ANALYSIS OF OBJECT DETECTION TECHNIQUE USING PARTICLE SWARM OPTIMIZATION

A Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

In

Electronic Instrumentation and Control

Submitted By

Sudhakar Singh

800951025

Under the Supervision of

Guide
Mr. Nirbhow Jap Singh
Assistant Professor, EIED
Thapar University, Patiala
Punjab, 147004

Co-Guide
Mr. Ankit Sharma
Assistant Professor, EIED
SVIET, Indore
M.P., 452001

Department of Electrical and Instrumentation Engineering

Thapar University

(Established under the section 3 of UGC act, 1956)

Patiala, 147004, Punjab, India

June 2011

DECLARATION

I hereby certify that the work presented in the thesis entitled, **"Analysis of Object Detection Technique using Particle Swarm Optimization"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in Electronics, Instrumentation and Control submitted in Electrical and Instrumentation Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of **Mr. Nirbhow Jap Singh** Assistant Professor, EIED Department of Electrical and Instrumentation Engineering, Thapar University, Patiala, Punjab and **Mr.Ankit Sharma** Assistant Professor, EIED Department of Electronics and Instrumentation Engineering Sri Vaishnav Institute of Engineering and Technology, Indore.

Singh

Date:

Sudhakar Singh
800951025

I certify that the above statement made by the student is correct to the best of my knowledge and belief.

Guide
Mr. Nirbhow Jap Singh
Assistant Professor, EIED
Thapar University, Patiala.
Punjab, 147004

Ankit Sharma

Co-Guide
Mr. Ankit Sharma
Assistant Professor, EIED
SVIET, Indore.
M.P., 452001

Countersigned By

S. Ghosh
14/7/11

Dr. Smarajit Ghosh
Head of Department, EIED
Thapar University, Patiala
Punjab, 147004

Dr. S. K. Mohapatra
Dean of Academic Affairs
Thapar University, Patiala
Punjab, 147004

DECLARATION

I hereby certify that the work presented in the thesis entitled, **“Analysis of Object Detection Technique using Particle Swarm Optimization’’**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in Electronics, Instrumentation and Control submitted in Electrical and Instrumentation Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of **Mr. Nirbhow Jap Singh** Assistant Professor, EIED Department of Electrical and Instrumentation Engineering, Thapar University, Patiala, Punjab and **Mr.Ankit Sharma** Assistant Professor, EIED Department of Electronics and Instrumentation Engineering Sri Vaishnav Institute of Engineering and Technology, Indore.

Date:

Sudhakar Singh
800951025

I certify that the above statement made by the student is correct to the best of my knowledge and belief.

Guide
Mr. Nirbhow Jap Singh
Assistant Professor, EIED
Thapar University, Patiala.
. Punjab, 147004

Co-Guide
Mr. Ankit Sharma
Assistant Professor, EIED
SVIET, Indore.
M.P., 452001

Countersigned By

Dr. Smarajit Ghosh
Head of Department, EIED
Thapar University, Patiala
Punjab, 147004

Dr. S. K. Mohapatra
Dean of Academic Affairs
Thapar University, Patiala
Punjab, 147004

ACKNOWLEDGEMENT

I am thankful to **Dr. Samarjit Ghosh**, Professor and Head, Electrical and Instrumentation Engineering Department, Thapar University, Patiala for providing us with adequate infrastructure for carrying out the work.

It is with the deepest sense of gratitude that I am reciprocating the magnanimity, which my guide **Mr. Nirbhow Jap Singh** Assistant Professor Electrical and Instrumentation Engineering Department, Thapar University, Patiala and co-guide **Mr**.**Ankit Sharma** Assistant Professor Electronics and Instrumentation Department, Sri Vaishnav Institute of Engineering and Technology, Indore have bestowed on me by providing individual guidance and support throughout the work. I am deeply indebted to him for help, stimulating suggestions and encouragement which helped me in all the time of writing this report.

I would also like to thank all the faculty members who were always there at the need of hour and provided with all the help and facilities, which I required for the completion of my report.

I am also thankful to the authors and researchers whose works I have consulted and quoted in this work.

Greatest thanks are to all who wished me success especially my parents my brothers and sisters. Above all I render my gratitude to the ALMIGHTY who bestowed self confidence, ability and strength in me to complete this work.

Sudhakar Singh
800951025

ABSTRACT

A key challenge in a surveillance system is the object detection task. Object detection in general is a non-trivial problem. From many years, among the object detection problems, the researchers have mainly focused on the problem of face detection in the field of image processing. Object detection is necessary, for guidance of autonomous vehicles, efficient video compression, for smart tracking of moving objects, for automatic target recognition (ATR) systems and for many other applications. Image matching is very important technique for wide range of applications, such as in guidance, navigation, robot vision, automatic surveillance, and in mapping sciences.

Numerous techniques have been proposed for object detection. Most biological vision systems have the talent to cope with changing world. Cross-correlation and related techniques have dominated the field since the early fifties. Conventional template matching algorithm based on cross-correlation requires complex calculation and large time for object detection, which makes difficult to use them in real time applications. In this thesis a different algorithmic approach having origin of biological principles is applied to detect the position of the selected object (part of an image). Applied algorithm performs better to detect the exact position of object when numbers of iteration are fixed but population size is limitedly increased.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Overview

Computer vision is fascinating topic. It is easy for a human to detect the positions of the letters, objects, numbers, etc. However, for a computer to solve these types of problems in fast manner is a very challenging task. Object detection is a fundamental component of artificial intelligence and computer vision. Object detection methods are used in various areas such as science, engineering, medical applications, etc. Interest in pattern recognition is fast growing in order to deal with the prohibitive amount of information, we encounter in our daily life. Object detection is necessary for surveillance applications, for guidance of autonomous vehicles, for efficient video compression, for smart tracking of moving objects, for automatic target recognition (ATR) systems and for many other applications. Automation is desperately needed to handle this information explosion.

In the last decades the computer's ability to perform huge amount of calculations, and handle information flows we never thought possible ten years ago has emerged. Despite this a computer can only extract little information from the image in comparison to human being. The way the human extracts useful information is not fully known and this skill has not been merged into computer vision science. The future goal of this thesis is to implement a system in MATLAB that is able to detection of object in an image in fast manner. The system should use both fast and advanced algorithms aiming to achieve the exact position of the object and time consumed in an image. The aim is to develop a system with the potential to be implemented in a real time environment. Therefore the system needs to be very fast.

This thesis is related to the broad subject of automatic detection of object in image. Automatically analyzing images and image sequences is the area of research usually called "computer vision". In order to be able to act intelligently a machine should be aware of its environment. The visual information is essential for humans. Finding targets in images involves two important steps: localization of candidates matches within the image, and each candidate is verified through a matching process. Most of the relevant literature in the field involves the task of matching. Image matching is necessary in mostly all of the image processing. Image matching has large numbers of applications which includes in guidance, navigation, robot vision, automatic surveillance, and in mapping sciences. Least Mean Square and related techniques are dominantly used in image matching applications. It is difficult to use this conventional template matching algorithm based on cross-correlation in real time applications due to requirement of complex calculation and large time for object detection applications. The shortcomings of this class of image matching methods have caused a slow-down in the development of operational automated correlation systems. To overcome this problem normalized cross–correlation method for image matching in object detection is implementing here.

The algorithm models the exploration of multi-dimensional solution space by a population of individuals where the success of each individual has an influence on the dynamics of other members of the swarm. It has been shown that the basic PSO is not guaranteed to converge to a local or global optimum. However, some optimization problems require the identification of global as well as local minima in a multimodal framework. In our method swarming is done at the classifier level in a space consisting of object location, scale, and classifier parameter dimensions and where each particle is a complete classifier. The particles swarm in this space in order to find the local optima that correspond to objects in the image. However, there have been attempts to use Genetic Algorithms (GAs) and Evolutionary Algorithms for object detection. The authors employ GA to detect and verify faces from images encoding only the position of the face. Object scale is handled by scaling the input image.

The proposed technique is substantially different from the previous work in that each particle from the population is a unique classifier. As the population swarms around, the classifiers adjust parameters to best detect the objects in the scene. Our method also differs from other vision algorithms that use swarm intelligence in that the other methods use swarms to build up features using ant colony pheromone based ideas.

The motive of this thesis is to study and implementation of particle Swarm optimization algorithm and algorithm for object detection in image and its variants. This thesis investigates the application of an optimization method, known as Particle Swarm Optimization, to the field of object detection and image processing. PSO solve optimization problems by simulating the social behavior of bird flocks.

1.2 Motivation

There are many difficult problems in the field of pattern recognition and image processing. These problems are the focus of much active research in order to find efficient approaches to address them. However, the outcome of the research is still unsatisfactory. Evolutionary algorithms are generally more suitable to solve these difficult problems because they are population-based stochastic approaches. Thus, evolutionary algorithms can avoid being trapped in a local optimum and can often find a global optimal solution. A PSO is a population-based stochastic optimization algorithm modeled after the simulation of the social behavior of bird flocks. PSO is easy to implement and has been successfully applied to solve a wide range of optimization problems. Thus, due to its simplicity and efficiency in navigating large search spaces for optimal solutions, PSOs are used in this research to develop efficient, robust and flexible algorithms to solve a selective set of difficult problems in the field of object detection and image processing.

1.3 Objective

In the presented work the particle swarm optimization algorithm is used to detect the position of selected object. The performance of PSO in terms of effectiveness and time taken to find the desired position will be analyzed. To further enhance the performance of basic PSO variations on the basis of cognitive behavior, social behavior, inertia weight, population size and number of iterations.

1.4 Thesis Outline

The whole of the work is divided into six chapters, the brief discussion are as follows. In chapter 1, a detailed introduction on object detection is given. In chapter 2, the brief literature review is done for different techniques for object detection. The chapter 3 deals with the basics of Digital Image Processing, object detection and pattern matching. The chapter 4 deals with the explanation of PSO technique with its terms, variants and algorithm. In chapter 5, the formulation of algorithm for object detection using PSO is explained in detail with the flow chart. In chapter 6, the result of object detection algorithm based on to detect the selected position of image and time consumed by proposed algorithm.

CHAPTER 2

LITERATURE REVIEW

Various techniques are proposed for detection of object in images due to its wide applications. Most of the schemes, which are pointed in literature, were more or less effective but their efficiency was greatly limited by either time or real-time implement ability. Most of these techniques also use prior heuristics about object geometry and some of them really based on very complex mathematical calculations which are almost not suitable for real-time implementation.

F. Ackermann [1]: A procedure for digital image correlation is described which is based on least squares window matching. The immediate aim was high precision parallax assessment, point transfer, and point measurement.

W. Forstner [2]: A new feature based correspondence algorithm for image matching is proposed. The interest operator is optimal for selecting points which promise high matching accuracy, for selecting corners with arbitrary number and orientation of edges or centers of disc, circles or rings. The similarity measure can take the seldomness of the selected points into account. The consistency of the solution is achieved by maximum likelihood type (robust) estimation for the parameters of an object model.

J. Bala et.al [3]: Address the problem of crafting visual routines for detection tasks. Emphasis was placed on both competition and learning to help with specific visual tasks involved in localization and identification. Crafting of visual routines presented difficult optimization problems and leads to evolutionary computation using a hybrid genetic architecture consisting of natural selection, learning, and their beneficial interactions.

Fan Mo et.al [4]: The proposed algorithm is characterized by its outstanding real time performance, making it the only applicable algorithm for object tracking this paper proposes a novel point patter matching among the existing ones.

C.F. Olson [5]: In image matching applications such as tracking and stereo matching, it is common to use the sum-of-squared differences (SSD) measure to determine the best match for an image template. However, this measure is sensitive to outliers and is not robust to template variations. He described a robust measure and efficient search strategy for template matching with a binary or grayscale template using a maximum likelihood formulation.

In addition to sub pixel localization and uncertainty estimation, these techniques allow optimal feature selection based on.

A. W. Gruent [6]: The Adaptive Least Squares Correlation is a very potent and flexible technique for all kinds of data matching problems, its application to image matching is outlined. The various tools of least squares estimation can be favorably utilized for the assessment of the correlation quality. Furthermore, the system allows for stabilization and improvement of the correlation procedure through the simultaneous consideration of geometrical constraints, e.g. the co linearity condition minimizing the localization uncertainty. He examined the use of these techniques for object recognition, stereo matching, feature selection, and tracking.

Kwan-Ho Lin et.al [7]: A new method for locating object based on valley field detection and measurement of fractal dimensions is proposed. Possible object position in an image with a complex background is identified by valley field detection.

Martin Berger [8]: Presented a generic matching algorithm suitable for many applications where feature extraction is difficult or inaccurate. Least squares template matching (LSM) is an area-based matching algorithm. It replaces the conventional multi-stage approach where feature detection is followed by thresholding, binarization and a discrete search. Thus, LSM does not depend on the extraction of binary (also called non-iconic) image features. This is a very important advantage especially in low-contrast and blurred image, where feature detection is mostly unreliable. Furthermore, unlike in most correlation methods, the optimum transformation is not searched by testing all possible cases, but approached using an optimization scheme. Assuming that a fair initial guess can be supplied, this was not only faster but also more accurate.

