. OBJECT DETECTION IN IMAGE USING PARTICLE SWARM OPTIMIZATION

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

Master of Engineering
in
Electronics Instrumentation & Control Engineering

By:
Ankit Sharma
Regn. No. 800851002

Under the supervision of:
Mr. Nirbhow Jap Singh
Assistant Professor, EIED

ELECTRICAL AND INSTRUMENTATION ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

July 2010

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "Object Detection In Image Using Particle Swarm Optimization", in partial fulfillment of the requirements for the award of degree of Master of Engineering in Electronics, Instrumentation & Control Engineering submitted in Electrical & 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** and refers other researcher's works which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

Ankit
28/06/10

(Ankit Sharma)
Regn. No.-800851002

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

28/6/10

(Mr. Nirbhow Jap Singh)
Assistant Professor
Electrical & Instrumentation Engineering Department
Thapar University
Patiala

Countersigned by

S. Ghosh
12/7/10

(Dr. Smarajit Ghosh)
Head, EIED
Thapar University,
Patiala.

16.7.10

(Dr. R.K.Sharma)
Dean (Academic Affairs)
Thapar University
Patiala

II

ACKNOWLEDGEMENT

I would like to express my gratitude to my guides **Mr. Nirbhow Jap Singh**, Assistant Professor, Electrical and Instrumentation Engineering Department, Thapar University, Patiala for their patient guidance and support throughout this thesis work. I am truly very fortunate to have the opportunity to work with him. I found this guidance to be extremely valuable.

I shall be failing in my duty, if I don't place on record my thanks and regards to **Mr. Nitin Narang**, Electrical & Instrumentation Engineering Department, Thapar University, Patiala for his unending support right from the elementary stage.

I shall be failing in my duties if I do not express my deep sense of gratitude towards **Dr. Smarajit Ghosh,** Professor and Head of Electrical and Instrumentation Department who has been a constant source of inspiration for me throughout this work.

I am also thankful to all the staff members of the Department for their full cooperation and help.

I am greatly indebted to all my friends, who have graciously applied themselves to the task of helping me with ample morale support and valuable suggestions. Finally, I would like to extend my gratitude to all those persons who directly or indirectly helped me in the process and contributed towards this work.

28/06/10

Ankit Sharma
(Regn. No. 800851002)

III

ABSTRACT

Image matching is a key component in almost any image analysis process. Image matching is crucial to a wide range of applications, such as in navigation, guidance, automatic surveillance, robot vision, and in mapping sciences. Any automated system for three-dimensional point positioning must include a potent procedure for image matching. Most biological vision systems have the talent to cope with changing world. Computer vision systems have developed in the same way. 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. 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. 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. The shortcomings of this class of image matching methods have caused a slow-down in the development of operational automated correlation systems. In the proposed work *particle swarm optimization* & its variants based algorithm is used for detection of object in image. Implementation of this algorithm reduces the time required for object detection than conventional template matching algorithm. Algorithm can detect object in less number of iteration & hence less time & energy than the complexity of conventional template matching. This feature makes the method capable for real time implementation. In this thesis a study of particle Swarm optimization algorithm is done & then formulation of the algorithm for object detection using PSO & its variants is implemented for validating its effectiveness.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Overview

As humans, it is easy (even for a child) to detect the position of the letters, objects, numbers, etc. However, making a computer 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. Automation is desperately needed to handle this information explosion. 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. 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 brain filters out useful information is not fully known and this skill has not been merged into computer vision science. The aim of this thesis is to implement a system in MATLAB that is able to faster detection of object in an image. The system should use both fast and advanced algorithms aiming to achieve the exact position of the object in image. The goal is to develop a system with the potential to be implemented in a real time environment. Therefore the system needs to be very fast.

Artificial intelligence is an important topic of the current computer science research. In order to be able to act intelligently a machine should be aware of its environment. The visual information is essential for humans. Therefore, among many different possible sensors, the cameras seem very important. Automatically analyzing images and image sequences is the area of research usually called "computer vision". This thesis is related to the broad subject of automatic detection of object in image.

Image matching is required in almost all of the image analysis processes. Image matching has large no. of applications which includes in navigation, guidance, automatic surveillance, robot vision, and in mapping sciences. Cross-correlation 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.

This thesis investigates the application of an efficient 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. In the proposed work particle swarm optimization based algorithm is used for detection of object in image. Implementation of this algorithm resulted in reduction of time required for object detection than conventional template matching algorithm. Hence, the method is capable of detecting the object in real time applications. In this thesis a study of particle Swarm optimization algorithm is done & formulation of the algorithm for object detection using PSO & its variants is done.

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 and pattern matching 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. The chapter 4, deals with the explanation of PSO technique & PPO technique with its algorithm. In chapter 5, the formulation of algorithm to object detection using different search methodologies is explained in detail with flow chart. In chapter 6, the result of object detection algorithm based on conventional cross-correlation algorithm and proposed algorithm are compared for different images.

CHAPTER 2
LITERATURE REVIEW

Many techniques are proposed for detection of object in image 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.

W. Forstner [9]: 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.

F. Ackermann [13]: 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.

A W Gruent [14]: 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. It allows for simultaneous radiometric corrections and local geometrical image shaping, whereby the system parameters are automatically assessed, corrected, and thus optimized during the least squares iterations. 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.

J. P. Lewis [15]: Proposed the method for fast normalized cross-correlation. Although it is well known that cross correlation can be efficiently implemented in the transform domain, the normalized form of cross correlation preferred for feature matching applications does not have a simple frequency domain expression. Normalized cross correlation has been computed in the spatial domain for this reason.

J. Bala, K. DeJong, J. Huang, H. Vafaie, H. Wechsler [16]: They 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.

Martin Berger [17]: He 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.

C.F. Olson [18]: 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 subpixel localization and uncertainty estimation, these techniques allow optimal feature selection based on

minimizing the localization uncertainty. He examined the use of these techniques for object recognition, stereo matching, feature selection, and tracking.

