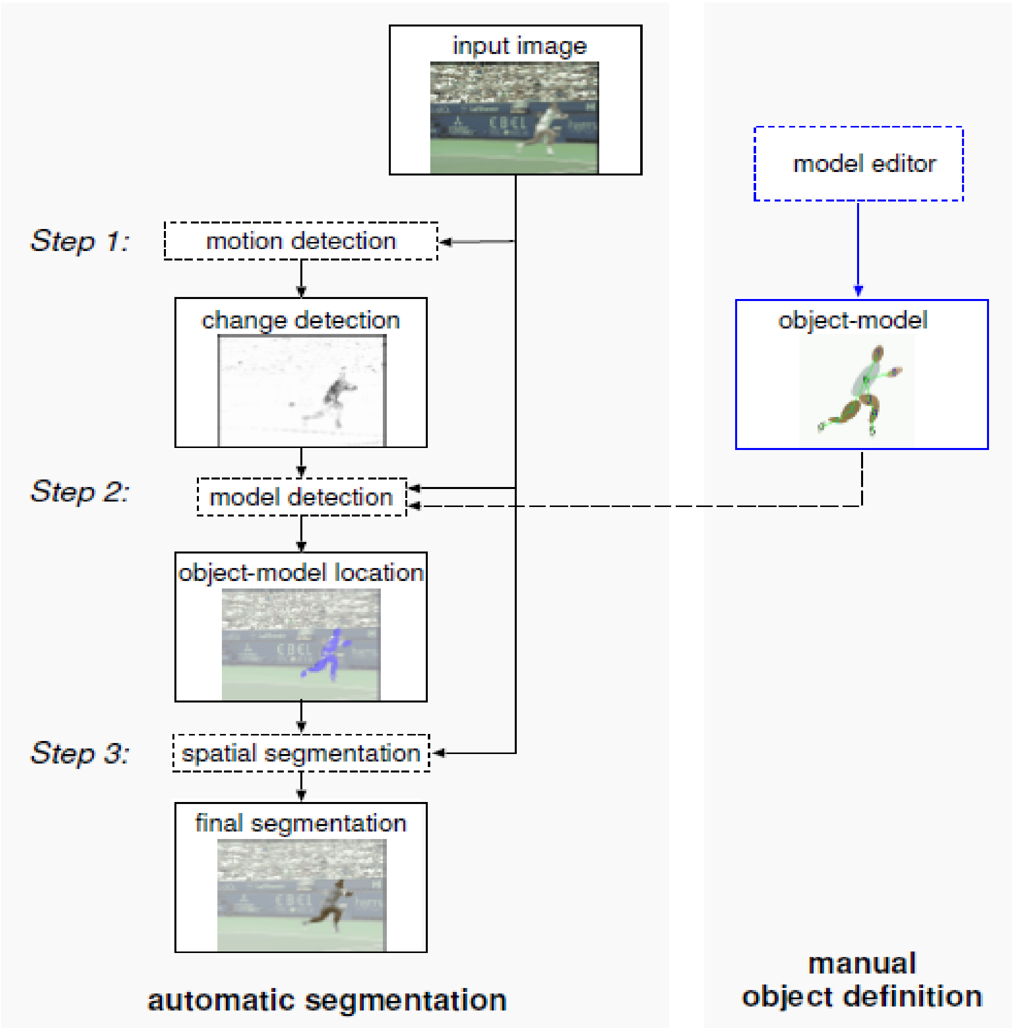


Fig.8. Image segmentation using graph partitioning method

Model based segmentation:

The central assumption of such an approach is that structures of interest/organs have a repetitive form of geometry. Therefore, one can seek for a probabilistic model towards explaining the variation of the shape of the organ and then when segmenting an image impose constraints using this model as prior. Such a task involves

1. Registration of the training examples to a common pose
2. Probabilistic representation of the variation of the registered samples
3. Statistical inference between the model and the image.

State of the art methods in the literature for knowledge-based segmentation involve active shape and appearance models, active contours and deformable templates and level-set based methods.

2.3 Global thresholding approach:

Thresholding:

Thresholding is one of the most powerful and important tools for image segmentation. The segmented image obtained from thresholding has the advantages of smaller storage space, fast processing speed and ease in manipulation compared with gray level image which usually contains 256 levels. The thresholding techniques, which can be divided into bi-level and multilevel category. In bi-level thresholding [Fig.9a], a threshold is determined to segment the image into two brightness regions which correspond to background and object. Several methods have been proposed to automatically select the threshold. Otsu et.al formulates the threshold selection problem as a discriminant analysis where the gray level histogram of image is divided into two groups and the threshold is determined when the variance between the two groups is the maximum. Even in the case of unimodal histogram images, that is, the histogram of a gray level image does not have two obvious peaks, Otsu's method can still provide satisfactory result. Therefore, it is referred to as one of the most powerful methods for bi-level thresholding. In multilevel thresholding [Fig.9b], more than one threshold will be determined to segment the image into certain brightness regions which correspond to one background and several objects.

The selection of a threshold will affect both the accuracy and the efficiency of the subsequent analysis of the segmented image. The principal assumption behind the approach is that the object and the background can be distinguished by comparing their gray level values with a suitably selected threshold value. If background lighting is arranged so as to be fairly uniform, and the object is rather flat that can be silhouetted against a contrasting background, segmentation can be achieved simply by thresholding the image at a particular intensity level.

The simplicity and speed of the thresholding algorithm make it one of the most widely used algorithms in automated systems ranging from medical applications to industrial manufacturing. The binarized image is especially suitable as the input for hardware implementation of template matching through correlation and moment based recognition. Besides the application of thresholding in image segmentation, it is also used in various classification problems in pattern recognition.

Suppose that the gray-level histogram shown in Fig 9(a) corresponds to an image, f(x; y), composed of light objects on a dark background, in such a way that object and background pixels have gray levels grouped into two dominant modes. One obvious way to extract the objects from the background is to select a threshold T that separates these modes. Then any (x; y) for which f(x; y) > T is called an *object point*; otherwise, the point is called a *background point*. Fig.9 (b) shows a slightly more general case of this approach, where three dominant modes characterize the image histogram (for example, two types of light objects on a dark background).

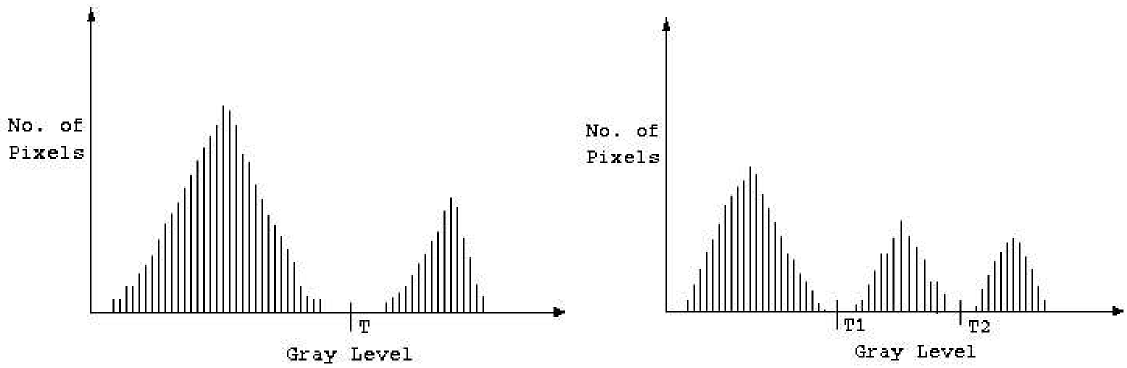


Fig.9. Gray-level histograms that can be partitioned by (a) a single threshold, and (b) multiple thresholds

Here, multilevel thresholding classifies a point (x; y) as belonging to one object class if T1 < f(x; y) ≤ T2, to the other object class if f(x; y) > T2, and to the background if f(x; y) ≤T1. Based on the preceding discussion, thresholding may be viewed as an operation that involves tests against a function T of the form

$$T = T[x; y; p(x; y); f(x; y)]$$

where f(x; y) is the gray level of point (x; y) and p(x; y) denotes some local property of this point-for example, the average gray level of a neighborhood centered on (x; y). A threshold image g(x; y) is defined as

$$g(x,y)=\begin{cases}0-if-f(x,y)\le T\\1-if-f(x,y)>T\end{cases}$$

Thus, pixels labeled 1 (or any other convenient gray level) correspond to objects, whereas pixels labeled 0 (or any other gray level not assigned to objects) correspond to the background. When T depends only on f(x; y) (that is, only on gray-level values) the threshold is called *global*. If T depends on f(x; y) and p(x; y), the threshold is called *local*. If, in addition, T depends on the spatial coordinates x and y, the threshold is called *dynamic* or *adaptive*.

One simple technique for finding a suitable threshold arises in situation where the proportion of the background that is occupied by objects is substantially constant in a variety of conditions. The technique that is most frequently employed for determining thresholds involves analyzing the histogram of intensity levels in the gray image.

This method is subject to the following major difficulties:

1. The valley may be so broad that it is difficult to locate a significant minimum.
2. There may be a number of minima because of the type of detail in the image, and selecting the most significant one will be difficult.
3. Noise within the valley may inhibit location of the optimum position.
4. There may be no clearly visible valley in the distribution because noise may be excessive or because the background lighting may vary appreciably over the image.
5. Either of the major peaks in the histogram (usually that dye to the background) may be much larger than the older, and this will then bias the position of the minimum.
6. The histogram may be inherently multimodal, making it difficult to determine which the relevant thresholding level is.

Otsu's global thresholding method:

This method is a nonparametric and unsupervised method of automatic threshold selection for image segmentation. An optimal threshold is calculated by the discriminant criterion, namely, so as to maximize the between-class variance or to minimize the within-class variance. The method is very simple, utilizing only the zeroth and first order cumulative moments of the gray level histogram.

Let the pixels of a given image represented in L gray levels [1; 2; :::; L]. The number of pixels at level i is denoted by n_i and the total number of pixels by $N = n_1+n_2+:::+n_L$. For simplification, the gray-level histogram is normalized and regarded as a probability distribution.

$$P_i = \frac{n_i}{N}, P_i \geq 0, \sum_{i=1}^{L} P_i = 1 \quad .....(1)$$

To emphasize the partitioned windows technique, only Otsu's thresholding method is considered among many other techniques. This method can be stated as follows: For a given image *f(x,y)* with m gray levels *0, 1, . . . ,m-1,* let the threshold be j, where *0< j< m-1*. Then, all pixels in image *f(x,y)* can be divided into two groups: group A with gray level values of pixels less than or equal to *j*; and group B with values greater than *j*. Also, let *(w_1(j), M_1(j)),(Eqn.2,3) (w_2(j),M_2(j))(Eqn.4,5)* be the number of pixels and the average gray level value in group A and group B, respectively. Then

$$\omega_1(j) = \sum_{i=0}^{j} n_i, 0 \leq j \leq m-1 \quad ....(2)$$

$$M_1(j) = \frac{\sum_{i=0}^{j}(i.n_i)}{\omega_1(j)}, 0 \leq j \leq m-1 \quad ....(3)$$

$$\omega_2(j) = \sum_{i=j+1}^{m-1} n_i, 0 \leq j \leq m-1 \quad ....(4)$$

$$M_2(j) = \frac{\sum_{i=j+1}^{m-1}(i.n_i)}{\omega_2(j)}, 0 \leq j \leq m-1 \quad ......(5)$$

where n_i is the number of pixels with gray level value *i*. Expressing the average gray level value M_T (Eqn.6)of all the pixels in image *f(x,y)* as

$$M_T = \frac{\omega_1(j)M_1(j) + \omega_2(j)M_2(j)}{\omega_1(j) + \omega_2(j)}, 0 \leq j \leq m-1 \quad ....(6)$$

the variance between the two groups, denoted as $\sigma_B^2(j)$,is

$$\sigma_B^2(j) = \omega_1(j)(M_1(j) - M_T)^2 + \omega_2(j)(M_2(j) - M_T)^2 \quad \text{…………(7)}$$

$$= \frac{\omega_1(j)\omega_2(j)(M_1(j) - M_2(j))^2}{\omega_1(j) + \omega_2(j)} \quad \text{………….(8)}$$

For j ranging from 0 to m-1, calculate each $\sigma_B^2(j)$ Using above Eqn(7,8), and the value j corresponding to the greatest $\sigma_B^2(j)$ is the resulting threshold T.

2.4 Applications:

- Medical Imaging

  1. Locate tumors and other pathologies
  2. Measure tissue volumes
  3. Computer-guided surgery
  4. Diagnosis
  5. Treatment planning
  6. Study of anatomical structure

- Locate objects in satellite images (roads, forests, etc.)
- Face recognition
- Fingerprint recognition
- Traffic control systems
- Machine vision

Chapter 3

Image segmentation using Fuzzy C Means Clustering

3.1 Introduction:

Cluster analysis or clustering [7],[16] is the assignment of objects into groups (called *clusters*) so that objects from the same cluster are more similar to each other than objects from different clusters. Often similarity is assessed according to a distance measure. Clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics. Data clustering is the process of dividing data elements into classes or clusters so that items in the same class are as similar as possible, and items in different classes are as dissimilar as possible. Depending on the nature of the data and the purpose for which clustering is being used, different measures of similarity may be used to place items into classes, where the similarity measure controls how the clusters are formed. Some examples of measures that can be used as in clustering include distance, connectivity, and intensity.

In *hard clustering,* data is divided into distinct clusters, where each data element belongs to exactly one cluster. In *fuzzy clustering,* data elements can belong to more than one cluster, and associated with each element is a set of membership levels. These indicate the strength of the association between that data element and a particular cluster. Fuzzy clustering is a process of assigning these membership levels, and then using them to assign data elements to one or more clusters.

One of the most widely used fuzzy clustering algorithms is the Fuzzy C-Means (FCM) Algorithm. The FCM algorithm attempts to partition a finite collection of elements $X = \{x_1 . . . , x_n\}$ into a collection of c fuzzy clusters with respect to some given criterion. Given a finite set of data, the algorithm returns a list of c cluster centers $C = \{c_1 . . . , c_c\}$ and a partition matrix $U = u_{ij}$ lies between (0, 1), where $i = 1, . . . , n$, $j = 1, . . . , c$, where each element tells the degree to which element x_i belongs to cluster c_j . Like the k-means algorithm, the FCM aims to minimize an objective function. The standard function which differs from the k-means objective function by the addition of the membership values u_{ij} and the fuzzifier m. The fuzzifier m determines the level of cluster fuzziness. A large m results in smaller memberships $u_{i,j}$ and hence, fuzzier clusters. In the limit $m = 1$, the memberships $u_{i,j}$ converge to 0 or 1, which implies a crisp partitioning. In the absence of experimentation or domain knowledge, m is commonly set to 2. The basic FCM Algorithm, given n data points $(x_1, . . . , x_n)$ to be clustered, a number of c clusters with $(c_1, . . . , c_c)$ the center of the clusters, and m the level of cluster fuzziness with.

