IMAGE TEMPLATE MATCHING USING VECTORIZED IMAGE FEATURE REPRESENTATION

By

Muhammad Anwaar Manzar

A dissertation submitted to I.I.U. in partial fulfillment of the requirements for the degree of

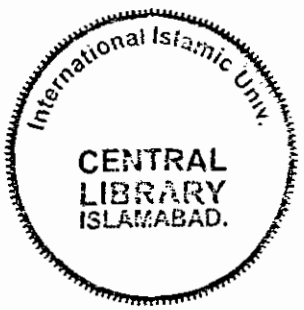
DOCTOR OF PHILOSOPHY

Department of Electronic Engineering

Faculty of Engineering and Technology

INTERNATIONAL ISLAMIC UNIVERSITY

2008



DEDICATED TO

HOLY PROPHET (P. B. U. H.)
THE GREATEST SOCIAL REFORMER

Certificate of Approval

It is certified that the research work contained in this dissertation has been carried out under the supervision of Dr. Ijaz Mansoor Qureshi, at International Islamic University, Islamabad. It is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Signature: _____
Supervisor:
Prof. Dr. Ijaz Mansoor Qureshi,
Dean Department of Electronic Engineering,
Faculty of Engineering and Technology,
International Islamic University.

Signature: _____
Co-supervisor:
Assistant Prof. Dr. Tanveer A. Cheema,
Department of Electronic Engineering,
Faculty of Engineering and Technology,
International Islamic University.

Signature: _____
External Examiner:

Signature: _____
Internal Examiner:

Abstract

Navigation is the science which tells about the position, orientation and velocity of a flying vehicle, relative to some geographical entities. Inertial systems and visual seekers are used for navigation with no external dependency and are always desirable as compared to the methods with external dependency because of their probability of outage and spoofing. Though the inertial based systems in self-sufficient methods are popular, but they are expensive and cannot guide the vehicle accurately on longer routes due to their built-in tendency for accumulating position errors over time. Therefore, the visual navigation is a low cost and efficient solution once supported by machine intelligence and computer vision algorithms. The flying vehicle can store a map of the area for matching the snap shots taken by an onboard camera to navigate efficiently. Thus, different image matching schemes have been presented in this dissertation for reliable aerial visual navigation.

The first approach proposed in this dissertation is through vector matching. The image is first converted into edges from which the prominent ones are vectorized in a hyper-dimension space. Once the image and the template are represented in this hyper space, the matching reduces merely to vector subtraction process. The result of this subtraction gives the matching co-ordinates of the template matched location with respect to the main image. The computation can further be reduced if the main image is represented in hyper-space offline. The whole process is further being supervised to gauge the level of confidence for the early termination of the matching process.

The second approach utilizes the gray values of the pixels in the image for vectorization. The image is first converted into a set of binary images through a gray level slicing process. Then, the connected components in the binary images are

expressed in the form of vectors. The image matching is performed in this vector domain. This algorithm gives a solution for scale and rotation invariance template matching along with a 30 dB of noise robustness.

The last approach forms an adaptive prediction mechanism which further enhances the image slicing scheme suitable to match many incoming templates with a single main image as in the form of a video sequence. The high rate of adaptive convergence impose a tighter bound on the next template location position estimation, which further reduces the computations as the search area is lesser. This adaptive scheme is more efficient in a variety of routes configurations and vehicle velocities as compared to many other techniques.

The proposed algorithms have been evaluated, using percentage of correct matches and computational analysis, under a variety of feature situations on database images as well as the real images. These algorithms have also been compared with some other well known techniques reported in the literature. The proposed schemes outperform the other competitive methods in terms of percentage of correct matches and computational complexity.

List of Publications and Submissions

1. **A. Manzar**, T. A. Cheema and I. M. Qureshi, "New Scale Invariant Template Matching Technique using Hyper Space Image Representation," Pattern Analysis and Application, 2008. (*Published online on 28th March 2008*).

2. **A. Manzar**, T. A. Cheema, A. Jalil and I. M. Qureshi, "A New Image Matching Technique Based on Hyper-Vectorization of Gray Level Sliced Binary Images," IET Image Processing. (*Accepted for publication on 10th August 2008*).

3. **A. Manzar**, T. A. Cheema, A. Jalil and I. M. Qureshi, "Visual Aerial Navigation through Adaptive Prediction and Hyper-Space Image Matching," IEICE transactions on Information and Systems. (*Accepted for publication on 10th November 2008*).

4. A. Jalil, T. A. Cheema, **A. Manzar**, and I. M. Qureshi, "New Rotation-Invariant Texture Analysis Technique using Radon Transform and Hidden Markov Models," IEICE transactions on Information and Systems, vol. E91-D, no. 12, 2008.

5. A. Jalil, **A. Manzar**, A. Zahoor, I. M. Qureshi, "Rotation-Invariant Features for Texture Image Classification," IEEE International Conference on Engineering of Intelligent System, pp. 42-45, 2006.

6. A. Jalil, T. A. Cheema, **A. Manzar**, and I. M. Qureshi, "Rotation and Gray-Scale-Invariant Texture Analysis using Radon and Differential Radon Transforms with Hidden Markov Models," submitted to IET.

7. A. Jalil, T. A. Cheema, **A. Manzar**, and I. M. Qureshi, "Rotation, Translation and Gray Scale Invariant Texture Analysis using Hidden Markov Models and Radon Transform," submitted to IET.

The research publications 1, 2 and 3 are included in this dissertation.

Acknowledgements

I thank to almighty Allah for guiding me at each step of this work through the blessings of health, supporting family, talented teachers and co-operative friends. Peace and prayer for marvelous human and the torch bearer of wisdom Muhammad (Peace be upon him). Who's objective was enlightenment of the whole world.

I offer my special and sincere thanks to my supervisor Dr. Ijaz Mansoor Qureshi whose continuous and motivating guidance, beneficial suggestions and condescended supervision remained with me throughout the research work. Despite his hyper dimensional commitments, he never let me relax.

I owe to express my thanks for fruitful advices and critical reviews of my research work co-supervisor Dr. Tanweer Ahmad Cheema. I found him eager to help me in my research. I am also grateful to Dr. Abdul Jalil and Dr. Aqdas Naveed Malik for their encouragement and inspiration from time to time.

I appreciate my mother and father who remain desirous for my health. Last but not the least I thank my wife Bushra who proved to be embodiment of endurance and sacrifice throughout untiring research years, who loves to see me flourishing and progressing. I also thank my daughters Lamia and Areej who inspired me all along with their little innocent gestures.

(Anwaar Manzar)

Contents

List of Figures

List of Tables

List of Abbreviations

GPS	Global Positioning System
MSE	Mean-Square Error
MAE	Mean-Absolute Error
BMA	Block Matching Algorithm
NMC	Normalized Min Correlation
SAD	Sum of Absolute Differences
BPC	Bounded Partial Correlation
NCC	Normalized Cross-Correlation
ZNCC	Zero Mean Normalized Cross-Correlation Function
NP	Non Parametric
DT	Distance Transform
GHMT	Grayscale Morphological Hit-or-Miss Transform
DCT	Discrete Cosine Transform
PMF	Probability Mass Functions
DP	Dynamic Programming
PISC	Probabilistic Increment Sign Correlation
HMM	Hidden Markov Model
CT	Computed Tomography
CBP	Component Block Projections
RT	Radon Transform
HVT	Hyper Vector Tables
RIMA	Robust Image Matching Algorithm
AWGN	Additive White Gaussian Noise
FDIM	Fourier Descriptor Image Matching
PCA	Principal Component Analysis
PCA-RITF	PCA-based Rotation Invariant Texture Features
3D-FR	Three Dimensional Face Recognition
LMS	Least Mean Square
RLS	Recursive Least Square

List of Symbols

$f(x,y)$	Main Image
$\bar{f}(\bar{x},\bar{y})$	Template Image
(M,N)	Dimensions of the Main Image (rows, columns)
$(\bar{M},\bar{N})$	Dimensions of the Template Image (rows, columns)
m	Template Scale
θ	Template Rotation Angle
$(\Delta x,\Delta y)$	Template Matching Location
$g(s,\theta)$	Radon Transform of the Image
Ψ , Θ	Confidence Factor
$g^{A}(x,y)$	Binary Sliced Image Bands
$S''(k,q)$	Boundaries of the Binary Images
$c(k,q)$	Centroid Co-ordinates
$G(k,q)$	Centroid to Boundary distance vector
$H(k,q)$	Length Normalized Shape Vector
$\psi(k,q)$	Header of the Shape Vector
$\zeta(k,q)$	Magnitude Normalized Shape Vector
$\xi(k,q)$	Consolidated Vector Signature for a Shape
$J(k,q)$	Centroid to Boundary phase vector
$\measuredangle d_n(k,q)$	The phase of the n^{th} angle with centroid of the q^{th} boundary in the k^{th} *binary image*
$D_n(k,q)$	Length Normalized Phase Vector
$\varepsilon(\mathbb{S})$	1-D Correlation Error Function
ε_{min}	The minima of the correlation error function $\varepsilon(\mathbb{S})$ located at $\mathbb{S}=\mathbb{S}_{min}$
γ	Threshold value for vector matching
Z	Solution set after vector matching

$\xi^{k,q}$	Complete Set of the Main Image Hyper-Vectors
$\bar{\xi}^{k,q}$	Reduced subset from the set $\xi^{k,q}$.
$\hat{V}_x, \hat{V}_y$	Predicted Location of the next template matching point

Chapter 1

Introduction

1.1 Problem Statement

Image template matching is an important field of research. Image matching can be defined as the process of finding the locations in an image that are similar to the given pattern template. The pattern template may have a different rotation or (and) scale than the main image. The matching of the template with the main image is a well known problem in the field of computer vision, artificial intelligence, visual navigation and pattern recognition.

The digital representation of the picture image is in the form of pixels which have discrete values in the spatial domain. Let $f(x,y)$ be the main image with dimensions (M,N) such that $1 \le x \le M$ and $1 \le y \le N$. Similarly, let $\bar{f}(\bar{x},\bar{y})$ be the template image with dimensions $(\bar{M},\bar{N})$ such that $1 \le \bar{x} \le \bar{M}$ and $1 \le \bar{y} \le \bar{N}$. We need to find a transformation:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} + m \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} \tag{1.1.1}$$

where Δx and Δy are the translations in x and y directions, respectively, whereas, m and θ are the scaling factor and the rotation angle of the template image, respectively, under which the template coordinates are best mapped onto the main image

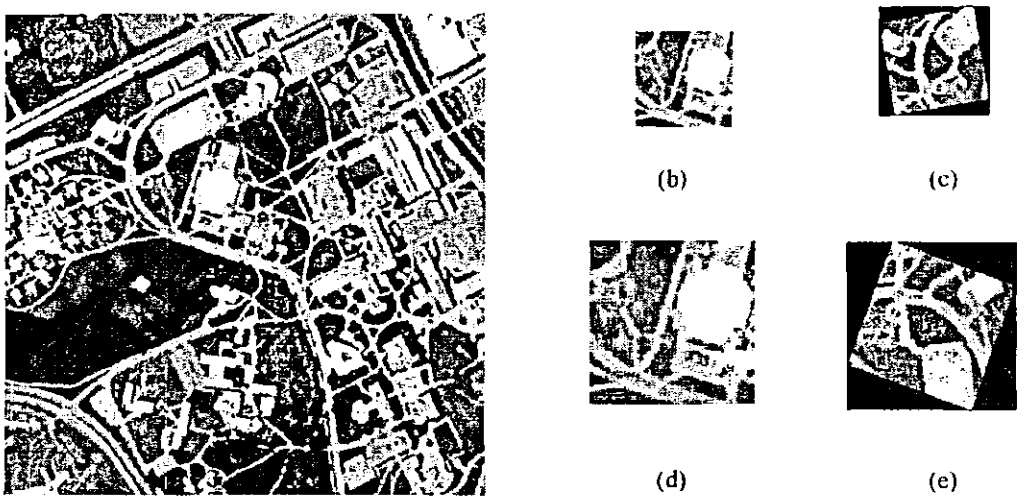


Fig. 1.2 (a) Main Image (b) Original Template (c) Rotated Template (d) Scaled Template (e) Scaled and Rotated Template.

template, rotated template, scaled template and scaled-rotated template, respectively. The research covered in this dissertation is only for rigid body registration only.

1.2 Contribution of the Dissertation

Many template matching techniques are available in the area of pattern matching. They can be broadly categorized into two classes. First of these are based on image pixel values and the second are based on image features. The methods which process the image pixel values are mostly those which are based on a correlation kernel for the search of the best fit location of the template image inside the main image. Advanced methodologies in this domain deal with the optimization of the algorithms and their faster convergence towards a unified solution while utilizing the same correlation kernel.

This dissertation presents a fast template matching scheme through a suitable proposed vectorized representation of the image features. Two different matching schemes are presented. The first scheme deals with the image edge matching, while the second scheme matches the gray level pixels in the image after necessary pre-processing. Matching the edges of both the images provides an effective and robust

way of the determination of best fit location between the two. Matching edges in the spatial domain leads us towards the correlation type of processing in general. The same matching in the frequency domain reduces a lot of computations, but the transform process between the two domains is the undesirable overhead in this area.

The first scheme presented in this dissertation deals with the transformation of the image edges in the form of a vector table. The edge extraction is done in a specific fashion to suit the upcoming stages of the method. Then the vectorization process is facilitated through a modified version of the conventional Radon Transform. Thus the inherent property of the Radon Transform to scan the image with the line slicing of the 2D function helps the realization of the vectors. These 4D hyper-vectors form a cluster of points in the same 4D hyper-space representing the significance of the edges in an image. The main vector matching scheme in fact fits the template edge cluster with that of the main image to ascertain the match or a mismatch situation. In case of a match, the algorithm calculates the best matching location of the template in the main image.

The second scheme converts the gray scale pixel information in the matching images into the form of normalized vector signatures. The conversion starts with the grouping of the image pixels based on their pixel values and their location. The pixel groups are then utilized to form binary images for further processing. These binary images are then used to form the connected component of the pixels for making different types of the vector signatures. These signatures are then normalized for further processing. The normalized vector signatures are subtracted with each other to establish a match or a mismatch condition with a good degree of scale and rotation invariance as well. These schemes are more robust against noise and provide a better and fast image matching solution over a wide range of image scale and rotation than many of the

methods existing in the literature. Furthermore, these schemes are used for the application of automated visual aerial navigation.

Navigation can be defined as the science which tells us about the position, orientation and velocity of an aerial vehicle, with reference to some geographical entities [1]. The methodologies used for navigation are broadly categorized into two classes, firstly those which are dependent on external sources like Global Positioning System (GPS) and guidance beacons etc., secondly the ones which have no external dependencies for their operation like inertial systems and visual seekers. However, the methods with external dependencies are always less desirable because of their probability of outage, spoofing and jamming [2] [3]. In self-sufficient methods, the inertial based systems are popular but they are expensive and cannot guide the vehicle accurately on long routes. This is due to their built-in tendency for accumulating position errors over time. This keeps on increasing the ambiguity in position. However, visual navigation is a low cost and efficient solution. The algorithms are based on machine intelligence and computer vision through which a flying vehicle can store a map of the area for matching the snap shots taken by an onboard camera to navigate efficiently [4] [5]. Thus an efficient image matching scheme is presented for reliable visual navigation. This scheme utilizes less computation to match two images by converting their edges in the form of hyper-dimensional vectors.

1.3 Organization of the Dissertation

Chapter 2 summarizes the previous work in the domain of template matching. Each technique is listed with the discussion covering its implementation. The chapter analyzes the strengths and weaknesses of the most common template matching techniques.

Chapter 3 covers the first scheme for the template matching problem. It is based on the vectorization of image edges. The template matching is done through vector matching after a similar conversion in a hyper dimension space.

Chapter 4 covers the second scheme of the same template matching problem. It also deals with the new idea of vectorization of the image features. The vectorization process includes formation of the normalized vectors suitable for matching two images over a wide range of rotation and scale difference.

Chapter 5 demonstrates a new hybrid scheme which involves the image vectorization for the purpose of matching and a supervisory adaptive prediction mechanism of the next matching location. We utilized this mechanism to demonstrate the application of the said techniques towards efficient target tracking. This kind of scheme is also suitable in a video sequence template matching with a further reduction in the computation level.

Chapter 6 concludes the dissertation. The summary is also narrated for comparison of the proposed scheme with others. Some of the future directions are also highlighted in this chapter.

Chapter 2

Image Template Matching: An Overview

Image matching problem appears in many fields. It is a vital field of research in automated robot movement and machine based navigation. Its application areas include military reconnaissance [6], medicine [7], and astronomy [8] [9], to name a few. The visual perception of an image can be processed for matching in a number of ways. Image edge matching is the matching procedure for the image edges which steer the research towards shape matching of the edged skeleton. Similarly, the matching can be based on gray values of the corresponding pixels which lead towards the correlation process. High level image feature identification and their use for the template matching is also a research area.

2.1 Classical correlation and its variants

The basic of all operations in the domain of image matching is the correlation process for template image matching [10]. Let the main image be $f(x,y)$ of size $M \times N$ and the template image be $w(x,y)$ of size $J \times K$ and we assume that $J \leq M$ and $K \leq N$. The correlation function is expressed as [11]:

$$c(x,y) = \sum_{s=1}^{J} \sum_{t=1}^{K} f(x+s, y+t) w(s,t) \qquad (2.1.1)$$

For $x = 1,2,...,M-J$, $y = 1,2,...,N-K$, and the summation is taken over the image region where w and f overlap. These schemes are computationally expensive as the

handle this method is used. The domain of parallel processing for a faster 2-D convolution was explored on dedicated correlation hardware [15]. A systolic design was also presented for a parallel 2D convolution [16]. The modular nature of the design made it possible to adjust a desirable balance between the computation speed and system I/O. Similarly, a new form of convolution algorithm was presented to be used in parallel processing setups enhancing the matching speed as compared to the conventional sequential schemes [17]. A further advancement in the parallel architecture schemes of the image convolution was presented, which utilizes non-broadcast mechanism to achieve still better performance in matching speeds [18]. The dedicated hardware design solution to the correlation problem does not cater for the increase in the level of computation with increase in the size of the matching images. Furthermore, the hardware realization always imposes a practical limitation between the selection of resources and I/O's.

2.1.2 Block Matching Approaches

Image block matching is a frequently used image matching technique specially employed in video coding. A new scheme in this regard was presented, which uses multi-resolution blocks of image to achieve a lower computation levels [19][20]. The algorithms gave 14-20% of speed improvement over different motion scenarios cases. Similarly, another block matching approach was presented for the estimation of blocks in the consecutive video frames [21]. This method was based on testing only four vital locations for the matching estimation. Another block motion estimation scheme was suggested, which was based on motion fields and pixel sub-sampling [22]. The blocks are fractioned and the sub-sampling space was determined using these fractional blocks. A similar motion vector based algorithm was suggested

identification strategy was also presented [30]. This method divides the template in sub-classes. The class of the template is determined in the first stage of search. This search is then further refined at the second stage for a specific and exact match. Test results were presented for an optical character recognition problem. A sub-template image matching approach was presented to cut down the heavy computations in the classical correlation image matching scheme [31][32]. The new point selection in the sub-template process reduces the overall computations to a greater extent. In the same context, another image matching scheme was presented on parallel virtual machine which was based on wavelets dealing with coarse to fine pixel processing [33]-[35]. This scheme was superimposed by the process of extraction of interesting feature from the image and representing them in the form of a point set. This scheme also utilizes an adaptive selection of the threshold value based on compactness measures of fuzzy sets. An efficient parallel and pipelined implementation of the convolution algorithm on Xilinx's Vertex FPGA is presented [36]. The scheme produces the correlation results in every clock cycle.

2.1.4 Early Truncation Approaches

Low complexity variants of the correlation problem have been suggested with statistical bounds to achieve partial correlations. A new form of normalized minimum correlation (NMC) was introduced which employs low complexity and proves to be robust against noise [37]. Similarly, a new direction in the sum of absolute differences (SAD) algorithm was explored with a hybrid approach with Monte Carlo algorithm to achieve a low computation level [38]. Another approximate solution scheme was suggested for the weighted graph matching problem which involves finding the optimum match between two weighted graphs [39][40]. In order to reduce the

computations to a great deal a fast image matching scheme was presented which skips the comparison location of the template on the basis of a lower bound of a distance parameter [41]. The scheme claimed to be five times faster than the ordinary search with the same accuracy in the results. Another attempt to reduce the computation related to the inherent correlation process is bounded partial correlation (BPC), based on the normalized cross-correlation (NCC) function [42]. This scheme searches a suitable elimination condition at each search location which forms the basis of the upper bound for NCC function. This upper-bounding function incorporates the partial information from the actual cross correlation function and can be calculated efficiently using the given recursive scheme. A simple improvement to the basic BPC formulation is also shown in this reference that provides additional computational benefits and renders the technique more robust with respect to parameters choice. A similar template searching approach was suggested which was based on the Zero mean Normalized Cross-Correlation function (ZNCC) [43]. This scheme achieves a reduced computation level by checking the proposed two sufficient conditions at each search location. These methods rely on the bounded partial correlation that eliminates the points which may not provide a better cross correlation score. A bounded partial correlation method [44] has also been presented, which checks two sufficient conditions at each image positions to achieve a reduced computational level for image matching. In this method, most of the computationally expensive calculations were skipped for those image points that may not improve the best correlation score.

Further attempts in the direction of image correlation are towards modification of the basic kernel design of the process to achieve a better, fast and robust solution. A new image registration method was devised in this context which utilizes the non-parametric (NP) sampling method [45]. This method performs better as compared to

existing registering techniques as the spatial image structures are utilized and no arbitrary selection of the kernel is required. In another proposed technique the basic square error kernel underlying the correlation operation was also modified in weighted least square image matching based target tracking [46]. Similarly, an advanced form of correlation filter is suggested to achieve rotation invariance during pattern recognition [47][48]. Also a general form of a multi-class rotation-invariant filter is suggested which minimizes the average correlation plane error by controlling the side-lobes of this signal resulting in the improvement in discrimination against false targets.

2.2 Chamfer image matching development

Images can also be compared by comparing their edges. An efficient method of image edge matching has been presented known as "Chamfer Matching" [49]. The method proceeds with transforming the edges of the main image into its Distance Transform (DT). This DT is a 2D representation of the image in which the edge pixels are set to zero and the pixels farther away from the edge have an increasing value. The method of transformation works in an iterative fashion starting from a state where edge pixels are set to zero and the rest as infinity. The individual pixel obtains a new value using the following expression at the end of each iteration:

$$\begin{aligned} v_{i,j}^{k} = \min(&v_{i-1,j-1}^{k-1}+4, v_{i-1,j}^{k-1}+3, v_{i-1,j+1}^{k-1}+4, v_{i,j-1}^{k-1}+3, v_{i,j}^{k-1}, \\ &v_{i,j+1}^{k-1}+3, v_{i+1,j-1}^{k-1}+4, v_{i+1,j}^{k-1}+3, v_{i+1,j+1}^{k-1}+4) \end{aligned} \tag{2.2.1}$$

where $v_{i,j}^{k}$ is the value of the pixel in position (i,j) at iteration k. The iterations continue until no value changes. The number of iterations is proportional to the longest edge distance occurring in the image. At the end of the process, the edge pixels attain the value zero and the pixels away from the edge will have higher values

in general. In order to match, the template image edges are superimposed over the distance transform of the main image already formed. An average of the DT pixel values which the template edges hit is called the edge distance. A perfect match will produce the edge distance of zero. Root mean square average can also be used for a better minima location, given as :

$$E(i,j)=\frac{1}{3}\sqrt{\frac{1}{n}\sum_{i=1}^{n}v_i^2} \tag{2.2.2}$$

where v_i are the distance values, n is the number of points in the template edges and $E(i,j)$ is the error function value for the location (i,j).

2.2.1 Modifications in Chamfer Approach

The search offered by the Chamfer image matching scheme is of greater robustness against noise but inherently it proceeds in a correlation fashion which takes a lot of the processing time for verification of the global minima value. The multi-resolution approach however, reduces the level of computations in signal processing [50]. Applying the similar concept to the technique of Chamfer image matching, a further advancement was suggested to utilize an image pyramid for the image edge matching [51]. Here, the matching is performed in a series of images depicting the same scene, but in different resolutions, i.e. in a resolution pyramid. Because of this hierarchical approach significant amount of computations are cut down. The search starts at a low resolution and the results from the low resolution guides the matching at finer levels. The hierarchical Chamfer matching is further refined with a new proposed matching scheme in which DT of both the image and the template are used [52]. The low resolution image search may ignore the fine details of the template which can sometimes be important for the discrimination.

Another approach to reduce the computations was presented which was based on the selection and usage of a few interesting points [53][54]. The algorithm extends the traditional method by introducing interesting points to replace edge points in distance transform for the matching measurement. A series of images with different number of interesting points to feature the original image is created in a pyramid structure through a dynamic threshold scheme. The matching is performed in this pyramid from course level to fine level by minimizing a given matching criterion in terms of distance between selected points of interest. The selection of a fewer points for the image processing is, however, a compromise towards the robustness in speed and performance.

A new image matching strategy was presented, known as center-on fit, based on simulated annealing and Chamfer matching [55]. Simulated annealing is a stochastic optimization technique guiding the results towards the matching solution and is based on the analogy between the annealing of solids and solving optimization problem. Simulated annealing has been applied to a wide variety of image processing applications which simulates the evolution to thermal equilibrium of a solid for a fixed value of temperature T. At each temperature T, the solid is allowed to reach thermal equilibrium, characterized by a probability of being in a state with energy E given by the Boltzmann distribution:

$$p(E) = e^{\left(-\frac{E}{KT}\right)} \tag{2.2.3}$$

where E denotes the energy, K is the Boltzmann constant and T is the absolute temperature (Kelvin). $p(E)$ denotes the probability of a state having energy E and temperature T. Another approach combining the Chamfer match process and the shape matching is presented [56]. This hybrid method utilizes a new form of the shape

matching based on shape context and its correspondences which covers for the shortfalls of Chamfer. The cost function is minimized using the viterbi algorithm around the image feature.