Kwan-Ho Lin, Kin-Man Lam and Wan-Chi [22]: 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.

R.M. Dufour, E.L. Miller, N.P. Galatsanos [23]: They 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, Qingming Huang, Wen Gao [28]: 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.

Mukesh Motwani, Rakhi Motwani, and Frederick C. Harris [29]: They 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.

Z.-H. Zhou and X. Geng [30]: 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, Hagit Hel-Or [31]: 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.

Kun Peng, Liming Chen, Su Ruan, Georgy Kukharev abstract [32]: They 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.

CHAPTER 3

DIGITAL IMAGE PROCESSING

3.1 Introduction

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, MIT, Bell Labs, University of Maryland, and a few other places, with application to satellite imagery, wire photo standards conversion, medical imaging, videophone, 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. As general-purpose computers became faster, these computers started to take over the role of dedicated hardware for all but the most specialized and compute-intensive operations.

With the fast computers and signal processors 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.

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. 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.

Digital image processing is a subset of the electronic domain wherein 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.

$$f(x,y) = \begin{bmatrix} f(0,0) & \cdots & f(0,N-1) \\ \vdots & \ddots & \vdots \\ f(M-1,0) & \cdots & f(M-1,N-1) \end{bmatrix} \tag{3.1}$$

3.2 Image Basics

The basic image characteristics include:

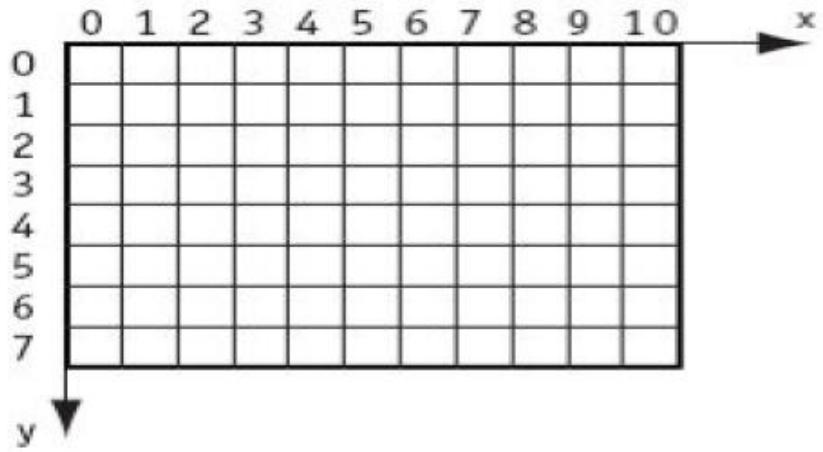
3.2.1 Image metrics

3.2.2 Image modes

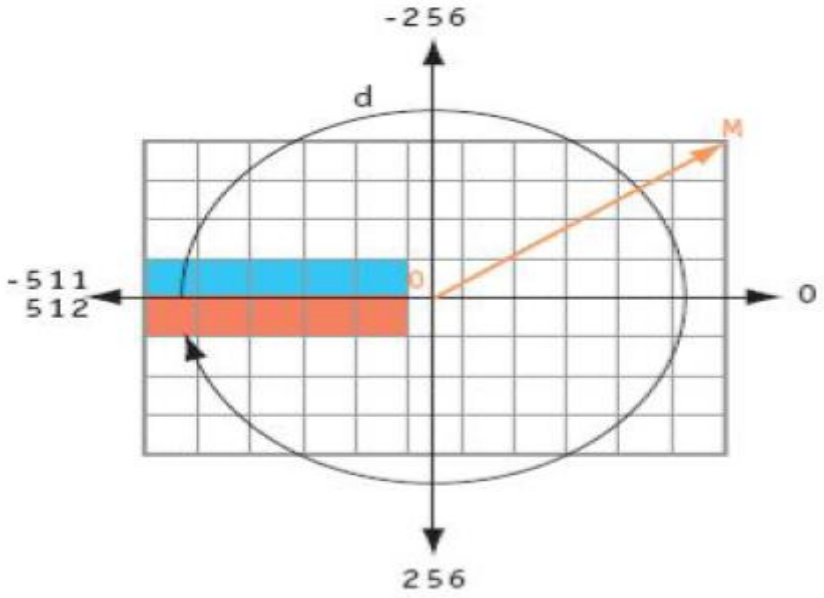
3.2.3 Image histograms

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.



(a)



(b)

Fig 3.1: Visualization of x, y, d, m variables and X, Y, M pseudo constants

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 is paper white. 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.

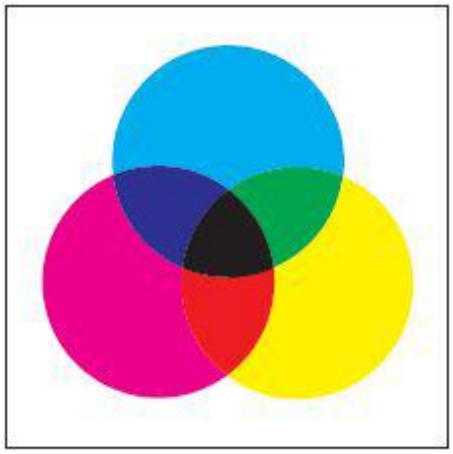


FIG 3.3: A print of three colored circles (cyan, magenta and yellow) on paper

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 Template Matching

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:

$$\lambda(x,y) = \frac{\sum_s \sum_t \delta_{I(x+s,y+t)} \delta_{T(s,t)}}{\sum_s \sum_t \delta^2_{I(x+s,y+t)} \sum_s \sum_t \delta^2_{T(s,t)}} \quad (3.2)$$

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}

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

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

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.

CHAPTER 4

SEARCH METHODOLOGIES

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 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.

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.

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 Grenander presented simulations of bird flocking. It became obvious during the development of the particle swarm concept that the neighbours of the population of agents is 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 Millonas , 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 The Particle Swarm Optimization Concept

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 p_{best} and g_{best}. 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 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.