Wei-feng LIU et.al [9]: It is one challenge to select a general feature for object representation fixed the unconstrained videos. An object detection method which is robust to the target rotation and Scales is proposed based on the histogram feature and particle swarm optimization. First, the characters of histogram are presented, and then the merits of histogram feature are analyzed. To cover the computation problem of pixel by pixel searching, particle swarm optimization (PSO) is employed. Then the flowchart of target detection algorithm using histogram and PSO is described.

The experimental result proved that the histogram processes the merits of robustness and efficiency for target detection, and that the computation could be improved due to the performance of PSO.

R.M. Dufour, et.al [10]: Examined the problem of locating an object in an image when size and rotation are unknown. Previous work has shown that with known geometric parameters, an image restoration method can be useful by estimating a delta function at the object location. When the geometric parameters are unknown, this method becomes impractical because the likelihood surface to be minimized across size and rotation has numerous local minima and areas of zero gradients. They proposed a new approach where a smooth approximation of the template is used to minimize a well-behaved likelihood surface. A coarse-to-fine approximation of the original template using a diffusion-like equation is used to create a library of templates.

Kun Peng et.al [11]: Presented a robust eye detection algorithm for gray intensity images. The idea of their method was to combine the respective advantages of two existing techniques, feature based method and template based method, and to overcome their shortcomings.

Yuhua Zheng et.al [12]: This paper presents an automatic object detection and tracking algorithm by using particle swarm optimization (PSO) based method, which is a searching algorithm inspired by the behaviors of social insect in the nature. A cascade of boosted classifiers based on Haar-like features is trained and employed to detect objects. To improve the searching efficiency, first the object model is projected into a high-dimensional feature space, and the PSO-based algorithm is applied to search over this high-dimensional space and converge to some global optima, which are well-matched candidates in terms of object features. Then, a Bayes-based filter is used to identify the best match with the highest possibility among these candidates under the constraint of object motion estimation. The proposed algorithm considers not only the object features but also the object motion estimation to speed up the searching procedure. Experimental results of tracking on vehicle and face demonstrate that the proposed method is efficient and robust under dynamic environment.

Mukesh Motwani et.al [13]: Suggested a novel method which is robust and efficient in extracting objects using Wavelets and Neural Networks. Wavelet analysis is used as a pre-processor for a back propagation neural network with conjugate gradient learning.

The inputs to the neural network are the wavelet maxima neighborhood coefficients of images at a particular scale.

HawladerAbdullah Al-Mamun et.al [14]: As face recognition and facial feature based human computer interaction have become the subjects of intense focus in recent decades, facial feature extraction has emerged as a challenging task in the field of computer vision. Eye is said to be the most salient feature on face because of its versatility of appearance and expression variety. Various eye detection schemes are proposed in the literature but most of them require massive mathematical processing which is a barrier against real-time implementation. A novel approach for eye detection is proposed in this paper which exploits the flexibility of deformable template and uses genetic algorithm to match the template for eye detection. Implementation of genetic algorithm reduces the time required for template matching than conventional template matching algorithm. Moreover the method does not require any prior knowledge about eye geometry or potential eye location tags on facial image. Experimental results show that the proposed scheme can easily be implemented in real-time as it can detect eye in few genetic epochs. The method was tested on ORL face image database which contains 400 images grouped into 40 persons having 10 different expressions each. The detection accuracy was 87.2%.

Jian Wu1et.al [15]: This paper proposes a kind of video object tracking method based on normalized cross-correlation matching by using the high precision characteristics of normalized cross-correlation image matching. Firstly, extract video background from the temporal information of video. Then, acquire the region of moving object using background subtraction. Lastly, carry out related matching and updating towards the extracted moving object by means of normalized cross-correlation. Experimental result shows that the adaptability of our method is strong, which can well solve the tracking problems when tracking objects have scale transform. It also has good anti-interference ability and robustness, and can track moving objects accurately under the condition of noise interference, lens dithering and background mutation.

R.M. Dufour, et.al [16]: Examined the problem of locating an object in an image when size and rotation are unknown. Previous work has shown that with known geometric parameters, an image restoration method can be useful by estimating a delta function at the object location. When the geometric parameters are unknown, this method becomes impractical because the likelihood surface to be minimized across size and rotation has numerous local minima and areas of zero gradients. They proposed a new approach where a smooth approximation of the template is used to minimize a well-behaved likelihood surface. A coarse-to-fine approximation of the original template using a diffusion-like equation is used to create a library of templates. Using this library, they can successively perform minimizations which are locally well-behaved.

Feng Zhao, et.al [17]: Proposed the method of image matching by normalized cross-correlation. Correlation is widely used as an effective similarity measure in matching tasks. They proposed a new correlation based method for matching two images with large camera motion. Their method is based on the rotation and scale invariant normalized cross-correlation. Both the size and the orientation of the correlation windows are determined according to the characteristic scale and the dominant direction of the interest points.

Z.-H. Zhou, et.al [18]: The generalized projection function (GPF) is defined. Both the integral projection function (IPF) and the variance projection function (VPF) can be viewed as special cases of GPF. Another special case of GPF, i.e. the hybrid projection function (HPF), is developed through experimentally determining the optimal parameters of GPF. Experiments on three face databases show that IPF, VPF, and HPF are all effective in object detection.

Yacov Hel-Or, et.al [19]: A novel approach to pattern matching is proposed in which time complexity is reduced by two orders of magnitude compared to traditional approaches. The suggested approach uses an efficient projection scheme which bounds the distance between a pattern and an image window using very few operations on average. The projection framework is combined with a rejection scheme which allows rapid rejection of image windows that are distant from the pattern.

CHAPTER 3

OBJECT DETECTION IN IMAGE

3.1 Digital Image Processing

In today's world of advanced technology where most remote sensing data are recorded in digital format, virtually all image interpretation and analysis involves some element of digital processing. Digital image processing may involve numerous procedures including formatting and correcting of the data, digital enhancement to facilitate better visual interpretation, or even automated classification of targets and features entirely by computer. Many of the techniques of digital image processing, or digital picture processing as it was often called, were developed in the 1960s at the Jet Propulsion Laboratory, University of Maryland, Bell Labs, MIT, and a few other places, with application to satellite imagery, videophone, wire photo standards conversion,` medical imaging, , character recognition, and photo enhancement. But the cost of processing was fairly high with the computing equipment of that era. In the 1970s, digital image processing proliferated, when cheaper computers and dedicated hardware became available. Images could then be processed in real time, for some dedicated problems such as television standards conversion.

In order to process remote sensing image digitally, the data must be recorded and available in a digital form suitable for storage on a computer tape or disk. Obviously, the other requirement for digital image processing is a computer system, sometimes referred to as an image analysis system, with the appropriate hardware and software to process the data. Several commercially available software systems have been developed specifically for remote sensing image processing and analysis. is a computer system, sometimes referred to as an image analysis system, with the appropriate hardware and software to process the data. Several commercially available software systems have been developed specifically for remote sensing image processing and analysis. The fast computers and signal processors are available in the 2000s; digital image processing has become the most common form of image processing, and is generally used because it is not only the most versatile method, but also the cheapest.

Digital image processing is a subset of the electronic domain where in the image is converted to an array of small integers, called pixels, representing a physical quantity such as scene radiance, stored in a digital memory, and processed by computer or other digital hardware. Digital image processing, either as enhancement for human observers or performing autonomous analysis, offers advantages in cost, speed, and flexibility, and with the rapidly falling price and rising performance of personal computers it has become the dominant method in use.

3.1.1 Definition of an Image

The term image refers to a two-dimensional light intensity f(x, y). Where *x* and *y* denote spatial coordinates and the value of f at any point (x, y) is proportional to the brightness (or gray level) of the image at that point.

3.1.2 Digital Image

The term digital image refers to an image f(x, y) that has been discretized both in spatial coordinates and brightness. A digital image can be considered a matrix whose row and column indices identify a point in the image and the corresponding matrix element value identifies the gray level at that point. The elements of such a digital array are called pels (picture elements), or more commonly pixels. Images are built up of pixels that contain color information and are aligned with the cartesian coordinate system. The zero point is found at the top-left corner of the image (in PostScript, for example, the zero point is found at the bottom-left corner of the page). The image's width is represented by the variable N the image's height with the variable M as shown below.

3.2 Image Basics

The basic image characteristics include:

3.2.1 Image metrics

3.2.2 Image modes

3.2.3 Image histograms

3.2.1 Image metrics

Images are built up of pixels that contain color information and are aligned with the Cartesian coordinate system. The zero point is found at the top-left corner of the image in PostScript, for example, the zero point is found at the bottom-left corner of the page.

The image’s width & height are represented by variable X and Y. Figure 2.1a shows the coordinates of an image with the width and height of 11 × 8 pixels. The zero point in polar coordinates is found at the middle of the image. The two coordinate axes are the angle (or direction) and magnitude (or distance from the image’s center) and are represented by the variables d and m, respectively. Fig 3.1(b) shows the computed polar coordinates of the same image.

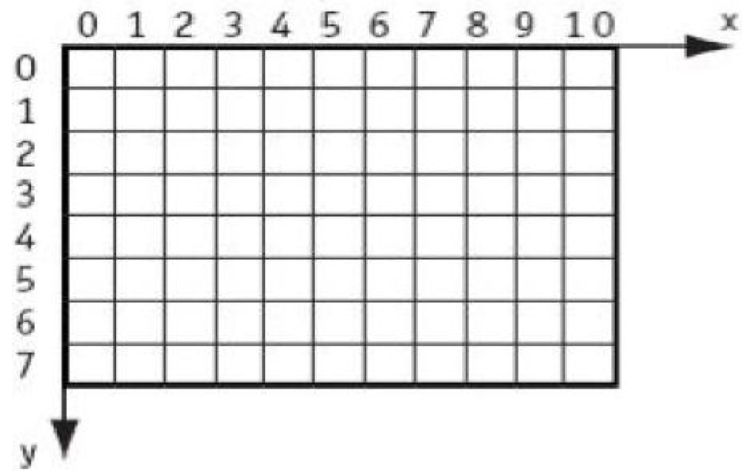

Fig 3.1: Pixels representation of an image.

3.2.2 Image Modes (color models)

When working with pixel-based images, one can work with black-and-white (bitmap), grayscale and colored images. The pixel information between these image modes is different. This also plays a part in the image file’s size and memory size needed.

a) Bitmap images

Bitmaps are black-and-white images, whose pixel information can only be black or white and nothing in between. One bit can have the information 0 (Black) or 1 (White) and nothing else, and that is enough to describe one pixel in a bitmap. The memory requirements for a 640 × 480 bitmap are 37.5 KB.

b) Grayscale images

In Grayscale images, a pixel is described by one byte or 8 bits. The image consists of eight bit planes in one channel. You can combine the 8 bits in up to 256 combinations.

So a pixel can, for example, have the following values: 0 (black), 64 (dark gray), 128 (gray), 192 (light gray) and 255 (white). The memory requirement for a 640 × 480 Grayscale image is 300 KB.

c) RGB images

Televisions, computer monitors, scanners and our eyes work with the emission respectively the absorption of Red, Green and Blue light rays. The combination of these three colors in different intensities can produce millions of different colors. These colors are actually light rays which have a certain frequency or wavelength. Imagine Fig. 3.2 as if one were in a dark room and were projecting three colored lamps onto the wall. When mixed together, the frequencies are added together. This is the additive color mixture. If you further imagine that each lamp can be set to 256 intensity settings, you could mix up to 256 × 256 × 256 = 16.7 million colors. Each color (red, green and blue) – in Photoshop called channels – is described thus in 8 bits or one byte. One pixel in your RGB-image is described by 3 × 8 bits = 24 bits. The memory requirements for a 640 x 480 RGB image are 900 KB.

Fig: 3.2 Projections of three lamps with the colors red, green and blue onto a wall in a dark room

A gray tone can be achieved by setting equal intensities of the three color channels:

(0, 0, 0) black

(128, 128, 128) gray

(192, 192, 192) light gray

(255, 255, 255) white

RGB images are mostly used for web graphic creation, presentations in general and CD-ROM productions.

d) CMYK Images

These images are used specially for print products such as booklets, magazines, catalogs, etc. Most CMYK images are converted after scanning (with the scanner's driver) or when converting images with other color spaces to CMYK.CMYK stands for the inks Cyan (blue),Magenta (pinkish),Yellow and Black. These colors are printed on media and are not emitted colors (such as from the monitor).Thus, light is absorbed from the ink on paper and our eyes see only the reflected light rays (see Fig. 3.3). Therefore, when nothing is printed, the media's color is seen, which in most cases paper white is. When Cyan is printed with Magenta, purplish Blue is obtained; Magenta with Yellow gets you Red and Yellow with Blue obviously Green. And when all colors are mixed together, all light rays are absorbed and you see a black area on the white paper. Since the pixel intensity in each channel is still eight bits or one byte, each CMYK pixel is described by 32 bits. The memory requirement for a 640 x 480 CMYK image is 1200 KB or 1.2 MB.

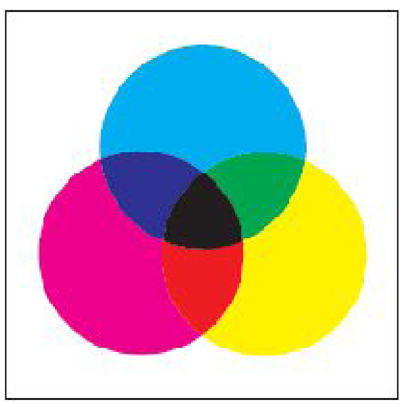

Fig: 3.3 A print of three colored circles (cyan, magenta and yellow) on paper

3.2.3 Image histogram

The histogram describes the relative color amount in a channel of an image. Since a pixel can have a value from 0 to 255, one can use the 256 get/put memory cells which contain the total amount of pixels having that certain pixel value. The visualization of a histogram is done with a bar graph.