2.3 Image matching in different domains

The image has got multiple forms of its representation. All the different image representations are devised to suite different needs of image coding, transmission and processing. The use of Hough transform for shape recognition in images was explored [57]. Hough transform exploits the straight line feature in an image. Similarly, the statistical processing of the image also makes it possible to reduce some computations while template matching [58][59]. The cost of template matching has been reduced utilizing the probabilistic domain knowledge of the matching images in these techniques.

The use of 2D digital filters for the template image matching was presented [60]. The scheme uses a simple lookup table along with the two-complement image representation to yield good results over a modest amount of hardware. Another template image matching scheme was presented and its limitations were explored where the matching process was expedited in the compressed image domain [61]. A similar new transformation for the image template matching is presented [62]. This transform is defined as grayscale morphological hit-or-miss transform (GHMT) and it was claimed to be robust against Gaussian noise.

A new approach towards image template matching in relation with multi-resolution approach utilizing the ortho-normal wavelet basis was discussed [63]. The use of symmetric convolution along with the discrete cosine transform (DCT) was explored [64] in the similar compression domain for images. A fast template matching method was suggested by using the dual transform of Fourier and the Karhunen-Loeve [65].

The vector subspace spanned by the Eigen-vectors is generated which is then used for the matching of distorted shapes as well. A new approach towards pattern matching was presented which creates a transform space and divides it sequentially to perform Boolean operations on the constraint sets that are defined by template and target points [66]. A fast template matching algorithm was suggested which was based on L-p norm [67][68]. This algorithm produces the same result as full exhaustive template search in Fourier domain once the template and image window is properly partitioned. The fast Fourier algorithm provides an edge over the conventional image matching as the iterative algorithm makes the realization of Fourier transform possible with a low computational level. However, the requirement of transforming the image back to spatial domain for the interpretation of the results is an unwanted overhead. Furthermore, these schemes show low endurance towards the scale and orientation differences between the matching images. Another scheme was proposed for fast template matching which takes the advantage of energy packing property of Walsh-Hadamard transformation [69]. All of these schemes provide efficient image matching directions. The common drawback is the interpretation of the processed data for which the inverse transform has to be done, causing an extra overhead.

2.4 Feature image matching

Image can be represented in the form of the features they possess. The representation can be directed towards achieving a specific goal of image coding, compression or matching. In the fields of image compression and coding, it is desired to achieve the extraction of the original image to a highest possible level, whereas in the domain of image matching, it is not required to regenerate the original image at all [70]. Especially in the application area of visual navigation, only matching coordinates are essentially required to fulfill the need.

Many methods have been proposed for image feature comparison and matching. A pattern matching scheme was proposed based on their correspondences [71]. The finite grid set and distance mapping strategies are utilized and tested on binary images in this scheme. It is also efficient to represent the feature in the images to be matched in the form of descriptors. Different descriptors as Legendre descriptors, Zernike descriptors and two-dimensional Fourier descriptors were explored [72]-[75]. Emphasis was also made on the method of template extraction through which the performance of image matching may be enhanced a lot. An entropy based method was presented which extracts the template from the binary images by eliminating the unreliable pixels [76]. A contour based image matching was presented which employs a key representation of the image contours [65]. This is achieved by a dual transformation scheme both Fourier transform and the Karhunen-Loeve transform and was utilized in matching objects with unknown distortions. Many other forms of image curves representation are discussed [77-87]. These representations facilitate the image matching process in one way or the other. The hybrid approach combining the shape representation along with the multi-scale platform was also explored [88]-[89]. Many other image representations were also explored to achieve a better enhancement goal [90]. A comprehensive comparison of the image features descriptors is presented [91].

New improved methods have been explored in the direction of image matching. Early jump out schemes have been proven to be far more efficient as they truncate the processing cycle when necessary and sufficient conditions for the image matching are attained [92]. This proposed technique is also applicable to most of the motion estimation algorithms. Similarly, the role of neural networks and different training based solutions in target recognition and pattern matching was explored [93]-[95].

Hough transform was explored in object detection was explored which uses clustering representation of the data in space [96][97]. Data bounds were also derived for the schemes to avoid false peaks. Tracking targets in video sequences through a low-order parametric model for motion estimation was also discussed [98]. The method is based on off-line learning of targets for a better and faster online recognition. Still newer and hybrid mechanisms are being explored towards better image perception and faster template matching challenge.

2.4.1 Shapes and Line approaches

Shapes and lines present in the image constitute a powerful feature set of the image suitable for effective image matching. Corners and junctions detection in image edges is utilized in current efficient techniques for the image matching [99-102]. Another algorithm with corner matching approach was presented with a great deal of robustness [103]-[105]. Many other algorithms explored the use of shape descriptors for the effective image matching under invariant parameters [106]-[107]. Many strategies to match images on the basis of shapes and exploring a reduced computation approach were presented in most recent techniques [110]-[113]. Similarly, extraction of geometric image features makes the comparison easier and faster and the results are robust in noisy environment. A new shape matching algorithm is presented [114]. Here the shapes to be matched are registered as a boundary encircling the represented cluster. The matching itself is correlation based once the boundaries are represented by a DT. Representation of the image in the form of shapes of its contents is discussed in many recent techniques [115-120]. Matching the contours present in the image is one way of feature comparison [121]. Inclined targets are also compared in this study using a voting mechanism.

Multiple similarity measures for the comparing images were devised for faster and robust image matching. A concise comparison between a few similarity-measures used in medical imaging was presented using the result of computed tomography (CT) [131]. Classification of image features is also a part of the image understanding. A new statistical measure for the identification of image spatial features as textures was proposed [132]. This scheme utilizes a directional RANK-strength vector for classification and has produced good results. Similarly, another texture classification technique was presented which is orientation and resolution (scale) insensitive [133]. The algorithm utilizes a 2D linked list to tune a mask over the normal and multi-scale and orientation Brodatz data set.

2.4.3 Image Vectorizations

The modern form of image vectorization is a new direction for image matching area. Image representation space is also explored to find better ways for faster matching [134]. A vectorization technique to be used in data compression was presented which utilizes half-L2-norm pyramid data structure for effective data encoding [135]. Hyper-dimensional vectors in association with larger codebooks outperform many of the other existing techniques. Similarly, an extension to the feature image matching is proposed [136] in which component block projections (CBP) are used in the form of concatenated directional vector for the efficient image template matching. Zernike moments have also been used [137] as a powerful image descriptors. A relationship between geometric and Zernike moments was suggested to reduce the inherent complexity of Zernike moments to achieve a real-time processing of images upto 4 Mega pixels. Another similar implementation of hyper-plane image representation

the use of reduced set of interesting points extracted from the image feature for reduction of computation in the template image matching. A study was presented towards the effect on this selection once the template image is scaled [147]. Similarly, the extraction of distinctive image feature point set was discussed [148-151] having a level of affine and illumination invariance. Another strategy for the selection of only useful image features for the template matching was presented based on principal component dimensionality reduction to form a reduced processing space [152][153][161]. Similarly, the redundant features were removed using the classical factorization theorem.

A feature based image matching method was presented for the selection of landmark in satellite images [154]. It extracts a small set of features by decomposing the world into small number of maximally sized regions for robotic navigation. Although, this scheme can work in a cluttered environment but it is not suitable for visual navigation in aerial vehicle.

The trend of the techniques used for image matching is now gradually tending towards advanced feature matching schemes. The emphasis of the proposed work is to establish a unique vectorized feature representation. The representation can thus be utilized for the effective image matching exploring the matching process in the transform domain.

Chapter 3

Template Image Matching Through Hyper Vectorization Using Modified Radon Transform

In this chapter, the theory of Radon Transform (RT) has been presented in context of its ability towards enhancement and projection of straight segmented image edges. Then, the formulation of the RT methodology is explained on the basis of simple examples towards image matching. The limitations of RT towards the practical use has also been highlighted which further becomes the basis of modifying the RT approach to formulate an algorithm to form a practical and robust image matching scheme. The new methodology is explained along with the experimental setup and discussion of the results.

3.1 Radon Transform

RT is a parametric transform which converts the spatial information in an image into the consecutive parametric domain. It is another way of representing the same information which highlights the edges in an image in the transform domain. RT of a 2-D function $f(x,y)$ is defined as follows [155]:

$$g(s,\theta)=\mathbb{R}\left[f(x,y)\right]=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(x,y)\delta(x\cos\theta+y\sin\theta-s)\,dxdy \qquad (3.1.1)$$

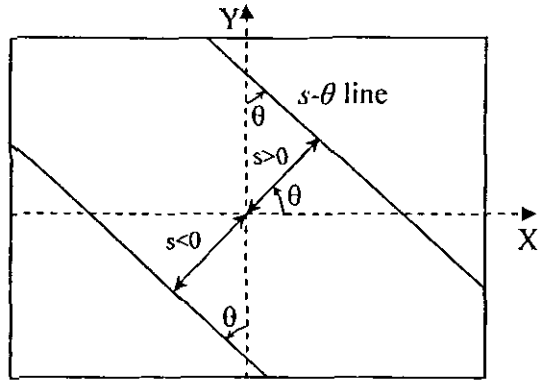
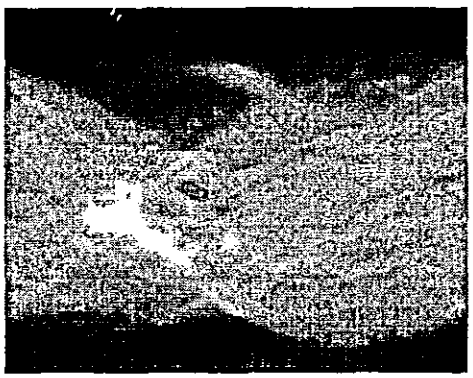


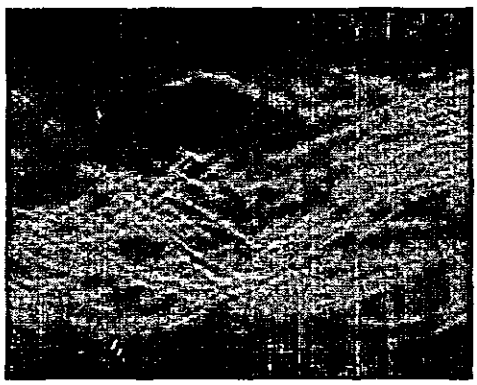
Fig. 3.1 Radon Transform (RT) Line of integration.

where $-\infty < s < \infty$ and $0 < \theta < \pi$. The angle θ is the angle formed by the line of integration with respect to y axis as shown in Fig. 3.1, while s is the perpendicular offset from the origin to the line of integration and it can be positive or negative. The positive and the negative values of s are interpreted as shown in Fig. 3.1. In this way the 2-D kronecker delta function $\delta(x\cos\theta + y\sin\theta - s)$ extracts a line slice out of the function $f(x,y)$. This line will be referred to as $s-\theta$ line. Hence, $g(s,\theta)$ is 1-D projection of $f(x,y)$ at an angle θ with offset s from the origin. In our case $f(x,y)$ is a discrete 2-D function (or image). Thus the integration operation in RT is converted to a summation. Hence, the RT simply adds up the gray scale pixel values lying on the $s-\theta$ line. Similarly, if the edge detected binary image is considered, its RT simply reduces to counting the ones constituting the edges lying on the $s-\theta$ line. The difference of the two is shown in Fig. 3.2 and Fig. 3.3. Fig. 3.2 (a) shows a gray scale image and its RT has been presented in Fig. 3.2 (b). Similarly, an edged binary image is shown in Fig. 3.3 (a) and the corresponding RT is displayed in Fig. 3.3 (b). The integration limits in equation (3.1.1) are from $-\infty$ to $+\infty$. The image function $f(x,y)$ has finite dimensions (M,N) such that $1 \le x \le M$ and $1 \le y \le N$, thus the integral limits will automatically reduce to a finite



(a) (b)

Fig. 3.2 RT of a gray scale image (a) Image (b) RT



θ (degrees)

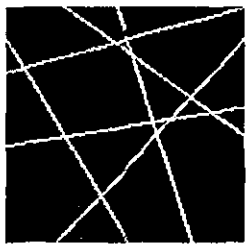
(a) (b)

Fig. 3.3 RT of an edged binary image (a) Image (b) RT.

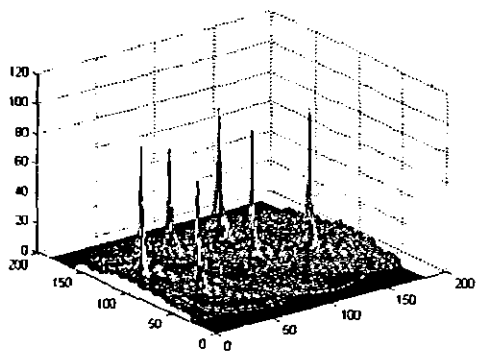
value and the maximum value of s for which $g(s,\theta)$ will be given as:

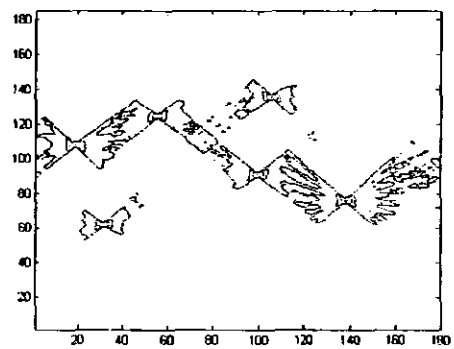
$$s_{max} = \sqrt{\left(\frac{M}{2}\right)^2 + \left(\frac{N}{2}\right)^2} \tag{3.1.2}$$

The brighter spots in Fig. 3.2 (b) represent the orientation of the edge patterns of the gray scale image of Fig. 3.2 (a). Similarly, in Fig. 3.3 (b) the same patterns are more visible and distinguishable because the edged binary image of Fig. 3.3 (a) only shows the edges. The white spots can easily be localized and spotted in Fig. 3.3 (b) than compared to the Fig. 3.2 (b).



(a) (b)





(c) (d)

Fig. 3.4 (a) A binary test image (b) Radon Transform (c) RT Surface (d) Contour plot of the RT peaks.

3.2 Surface Interpretation in Radon Transform

A test image of 200×200 pixels, is considered in Fig. 3.4 (a) with six simulated vital edges having different orientation across the image. Its RT is shown in Fig. 3.4 (b) in which six bright spots can be easily demarcated. The location of these bright spots in the transform domain corresponds to the prominent edges in the images in the spatial domain. If the values in the transform domain are plotted as a 3-D mesh surface, the peaks are well prominent as compared to the rest of the radon surface as shown in Fig. 3.4 (c). Similarly, the contour plot of Fig. 3.4 (d) also demonstrates the sudden change in altitude of the radon surface.

(a) Template 1 from (10, 75)

(b) Template 2 from (93, 28)

(c) Template 3 from (300, 260)

(d) Template 4 from (80, 260)

Fig. 3.5 Template taken from different locations of the test image.

Table 3.1 Radon transform parameters θ and s for the image of Fig. 3.4 (a)

θ	s
19	60
32	-124
56	128
100	-5
106	173
138	-66

The spread of the radon surface corresponding to each peak constitute a butterfly type pattern signifying the near zero values as the $s-\theta$ line is getting perpendicular to the image edge. It is also to be noted that six edges in Fig. 3.4 (a) correspond to the relative six peaks of the radon surface of Fig. 3.4 (c). The peaks corresponding to each image edge can thus be detected from the radon surface and are listed in table 3.1 displaying the orientation θ and offset s for each of the six edges. Four example templates each of size 100×100 pixels are then extracted from the image of Fig. 3.4 (a) from the locations (row, column) as shown in Fig. 3.5 (a) to (d). Each of the templates thus obtained has been subjected to RT and peak detection and location of the peaks were found. These peaks corresponding to each template image

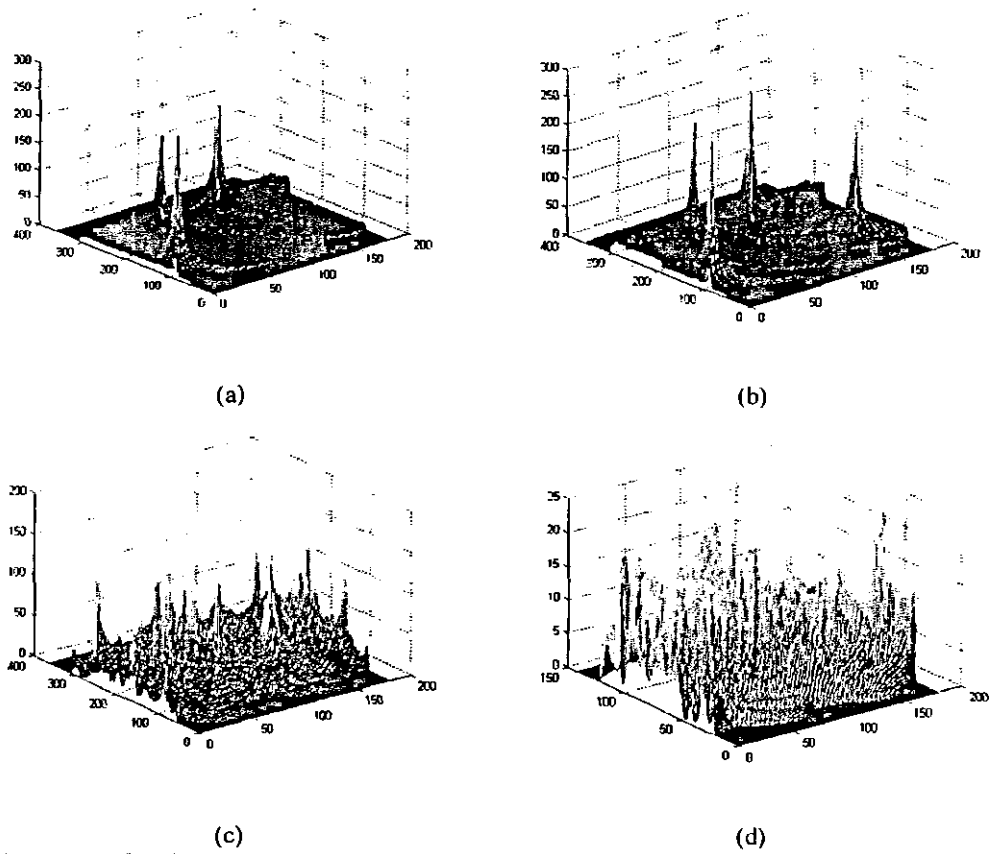
Table 3.2 Edge orientation listings of the template images of Fig 3.5

Template 1		Template 2		Template 3		Template 4	
θ	s	θ	s	θ	s	θ	s
31	-24	56	47	19	-12	32	10
106	-5	106	-28	138	38	100	27

have been listed in table 3.2 corresponding to template 1 to 4, respectively. It is to be noted that the data given in these tables do not directly corresponds to the data of the main image as the values of θ and s are relative to the particular image center. As each template image contains two edges and their orientation can be matched with that of the edges in the main image of Fig. 3.4, mathematical correlation shows the estimate of the template location as listed in table 3.3. It is required to have a minimum of two different edges to be present in the template image as essential condition for the estimation of the template extracted location. The accuracy of this estimation can be enhanced if either more edges are present in the template image or bigger vital edges with accurate orientation data can be recorded. Finding the location of the given template images of Fig. 3.5 in the main image of Fig. 3.4 (a) through conventional correlation based techniques requires a lot of computations as it involves the comparison of all template pixels with that of corresponding main image pixels for each probable template matching location. These comparison operations count to a total of 10^8 for the main image of size 200×200 pixels and template of size 100×100 pixels (10^4 pixels comparison for one template location by 10^4 probable template matching locations). The comparison may differ from that of a simple subtraction to a correlation based square error sum. Whereas, the estimation of the template image locations through the orientation information of the edges only take a few trigonometric and logical operations for the same results. In the quoted example,

Table 3.3 Actual and Calculated Locations.

Template Locations		Calculated Locations	
X	Y	X	Y
10	75	7.8	77.6
93	28	90.1	30.1
300	260	299.8	257.2
80	260	81.9	260.3



(a) (b)

(c) (d)

Fig. 3.6 The flooding of peripheral surface as the number of peaks increases in images with (a) 3 (b) 4 (c) 11 and (d) 50 edges.

once the orientation θ and offset s of the template edges are given, the proposed method only takes 6 logical comparisons 2 subtractions and a vector addition to locate the template image center with respect to the main image center.

3.3 Problems in a Practical Case

RT is a parametric transform and it detects the orientation of an edge in the image. The integration or summation operation involved in RT creates a peak on that location. Any image with three vital edges will create three peaks at different location in RT domain as shown in Fig. 3.6 (a). Although, the peaks are prominent and can easily be located in the transform signal, yet the peripherals produced which is associated by the peak cannot be ignored completely. These insignificant peripheral values of the radon surface cannot be indistinguishably neglected once the number of

image edges increased. Images with 3, 4, 11 and 50 edges are considered and there RT surfaces are plotted in Fig. 3.6 (a) to (d), respectively. With the increase in image edges, the peaks of the radon surface losses their significance as they submerge in the flooding peripheral surface. Hence, the RT cannot be directly used for the practical image matching case because the number of edges may cause a bigger portion of the radon surface peaks to become undistinguishable. Thus a modified form of RT is presented for the use in image matching applications.

3.4 Proposed Image Matching Technique

Image matching is the process of finding the location in the given main image that is similar to a given pattern template. Let the main image be represented as $f(x,y)$ with dimensions (M,N) such that $1 \le x \le M$ and $1 \le y \le N$. Similarly, the pattern template image be represented as $\bar{f}(\bar{x},\bar{y})$ with dimensions $(\bar{M},\bar{N})$ such that $1 \le \bar{x} \le \bar{M}$ and $1 \le \bar{y} \le \bar{N}$. The proposed image matching scheme finds whether the template image $\bar{f}$ is a part of the main image f or not. If $\bar{f}$ is a part of f, it also finds the best matching location as:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} C_{fl} \\ R_{fl} \end{bmatrix} + \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} \tag{3.4.1}$$

where C_{fl} and R_{fl} are the translations in x and y direction, respectively, to be found. Fig. 3.7 shows the main steps of the proposed scheme. First of all edge detection is performed on both the main image and the template. The binary images, thus, produced are then vectorized to form hyper vector tables (HVT) for both images. Finally, vector matching is performed on these HVT to ascertain the match/mismatch and the best template matching location.

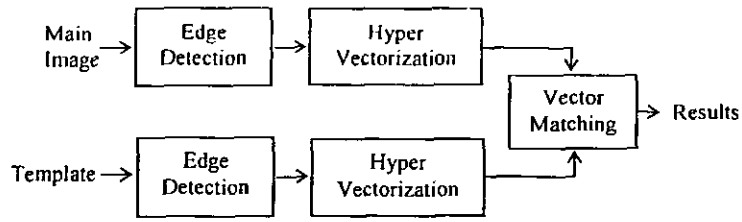
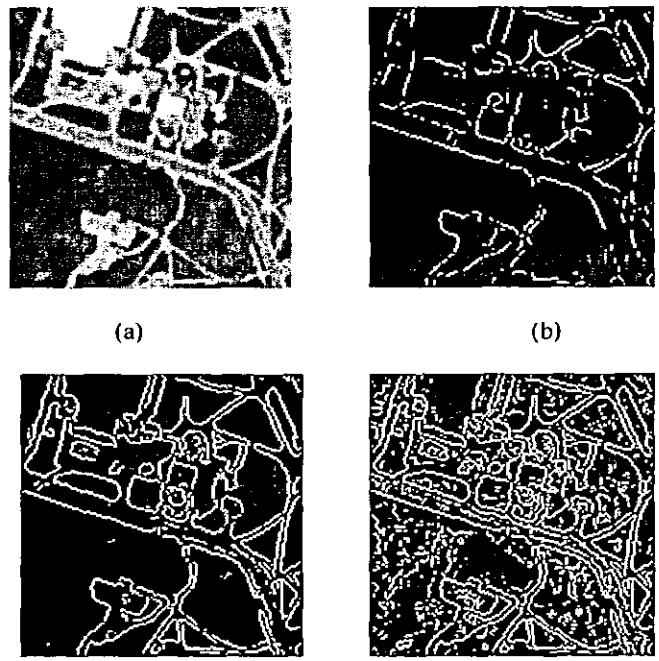


Fig. 3.7 Main steps for the proposed image matching scheme.



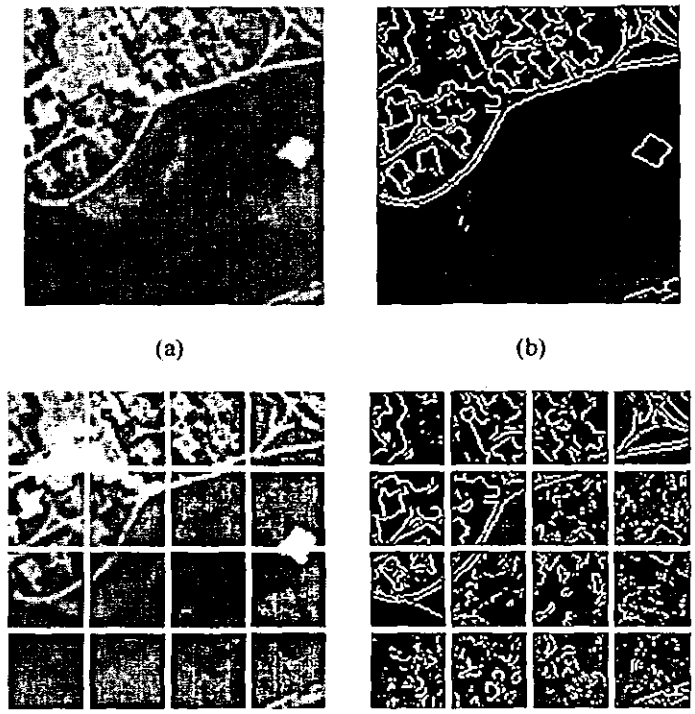
(a) (b)

(c) (d)

Fig. 3.8 (a) Original Image (b) Less Amount of Edges
(c) Appropriate Edge Detection (d) More then desired Edges.