4.4.1 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 farther 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.

4.4.2 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.

4.4.3 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.

4.5 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.

4.5.1 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.

4.5.2 Number of Individuals

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

4.5.3 Inertia weight

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

$$v_j(i) = w * v_j(i - 1) + C_1(p_{best} - x_j(i)) + C_2(g_{best} - x_j(i)) \quad (4.1)$$

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.

According to the discussion in above sections, the following procedure can be used for implementing the PSO algorithm.

1. Initialize the swarm by assigning a random position in the problem search space to each particle.
2. Evaluate the fitness function for each particle and find out the p_{best}.
4. For each individual particle, compare the particle's fitness value with its p_{best}. If the current value is better than the p_{best} value, then set this value as the current particle's position, x_j, as p_j.
4. Identify the particle that has the best fitness value. The value of its fitness function is identified as g_{best} and its position as p_g.
5. Update the velocities and positions of all the particles using equations,

$$v_j(i) = v_j(i-l) + C_1(p_{best(j)} - x_j(i)) + C_2(g_{best} - x_j(i)) \quad (4.2)$$

$$x_j(i) = x_j(i - 1) + v_j(i)$$

6. Repeat steps 2–5 until a stopping criterion is met (e.g., maximum number of iterations or a sufficiently good fitness value). The flow chart is given as under:

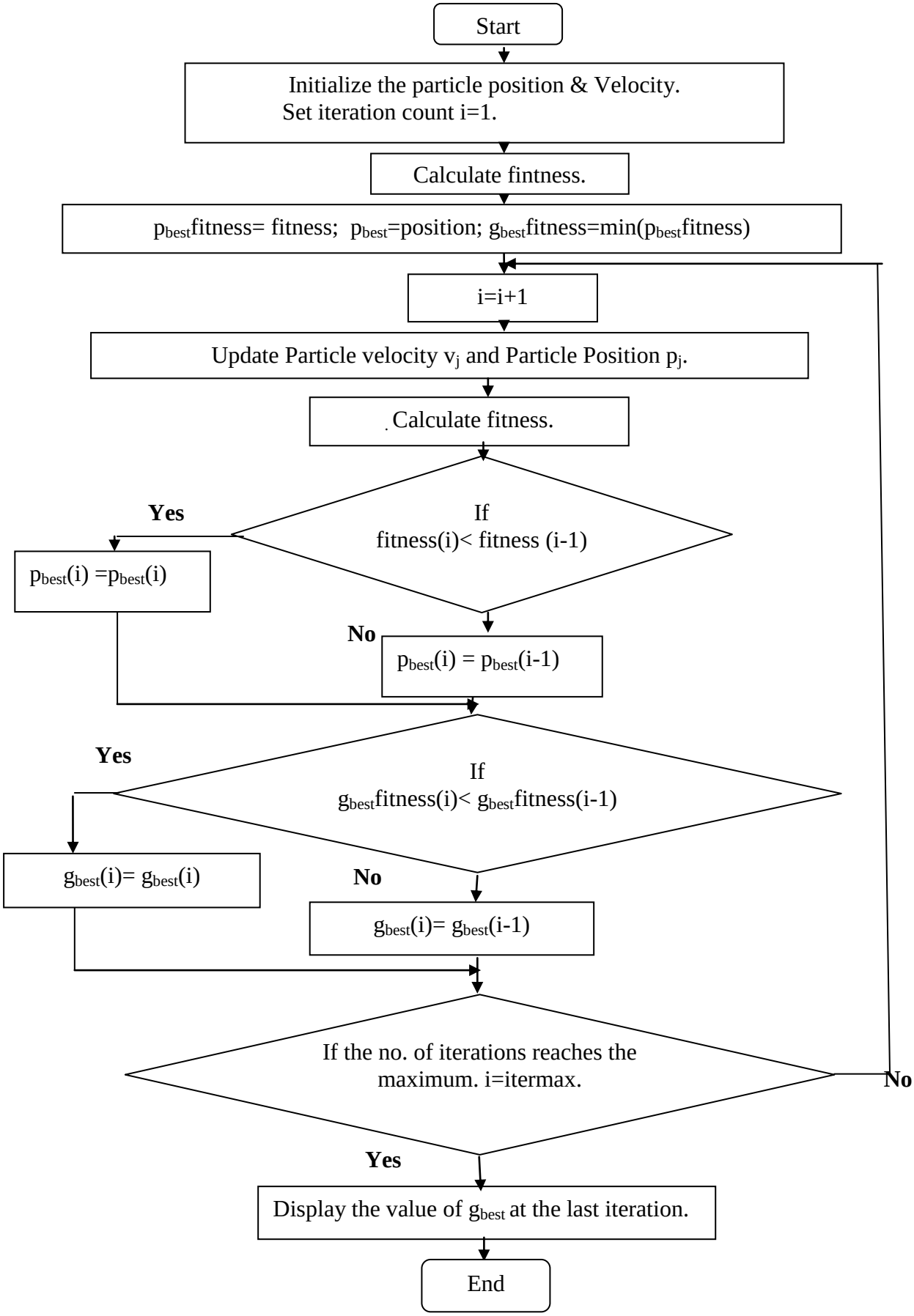


Fig 4.1: The flow chart of PSO.

4.5 The Predator-Prey Optimizer

Motivation for developing the predator-prey model was mainly to introduce a mechanism for creating diversity in the swarm at any moment during the run of the algorithm, not depending on the level of convergence already achieved. This allows the "escape" of particles even when convergence of the swarm around a local suboptimum had already occurred. A second, and less practical, motive was to maintain the biological metaphor. The predator-prey model is inspired in the hunt in nature of animals grouped in flocks by one or more predators. When chased, animals have more difficulty to stay around their most preferable places (better pastures, water sources…) and have to search for other locations, free of predators and perhaps even better. This is the effect which is modeled in this algorithm, where the metaphorical better pastures are the function's local sub optima. The predator's objective is to pursue the best individual in the swarm, i.e. the individual that has found the best point in the search space corresponding to the function being optimized. The predator update equations are:

$$v_p(i) = C_4(x_g(i\text{-}1) - x_p(i\text{-}1)) \qquad (4.3)$$
$$x_p(i) = x_p(i\text{-}1) + v_p(i)$$

C_4 is another random number distributed between 0 and an upper limit and x_g is the present position of the best particle in the swarm. The upper limit on C_4 allows us to control how fast the predator "catches" the best individual.