3.3 Object Detection

Object detection attempts to determine the existence of specific objects in a set of images and, if present, to determine the locations, sizes and shapes of these objects. It is a challenging problem because objects can occur under different orientations, lighting conditions, backgrounds and clutter. It often utilises a trained binary classifier that can distinguish the objects of interest from the background (including objects of other classes). In computer vision, object detection and tracking is an active research area which has attracted extensive attentions from multi-disciplinary fields, and it has wide applications in many fields like service robots, surveillance systems, public security systems, and virtual reality interfaces. Detection and tracking of an object like car and people, eye ear are more concerned [7], especially flexible and robust tracking algorithms under dynamic environments, where lightening condition may change and occlusions may happen.

The general process of object detection consists of two steps. The first step is building models according to the prior knowledge of the interested objects, the feature model is built up to describe the target object and separate it from other objects and backgrounds. And since most images are noisy, statistic information are usually adopted to quantify features. The second step is to find a particular region in the image; called area of interest (AOI), which either can best fit the object model or has the highest similarity with the model. If the detection is executed successfully for each frame of an image sequence, it means that the object is tracked, where every frame is treated independently. Another category of detection algorithms takes advantage of correlations between image frames to accelerate tracking process. Basically, the features of the object itself are local information, and the features of an image sequence belong to global information. For instance, movement correlations include predicted object heading and velocity.

Extensive works have been conducted for object detection and tracking. Most available algorithms focus on estimating movements of AOI using probabilistic theories. Some popular models and approaches, like HMM (Hidden Markov Model), Kalman filter , condensation , and particle filter predict discrete probability distributions, while other algorithms, for example, mean shift methods study how to search the object model in a more robust manner.

For Kalman filter based methods, some researcher's proposed different control and noise models into the zrecursion function; however those assumptions are dependent on specific applications and need to be tuned carefully. Condensation and Particle filter methods mainly focus on how to sample probability and likelihood, so as to represent simultaneous alternative hypotheses of the object which could not be handled by Kalman filter. Generally speaking, the mean-shift, method, as a well-known kernel-based local searching algorithm, is efficient for object detection. However, the searching window may drift away from the object under dynamic conditions. For example, if the kernel is lost from the tracked target in one frame under some emergent situations, such as the changes in illumination condition, it would be difficult for the tracker to recover itself from this unpredicted event. Some other methods have also been proposed. Wren et al. proposed a Pfinder real time system for people tracking, where a multi-class statistical model of color and shape was proposed to segment people from background scenes. Viola and Jones proposed a boosted cascade algorithm, which is a learning method by combining a set of weak 0/1 classifiers to quickly detect objects. Olson and Brill built a general-purpose system for moving object detection and event recognition, where objects were detected and tracked by both first-order prediction and nearest neighbor matching. The system proposed in extracted moving targets from a real-time video stream, and classified them into pre-defined categories and tracked them. Tao et al. Tried to build configurations for each target to naturally handle appearance, disappearance and occlusion. Beleznail et. al adopted the fast mean shift to cluster and track people. In the Hydra system, a silhouette-based shape model and the correlation-based matching method were combined together to conduct classification.

3.4 Detection, Recognition, Tracking

The area of object detection and recognition has made significant progress in the last few years. Many algorithms developed recently in this area relate to human face detection and recognition due to its potential applications in security and surveillance. Yet, generic, reliable, and fast human face detection was, until very recently, impossible to achieve in real-time. The concepts involved in object detection, object recognition, and object tracking often overlap. Each of these computer vision techniques tries to achieve the following:

- Object Tracking: dynamically locates objects by determining their position in each frame.
- Object Detection: locate generic classes of objects in the image (such as faces).
- Object Recognition: classify specific objects in the image (such as a face that belongs to one individual, a certain printed character etc). In practice, the same methods with minor variations are used to achieve one of the three tasks presented above. The area of Object Detection and Recognition has made significant progress in the last few years. Many algorithms developed recently in this area relate to human face detection and recognition due to its potential applications in security and surveillance.

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 hands, torso and feet by using the Cardboard Model which represents relative positions and sizes of body parts.

There are different methods of object detection but, the proposed technique is based on template matching. On the surface; this may seem trivial; people are able to immediately detect objects, with little, if any thought.

3.5 Background

Detection of a known object of interest within a given image by a human observer may occur quickly and accurately when images are obtained under ideal conditions. However, the potential for the observer to overlook the object increases when the image is obtained under non-ideal conditions, such as when brightness and contrast changes visually blend the object into its surroundings [8]. Object edges that human observers use to define the boundary between an object and its surroundings become blurred or broken, which causes the observer to overlook the object. Unfavorable consequences may arise if the human operator fails to locate an object of interest in the event that the object were a potential target, such as an enemy aircraft, tank, or weapon system in military images, or a malignant mass or vascular obstruction in medical images. In addition, incorrectly assigning positive detection of enemy aircraft when the aircraft is an ally, or detecting a malignant mass when none is present may result in ill-fated consequences, such as friendly fire or unnecessary surgeries. For these reasons, studies in object detection and recognition remain in the forefront of image processing research and development.

With current technology, automatic target recognition (ATR) might provide the most advantageous solution for enemy target detection, and a great deal of research in the area of ATR continues. However, ATR methods do not offer 100% object detection accuracy, making it necessary to employ the human observer in the process. Rather than exclusively using autonomous decision-making algorithms, object detection algorithms are used to reduce information overload by offering fewer target candidates and to allow a second opinion for selected object verification by human observers. This can potentially reduce the number of bad choices and tragic results.

3.5.1 Information overload

Military information gatherers capture large numbers of images world-wide every second from sources such as Unmanned Aerial Vehicles (UAV) and satellite cameras. In fact, terabytes of image data are acquired, stored, and assessed by the military daily [9]. Scanning terabytes of images for potential enemy targets is incredibly time consuming and may overwhelm military image analysts. Similarly, more than 300 million medical images are taken in the form of radiographs or Computed Tomography (CT), Magnetic Resonance Imaging (MRI), and Positron Emission Tomography (PET) scans annually in hospitals and medical centers. Radiologists spend a substantial amount of time visually scanning each image for abnormalities. Software that is able to assist in the detection of objects of interest in military and medical images potentially saves valuable time and lives.

3.5.2 Second opinion

Instead of solely relying on technology to detect a potential enemy target, placing the human in the final decision-making process provides a second opinion about the target.

Objects having similar shapes, lines, or material surfaces may deceive object detection software. Allowing the human to interact and make final decisions based on the information provided by the software and prior knowledge about material identification, edge observation, and situation awareness decreases the likelihood of incorrectly identifying targets.

3.6 Standard terminology in computer-aided object detection

3.6.1 Histograms

The histogram of an image is a graphical representation of the frequency of pixel intensity occurrence in the image. In an 8-bit grayscale image, such as a scanned document, 256 grayscale values are possible. Another image acquisition method, for instance, a digital camera, might produce 16-bit images, which results in 65,536 possible grayscale values. A large peak in an image histogram identifies a group of common grayscale values, whereas a valley indicates that one or several adjacent grayscale values are less common in the image. In a color image, the individual red, green, and blue (RGB) color components can be similarly viewed as histograms.

Though the histogram represents every pixel within an image, spatial information about each pixel is lost, i.e., the number of pixels at a specific grayscale value is represented in the histogram, but the histogram does not provide the location of those pixels within the original image. Also, because the pixel location information is lost, an image cannot be regenerated from its histogram. Even though the histogram does not provide pixel locations, knowing the shape of the histogram may be valuable. In some instances, manipulating the histogram can provide a more visually pleasing image. For example, in histogram equalization, a transfer function is applied to the histogram that fills in some of the valleys and spreads out some of the peaks to utilize more of the available grayscale [11, 12]. However, since histogram equalization assigns new grayscale levels to some pixels, quantitative analysis is no longer possible. Histogram equalization also fails in several situations; its effectiveness diminishes when variation in contrast occurs across an image, as might be the case with partial cloud cover [13]. Adaptive histogram equalization considers the issue of contrast changes by applying similar transfer functions to the histogram of each pixel’s neighborhood.

Even though this method outperforms traditional (global) histogram equalization for low contrast images, there are at least two drawbacks associated with adaptive histogram equalization: the process is computationally intensive, and the result is an artificial representation of the original image.

3.6.2 Thresholding

Thresholding an image generates a black and white, or binary, image by defining one range of grayscale values as the foreground and the remaining grayscale values as the background. Localizing a specific range of grayscale values based on object intensity sets the object as the foreground and facilitates object detection. However, setting a fixed threshold for use across multiple images is ineffective in locating features because changes in brightness levels result in fixed threshold method failure. To circumvent the limitations and disadvantages of fixed threshold techniques, adaptive thresholding methods provide localized histogram equalization at each pixel that can accentuate subtle texture and detail in some cases. Adaptive thresholding techniques are generally suited to illumination normalization, textural enhancement, and anomaly detection [13]. However, time becomes the price paid for using pixel-by-pixel histogram equalization calculations of adaptive thresholding.

3.6.3 Edge detection

All object detection methods key on various important features of the target. Manmade objects tend to have straight edges, which make edge detection algorithms highly effective in many detection applications. An important feature of any edge detection algorithm is the preservation of the "useful structural information about object boundaries" [14]. John Canny, in 1986, defined three important criteria for an effective edge detection filter. The filter must return a low number of errors, maintain localized edge points, and only provide one response to a local edge [14].

3.6.4 Shape-based detection

As an additional method for the larger-scale project, a shape-based object detection method is under study. The first step in this method is object segmentation using a band-pass edge detection filter that, in this instance, consists of the difference of two low-pass Gaussian filtered images with sequentially incremented standard deviations.

Thresholding the difference image produces a binary image of edges and contours. Additional processing with morphological and logical operations eliminates specific unwanted features, for example regions that exceed predicted object size, and fill broken boundaries and holes. Next, objects of a specified shape are identified through a solidity measurement. Object solidity is determined as the ratio of the object ROI to the area of the smallest polygon required to enclose the object. This allows objects resembling a known polygon to be extracted from the image by implementing a solidity-based thresholding procedure. Another relatively new and popular shape-based detection method is active shape modeling (ASM). ASM uses a deformable model, or *atlas*, which is an elastic-like template that can be stretched or shaped to match target boundaries in an image.

To provide an automated system, the variability of feature shapes throughout a set of independently obtained images requires training models using images with identified landmark points to identify similar features [19]. ASM could potentially be an effective method of target detection for the overall scope of the project; however, because another shape-based method is being pursued by our lab as part of the larger project, we elected not to explore the ASM method at this time. Image processing methods that have the ability to extract features from images also fall into the shape-based detection category. The top hat filter, a common feature detection method that is a Fourier transform-based kernel, looks for areas of bright and dark pixels in definite patterns.

The kernel consists of a center region, or crown, and a larger outer region, or brim. If the difference in the brightest (or darkest) brim and crown pixels exceeds a chosen threshold, then the central pixel in the top hat is maintained, otherwise it is replaced by an average value of brightness from the brim. The elimination of undesirable pixels makes the top hat filter a noise suppression operator. As with all thresholding operations, choosing the appropriate difference threshold to retain the desired pixels becomes challenging with different images.

3.6.5 Pattern matching

In pattern matching, algorithms search for known textures, homogeneous patches, and spatial patterns of intensity values. Determining the difference between brightest and darkest values in an area will determine if the texture in an area is uniform, a small difference, or rough, a large difference [20]. Pattern matching by means of frequency analysis is another detection method that is under investigation for our larger-scale project. The pattern is representative of the object's geometric signature; for example, the known geometric signature found in a scaled model of a Scud missile launcher is a set of three or four parallel lines. These parallel lines are represented as peaks in the Fourier domain. Information such as the frequency of the pattern and the object orientation can thus be extracted using Fourier analysis. Analysis of images with a known source-to-object distance enables prediction of the edge width and spacing between edges so that a band-pass filter can be used to extract the Fourier peaks that correspond to the target edges. Unwanted noise is reduced by peak amplitude thresholding.

The aim of object detection is to establish a correspondence between objects or object parts in consecutive frames and to extract temporal information about objects such as trajectory, posture, speed and direction. Tracking detected objects frame by frame in video is a significant and difficult task. It is a crucial part of smart surveillance systems since without object tracking, the system could not extract cohesive temporal information about objects and higher level behavior analysis steps would not be possible. On the other hand, inaccurate foreground object segmentation due to shadows, reflectance and occlusions makes tracking a difficult research problem. We used an object level tracking algorithm in our system.

That is, it do not track object parts, such as limbs of a human, but it track objects as a whole from frame to frame. The information extracted by this level of tracking is adequate for most of the smart surveillance applications. The approach makes use of the object features such as size, center of mass, bounding box and color histogram which are extracted in previous steps between objects in consecutive frames. Furthermore, the detection algorithm detects object occlusion and distinguishes object identities after the split of occluded objects. By analyzing the object trajectory information, our tracking system is able to detect left and removed objects as well.