3.4.1 Edge Detection

The purpose of edge detection is to detect a suitable amount of prominent image edges which can be considered as a good representation of the main features in the image. The Sobel method is used to detect the edges with both the horizontal and vertical high-pass masks [11]. The threshold value used in the final binarization step of Sobel algorithm is directly related to the number of detected edges in the image. The variation of the threshold value and its impact on edge extraction is elaborated in Fig. 3.8. The threshold value is low in the image of Fig. 3.8 (b) and it is respectively



(a) (b)

(c) (d)

Fig. 3.9 The effect of image segmentation in spatial domain on the number of extracted edges. (a) Example image (b) Edge detection with one global threshold (c) Segmented image (d) Edge detection after image segmentation.

increased in Fig. 3.8 (c) and Fig. 3.8 (d). The edges are also increased in the same manner. It is desired that the value of this threshold is so selected that 10-15% of the total image pixels may appear in the form of edges for the uniformity in the comparison process. This method gives essentially enough number of edges suitable for the proposed edge matching scheme. Similarly, the use of one global threshold for edge extraction may sometimes create bigger black portions in the image without any significant edges. An example in point is the image considered in Fig. 3.9 (a). It uses one global threshold value for edge detection results in the binary image as shown in Fig. 3.9 (b). It is to be noted that Fig. 3.9 (b) has got a large portion of black patch revealing no information about the edges there. The segmentation of the image in Fig. 3.9 (a) can resolve this problem. The segmented image is shown in Fig. 3.9 (c) and the application of different threshold values for each segment image results in better edge

extraction as shown in Fig. 3.9 (d). Let b and $\bar{b}$ be the edged binary images of the gray images f and $\bar{f}$, respectively.

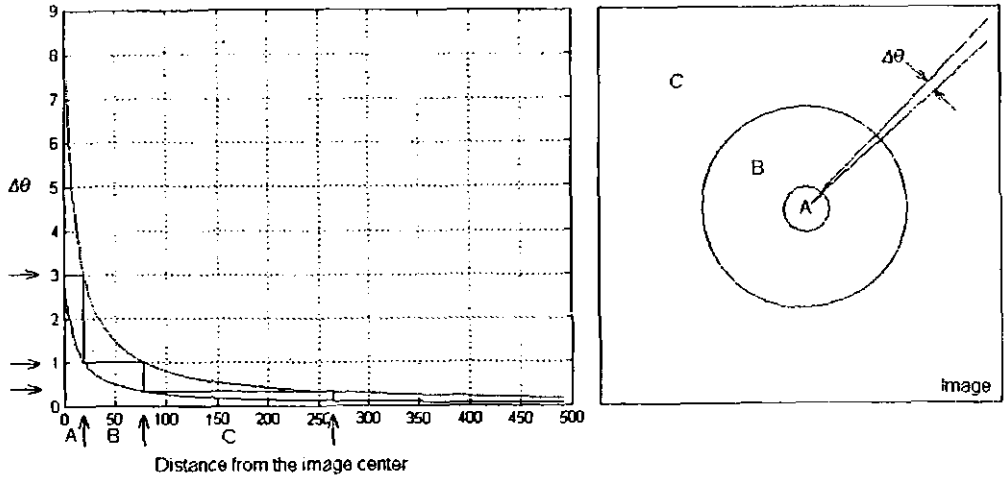
3.4.2 Hyper Vectorization

A modified Radon Transform based hyper-vectorization method has been proposed to represent the edged binary images in the form of feature vectors. A 2-D kronecker delta function $\delta(x\cos\theta + y\sin\theta - s)$ has been multiplied to extract a line slice out of the function $b(x,y)$ to form $g(s,\theta)$ for each value of s and θ, in order to vectorize the binary image b.

$$g(s,\theta) = b(x,y)\delta(x\cos\theta + y\sin\theta - s) \qquad -s_{max} \le s \le s_{max}, 0 \le \theta < \pi \tag{3.4.2}$$

The factor s varies in the range $-s_{max}$ to $+s_{max}$ beyond which the function $g(s,\theta)$ will have zero values. In order to scan the whole image in a complete manner, it is recommended to use the discrete steps of parameter s as $\Delta s = 1$. Whereas, the angle θ varies from 0° to 179° with discrete steps of $\Delta\theta = 1''$. The value of $\Delta\theta = 1''$ is selected for the sake of simplicity and to scan the image with a reasonably good resolution. Too low a value for the $\Delta\theta$ will cause over scanning of the image edges and one vector will be recounted many times in this way, whereas higher values of $\Delta\theta$ may cause some missing radial segments specially in the farther regions of the mage. The recommended value of $\Delta\theta$ step is plotted in Fig. 3.10 (a) between the two bounds of acceptable regions.

One can use a value less than this, but it will produces more vectors and will also increase the computations as well. Furthermore, the technique presented in this dissertation does not need all of the image edges to qualify a match or a mismatch.



(a) (b)

Fig. 3.10 (a) Recommended value for the factor $\Delta\theta$ step size (b) Scanning the circular regions.

The corresponding Fig. 3.10 (b) shows three regions in the image marked as "A", "B" and "C". Region "A" corresponds to the circle of 20 pixel radius in which the value of $\Delta\theta$ should be taken as 3° as shown in Fig. 3.10 (a). Similarly, for the region "B" and "C" these recommended values are 1° and 0.5°, respectively. For the regions larger than "C" extending to more than 260 pixels from the image center, this value must further be reduce to 0.25 for a good image coverage. The upper and lower bounds of $\Delta\theta$ for a good coverage are given as:

$$\tan^{-1}\left(\frac{1}{x}\right) \leq \Delta\theta \leq 2\tan^{-1}\left(\frac{1}{x}\right) \tag{3.4.3}$$

where x is the distance from the image center in pixels. The upper and lower bounds of the $\Delta\theta$ given by eq. (3.4.3) ensures the step size in a suitable range as to pick all pixels integrally and to avoid sub-pixels interpolated values.

Each binary image $g(s,\theta)$ will then be checked for a connected digital path from pixel $p(c_0, r_0)$ to pixel $q(c_L, r_L)$, such that pixels (c_i, r_i) and (c_{i-1}, r_{i-1}) are 8-adjacent

for $1 \le i \le L$ with $V = \{1\}$ [11]. This digital path will be qualified to be listed in the HVT if $L > \tau$, where τ is a small positive integral constant value whose value is considered as 8 pixels for the experiments whose results are quoted here. Once the digital path is qualified, only one of the two end points will be listed in HVT as (θ, L, C, R) as follows:

$$(\theta, L, C, R) = \begin{cases} (\theta, L, c_n, r_n) & \text{if } c_n > c_n \text{ or if } c_n = c_n \text{ and } r_n > r_n \\ (\theta, L, c_n, r_n) & \text{if } c_n < c_n \end{cases} \tag{3.4.4}$$

Similarly, all the entries from all binary images $g(s, \theta)$ will be listed in HVT. Let the main image HVT have N_I vectors denoted as $I_n(\theta_n, L_n, C_n, R_n)$ with $1 \le n \le N_I$. Similarly, the template has N_T vectors denoted as $T_m(\theta_m, L_m, C_m, R_m)$ with $1 \le m \le N_T$.

3.4.3 Vector Matching

The proposed vector matching algorithm consists of pair formation, main image subset extraction, evaluation of potential solutions, vector matching, solution grid formation and template location estimation. The flow chart of the matching algorithm has also been shown in Fig. 3.11. The steps of the algorithm are explained as follow:

Pair Formation: First of all pairs are formed which consist of two distinct hyper-vectors taken from the template HVT. For this purpose, the template HVT is divided into two groups and the pair formation is done as follows:

$$(T_j, T_k) \qquad \begin{matrix} 1 \le j \le N_T - 1 \\ j+1 \le k \le N_T \end{matrix} \tag{3.4.5}$$

Main Image Subset Extraction: Each pair formed, in the previous step, will then be used to formulate two subsets from the main image HVT on the basis of their edge orientation 'θ', i.e. close to the orientation of the template pairs. Consider the following pair is selected:-

$$T_j(\theta_j, L_j, R_j, C_j), \quad T_k(\theta_k, L_k, R_k, C_k) \tag{3.4.6}$$

then, the two subsets from the main image HVT are formulated as follows:

$$I_p(\theta_p, L_p, R_p, C_p) \qquad U \le p \le V \tag{3.4.7}$$

and

$$I_q(\theta_q, L_q, R_q, C_q) \qquad W \le q \le X \tag{3.4.8}$$

such that $abs(\theta_q - \theta_k) \le 2$ degrees and $abs(\theta_p - \theta_j) \le 2$ degrees where $1 \le U, V, W, X \le N_I$.

Potential Solutions: The potential solutions can be calculated by taking one entry from subset I_p and the second entry from subset I_q. There will be $(X - W + 1)(V - U + 1)$ pairs. All those pairs that satisfy the following two constraints will be considered as potential solutions if it is known that, the template is of the same scale as that of the main image.

$$abs\big((R_p - R_q) - (R_j - R_k)\big) \le 2 \qquad pixels \tag{3.4.9}$$

$$abs\big((C_p - C_q) - (C_j - C_k)\big) \le 2 \qquad pixels \tag{3.4.10}$$

The pairs satisfying the above constraints will qualify as potential solutions to be used in the next step of the algorithm (vector matching) with ξ_R and $\xi_{t'}$ both equal to 1.

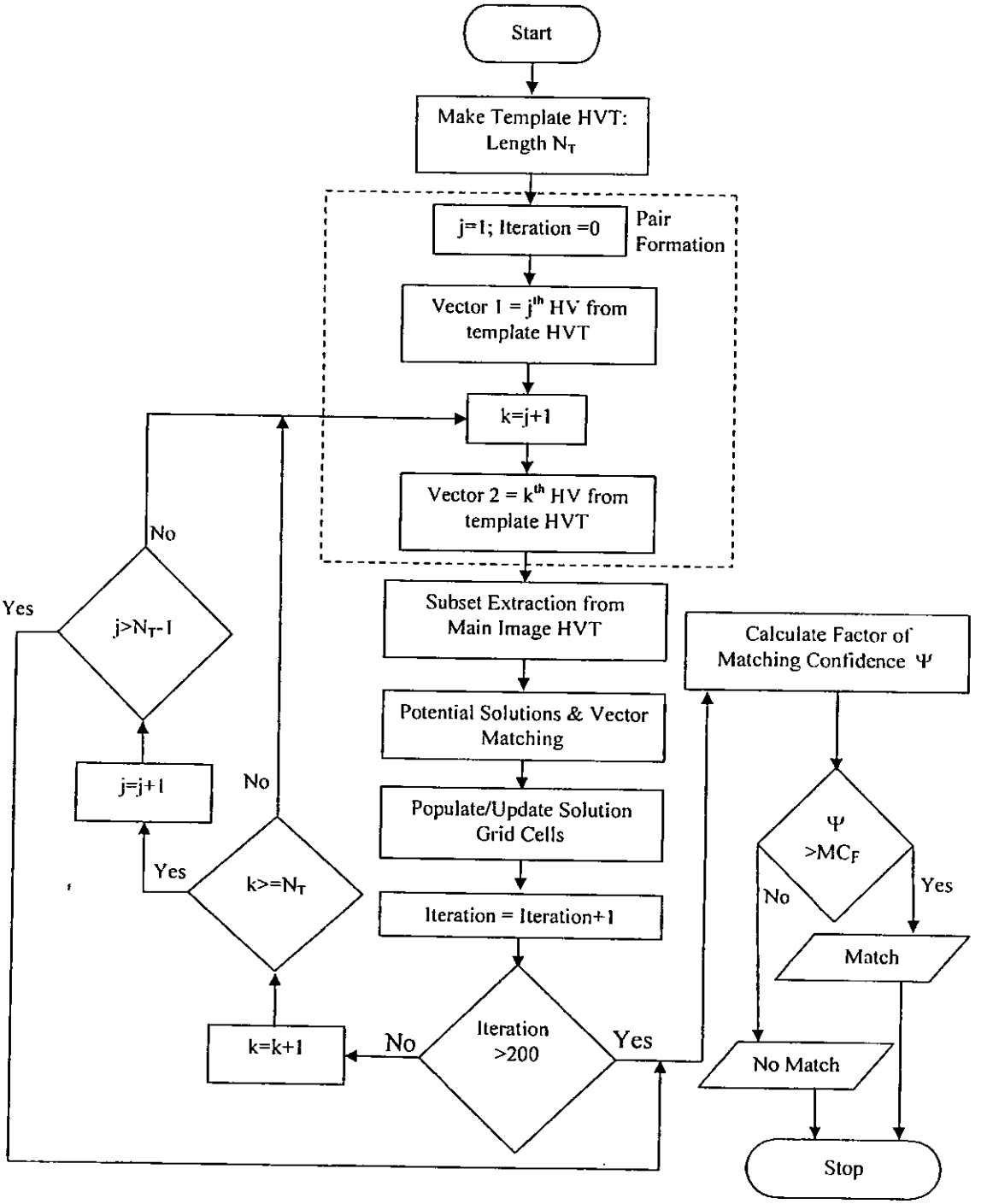


Fig. 3.11 Flow Chart for vector matching in the proposed scheme.

The two scaling factors have been defined to achieve the scale invariant matching which are given below:

$$\xi_R = \left(\frac{R_p - R_q}{R_j - R_k} \right) \tag{3.4.11}$$

$$\xi_C = \left(\frac{C_p - C_q}{C_j - C_k} \right) \tag{3.4.12}$$

All of the pairs which satisfy the following three constraints will qualify as potential solution:

$$\xi_C > 0 \tag{3.4.13}$$

$$\xi_R > 0 \tag{3.4.14}$$

$$\begin{aligned} abs[\xi_R - \xi_C] < 0.2 \quad & if \quad \xi_R < 1 \quad and \quad \xi_C < 1 \\ abs\left[\frac{1}{\xi_R} - \frac{1}{\xi_C}\right] < 0.2 \quad & if \quad \xi_R > 1 \quad and \quad \xi_C > 1 \end{aligned} \tag{3.4.15}$$

The pairs satisfying the above constraints will qualify as potential solutions to be used in the next step of the algorithm. The constant value of 0.2 (which should ideally be 0) is determined experimentally to accommodate the discrete integral mapping of the floating values in the image domain and, at the same time, to avoid the wrongly scaled potential solutions to become the part of the grid arrays. Also if $\xi_R < 1$ and $\xi_C < 1$, the template is of greater scale and vice versa. It is assumed that the aspect ratio is maintained while scaling the template image. The difference of the predicted scales ξ_R and ξ_C can be directly compared if both of them are less than 1, otherwise it is preferred to compare the reciprocals of the both, as mentioned in the second part of the equation (3.4.15).

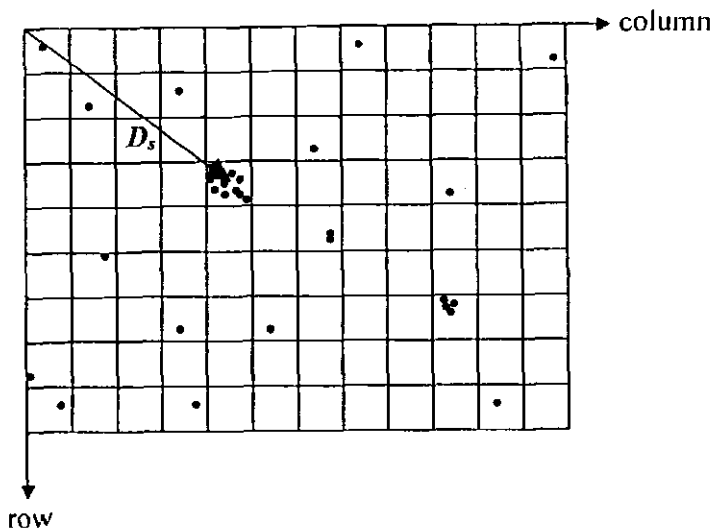


Fig. 3.12 The concept of solution mapping in a grid cell array.

Vector Matching: Each potential solution from the last step will then produce a displacement vector $D(R_D, C_D)$, where either

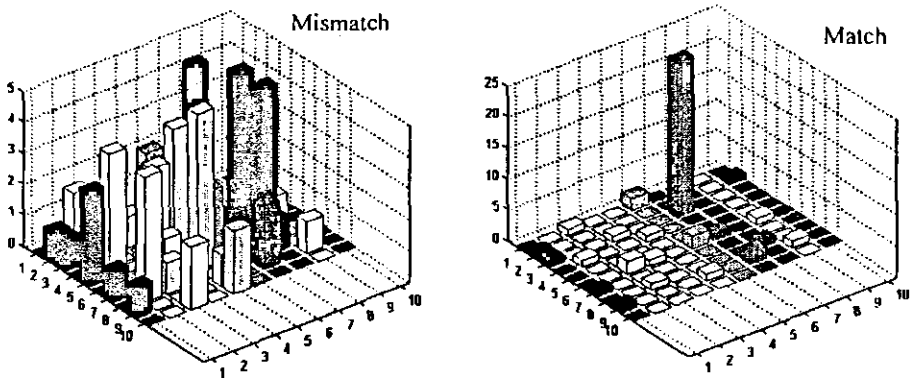
$$\begin{aligned} R_D &= R_p - R_j.\xi_R \\ C_D &= C_p - C_j.\xi_C \end{aligned} \tag{3.4.16}$$

or

$$\begin{aligned} R_D &= R_q - R_k.\xi_R \\ C_D &= C_q - C_k.\xi_C \end{aligned} \tag{3.4.17}$$

Both the options (either equation (3.4.16) or equation (3.4.17)) are acceptable as they cause a difference of 2 pixels within the solutions. The factors R_D and C_D formulate the required potential answer to the matching problem and it describes the template location in terms of rows and column in the main image.

Solution Grid Formation: Each displacement vector entry D from the last step will then be used to map the solution in a grid cell array. Grid cell array is composed of square cells in a grid formation. Each grid cell may get one or more solution counts



(a) (b)

Fig. 3.13 Grid array population distribution for (a) Non-matching and (b) Matching template cases.

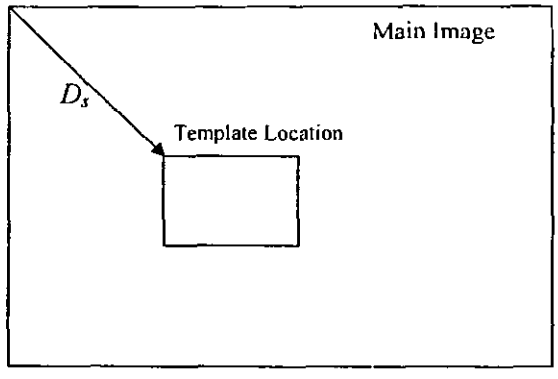


Fig. 3.14 Vector matching results with final displacement vector D_s

during the process of mapping of these displacement vectors. The center of the grid cell with maximum number of solution counts will be considered as the final template matching solution D_s as shown in Fig. 3.12.

Template Location Estimation: If the distribution of the population in the different grid cells is not varying much as compared to each other, the algorithm will then declare a mismatch as shown in Fig. 3.13 (a). In case if one of the grid cells is heavily populated as compared to the rest, as shown in Fig. 3.13 (b), it will be considered as a match. The center of this grid cell will then be considered as the final solution with distance D_s from the top left corner. Fig. 3.14 shows the location of the template in the main image through the vector D_s.

The confidence factor expressed in the flow chart is defined as follows:

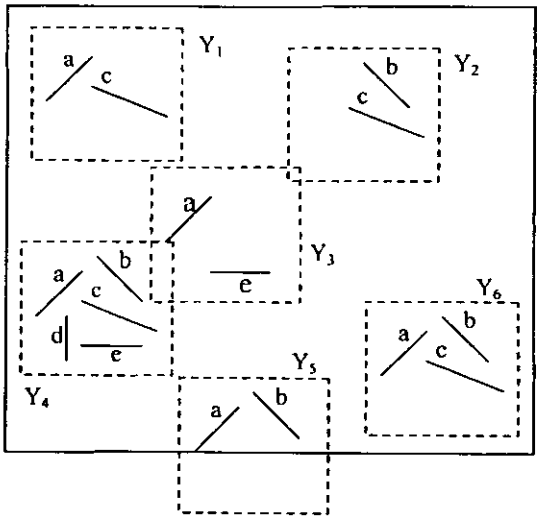
$$\Psi = \frac{\gamma_{MAX} - \bar{\gamma}_R}{\gamma_{MAX}} \times 100 \qquad (3.4.18)$$

where γ_{MAX} is the count of the most densely populated grid cell and $\bar{\gamma}_R$ the mean count of the rest of the non empty grid cells with count more than $0.25\gamma_{MAX}$. This figure is determined experimentally to ignore the grid cells with smaller count and the matched and non-matched cases are thus totally separated out on the basis of the value of Ψ. The plots of the confidence factor Ψ show initial random transients in the value as the grid cell array is getting populated. It is found experimentally that after about 200 iterations, the value of Ψ settles greater than MC_F for the matching template cases and it remains less than MC_F for non-matching templates.

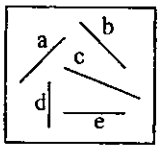
3.5 Results and Discussion

The experiments have been designed to evaluate the performance of the proposed algorithm. The algorithm has been tested on a single test image, a real satellite image and on 30 different satellite images from USC-SIPI-HAAI database. The USC-SIPI-HAAI images contain a variety of aerial satellite images having multiple ground features covering almost all kind of scenarios to be faced by an aerial vehicle. These multiple experiments confirm the scale and rotation endurance of the proposed algorithm in comparison with three other image matching techniques. Similar experiments have also been performed separately for the speed analysis of the proposed algorithm with respect to others methods under similar testing conditions. The noise analysis has been provided to evaluate the performance of the proposed algorithm under noisy conditions.

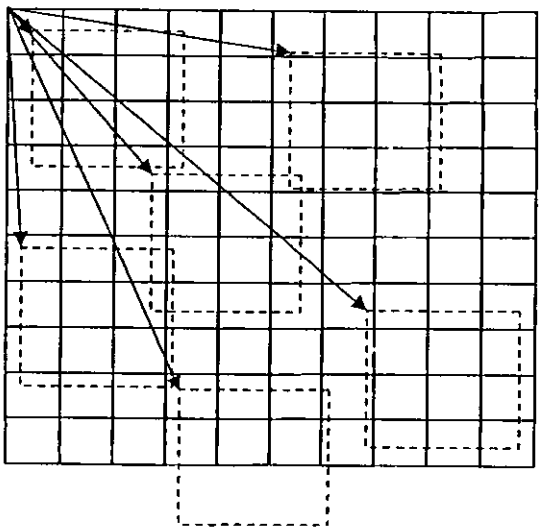
A simple example for the simulation of the proposed image matching algorithm is presented. The main image is shown in Fig. 3.15 (a) with many edges. These edges



(a)



(b)



(c)

Fig. 3.15 The main image, template and grid cell array of a simple example
(a) Main image for simulation test (b) Template image with five edges (c)
Grid cell array solution superimposed on the displacement vectors.

are denoted by small alphabets. The template is shown in Fig. 3.15 (b). It has five straight edges 'a' to 'e'. The main image has got the same orientations of the edges as present in the template but the perfect match should refer to solution Y_4. Fig. 3.15 (c)

Table 3.4 Edge Pairs and their contribution towards solution.

Edge Pair		Solution Counts						
		Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	ψ
a	b	0	0	0	1	1	1	0
a	c	1	0	0	1	0	1	33.3
a	d	0	0	0	1	0	0	33.3
a	e	0	0	1	1	0	0	50.0
b	c	0	1	0	1	0	1	40.0
b	d	0	0	0	1	0	0	50.0
b	e	0	0	0	1	0	0	57.1
c	d	0	0	0	1	0	0	62.5
c	e	0	0	0	1	0	0	66.6
d	e	0	0	0	1	0	0	70.0
Total		1	1	1	10	1	3	

shows the potential solution locations superimposed on a grid cell array underneath. Let us consider the template edge pair 'a' and 'b'. For this edge pair, three solutions in the main image will qualify as this edge pair is present in the main image with the same orientation at three different locations (i.e. Y_4, Y_5 and Y_6). The next step is to list out all considered edge pairs and their results in the Table 3.4. The '1' in the Table 3.4 means that it qualifies as a potential solution and '0' means that it does not qualify. The solution count for Y_4 is maximum i.e. 10 and hence, will be considered as the final matching point. The grid array population distribution is shown in Fig. 3.16 at the end of the matching algorithm. Grid cell at sixth row and first column have got the highest solution score of '10' pointing towards Y_4 of Fig. 3.15 (a). The value of the confidence factor Ψ is updated with each new qualified edge pair entry and the new value is listed in the last column of Table 3.4. This trend of confidence factor Ψ has been plotted in Fig. 3.17. As more and more edge pairs are taken into account, the general trend of the confidence factor gets closer to 100%. The confidence factor Ψ serves as discriminatory property which classifies between a matching and an unmatched template image.

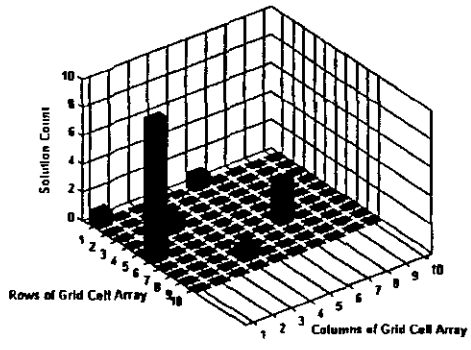


Fig. 3.16 Grid Array Population Distribution.

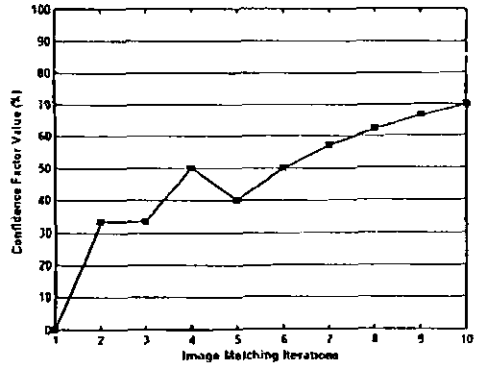


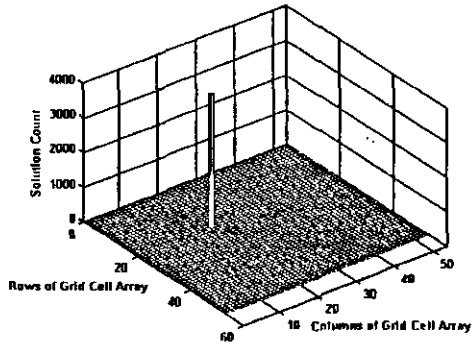
Fig. 3.17 The value for the Confidence Factor for each vector matching iteration.