The influence of the predator on any individual in the swarm is controlled by a "fear" probability P_f, which is the probability of a particle changing its velocity in one of the available dimensions due to the presence of the predator.

The dimension where the change will occur is randomly chosen. For some particle *j*, if there is no change in the velocity in a dimension k the update rules in that dimension still are:

$$v_{jk}(i) = wv_{jk}(i\text{-}l) + C_1(p_{best(jk)} - x_{jk}(i\text{-}1)) + C_2(g_{best} - x_{jk}(i\text{-}1)) \qquad (4.4)$$
$$x_{jk}(i) = x_{jk}(i - 1) + v_{jk}(i)$$

But if the predator influences the velocity in dimension k, the rule becomes:

$$v_{jk}(i) = wv_{jk}(i\text{-}1) + C_1(p_{best(jk)} - x_{jk}(i\text{-}1)) + C_2(g_{best} - x_{jk}(i\text{-}1)) + C_3 D(d) \qquad (4.5)$$

$$x_{jk}(i) = x_{jk}(i - 1) + v_{jk}(i)$$

The fourth term in the first equation in (4.5) quantifies the repulsive influence of the predator by modifying the velocity adding a value that is a function of the difference between the position of the predator and the particle. *d* is the averaged sum of the absolute values of the distances in each dimension. *D(x)* is an exponential decreasing distance function defined as:

$$D(x) = ae^{-bx} \qquad (4.6)$$

D(x) makes the influence of the predator grow exponentially with proximity. The objective of its use is to introduce more perturbation in the swarm when the particles are nearer the predator, which usually happens when convergence occurs. When the distance is bigger (e.g. during the initial exploration phase of the swarm, when *w* is still big), the predator's influence is smaller and usual swarm dynamics take control. The *a* and *b* parameter defines the form of the *D* function: *a* represents the maximum amplitude of the predator effect over a prey and *b* permits to control the distance at which the effect is still significant.

The predator effect was designed to take advantage of the use of *w* as an inertia parameter in the swarm update equations. The idea is to lower the values of *w*, thus forcing a faster convergence, while relying on the predator to maintain population diversity.

CHAPTER 5
OBJECT DETECTION IN IMAGE USING SEARCH METHODOLOGIES

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 & Cross-correlation Coefficient

Template matching is a popular method for pattern recognition. Its 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:

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

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}

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

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

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.

5.2.2 Complexity of Deformable Template Matching Algorithm

The worst case complexity of deformable template matching can be defined by the following lemma.

Lemma 1: If the deformable template has N degrees of freedom with variation ranges $\{R_1, R_2, R_3, \ldots, R_N\}$, an image of size m×n posses a worst case complexity of $O(R_1 R_2 R_3 ... R_N mn)$ for template matching.

Proof: It is apparent from Algorithm 1 that calculation of γ over the entire image, requires time of $O_{(mn)}$. In that algorithm the variation range of x and y were 1 and 1 respectively since x and y were not changed while calculating $\gamma(x,y)$ at a particular point (x, y). So worst case complexity for fixed size template matching is $O_{(1\times1\times mn)}$. But if each degree of freedom varies in a range 0 to R_i then γ (x, y) at point (x, y) will be calculated $O(R_1 R_2 R_3 ... R_N)$ times. So the worst case complexity for template matching is $O(R_1 R_2 R_3 ... R_N mn)$.

This time is huge for real-time implementation. That's why a new template matching algorithm is required which should be based on particle swarm optimization.

5.3 Search methodologies for Object Detection

The Particle swarm optimization (PSO) and Predator prey approach (PPO) has been briefed in chapter 4. 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.

$$upper\ limit \geq p_j \geq lower\ limit$$

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

$$\lambda(x,y) = \frac{\sum_s \sum_t \delta_{I(x+s,y+t)} \delta_{T(s,t)}}{\sum_s \sum_t \delta^2_{I(x+s,y+t)} \sum_s \sum_t \delta^2_{T(s,t)}} \quad (5.2)$$

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}

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

$$T = \frac{1}{pq}\sum_{s}\sum_{t} T(s,t)$$

And inequality constraints as

$$p_j^{min} \leq p_j \leq p_j^{max}$$

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

$$V_j(i) = v_j(i-1) + C_1(p_{best(j)} - p_j(i)) + C_2(g_{best} - p_j(i)) \quad (5.3)$$

6. Modify the individual's position p_j as

$$P_j(i) = p_j(i-1) + v_j(i) \quad (5.4)$$

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 is obtained from g_{best}.

The following steps are used by the PPO 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.

$$upper\ limit \geq p_j \geq lower\ limit$$

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. Initialize the predator position and velocity.

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

$$\lambda(x,y) = \frac{\sum_s \sum_t \delta_{I(x+s,y+t)} \delta_{T(s,t)}}{\sum_s \sum_t \delta^2_{I(x+s,y+t)} \sum_s \sum_t \delta^2_{T(s,t)}} \quad (5.5)$$

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}

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

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

And inequality constraints as

$$p_j^{min} \leq p_j \leq p_j^{max}$$

5. Compare each individual's fitness value with its p_{best}. The best fitness value among p_{best} is denoted as g_{best}.