Types of clustering:

- Hierarchical clustering
- Partitional clustering
    - K-means and derivatives
        - k-means clustering
        - Fuzzy c-means clustering
        - QT clustering algorithm
    - Locality-sensitive hashing
    - Graph-theoretic methods
- Spectral clustering

Data clustering algorithms can be hierarchical. Hierarchical algorithms find successive clusters using previously established clusters. These algorithms can be either agglomerative ("bottom-up") or divisive ("top-down"). Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters. Partitional algorithms typically determine all clusters at once, but can also be used as divisive algorithms in the hierarchical clustering.

Determining the number of clusters in a data set, a quantity often labeled k, is fundamental to the problem of data clustering, and is a distinct issue from the process of actually solving the clustering problem. In most cases, k must be chosen somehow and specified as an input parameter to clustering algorithms, with the exception of methods such as correlation clustering, which are able to determine the optimal number of clusters during the course of the algorithm. The correct choice of k is often ambiguous, with interpretations depending on the shape and scale of the distribution of points in a data set and the desired clustering resolution of the user. In addition, increasing k without penalty will always reduce the amount of error in the resulting clustering, to the extreme case of zero error if each data point is considered its own cluster (i.e., when k equals the number of data points, n). Intuitively then, the optimal choice of k will strike a balance between maximum compression of the data using a single cluster, and maximum accuracy by assigning each data point to its own cluster.

An important step in any clustering is to select a distance measure, which will determine how the *similarity* of two elements is calculated. This will influence the shape of the clusters, as some elements may be close to one another according to one distance and farther away according to another. For example, in a 2-dimensional space, the distance between the point (x=1, y=0) and the origin (x=0, y=0) is always 1 according to the usual norms, but the distance between the point (x=1, y=1) and the origin can be 2, $\sqrt[2]{2}$ or 1 if you take respectively the 1-norm, 2-norm or infinity-norm distance.

Common distance functions:

- The Euclidean distance, also called the 2-norm distance. A review of cluster analysis in health psychology research found that the most common distance measure in published studies in that research area is the Euclidean distance or the squared Euclidean distance.
- The Manhattan distance (aka taxicab norm or 1-norm)
- The maximum norm (aka infinity norm)
- The Mahalanobis distance corrects data for different scales and correlations in the variables
- The angle between two vectors can be used as a distance measure when clustering high dimensional data. Check for Inner product space.
- The Hamming distance measures the minimum number of substitutions required to change one member into another.

3.2 The Fuzzy C Means Clustering technique:

Fuzzy c-means algorithm [4],[5] is a *clustering* algorithm. Clustering can be considered as the most important unsupervised learning problem. So, as every other problem of this kind, it deals with finding a structure in a collection of unlabeled data.

3.2.1 Classical Sets:

A classical set is a set that has a crisp boundary. For example, a classical set X of real numbers greater than 6 is expressed as

$$A = \{x \mid x > 6\} \quad \text{……….(9)}$$

In this set of real numbers there is a clear unambiguous boundary 6 such that if *x* is greater than this number. In this case *x* either belongs to this set *'A'* or it does not belong to this set. These types of sets are called Classical Sets and the elements in this set are a part of the set or they are not a part of the set. Classical sets are an important tool in mathematics and computer science but they do not reflect the nature of human concepts and thought.

In contrast to a classical set, a fuzzy set is a set without crisp boundaries. That is, the process of an element "belongs to a set" to "does not belong to a set" is gradual. This transition is decided by the membership function of a fuzzy dataset. Real life problems have data which most of the time has a degree of "trueness" or "falseness" that is the data cannot be expressed in terms of classical set. A good example of this is; the same set A is a set of tall basketball players. According to the classical set logic a player 6.01 ft tall is considered to be tall whereas a player 5.99 ft tall is considered to be short.

3.2.2 Fuzzy Sets and Membership Function:

Membership functions give fuzzy sets the flexibility in modeling commonly used linguistic terms such as "the water is hot" or "the temperature is high." Zadeh (1965) points out that, this imprecise data set information plays an important role in human approach to problem solving. It is important to note that fuzziness in a dataset comes does not come from the randomness of the elements of the set, but from the uncertain and imprecise nature of the abstract thoughts and concepts. If *X* is a collection of objects denoted by *x*, then a fuzzy set *'A'* in *'X'* is defined as a set of ordered pairs

$$A = \{(x, \mu A(x)) \mid x \in X\} \quad \text{........(10)}$$

Fig.10a. input image

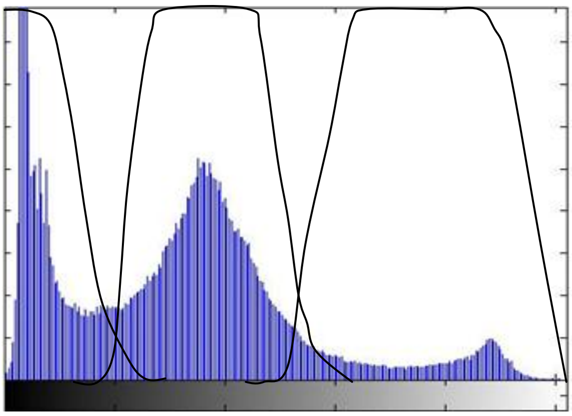


Fig.10.b. Membership values assigned image histogram

where $\mu A(x)$ is called the membership function (MF) for the fuzzy set *A*. The membership function maps each element of *X* to a membership grade between 0 and 1. If the value of the membership function is restricted to either 0 or 1, then *A* is reduced to a classical set and $\mu A(x)$ is the characteristic function of *A*. Usually X is referred to as the universe of discourse and may consist of discrete objects or continuous space. Here U matrix gives the membership values of all pixels in the image shown in [Fig.10a] for the three clusters considered in it's membership values assigned histogram diagram [Fig.10b].

$$U_{N\times C} = \begin{pmatrix} 0.4 & 0.3 & 0.3 \\ 0.5 & 0.2 & 0.3 \\ 0.2 & 0.6 & 0.1 \end{pmatrix} \quad \text{here N=3,C=3}$$

3.2.3 Significance of Membership Function in Cluster Analysis:

In FCM data is bound to each cluster by means of a membership function, which represents the fuzzy behavior of this algorithm. To do that, we build an appropriate matrix named U whose factors are numbers between 0 and 1, and represent the degree of membership between data and centers of clusters. In the FCM approach, instead, the same given datum does not belong exclusively to a well-defined cluster, but it can be placed in a middle way.

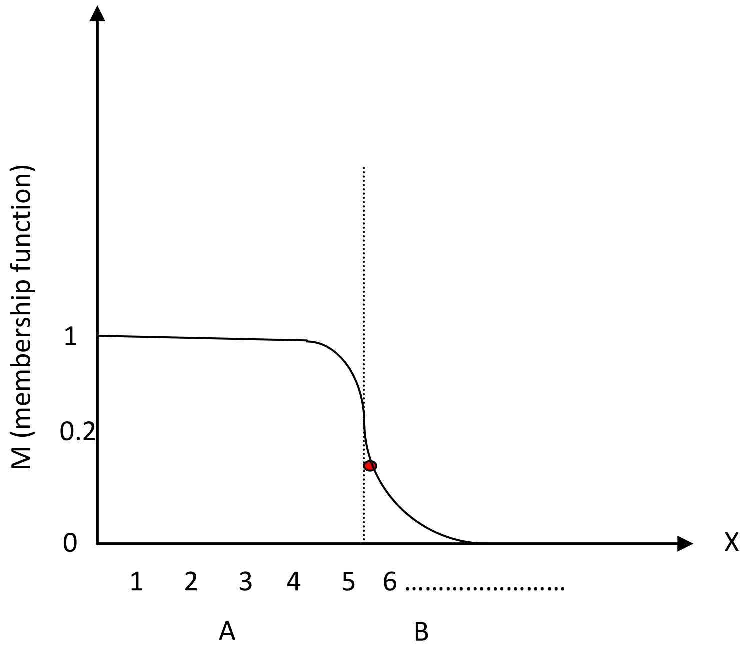


Fig.11.Membership Function for FCM Algorithm

In the case of FCM, the membership function follows a smoother line to indicate that every datum may belong to several clusters with different values of the membership coefficient. In Fig.11 (George and Yuan, 1995), the datum shown as a red marked spot belongs more to the cluster B rather than the cluster A. The value 0.2 of membership function indicates the degree of membership to A for such datum.

Now, instead of using a graphical representation, we introduce a matrix $U_{N\text{x}C}$ whose factors are the ones taken from the membership functions. The number of rows and columns depends on how many data and clusters we are considering. Here C (columns) is the total number of clusters and N (rows) is the total data points.

3.2.4 FCM Algorithm:

There are many fuzzy clustering methods being introduced. Fuzzy *C*-means clustering algorithm is one of most important and popular fuzzy clustering algorithms. At present, the FCM algorithm has been extensively used in feature analysis, pattern recognition, image processing, classifier design, etc. However, the FCM clustering algorithm is sensitive to the situation of the initialization and easy to fall into the local minimum or a saddle point when iterating. To solve this problem several other techniques have been developed that are based global optimization methods (e.g. genetic algorithms, simulated annealing). However, in many practical applications the clustering method that is used is FCM with multiple restarts to escaping from the sensibility to initial value.

For a set of unlabeled data $X = \{x_1, \ldots, x_N\}$, where N is the number of data points. Its constrained fuzzy *C*-partition can be briefly described as follows: Given that the membership function of the i^{th} ($i = 1,\ldots,N$) vector to the j^{th} ($j = 1,2,\ldots,C$) cluster is denoted as u_{ij}. The membership values are often constrained as given by [Eqn.11]

$$\forall j, \sum_{i=1}^{C} u_{ij} = 1; \forall i, j, u_{ij} \in [0,1]; \forall i, \sum_{K=1}^{C} u_{ik} > 0 \quad \ldots\ldots.(11)$$

The dissimilarity function that is used in FCM is given as:

$$J(U, C_1, C_2, \ldots.. C_C) = \sum_{i=1}^{C} J_i = \sum_{i=1}^{C}\sum_{j=1}^{n} \mu_{ij}{}^{m} d_{ij}{}^{2} \quad \ldots\ldots\ldots..(12)$$

Here, $\mu_{ij} \in [0,1]$, C_i is the centric of i^{th} cluster, d_{ij} is the Euclidian distance between i^{th} centric and j^{th} data point and $m \in [1,\infty]$ is a weighting exponent. To reach a minimum of dissimilarity function there are two conditions. These are given in [Eqns (13) and(14)].

$$C_i = \left(\sum_{j=1}^{n} \mu_{ij}{}^{m} x_j \Big/ \sum_{j=1}^{n} \mu_{ij}{}^{m} \right) \quad \text{……….(13)}$$

$$\mu_{ij} = \frac{1}{\left(\sum_{k=1}^{C} \frac{d_{ij}}{d_{kj}} \right)^{\frac{2}{m-1}}} \quad \text{………(14)}$$

With these results, well known FCM algorithms are listed below

Step 1: Initialize membership matrix *U*

Step 2: Calculate centric C_i , i =1,….c by Equation (3)

Step 3: Compute (2). Stop if either it is below a certain tolerance;

Step 4: Compute a new *U* using Equation (4), Go to Step 2.

Chapter 4
Particle Swarm Optimization

4.1 Introduction:

Particle swarm optimization algorithm [21] is a kind of evolutionary computation technique developed by Kennedy and Eberhart in 1995. It is similar to other population-based evolutionary algorithms in that the algorithm is initialized with a population of random solutions. Unlike most of the other population- based evolutionary algorithms, however, each candidate solution (called particle) is associated with a velocity and 'flies' through search space. PSO algorithm rapidly attracted researchers' attention and has been applied in neural network optimization, data clustering, engineering design, etc. In this paper, a hybrid cluster algorithm using partial swarm optimization is proposed to improve the FCM algorithm.

As described in evolutionary techniques, PSO also uses a population of potential solutions to search the search space. However, PSO differs from other evolutionary algorithms such that there are no DNA inspired operators in order to manipulate the population. Instead, in PSO[19], the population dynamics resembles the movement of a "birds' flock" while searching for food, where social sharing of information takes place and individuals can gain from the discoveries and previous experience of all other companions. Thus, each companion (called particle) in the population (called swarm) is assumed to "fly" over the search space in order to find promising regions of the landscape. In the case of minimizing a function, such regions possess lower function values than other visited previously. In this context, each particle is treated as a point in a D dimensional space, which adjusts its own "flying" according to its flying experience as well as the flying experience of other particles (companions). In our experiments, a version of this algorithm is used adding an inertia weight to the original PSO dynamics.

4.2 Notion of Particle swarm Optimization:

PSO concept is based on a metaphor of social interaction such as bird flocking [Fig.12.a] and fish schooling [Fig.12.b]. Particle Swarm Optimization (PSO) is based on the collective motion of a flock of particles: the particle swarm. In the simplest and original version of PSO, each member of the particle swarm is moved through a problem space by two elastic forces. One attracts it with random magnitude to the best location so far encountered by the particle. The other attracts it with random magnitude to the best location encountered by any member of the swarm. PSO consists of a swarm of particles and each particle flies through the multidimensional search space with a velocity, which is constantly updated by the particle's previous best performance and by the previous best performance of the particle's neighbors. PSO can be easily implemented and is computationally inexpensive in terms of both memory requirements and CPU speed. The position and velocity of each particle are updated at each time step (possibly with the maximum velocity being bounded to maintain stability) until the swarm as a whole converges to an optimum.

Fig.12.a. Bird Flocking

Fig.12.b.Fish Schooling

Reynolds proposed a behavioral model in which each agent follows three rules:
Separation: Each agent tries to move away from its neighbors if they are too close.
Alignment: Each agent steers towards the average heading of its neighbors.
Cohesion: Each agent tries to go towards the average position of its neighbors.

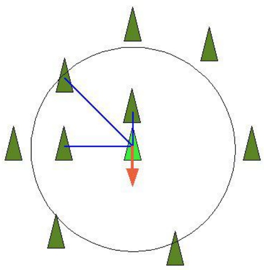


Fig.13.a. Separation

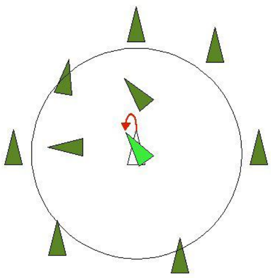


Fig.13.b.Alignment

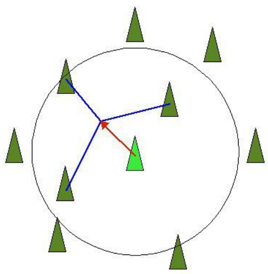


Fig.13.c.Cohesion

Kennedy and Eberhart included a 'roost' in a simplified Reynolds-like simulation so that each agent was attracted towards the location of the roost. Each agent 'remembered' where it was closer to the roost. Each agent shared information with its neighbors about its closest location to the roost.