Fig. 3.18 Example Real Satellite Image, with white square pointing out the location of the template and the vector D_S giving the final solution.

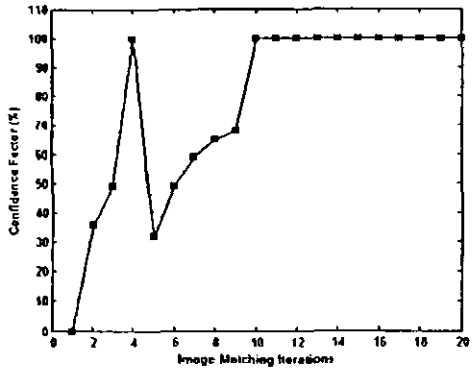
3.5.1 Template Image Matching of a Real Satellite Image

In order to explain the performance of the proposed algorithm on a real satellite image, a sample image is considered as shown Fig. 3.18. The template image has been extracted from the location D_s(200,200). The white square in the Fig. 3.18 shows the template image boundary. This matching template image is shown separately in Fig. 3.19 (a). The main image of 512×512 pixels has been vectorized to form 3896 vectors. Similarly, the matching template has also been vectorized to form 124

(a)



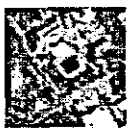
(b)



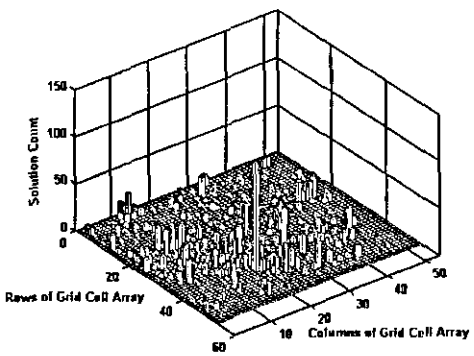
(c)

Fig. 3.19 The results of a matching template image that belongs to Fig. 9
(a) Matching template image (b) Solution Distribution in grid cell array
(c) Confidence factor growth trend.

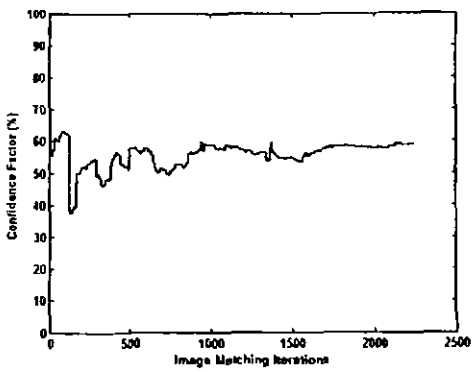
vectors. This makes 7626 template edge pairs to be compared with the HVT of the main image. Each template edge pair can score multiple qualifying solution entries in the grid cell array. A total of 10368 qualified solutions have been obtained. These solutions are populated in a grid cell array. The size of each cell is of 10×10 pixel. The maximum score is contained by the grid cell at 20^{th} row and 20^{th} column of 3841 which points towards the final solution vector $D_s(200,200)$, as shown in Fig. 3.20 (b). The rest of the qualified solutions are scattered insignificantly over the rest of the grid



(a)



(b)



(c)

Fig. 3.20 The results of a non matching template image that does not belongs to the image shown in Fig. 9 (a) Un-matched template image (b) Solution distribution in grid cell array (c) Confidence factor growth trend.

cells. The value of the confidence factor Ψ was updated with each entry of the qualified solution in the grid cell array. The trend of this factor was plotted in Fig. 3.19 (c) for the first 20 iterations. This value settles to 100% at 10^{th} iteration and remains there for the rest of all iterations as the template is a perfect matching one.

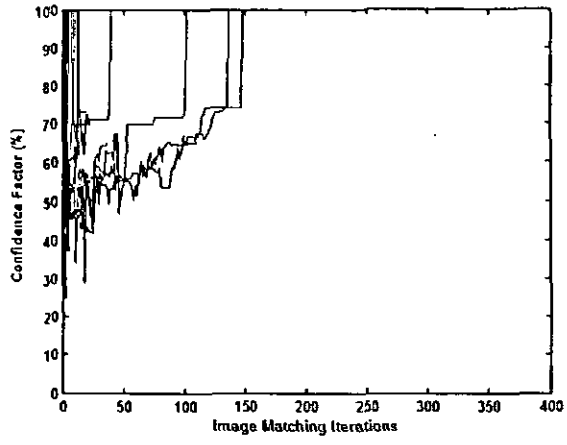
Similarly, another template image was taken, as shown in Fig. 3.20 (a), which is not the part of the main image in Fig. 3.18. The vectorization forms 65 vectors from this template image. This makes a total edge pairs of 2080. A total of 2250 qualified

solution entries were obtained which were populated on a grid cell array of each cell size of 10×10 pixels. This population distribution is shown in Fig. 3.20 (b). The maximum score is 114 of the cell at 40th row and 20th column which shows a best matching location of this template at D_s(400,200). This figure of 114 is also not significantly larger than the rest of the grid cell score. The trend of the confidence factor Ψ is also plotted in Fig. 3.20 (c) for the complete 2250 qualifying solution entries. The final value of the confidence factor Ψ is 67.41%. It is to be noted that after the initial transients, this value remain below 70% throughout the matching process.

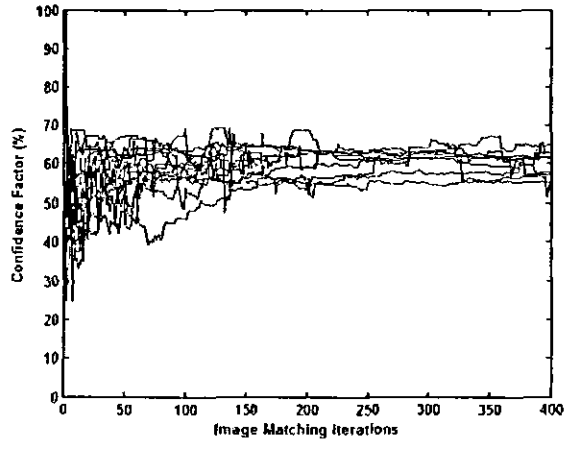
Therefore, the steady state value of confidence factor Ψ at 100% shows a match and less then 70% can be considered as un-matched template. The trend of the confidence factor growth with respect to vector matching iterations serves two important functions. Firstly, it forms a basis of identification between a match and a mismatch image matching. Secondly, it is used to truncate the image matching process whenever essential conditions for the decision regarding image matching are met.

3.5.2 Template Image Matching on USC-SIPI-HAAI Database

Thirty real satellite images of size 512×512 pixels have been considered as main image one by one. Templates of size 100×100 pixels have been extracted from 16 different locations from each main image. The value of confidence factor is evaluated and recorded after each vector matching iterations. The resulting plots for 480 trends are recorded and fifteen cases are plotted on the same graph for comparison in Fig. 3.21 (a) for matching templates. Similar experiments have also been conducted for non matching templates and the fifteen such plots are consolidated in Fig. 3.21 (b). It is evident that the matched case plots lead to the final value of 100%, whereas the



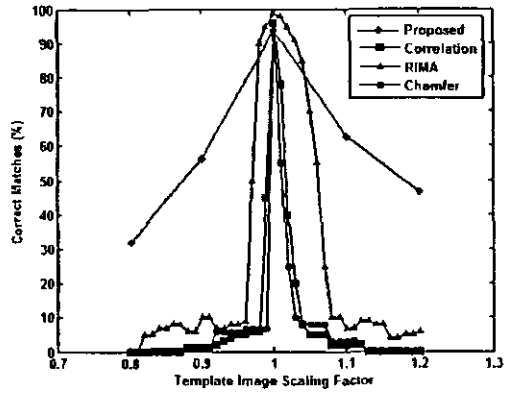
(a)



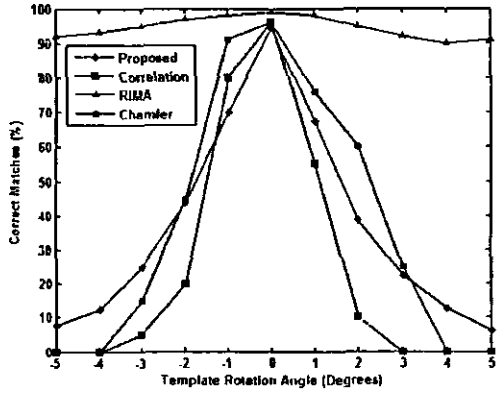
(b)

Fig. 3.21 Confidence factor growth trends of database images for matching and non matching templates (a) Matched template cases (b) Un-matched template cases.

unmatched case settles around less the 70% for the same plots. Therefore, it is recommended that the value of MC_F can be taken as 75%. It is observed that 85% of the matching cases converge to 100% value within first 70 iterations whereas 95% are converged after 200 iterations. There are 4% solutions which could not converge at all giving wrong results. A similar trend is observed in non-matching template cases as 87% of the cases settle to a value less than 70% within first 70 image matching iterations. Similarly, 2% cases wrongly converge to a 100% level giving matching



(a)



(b)

Fig. 3.22 The percentage correct matches for template image scaling and rotation on database images (a) Percentage of correct matches vs. Template scaling factor (b) Percentage of correct matches vs. Template rotation angle.

conditions. The first two hundred image matching iterations must be skipped to avoid the initial random transients of the value of Confidence factor. This quantity is determined experimentally after analyzing the statistical results including noisy data.

3.5.3 Scale and Rotation Invariant Template Image Matching

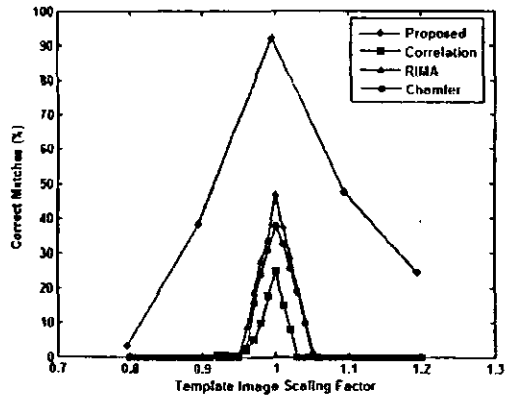
The effect on performance of the difference of scale in the template image with respect to the main image was evaluated on USC-SIPI-HAAI database. Template images were extracted from 16 different locations from the available 30 main images. The scale of each template image was varied from 0.8 to 1.2 times the original with a

step size of 0.1. The scale factor of 1.0 means no scale change in the template image. The percentages of correct matches are thus plotted in Fig. 3.22 (a). Similarly, the results of Correlation [10], Chamfer [14] and RIMA (Robust Image Matching Algorithm) [18] are also plotted on the same graph of Fig. 3.22 (a) with template scale varying from 0.8 to 1.2 with a smaller step of 0.01. The proposed technique has shown greater endurance with varying template scale as compared to the rest of the techniques. All of the other techniques have shown less than 10% results at a scaling factor of 1.1 whereas the proposed technique has shown 63.07%.

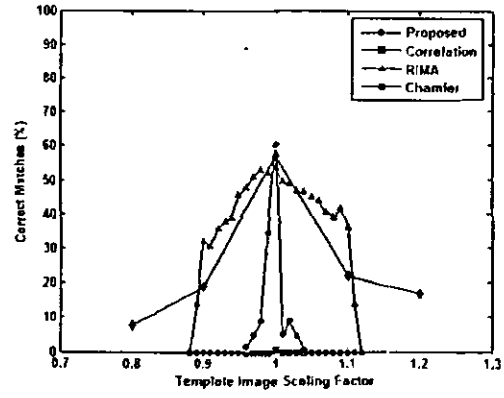
A similar test was conducted to evaluate the performance of the proposed algorithm when the template is rotated. All of the template images were rotated from an angle of -5° to $+5^{\circ}$ prior to image matching. The results are consolidated in Fig. 3.22 (b). The proposed algorithm has shown an effective endurance of $\pm 2^{\circ}$ after which the performance drops rapidly. This is primarily due to the effect of considering $\pm 2^{\circ}$ in the process of subset extraction from the main image HVT. Increasing this range may increase the rotation endurance, but will have negative impact on the speed and performance of the algorithm.

3.5.4 Template Image Matching with Impulsive and AWGN noise

The performance of the proposed algorithm was evaluated on USC-SIPI-HAAI database image matching experiment with 30 dB of impulsive noise with varying template image scale. The impulsive noise occurs due to the malfunction or saturation of some of the sensor elements. This noise introduces black and white dots in the image and it is also known as salt and pepper noise. The resulting plot of percentage of correct matches is shown in Fig. 3.23 (a). The statistical data shows a reduction of correct matches from 53.4% to 38.1% at 0.9 scaling factor for the template image.



(a)



(b)

Fig. 3.23 The percentage correct matches for noisy and scaled template images (a) Percentage of correct matches vs. Template scaling factor with 30 dB impulsive noise (b) Percentage of correct matches vs. Template scaling factor with 30 dB Gaussian noise.

The performance of other techniques has shown significant degradation in the results as compared with the proposed method as they have failed to match for the same scaling factor under the presence of 30dB impulsive noise. Similar experiments were conducted in the presence of 30dB Additive White Gaussian Noise (AWGN) and the results are shown in Fig. 3.23 (b). A drop in performance is observed from 93% to 57.8% at scale 1.0 for the proposed method. The AWGN noise has more effect on the image edges as it tends to change the gray level value randomly, whereas the

impulsive noise is observed to be deteriorating only few edges disturbing only those edges where it effects.

3.5.5 Computational Analysis of the Proposed Algorithm

The computational load of the proposed algorithm has been investigated by calculating the number of operations required to estimate the position of the template in the main image. The speed comparison has also been provided with different image sizes and on noisy images. The main image has size $M \times N$ pixels and the template has size $\bar{M} \times \bar{N}$ pixels. In proposed algorithm 10% of the total pixels will constitute as the edges. Each edge will have length of 8 pixels for the worst case. If the vectorization process counts 100% of the total edge pixels as vectors, we will have the E_T template edges or hyper-vectors given as

$$E_T = \frac{\bar{M}\bar{N}}{80} \tag{3.5.1}$$

whereas the main image will have E_M edges or hyper-vectors

$$E_M = \frac{MN}{80} \tag{3.5.2}$$

Edge pair formation from the template image will form P edge pairs, where

$$P = \frac{\left(\bar{M}\bar{N}\right)^2}{8} \tag{3.5.3}$$

If we assume an equal distribution of the edges in the main image for all angles in the range of θ, a total of $(5MN)/(80\times180)$ vectors will be extracted as a subset from the HVT of the main image. Hence, each vector matching iterations will cost

Table 3.5 Total Number of Operations in Image Matching with respect to the Main Image Size ($\times 10^8$).

Main Image Size / Methods	200×200 pixels	400×400 pixels	600×600 pixels
Correlation Method	31.6	5280	74100
Chamfer Algorithm	3.74	434	3380
RIMA	3.84	427	3710
Proposed Algorithm	2.47	39.5	200

$\left((5MN)/(80\times 180)\right)^2$ subtractions, comparisons and equal multiplications (operations). The total computations C given in terms of image edges as

$$C = P\left(\frac{5\times E_M}{180}\right)^2 \tag{3.5.4}$$

or

$$C = \left(\frac{25}{18}\right)\bar{M}\bar{N}E_T E_M{}^2 \tag{3.5.5}$$

This process of hyper vectorization comprises $\left(\left(\bar{M}^2+\bar{N}^2\right)\times 180\right)$ logical comparisons only if we assume that the main image is vectorized offline. In case of chamfer based correlation approach, a typical of $\left(M-\bar{M}+1\right)\left(N-\bar{N}+1\right)\left(\bar{M}\bar{N}/10\right)$ multiplications, along with $\left(M-\bar{M}+1\right)\left(N-\bar{N}+1\right)$ square root operations and $\left(M-\bar{M}+1\right)\left(N-\bar{N}+1\right)$ divisions are required. This is indeed a heavy computational load as compared to our technique which has little real time multiplications. Table 3.5 shows the comparison of number of operations required to estimate the position of template image in the main image with different image sizes. The computations of compared techniques were observed after implementing these techniques on similar platforms. The table clearly shows that the number of computations of the proposed algorithm is much less than the other schemes i.e. Correlation [10], Chamfer [14] and RIMA [18].

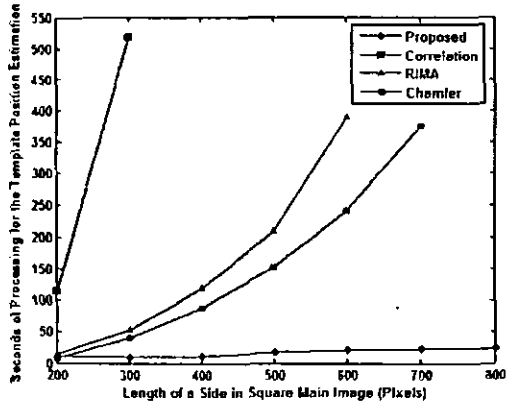


Fig. 3.24 Database experiment results with varying size of the square main image vs. the time required for the matching operation.

The size of the main image in image matching experiments determines the search area and is a vital parameter for evaluating the speed and the variation of speed with respect to main image size. Main image size is varied from 200×200 pixels to 800×800 pixels. Templates were extracted from the 9 different locations from all of the 30 sets of main images. Experiments were conducted on a 2.0 GHz Intel based Centrino Core2 Duo PC using Matlab® 7.1.

The average time taken by the experiments for all configurations is shown in Fig. 3.24. Similar experiments were conducted for Correlation [10], Chamfer [14] and RIMA [18] and their plots are also displayed in Fig. 3.24 for comparison. The speed of the proposed algorithm is much better than the techniques with which it is compared. Furthermore, the proposed algorithm shows a linear proportional dependency on the main image size whereas the rest of the techniques show a non-linear rise with the increase in the search area. The proposed algorithm took 10.15 seconds for the main image size of 300×300 pixels, whereas the Correlation [10], Chamfer [14] and RIMA [18] took 518.74 seconds, 48.7 seconds and 54.57 seconds, respectively. The proposed algorithm took 18.65 seconds for the main image size of 500×500 pixels, whereas the Correlation [10], Chamfer [14] and RIMA [18] took

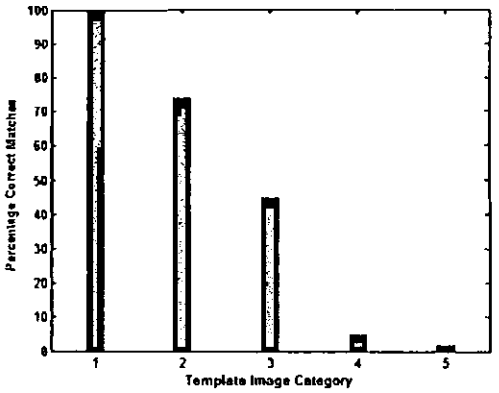


Fig. 3.25 Percentage of Correct Image Matching Experiments
vs. Template Image Category.

2655.1 seconds, 154.2 seconds and 209.85 seconds, respectively. This time can further be decreased if a dedicated hardware with DSP processor may be implemented.

The proposed image edge matching algorithm depends on proper image edge extraction and presence of vital image edges. Images with less number of edges tend to degrade the performance heavily. The template images of different categories are thus investigated for the analysis of the performance in each case. Each category consisted of 104 template images. Category 1 was of the templates with good and prominent edges. This category refers to the satellite images with clear permanent ground features likes roads and constructions etc. Category 2 templates were those with sparsely populated rural areas. Category 3 images were those with mixed vegetation and cultivation lands. Category 4 consisted of the images with sand and desert lands with insignificant and temporary ground features whereas category 5 is composed of mostly water and lake type images. The results are shown in Fig. 3.25.

The effect of noise on algorithm speed is also investigated. The value of confidence factor quickly converges towards 100% in the template image without any noise, as shown in plot of Fig. 3.26. Similar study has been conducted on a template with 30

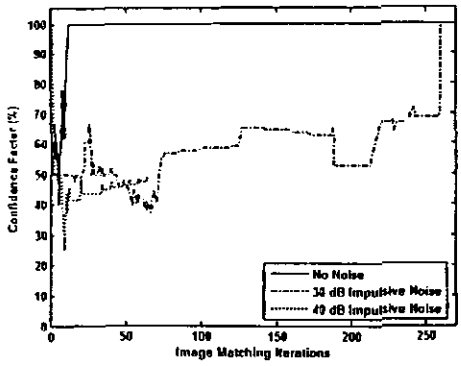


Fig. 3.26 Confidence Factor Trends for Template with no noise and Similar Trends for Templates with 30dB and 40 dB impulsive noise.

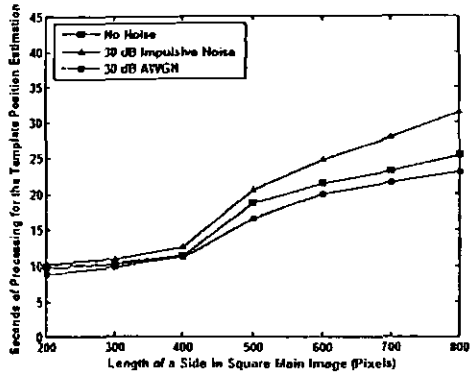


Fig. 3.27 Speed Comparison for Template Location Estimation with no noise and in the presence of 30 dB Impulsive Noise and 30 dB AWGN.

dB impulsive noise in which the value of confidence factor took greater iterations to converge to 100%. If the noise is further increased, two facts are observed. Firstly, fewer solutions qualify and cause lesser number of iterations. Secondly, the final value of the confidence factor is as low as 50% which puts this experiment in the category of unmatched template because of the presence of noise as shown in Fig. 3.26. At this level of noise, the algorithm actually fails to recognize a matching template and may declare an unmatched result. Speed analysis is also performed with varying main image size. A slight increase in processing time is observed with 30 dB impulsive noise because some edges may be broken or distorted in the presence of impulsive noise. Therefore, the total number of edges slightly increases, causing a slight increase in the processing time. However, in case of AWGN, the edges number was slightly decreased. Therefore, the speed of the proposed algorithm slightly increases, as shown in Fig. 3.27. The time required for the matching a template with impulsive noise is 20.52 sec and the one with AWGN is 16.58 sec as compared to the template with no noise which is 18.81 sec for the main image size of 500×500 pixels.

3.6 Conclusions

This scheme presents a new approach towards image edge matching. Once the important properties of the edges are converted into a cluster of hyper-vectors, the matching problem simply reduces to a few subtractions and comparisons and nominal multiplications if scale invariance template matching is desired. Increasing the dimensionality of the image feature representation has found to be helpful in reducing the complexity of the solution procedure. Listing more parameters of an image feature helps in elaborated image description. The algorithm is simpler and much faster for the use in real time systems. As the algorithm proceeds, the grid cells get populated and the pattern of the population distribution tends the solution either towards a match or a mismatch. Due to this methodology, the matching process can be stopped at any time once the direction of the solution is clear which gives a further benefit towards the overall computations. The results clearly demonstrate that the performance of the proposed algorithm is much better than the competitor techniques under different situations.

Chapter 4

Template Matching Through Hyper Vectorization Using Gray Level Sliced Binary Images

In this chapter, a feature based image matching approach suitable for aerial visual navigation is discussed. The image features were extracted through quantization of gray levels in an image to form sub-band binary images. These binary images are then subjected to the boundary extraction of connected patches. The boundaries were vectorized and then normalized for sorting in an order of significance. Hence, the complicated image matching process was reduced to only a few vector subtractions. The main advantage of this approach lies in their low computational overhead, which is primarily due to the small size of feature vectors and early truncation of the algorithm as the position of the vehicle is ascertained. Another important advantage of this algorithm is that it provides a rotation and scale invariant image matching. High level of noise immunity has also been observed which makes the scheme more robust.

4.1 Problem Formulation

Image matching is the process of finding the locations that are similar to the given pattern template from the given main image. The pattern template may have a different scale and rotation. Let $f(x,y)$ be the main image with dimensions $M \times N$. Similarly, let $\bar{f}(\bar{x},\bar{y})$ be the template image having smaller dimensions $\bar{M} \times \bar{N}$ than the main image. We need to find a transformation:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} + m \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} \tag{4.1.1}$$

where Δx and Δy represents the translations of the template image in x and y directions, respectively. m and θ are the scaling factor and the rotation angle of the template image, respectively. The m scaled and θ rotated template image gray levels best match with the gray levels of the main image at the displacement of Δx and Δy. Therefore, the template matching requires the estimation of these four parameters $(\Delta x, \Delta y, m, \theta)$ for any given image set f and $\bar{f}$.

4.2 Image Feature Extraction and Matching in the Proposed Scheme

Image matching is performed in three major steps, i.e. gray level slicing, hyper-vectorization and vector matching. The gray level slicing is performed on both the main image $f(x, y)$ and the template image $\bar{f}(\bar{x}, \bar{y})$ to form sub-band binary image set. Then, these binary images are hyper-vectorized to form shape signatures for both the image sets. Finally, vector matching is performed to obtain the best matching location $(\Delta x, \Delta y)$ and orientation (m, θ) of the template image in the main image. The main steps of the proposed algorithm are implemented in a sequence shown in Fig. 4.1. All these steps are explained as follows.