6. Modify the predator velocity and positions, predator update equations are:

$$v_p(i) = C_5(x_g(i\text{-}1) - x_p(i\text{-}1)) \quad (5.6)$$
$$x_p(i) = x_p(i\text{-}1) + v_p(i)$$

C5 is another random number distributed between 0 and an upper limit and *xg* is the present position of the best particle in the swarm. The influence of the predator on any individual in the swarm is controlled by a "fear" probability P_f. The dimension where the change will occur is randomly chosen. For some particle *j*, if there is no change in the velocity in a dimension *k* the update rules in that dimension still are:

$$v_{jk}(i) = w * v_{jk}(i\text{-}l) + C_1(p_{best(jk)} - x_{jk}(i\text{-}1)) + C_2(g_{best} - x_{jk}(i\text{-}1)) \quad (5.7)$$
$$x_{jk}(i) = x_{jk}(i-1) + v_{jk}(i)$$

But if the predator influences the velocity in dimension k, the rule becomes:

$$v_{jk}(i) = w * v_{jk}(i\text{-}l) + C_1(p_{best(jk)} - x_{jk}(i\text{-}1)) + C_2(g_{best} - x_{jk}(i\text{-}1)) + C_3 D(d) \quad (5.8)$$
$$x_{jk}(i) = x_{jk}(i-1) + v_{jk}(i)$$

The fourth term in the first equation in (5) quantifies the repulsive influence of the predator by modifying the velocity adding a value that is a function of the difference between the position of the predator and the particle. *d* is the averaged sum of the absolute values of the distances in each dimension. *D(x)* is an exponential decreasing distance function defined as:

$$D(x) = ae^{-bx} \quad (5.9)$$

D(x) makes the influence of the predator grow exponentially with proximity. The *a* and *b* parameter defines the form of the *D* function: *a* represents the maximum amplitude of the predator effect over a prey and *b* permits to control the distance at which the effect is still significant.

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 4.

10. The individual point that provides the exact position of template on image is obtained.

The following steps are used by PSO algorithm with the concept of regenerating the population 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.

$$upper\ limit \geq j \geq lower\ limit$$

Here p_i 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

$$\lambda(x,y) = \frac{\sum_s \sum_t \delta_{I(x+s,y+t)} \delta_{T(s,t)}}{\sum_s \sum_t \delta^2_{I(x+s,y+t)} \sum_s \sum_t \delta^2_{T(s,t)}} \quad (5.10)$$

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}

$$I(x,y) = \frac{1}{pq}\sum_{s}\sum_{t} I(x+s, y+t)$$

$$T = \frac{1}{pq}\sum_{s}\sum_{t} T(s,t)$$

And inequality constraints as

$$p_j^{min} \leq p_j \leq p_j^{max}$$

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_d of each individual p_j as

$$v_j(i) = v_j(i\text{-}l) + C_1(p_{best(j)} - p_j(i)) + C_2(g_{best} - p_j(i)) \quad (5.11)$$

Modify the individual's position p_j as

$$p_j(i) = p_j(i - 1) + v_j(i) \quad (5.12)$$

where i is the j^{th} unit and i is for iteration.

6. 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}.

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

8. If the value of fitness remains constant for continuous 10 times then initialize the population again.

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 is obtained.

CHAPTER 6
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 & Predator-prey 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 & Templates

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

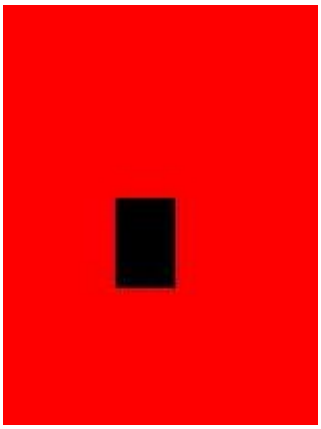


Image (a)

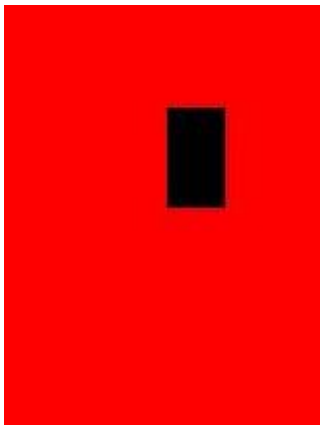


Image (b)

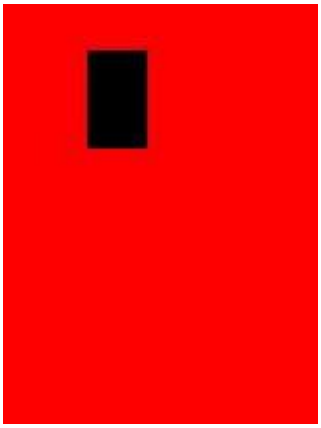


Image (c)

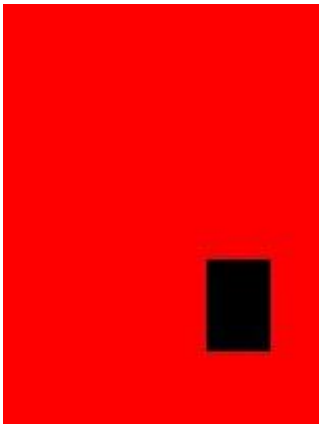


Image (d)

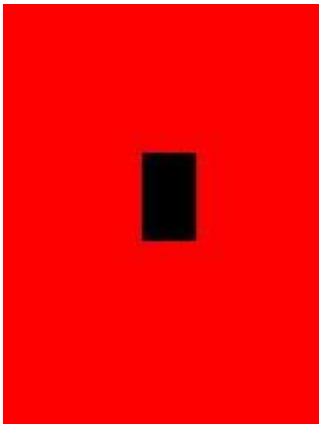


Image (e)

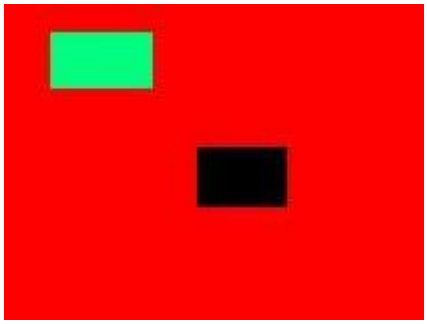


Image (f)