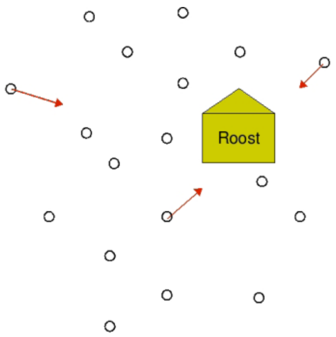


Fig.14.a. Attraction

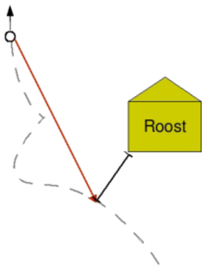


Fig.14.b. Nearest path

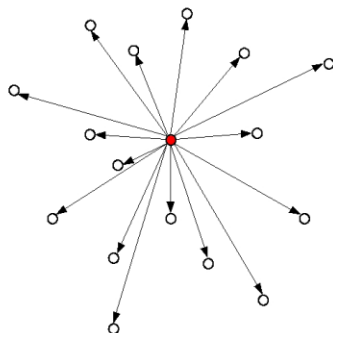


Fig.14.c. Information sharing

First, let us define the notations used in this paper. Particle i of the swarm is represented by the *D-dimensional* vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ and the best particle of the swarm, i.e., the particle with the smallest function value, is denoted by the index g. The best previous position, representing the best function value, of particle i is recorded and represented as $p_i = (\rho_{i1}, \rho_{i2}, \ldots, \rho_{id})$. The position change (velocity) of particle i is $V_i = (V_{i1}, V_{i2}, \ldots, V_{id})$. Particles update their velocity and position through tracing two kinds of 'best' value. One is its personal best (*pbest*), which is the location of its highest fitness value [Eqn23]. In global version, another is the global best (*gbest*), which is the location of overall best value, obtained by any particles in the population. Particles update their positions and velocities according to Eqns (15) and (16).

$$V_{id}(t+1) = \chi(wV_{id}(t) + c_1\phi_1(\rho_{id}(t) - x_{id}(t)) + c_2\phi_2(\rho_{gd}(t) - x_{id}(t))) \quad \ldots\ldots\ldots (15)$$

$$x_{id}(t+1) = x_{id}(t) + V_{id}(t+1) \quad \ldots\ldots\ldots(16)$$

here, $V_{id}(t)$ is the velocity of the d^{th} dimension of the i^{th} particle in the t^{th} iteration, $x_{id}(t)$ is the corresponding position, $P_{id}(t)$ and $P_{gd}(t)$ are the corresponding personal best and global best respectively, the variables ω is the inertia weight, the variables φ_1 and φ_2 are the accelerate parameters, which respectively adjust the maximal steps particles flying to the personal best and the global best, $rand_1$ and $rand_2$ are two random numbers in [0,1]. The hybrid clustering approach to image segmentation is very similar to the Fuzzy C-means strategy. It starts by choosing the number of clusters and a random initial cluster center for each cluster. PSO plays its part in assigning each pixel to a cluster. This is done according to a probability which is inversely dependent to the distance (similarity) between the pixel and cluster centers. The continuous Optimization problem can be stated as:

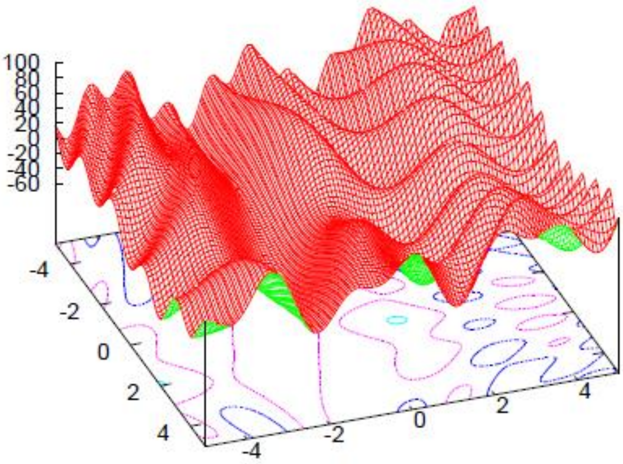


Fig.15. 3D representation of Objective function

Find $X^* \subseteq X \subseteq R^n$ such that

$$X^* = \arg\min_{x \in X} \{f(x)\} = \{x^* \in X : f(x^*) \leq f(x) \forall x \in X\} \quad \text{........(17)}$$

4.3 PSO Algorithm:

Algorithm 1 *Population Based Searches*

*1: **procedure** PBS*

2: Initialize the population

*3: **repeat***

*4: **for** i = 1 to number of individuals **do***

5: $G(\vec{x}_i)$ *% G() evaluates goodness*

*6: **end for***

*7: **for** i = 1 to number of individuals **do***

8: $P(\vec{x}_i, \theta)$ *% Modify each individual using parameter*

*9: **end for***

*10: **until** stopping criteria*

*11: **end procedure***

Algorithm 2 *Particle Swarm algorithm*

*1: **procedure** PSO*

*2: **repeat***

```
    for i = 1 to number of individuals do
        if G(x→i) > G(P→i) then.                    % G() evaluates goodness
            for d = 1 to dimensions do
                pid = xid                           % pid is the best state found so far
            end for
        end if
        g = i                                       %arbitrary
        for j = indexes of neighbors do
            if G(P→j) > G(P→g) then
                g = j                               % g is the index of the best performer in the
                                                      neighborhood
            end if
        end for
        for d = 1 to number of dimensions do
            vid(t) = f(xid(t − 1), vid(t − 1), pid, pgd)     % update velocity
            vid ∈ (−Vmax,+Vmax)
            xid(t) = f(vid(t), xid(t − 1))                   %update position
        end for
    end for
  until stopping criteria
end procedure
```

4.4 Proposed hybrid algorithm using PSO and FCM technique:

The C-means algorithm which is based on the similarity between pixels and the specified cluster centers. The behavior of the C-means algorithm mostly is influenced by the number of clusters specified and the random choice of initial cluster centers. Fuzzy C Means (FCM) is one of the most commonly used fuzzy clustering techniques for different degree estimation problems. It provides a method that shows how to group data points that populate some multidimensional space into a specific number of different clusters. FCM restriction is the clusters number which must be known a priori. FCM employs fuzzy partitioning such that a data point can belong to several groups with the degree of membership grades between 0 and 1 and the membership matrix U is constructed of elements that have value between 0 and 1. The aim of FCM is to find cluster centers that minimize a dissimilarity function. U is the membership matrix, is randomly initialized. In the fuzzy clustering, a single particle represents a cluster center vector, in which each particle P_l is constructed as follows

$$P_l = (V_1, V_2, \ldots. V_i \ldots V_C) \quad \ldots\ldots(18)$$

Where l represent the number of clusters and $l = 1, 2, \ldots, n$ and V_i is the vector of *c-th* cluster center.

$$V_i = (V_{i1}, V_{i2}, \ldots\ldots V_{iD}) \quad \ldots\ldots.(19)$$

where $1, 2, \ldots, D$ are dimensions of cluster center vectors. Therefore, a swarm represents a number of candidates clustering for the current data vector. Each point or data vector belongs to every various cluster by different membership function, thus, a fuzzy membership is assigned to each point or data vector. Each cluster has a cluster center and each iteration presents a solution giving a vector of cluster centers. We determine the position of vector P_i for every particle and update it. We then change the position of cluster centers based on these particles. For the purpose of this algorithm, we define the following notations:

n : number of data vector

C : number of cluster center

$V_l^{(t)}$: position of particle l in stage t

$Vel_l^{(t)}$: velocity of particle l in stage t

X_k : vector of data and $k = 1, 2, \ldots, n$

$\rho_l^{(t)}$: best position funded by particle l in stage t

$\rho_g^{(t)}$: best position funded by all particles in stage t

$p^{(t)}$: fuzzy pseudo partition in stage t

$A_i^{(t)}(X_K)$: membership function of data k vector in stage t into cluster i

The fitness of particles is easily measured as follows:

$$\rho_l^{(t)} = \sum_{K=1}^{n} \sum_{i=1}^{C} [A_i(X_K)]^m \left\| X_K - V_i \right\|^2 \quad , \; l = 1,2,3\ldots\ldots n\_part \ldots\ldots(20)$$

The following algorithm finds a cluster for each data vector or data point:

Step 1: Let t =0, select initial parameters such as C, initial position of particle, initial velocity of particles, c_1, c_2, χ, w, a real number $m \in (1, \infty)$, a small positive number ε ,and stopping criterion.

Step 2: Calculate the $A_i^{(t)}(X_K)$ for all particles by the following procedure, where i =1, 2,.C; k =1, 2, …, n .

For each $X_k \in X$, if $\left\|X_K - V_i^{(t)}\right\|^2 > 0$ for all i =1, 2, …, C; then define:

$$A_i^{(t+1)}(X_K) = \left[\sum_{j=1}^{C} \left(\frac{\left\|X_K - V_i^{(t)}\right\|^2}{\left\|X_K - V_j^{(t)}\right\|^2} \right)^{\frac{1}{m-1}} \right]^{-1} \quad \text{……..(21)}$$

If $\left\|X_K - V_i^{(t)}\right\|^2 = 0$ for some $i \in I \subseteq \{1, 2, …, C\}$, then define $A_i^{(t+1)}(X_K)$ for $i \in I$ by any nonnegative real numbers satisfying the following equation.

$\sum_{i \in I} A_i^{(t+1)}(X_K) = 1$ and define $A_i^{(t+1)}(X_K) = 0$ for $i \in \{1,2,…..C\} - I$

Step 3: For each particle calculate the fitness using Eqn.(17).

Step 4: Update the global best and local best position.

Step 5: Update $Vel_l^{(t)}$ and $V_l^{(t)}$ for all l=1, 2, …, *number of particle* as follows:

$$Vel_{id}^{(t+1)} = \chi(wVel_{id}^{(t)} + C_1\phi_1(\rho_{id}^{(t)} - V_{id}^{(t)}) + C_2\phi_2(\rho_{gd}^{(t)} - V_{id}^{(t)}) \quad \text{……(22)}$$

$$V_{id}^{(t+1)} = V_{id}^{(t)} + Vel_{id}^{(t+1)} \quad \text{……..(23)}$$

Step 6: Update $\rho^{(t+1)}$ by Step 2.

Step 7: Compare $\rho^{(t)}$ and $\rho^{(t+1)}$. If $p^{(t+1)} - p^{(t)} \le \varepsilon$, then stop; otherwise, increase t by one and continue form Step 3.

Chapter 5
Pattern Recognition

5.1 Introduction:

The term *Pattern recognition* [11] encompasses a wide range of information processing problems of great practical significance, from speech recognition and the classification of handwritten characters, to fault detection in machinery and medical diagnosis. Pattern recognition is a field within the area of machine learning. Alternatively, it can be defined as “the act of taking in raw data and performing an action based on the category of the data”. As such it is a collection of methods for supervised learning. Pattern recognition aims to classify data (patterns) based on either a priori knowledge or on statistical information extracted from the patterns. The patterns to be classified are usually groups of measurements or observations, defining points in an appropriate multidimensional space. Pattern recognition system consists of two-stage process.

The first stage is feature extraction and the second stage is classification. Feature extraction is the measurement of population of entities that will be classified. This assists the classification stage by looking for features that allows fairly easy to distinguish between the different classes. Several different features have to be used for classification. The set of features that are used makes up a feature vector, which represents each member of the population. Then, pattern recognition system classifies each member of the population on the basis of information contained in the information vector.

Pattern recognition is the scientific discipline whose goal is the classification of objects into a number of classes or categories. Depending on the application, these objects can be images or signal waveforms or any other type of measurements that need to be classified. We will refer to these objects using the generic term *patterns.* Pattern recognition has a long history, but before 1960s it was mostly the output of theoretical research in the area of statistics. As with everything else, the advent of computers increased the demand for practical applications of pattern recognition, which in turn set new demands for further theoretical developments. As our society evolves from the industrial to its postindustrial phase, automation in industrial production and the need for information handling and retrieval are becoming increasingly important. This trend pushed pattern recognition to the high edge of today’s engineering applications and research.

Pattern recognition is an integral part in most machine intelligent systems built for decision making. A complete pattern recognition system [Fig.16] consists of a sensor that gathers the observations to be classified or described, a feature extraction mechanism that computes numeric or symbolic information from the observations, and a classification or description scheme that does the actual job of classifying or describing observations, relying on the extracted features. The classification or description scheme is usually based on the availability of a set of patterns that have already been classified or described. This set of patterns is termed the training set, and the resulting learning strategy is characterized as *supervised learning*. Learning can also be *unsupervised*, in the sense that the system is not given an *a priori* labeling of patterns, instead it itself establishes the classes based on the statistical regularities of the patterns.