4.2.1 Gray Level Slicing

Gray level slicing is often used to highlight a specific range of gray levels in an image [41] [42]. The gray level slicing has been utilized to form a binary image by putting the range of interest to maximum intensity while keeping all the other pixels to a zero

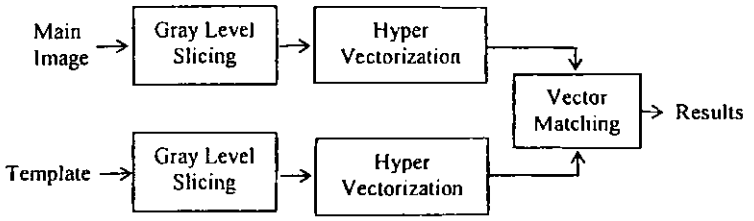


Fig. 4.1 Main steps for the proposed image matching scheme.

level. This facilitates the vectorization of the images in the next step for all of the image features. Gray level slicing process will produce B number of binary images $\{g^1(x,y), g^2(x,y), \ldots, g^B(x,y)\}$ of the same size. Hence, let $g^k(x,y)$ be the k^{th} binary image formed from the main image $f(x,y)$ where $1 \le k \le B$. The process of obtaining binary images from $f(x,y)$ can be represented as:

$$g^k(x,y) = \begin{cases} 1, & \left(\frac{256(k-1)}{B}\right) \le f(x,y) \le \left(\frac{256k}{B} - 1\right) \\ 0, & otherwise \end{cases} \tag{4.2.1}$$

Similarly, the template $\bar{f}(\bar{x},\bar{y})$ will also be converted into B number of binary images $\{\bar{g}^1(\bar{x},\bar{y}), \bar{g}^2(\bar{x},\bar{y}), \ldots, \bar{g}^B(\bar{x},\bar{y})\}$. Hence, let $\bar{g}^k(\bar{x},\bar{y})$ be the k^{th} binary image formed from the template $\bar{f}(\bar{x},\bar{y})$ with $1 \le k \le B$. In the gray level slicing of a satellite image, we have proposed for four band slicing for the use in rotation and scale invariant image matching under the scope of visual aerial navigation as shown in Fig. 4.2.

4.2.2 Hyper Vectorization

Vectorization of image is an effective method to represent the image features in the desired form [43] [44]. A scale and rotation invariant hyper-vectorization based methodology is proposed for shape matching. The goal of this work is not to

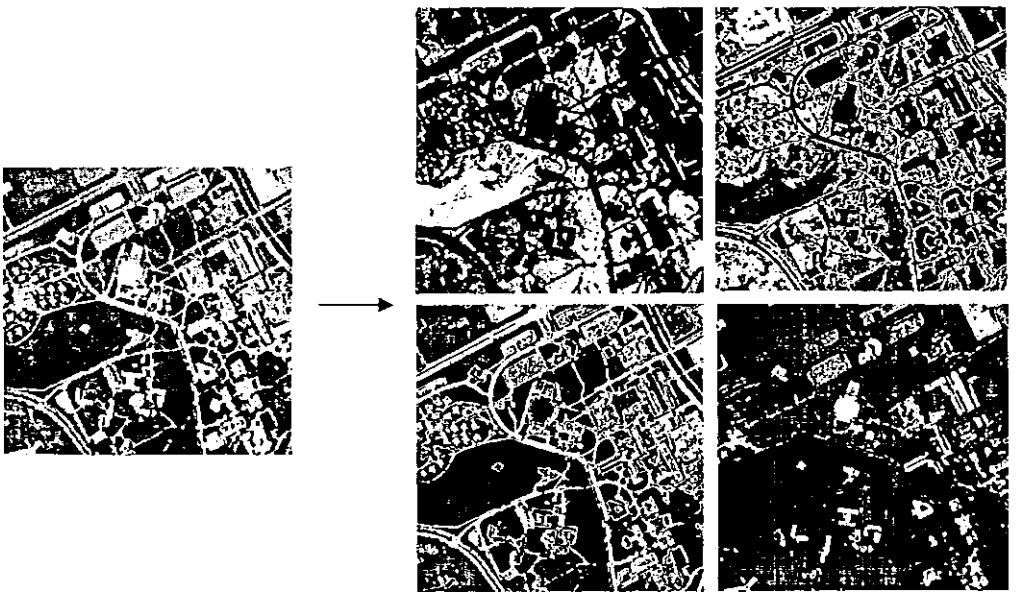


Fig. 4.2 Real Gray Scale Satellite Image and its Sliced Bands for B = 4.

reconstruct the matching images but to estimate the best matching location $(\Delta x, \Delta y)$ and orientation (m, θ) of the template in the main image. The process of vectorization includes the following four steps:

Boundary Extraction of Shapes: Each binary image formed in the previous step, $g^k(x,y)$ or $\bar{g}^k(x,y)$, is then subjected to a boundary extraction process [37]. In this process, the image may be subtracted from its eroded image to extract all the boundaries of connected patches of the binary images. Each binary image may produce more than one boundary. Each boundary may be represented by the co-ordinates of its perimeter. The boundary co-ordinates of the q^{th} boundary and of the k^{th} binary image, $g^k(x,y)$, can be represented as $S^n(k,q)$ where $1 \le n \le L^{k,q}$ and $L^{k,q}$ is the number of points in the perimeter. The $S^n(k,q)$ may also be represented in complex form as:

$$S^n(k,q) = \left(x^n(k,q) + jy^n(k,q)\right) \qquad 1 \le n \le L^{k,q} \qquad (4.2.2)$$

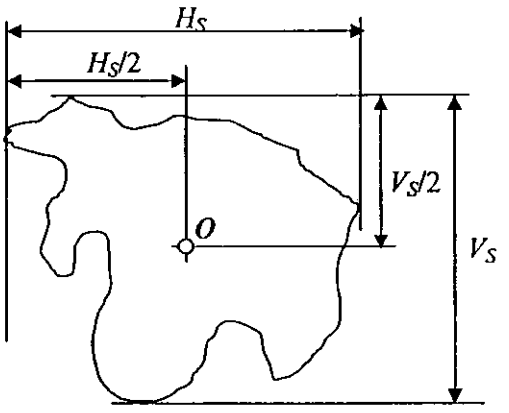


Fig. 4.3. Centroid of a given boundary.

Similarly, the boundaries of the binary image $\bar{g}^k(\bar{x},\bar{y})$ may be extracted and can be represented as:

$$\bar{S}^n(k,\bar{q}) = \left(\bar{x}^n(k,\bar{q}) + j\bar{y}^n(k,\bar{q})\right) \qquad 1 \le n \le \bar{L}^{k,\bar{q}} \tag{4.2.3}$$

Centroid of the Shape: The first step is to find out the centroid of the shape. The centroid of a shape is defined as the midpoint of the extremes of the boundary in both the horizontal and vertical directions. The procedure of obtaining the centroid is explained in the Fig. 4.3. In this figure, the centroid is labeled as "***O***" and is located at

$$c(k,q) = \left(c_x(k,q) + jc_y(k,q)\right) \tag{4.2.4}$$

in the main image corresponding to k^{th} binary image and q^{th} boundary, whereas the first co-ordinate of the centroid $c_x(k,q)$ is defined as

$$c_x(k,q) = mean\left\{\max\left(\forall x_n(k,q) \mid 1 \le n \le L^{k,q}\right), \min\left(\forall x_n(k,q) \mid 1 \le n \le L^{k,q}\right)\right\} \tag{4.2.5}$$

In this equation, the $mean(...,...)$ finds the mean value of the two entries in the parenthesis, $\max\{...\}$ gives the maximum value of the quantities in the respective argument sets and $\min\{...\}$ gives the minimum values of the quantities in the respective argument sets. The other co-ordinate $c_y(k,q)$ of the centroid is given as

$$c_y(k,q) = mean\left\{\max\left(\forall y_n(k,q) \mid 1 \le n \le L^{k,q}\right), \min\left(\forall y_n(k,q) \mid 1 \le n \le L^{k,q}\right)\right\} \quad (4.2.6)$$

Similarly, the centroid of the template will be

$$\bar{c}(k,\bar{q}) = \left(\bar{c}_x(k,\bar{q}) + j\bar{c}_y(k,\bar{q})\right) \quad (4.2.7)$$

Magnitude Signatures: Then, a row matrix $G(k,q)$ is formulated which is composed of Euclidean distances $d_n(k,q)$ of all the boundary point in $S^n(k,q)$ with the centroid point $O\left(c_x(k,q) + jc_y(k,q)\right)$. This vector is represented as

$$G(k,q) = \left[d_1(k,q) \quad d_2(k,q) \quad d_3(k,q) \quad \ldots \quad d_{L^{k,q}}(k,q)\right] \quad (4.2.8)$$

where

$$d_n(k,q) = \sqrt{\left(c_x(k,q) - x_n(k,q)\right)^2 + \left(c_y(k,q) - y_n(k,q)\right)^2} \qquad 1 \le n \le L^{k,q} \quad (4.2.9)$$

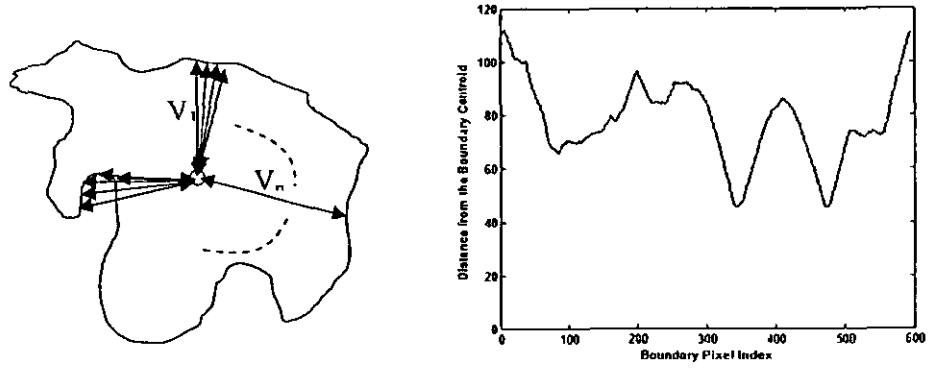
which is the n^{th} distance of the q^{th} boundary in the k^{th} binary image formed from the main image. Similarly, for the template, the list will be represented as

$$\bar{G}(k,\bar{q}) = \left[\bar{d}_1(k,\bar{q}) \quad \bar{d}_2(k,\bar{q}) \quad \bar{d}_3(k,\bar{q}) \quad \ldots \quad \bar{d}_{\bar{L}^{k,\bar{q}}}(k,\bar{q})\right] \quad (4.2.10)$$

where

$$\bar{d}_n(k,\bar{q}) = \sqrt{\left(\bar{c}_x(k,\bar{q}) - \bar{x}_n(k,\bar{q})\right)^2 + \left(\bar{c}_y(k,\bar{q}) - \bar{y}_n(k,\bar{q})\right)^2} \qquad 1 \le n \le \bar{L}^{k,\bar{q}} \quad (4.2.11)$$

The pictorial representation of these distances is shown in Fig. 4.4(a) and a corresponding plot is shown in Fig. 4.4(b). If the particular boundary point in the binary image is farther away from the centroid, the distance in the plot will reveal the same. Hence, the plot is another way of representing the boundary in the binary image and will remain a unique one for one particular shape. This plot is then normalized in



(a) (b)

Fig. 4.4 (a) Distances of boundary points from centroid (b) Corresponding plot.

both the horizontal and vertical axes. The horizontal normalization process was applied on vector $G(k,q)$ to form a fixed length representation having "$\Im$" elements as

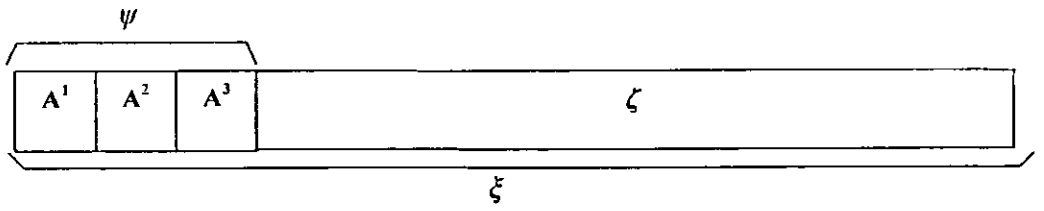
$$H(k,q)=\left[h_1(k,q) \quad h_2(k,q) \quad h_3(k,q) \quad \ldots \quad h_{\Im}(k,q)\right] \tag{4.2.12}$$

where $$h_n(k,q)=mean\left(d_{floor\left(\frac{nl_q^k}{\Im}\right)}(k,q), d_{ceil\left(\frac{nl_q^k}{\Im}\right)}(k,q)\right) \tag{4.2.13}$$

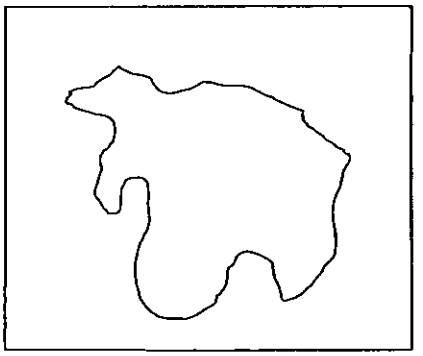
Similarly, the horizontal normalization process for template image will yield $\bar{H}(k,\bar{q})$ with fixed length representation.

$$\bar{H}(k,\bar{q})=\left[\bar{h}_1(k,\bar{q}) \quad \bar{h}_2(k,\bar{q}) \quad \bar{h}_3(k,\bar{q}) \quad \ldots \quad \bar{h}_{\Im}(k,\bar{q})\right] \tag{4.2.14}$$

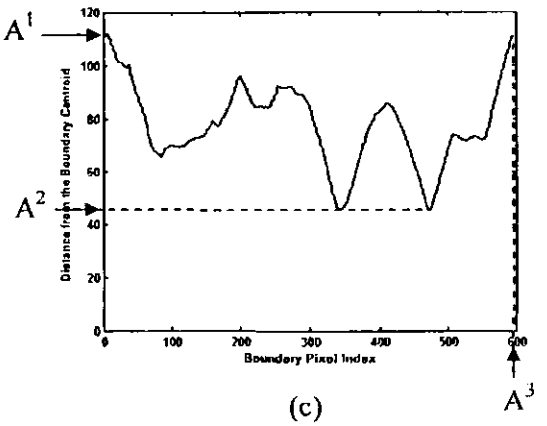
The values thus formulated are expressed in the form of a row matrix, which will be used to formulate a unique signature of this particular boundary in the next stage. The vertical normalization is achieved by dividing each magnitude with the maximum distance in the set as shown here



(a)



(b)



(c)

Fig. 4.5 The components of the vector header (a) Signature (b) Shape (c) Header Components.

$$\zeta(k,q)=\frac{H(k,q)}{\max\left(H(k,q)\right)}. \tag{4.2.15}$$

Similarly, the template image is also normalized, as shown below

$$\bar{\zeta}(k,\bar{q})=\frac{\bar{H}(k,\bar{q})}{\max\left(\bar{H}(k,\bar{q})\right)} \tag{4.2.16}$$

A three valued header $\psi(k,q)$ (consisting of quantities $A^1(k,q)$, $A^2(k,q)$ and $A^3(k,q)$) is proposed to generate and place before the row matrix $\zeta(k,q)$ to form a complete "$\Im$+3" valued signature as given below

$$\xi(k,q)=[\psi(k,q) \quad \zeta(k,q)] \tag{4.2.17}$$

The shape of this particular boundary is also shown in Fig. 4.5(b). $A^1(k,q)$ and $A^2(k,q)$ are the maximum and the minimum distances from the centroid of the boundary prior to normalization i.e. $\max(G(k,q))$ and $\min(G(k,q))$, respectively. These distances are also represented graphically in Fig. 4.5(c). $A^3(k,q)$ is the perimeter length of the boundary before normalizing the signature to length $\Im$. Similarly, we have a complete normalized shape signature for the template image shown in Fig. 4.5(a) as

$$\bar{\xi}(k,\bar{q}) = [\bar{\psi}(k,\bar{q}) \quad \bar{\zeta}(k,\bar{q})] \tag{4.2.18}$$

Phase Signatures: A row matrix is then formed which is composed of the angles of vectors d_n. This row matrix is represented as

$$J(k,q) = \left[\measuredangle d_1(k,q) \quad \measuredangle d_2(k,q) \quad \measuredangle d_3(k,q) \quad \ldots \quad \measuredangle d_{L^{k,q}}(k,q) \right] \tag{4.2.19}$$

where

$$\measuredangle d_n(k,q) = \tan^{-1} \frac{\left(c_y(k,q) - y_n(k,q)\right)}{\left(c_x(k,q) - x_n(k,q)\right)} \qquad 1 \le n \le L^{k,q} \tag{4.2.20}$$

The $\measuredangle d_n(k,q)$ is the phase of the n^{th} angle with centroid of the q^{th} boundary in the k^{th} binary image formed from the main image. This representation is also expanded or shrunk (re-scaled) to form a fixed length representation having "$\Im$" elements in horizontal direction only as

$$\measuredangle \zeta(k,q) = \left[D_1(k,q) \quad D_2(k,q) \quad D_3(k,q) \quad \ldots \quad D_{\Im}(k,q) \right] \tag{4.2.21}$$

where $$D_n(k,q) = mean\left\{ \measuredangle d_{floor\left(\frac{n L^{k,q}}{\Im}\right)}(k,q), \measuredangle d_{ceil\left(\frac{n L^{k,q}}{\Im}\right)}(k,q) \right\} \tag{4.2.22}$$

Similarly, for the template image we have

$$\measuredangle \bar{\zeta}(k,\bar{q}) = \left[\bar{D}_1(k,\bar{q}) \quad \bar{D}_2(k,\bar{q}) \quad \bar{D}_3(k,\bar{q}) \quad \ldots \quad \bar{D}_{\Im}(k,\bar{q}) \right] \tag{4.2.23}$$

This completes the process of vectorization in which the important features of both the main image and the template are represented by a normalized magnitude signature along with a header component and a normalized phase signature. These signatures constitute a $\Im$-D hyper-space which will be processed in the next stage for solution estimation. In this section our contribution is the utilization of an associated phase signature along with the normalized magnitude signature. The combined arrangement gives precise rotation information of the template feature shape with respect to that of the main image.

4.3 Vector Matching

Vector matching is the most important step after the vectorization process. In the vectorization process, both the main image and the template are in the form of a vector sets. Each binary image may produce many vectors after the boundary extraction process and each boundary produces one vector. The vectors in the template set are then sorted with respect to perimeter length in descending order and considered one by one for matching. We consider only those vectors with $\overline{A}^1(k,\overline{q}) - \overline{A}^2(k,\overline{q}) \geq \Upsilon$ and $\overline{A}^3(k,\overline{q}) \geq \wp$. The reason for this selection is to ignore the shapes which are close to a circle and those which are insignificantly small. Such shapes may produce false results as the signatures for the circular or close to circular shapes tend to become quite indistinctively flat. The qualified vector of the template image is then matched with all of the vectors of the main image belonging to the same binary band.

4.3.1. Rotation Estimation

First step of vector matching is to find out the template rotation angle for which the magnitude signatures are correlated, as follows, to form an error function $\varepsilon(S)$. We

have proposed the following 1-D SAD (sum of absolute difference) for estimation of the template matching location as:

$$\varepsilon(\mathbb{S}) = sum\left(abs\left(\zeta(k,q) - \bar{\zeta}^{(\mathbb{S})}(k,\bar{q})\right)\right) \qquad 1 \le \mathbb{S} \le \Im \tag{4.3.1}$$

where $\bar{\zeta}^{(\mathbb{S})}(k,\bar{q})$ is defined as a shifted and rotated version of the template signature $\bar{\zeta}(k,\bar{q})$ with shifts $\mathbb{S}$ varying from 1 to $\Im$ given as:

$$\begin{aligned}\bar{\zeta}^{(\mathbb{S})}(k,\bar{q}) = [&\bar{H}_{\Im-\mathbb{S}+1}(k,\bar{q}) \quad \bar{H}_{\Im-\mathbb{S}+2}(k,\bar{q}) \quad \cdots \quad \bar{H}_{\Im}(k,\bar{q}) \\ &\bar{H}_{1}(k,\bar{q}) \quad \bar{H}_{2}(k,\bar{q}) \quad \cdots \quad \bar{H}_{\Im-\mathbb{S}}(k,\bar{q})]\end{aligned} \tag{4.3.2}$$

The minima of this correlation error function $\varepsilon(\mathbb{S})$ is located at $\mathbb{S} = \mathbb{S}_{min}$ and is given as

$$\varepsilon_{min} = \varepsilon(\mathbb{S}_{min}) \tag{4.3.3}$$

If this $\varepsilon_{min} \le \gamma$, then this vector pair will be considered as matched, where γ is a threshold value for vector matching.

$$\theta = mean\left(\left(\measuredangle\zeta(k,q)\right) - \left(\measuredangle^{(\mathbb{S}_{min})}\bar{\zeta}(k,\bar{q})\right)\right) \tag{4.3.4}$$

4.3.2. Scale Estimation

The second step is to estimate the scale of the template with respect to the main image. The perimeter of the boundary extracted in the template can be compared with that of the main image prior to normalization process for this estimation. Hence, the proposed scale estimate for this particular vector pair is simply a division process and is given as

$$m = \frac{A^{3}(k,q)}{\bar{A}^{3}(k,\bar{q})} \tag{4.3.5}$$

4.3.3. Location Estimation

The third and last step of vector matching is to find out the matching co-ordinates of the template in the main image. The centroid of the template shape has to be rotated

and scaled back in order to match the orientation and the scale of the shape in the main image before finding the matching co-ordinates. The co-ordinates of the centroid of this template shape is given as $\bar{O}\left(\bar{c}_x(k,\bar{q})+j\bar{c}_y(k,\bar{q})\right)$. The center of the template image is given as $\bar{P}\left(\frac{\bar{M}}{2}+j\frac{\bar{N}}{2}\right)$. We can define a vector $\bar{Q}$ extending from the template image center towards the respective shape centroid, as given below

$$\bar{Q}=\bar{O}-\bar{P} \tag{4.3.6}$$

This vector can also be represented in polar form as

$$\bar{Q}=\left|\bar{Q}\right|e^{j\angle(\bar{Q})} \tag{4.3.7}$$

This vector is then subjected to a rotation to an angle "$-\theta$" given in equation (4.3.4) to form another vector $\bar{R}$ as

$$\bar{R}=\left|\bar{Q}\right|<e^{j(\angle(\bar{Q})-\theta)} \tag{4.3.8}$$

Another vector is then defined as $\bar{T}$, which is extending from the template image upper left corner to the new rotated position of $\bar{Q}$ given as

$$\bar{T}=\bar{P}+\bar{R}=\left(\bar{T}_x+j\bar{T}_y\right) \tag{4.3.9}$$

This vector is then finally scaled to form a vector U as

$$U=m\bar{T}=\left(m\bar{T}_x+jm\bar{T}_y\right) \tag{4.3.10}$$

The co-ordinates of the respective shape in the main image is given as $O\left(c_x(k,q)+jc_y(k,q)\right)$. The final displacement co-ordinates of the template image in the main image are represented by a vector V as given below

$$V=O-U=\left(\Delta x+j\Delta y\right) \tag{4.3.11}$$

4.3.4. Confidence Factor Estimation

The solution set $(\Delta x, \Delta y, m, \theta)$ is an estimated solution which is obtained from comparing one template vector from the set of main image vectors. Many such solution set will be created when the comparison process goes on and ultimately a solution space will be formed with the total number of $\Re$ solutions. Let this solution space be represented by the set Z as

$$Z = \bigcup_{r=1}^{\Re} Z_r = \bigcup_{r=1}^{\Re} \{(\Delta x_r, \Delta y_r, m_r, \theta_r)\} \tag{4.3.12}$$

Then a fifth index, s_r will be included in the ordered pairs of this set to make the new representation as

$$Z_r^s = \bigcup_{r=1}^{\Re} \{(\Delta x_r, \Delta y_r, m_r, \theta_r, s_r)\} \tag{4.3.13}$$

such that

$$s_r = n\left(\{\forall Z_r \in Z \,|\, |Z_r - Z_t| \le \eta, 1 \le r, t \le \Re, r \ne t\}\right) \tag{4.3.14}$$

where $|Z_r - Z_t|$ gives the Euclidean distance between the vector Z_r and Z_t. The argument $n(\cdots)$ gives the cardinal number of the set, where η is a small positive quantity showing the limits of the cluster neighborhood. Then the final predicted solution of the template matching location and orientation is given as Z_r in which s_r has the highest value. It is not necessary to match all the template features with the main image to ascertain the best matching position of the template. The earlier truncation of the matching process can be achieved using the following proposed confidence factor Θ, which is defined as follows:

$$\Theta = \frac{S_{Max} - \overline{S}_R}{S_{Max}} \times 100 \tag{4.3.15}$$

where s_{Max} is the maximum neighborhood count and $\overline{s}_R$ is the mean neighborhood count of the rest of the solutions with count more than $0.25s_{Max}$. The value of Θ can be updated with each new entry in the set Z_r and it also serves as a distinguishing factor for matching and non-matching template. If $\Theta > \Theta_{threshold}$ the truncation process will be truncated declaring a match.

4.3.5. Pseudo Code of the Proposed Algorithm

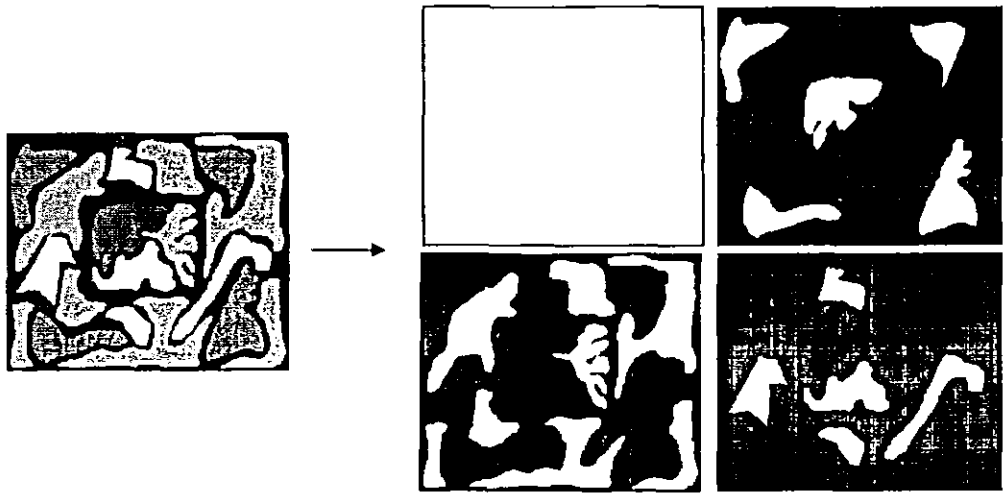
The procedure for the estimation of template image matching location can also be summarized as given in Table 4.1 in the form of tabulated pseudo-code:

Table 4.1 The Pseudo code of the Proposed Scheme.