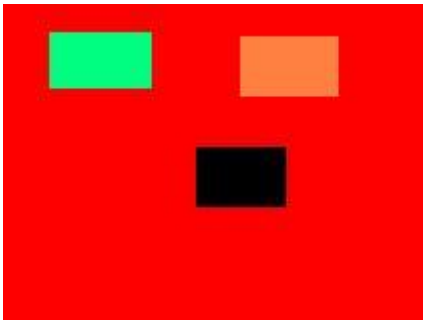


Image (g)

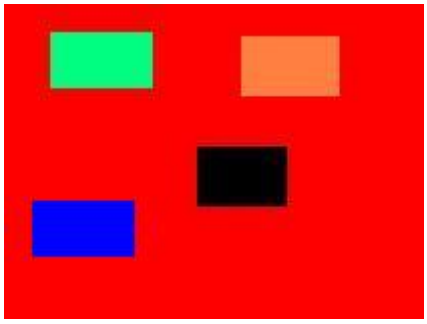


Image (h)

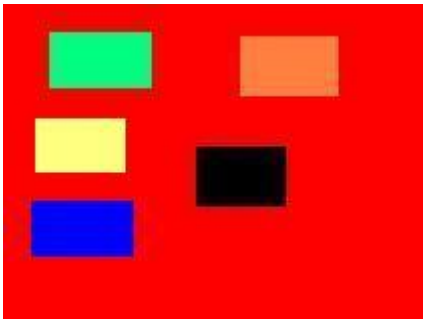


Image (i)

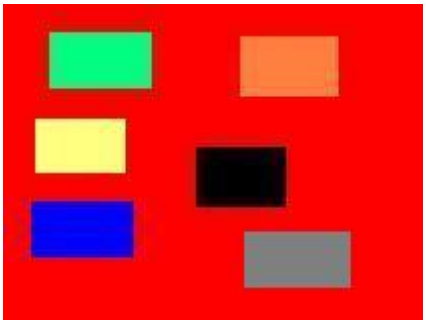


Image (j)

Fig. 6.1 Different test images

6.3 Results from different search algorithms

Above test images are tested on PSO based algorithm and then on PPO based algorithm for detecting the position of black template & both the algorithms are capable of detecting the position of object in image with very less time as compared to conventional template matching algorithm. “Image a”, “image b”, “image c”, “image d” & “image e” are of size 209*157 & “image f”, “image g”, “image h”, “image i” & “image j” are of size 157*209. In conventional template matching algorithm the time taken for detection of object is also calculated & it is revealed that in the proposed algorithms the time consumed in the process of object detection is reduced by around 10 times.

6.3.1 PSO based technique

The proposed PSO based technique was implemented in MATLAB and run under Pentium-4 (2.79 GHz) machine with 512 MB of RAM. Each test image is tested ten times by PSO based object detection program.

Image	PSO Based Algorithm						Old Algorithm
	Minimum Iteration Taken	Minimum Time Taken (In sec.)	Maximum Iteration Taken	Maximum Time Taken (In sec.)	Average Iteration Taken	Average Time Taken (In sec.)	Time Taken (In sec.)
(a)	7	0.18	131	2.03	38	0.60	3.85
(b)	11	0.17	65	1.00	32	0.50	5.56
(c)	11	0.17	90	1.39	33	0.5	2.87
(d)	18	0.27	170	2.63	51	0.81	7.33
(e)	18	0.27	74	1.14	48	0.70	4.19
(f)	3	0.04	73	1.16	32	0.50	5.70
(g)	4	0.06	130	2.01	38	0.60	5.70
(h)	15	0.23	82	1.27	45	0.70	5.70
(i)	10	0.15	152	2.35	56	0.8	5.70
(j)	6	0.09	82	1.27	44	0.60	5.70

Table 6.1 Time consumed and No. of iterations by PSO based algorithm & Old algorithm

6.3.2 PPO based technique

The proposed PPO based technique was implemented in MATLAB and run under Pentium-4 (2.79 GHz) machine with 512 MB of RAM. Each test image is tested ten times by PPO based object detection program.

Image	PPO Based Algorithm						Old Algorithm
	Minimum Iteration Taken	Minimum Time Taken (In sec.)	Maximum Iteration Taken	Maximum Time Taken (In sec.)	Average Iteration Taken	Average Time Taken (In sec.)	Time Taken (In sec.)
(a)	4	0.06	97	1.54	36	0.51	3.85
(b)	3	0.04	72	1.14	34	0.50	5.56
(c)	6	0.09	97	1.54	42	0.60	2.87
(d)	5	0.07	164	2.54	39	0.60	7.33
(e)	19	0.30	93	1.47	41	0.60	4.19
(f)	9	0.10	56	0.80	23	0.36	5.70
(g)	1	0.01	76	1.2	31	0.49	5.70
(h)	4	0.06	66	1.04	30	0.47	5.70
(i)	6	0.09	74	1.17	31	0.49	5.70
(j)	10	0.15	93	1.47	38	0.60	5.70

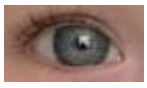
Table 6.2 Time consumed and No. of iterations by PPO based algorithm & Old algorithm

6.4 Real time images & test results with modified algorithm

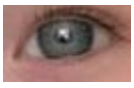
PSO & PPO based algorithms are tested on some real time images, & it is found that these algorithms are not performing well for finding the position of object in image. Concept of regenerating the population is used for the detection of object in such type of images & on testing it is found that this algorithm is capable of detecting the position of the object with 10 times reduction in time as compared to conventional

template matching algorithms. Different test images & templates on which program is tested is shown below.