5.2 Block diagram of pattern recognition algorithm:

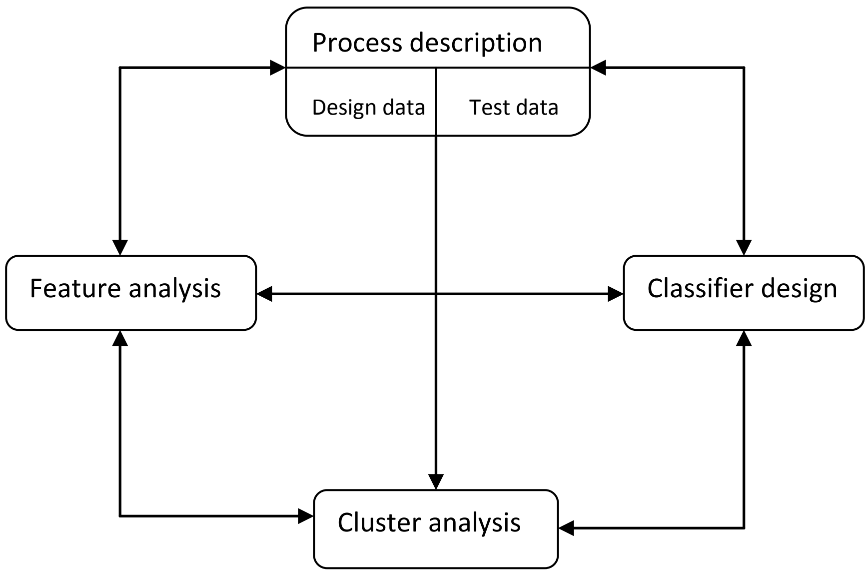


Fig.16. Elements of typical pattern recognition system

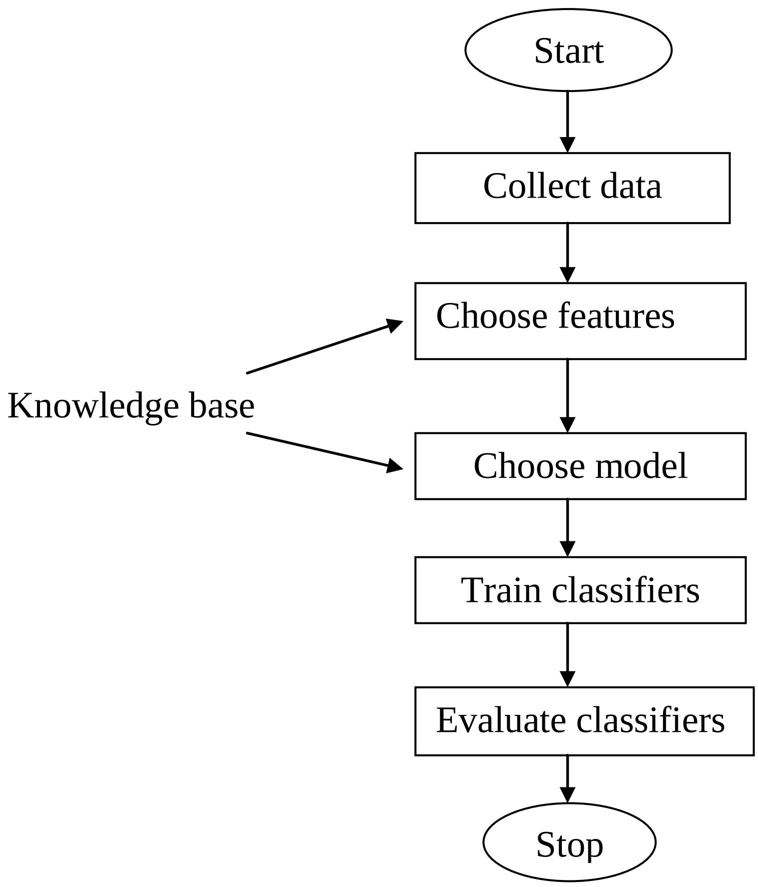
5.3 Applications of pattern recognition:

Pattern recognition is concerned with the automatic detection or classification of objects or events. Here are some examples of the problems to which pattern recognition techniques have been applied:

- Automated analysis of medical images obtained form microscopes and CAT scanners. Magnetic resonance images, nuclear medicine images, X-rays, and photographs
- Automated inspection of parts on an assembly line
- Human speech recognition by computers
- Automatic grading of plywood, steel and other sheet material
- Classification of seismic signals for oil and mineral exploration, and earthquake prediction

- Selection of tax returns to audit, stocks to buy, and people to insure
- Identification of people from fingerprints, hand shape and resize, retinal scans, voice characteristics, typing patterns and handwriting
- Automatic inspection of printed circuits and printed character and handwriting recognition
- Automatic analysis of satellite pictures to determine the type and condition of agricultural crops, weather conditions, snow and water reserves, and mineral prospects.
- Selection of good prospects from a mail-order list
- Classification of electrocardiograms into diagnostic categories of heart disease, detection of spikes in electroencephalograms, and other medical waveforms analyses.

Flow chart of pattern recognition system:



5.4 Feature Extraction:

Feature extraction is a phase of the recognition process in which the objects are measured. A measurement is the value of some quantifiable property of an object. A feature is a function of one or more measurements, computed so that it quantifies some significant characteristic of the object. This process produces a set of features that, taken together, comprise the feature vector. There are various ways to generate features from the raw data set. A number of transformations can be used to generate features. The basic idea is to transform a given set of measurements to a new set of features. Transformation of features can lead to a strong reduction of information as compared with the original input data. So for most of the classification a relative small number of features is sufficient for correct recognition. Obviously feature reduction is a sensitive procedure since if the reduction is done incorrectly the whole recognition system may fail or not present the desired results. Examples of such transformations are the Fourier transform, the Karhunen-Loeve transform, and the Haar transform. However feature generation via linear transformation techniques is just one of the many possibilities. There is a number of alternatives which are very much application dependent. Examples of such features are moment-based features, chain codes, and parametric models.

In pattern recognition and in image processing, Feature extraction is a special form of dimensionality reduction. When the input data to an algorithm is too large to be processed and it is suspected to be notoriously redundant (much data, but not much information) then the input data will be transformed into a reduced representation set of features (also named features vector). Transforming the input data into the set of features is called *features extraction*. If the features extracted are carefully chosen it is expected that the features set will extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input.

5.5 Cluster Analysis:

The main objective in clustering techniques is to partition a given data set into homogeneous clusters. The term homogeneous is used in the sense that all points in the same group are similar to each other and are not similar to points in other groups. The similarity of these points is defined according to some established criteria. While the use of clustering in pattern recognition[11] and image processing is relatively recent, cluster analysis is not a new field. Since it has been used in other disciplines, such as biology, psychology, geology and information retrieval. The majority of the clustering algorithms find clusters of a particular shape. Humans are the best cluster seekers in two dimensions, but most of the real problems involve clustering in higher dimension. And the difficulties with an intuitive interpretation of data embedded in a high dimensional space are evident. Clustering method is a very active field in pattern recognition and data miming. Consequently a large amount of clustering algorithms continues to appear in the literature. Most of these algorithms are based on proximity measures. Even though, there are a class of algorithm based on different combinations of a proximity measure and a clustering scheme. Clustering is a major tool used in a number of applications, which can be basically used in four different ways namely data reduction, hypothesis generation, hypothesis testing and prediction based on group.

5.6 Classifiers Design:

Classifiers are systems designed to perform the classification stage of the recognition system. There are several approaches for the design of the classifier in a pattern recognition system and they can be grouped in three classes: classifiers based on bayes decision theory, linear and nonlinear classifiers. The first approach builds upon probabilistic arguments stemming from the statistical nature of the generated features. This is due to the statistical variation of the patterns as well as to possible noise obtained in the signal acquisition phase. The objective of this type of design is to classify an unknown pattern in the most probable class as deduced from the estimated probability density functions. Even though linear classifiers are more restricted in their use, the major advantage is their simplicity and computational demand in solving problems which do not require more sophisticated nonlinear model. Examples of linear classifiers are the perceptron algorithm and least squares methods. For problems that are not linearly separable and for which the design of a linear classifier, even in

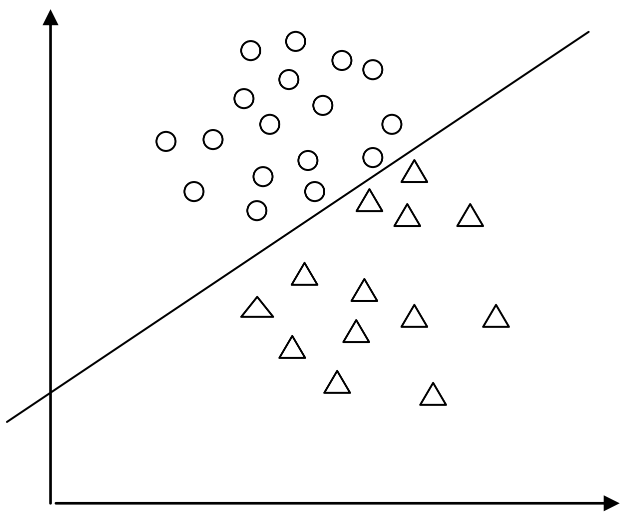


Fig.17.a.Illustration of uniform pattern Classification

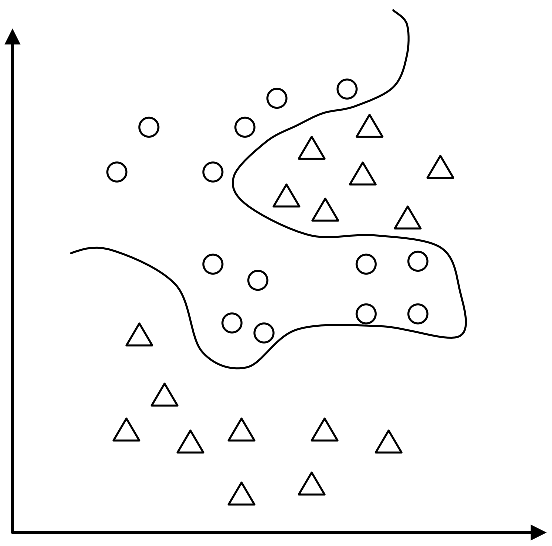
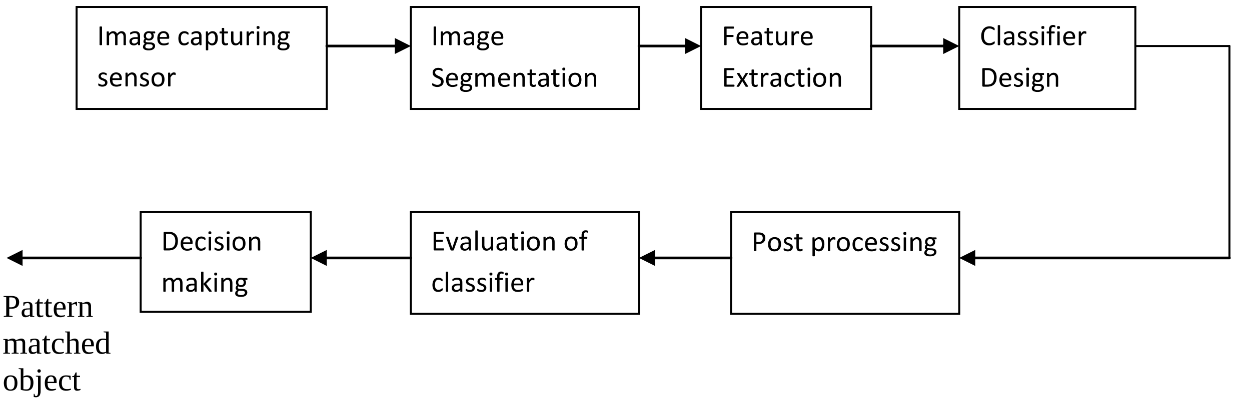


Fig.17.b. Illustration of nonuniform pattern Classification

an optimal way, does not lead to satisfactory performance, the use of nonlinear classifier are mandatory.

5.7 Pattern recognition system for image processing:



5.8 Proposed pattern matching algorithm:

The proposed algorithm for object tracking exploits pattern matching with the features above and makes use of the minimum distance search in the feature space. We now go into more details of our algorithm. Using the image segmentation result of the object i in the t-th frame, we first extract the features of the object (t, i). Here, the notation (t, i) stands for the objects i in the t-th frame. Then we perform the minimum distance search in the feature space between (t, i) and $(t-1, j)$ for all objects j in the preceding frame. Finally, the object (t, i) is identified with the object in the preceding frame which has the minimum distance from (t, i). Repeating this matching procedure for all segments in the current frame, we can identify all objects one by one and can keep track of the objects between frames.

After the image is segmented, we can view all objects (connected regions satisfying similarity criteria) in an image no matter whether they are moving or not. Then we perform pattern matching on the successive segmented frames having desired object (Object we are interested to track) features extracted in hand. Then the features extracted for segments are described as

Feature Extraction for Segments:

In this subsection, we describe the extracted features of segmented objects. Figure 19 shows an example of a segment for explanation purposes.

1. Area*:* By counting the number of pixels included in segment *I* of the t^{th} frame, we calculate the area of the object $a_i(t)$.

2. Width and Height*:* We extract the positions of the pixel P_{xmax} (P_{xmin}) which has the maximum (minimum) *x*-component.

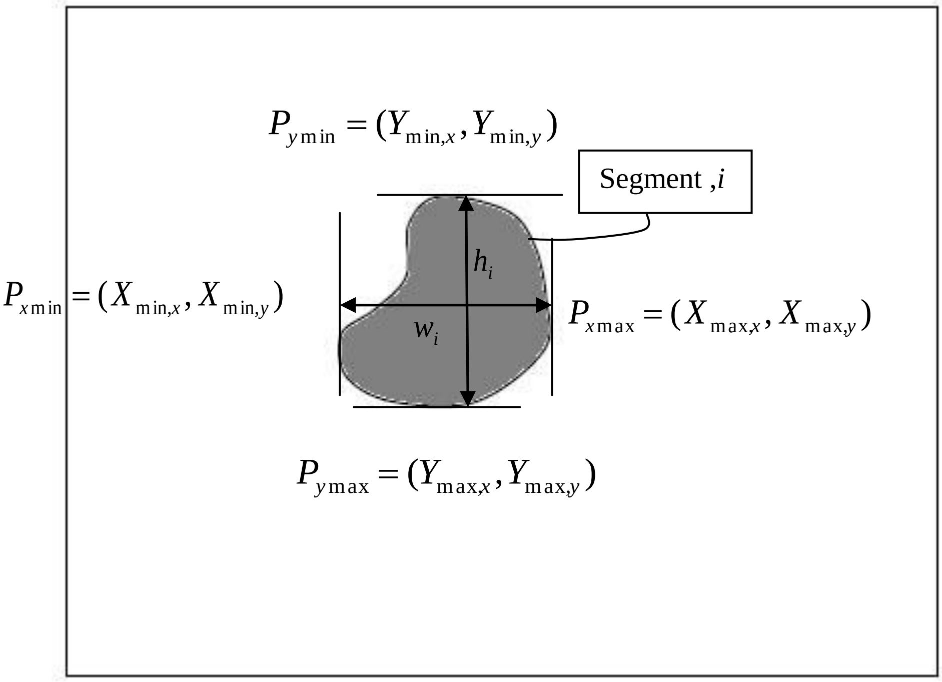


Fig.19. Explanation of the proposed feature extraction for segment i in the image segmentation result

$$p_{x\max} = (X_{\max,x}, X_{\max,y}) \quad \text{.....(24)}$$

$$p_{x\min} = (X_{\min,x}, X_{\min,y}) \quad \text{.....(25)}$$

where *Xmax,x, Xmax,y*, *Xmin,x*, and *Xmin,y* are the *x*- and *y* coordinates of the rightmost and leftmost boundary of segment *i*, respectively. In addition, we also extract

$$p_{y\max} = (Y_{\max,x}, Y_{\max,y}) \quad \text{.....(26)}$$

$$p_{y\min} = (Y_{\min,x}, Y_{\min,y}) \quad \text{.....(27)}$$

Then we calculate the width *w* and the height *h* of the objects as follows

$$w_i(t) = X_{\max,x} - X_{\min,y} \quad \text{.....(28)}$$

$$h_i(t) = Y_{\max,x} - Y_{\min,y} \quad \text{.....(29)}$$

3. Position: We define the positions of each object in the frame as follows

$$X_i(t) = \frac{X_{\max,x} + X_{\min,x}}{2} \quad \text{.....(30)}$$

$$Y_i(t) = \frac{Y_{\max,x} + Y_{\min,x}}{2} \quad \text{.....(31)}$$

4. Colour: Using the image data at *Pxmax*, *Pxmin*, *Pymax* and *Pymin*, we define the color feature of each object as in for the *R* (Red) component as well as by equivalent equations for the *G* and *B* components.

$$R_i(t) = \frac{[R(P_{x\max}) + R(P_{x\min}) + R(P_{y\max}) + R(P_{y\min})]}{4} \quad \text{.....(32)}$$

5.8.1 Motion Vector Determination:

Video sequences generally comprise two types of motion: the motion of objects visible in the scene and the global motion that is caused by the moving camera. Object motion is usually also difficult to describe, since in general, objects can show articulated or deformable motion. In general, the motion of objects can be rather complex and very difficult to model. Examples are water waves, explosions, traffic lights, or other sudden changes. On the other hand, camera motion is restricted to only a few degrees of freedom like the camera rotation angles or its focal length. When analyzing a video sequence, the inverse problem occurs: find the parameters which describe the apparent motion in the video sequence. Techniques for estimating the motion parameters generally follow one of two fundamental approaches: direct estimation algorithms and feature based algorithms.