Steps	Process	Input	Output
Problem Statement			
	To find template image $(\overline{f})$ best matching location $(\Delta x, \Delta y)$, and orientation (m, θ), in the main image (f).	$f(x,y)$ size: $M \times N$ $\overline{f}(\overline{x},\overline{y})$ size: $\overline{M} \times \overline{N}$	$(\Delta x, \Delta y, m, \theta)$.
Proposed Algorithm			
		Stage 1: Gray Level Slicing	
1	Main image and the template image are sliced into B binary images of the same size.	$f(x,y)$, $\overline{f}(\overline{x},\overline{y})$	$\{g^1(x,y), g^2(x,y), \ldots, g^B(x,y)\}$ $\{\overline{g}^1(\overline{x},\overline{y}), \overline{g}^2(\overline{x},\overline{y}), \ldots, \overline{g}^B(\overline{x},\overline{y})\}$
		Stage 2: Hyper Vectorization	
2	All the binary images of step 1 are to be subtracted from their own eroded copy to extract the perimeter co-ordinates boundary of this q^{th} connected patch.	$g^k(x,y)$, $\overline{g}^k(\overline{x},\overline{y})$ $1 \le k \le B$	$S^n(k,q) = x^n(k,q) + iy^n(k,q)$ $\overline{S}^n(k,\overline{q}) = \overline{x}^n(k,\overline{q}) + i\overline{y}^n(k,\overline{q})$ $1 \le n \le L^{k,q}$ $L^{k,q}$: No. of perimeter points in k^{th} binary image and q^{th} boundary.
3	Centroid point "O" is found for each boundary of step 2.	$S^n(k,q)$, $\overline{S}^n(k,\overline{q})$	$c(k,q) = (c_x(k,q) + jc_y(k,q))$ $\overline{c}(k,\overline{q}) = (\overline{c}_x(k,\overline{q}) + j\overline{c}_y(k,\overline{q}))$
4	Row matrix is formed which consist of the distances from all the perimeter points to the centroid.	$S^n(k,q)$, $\overline{S}^n(k,\overline{q})$ $c(k,q)$, $\overline{c}(k,\overline{q})$	$G(k,q)$, $\overline{G}(k,\overline{q})$
5	All of the row matrices obtained in step 4 are re-scaled to a fixed length "$\mathfrak{I}$"	$G(k,q)$, $\overline{G}(k,\overline{q})$	$H(k,q)$, $\overline{H}(k,\overline{q})$
6	The matrices obtained in step 5 are normalized by dividing each element by the maximum of the array.	$H(k,q)$, $\overline{H}(k,\overline{q})$	$\zeta(k,q)$, $\overline{\zeta}(k,\overline{q})$

7	The maximum and the minimum distance (A^1 and A^2), and the length (A^3) of the boundary perimeter are placed as header to form full magnitude signature	$\zeta(k,q), \bar{\zeta}(k,\bar{q})$ $\psi(k,q) = A^1, A^2, A^3$ $\bar{\psi}(k,\bar{q}) = \bar{A}^1, \bar{A}^2, \bar{A}^3$	$\xi(k,q) = [\psi(k,q) \quad \lvert\zeta(k,q)\rvert]$ $\bar{\xi}(k,\bar{q}) = [\bar{\psi}(k,\bar{q}) \quad \lvert\bar{\zeta}(k,\bar{q})\rvert]$
8	Phase signature is also formed which consist of the angles of all of the boundary point from the centroid with respect to positive x axis.	$S''(k,q)$, $\bar{S}''(k,\bar{q})$ $c(k,q)$, $\bar{c}(k,\bar{q})$	$J(k,q), \bar{J}(k,\bar{q})$
9	All of the row matrices obtained in step 8 are re-scaled to a fixed length "$\Im$" as a final phase signature	$J(k,q), \bar{J}(k,\bar{q})$	$\measuredangle\zeta(k,q)$, $\measuredangle\bar{\zeta}(k,\bar{q})$
		Stage 3: Vector Matching	
10	Magnitude signatures are sorted with the lengthiest perimeter placed first. The magnitude signatures of the template from this list will then be correlated in 1-D form by all of the magnitude signatures of the main image of the same binary band.	$\zeta(k,q), \bar{\zeta}(k,\bar{q})$, for $\bar{A}^1(k,\bar{q}) - \bar{A}^2(k,\bar{q}) \geq \bar{A}^3(k,\bar{q}) \geq \wp$ as qualifying conditions	$\varepsilon(\mathbb{S})$: correlation error function with minima $\varepsilon_{\min} = \varepsilon(\mathbb{S}_{\min})$ at shift $\mathbb{S} = \mathbb{S}_{\min}$
11	Template image rotation angle "θ" estimation.	$\zeta(k,q), \bar{\zeta}(k,\bar{q})$ for $\varepsilon_{\min} \leq \gamma$,	$\theta = mean\left(\measuredangle\zeta(k,q) - \measuredangle^{(\mathbb{S}_{\min})}\bar{\zeta}(k,\bar{q})\right)$ $(\cdots)^{(\mathbb{S}_{\min})}$: Shifted angle signature
12	Template image scale "m" estimation.	$\zeta(k,q), \bar{\zeta}(k,\bar{q})$ for $\varepsilon_{\min} \leq \gamma$,	$m = \dfrac{A^3(k,q)}{\bar{A}^3(k,\bar{q})}$
13	The vector Q is defined as the one extending from template image center towards the shape's centroid. Whereas R is a new vector with rotated orientation and same length as Q.	$\bar{c}(k,\bar{q})$ $\bar{P}\left(\dfrac{\bar{M}}{2} + j\dfrac{\bar{N}}{2}\right)$ $\bar{Q} = \bar{c} - \bar{P} = \lvert\bar{Q}\rvert e^{j\measuredangle(Q)}$	$\bar{R} = \lvert\bar{Q}\rvert < e^{j(\measuredangle(Q)-\theta)}$ $\bar{R}$ has got the same magnitude as $\bar{Q}$ and with rotated orientation as estimated in step 11
14	Vector from upper left corner of the template image to R	$\bar{P}, \bar{R}$	$\bar{T} = \bar{P} + \bar{R} = (\bar{T}_x + j\bar{T}_y)$
15	The vector of step 14 is then scaled to match the scale of the main image	$\bar{T}$	$U = m\bar{T} = (m\bar{T}_x + jm\bar{T}_y)$
16	The reference is then changed to the main image upper left corner	c, U	$V = c - U = (\Delta x + j\Delta y)$
17	4-D hyper space cluster formulation for the qualifying solutions and its neighborhood count "s" is calculated at the same time.	$(\Delta x, \Delta y, m, \theta)$	$(\Delta x, \Delta y, m, \theta, s)$
18	Confidence factor for the match is found and the trend of this factor leads to early truncation of the iterative vector matching process.	s_{Max}: Max neighbor count $\bar{s}_R$: Mean neighbor count with count > $0.25\, s_{Max}$	$\Theta = \dfrac{s_{Max} - \bar{s}_R}{s_{Max}} \times 100$
19	If $\Theta > \Theta_{threshold}$ then the matching process is terminated and the solution in the solution space with maximum neighbor count will be declared are matching parameter otherwise pick the next vector for matching and go to step 10.		

20	If all of the vector pairs are processed and still the $\Theta < \Theta_{threshold}$ then the match is declared as a non-matching case.



(a) (b)

Fig. 4.6 Test image and its four binary bands.

4.3.6 Test Case

A test image is considered as shown in Fig. 4.6 (a). It is apparent that the image contains eighteen connected patches of different gray scale values. Gray level slicing with $B=4$ form four binary images b_{I1}, b_{I2}, b_{I3} and b_{I4} as shown in Fig. 4.6 (b). The boundary algorithm then extracts the outer boundary of the connected white pixels objects contained in the binary images. The connected components extracted for each binary band are elaborated separately in Fig. 4.7 for clarity. Similarly, Fig. 4.8 (a) shows the template image, Fig. 4.8 (b) shows gray level sliced binary bands and in Fig. 4.8 (c), the four shapes are elaborated. Fig. 4.9 shows the bar plots of the normalized vector signatures "ζ" for the respective shapes shown in Fig. 4.7. Similarly, Fig. 4.10 shows the same bar plots for the respective template image shapes. It is evident that ζ_{T21} is matching with ζ_{I22}, ζ_{T31} with ζ_{I34} and ζ_{T41} with ζ_{I44}.

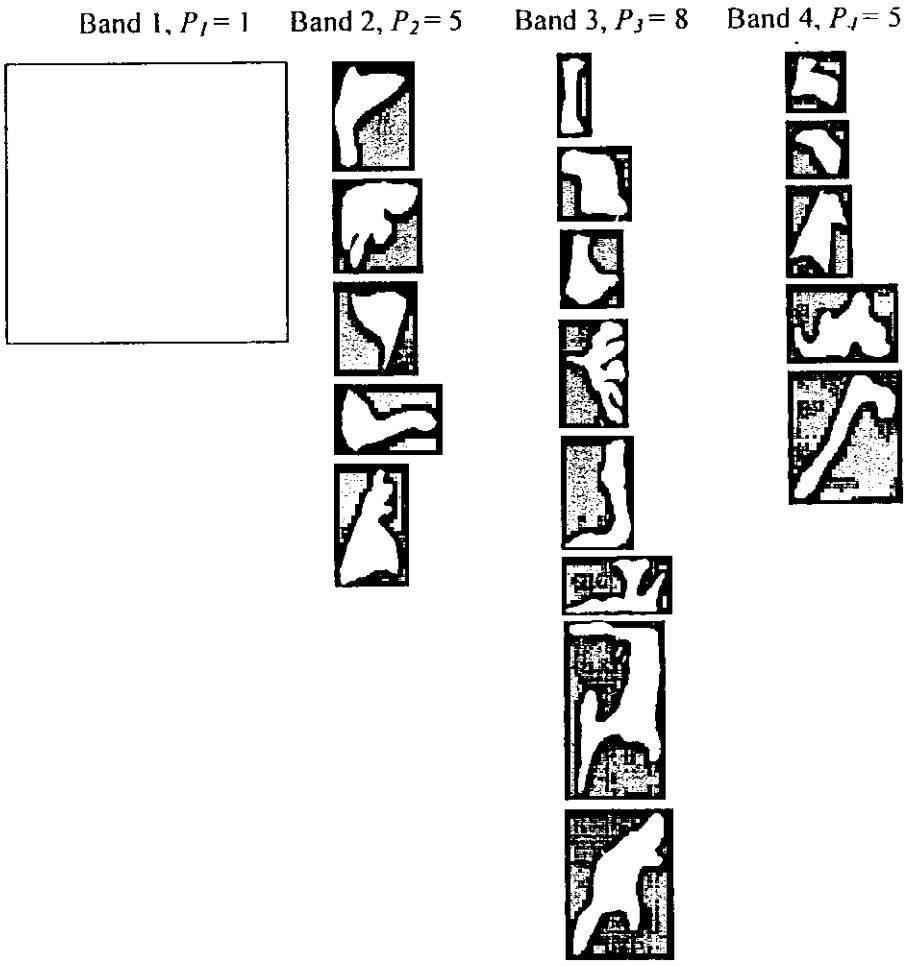


Fig. 4.7 Boundaries extracted from the test image bands.

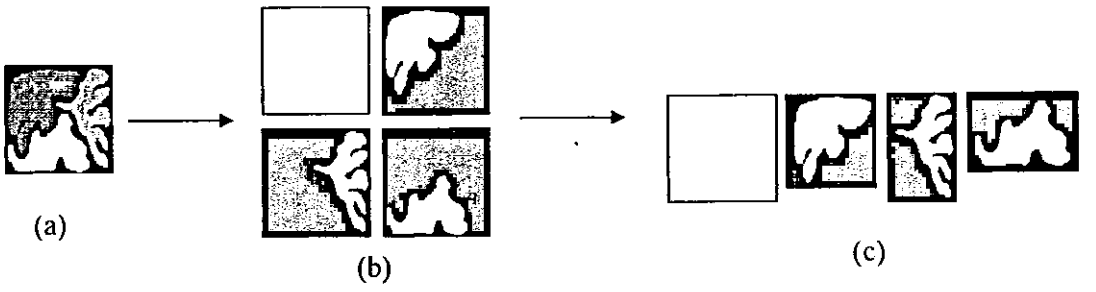


Fig. 4.8 (a) Template Image (b) Binary Images Bands (c) Shapes.

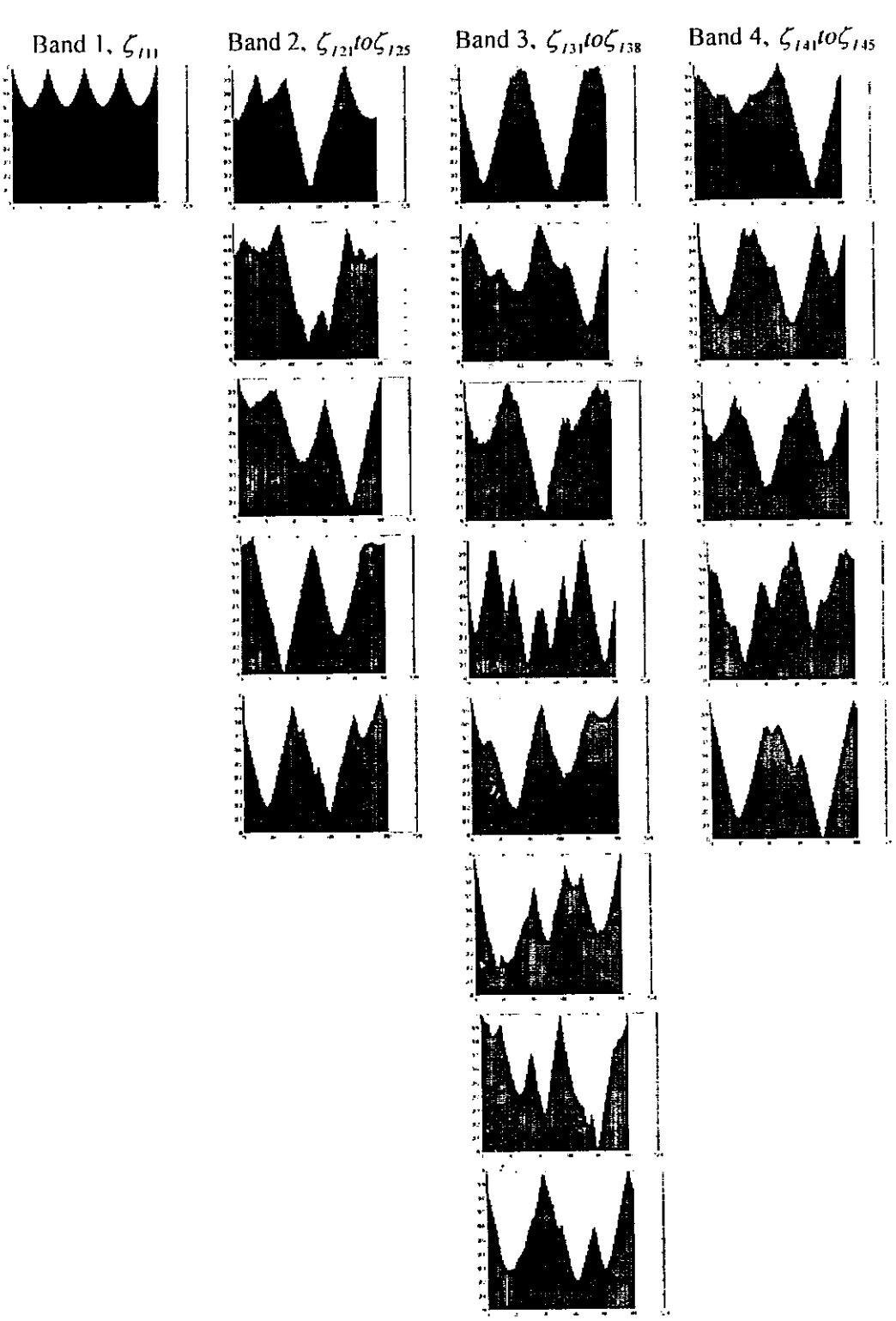


Fig. 4.9 Signatures for the shapes in the main image.

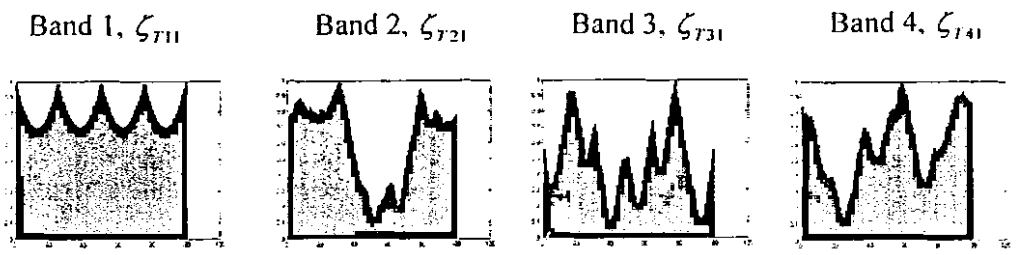


Fig. 4.10 Signatures of the shapes in the template image.

4.4 Results and Discussion

The experiments are designed to test our following claims:

1. Image matching experiments on the bulk of satellite imagery database for the verification of robustness and versatility of the algorithm.
    a. Intelligent detection and segregation of the non matching template images.
    b. Early truncation of the matching process after establishing enough matching operations.
2. Scale invariance image matching and comparison of results.
3. Rotation invariant image matching and full range verification.
4. Computations comparison of the proposed algorithm.
5. Effect of computation on image size.
6. Noise effect and the level of endurance in both AWGN and impulsive noise types.

4.4.1 Database Image Matching Experiments

The proposed algorithm was implemented and was evaluated on USC-SIPI-HAAI database. This database of images contains a variety of aerial images having multiple ground features. The images used as main image have the size of 512×512 pixels. The template images for matching had the size of 100×100 pixels. The fixed length normalization, $\Im$, for boundary signatures was selected to be 100. Multiple experiments were conducted to test the performance of the matching algorithm. Thirty two real satellite images were considered for this purpose. Templates were extracted from 64 different locations from each image. The gray level slicing was performed with four bands i.e. $B=4$. Other parameters are kept as $\Upsilon=10$, $\varepsilon=10$ and $\eta=10$. The proposed algorithm was evaluated using the right prediction percentage factor P defined as

$$\mathrm{P}=\frac{Correct\ Matching\ Experiments}{Total\ Experiments}\times 100 \qquad (4.4.1)$$

This right prediction percentage factor has been utilized to evaluate the robustness and versatility of the proposed algorithm as compared with the ones reported earlier in literature.

4.4.2 Computation Load of the Algorithm

The proposed algorithm is based on extraction of shape contour in the different gray level sliced binary images, as features and simpler computations as integer comparison and additions/subtractions for vector matching. A relative computational analysis of the algorithm with that of Chamfer image matching algorithm [14], Robust Image Matching Algorithm (RIMA) [18], PCA-base Rotation Invariant Texture Features (PCA RITF) [159], Fourier Descriptor Image Matching (FDIM) [157] and

Table 4.2 Comparison of Computations for Different Image Matching Schemes for a 100×100 pixel template in × 10^4 computations.

Algorithm / Computations	Chamfer		RIMA		PCA RITF		3D-FR		Hyper Vector Matching	
	Pre	Online	Pre	Online	Pre	Online	Pre	Online	Pre	Online
Additions / Subtractions	300	16012	300	176012	142440	81773	144	32	16	24
Multiplications/ Divisions	300	12	300	12	154980	78330	96	45	32	32
Square Roots	--	16	--	--	--	--	--	--	8	8
Logical Comparisons	22500	160016	22500	160900	121	449	877	22	375	15

Three-dimensional Face Recognition (3D-FR) [158] has been presented in Table 4.2. The 3D-FR [158] is a face recognition scheme which converts the facial gray level features in the form of contours and the distances between two contour lines in a convenient form for the purpose of face matching. The table shows that only the proposed algorithm utilizes much less computations as compared with the other algorithms.

The computations for the image matching experiments increase with increase in the image size. We have also compared the processing time of the proposed algorithm with the Chamfer [14], RIMA [18], PCA RITF [159], Fourier Descriptor Image Matching (FDIM) [157] and 3D-FR [158]. Results of this comparison are provided in Fig. 4.11. Size of the template is taken fixed as 100×100 pixels, whereas the size of the main image is varied from 200 to 800 pixels side for square images in 100 pixels step.

Platform used for the experiments was Matlab® 7.3.on a PIV based 3.0 Ghz PC. It is evident that other techniques take more time for the same image matching a faster rate

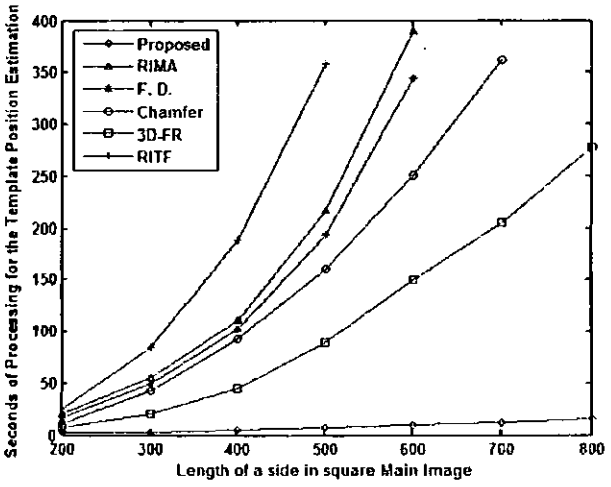


Fig. 4.11 Comparison of the processing time of the proposed algorithm with other techniques.

experiments for any given image size. It is clear from Fig. 4.11 that as the size of the main image is increased, the time required to perform the template search increases at showing a non-linear rising trend. The results of proposed scheme, however, show a little dependency on the size of the main image.

The proposed algorithm can be intelligently terminated earlier without performing the full matching iterations. The matching confidence factor given in equation (4.3.15) serves as discrimination between the matched and the unmatched image cases. To verify this property, ten examples of both the matched and unmatched cases have been considered. The value of the confidence factor has been recorded at each image matching iterations and has been plotted in Fig. 4.12 for all of these cases. These ten experiments show that matched and the unmatched templates are well distinguished at the earlier stages of vector comparison. Furthermore, the unmatched template confidence factor stays close to 0% as the solution set in this case is

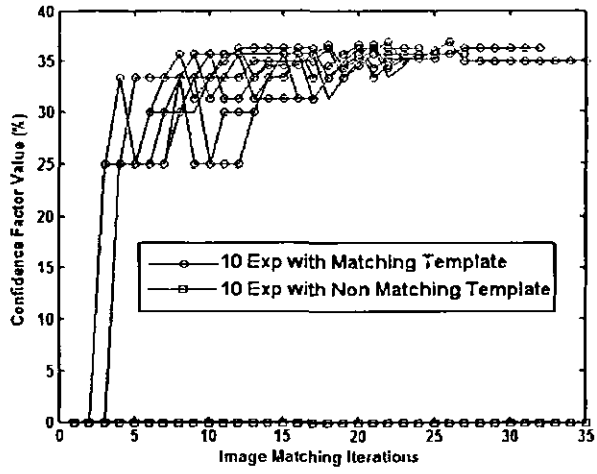


Fig. 4.12 Confidence Factor Value and its growth vs. Matching and Un-matched templates.

randomly populated, whereas in the case of a matched template many solutions show a fixed convergence trend towards the true match point which increases the value of this matching confidence factor rapidly.

4.4.3 Scale Invariance

A test run of the proposed algorithm was performed on database of satellite images in order to observe the scale invariance of the algorithm. Therefore, the template images were scaled from 0.4 to 2.0 with a step of 0.1. Scale value of "1" shows that the template image used in matching experiment is of the same scale as that of the main image. The scale greater than "1" shows expansion in the template prior to matching and vice versa. The results were consolidated in Fig. 4.13 for the proposed algorithm as well as the other schemes, i.e. Chamfer [14], RIMA [18], PCA RITF [159], FDIM [157] and 3D-FR [158]. These results are provided in the percentage P of right prediction of the templates for each scale factor. The scale range of 0.8 to 1.2 has been studied with a smaller step of 0.01 because the Chamfer [14] and RIMA [18] shows a degraded performance within this range. Chamfer [14] and RIMA [18] have failed to match outside the scaling factor of 0.8 to 1.2. However, FDIM [157] and 3D-

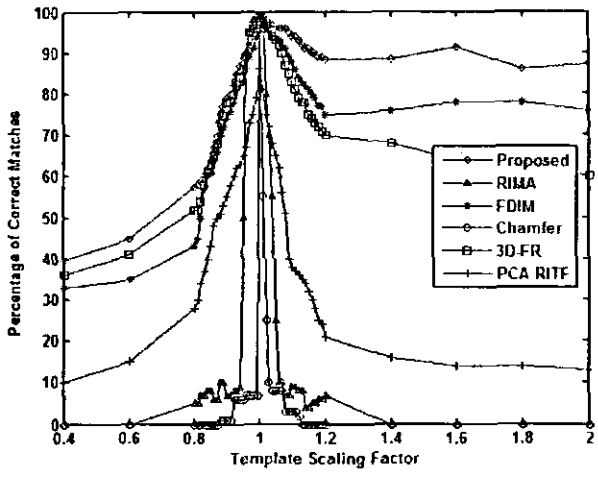


Fig. 4.13 Template matching error vs. the original template scaling factor.

FR [158] produced result comparatively less then the proposed algorithm. These schemes produce fairly accurate matching statistics in case of no scaling i.e. with scaling factor "1". The proposed scheme produces comparable image matching for the scaling factor of "1". However, the proposed scheme produces much better results than the other five schemes for scaling factors other than 0.9 to 1.1. For example, at a scaling factor of 1.2, Chamfer [14] fails to match, RIMA [18] shows 7% results, 3D-FR [158] shows 68% results, FDIM [157] shows 72% results, PCA RITF [159] shows 19%, whereas the proposed scheme shows 88.41% of correct matches. The scheme presented in chapter 3 shows 45% results at scale 1.2 as shown in Fig 3.23 (a). The proposed scheme matches image features using a normalized representation of feature signatures which enable the comparison with scale invariance over a greater scale range. However, the image features degrades more if it is shrunk then in the case of image expansion. This fact has affected the results in the same pattern as the factor P drops to 40% at a template scale of "0.4".