3.7 Image Matching Approaches

The definition for best match criteria obviously plays an important role in each matching algorithm, whether is image-based, feature-based or structural-based. In reality, there will not be an exact match since some part of the image content is usually corrupted in real images by noise, and distorted by geometric distortion, occlusion, illumination condition and others. Area-based matching work well and has a high accuracy potential when the image gradient surfaces are continuous and the image regions are well textured, which contain sufficient common feature (visual texture) to allow corresponding matches to be obtained. Mismatch and ambiguities may encounter when the images are of the scene of repetitive texture or does not contain adequate texture, or with many depth discontinuity. In addition, blunders can be occurring in image areas of occlusions and noise.

Area-based matching is weak in handling the sensitivity of the grey values to changes in radiometry due to illumination changes. However, in these situations, e.g. in the presence of image noise, area correlation degrades gracefully it usually continues to find the matching answer, but with reduced confidence measure and increased ambiguities. In addition, this method only works well under the condition that images are acquired geometrically alike. Area-based method encounters difficulties when the two images are taken from extremely different viewpoints. It requires a very good initial position of the two areas before the actual matching step is taking, to avoid large search space.

3.7.1 Proposed Template Matching

Template matching is a popular method for object detection. It is defined below:
Definition: Let I be an image of dimension m×n and T be another image of dimension p×q such that p<m and q<n then template matching is defined as a search method which finds out the portion in I of size p×q where T has the maximum cross correlation coefficient with it.
The normalized cross correlation coefficient is defined as:

$$\lambda(x,y) = \frac{\sum_s \sum_t \delta_{I(x+s,y+t)}\delta_{T(s,t)}}{\sum_s \sum_t \delta^2_{I(x+s,y+t)} \sum_s \sum_t \delta^2_{T(s,t)}} \tag{3.1}$$

Where,

$\delta_{I(x+s,y+t)} = I(x+s, y+t) - I(x,y)$, $\delta_{T(s,t)}$ = T(s,t)-T

S € {1, 2, 3,…..p}, and t € {1, 2, 3,…..q}.

X € {1, 2, 3,……m-p+1}, y € {1, 2, 3,……n-q+1}

$$I(x,y) = \frac{1}{pq}\sum_s \sum_t I(x+s, y+t) \tag{3.2}$$

$$T = \frac{1}{pq}\sum_s \sum_t T(s,t) \tag{3.3}$$

For template matching the template, T slides over I and γ is calculated for each coordinate (x, y). After completing this calculation, the point which exhibits maximum γ is referred to as the match point.

3.8 Model based object detection

The basic model based detection system begins some form of images formation where by 3D scene is reduced to a 2D image array, via an orthographic or perspective projection. Commonly used sensors which achieve this goal are the standard cameras, laser range finder and standard range finder and structured light range cameras. After preprocessing which accounts for the sensor noise and spurious data, the image is processed such that the pixels are grouped in a manner that reflects the objects in scene and the nature of their interrelationship.

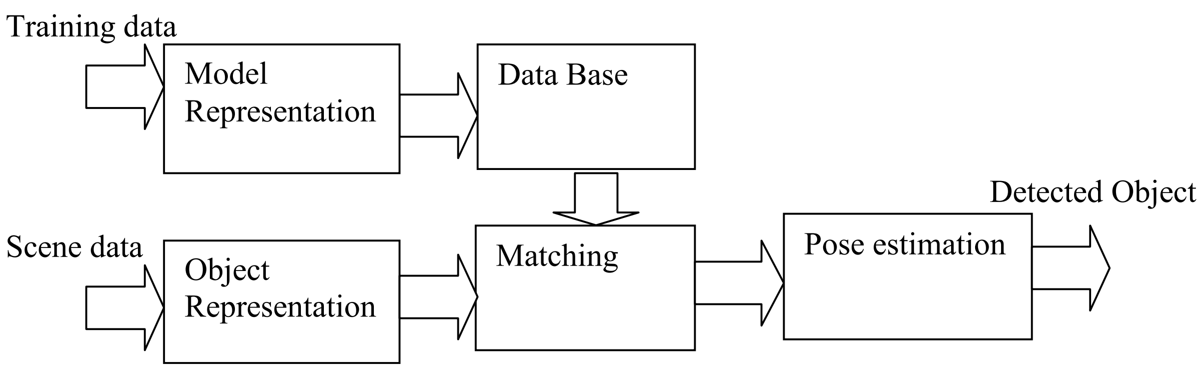

Fig: 3.4 Block diagram of object detection

CHAPTER 4
PSO TECHNIQUE

4.1 Introduction

Particle swarm optimization is a stochastic, population-based search and optimization algorithm for problem solving. It is a kind of swarm intelligence that is based on social-psychological principles and provides insights into social behavior, as well as contributing to engineering applications. The particle swarm optimization algorithm was first described in 1995 by James Kennedy and Russell C. Eberhart. The techniques have evolved greatly since then, and the original version of the algorithm is barely used at present. Social influence and social learning enable a person to maintain cognitive consistency. People solve problems by talking with other people about them, and as they interacts their beliefs, attitudes, and behavior changes, the changes could typically be depicted as the individuals moving toward one another in a socio-cognitive space.

The particle swarm optimization (PSO) algorithm is a population-based search algorithm inspired by the social behavior of birds within a flock. The initial intent of the particle swarm concept was to graphically simulate the graceful and unpredictable choreography of a bird flock, the aim of discovering patterns that govern the ability of birds to fly synchronously, and to suddenly change direction with a regrouping in an optimal formation. From this initial objective, the concept evolved into a simple and efficient optimization algorithm. In PSO, individuals, referred to as particles, are "flown" through hyper dimensional search space. Changes to the position of particles within the search space are based on the social-psychological tendency of individuals to emulate the success of other individuals.

The changes to a particle within the swarm are therefore influenced by the experience, or knowledge, of its neighbors. The search behavior of a particle is thus affected by that of other particles within the swarm therefore PSO is the kind of symbiotic cooperative algorithm. The consequence of modeling this social behavior is that the search process is such that particles stochastically return toward previously successful regions in the search space. The operation of the PSO is based on the neighborhood principle as social network structure. The particle swarm simulates a kind of social optimization.

A problem is given, and some way to evaluate a proposed solution to it exists in the form of a fitness function. A communication structure or social network is also defined, assigning neighbors for each individual to interact with a population of individuals defined as random guesses as the problem solutions is initialized. These individuals are candidate solutions and are also known as the particles, hence the name particle swarm. An iterative process to improve these candidate solutions is set in motion. The particles iteratively evaluate the fitness of the candidate solutions and remember the location where they had their best fitness value. The individual's best solution is called the particle best or the local best. Each particle makes this information available to their neighbors. They are also able to see where their neighbors have had best fitness value. Movements through the search space are guided by these successes, with the population usually converging, by the end of a trial, on a problem solution better than that of non-swarm approach using the same methods.

4.2 Flocks, Swarm and Particle

A number of scientists have created computer simulations of various interpretations of the movement of organisms in a bird flock or fish school. Notably, Reynolds and Heppner and Germander presented simulations of bird flocking. It became obvious during the development of the particle swarm concept that the neighbours of the population of agents are more like a swarm than a flock. The term swam has a basis in the literature. In particular, the authors use the term in accordance with a paper by Millons, who developed his models for applications in artificial life, and articulated five basic principles of swarm intelligence.

- First is the proximity principle: the population should be able to carry out simple space and time computations.
- Second is the quality principle: the population should be able to respond to quality factors in the environment.
- Third is the principle of diverse response: the population should not commit its activities along excessively narrow.
- Fourth is the principle of stability: the population should not change its mode of neighbour every time the environment changes.

- Fifth is the principle of ability: the population must be able to change the behaviour mode when it’s worth the computational price.

Note that principles four and five are the opposite sides of the same coin. Particle swarm optimization concept and paradigm presented seem to adhere to all five principles. Basic to the paradigm are n-dimensional space calculations carried out over a series of time steps. The population is responding to the quality factors local best. Further, liccvcs discusses particle systems consisting of clouds of primitive particles as models of diffuse objects such as clouds, fire and smoke. Thus the label the authors have chosen to represent the optimization concept is particle swarm.

4.3 Notion of Particle swarm Optimization

PSO concept is based on a metaphor of social interaction such as bird flocking [Fig.4.1a] and fish schooling [Fig.4.2b]. 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 multi-dimensional 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.

Particle swarm optimization is similar to a genetic algorithm in that the system is initialized with a population of random solutions. It is unlike a genetic algorithm, however, in that each potential solution is also assigned a randomized velocity, and the potential solutions, called particles, are then “flown” through hyperspace. Each particle keeps track of its coordinates in hyperspace which are associated with the best solution (fitness) it has achieved so far. (The value of that fitness is also stored.) This value is called P_{best} (local best).

Another “best” value is also tracked. The “global” version of the particle swarm optimizer keeps track of the overall best value, and its location, obtained thus far by any particle in the population; this is called G_{best} (global best). The particle swarm optimization concept consists of, at each time step, changing the velocity (accelerating) each particle toward its local best and global best.

Fig: 4.1 Bird Flocking

Fig: 4.2 Fish Schooling

Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward P_{best} and G_{best}. A “local” version of the optimizer is introduced in which, in addition to P_{best}, each particle keeps track of the best solution, called G_{best}, attained within a local topological neighbourhood of particles. Both the global and local versions are described in more detail below. Acceleration constant is also specified, but in the experience of the authors, is not usually varied among applications.

A new form of particle swarm optimizer is introduced which examines how changes in the paradigm affect the number of iterations required to meet an error criterion, and the frequency with which models cycle interminably around a non global optimum. Three versions were tested: the “G_{best}” model, in which every agent has information about the group’s best evaluation, and two variations of the “P_{best}” version, one with a neighbourhood of six, and one with a neighbourhood of two.

It appears that the original G_{best} version performs best in terms of median number of iterations to convergence, while the P_{best} version with a neighbourhood of two is most resistant to local minima. Particle swarm optimization is an extremely simple algorithm that seems to be effective for optimizing a wide range of functions. It is a mid-level form of a life or biologically derived algorithm, occupying the space in nature between evolutionary search, which requires eons, and neural processing, which occurs on the order of milliseconds. Conceptually, it seems to lie somewhere between genetic algorithms and evolutionary programming. It is highly dependent on stochastic processes, like evolutionary programming. The adjustment toward P_{best} and G_{best} by the particle swarm optimizer is conceptually similar to the crossover operation utilized by genetic algorithms. It uses the concept of fitness, as do all evolutionary computation paradigms.

The concept of particle swarm optimization is flying potential solutions through hyperspace, accelerating toward "better" solutions. Other evolutionary computation schemes operate directly on potential solutions which are represented as locations in hyperspace. Much of the success of particle swarms seems to lie in the agent's tendency to hurtle past their target. The stochastic factors allow thorough search of spaces between regions that have been found to be relatively good, and the momentum effect caused by modifying the extant velocities rather than replacing them results in overshooting, or exploration of unknown regions of the problem domain. Much further research remains to be conducted on this simple new concept and paradigm. The goals in developing it have been to keep it simple and robust, and it seems to have succeeded at that. The algorithm is written in a very few lines of code, and requires only specification of the problem and a few parameters in order to solve it.

4.4 PSO Neighborhood Topologies

Different neighborhood topologies have been investigated [22]. Two common neighborhood topologies are the star (or wheel) and ring (or circle) topologies. For the star topology one particle is selected as a hub, which is connected to all other particles in the swarm. However, all the other particles are only connected to the hub. For the ring topology, particles are arranged in a ring. Each particle has some number of particles to its right and left as its neighborhood. Recently, Kennedy and Mendes [22] proposed a new PSO model using a Von Neumann topology.

For the Von Neumann topology, particles are connected using a grid network (2-dimensional lattice) where each particle is connected to its four neighbor particles (above, below, right and left particles). Figure 4.1 illustrates the different neighborhood topologies. The choice of neighborhood topology has a profound effect on the propagation of the best solution found by the swarm. Using the global model the propagation is very fast (i.e. all the particles in the swarm will be affected by the best solution found in iteration *t*, immediately in iteration *t*+1). This fast propagation may result in the premature convergence problem.

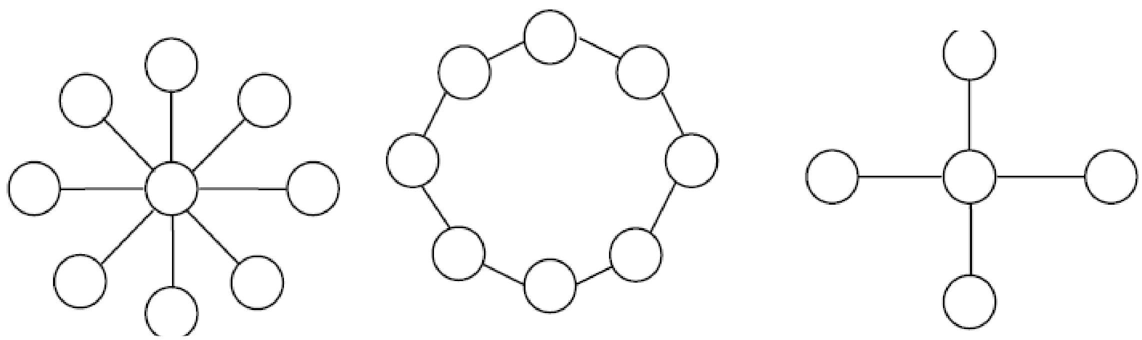

Fig: 4.3 (a) Star Topology (b) Ring Topology (c) Von Newman Topology

However, using the ring and Von Neumann topologies will slow down the convergence rate because the best solution found has to propagate through several neighborhoods before affecting all particles in the swarm. This slow propagation will enable the particles to explore more areas in the search space and thus decreases the chance of premature convergence.