The second approach for motion estimation [12],[13] is feature-based techniques. They determine a small set of feature points in each of the input frames and establish correspondences between matching points. The feature points are selected such that the motion of the point can be computed with high reliability and accuracy. After the point correspondences have been established, the motion parameters [Eqns 35, 36] are determined by fitting the motion model to the point-correspondence data. Since the feature-based methods primarily use the position of feature points instead of a direct comparison of the image data, the feature-based methods are more robust to changes of illumination or noise than the direct methods. Furthermore, they allow large motions between images and they are also faster to compute in one direction, namely perpendicular to the line direction. Along the line, the texture is uniform, making it impossible of finding a best-matching position. The problem becomes even worse if the pattern only shows uniform color, providing no directed structure in the texture at all. This indeterminism is known as the aperture problem. It is only possible to determine the position of the pattern reliably if the pattern shows variations in two directions.

5.8.2 Proposed concept for Motion vector determination:

The proposed algorithm for object tracking exploits pattern matching with the features above and makes use of the minimum distance search in the feature space. We now go into more details of our algorithm. Using the image segmentation result of the object *i* in the *t*-th frame, we first extract the features of the object (*t, i*). Here, the notation (*t, i*) stands for the objects *i* in the *t*-th frame. Then we perform the minimum distance search in the feature space between (*t, i*),(*t* − 1, *j*) for all objects *j* in the preceding frame. Finally, the object (*t, i*) is identified with the object in the preceding frame which has the minimum distance from (*t, i*).

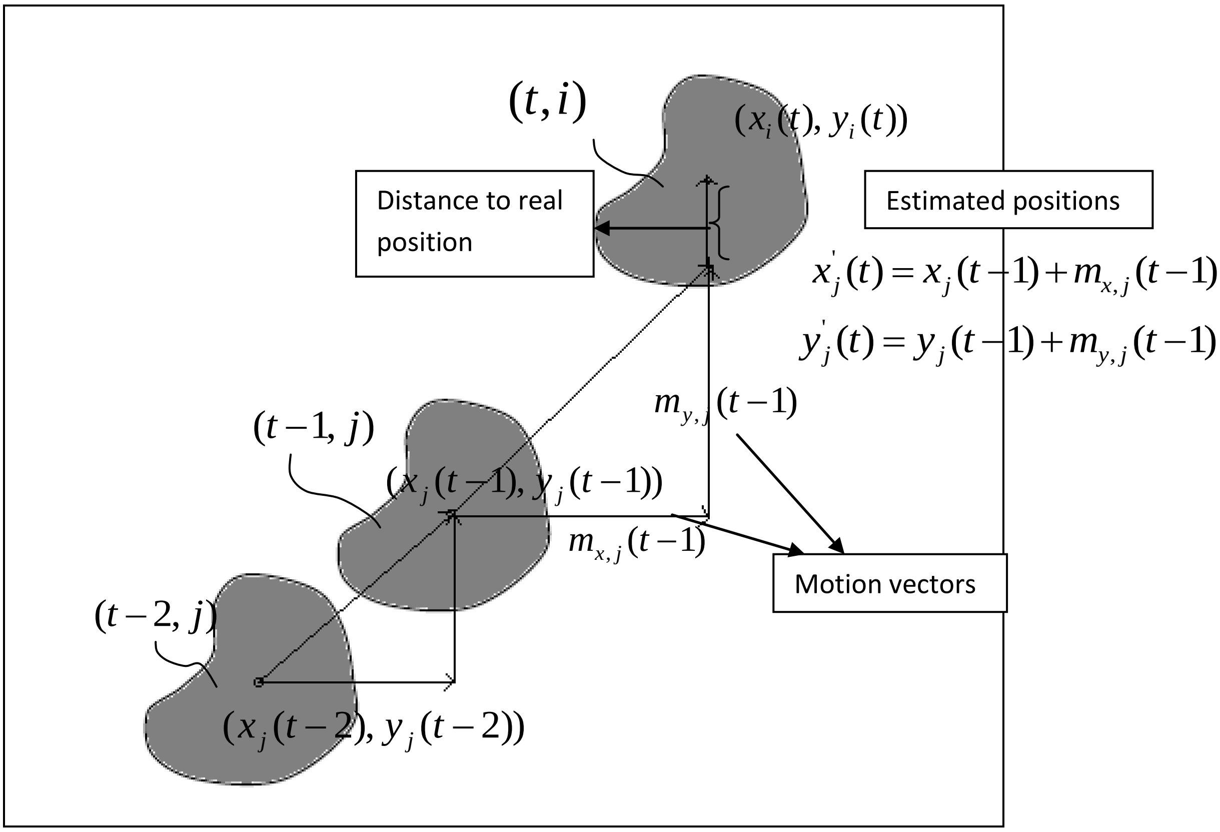


Fig.20.Estimation of positions in the next frame

Repeating this matching procedure for all segments in the current frame, we can identify all objects one by one and can keep track of the objects between frames. A few comments on further refinements of the proposed algorithm are in order.

(1) In calculation of the distance between (t, i) and ($t-1$, j) in position space, it is more appropriate to take account of motion determination and use estimated positions [Eqns 33,34] instead of raw positions $x_j(t-1)$ and $y_j(t-1)$. The quantities $m_{x,j}(t-1)$ and $m_{y,j}(t-1)$

$$x_i^{'}(t+1) = x_i(t) + m_{x,i}(t) \quad \text{.....(33)}$$

$$y_j^{'}(t+1) = y_j(t) + m_{y,j}(t) \quad \text{.....(34)}$$

$$m_{x,j}(t-1) = x_j(t-1) - x_j(t-2) \quad \text{.....(35)}$$

$$m_{y,j}(t-1) = y_j(t-1) - y_j(t-2) \quad \text{.....(36)}$$

correspond to the motion vector of x- and y-directions of the object j. These replacements are available and used from the third frame onwards.

(2) We have not specified the distance measure [Eqns37,38] used for matching yet. In the simulation experiments we could confirm that besides the Euclidean distance D_E the simpler Manhattan distance D_M is already sufficient for object tracking purposes. These two distances between vectors $(x_1, \cdots, x_n)$ and $(y_1, \cdots, y_n)$ are defined as

$$D_E = \sqrt{(x_1 - y_1)^2 + \ldots\ldots + (x_n - y_n)^2} \quad \text{.....(37)}$$

$$D_M = |x_1 - y_1| + \ldots\ldots + |x_n - y_n| \quad \text{.....(38)}$$

(3) In order to treat all object features with equal weights, it is necessary to normalize the features. One possible way is dividing them by their maximum values. Dividing by 2n, where the integer n is determined for each feature so that approximately equal weights results, is another possibility. The second possibility has the advantage that the division can be realized by a shifting operation in a hardware realization.

5.8.3 Objects tracking algorithm:

1) Feature Extraction

a) Extraction of the area *ai*(*t*) and the positions of the pixels *Pxmax*, *Pxmax*, *Pxmax* and *Pxmax* for the segment *i*.

b) Calculation of the width, height of the segment *i*

$$w_i(t) = X_{\max,x} - X_{\min,x}$$

$$h_i(t) = Y_{\max,y} - Y_{\min,y}$$

c) Calculation of the (present) positions ($x_i(t)$, $y_i(t)$) of the segment *i*.

$$x_i(t) = \frac{X_{\max,x} + X_{\min,x}}{2}$$

$$y_i(t) = \frac{Y_{\max,y} + Y_{\min,y}}{2}$$

d) Calculation of the colour features of the segment *i*.

$$R_i(t) = \frac{[R(P_{x\max}) + R(P_{x\min}) + R(P_{y\max}) + R(P_{y\min})]}{4}$$

$$G_i(t) = \frac{[G(P_{x\max}) + G(P_{x\min}) + G(P_{y\max}) + G(P_{y\min})]}{4}$$

$$B_i(t) = \frac{[B(P_{x\max}) + B(P_{x\min}) + B(P_{y\max}) + B(P_{y\min})]}{4}$$

2) Pattern Matching in the Feature Space

if (*t* == 1) then

a) Perform feature-extraction for segments.

b) go to (image segmentation of the next frame).

if ($t \geq 2$) then

a) Perform feature-extraction for segment i.

b) Calculation of distances $D(t, i; t-1, j)$, $\forall j$.

c) Search for the minimum distance among the distances

$D(t, i; t-1, k) \equiv \min D(t, i; t-1, j)$, $\forall j$.

d) Identify (t, i) with $(t-1, k)$ and remove $(t-1, k)$ from reference data.

e) Estimation of the positions of the segment i in the next frame

$$x_i^{'}(t+1) = x_i(t) + m_{x,i}(t)$$

$$y_i^{'}(t+1) = y_i(t) + m_{y,i}(t)$$

f) Repeat the matching procedure [from b) to e)] for all segments in the t-th frame.

g) go to (image segmentation of the next frame)

Chapter 6
Simulation Results

In this chapter all the simulation results for all the techniques are explained. Initially the moving objects in video images are tracked based on image segmentation and pattern matching techniques. The Object tracking is done in three ways, those are

1. Histogram based thresholding by manual threshold selection and Pattern matching.
2. Segmentation by Otsu's global thresholding and Pattern matching.
3. Fuzzy C means clustering with particle swarm optimization and Pattern matching.

After the segmentation, it is possible to detect all the objects or connected regions in the image, no matter whether they are moving or not. Later we extract the features of the object patterns which we are interested to track. The segmentation is done for all the consecutive video images in the sequence and then sent to feature extraction process. Then feature matching (pattern matching) is done with next frame, among all the patterns the pattern matches with our desired object only and subject to the condition that desired object only satisfies minimum distance search in the feature space. i.e. after segmentation we find the distances of an object i in t^{th} frame to all the objects $(j=1,2,3....n)$in the $(t+1)^{th}$ frame. let us consider a video sequence in which 3 objects are in motion, if the frame rate is high and all the frames in the sequence are taken in to account, then one object in a frame gives minimum distance with the same object in next frame. So the object is tracked in the next frame based on two conditions one is pattern matching and the second is minimum distance search in the feature space. Once the pattern matching is over, then the motion vectors are calculated for each object in the image. But in real life nonlinear motion dominates linear motion. In video sequences the movement of the objects may be either linear or non linear, if the movement is linear then estimated positions (adding distance in x,y directions to previous centroid) in the next frame becomes the real positions, whereas nonlinear case in addition to this, distance to real position will have to be added. Then we can move the region of interest with the pattern we want to track.

Here, the 3 techniques have been repeated for ball video sequence and tennis video sequence. Each image in ball video sequence is of size 320 X 240 and tennis video sequence is of size 352 X 220. In the ball video sequence the frames from 9 to 35 are taken, whereas in tennis video sequence the frames from 21 to 35 are taken. The ROI (region of interested) for ball video sequence is of size 50 X 50, and it is 16 X 16 for tennis video sequence.

The simulation results of histogram based tracking are shown below:

6.1 Histogram based segmentation and pattern matching:

Fig.21. Frame 10 of ball video sequence

Fig.22. Frame 11 of ball video sequence

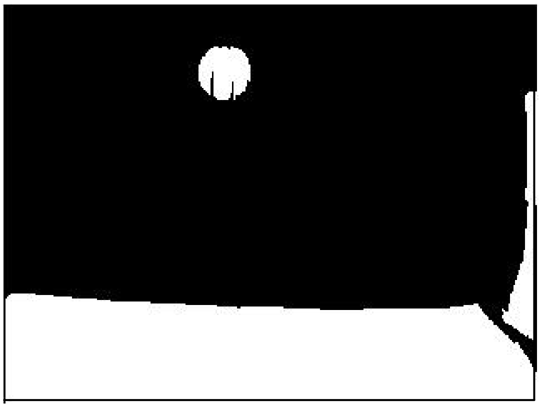


Fig.23. Segmented image of frame 10 of ball video sequence

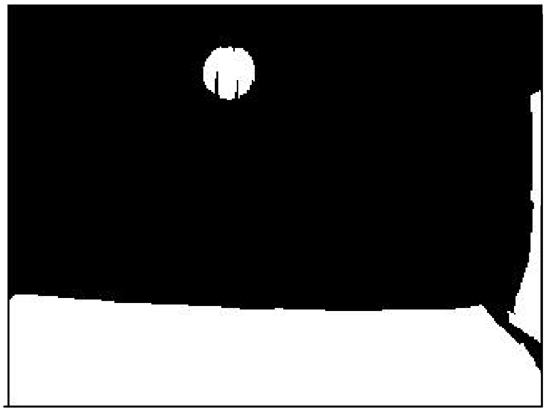


Fig.24. Segmented image of frame 11 of ball video sequence

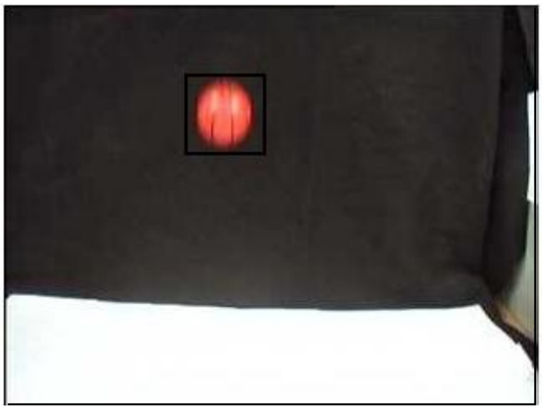


Fig.25. ball in frame of 10 of ball video Sequence is tracked.