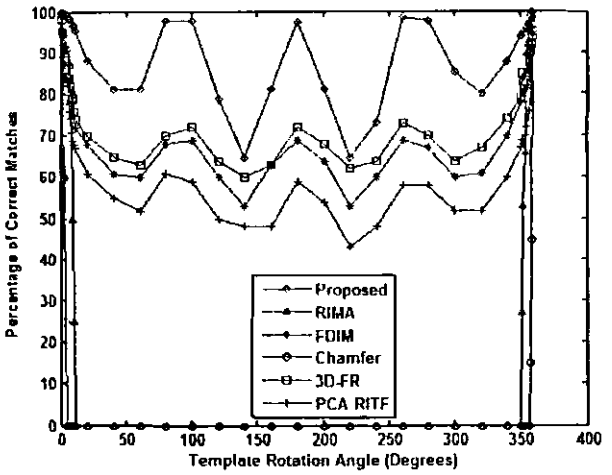


Fig. 4.14 Template rotation angle plotted vs. the percentage of correct solutions.

4.4.4 Rotation Invariance

A consolidated test run was made on 30 real satellite images. Templates were extracted from 16 different locations. Each template was given a rotation of 0° to 359° with the step of 20°. The angular range of 0° to 10° and 350° to 359° is, however, covered with a smaller step of 1°. Fig. 4.14 shows the value of factor P plotted against each rotated angle. The proposed scheme shows a good percentage for template matching for the whole 360° template rotation. RIMA [18] and Chamfer [14] show a limited range of rotation coverage close to 0° when the template image is only slightly rotated. For the rotation of 20° 3D-FR [158] shows 70% results, Chamfer [14] shows 0%, RIMA [18] shows 0%, FDIM [157] shows 67%, PCA RITF [159] shows 60% and the proposed scheme shows 90% of correct matches. The previous scheme of chapter 3 has shown 39% at 2° and 6.2% at 5° as shown in Fig 3.23(b) as a great distortion effect has been observed on the edges with minor rotations. This new proposed scheme, however, has made it possible to match feature shapes with a high degree of rotation invariance as it is representing each shape in the form of

normalized magnitude and phase signatures as shown in Fig. 4.14. Each magnitude shape signature is correlated in a 1D fashion to establish the best fit at a particular rotation angle. The angle estimate is then made through the phase representation of the same signature. Finally, the clustering of qualified solutions in a 4D space filters out the unwanted solutions and let the result converge towards the true rotation orientation. Note that the results are near ideal when the rotation is through a multiple of 90°, because in this way the rotation has the least effect on feature shapes. It is evident from Fig. 4.14 that the results of the proposed scheme deteriorate significantly for rotations of 45°. This is due to the fact that at this rotation the maximum number of pixels of the image has been estimated by an interpolator. We have used a bilinear interpolator. The results may improve significantly if a better interpolator is used.

4.4.5 Effect of Noise in the Image Matching

The proposed algorithm has also been evaluated for two different types of noises i.e. additive white Gaussian noise (AWGN) and impulsive noise. The behavior of the proposed algorithm was evaluated for 30 dB AWGN. The results were also compared at different scales of the template image as well as at different rotations of the template. The results were also compared with the other five techniques as well. Fig. 4.15 shows the percentage of correct matches for 0.4 to 2.0 scale range of the template image in the presence of 30 dB AWGN. The results show that the proposed algorithm performs better than the Chamfer [14], RIMA [18], 3D-FR [158], FDIM [157] and PCA RITF [159]. It has also been observed that proposed algorithm provides 85% of correct matches at 1.0 scale which is significantly greater than 73% of 3D-FR [158], 0% of Chamfer [14], 65% of RIMA [18], 68% of PCA RITF [159], 71% of FDIM [157] and 61% of the edge based scheme of chapter 3 as shown in Fig 3.24 (b). The results for different rotations in the presence of 30 dB AWGN are compared with

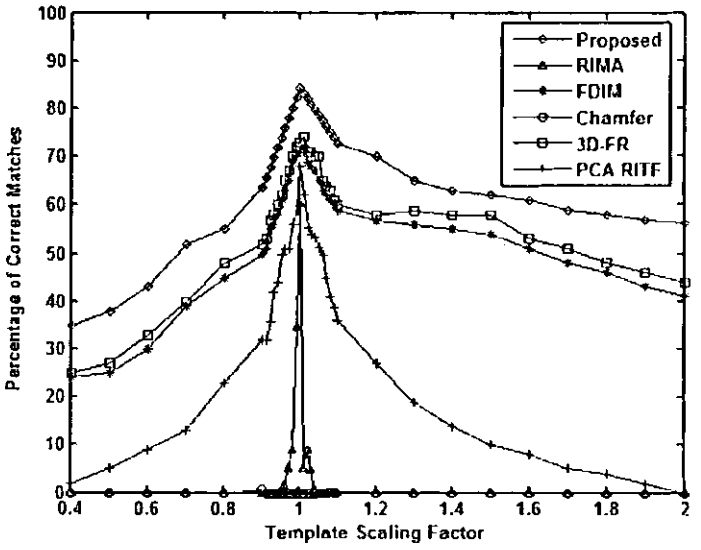


Fig. 4.15 Template matching error vs. the original template scaling factor with 30 dB AWGN.

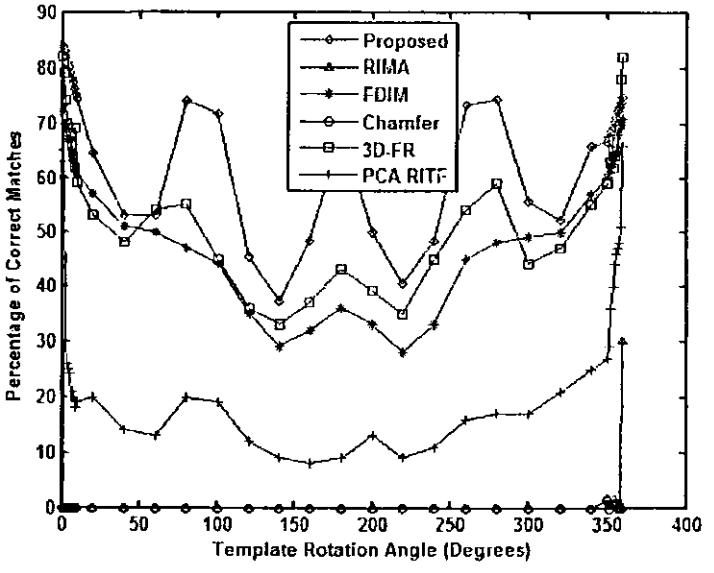


Fig. 4.16 Template rotation angle plotted vs. the percentage of correct solutions with 30 dB AWGN.

other methods in Fig. 4.16. The behavior of the results have similar pattern which was elaborated in Fig. 4.14. However, the results deteriorate slightly due to the presence of AWGN.

The results have also been assessed for excessive impulsive noise which is primarily due to imperfection/malfunctioning of different sensors. We have evaluated the proposed algorithm at different template scales as well as different template image rotations in the presence of 30 dB impulsive noise. Fig. 4.17 shows the comparison of the percentage of correct matches for 0.4 to 2.0 template scale range. The results in Fig. 4.17 shows that the proposed algorithm out performs than that of the Chamfer [19], RIMA [23], PCA RITF [17], FDIM [35] and 3D-FR [16]. A similar behavior is also observed at different template rotation angles in the presence of 30dB impulsive noise, as shown in Fig. 4.18.

The performance of the algorithm with respect to different noise levels is analyzed as shown in Fig 4.19. AWGN has more effect over the performance as the matching degrades to 66% with 40 dB level. Whereas the impulsive noise degrades the performance to 80% at 80 dB noise level. This is due to the fact that AWGN distorts the shapes of the image features more than the impulsive noise. The effect of the performance is analyzed with gray level shift for the template image. The results are consolidated in Fig 4.20. The shift of the gray level in the pixels of template image causes either the shape registration from one band to another or the distortion of the image shape feature. This distortion causes the degradation of the performance quoted earlier.

4.4.6 The Real Application Results

We have obtained several real images of district Swabi, NWPF, Pakistan. The images were obtained by a gray scale Sony camera (XC-ST70CE) with analog video out. This video was digitized at a resolution of 720×480 pixels at the height of 700 m. These snapshots represent real noisy/degraded, scaled and rotated images with varying illumination. A sample image of this class is shown in Fig. 4.21. The satellite image

of this area is also shown in Fig. 4.22. These 1000 real images were matched with the satellite images of the area using the proposed algorithm and the results show that the percentage of correct classification was 86%. However, the FDIM [157], RIMA [18], Chamfer [14], 3D-FR [158] and PCA RITF [159] showed the percentage of correct classification as 18.7%, 32%, 39.1%, 61.5% and 45.2%, respectively. The results clearly show that the proposed algorithm is capable of image matching under a variety of real environment.

4.5 CONCLUSIONS

This method presents a new image matching method using hyper vectorization of gray level sliced binary image. It is proposed to utilize the gray level binary sliced images for boundary extraction of connected patches. The process of signaturization has been proposed to form image feature vectors of fixed dimensions. Therefore, the proposed method represents the image gray shape features in terms of 1D normalized vector cluster. These vectors were proposed to be compared to find the image matching solution in an iterative way. Thus shape matching is performed in a much more effective way as the computation has been reduced to vector subtraction only. The image representation suggested in this scheme is devised in a way to facilitate the image matching in much lesser computations without the need for any inverse transformation. The results of the algorithm are fairly robust and computationally attractive. Image matching shows good performance for a complete 360^{O} rotated template and over a wide range of template scaling. The immunity against noise also shows to be much better than many techniques known in the literature.

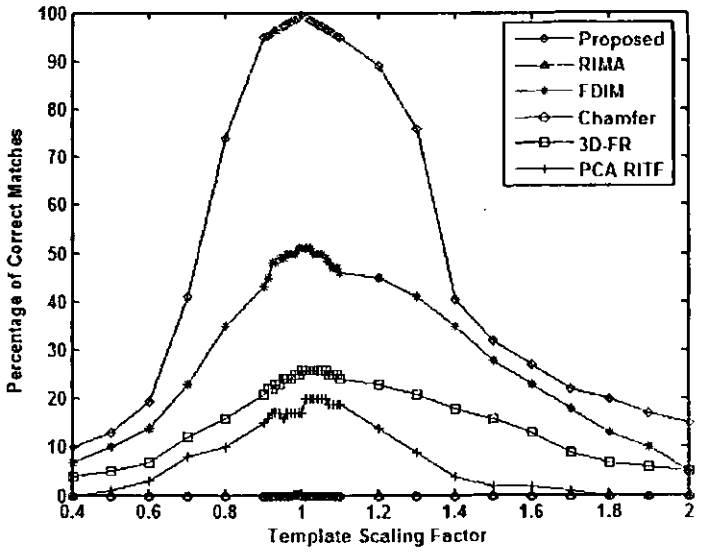


Fig. 4.17 Template matching error vs. the original template scaling factor with 30 dB Impulsive noise.

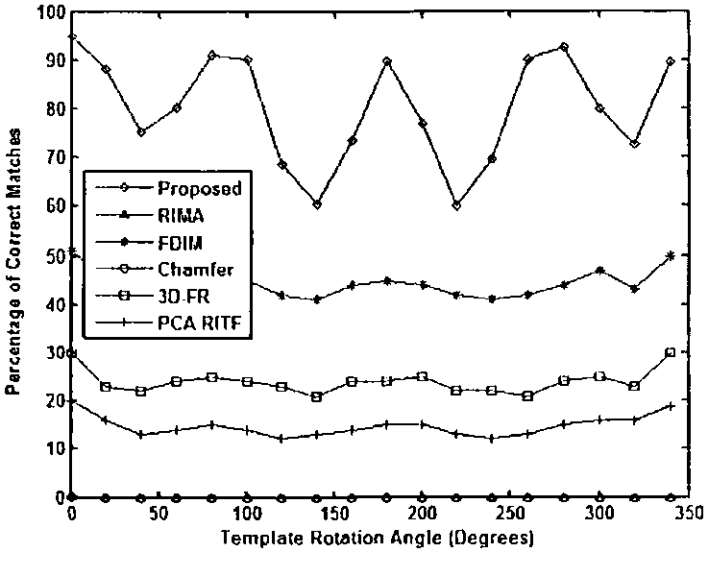


Fig. 4.18 Template rotation angle plotted vs. the percentage of correct solutions with 30 dB Impulsive noise.

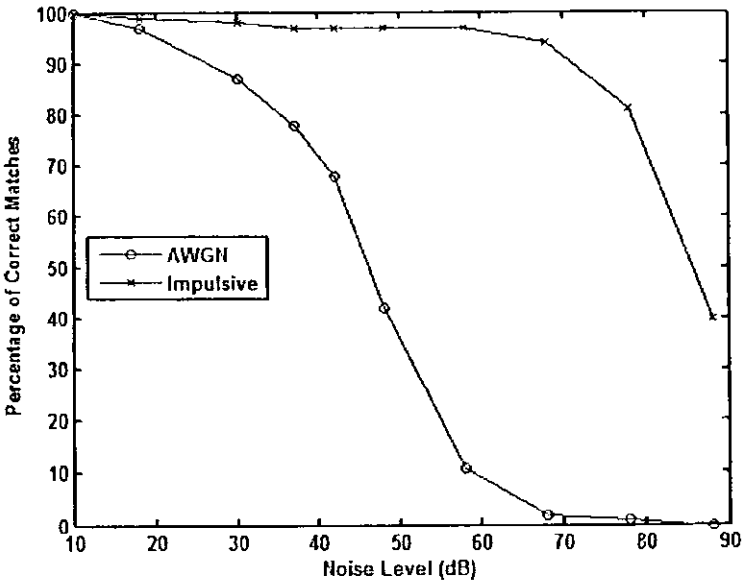


Fig. 4.19 Template matching error vs. the noise level of AWGN and Impulsive.

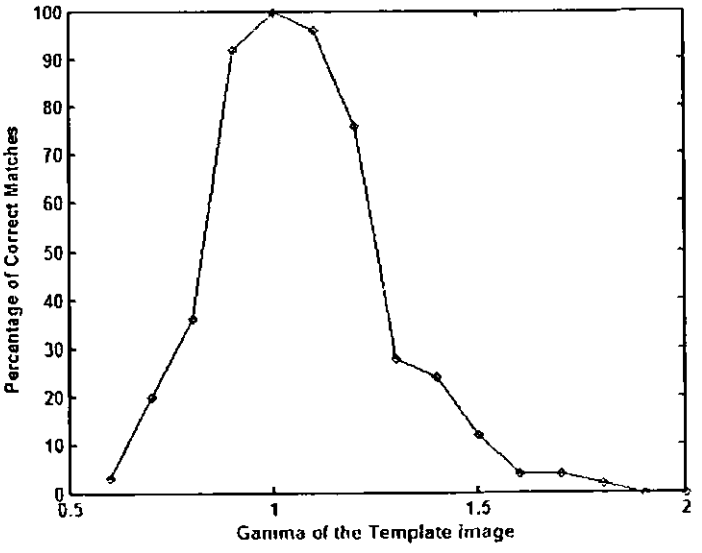


Fig. 4.20 Template matching error vs. the Gamma of the Template Image.

Fig. 4.21 A Real Template Image Sample.

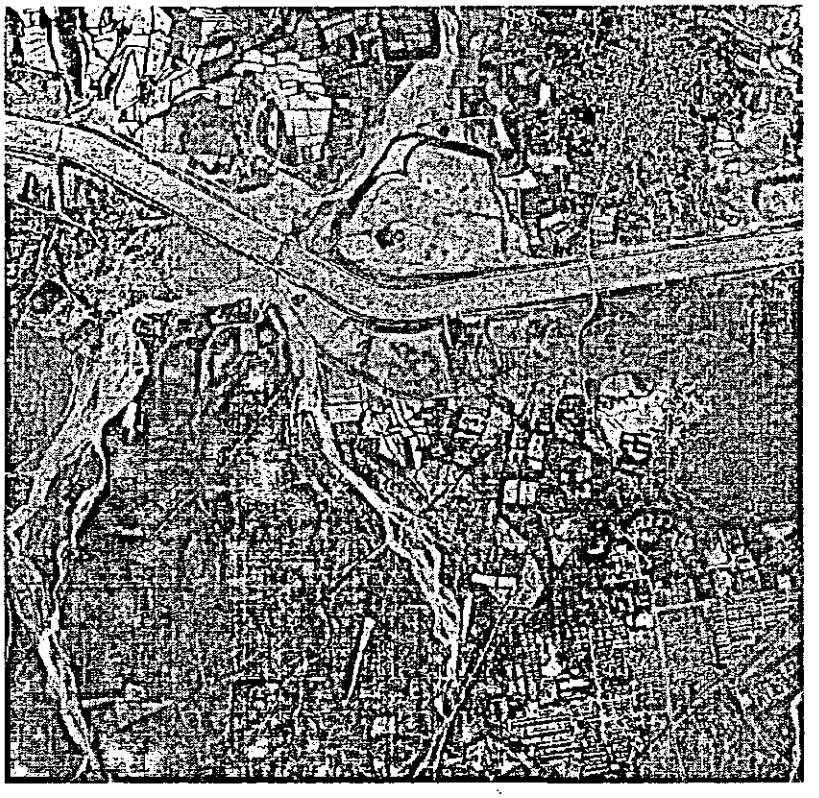


Fig. 4.22 Satellite Image of the Operation Area.

Chapter 5

Template Matching in a Video Sequence Using Hyper Vectorization and Adaptive Tracking

5.1 Adaptive Tracking and its Impact on the Computations

The proposed image matching scheme in the last chapter employ the image representation in a vectorized form. Hence, the main image as well as the template image will ultimately be represented in the form of a hyper-vector cluster. Image matching is achieved by comparing the two clusters representing the two images to be matched. The adaptive tracking scheme presented in this chapter facilitates the reduction of the cluster size of the main image. This reduction provides a vital impact on the processing time of the experiments.

5.2 Subset Extraction

Matching the template vectors with that of the complete set of hyper-vectors from the main image takes a lot of processing time. The reduction in the quantity of these hyper-vectors has a direct impact on the computations and speed of the algorithm. This reduction is achieved by making the subset $\bar{\xi}^{k,q}$ of the set of complete hyper-vectors $\xi^{k,q}$ of the main image as given by the following equation (5.2.1).

$$\tilde{\xi}^{k,q} = \left\{ \forall a \in \xi^{k,q} \mid \left| c_x^{k,q} - \left(\hat{V}_x + \frac{\kappa M_{\bar{f}}}{2\sin(\theta)} \right) \right| \le \alpha e, \left| c_y^{k,q} - \left(\hat{V}_y + \frac{\kappa N_{\bar{f}}}{2\sin(\theta)} \right) \right| \le \alpha e \right\} \quad (5.2.1)$$

This is achieved using the information of the current matching scale κ and rotation θ of the template image, along with the information about the next predicted matching location as estimated by the supervisory adaptive algorithm $\left(\hat{V}_x, \hat{V}_y\right)$. The actual size of the current template image size $\left(M_{\bar{f}}, N_{\bar{f}}\right)$ is scaled with the scaling factor κ and rotated by an angle θ to find the area of bounding rectangle placed at the next predicted matching location $\left(\hat{V}_x, \hat{V}_y\right)$. The shape centroid $\left(c_x^{k,q}, c_y^{k,q}\right)$ of all the vectors in $\xi^{k,q}$ are checked with an enlarged factor of safety "α" used to enhance the last prediction error "e".

5.3 Adaptive Prediction

Aerial vehicles mostly moves on a route which consists of legs and waypoints. The flying altitude and vehicle speed may vary according to the mission profile. The adaptive mechanism can be trained quickly to predict the next location of the vehicle according to the current speed. Similarly, changing the course and speed will again produce a larger error which will be used to modify the weights of the adaptive filter for the convergence towards a lower error solution. Least-Mean-Square and Recursive-Least-Square algorithm have been used for the prediction of the next matching location [156].

Adaptive supervisory algorithm is elaborated in Fig. 5.1. The image matching location for the mth template frame is converted to a complex input $\left(V_x + iV_y\right)_m$. The input for the adaptive filter is actually the difference of the two consecutive matching solutions.

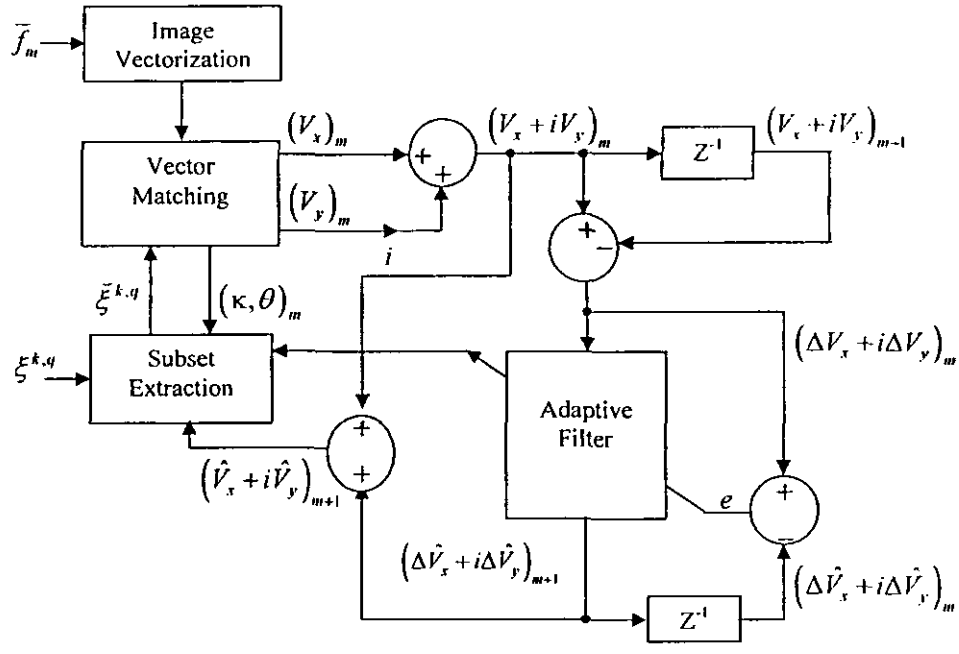


Fig. 5.1 Block diagram of the hybrid vectorized image matching and adaptive prediction mechanism.

$$\left(\Delta V_x + i\Delta V_y\right)_m = \left(V_x + iV_y\right)_m - \left(V_x + iV_y\right)_{m-1} \tag{5.3.1}$$

The adaptive filter is required to predict the next matching coordinate difference $\left(\Delta \hat{V}_x + i\Delta \hat{V}_y\right)_{m+1}$. This is then used to evaluate the predicted location of the next match as:

$$\left(\hat{V}_x + i\hat{V}_y\right)_{m+1} = \left(V_x + iV_y\right)_m + \left(\Delta \hat{V}_x + i\Delta \hat{V}_y\right)_{m+1} \tag{5.3.2}$$

whereas the error is found as:

$$e = \left(\Delta V_x + i\Delta V_y\right)_m - \left(\Delta \hat{V}_x + i\Delta \hat{V}_y\right)_m \tag{5.3.3}$$

5.4 Result and Discussion

The experiments are designed to test the accuracy and speed of the proposed image matching scheme. The effect of adaptive prediction on the performance towards

computation reduction is also emphasized. Statistical results are presented covering a wide variety of experimental aspects to show the robustness in the presence of noise.

5.4.1 Sample Image Experiment

A sample experiment is explained as follows. The main image is shown in Fig. 5.2 along with the route shown superimposed with black lines. 450 images have been extracted along the route and are shown in Fig. 5.3 separately. The size of the main image is of 1128×1592 pixels whereas each template image is of 100×100 pixels. The motion of the aerial vehicle is assumed to be with uniform speed and at constant altitude creating footprints of equal size and spaced equally apart. The main image has been vectorized and stored before hand. At the real time the incoming template images are vectorized and this vector set is matched with the reduced vector set of the main image. This reduce vector set is generated using the information about the prediction of the next vehicle location given by the adaptive filter associated with this matching mechanism. The error in prediction is used to train the weights of the adaptive filter for the next iteration as well as to extract the subset from the main image with a reasonable boundary margin. Experimental perimeters were taken as $B=4$, $\Im=100$, $\Upsilon=10$, $\varepsilon=10$, $\eta=10$ and $\alpha=2$. LMS (Least Mean Square) algorithm is used for the adaptive prediction with tap length of 32 and step size of 9×10^{-7}. For the first matching process, whole of the main image is considered whereas the subsequent region of search shrinks to a low value. Both the actual route and the predicted route are plotted in Fig. 5.4. It is to be noted that over the straight path, the error gradually reduces as the output of the adaptive filter tends towards the desired output. The search area after incorporating the predicted matching location, image scale and rotation is plotted in Fig. 5.5. The error is low as compared to the image size and

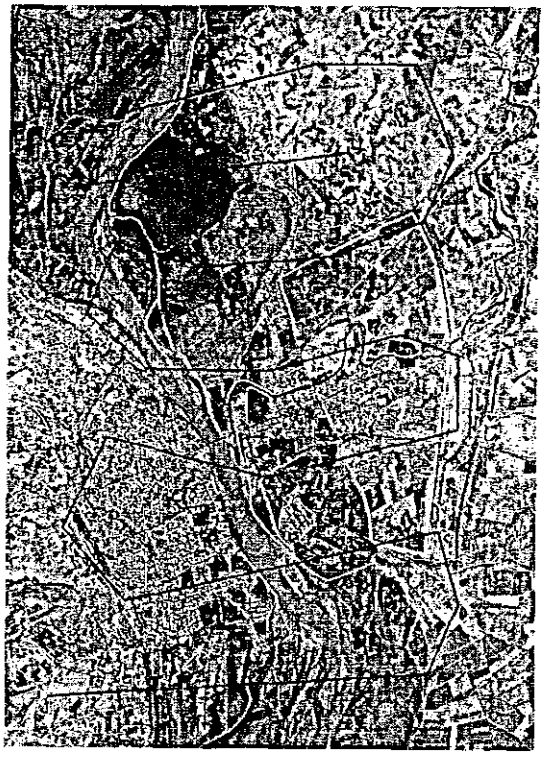


Fig. 5.2 Satellite imagery considered for sample test as shown along with the simulated route super-imposed.