4.5 Optimization

Optimization is the mechanism by which one finds the maximum or minimum value of a function or process. This mechanism is used in fields such as physics, chemistry, economics, and engineering where the goal is to maximize efficiency, production, or some other measure. Optimization can refer to either minimization or maximization; maximization of a function f is equivalent to minimization of the opposite of this function, -f . Mathematically, a minimization task is defined as:

Given f: R^n € R

Find ˆx € R^n such that f(ˆx) ≤f(x), ∨x € R^n

Similarly, a maximization task is defined as:
Given f: R^n € R

Find ˆx 2 R^n such that f(ˆx) ≤ f(x), ∨x ≤ R^n

The domain R^n of f is referred to as the search space (or parameter space [24]). Each element of R^n is called a candidate solution in the search space, with ˆx being the optimal solution. The value n denotes them number of dimensions of the search space, and thus the number of the function f is called the objective function, which maps the search space to the function space. Since a function has only one output, this function space is usually one-dimensional. The function space is then mapped to the one-dimensional fitness space, providing a single fitness value for each set of parameters. This single fitness value determines the optimality of the set of parameters for the desired task.

In most cases, including all the cases discussed in this paper, the function space can be directly mapped to the fitness space. However, the distinction between function space and fitness space is important in cases such as multi objective optimization tasks, which include several objective functions drawing input from the same parameter space [24, 25]. For a known (differentiable) function f, calculus can fairly easily provide us with the minima and maxima of f. However, in real-life optimization tasks, this objective function f is often not directly known. Instead, the objective function is a "black box" to which we apply parameters (the candidate solution) and receive an output value. The result of this evaluation of a candidate solution becomes the solution's fitness.

The final goal of an optimization task is to find the parameters in the search space that maximize or minimize this fitness. In some optimization tasks, called constrained optimization tasks, the elements in a candidate solution can be subject to certain constraints (such as being greater than or less than zero). For the purposes of this paper, we will focus on unconstrained optimization tasks. A simple example of function optimization can be seen in figure 4.4. This figure shows a selected region the function f, demonstrated as the curve seen in the diagram

. This function maps from a one-dimensional parameter space—the set of real numbers R on the horizontal x-axis to a one-dimensional function space— the set of real numbers R on the vertical y-axis.

The x-axis represents the candidate solutions, and the y-axis represents the results of the objective function when applied to these candidate solutions. This type of diagram demonstrates what is called the fitness landscape of an optimization problem. The fitness landscape plots the n-dimensional parameter space against the one-dimensional fitness for each of these parameters. Figure 4.4 also shows the presence of a local maximum in addition to the marked global maximum.

A local maximum is a candidate solution that has a higher value from the objective function than any candidate solution in a particular region of the search space. For example, if we choose the interval [0, 2.5] in figure 4.4, the objective function has a local maximum located at the approximate value x = 1.05.Many optimization algorithms are only designed to find the local maximum, ignoring other local maxima and the global maximum. However, the PSO algorithm as described to find the global maximum.

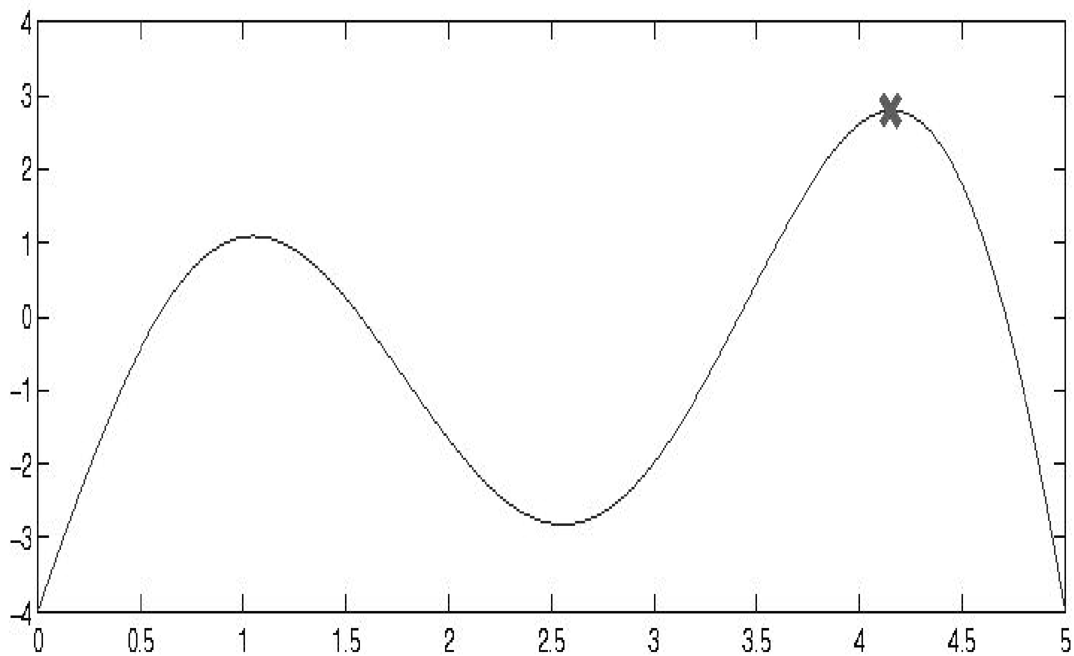

Fig: 4.4 The objective function has a local maximum located

The PSO algorithm works by simultaneously maintaining several candidate solutions in the search space. During each iteration of the algorithm, each candidate solution is evaluated by the objective function being optimized, determining the fitness of that solution. Each candidate solution can be thought of as a particle "flying" through the fitness landscape finding the maximum or minimum of the objective function. Initially, the PSO algorithm chooses candidate solutions randomly within the search space. Figure 4.5 shows the initial state of a four-particle PSO algorithm seeking the global maximum in a one-dimensional search space. The search space is composed of all the possible solutions along the x-axis; the curve denotes the objective function. It should be noted that the PSO algorithm has no knowledge of the underlying objective function, and thus has no way of knowing if any of the candidate solutions are near to or far away from a local or global maximum.

The PSO algorithm simply uses the objective function to evaluate its candidate solutions, and operates upon the resultant fitness values. Each particle maintains its position, composed of the candidate solution and its evaluated fitness, and its velocity. Additionally, it remembers the best fitness value it has achieved thus far during the operation of the algorithm, referred to as the individual best fitness, and the candidate solution that achieved this fitness, referred to as the individual best position or individual best candidate solution.

Finally, the PSO algorithm maintains the best fitness value achieved among all particles in the swarm, called the global best fitness, and the candidate solution that achieved this fitness, called the global best position or global best candidate solution.

The PSO algorithm consists of just three steps, which are repeated until some stopping condition is met [26]:

1. Evaluate the fitness of each particle
2. Update individual and global best fitnesses and positions
3. Update velocity and position of each particle

Below from figure we can see how the solution point move from one position to another position and how the fitness is changing in given example. From it is clear that the position of particle is changing in very fast manner, and providing optimal solution. The velocity and position update step is responsible for the optimization ability of the PSO algorithm. The velocity of each particle in the swarm is updated using the following equation:

$$V_i(t + 1) = W*V_i(i) + C_1*r1[\hat{}X(i) - X_i(i)] + C_2*r2[G(i) - X_i(i)] \qquad (4.1)$$

The index of the particle is represented by i. Thus, V_i (t) is the velocity of particle i at time t and $X_i(i)$ is the position of particle i at time t.

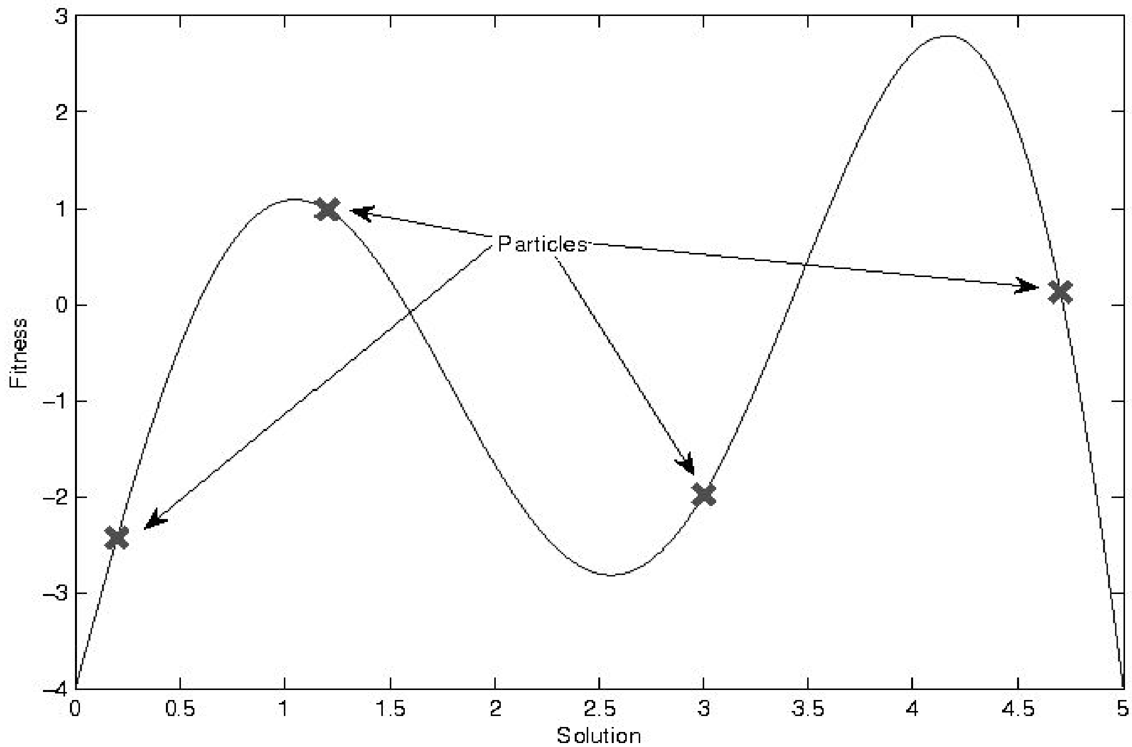

Fig 4.5: Initial state of a four-particle

The parameters W, C_1, and C_2 ($0 \leq w \leq 1.2$, $0 \leq C_1 \leq 2$, and $0 \leq C_2 \leq 2$) are user-supplied coefficients. The values r1 and r2 ($0 \leq r1 \leq 1$ and $0 \leq r2 \leq 1$) are random values regenerated for each velocity update. The value $\hat{}X_i(t)$ is the individual best candidate solution for particle i at time i, and G(i) is the swarm's global best candidate solution at time i. Each of the three terms of the velocity update equation has different roles in the PSO algorithm. The first term $W*V_i(i)$ is the inertia component, responsible for keeping the particle moving in the same direction it was originally heading. The value of the inertial coefficient W is typically between 0.8 and 1.2, which can either dampen the particle's inertia or accelerate the particle in its original direction [26]. Generally, lower values of the inertial coefficient speed up the convergence of the swarm to optima, and higher values of the inertial coefficient encourage exploration of the entire search space. The second term called the cognitive component, acts as the particle's memory, causing it to tend to return to the regions of the search space in which it has experienced high individual fitness. The cognitive coefficient C_1 is usually close to 2, and affects the size of the step the particle takes toward its individual best candidate solution $\hat{}X_i$. The third term called the social component causes the particle to move to the best region the swarm has found so far. The social coefficient C_2 is typically close to 2, and represents the size of the step the particle takes toward the global best candidate solution G(i) the swarm has found up until that point. The random values r1 in the cognitive component and r2 in the social component cause these components to have a stochastic influence on the velocity update.

This stochastic nature causes each particle to move in a semi-random manner heavily influenced in the directions of the individual best solution of the particle and global best solution of the swarm. In order to keep the particles from moving too far beyond the search space, we use a technique called velocity clamping to limit the maximum velocity of each particle [27]. For a search space bounded by the range $[-X_{max}, X_{max}]$, velocity clamping limits the velocity to the range $[-V_{max}, V_{max}]$, where $V_{max} = k \times X_{max}$. The value k represents a user-supplied velocity clamping factor, $0.1 \leq k \leq 1.0$. In many optimization tasks, such as the ones discussed in the paper, the search space is not centered on 0 and thus the range $[-X_{max}, X_{max}]$ is not an adequate definition of the search space. In such a case where the search space is bounded by $[X_{min}, X_{max}]$, we define $V_{max} = k \times (X_{max} - X_{min})/2$. Once the velocity for each particle is calculated, each particle's position is updated by applying the new velocity to the particle's previous position:

$$X_i(i + 1) = X_i (i) + V_i(i + 1) \tag{4.2}$$

This process is repeated until some stopping condition is met. Some common stopping conditions include: a preset number of iterations of the PSO algorithm, a number of iterations since the last update of the global best candidate solution, or a predefined target fitness value.