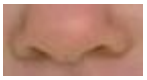
Image (1) Size: 500*500



Template 1.1



Template 1.2



Template 1.2

Fig. 6.2 Test image (1)

Image (2) Size: 339*247

Template 2.1

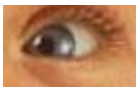
Template 2.2

Template 2.3

Fig. 6.3 Test image (2)

Image (3) Size: 459*402

Template 3.1



Template 3.2

Template 3.3

Fig. 6.4 Test image (3)

Image (4) Size: 486*388

Template 4.1

Template 4.2

Template 4.3

Fig. 6.5 Test image (4)

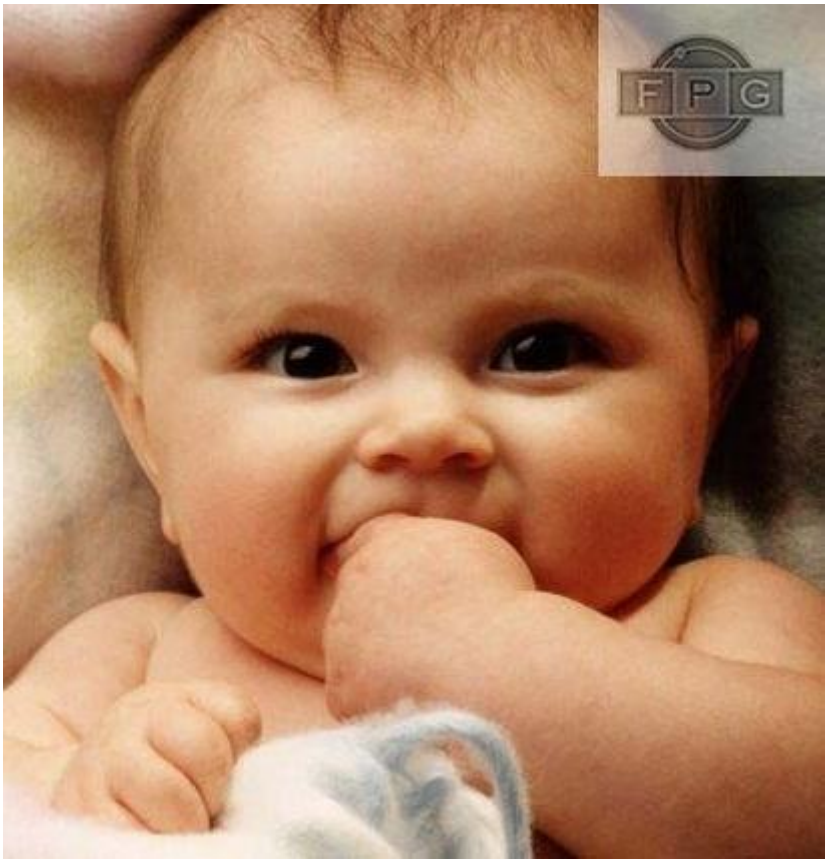
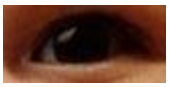


Image (5) Size: 427*410

Template 5.1



Template 5.2

Template 5.3

Fig. 6.6 Test image (5)

The proposed technique was implemented in MATLAB and run under Pentium-4 (2.79 GHz) machine with 512 MB of RAM. Population size was fixed to 20.

Image No.	Template No.	Modified PSO Based Algorithm		Old Algorithm
		No. of iteration	Time Taken (In sec.)	Time Taken (In sec.)
Image (1)	Template 1.1	200	4.50	71.38
	Template 1.2	200	4.60	276.90
	Template 1.3	200	4.50	142.59
Image (2)	Template 2.1	200	3.02	3.09
	Template 2.2	200	2.93	29.88
	Template 2.3	200	2.60	9.03
Image (3)	Template 3.1	200	4.78	61.38
	Template 3.2	200	4.71	210.14
	Template 3.2	200	5.26	164.50
Image (4)	Template 4.1	200	4.09	10.70
	Template 4.2	200	4.10	49.77
	Template 4.3	200	4.46	22.66
Image (5)	Template 5.1	200	5.10	29.17
	Template 5.2	200	5.31	147.85
	Template 5.3	200	5.00	77.90

Table 6.3 Time consumed and No. of iterations by Modified PSO based algorithm & Old algorithm

6.5 Conclusion

In the proposed work, we have successfully employed the PSO based methods to solve the object detection problem. The PSO based algorithm has superior features, including high-quality solution, stable convergence characteristic and good computation efficiency. The results show that the proposed method is capable of obtaining higher quality solution efficiency. It is clear from the results that the proposed PSO based method can avoid the shortcoming of Old template matching algorithm and can provide higher quality solution with better computation efficiency.

When the simple test images are tested on PSO based algorithm & on PPO based algorithm for detecting the position of object then it is found that both of the algorithms are capable of detecting the position of object in image with very less time as compared to conventional template matching algorithm. In conventional template matching algorithm the time taken for detection of abject is nearly 5 sec., while in the proposed algorithms the time consumed in the process of object detection is reduced by 10 times. PSO & PPO based algorithms are tested on some real time images, & it is found that these algorithms are not performing well for finding the position of object in image. Concept of regenerating the population is used for the detection of object in such type of images & on testing it is found that this algorithm is capable of detecting the position of the object with 10 times reduction in time as compared to conventional template matching algorithms.

6.6 Future Scope of Work

Algorithms based on PSO proposed 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. 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. In all of the above application the proposed algorithm can provide faster detection of object. Anti predatory particle swarm optimization, Differential evolution & Evolutionary programming can also be implemented in order to provide faster detection of object in image.