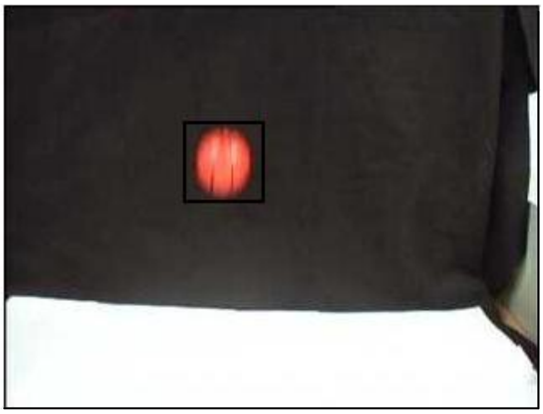


Fig.26. ball in frame 11 of ball video sequence is tracked.

6.2 Drawback of Histogram based thresholding:

The disadvantage of histogram based thresholding is the pattern in the segmented image will not be clear. So in the segmented image shown in Fig.27, there are three patterns ball, floor and side wall. In the Fig.28 also the same three patterns exists but with some variation. So may it happen sometimes that the pattern in the previous frame may not be recognized in next frame because of segmentation fault and in histogram based segmentation the threshold is selected manually, if the proper threshold is not selected then we can not differentiate foreground and background objects. So the object tracking for desired object is not possible. The disadvantage of this method is, it can't be used for complex image sequences where the background is changed or the camera is moving. It can be used with video sequences with stationary camera and with less tracking efficiency.

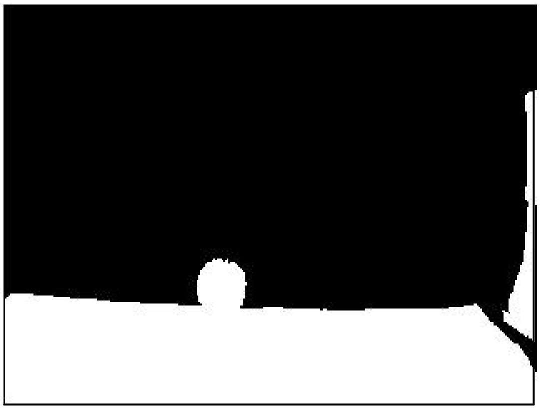


Fig.27. Segmented image of frame 27 of ball video sequence

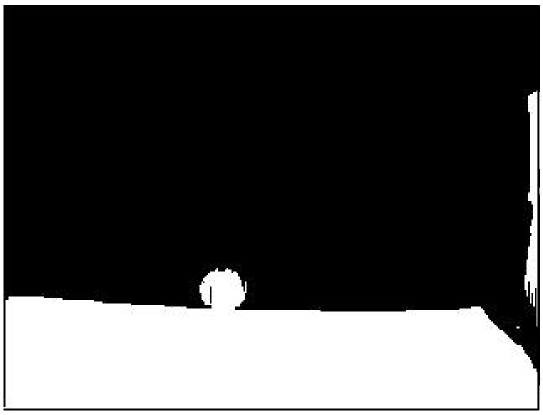


Fig.28. Segmented image of frame 28 of ball video sequence

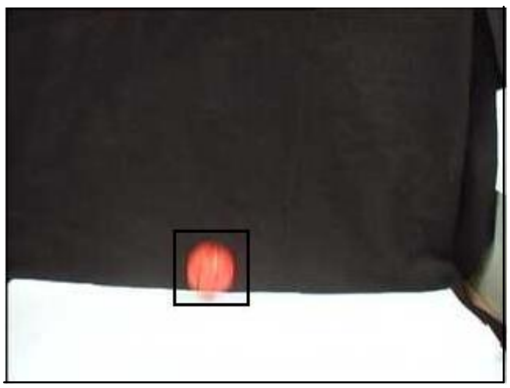


Fig.29. Pattern matched object in frame 27 of ball video sequence

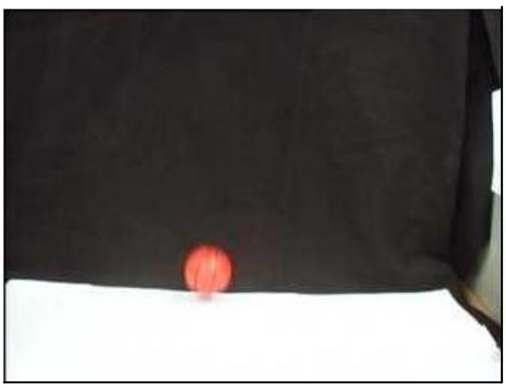


Fig.30. Pattern matching fails in frame 28 of ball video sequence

In the above figure 27 the 3 patterns in the image are clearly visible in frame 27, but in figure 28 the ball, wall patterns are not clearly visible, pattern matching fails here and the same is repeated for frames 52 and 57 also.

6.3. Tracking using Otsu's global thresholding:

The second approach for segmentation is Otsu's global thresholding. In this approach the segmentation accuracy increases, but the difficulty arises when the illumination changes from one frame to another frame in video sequence. This can be observed in the below shown figures. In original figures shown in Fig.21, 22 it is found that there is non uniform illumination present in the image. This approach is better than histogram based segmentation.

6.3.1 Ball video sequence:

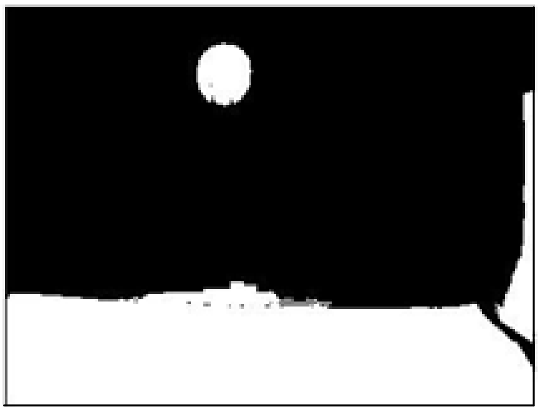


Fig.31. Segmented image of frame 10 of ball video sequence

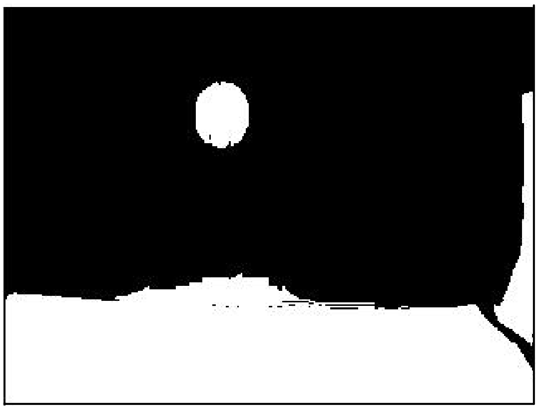


Fig.32. Segmented image of frame 11 of ball video sequence.

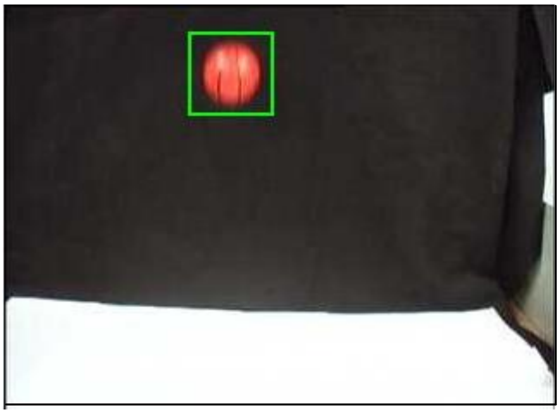


Fig.33. Pattern matched object in frame 10 of ball video sequence

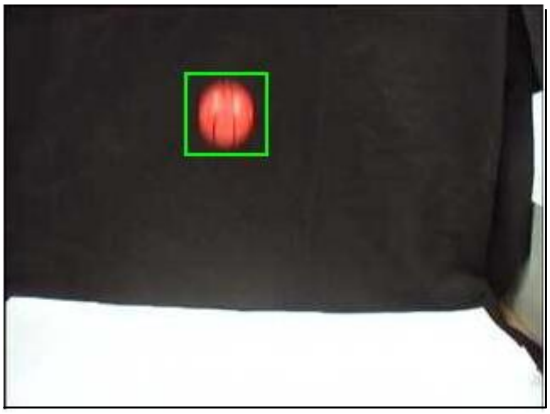


Fig.34. Pattern matched object in frame 11 of ball video sequence

TABLE.I

EXTRACTED FEATURES FOR BALL VIDEO SEQUENCE

Object	Area	width	height	cenx	ceny	mv-x	mv-y
(10,1)	20988	66	318	207	161	0	0
(10,2)	961	33	31	22	134	20	1
(10,3)	3020	151	21	128	310	0	0

Object	area	width	height	cenx	ceny	mv-x	mv-y
(11,1)	20988	66	318	207	161	0	0
(11,2)	954	37	31	22	142	44	1
(11,3)	3029	151	21	128	310	0	0

TABLE.II

THE EUCLIDEAN DISTANCE BETWEEN SUCCESSIVE FRAMES FOR BALL VIDEO

(Frame, Object)	(10,1)	(10,2)	(10,3)
(11,1)	**0**	176.3589	168.6476
(11,2)	186.9599	**20.0250**	205.4556
(11,3)	168.6476	196.7867	**0**

6.3.2 Tennis video sequence:

In tennis video sequence, there are 2 moving objects in the sequence, one is ball and the other is hand of the player, In segmented image three patterns appears as the connected regions.

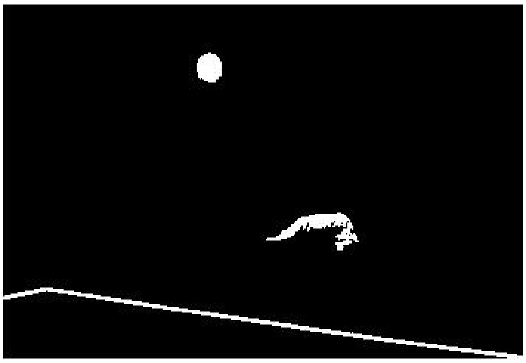


Fig.35. Segmented image of frame 22 of tennis video sequence

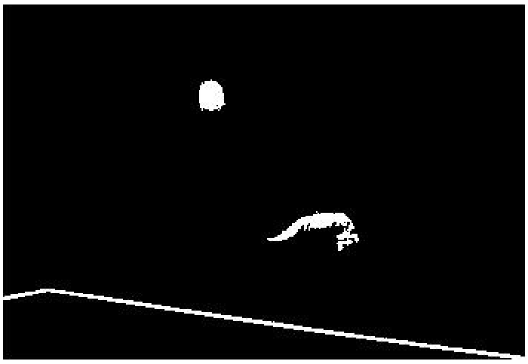


Fig.36.Segmented image of frame 23 of tennis video sequence

Features are extracted for these three patterns and matched with all the patterns in the next frame, and the ROI is moved accordingly. In this case, the camera is moved in a long shot, appearance of the images in the video sequence changes, though the algorithm works. But the segmentation results are less accurate when compared to FCM-PSO method. But this method is faster than FCM-PSO, can be used for real time applications. The efficiency of the tracker is less when compared to FCM-PSO technique. Each frame in the video is of size 352 X 220, the mask regions are 24 X 24 for ball and 35 X 35 for hand. In this video the ball and the hand are moving so the motion vectors of these two objects are finite, whereas the edge of the table being the 3rd pattern in the segmented image is stationary in all frames so motion vectors for this object is always zero. If there exists any stationary objects in the image then nonzero minima in all distances have to be considered, if that is not the case then minimum distance search in the feature space is to be done.

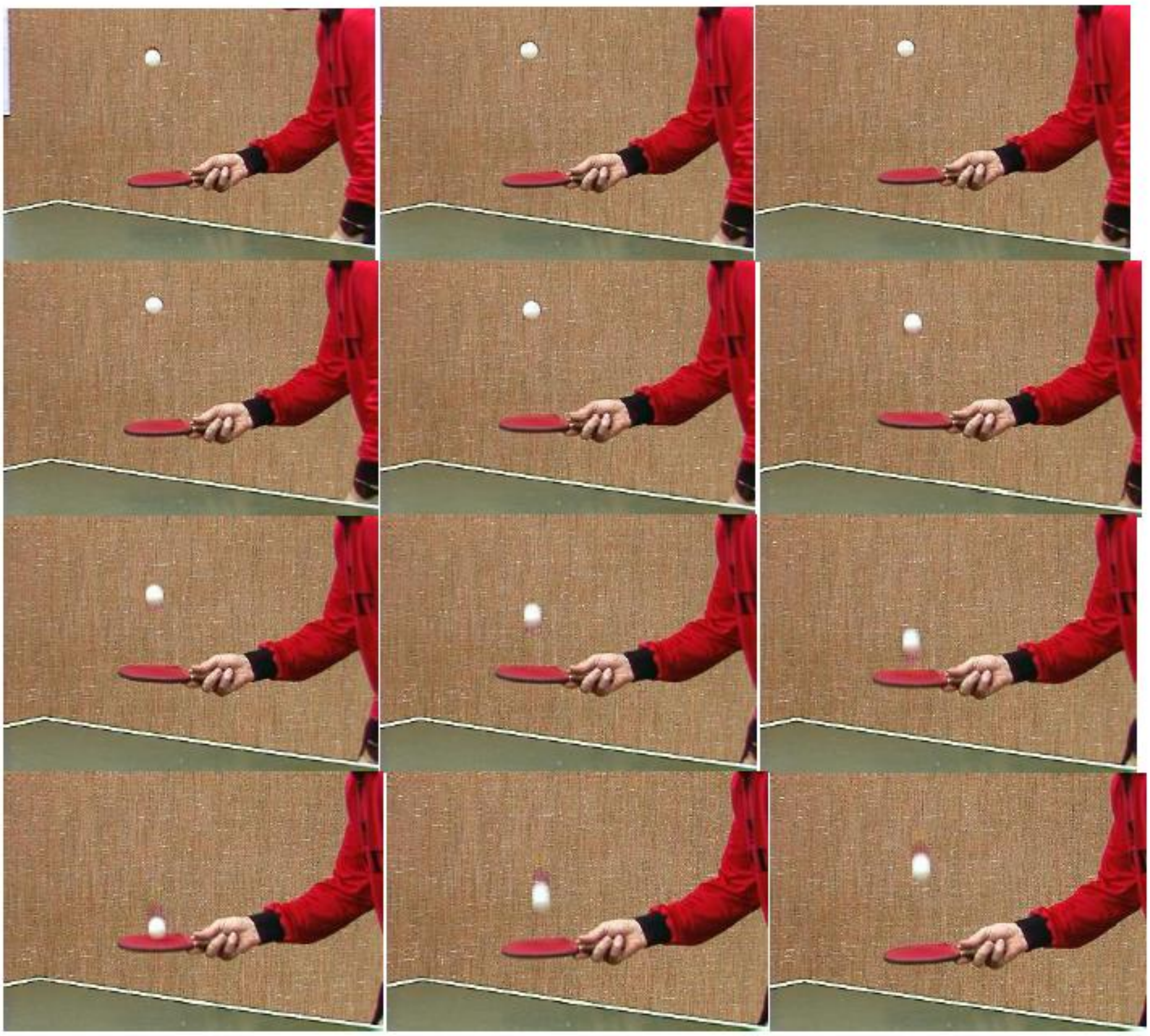


Fig.37. Frames 22 to 34 of tennis video sequence.