4.6 Particle Swarm Optimization Terms

A swarm consists of a set of particles, where each particle represents a potential solution. Particles are then flown through the hyperspace, where the position of each particle is changed according to its own experience and that of its neighbors. Let $X_j(i)$ denotes the position of particle p_j in search space, at time step i.

The position of P_j is then changed by adding a velocity $V_j(i)$ to the current position. The velocity vector drives the optimization process and reflects the socially exchanged information.

4.6.1 Individual Best (P_{best})

The local best reflects the circle neighborhood structure. Particles are influenced by the best position within their neighborhood, as well as their own past experience. Individual best is also called as local best. While local best is slower in convergence than global best, local best results in much better solution and searches a larger part of the search space. The further away a particle is from its previously found best solution, the larger the change in velocity to return the individual toward its best solution. The upper limit of the random value positions is a system parameter specified by the user. The larger the upper limit of positions, the more the trajectory of the particles oscillates. Smaller the value of positions ensures smooth trajectories.

4.6.2 Global Best (G_{best})

The global best, G_{best}, of PSO reflects the star neighborhood structure. The social knowledge used to drive the movement of particles includes the position of the best particle from the entire swarm. In addition, each particle uses its history of experiences in terms of its own best solution thus far. The further away a particle is from the global best position and its own best solution, the larger the change in velocity to move the particle back toward the best solutions.

4.6.3 Convergence

The algorithms above continue until convergence has been reached. Usually, a PSO algorithm is executed for a fixed number of iterations, or fitness function evaluations. Alternatively, a PSO algorithm can be terminated if the velocity changes are close to zero for all the particles, in which case there will be no further changes in particle positions.

4.7 PSO System Parameters

PSO has shown to perform better on higher-dimensional problems. Standard PSO is influenced by different system parameters, namely the dimension of the problem, number of individuals and inertia weight. The influence of the upper limit has been discussed previously. Other system parameters are discussed below.

4.7.1 Dimension of the Problem

Dimension of the problem deals with the variable or number of particle, as the dimension of the problem increases the complexity also increases.

4.7.2 Number of Individuals

The number of individual refers to the population size and it depends upon the dimension of the problem.

4.7.3 Inertia weight

Improved performance can be achieved through application of an inertia weight applied to the previous velocity:

$$V_j(i) = W * V_j(i - 1) + C_1(P_{best} - X_j(i)) + C_2(G_{best} - X_j(i)) \tag{4.3}$$

Where W is the inertia weight. The inertia weight controls the influence of previous velocities on the new velocity. Large inertia weights cause larger exploration of the search space, while smaller inertia weights focus the search on a smaller region. Typically, PSO is started with a large inertia weight, which is decreased over time. Here from fig we can see the how the particle improves their position and updated their corresponding global best and local best.

4.8 Stochastic Algorithms

Stochastic search algorithms are used to find near-optimal solutions for NP-hard problems in polynomial time. This is achieved by assuming that good solutions are close to each other in the search space. This assumption is valid for most real world problems [29]. Since the objective of a stochastic algorithm is to find a near-optimal solution, stochastic algorithms may fail to find a global optimal solution.

While an exact algorithm generates a solution only after the run is completed, a stochastic algorithm can be stopped any time during the run and generate the best solution found so far [29]. Stochastic search algorithms have several advantages compared to other algorithms [38]:

- Stochastic search algorithms are generally easy to implement.
- They can be used efficiently in a multiprocessor environment.
- They do not require the problem definition function to be continuous.
- They generally can find optimal or near-optimal solutions.
- They are suitable for discrete and combinatorial problems.

Three major stochastic algorithms are Hill-Climbing [39], Simulated Annealing [40] and Tabu search [41]. In Hill-Climbing, a potential solution is randomly chosen. The algorithm then searches the neighborhood of the current solution for a better solution. If a better solution is found, then it is set as the new potential solution. This process is repeated until no more improvement can be made. Simulated annealing is similar to Hill-Climbing in the sense that a potential solution is randomly chosen. A small value is then added to the current solution to generate a new solution.

If the new solution is better than the original one then the solution moves to the new location. Otherwise, the solution will move to the new location with a probability that decreases as the run progresses. Tabu search is a heuristic search algorithm where a tabu list memory of previously visited solutions is maintained in order to improve the performance of the search process. The tabu list is used to "guide the movement from one solution to the next one to avoid cycling" [45], thus, avoid being trapped in a local optimum. Tabu search starts with a randomly chosen current solution. A set of test solutions are generated via moves from the current solution. The best test solution is set as the current solution if it is not in the tabu list, or if it is in the tabu list, but satisfies an aspiration criterion. A test solution satisfies an aspiration criterion if it is in the tabu list and it is the best solution found so far [48]. This process is repeated until a stopping criterion is satisfied.

4.9 Basic Algorithm of PSO

The following steps are used by the PSO technique.

1. Initialize the parameters such as the size of population, lower limit, upper limit, inertia weight, random velocity and position of each particle and acceleration constant etc.

2. Calculate the fitness of each individual population using the fitness function or cost function

3. Compute local best and global best fitness, corresponding note down their positions.

4. Modify the individual's velocity $V_j(i)$ of each individual $P_j(i)$ and weight for each population.

$$W = Wmax - (Wmax - Wmin) * it/iteration \tag{4.4}$$

Modify the individual's velocity $V_j(i)$ as

$$V_j(i) = V_j(i-l) + C_1(P_{best(j)} - X_j(i)) + C_2(G_{best} - X_j(i)) \tag{4.5}$$

Modify the individual's position P_j as

$$P_j(i) = P_j(i-1) + V_j(i) \tag{4.6}$$

6. Calculate the fitness of each individual in the population using the fitness function. After that compares these fitness values with each other.

7. Minimum value of fitness will be the global best fitness.

8. Again calculate the new velocity and position by equations (4.5) and (4.6).

9. If the number of iteration reaches the maximum then go to step 10. Otherwise go to step 3.

10. The individual that generates the latest is the optimal position of each particle with the minimum fitness.

11. Stop

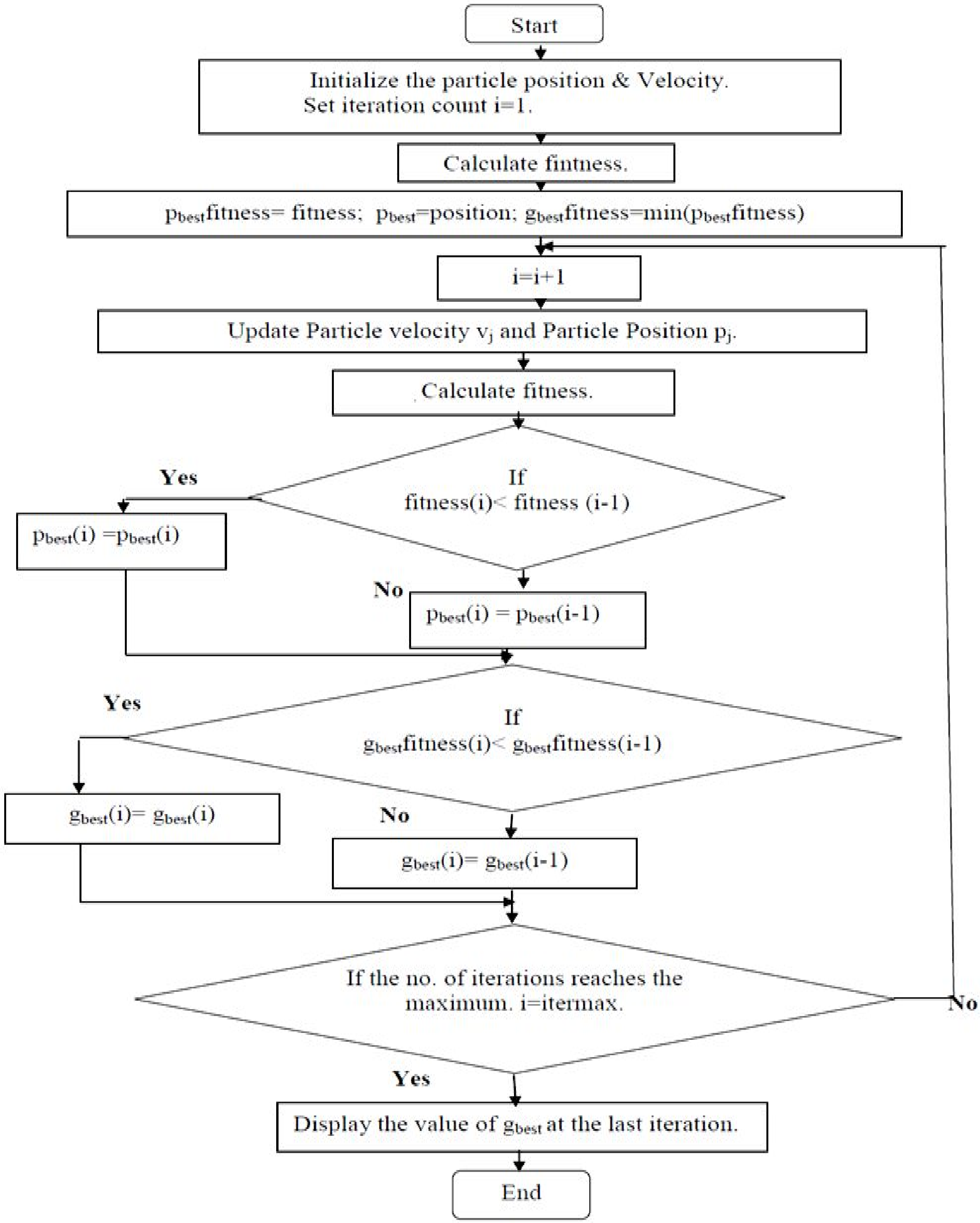

Fig: 4.6 The flow chart of PSO

CHAPTER 5

OBJECT DETECTION IN IMAGE BY PSO TECHNIQUE

5.1 Introduction

There are many methods to detect objects in image but in the presented work, the concept of particle swarm optimization is applied. The algorithm for detection of object is discussed in detail in this chapter. Here the parameters of the image (Height & Width) are behaving like dimension of the system, in PSO. The positions or points in the image, acting as solution of detection problem, are generated randomly between the upper and lower limit. After that, these values are updated till the desired optimization level is reached. Here the objective of detection of object in image is achieved by PSO (Particle Swarm Optimization) and its variants. The formulation and algorithm is simple and required less iteration as well as time and thus poised to overcome various shortcomings of traditional and other methods of object detection.

5.2 Problem Formulation

The detailed introduction to search methods is given in chapter 4, in this chapter the application of search techniques to image processing problem is considered. The objective of problem under consideration is to detect the point of maximum correlation between the template or cropped object (a part of image whose position is to be detected) and complete image, on which the position of the object (template) is to be determined. In short, the overall intention is to find out the point of maximum correlation between the cropped object and image on which position of object is to be determined.

5.2.1 Template Matching and Cross-correlation Coefficient

Template matching is a popular method for pattern recognition. It is defined below:

Definition: Let I be an image of dimension m×n and T be another image of dimension p×q such that p<m and q<n then template matching is defined as a search method which finds out the portion in I of size p×q where T has the maximum cross correlation coefficient with it. The normalized cross correlation coefficient is defined as:

$$\lambda(x, y) = \frac{\sum_s \sum_t \delta_{I(x+s,y+t)} \delta_{T(s,t)}}{\sum_s \sum_t \delta^2_{I(x+s,y+t)} \sum_s \sum_t \delta^2_{T(s,t)}} \tag{5.1}$$

Where,

$\delta_{I(x+s,y+t)} = I(x+s, y+t) - I(x,y)$, $\delta_{T(s,t)}$ = T(s,t)-T

S € {1, 2, 3...p}, and t € {1, 2, 3,…..q}.

X € {1, 2, 3 …m-p+1}, y € {1, 2, 3,……n-q+1}

$$I(x,y) = \frac{1}{pq}\sum_s \sum_t I(x+s, y+t) \tag{5.2}$$

$$T = \frac{1}{pq}\sum_s \sum_t T(s,t) \tag{5.3}$$

The value of cross-correlation coefficient γ ranges in [-1, +1]. A value of +1 indicates that T is completely matched with I(x, y) and -1 indicates complete disagreement. For template matching the template, T slides over I and γ is calculated for each coordinate (x, y). After completing this calculation, the point which exhibits maximum γ is referred to as the match point.

5.3 PSO Technique for Object Detection

The Particle swarm optimization PSO is a population based searching algorithm. This approach simulates the simplified social system such as fish schooling and birds flocking. PSO is initialized by a population of potential solutions called particles. Each particle flies in the search space with a certain velocity. The particle's flight is influenced by cognitive and social information attained during its exploration. It has very few tunable parameters and the evolutionary process is very simple. It is capable of providing quality solutions to many complex power system problems. One such problem is the unit commitment of thermal units in the power system. PSO is used to minimize the total operating cost by committing those optimal combinations of the units which satisfy the constraints and gives the minimum cost corresponding to that combination. Our main aim is to minimize the calculation effort and time required in traditional template matching algorithm.

5.3.1 Algorithms

The Object detection algorithms are developed by using three behaviorally different search techniques, first using PSO algorithm, then using PPO algorithm and then using PSO algorithm with the concept of regenerating the population.