REFERENCES

1. R.O. Duda and P.E.Hart, “Pattern Classification and Scene Analysis”, New York: Wiley, 1973.
2. W.A.Pratt,"Correlation techniques of image registration," IEEE Trans.. AES-IO, pp.353-358, May 1974.
3. R.Y. Wong, E.L. Hall, “Sequential hierarchical scene matching”, IEEE Transactions on Computers 27 (1978) 359–366.
4. R.Y. Wong, E.L. Hall, “Scene matching with invariant moments”, Computer Graphics and Image Processing 8 (1978) 16–24.
5. T. Peli, “An algorithm for recognition and localization of rotated and scaled objects”, Proceedings of the IEEE 69 (1981) 483–485.
6. G. Stockman, S. Kopstein, S. Benett, “Matching images to models for registration and object detection via clustering”, IEEE Transactions on Pattern Analysis and Machine Intelligence 4 (1982) 229–241.
7. A. Rosenfeld and A. Kak: “Digital Picture Processing”, Academic Press, 1982.
8. Y. Hsu, H.H. Arsenault and G. April: “Rotation-invariant digital pattern recognition using circular harmonic expansion," Applied Optics, Vol. 21, pp. 4012-4015, 1982.
9. FOERSTNER, W., “Quality assessment of object location and point transfer usingdigital image correlation techniques. International Archives of Photogrammetry and Remote Sensing” vol. XXV, A3a, Commission III, Rio de Janeiro, 1984.
10. A. Goshtasby, S.H. Gage, and J.F. Bartholic, “A Two-Stage Cross Correlation Approach to Template Matching”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 6, no. 3, pp. 374-378, 1984.
11. ROSENFELD, A. 1984. “Image analysis: problems progress and prospects.” *Pattern Recognition, 1984.*

12. A. Goshtasby, S.H. Gage, and J.F. Bartholic, “A Two-Stage Cross Correlation Approach to Template Matching”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 6, no. 3, pp. 374-378, 1984.
13. ACKERMANN, F. 1984. “Digital image correlation: Performance and potential application in photogrammetry”. Photogrammetric Record 11 (64):429-439.
14. A W Gruent, “Adaptive least squares correlation: A powerful image matching Technique.”, South African Journal of Photogrammetry, Remote Sensing & Cartography, 14(3):175–187, 1985.
15. J. P. Lewis, “Fast normalized cross-correlation”, Canadian Image Processing and Pattern Recognition Society, Quebec City, Canada, May 15-19, 1995.
16. 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.
17. Martin Berger, “The framework of least squares template matching.”**,** 1998.
18. C.F. Olson, “Maximum-likelihood template matching,” IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 52-57, 2000
19. L. Ding, A. Goshtasby, M. Satter, Volume “image registration by template matching, Image and Vision Computing” 19 (2001) 821–832.
20. E. Guest, E. Berry, R.A. Baldock, M. Fidrich, M.A. Smith, “Robust point correspondence applied to two- and three-dimensional image registration”, IEEE Transaction on Pattern Analysis and Machine Intelligence 23 (2001) 165–179.
21. J.V. Hajnal, D.L.G. Hill, D.J. Hawkes, “Medical Image Registration”, CRC Press, Baton Rouge, Florida, 2001, ISBN 0-8493-0064-9.
22. Kwan-Ho Lin, Kin-Man Lam and Wan-Chi Siu, "Locating the Eye in Human Face Images Using Fractal Dimensions," IEE Proceedings - Vision, Image and Signal Processing, vol. 148, no. 6, pp. 413-421, 2001.
23. 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, issue.12, pp. 1385 – 1396, 2002.
24. Rafael C. Gonzalez, Richard E. Woods, “Digital Image Processing,” 2nd ed. Prentice Hall India, 2002.

25. S. Kaneko, I. Murase, S. Igarashi, “Robust image registration by increment sign correlation, Pattern Recognition” 35 (2002) 2223–2234.
26. Barbara Zitova, Jan Flusser, “Image registration methods: a survey”, Image and Vision Computing, 2003.
27. H.S. Alhichri, M. Kamel, “Virtual circles: a new set of features for fast image registration”, Pattern Recognition Letters 24 (2003) 1181–1190.
28. Feng Zhao, Qingming Huang, Wen Gao, “Methods of image matching by normalized cross-correlation”.
29. Mukesh Motwani, Rakhi Motwani, and Frederick C. Harris, Jr. “Eye Detection using wavelets and ANN,” GSPx 2004, 2004.
30. Z.-H. Zhou and X. Geng, “Projection functions for eye detection,” Pattern Recognition, 37(5), pp. 1049-1056, 2004.
31. Yacov Hel-Or, Hagit Hel-Or, “Real-Time Pattern Matching Using Projection Kernels”, IEEE transactions on pattern analysis and machine intelligence, Vol. 27, No. 9, September, 2005.
32. Kun Peng, Liming Chen, Su 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, pp. 302-308, 2005.
33. Muhlenbein, H., & Schlierkamp-Voosen, D. (1993). “Predictive models for the breeder genetic algorithm: I. Continuous parameter optimization.” Evolutionary Computation, 1 (1), 25-49.
34. Kennedy, J. and Eberhart, R. C., “Particle swarm optimisation”, Proc. IEEE International Conference on Neural Networks. Piscataway, NJ, pp. 1942-1948, 1995.
35. Angeline, P. J., “Evolutionary optimization versus particle swarm optimization: philosophy and performance differences”, The Seventh Annual Conf. on Evolutionary Programming, 1998.
36. Shi, Y. and Eberhart, R. C., “Parameter selection in particle swarm optimisation” Evolutionary Programming VII: Proc. EP 98. New York, pp. 591-600, 1998.
37. Shi, Y. and Eberhart, R. C., “Empirical study of particle swarm optimisation”, Proceedings of the 1999 Congress on Evolutionary Computation. Piscataway, NJ, pp. 1945-1950, 1999.

38. Kennedy, J., “Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance”, Proc. Congress on Evolutionary Computation 1999. Piscataway, NJ, pp. 1931-1938, 1999.
39. Kennedy, J., Eberhart, R. C., and Shi, Y., “Swarm intelligence”, Morgan Kaufmann Publishers, San Francisco. 2001.
40. Lovbjerg, M.; Rasmussen, T. K.; and Krink, T., “Hybrid particle swarm optimiser with breeding and subpopulations”, Proceedings of the third Genetic and Evolutionary Computation Conference (GECCO-2001). 2001.
41. A. Silva, A. Neves, E. Costa, “Chasing the Swarm: A Predator Prey Approach to Function Optimization”, in: Proceedings of the MENDEL2002 8th International Conference on Soft Computing, Brno, Czech Republic. 2002.