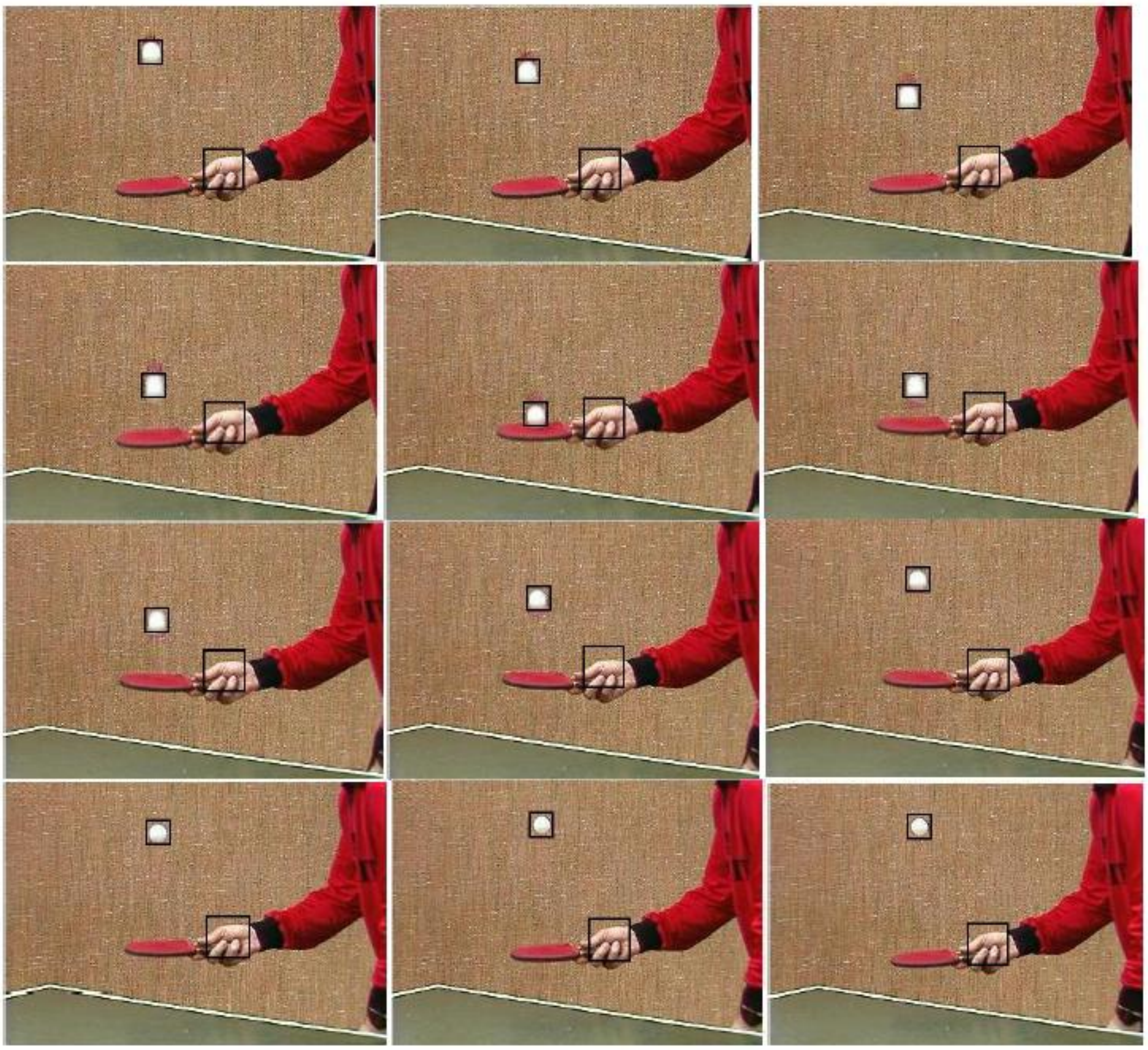


Fig.38. The objects tracked (ball, hand) images (frame 22 to 34) of tennis video sequence.

TABLE.III

EXTRACTED FEATURES FOR TENNIS VIDEO SEQUENCE

(Frame, Object)	Area	Width	Height	Cenx	Ceny	mv-x	mv-y
(22,1)	16497	47	351	217	177	2	1
(22,2)	304	19	16	44	140	14	0
(22,3)	1550	25	62	155	210	0	0

(Frame, object)	Area	Width	Height	Cenx	Ceny	mv-x	mv-y
(23,1)	16497	47	351	217	177	1	0
(23,2)	340	20	17	62	142	32	2
(23,3)	1586	61	26	54	210	0	0

TABLE .IV

THE EUCLIDEAN DISTANCE BETWEEN SUCCESSIVE FRAMES FOR TENNIS VIDEO

(Frame no, Object)	(23,1)	(23,2)	(23,3)
(22,1)	**0**	158.9025	71.196
(22,2)	190.6253	**32.0624**	142.3938
(22,3)	70.2353	115.2085	**1**

6.4 Tracking using FCM-PSO segmentation and pattern matching:

This method is more robust, accurate and the efficiency of the tracker is high in all the three methods because the segmentation accuracy is dependent on the no of clusters, which is known apriori. If the clusters are more then the segmented image consists of all the finer details of the image so that several patterns exists in the segmentation result and takes some time to match the pattern from one frame to another frame. In this case, the tracker does not mislead, because the segmentation is so clear and it is very easy to perform pattern matching.

These are some of the segmentation results with various cluster numbers and their error plots,

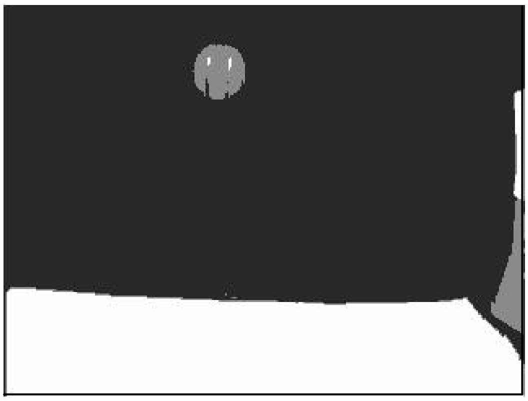


Fig.39. Segmentation using FCM-PSO with c=3

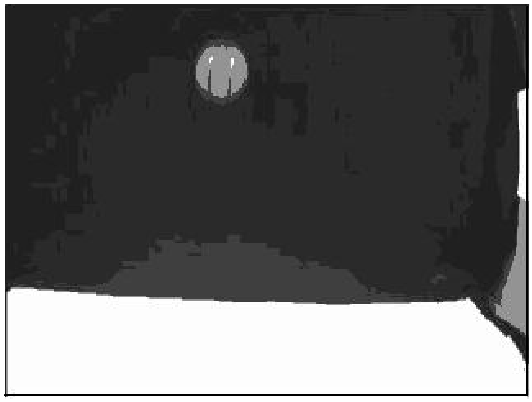
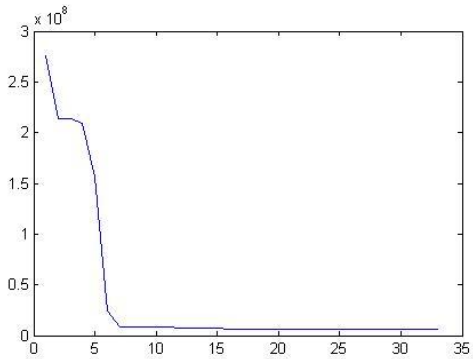
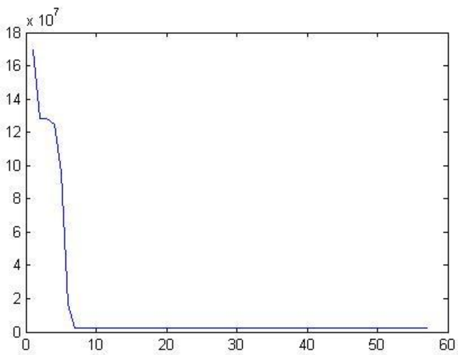


Fig.40.Segmentation using FCM-PSO with c=5





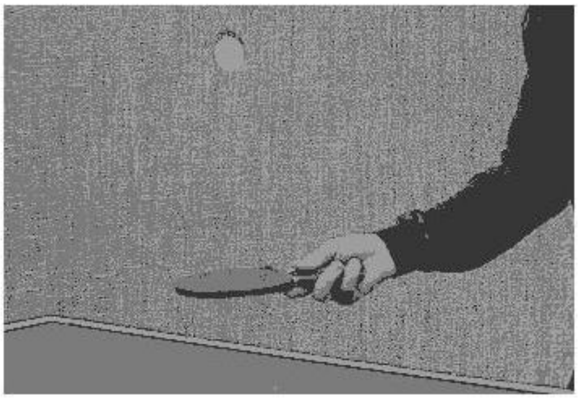


Fig.43. Segmentation of frame 21 of tennis video sequence with clusters=3

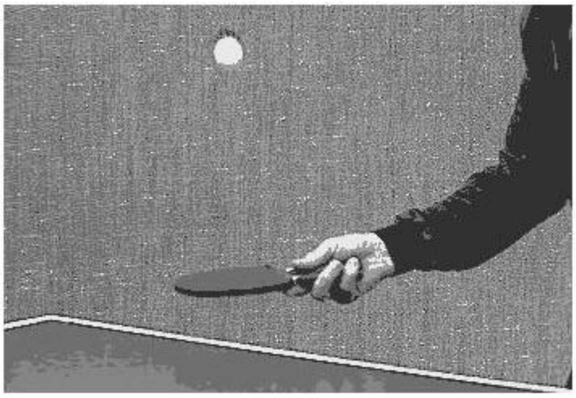
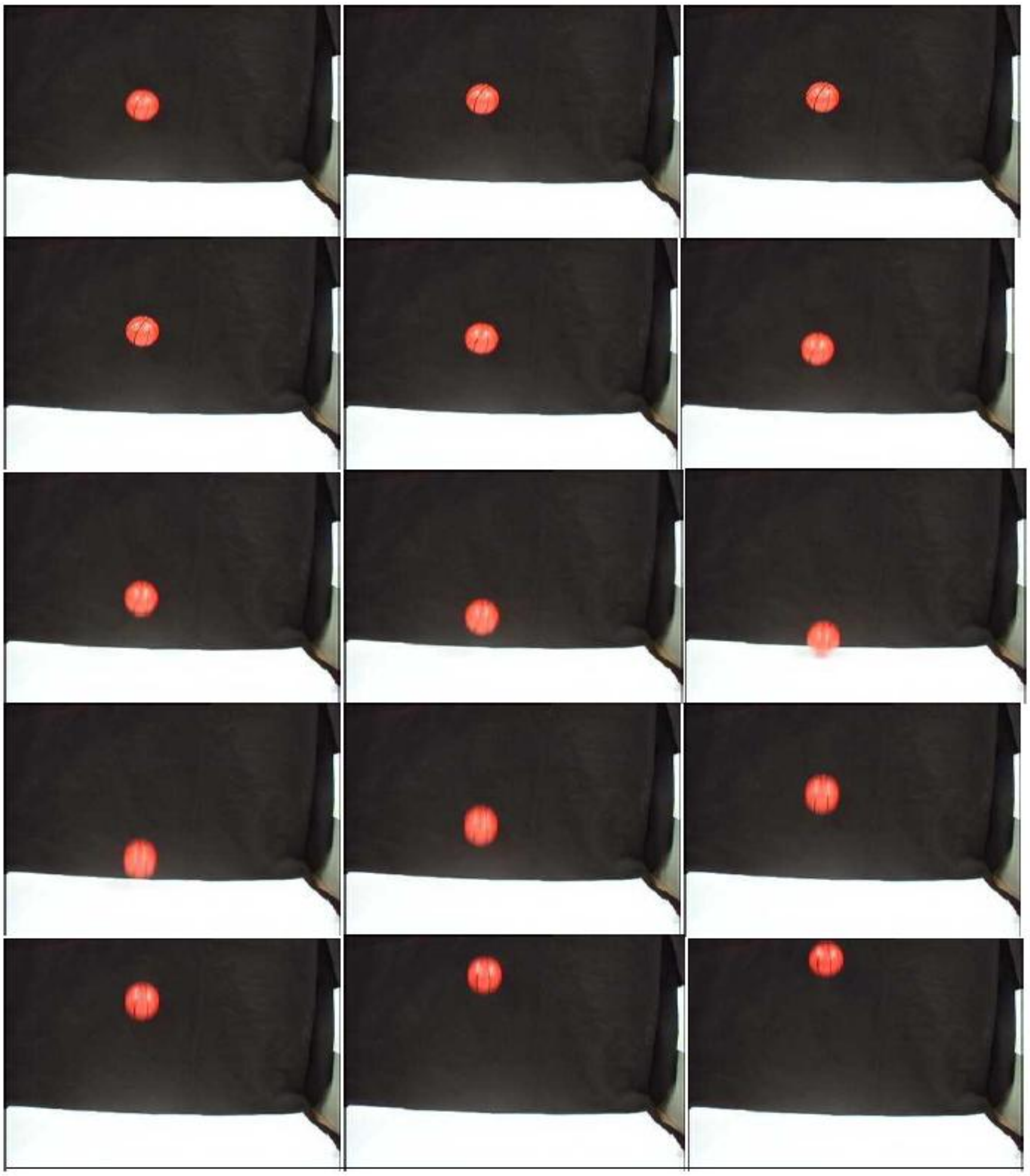