The following steps are used by the PSO technique to solve the object detection problem:

1. Initialize a population of particles P_j and other variables. Each particle is usually generated randomly within allowable range. Here P_j represented as j_{th} random point in image.

$$\text{upper limit} \geq P_j \geq \text{lower limit}$$

Here p_j represented as j_{th} random point in image.

2. Initialize the parameters such as the size of population, inertia weight, random velocity of particle etc.

3. Calculate the fitness of each individual in the population using the fitness function

$$\lambda(x,y) = \frac{\sum_s \sum_t \delta_{I(x+s,y+t)} \delta_{T(s,t)}}{\sum_s \sum_t \delta^2_{I(x+s,y+t)} \sum_s \sum_t \delta^2_{T(s,t)}} \tag{5.4}$$

Where,

$\delta_{I(x+s,y+t)} = I(x+s, y+t) - I(x,y)$, $\delta_{T(s,t)}$ = T(s,t)-T

S € {1, 2, 3…p}, and t € {1, 2, 3,…..q}.

X € {1, 2, 3 …m-p+1}, y € {1, 2, 3,……n-q+1}

$$I(x,y) = \frac{1}{pq} \sum_s \sum_t I(x+s, y+t) \tag{5.5}$$

$$T = \frac{1}{pq} \sum_s \sum_t T(s,t) \tag{5.6}$$

4. Compare each individual's fitness value with its P_{best}. The best fitness value among P_{best} is denoted as G_{best}.

5. Modify the individual's velocity v_j of each individual p_j as

$$V_j(i) = v_j(i-1) + C_1 (P_{best(j)} - P_j(i)) + C_2(G_{best} - P_j(i)) \tag{5.7}$$

6. Modify the individual's position P_j

$$P_j(i) = P_j(i-1) + V_j(i) \tag{5.8}$$

Where i is the j_{th} unit and i is for iteration.

7. If the evaluated value of each individual is better than the previous P_{best}, the current value is set to be P_{best}. If the best P_{best} is better than G_{best} the value is set to be G_{best}.

8. Maximize the fitness function using PSO method for the number of points generated at that time.

9. If the number of iteration reaches the maximum then go to step 10. Otherwise go to step 3.

10. The individual point that provides the exact position of template on image I obtained from G_{best}.

CHAPTER6

RESULT AND DISSCUSSION

6.1 Introduction

The previous chapters that have been studied provide the complete knowledge of object detection problem and its formulation using search methodologies. The algorithms of particle swarm optimization approach which are presented in chapter 4, have been applied for solving object detection problem. The performance has been studied for different images and different templates. The results are discussed as –

6.2 Test Images and Templates

Different test images and templates on which the developed algorithms are tested are shown below:

1. In the first table test images are tested by varying the population size and keep unchanging iterations.
2. In the second table test images are tested by changeable iterations and keep invariable population size.

(a)

(b) (c) (d)

Fig: 6.1 (a) Test mages (b) Template 1 (c) Template 2 (d) Template 3

(a)

(b)

(c)

Fig: 6.2 (a) Test Image 2 (b) Template 1 (c) Template 2

(a)

(b)

(c)

Fig : 6.3 (a) Image (b) Template 1 (c) Template 2

(a) (b)

Fig 6.4: (a) Test Image 4 (b) Template 1

(a) (b)

Fig: 6.5 (a) Test Image 5 (b) Template 1

Table1

Images	Template	Population size	Position of Template	Searched Position	Time (sec)	Error
Image 1	Template1	30	161 182	162 180	22.2187	2.2360
		40		161 182	28.2371	0.0000
		50		160 181	31.1073	1.4142
	Template2	30	273 211	273 211	10.8147	0.0000
		40		272 211	17.9928	1.0000
		50		273 211	24.6683	0.0000
	Template3	30	236 229	236 230	10.6639	1.0000
		40		236 229	16.6795	0.0000
		50		236 229	18.2435	0.0000
Image 2	Template1	30	54 63	54 63	7.5605	0.0000
		40		56 64	9.3651	2.2360

		50		54 63	10.7338	0.0000
	Template2	30	24 89	24 89	4.9898	0.0000
		40		25 89	5.6302	1.0000
		50		24 89	5.6118	0.0000
Image 3	Templat1	30	195 33	195 33	7.8439	0.0000
		40		195 36	8.1813	3.0000
		50		196 34	8.8867	1.4142
	Template2	30	116 37	116 38	8.3230	1.0000
		40		116 37	8.6887	0.0000
		50		116 38	7.5710	1.0000
Image 4	Template1	30	106 34	106 35	5.3976	1.0000
		40		106 34	7.6041	0.0000
		50		106 34	6.0731	0.0000
Image 5	Templat1	30	62 48	62 48	5.6870	0.0000
		40		62 48	4.2140	0.0000
		50		62 48	6.5430	0.0000

Table2

Images	**template**	**Iterations**	**Position of Template**	**Searched Position**	**Time**	**Error**
Image 1	Template1	400	161 181	161 180	22.6990	0.0000
		450		161 182	20.8246	1.0000

		500		161 180	20.5140	1.0000
	Template2	400	211 273	273 211	11.8869	0.0000
		450		274 211	14.5350	1.0000
		500		276 211	15.7090	3.0000
	Template3	400	230 237	236 229	7.5291	1.4142
		450		236 229	9.8388	1.4142
		500		237 230	9.1963	0.0000
Image 2	Template1	400	54 63	68 66	5.0577	14.3178
		450		69 66	5.6266	15.2970
		500		54 63	6.3074	0.0000
	Template2	400	24 89	24 89	4.4513	0.0000
		450		22 91	6.7875	2.8284
		500		24 89	6.4826	0.0000
Image 3	Template1	400	195 33	195 33	4.4166	0.0000
		450		197 30	8.3817	3.7416
		500		199 33	5.2048	2.0000
	Template2	400	116 37	116 37	7.3286	0.0000
		450		116 36	10.3075	2.0000
		500		116 37	6.6878	0.0000
Image 4	Template1	400	106 34	106 34	3.8438	0.0000
		450		107 35	5.3623	1.4142

		500		106 34	10.5666	1.0000
Image 5	Template1	400	62 48	62 48	4.7194	0.0000
		450		62 48	6.2320	0.0000
		500		62 48	5.6870	0.0000

6.3 Discussion

In the presented work proposed PSO based algorithm has been successfully applied for the object detection in a given image. Table (1) shows the results for image 1, when iterations are fixed at 500 and the population size is varied. It is confirmed that for a selected part of an image1 i.e.template1 shown in fig (6.1), the position of detected object is exactly matched with selected object when population size fixed at 40 and for this case the time consumed to detect the selected object is 28.2371 sec. In the case of test image(1) and the part of image i.e.template2 as shown in fig (6.1) the position of detected object is completely matched with selected template when iterations are fixed at 500 and population size is 30, in this case time consumed is 10.8147 sec. From fig (6.1) the test image1 and selected part of image i.e. temlate3 is taken the complete matching of detected object with selected object when iterations are fixed at 500 and population sizes are 40 and 50, in this case time consumed by algorithm are 16.6795 sec. and 18.2435 sec respectively.

When the test image 2 and selected part of image i.e. template1 as shown in fig (6.2), is taken then detected object completely matched with selected template when iterations are fixed at 500 and population sizes are 30 and 50, in this case time consumed by algorithm are 7.5605 sec. and 10.7338 sec. respectively. When test image2 and selected part of test image i.e. template2 is taken as shown in fig (6.2) the detected object completely matched with selected template when iterations are fixed at 500 and population sizes are 30 and 50, time consumed by the algorithm are 4.9898 sec.and 5.6118 sec. respectively.

When the test image 3 and selected part of image i.e. template1 is taken as shown in fig (6.2), the detected object correctly matched with the selected template when iterations are fixed at 500 and population size is 30, time consumed is 7.8439 sec. When test image3 and selected part of image i.e. template2 is taken as shown in fig(6.3), then the detected object totally matched with the template when iterations are fixed at 500 and population sizes are 40 and time consumed is 8.6887 sec.

When the test image4 and selected part of image i.e. template1 is taken as shown in fig(6.4), the detected object completely matched with the selected template when iterations are fixed at 500 and population sizes are 40 and 50, time consumed b the algorithm are 7.6041sec.and 6.0731 sec. respectively.

When test image5 and selected part of image (template1) is taken, as shown in fig (6.5) the detected object matched with selected template when iterations are fixed at 500 and population sizes are 30, 40 and 50 and time consumed are 5.6870 sec. 4.2140 sec.and 6.5430 sec. respectively.

From table 2 it can be seen that proposed algorithm has been successfully employed for object detection in images by unstable the iterations and keep steady the population size all the test images and templates are same as for the table1.

When test image1 and selected part of image i.e.template1 shown in fig (6.1) is taken, the detected object completely matched with selected template when population size is fixed at 50 and iterations are 400, time consumed by the algorithm is 22.6990 sec. When the test image1 and selected part of test image i.e. template 2 as shown in fig (6.1) the detected object entirely matched with selected template when population size is fixed at 50 and iterations are 400, time consumed by the algorithm is 11.8869 sec. When the test image 1 and selected part of test image (template3) as shown in fig (6.1) is taken, the detected object exactly matched with the selected template when population size is fixed at 50 and iteration are 400, time consumed by the algorithm is 7.5291 sec.

When the test image2 and selected part of test image (template1) as shown in fig (6.2) is taken, the detected object completely matched with selected object when population size is fixed at 50 and iterations are 500, time consumed by the algorithm is 6.3074sec. When test image2 and selected part of test image (template2) as shown in fig (6.2), is taken the detected object exactly matched with selected template when population size is fixed at 50 and iterations are 500, time consumed by the algorithm is 6.4826 sec.

In the case of test image3 and selected part of image (template1) as shown in fig (6.3) is taken, the detected object totally matched with the selected template when population size is fixed at 50 and iterations are 400, time consumed by the algorithm is 4.4166 sec.

When the test image4 and selected part of image (template2) as shown in fig (6.3) is taken, the detected object totally matched with the selected template when population size is fixed at 50 and iterations are 400 and 500 respectively, time consumed by the algorithm are 7.3286 sec. and 6.6878 sec. respectively. When the test image4 and selected part of image (template2) as shown in fig (6.4) is taken, the detected object totally matched with the selected template when population size is fixed at 50 and iterations are 400, time consumed by the algorithm is 3.8438 sec. In the case of test image3 and selected part of image (template1) as shown in fig (6.3) is taken, the detected object exactly matched with the selected template when population size is fixed at 50 and iterations are 400, time consumed by the algorithm is 4.7194 sec.

6.4 Conclusion

The proposed work has successfully employed the PSO based technique to solve the object detection problem in image. The obtained results (from table1 and table2) show that the proposed technique is capable of obtaining the position of object and consumes less time.

When the test images are tested on PSO based algorithm for detecting the position of object and time consumed by the algorithm the obtained result in the case of image1, for template1 (part of an image1) is completely matched with the detected object and time consumed by algorithm increases when iterations are fixed as compare to the case when population size is fixed. The obtained result for the template 2 (part of an image1) is completely matching with searched object and time consumed by algorithm is more when iterations are fixed as compared to the case when population size is fixed. It is obtained in case of template 3 (part of an image1) , is completely matched with the detected object and consumed time by algorithm is more when iterations are fixed as compared to the case when population is fixed.

The obtained result for the case of image 2, template 1(part of an image2) is completely matching with the detected object and time consumed is more when iterations are fixed as compare to case of when population size is fixed. The result obtained in case of template2 (part of an image2) is exactly matched with the detected object and consumed time more when iterations are fixed as compare to the case when population size is fixed.

For the case of image 3, template 1 (part of an image3) is completely matching with the detected object and time consumed is more when iterations are fixed as compare to case of when population is fixed. In the case of template 2 (part of an image3), completely matched with the detected object and consumed time is more when iterations are fixed as compare to when population is fixed.

For the case of image 4, template 1 (part of an image4) is completely matching with the detected object and time consumed is more when iterations are fixed as compare to case of when population size is fixed.

For the case of image 5, template 1 (part of an image5) is completely matching with the detected object and time consumed by the algorithm is less when iterations are fixed as compare to case of when population size is fixed.

It is clear from the obtained results that the proposed PSO based technique, when iterations are fixed consumed more time as compare to when population size is fixed. Hence, algorithm performs better for the considered image.

6.5 Future Scope

Proposed PSO based agorithms in this thesis for object detection can be implemented in wide range of applications, e.g. to navigation, guidance, automatic surveillance, robot vision, and to the mapping sciences. Any automated system for three-dimensional point positioning can use proposed algorithms for object detection. For a computer vision system, the ability to cope with moving and changing objects, changing illumination, and changing viewpoints is essential to perform several tasks. Genetic Algorithm, Anti predatory particle swarm optimization, Differential evolution and Evolutionary programming, PPO can also be implemented in order to provide faster detection of object in image. A new moving object segmentation method for video surveillance based on the particle swarm optimization algorithm can also be implemented.

REFRENCES

1. F. Ackermann, "Digital image correlation: Performance and potential application in photogrammetry", *Photogrammetric Record,* vol.11, pp.429-439, 1984.
2. J. Bala, K. DeJong, J. Huang, H. Vafaie, H. Wechsler, "Visual routine for eye detection using hybrid genetic architectures*", International Conference on Pattern Recognition,* vol. 3, pp. 606-610, 1996.
3. A.W. Gruent, "Adaptive least squares correlation: A powerful image matching Technique", *Journal of Photogrammetry, Remote Sensing and Cartography*, vol.14, pp.175–187, 1985.
4. C.F. Olson, "Maximum-likelihood template matching", *IEEE Conference on Computer Vision and Pattern Recognition,* vol.2, pp. 52-57, 2000
5. T. Peli, "An algorithm for recognition and localization of rotated and scaled objects", *Proceedings of the International Conference on Advances in Pattern Recognition of the IEEE,* vol.69, pp.483–485 1981.
6. M. Kass, A. Witkin, D. Terzopoulos, "Snakes: Active contour models", *International Journal Computer Vision*, vol. 1, pp. 321–331, 1987.
7. Y. Hsu, H.H. Arsenault, G. April: "Rotation-invariant digital pattern recognition using circular harmonic expansion", *Applied Optics,* vol. 21, pp. 4012-4015, 1982.
8. W.Foerstner, "Quality assessment of object location and point transfer using digital image correlation techniques", *International Archives of Photogrammetry and Remote Sensing,* vol. 554, pp.342-352, 1984.
9. K.H. Lin, K.M. Lam, W. Ch. Siu, "Locating the Eye in Human Face Images Using Fractal Dimensions", *IEEE Proceedings - Vision, Image and Signal Processing,* vol. 148, pp. 413-421, 2001.
10. Gonzalez, Woods, Riddins "Digital Image Processing using Mat lab" Pearson Education (Singapore) Pvt., 2004.
11. J.A.Stark, W.J. Fitzgerald "an alternative algorithm for adaptive histogram equalization'', *Graphical Models and Image Processing,* vol. 58, pp.180-185, 1996.
12. J.Canny "A computational approach to edge detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol.8 (6), 679-698, 1986.

13. A. Ng, M. Jordan, Y. Weiss. "On Spectral Clustering Analysis and an Algorithm", *Proceedings of Neural Information Processing Systems*, vol.36 (65), pp.542-556, 2001.
14. R.M. Dufour, E.L. Miller, N.P. Galatsanos, "Template matching based object recognition with unknown geometric parameters", *IEEE Transactions on Image Processing*, vol.11, pp. 1385 – 1396, 2002.
15. K. Peng, L. Chen, S. Ruan, Georgy Kukharev, "A Robust and Efficient Algorithm for Eye Detection on Gray Intensity Face", *Lecture Notes in Computer Science – Pattern Recognition and Image Analysis*, vol.64(7), pp. 302-308, 2005.
16. M. Motwani, R. Motwani, C. Harris Frederick, "Eye Detection using wavelets and ANN", *IEEE Transactions on Image Processing*, vol.23, pp.3456-3465, 2004.
17. K. Lin, K. M. Lam, W.C. Siu, "Locating the Eye in Human Face Images Using Fractal Dimensions", *IEE Proceedings - Vision, Image and Signal Processing,* vol. 148, pp. 413-421, 2001.
18. H.V. Karvir, J.A Skipper, "Power spectrum weighted edge analysis for straight edge detection in images'', *SPIE Defense and Security Symposium: Visual Information Processing* vol.556, pp.6575-6582, 2007.
19. R. C. Eberhart, Y. Shi, "Particle Swarm Optimization: Developments, Applications, and Resources", *Proceedings Congress on Evolutionary Computation IEEE service center*, vol.91 (23), pp.3452-3467, 2001.
20. F. Zhao, Q. Huang, W. Gao, "Methods of image matching by normalized cross-correlation", *Proceedings - Vision, Image and Signal Processing,* vol.34,pp.243-2009 .
21. J.U.K. Cho, "A Real-Time Object Tracking System Using a Particle Filter", *International Conference on Intelligent Robotics and Systems*, vol.43, pp. 2822-2827, 2006.
22. J. Kennedy and R. Eberhart, ''Particle Swarm Optimization'', *Proceedings of IEEE International Conference on Neural Networks*, *Perth, Australia,* vol. 4, pp. 1942-1948, 1995.
23. L. A.Canalis, M. Hernandez-Tejera, E. Sanchez-Nielsen, "Particle Swarms as Video Sequence Inhabitants For Object Tracking in Computer Vision", *Proceedings - Vision, Image and Signal Processing*, vol. 2, pp. 604-609, 2006.
24. Y. Shi, R. Eberhart, "A modified particle swarm optimizer'', *Proceedings of the IEEE International Conference on Evolutionary Computation*, vol.65, pp 69–73, 1998

25. F. van den Bergh, “An Analysis of Particle Swarm Optimizers’’, *PhD thesis, University of Pretoria*, 2001.
26. S. Prakash, U. Jayaraman, P. Gupta, “A Skin-Color and Template Based Technique for Automatic Ear Detection’’, *ICAPR Proceedings of the International Conference on Advances in Pattern Recognition*, vol. 978, pp.3520-3528, 2009.
27. J. W. Heng-jun, Y. C., Zhi-ming Cui1 “Video Object Tracking Method Based on Normalized Cross-correlation Matching**”,** *International Symposium on Distributed Computing and Applications to Business*, *Engineering and Science,* vol.12, pp.112-128, 2010.
28. J. Kennedy, “Effects of Neighborhood Topology on Particle Swarm Performance’’, *Proceedings of the Congress on Evolutionary Computation*, vol.76, pp. 1931-1938, 1999.
29. Y. Hel-Or, H. Hel-Or, “Real-Time Pattern Matching Using Projection Kernels”, *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, September, 2005.
30. Feng Zhao, Qingming Huang, Wen Gao, “Methods of image matching by normalized cross-correlation”, *International Technology Journal,* vol. 56(12), pp.324-345, 2010.
31. Z.H. Zhou, X. Geng, “Projection functions for eye detection”, *Pattern Recognition*, vol. 37(5), pp. 1049-1056, 2004.
32. K. Peng, L. Chen, Su Ruan, G. Kukharev, “A Robust and Efficient Algorithm for Eye Detection on Gray Intensity Face”, *Lecture Notes in Computer Science – Pattern Recognition and Image Analysis*, vol.27, pp. 302-308, 2005.
33. Liu G, R.M.Haralick, “Optimal matching problem in detection and recognition performance evaluation’’, *The Journal of the Pattern Recognition Society* vol. 35, pp. 2125-2139, 2002.
34. F. Han “A two-stage approach to people and vehicle detection with HOG-based SVM”, *The Performance Metrics for Intelligent Systems Workshop*, vol.56, pp. 133-140, 2006.
35. P. Angeline, “Evolutionary optimization versus particle swarm optimization: Philosophy and performance difference”, *The 7th Annual Conference on Evolutionary Programming*, vol.24, pp.654-669, 1998.
36. R. C. Eberhart, Y. Shi, “Particle swarm optimization: Developments, applications and resource”, *Proceedings of the International Congress on Evolutionary Computation,* vol. 1, pp. 81–86, 2001.

37. J. Kennedy, R. Eberhart, “A Discrete Binary Version of the Particle Swarm Algorithm’’, *Proceedings of the Conference on Systems, Man, and Cybernetics*, vol.2, pp. 4104-4109, 1997.
38. Y. Shi, R. Eberhart, “A modified particle swarm optimizer’’, *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp 69–73, 1998.
39. Z. Yan, Y. Lian, “A New Background Subtraction Method for on-road Traffic”, *Journal of Image and Graphics*, vol.13, No.3, pp.593-599, March, 2008.
40. Wei-feng LIU, “A Target Detection Algorithm Based on Histogram Feature and Particle Swarm Optimization’’, *International Conference on optical networks and communication,* vol.3, pp.54-72, 2009.
41. C.F. Olson, “Maximum-likelihood template matching”, *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 52-57, 2000.
42. S. Kaneko, I. Murase, S. Igarashi, “Robust image registration by increment sign correlation, Pattern Recognition”, *Annual Conference on Evolutionary Programming,* vol.35, pp. 2223–2234, 2002.
43. R. Reynolds, B. Peng, J. Brewster. “Cultural swarms II: Virtual algorithm emergence’’, *Proceedings of IEEE Congress on Evolutionary Computation,* pp. 1972-1979, 2003.
44. S. Chu, J. Roddick, “A Clustering Algorithm Using Tabu Search Approach with Simulated Annealing for Vector Quantization’’, *Chinese Journal of Electronics*, vol. 12, pp. 349-353, 2003.
45. R. Reynolds, B. Peng, J. Brewster. “Cultural swarms II: Virtual algorithm emergence’’, *Proceedings of IEEE Congress on Evolutionary Computation,* pp. 1972-1979, 2003
46. A. Sharma, N. J. Singh “Object detection in image using PSO”, *Signal and Image Processing : An International Journal,* vol.2, pp.205-221, March 2011.
47. Y. Zheng, Y. Meng, “Swarm intelligence based dynamic object tracking”, *IEEE World Congress on Computational Intelligence,* vol.10, pp. 405 – 412, 2008.
48. H. A. Al-Mamun, N. Jahangir, Md. Shahedul Islam, Md. Ashraful Islam “Eye Detection in Facial Image by Genetic Algorithm Driven Deformable Template Matching’’, *International Journal of Computer Science and Network Security,* vol.9 pp.762-785, August 2009.

APPENDIX

```
% Code developed for object detection in image using PSO technique.
        %pso main program
%p_s- population size, it- no of iterations
clc;
clear workspace;
p_s=30;it=500;n_v=2;%p_s: population size   it: no. of iteration  n_v: dimension size
c1=2; c2=2;
wmin=0.4; wmax=0.9;
%////////////////////////////////////////////////////////////
tic;
K=imread('C:\eye\d.jpg');
I=rgb2gray(K);
imshow(I);
%templateR=imread('C:\Users\sudhakar\Desktop\eye\sa12.jpg');
%templateR=rgb2gray(templateR);
[mm nn oo] = size(I);%mm=(rows), nn=(column)
[pp qq rr] = size(templateR);%pp=26(rows), qq=56(column)
 pmin=1;
ps=rand(n_v,p_s);  %% random generation of population
vl=rand(n_v,p_s);
 for k=1:p_s
       j=1;
       ps(1,k)=pmin+(pmax-pmin)*ps(1,k);
       ps(2,k)=pmin2+(pmax2-pmin2)*ps(2,k);
    end
   for k=1:p_s
       j=1;
       vl(j,k)=vmin+(vmax-vmin)*vl(j,k);
    end
 vmax2=pmax2/100;
vmin2=-vmax;
   for k=1:p_s
       j=2;
       vl(j,k)=vmin2+(vmax2-vmin2)*vl(j,k);
    end
 %calculate fit1ness
for j=1:2
   for k=1:p_s
       ps(j,k)=round(ps(j,k));
    end
end
fit1 = fit1ness(p_s,ps,pp,qq,I,templateR);
%fit1
%find global and local best
l_b=ps;
l_f=fit1;
[a,b]=max(fit1);
g_f(1)=a;
% iterations begins
for i=2:it
    w=wmax-(wmax-wmin)*i/it;
    for j=1:n_v
        for k=1:p_s
            nv(j,k)=w*vl(j,k)+c1*rand(1)*(l_b(j,k)-ps(j,k))+c2*rand(1)*(g_b(j,i-1)-ps(j,k));
            %nv(j,k)=vl(j,k)+c2*rand(1)*(g_b-ps(j,k));
            if(j==1)
                if(nv(j,k)>vmax)
                    nv(j,k)=vmax;
                end
                if(nv(j,k)<vmin)
                    nv(j,k)=vmin;
                end
                ps(j,k)=ps(j,k)+nv(j,k);
                if(ps(j,k)>pmax)
```

```
                ps(j,k)=pmax;
            end
            if(ps(j,k)<pmin)
                ps(j,k)=pmin;
            end
        end
        if(j==2)
            if(nv(j,k)>vmax2)
                nv(j,k)=vmax2;
            end
            if(nv(j,k)<vmin2)
                nv(j,k)=vmin2;
            end
            ps(j,k)=ps(j,k)+nv(j,k);
            if(ps(j,k)>pmax2)
                ps(j,k)=pmax2;
            end
            if(ps(j,k)<pmin2)
                ps(j,k)=pmin2;
            end
        end
    end
end
%change
%calculate fit1ness
for j=1:2
    for k=1:p_s
        ps(j,k)=round(ps(j,k));
    end
end
fit1 = fit1ness(p_s,ps,pp,qq,I,templateR);
%find global and local best
for j=1:p_s
    if(fit1(j)>l_f(j));
        l_f(j)=fit1(j);
        l_b(:,j)=ps(:,j);
    end
end
     g_f(i)=a;
g_b(:,i)=l_b(:,b);
%g_b=l_b(b);
vl=nv;
end
toc;
%file write for the global best and global fitness
    fid = fopen('global_best.xls', 'at');
    % fprintf (fid, '%d\n',i);
    fprintf(fid, '%6.4f  %6.4f\n', g_b);
    fclose(fid);
    fid = fopen('global_fit.xls', 'at');
    fprintf(fid, '%6.4f\n', g_f);
    fclose(fid)
    plot(1:1:400,g_f,'r')
    %file end
show=I(g_b(1,it):g_b(1,it)+pp-1,g_b(2,it):g_b(2,it)+qq-1,1);
figure; imshow(show);
clc;
```