Fig.44. Segmentation of frame 21 of tennis video sequence with clusters=5

6.4.1 Ball video sequence:



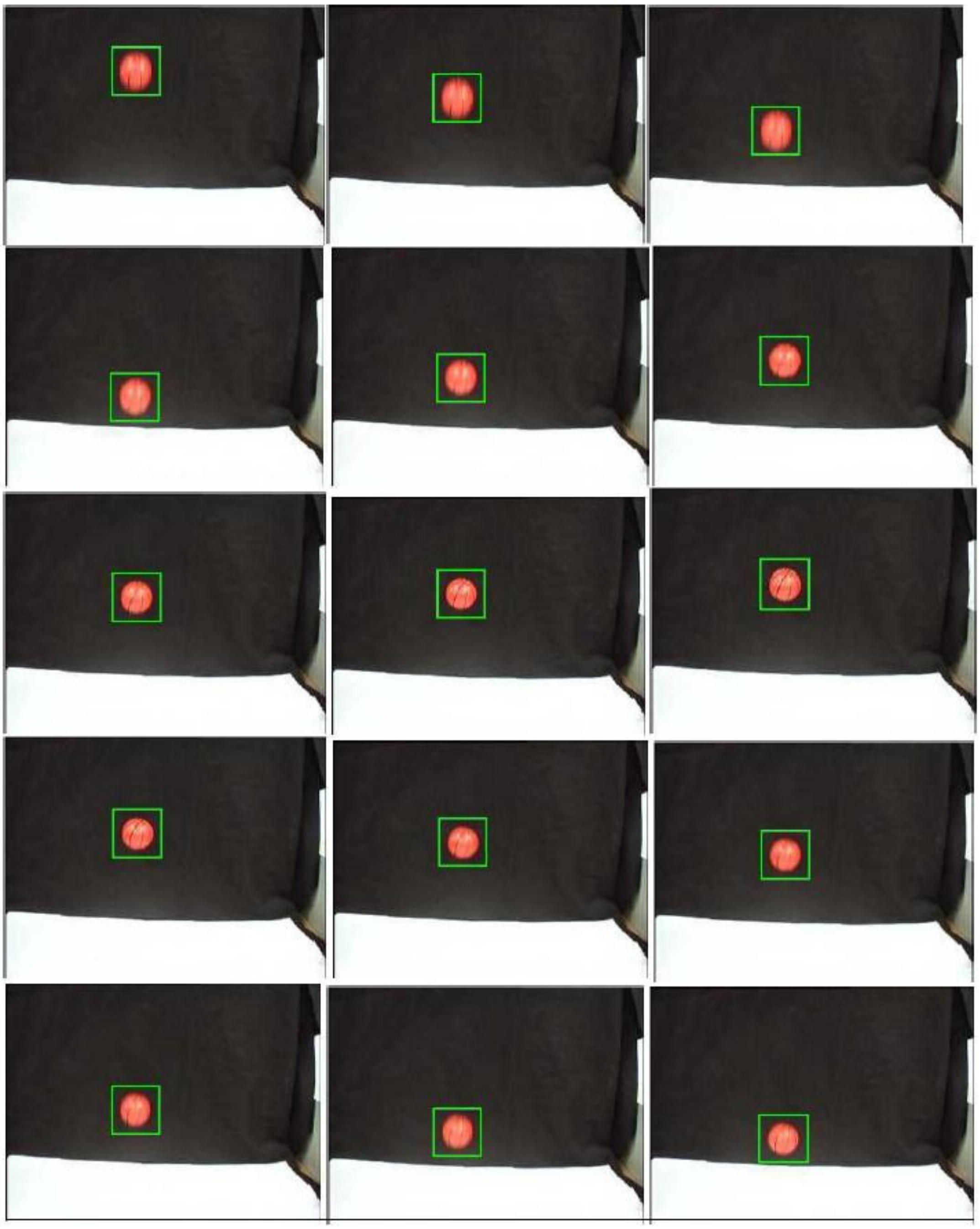


Fig.46. The object tracked images of ball video sequences using FCM-PSO

TABLE.V

EXTRACTED FEATURES FOR BALL VIDEO SEQUENCE

Object	Area	width	height	cenx	ceny	mv-x	mv-y
(8,1)	20976	61	316	207	160	0	0
(8,2)	948	32	34	22	135	18	2
(8,3)	3005	150	21	126	312	0	1

Object	area	width	height	cenx	ceny	mv-x	mv-y
(11,1)	20982	64	320	211	161	0	0
(11,2)	952	37	31	24	147	44	2
(11,3)	3021	151	24	129	310	0	0

TABLE.VI

THE EUCLIDEAN DISTANCE BETWEEN SUCCESSIVE FRAMES FOR BALL VIDEO

(Frame, Object)	(10,1)	(10,2)	(10,3)
(11,1)	**0**	174.3876	162.6476
(11,2)	182.9599	**18.0250**	259.4556
(11,3)	168.1856	192.4864	**0**

6.4.2. Tennis video sequence:

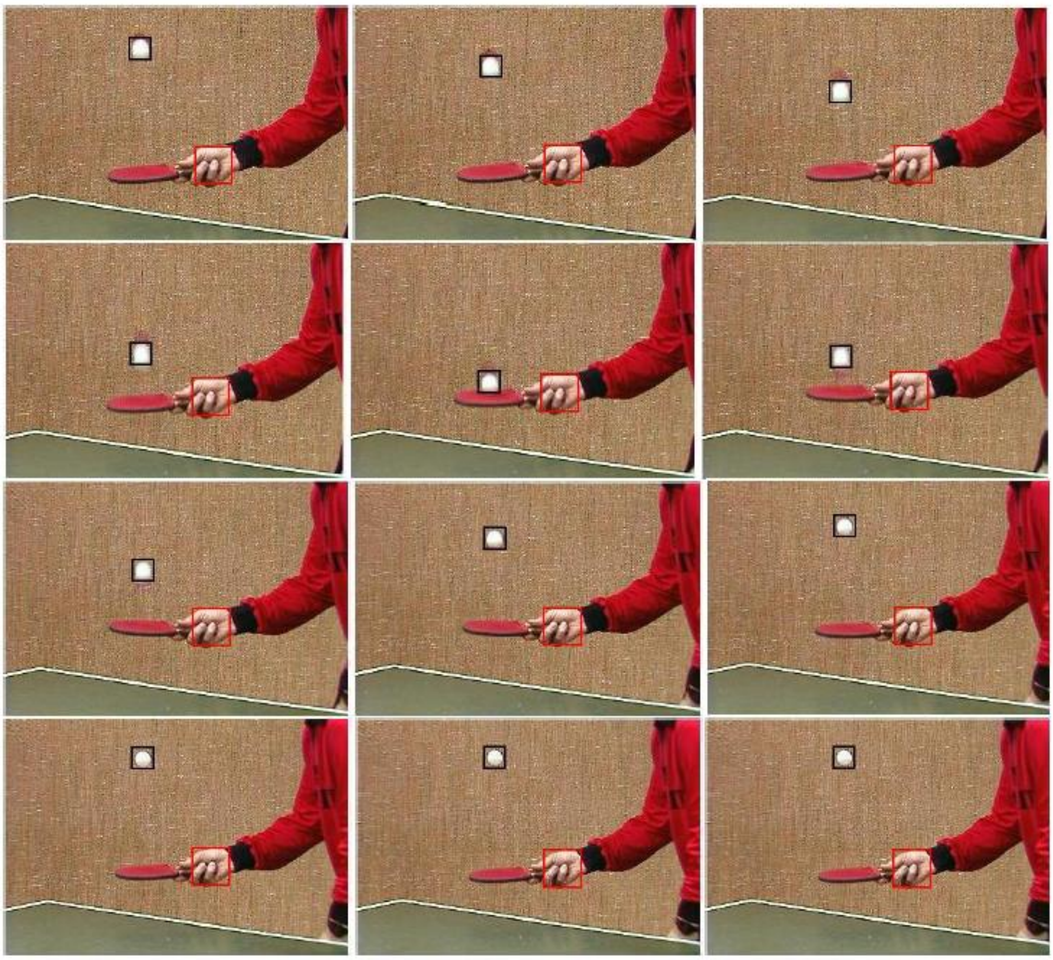


Fig.47. Object tracking using segmentation by FCM-PSO and pattern matching.

TABLE.VII

EXTRACTED FEATURES FOR TENNIS VIDEO SEQUENCE

(Frame, Object)	Area	Width	Height	Cenx	Ceny	mv-x	mv-y
(22,1)	16497	47	351	217	177	0	0
(22,2)	304	19	16	44	140	10	12
(22,3)	1550	25	62	155	210	26	55

(Frame, object)	Area	Width	Height	Cenx	Ceny	mv-x	mv-y
(23,1)	16497	47	351	217	177	1	0
(23,2)	340	20	17	62	142	10	9
(23,3)	1586	26	61	54	210	44	57

TABLE .VIII

THE EUCLIDEAN DISTANCE BETWEEN SUCCESSIVE FRAMES FOR TENNIS VIDEO

(Frame no, Object)	(23,1)	(23,2)	(23,3)
(22,1)	**1**	175.9346	69.3542
(22,2)	185.4562	**29.2373**	60.8353
(22,3)	185.7148	158.3824	**52.9528**

CONCLUSION:

We have proposed an object tracking algorithm for video pictures, based on image segmentation and pattern matching of the segmented objects between frames in a simple feature space. Simulation results for frame sequences with moving ball and Tennis video sequences verify the suitability of the algorithm for reliable moving object tracking. We also have confirmed that the algorithm works very well for more complicated video pictures including rotating objects and occlusion of objects. In order to extract color features of segmented objects, we used the gray value at the center pixel of an object. Thus, we cannot extract correct color features of an object that has gradation or texture. Nevertheless, gray value turns out to sufficiently represent the object's color features for the tracking purpose. A multicolored object would be segmented into several parts by the image segmentation algorithm. It would be recognized as a more complicated object through the identical movement of these parts.

There may be also the concern that the linear motion estimation is too simple and may fail for objects moving in a complicated nonlinear way. However, if the movement is not extremely fast, the deviation from estimated positions between successive frames is so small, that correct tracking is reliably achieved. Furthermore, if mistracking occurred at some frame by reason of occlusion, newly appearing or disappearing objects, the proposed algorithm could recover correct tracking after a couple of frames. This stability characteristic of the algorithm results from the fact that the object matching is performed in feature space between all objects in successive frames. It is sufficient for the tracking to use the simple Manhattan distance. The algorithm works even in the case of moving camera also, since the segmentation is done for each and every frame and different patterns in the image are viewed after segmentation. Pattern recognition males the algorithm more robust, such that the tracker will recover tracking if occlusion takes place.

FUTURE WORK:

The relative simplicity of this tracking algorithm promises that an FPGA implementation is possible and already sufficient for real time applications with a few moving objects. Thus, VLSI implementation of the algorithm is possible by using our developed architectures for image segmentation and a fully parallel associative memory with high-speed minimum Manhattan distance search, both of which have been already realized as VLSI circuits.

REFERENCES:

[1]. Takashi morimoto, Osama kiriyama, youmei harada,"Object tracking in video images based on image segmentation and pattern matching" IEEE conference proceedings, vol no 05, page no, 3215-3218, 2005

[2]. Qingming haung, wen gao, wenjian cai, "Thresholding technique with adaptive window selection for uneven lighting image", Elsevier, Pattern recognition letters, vol 26, page no 801-808, 2005

[3]. Junqiu wang [Member IEEE], yasushi yagi [Member IEEE],"Integrating colour and shape texture features for adaptive real time object tracking", IEEE transactions on image processing, vol no 17,no 2, page no 235-240,2007

[4]. Wei chen, Kangling fang, "A hybrid clustering approach using particle swarm optimization for image segmentation", ICALIP 2008, page no 1365-1368, 2008

[5]. Zhou Xian cheng, Shen Quan-tai, Liu li mei, "New two dimensional fuzzy C means clustering algorithm for image segmentation", Springerlink, vol no 15, page no 882-887, 2008.

[6]. E.Mahdizadeh, R. Tavakkoli-moghaddam, "A hybrid Fuzzy clustering PSO algorithm for a clustering supplier problem", Proceedings of IEEE IEEM 2007,vol no 7, page no 1466-1470, 2007

[7]. Weina wang, Yunjie zhang, Yi Li and xiaona zhang, "The global fuzzy C means clustering algorithm", Proceedings of 6th world congress on intelligent control and automation, june 21-23, dalian, china, vol no 6, page no 3604-3607, 2006.

[8]. Oleg Michailovich, Member, IEEE, And Allen Tannenbaum, Member, IEEE,"segmentation Of Tracking Sequences Using Dynamically Updated Adaptive Learning", IEEE Transactions on Image Processing, Vol. 17, No. 12, Page No 2403-2412, December 2008.

[9]. Wen-Yan Chang, Chu-Song Chen, and Yong-Dian Jian," Visual Tracking in High-Dimensional State Space by Appearance-Guided Particle Filtering", IEEE Transactions on Image Processing, Vol. 17, No. 7, page no 1154-1168, July 2008.

[10]. G.L. Foresti, "A real-time system for video surveillance of unattended outdoor environments," IEEE Trans. Circuits and Systems for Vid. Tech.,Vol. 8, No. 6, pp. 697–704, 1998.

[11]. C. Stauffer and W.E.L. Grimson, "Learning patterns of activity using realtime tracking," IEEE Trans. PAMIL, Vol. 22, No. 8, pp. 747–757, 2000.

[12]. S. W. Seol *et al.*, “An automatic detection and tracking system of moving objects using double differential based motion estimation,” Proc. of Int. Tech. Conf. Circ./Syst., Comput. and Comms. (ITC-CSCC2003), page no. 260– 263, 2003.

[13]. Wei Qu, Dan Schonfeld, ”Real-Time Decentralized Articulated Motion Analysis and Object Tracking From Videos”, IEEE Transactions on Image Processing, Vol.16, No.8, page no 2129-2139, August 2007.

[14]. D. M. Tsai and Y. H. Chen, “A fast histogram-clustering approach for multilevel thresholding,” Pattern Recognition Letters, Vol. 13, No. 4, pp. 245-252,1992.

[15]. N. Otsu, “A threshold selection method from gray-level histogram,” IEEE Transactions on System Man Cybernetics, Vol. SMC-9, No. 1,pp. 62-66,1979.

[16]. Lin Zhu, Fu-Lai Chung, And Shitong Wang,” Generalized Fuzzy C-Means Clustering Algorithm With Improved Fuzzy Partitions”, IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics, Vol. 39, No. 3, Page No 578-584, June 2009.

[17]. Mahamed G. H. Omran , Ayed Salman Andries P. Engelbrecht, “Dynamic clustering using particle swarm optimization with application in image segmentation”, Pattern Analysis Applications , page no 332–344, 2006.

[18]. Yuhua Zheng and Yan Meng,” Adaptive Object Tracking using Particle Swarm Optimization”, Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation, Jacksonville, FL, USA, June 20-23, Page no 43-49,2007.

[19]. J. Kennedy, R. C. Eberhart, and Y. Shi, “Swarm intelligence”, San Francisco, Morgan Kaufmann Publishers, 2001.

[20]. Lipton, H. Fujiyoshi, and R. Patil, “Moving Target Detection and Classification from Real-Time Video”, In Proceedings of IEEE Workshop on Application of Computer Vision, 1998,

[21]. J. Kennedy, R. C. Eberhart, Particle Swarm Optimization, in Proceeding of IEEE International Conference on Neural Networks, IEEE Press, Perth, WA, Vol. 4, pp. 1942-1948,1995.