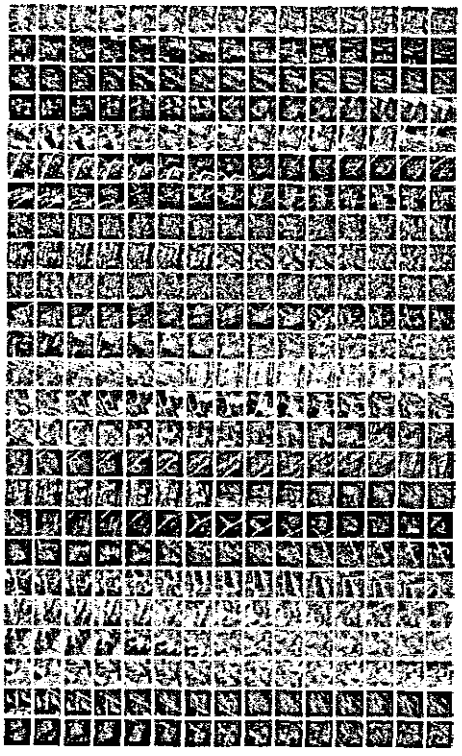


Fig. 5.3 Sample template images extracted along the route of Fig. 5.2.

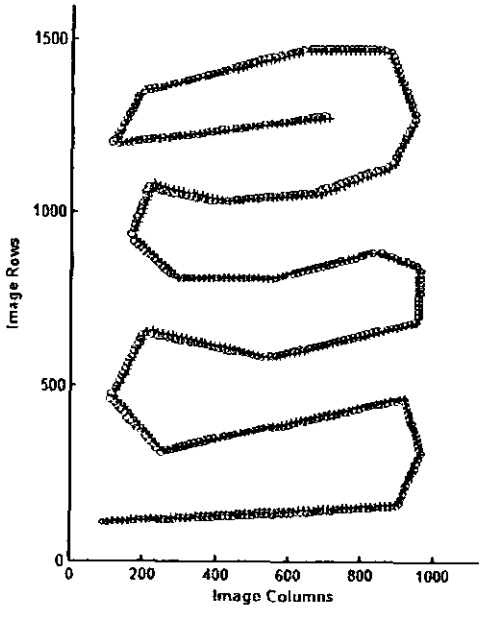


Fig. 5.4 Actual Route (o) Predicted Route (+).

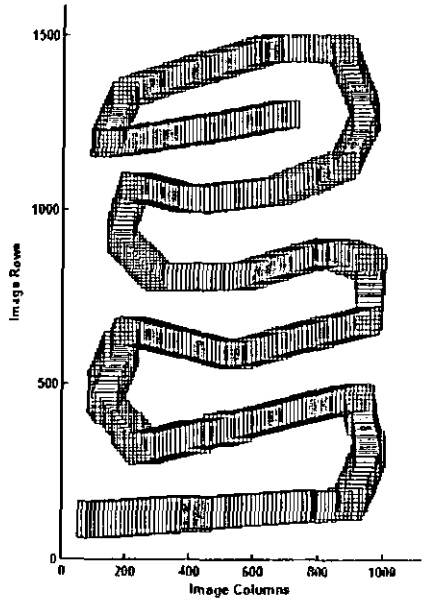


Fig. 5.5 Search area based on the prediction mechanism.

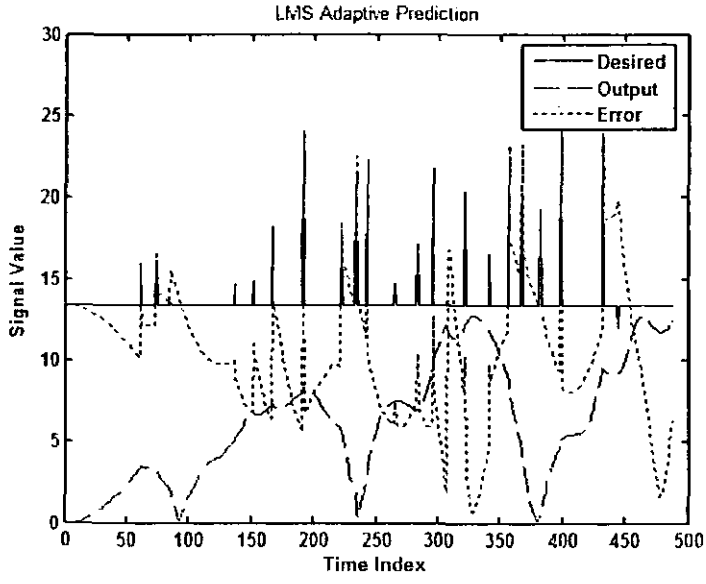


Fig. 5.6 Result of the adaptive prediction on the sample route of Fig. 5.2.

hence search area strictly follows the bounds given by the adaptive filter. The absolute values of the desired and the actual outputs along with the error is plotted in Fig. 5.6.

5.4.2 Database Image Matching Experiments

The speed of the image matching processing directly depends upon the search area. Multiple sizes of square main image have been considered varying from 100 to 1000 pixels with a step of 50 pixels. Time required for the image matching has been observed for a fixed template size of 100×100 pixels. Experiments has been performed on 2.0 GHz Intel Centrino Core2 Duo PC using Matlab® 7.3 on USC-SIPI-HAAI image database. This database of images contains a verity of aerial images having multiple ground features. Results are consolidated in Fig. 5.7 in which the main image size is plotted on the horizontal axis whereas the average time required for 250 image matching experiments is plotted on the vertical axis. It is to be noted that the variation of time with respect to image size is following a linear rise trend whereas the search area is being increased with square power of the image side.

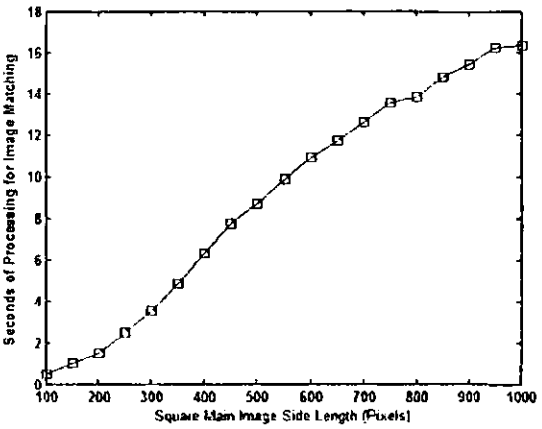


Fig. 5.7 Main image search area vs. the processing time.

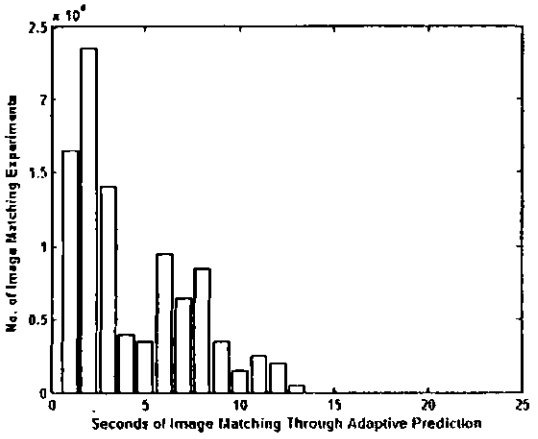
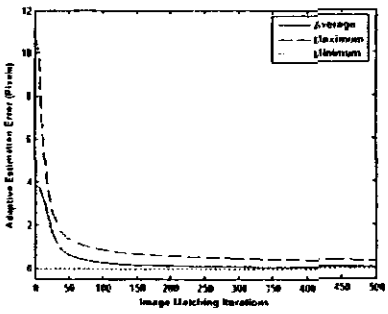


Fig. 5.8 Time required for the image matching experiments.

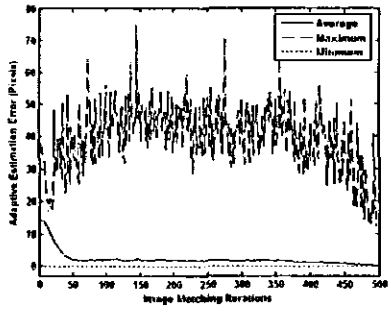
Hence, the reduction of the search area is desired to achieve a high level of speed performance for the proposed vector based image matching scheme.

The reduction of processing time has been observed once the adaptive prediction mechanism is incorporated along with the vector image matching scheme. For the experimental setup 16 satellite images has been considered. 12 routes have been plotted over these images with random waypoints coordinates. Each route is divided into 500 segments for the acquisition of the camera snapshots. This constitute 6000 image matching experiments per satellite image and a total of 96000 experiments. The time required for the image matching is plotted statistically in Fig. 5.8. When the error of the adaptive prediction is more, it causes a wider search area selection which requires more time for the proposed matching scheme. Where as the reduction of the error value along with the information about the current matching coordinates, scale and image rotation allows this search area to reduce considerably which in turns has a direct impact on the processing of the proposed scheme.

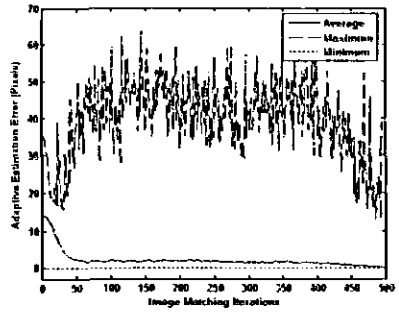
The convergence of adaptive prediction in this scenario is important as it is directly related to the performance of the proposed image matching scheme. Two types of adaptive setups are considered namely LMS and RLS (Recursive Least



(a) Straight Routes

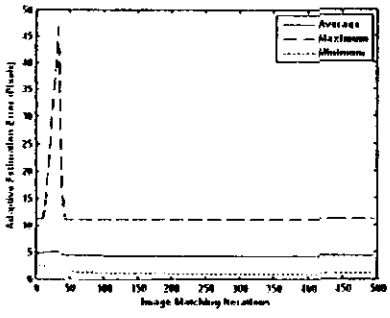


(b) Routes with multiple waypoints

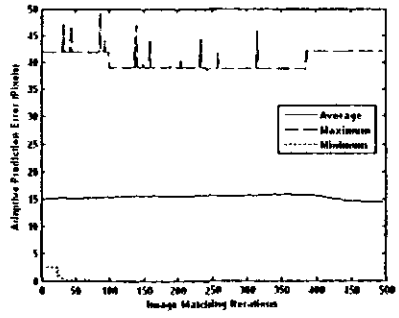


(c) Routes with multiple waypoints and varying velocity

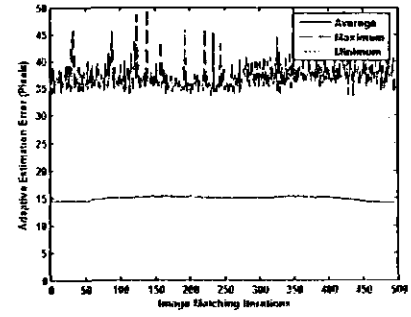
Fig. 5.9 Adaptive prediction of LMS algorithm for 96000 experiments. Absolute values of the average, maximum and minimum error vs. each image matching iteration.



(a) Straight Routes



(b) Routes with multiple waypoints



(c) Routes with multiple waypoints and varying velocity

Fig. 5.10 Adaptive prediction of RLS algorithm for 96000 experiments. Absolute values of the average, maximum and minimum error vs. each image matching iteration.

Square). Three route configurations are considered. Firstly straight routes, secondly routes with multiple waypoints and thirdly routes with multiple waypoints and randomly varying vehicle velocity. The experimental setup constitute of 16 satellite images, 12 routes per image, 500 waypoints per route for each adaptive setup and corresponding to each route configuration. For straight routes the average error value shows a fast reduction trend as shown in Fig. 5.9(a) and Fig. 5.10(a). For a multiple waypoint route, the average error trend shows an overall gradual convergence towards a zero valued error as shown in Fig 5.9(b, c) and Fig 5.10(b, c). Thus, the adaptive prediction mechanism supervises the subset extraction process and provides a tight bound in the search area boundaries. The proposed algorithm is compared with six different algorithms, classical correlation scheme [10], NCC [42], SSDA [160],

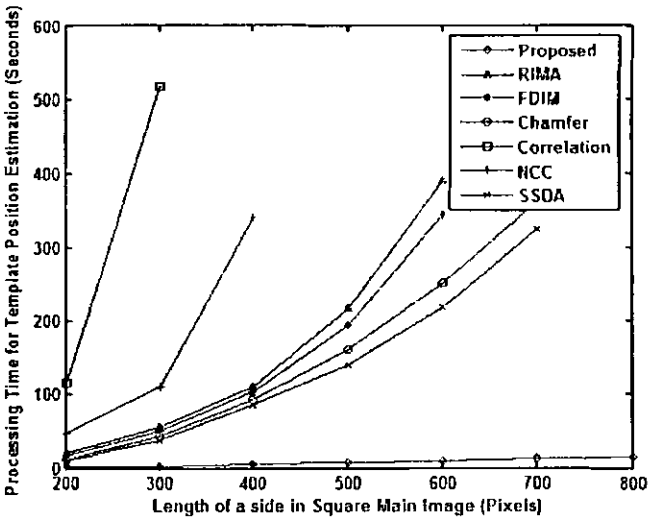


Fig. 5.11 Processing Time vs. the Search Area.

Chamfer image matching [14], Robust image matching algorithm (RIMA) [18] and Fourier Descriptors Image Matching (FDIM) [157]. The classical correlation scheme [10] accumulates pixel errors on the basis of their gray value difference. NCC [42] utilizes an advance form of the correlation. SSDA is a block matching algorithm used with early truncation approach [160]. Chamfer image matching [14] and RIMA [18] are edge matching algorithms using distance transforms of the images. FDIM [157] is a contour matching algorithm which compares image contour features. Time required for each scheme with respect to the main image search area on the similar platform is shown in Fig 5.11. The result shows that the proposed algorithm has the minimum processing time as compared to the rest of the techniques and it has also shown a linear dependency on the main image size. The scheme presented in chapter 4 shows 18.7 sec processing time as shown in Fig 4.6, whereas the incorporation of adaptive prediction for the reduction of search area in this proposed improvement shows 16.6 sec on the main image size of 800 pixels.

Fig. 5.12 Sample template image from the camera.

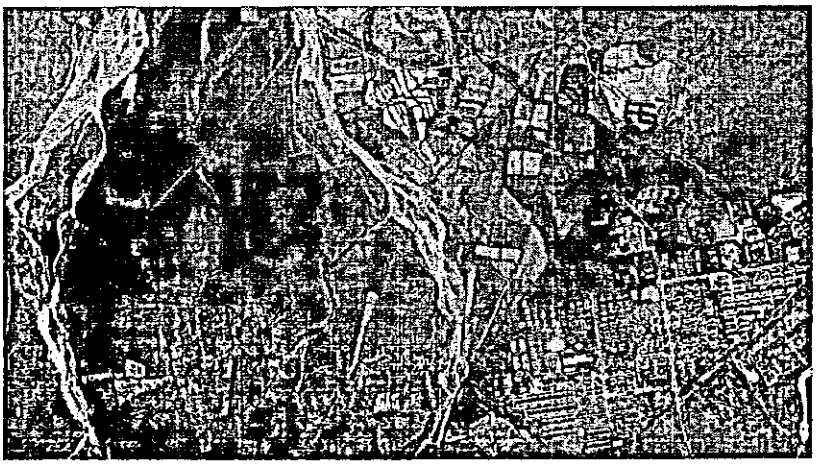


Fig. 5.13 Main image.

5.4.3 Real Application Results

We have obtained several real aerial images of district Mansehra, NWPF, Pakistan. The images were obtained by a gray scale Sony camera (XC-ST70CE) with analog video out. This video was digitized at a resolution of 720 × 480 pixels at the height of 700 m. These snapshots represent real noisy/degraded, scaled and rotated images with varying illumination. A sample image of this class is shown in Fig. 5.12. The satellite image of this area is also shown in Fig. 5.13. These 1000 real images were matched with the satellite images of the area using the proposed algorithm and the results show that the percentage of correct classification was 86%. However, the classical correlation [10], NCC [42], SSDA [160], Chamfer image matching algorithm [14], RIMA [18] and FDIM [157] showed the percentage of correct classification as 54.6%,

57.3%, 62.1%, 1.2%, 59.8% and 71.2%, respectively. The results show that the proposed algorithm is capable of image matching under a variety of real environment.

5.5 Conclusion

This scheme presents an improvement in the method explained in previous chapter. The supervisory modular algorithm checks the current matching parameters and predicts about the probable next matching location. The prediction is realized through an adaptive filter. The prediction helps the scheme to reduce the search area in the main image which has a direct effect on the speed of the image matching process. The results of the algorithm are fairly robust as the prediction error stays within a hundred pixel of the search area for both the prediction schemes. The prediction scheme is also computationally attractive as the search area in the main image is reduced to 10% of the total main image size. Image matching shows good performance for a complete 360^{O} rotated template and over a wide range of template scaling. The immunity against noise also shows to be much better than many techniques known in the literature.

Chapter 6

Conclusion

6.1 Summary of Results

Image matching through a high level feature comparison was presented in this dissertation. The goals were to achieve a fast, reliable and robust image matching. Three new schemes for the said purpose were discussed along with their test and statistical results.

The first scheme converts the image in the binary form through the process of edge detection. These edges can be listed in the form of hyper-vectors and the comparison of these vectors yield fast results as shown in the relevant chapter of this dissertation. This method gives good immunity against both impulsive and AWGN noises. As the process is accomplish in binary domain, higher level of gray scale invariance is also noted to be present in the scheme. Similarly, a wide range of scaling difference between the matching images is also covered. This range of the template scaling easily provides an efficient onboard solution for the application like visual aerial navigation. The scheme however, does not provide a reliable practical range of rotation invariance because of the wrong projection of the discrete hyper vectors data, once the template image is rotated.

The second scheme explores the use of gray scale information in the image after converting them into a hyper vector feature cluster. This hyper space cluster comparison gives a faster and robust matching solution over a wide range of template image scale and rotation difference. The iterative search mechanism also provides an attractive computation-reduction process. It is being supervised by an early truncation of the final solution.

The third hybrid scheme utilizes an adaptive supervisory mechanism for the prediction of the next template which is suitable if many templates are to be matched with one main image. This mechanism can be linked with any of the schemes proposed above. The impact of this prediction on computational overhead is discussed in the relevant section of this dissertation as the reduction in computation is vital because the adaptive prediction reduces the search area to a considerable limit.

The overall work of this dissertation emphasizes efficient methods of image matching for the application domain of visual navigation. The classical style of image matching starts the journey from correlation and its associated methods. The Chamfer and the methods associated with different image representation domains attempts to speed up this image matching. The modern methods are directed towards feature based image representation. The matching and image comparison is thus modified to become a high level feature processing. The presented work deals with the representation of image features in the form of hyper-vectors. Thus the complex form of feature matching reduces to a simple vector comparison yielding fast and robust image matching results.

6.2 Directions of Future Work

- Practical implementation of these schemes on hardware for a real time matching solution.

- Parallel processing and pipelined architecture is also to be explored for the maximum throughput of the performance in the proposed scheme.
- The effect of multi-dimensional data can also be explored for further improvement in data match time as a hybrid approach for future potential.
- Incorporation of texture classification can provide one good edge over the performance for the image matching.
- The scheme can be upgraded for an affine invariant transform to generate practical controls and navigation queues for an aerial vehicle.
- The high resolution and high speed cameras can be incorporated in the practical experiments for enhancing the accuracy of the system.
- High level ground features may also be vectorized through intelligent pattern recognition algorithms which can facilitate the identification of local and global navigation.
- Matching two images with different viewing angles may be looked into through a simulated 3-D world generation and correlating the features with actual camera shots.
- The feature identification of the images captured from the camera can be positions stamped and can further be used after integration for the curve fitting in hyper-dimensional representation of the main image features in the similar fashion.
- Image integration can be performed which employs matching of one frame image with the consecutive. This integration can also be used for the position estimation of the observer as well as its roll, pitch and yaw parameters. This

can provide a complete attitude picture for the control of an aerial vehicle on the basis of visual perception.

- The effect of infra red region imaging and multi sensor data fusion may also be explored and the features representation methodology may be revised in order to match the higher image features for a day-night operation.
- The role of edgelets and curvlets can also be explored for the sake of image vectorization and effective representation of image edges.

References:

[1] S. E. Hrabar and G. Sukhatme, "Omni directional vision for an autonomous helicopter," IEEE International Conference on Robotics and Automation, vol. 1, no. 1, pp. 558-563, 2003.

[2] P. D. Groves, R. J. Handley and A. R. Runnalls, "Optimizing the integration of terrain referenced navigation with INS and GPS," Journal of Navigation, vol. 59, no. 1, pp. 71-89, 2006.

[3] P. Ward, "GPS receiver RF interference monitoring, mitigation and analysis techniques," Navigation- Journal of the institute of Navigation, vol. 41, no. 4, pp. 367-392, 1994.

[4] S. D. Jones, C. Andersen and J. L. Crowley, "Appearance based process for visual navigation," Proceedings of the 1997 IEEE International Conference on Intelligent Robots and Systems, vol. 2, no. 1, pp. 551-557, 1997.

[5] Y. Bar-Shalon, H. M. Shertukde and K. R. Pattipati, "Extraction and optimal use of measurements from an imaging sensor for precision target tracking," IEEE International Conference on Control and Applications, vol. 1, no. 1, pp. 305-310, 1989.

[6] M. Soumekh, "Wide-bandwidth continuous-wave monostatic/bistatic synthetic aperture radar imaging," Proceedings of the 1998 International Conference on Image Processing ICIP 98, vol. 3, no. 1, pp. 361-365, 1998.

[7] J. P. Thiran and C. Michel, "Morphological registration of 3D medical images," Proceedings of IEEE International Conference on Image Processing, vol. 2, no. 1, pp. 253-256, 1996.

[8] R. Molina, "On the hierarchical Bayesian approach to image restoration: applications to astronomical images," IEEE Pattern Analysis and Machine Intelligence, vol. 16, no. 11, pp. 1122-1128, 1994.

[9] P. Lambert, S. Pires, J. Ballot and R. A. García, "Curvelet analysis of asteroseismic data, method description and application to simulated sun-like stars," Astronomy and Astrophysics, vol. 454, no. 3, pp. 1021-1027, 2006.

[10] N. L. Johnson and S. Kotz, George Udny Yule, in N L Johnson and S Kotz (eds.), "Leading personalities in statistical sciences (New York, 1997)," pp. 168-169, 1997.

[11] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 2nd Ed., Pearson Education, 2002.

[12] R. N. Nagel and A. Rosenfeld, "Ordered search techniques in template matching," Proceedings of the IEEE, vol. 60, no. 2, pp. 242-244, 1972.

[13] A. Goshtasby, S. H. Gage and J. F. Bartholic, "A two-stage cross correlation approach to template matching," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 6, no. 3, pp. 374-378, 1984.

[14] M. G. Alkhansari, "A fast globally optimal algorithm for template matching using low-resolution pruning," IEEE Transactions on Image Processing, vol. 10, no. 4, pp. 526-533, 2001.

[15] Z. Fang, X. Li and L. M. Ni, "Parallel algorithms for 2-D convolution," Proceedings of the 1986 International Conference on Parallel Processing, pp. 262-269, 1986.

[16] H. T. Kung and S. W. Song, "A systolic 2-D convolution chip," Proceedings of the IEEE Computing Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management, pp. 159-160, 1981.

[17] J. H. Chang, O. H. Ibarra, T. C. Pong and S. M. Sohn, "Two-dimensional convolution on a pyramid computer," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 4, pp. 590-593, 1988.

[18] S. Ranka and S. Sahni, "Convolution on mesh connected multi-computers," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 3, pp. 315-318, 1990.

[19] Y. Q. Shi and X. Xia, "A thresholding multi-resolution block matching algorithm," IEEE Transactions on Circuits and Systems for Video Technology, vol. 7, no. 2, pp. 437-440, 1997.

[20] K. M. Nam, J. S. Kim, R. H. Park and Y. S. Shim, "A fast hierarchical motion vector estimation algorithm using mean pyramid," IEEE Transactions on Circuits and Systems for Video Technology, vol. 5, no. 4, pp. 344-351, 1995.

[21] M. Ghanbari, "The cross-search algorithm for motion estimation," IEEE Transactions on Communications, vol. 38, no. 7, pp. 950-953, 1990.

[22] B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," IEEE Transactions on Circuits and Systems for Video Technology, vol. 3, no. 2, pp. 148-157, 1993.

[23] J. Chalidabhongse and C. C. J. Kuo, "Fast motion vector estimation using multi-resolution-spatio-temporal correlations," IEEE Transactions on Circuits and Systems for Video Technology, vol. 7, no. 3, pp. 477-488, 1997.

[24] Y. C. Lin and S. C. Tai, "Fast full-search block-matching algorithm for motion-compensated video compression," IEEE Transactions on Communications, vol. 45, no. 5, pp. 527-531, 1997.

[25] S. C. Cheng; H. M. Hang, "A comparison of block-matching algorithms mapped to systolic-array implementation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 7, no. 5, pp. 741-757, 1997.

[26] Y. S. Chen, Y. P. Hung and C. S. Fuh, "Fast block matching algorithm based on the winner-update strategy," IEEE Transactions on Image Processing, vol. 10, no. 8, pp 1212-1222, 2001.

[27] S. Mattoccia, F. Tombari, L. D. Stefano and M. Pignoloni, "Efficient and optimal block matching for motion estimation," 14th IAPR International Conference on Image Analysis and Processing (ICIAP 2007), vol. 1, no. 1, pp. 213-221, 2007.

[28] T. P. Plaks, "Mesh of linear arrays for template matching," Real Time Imaging, vol. 2, no. 6, pp. 373-382, 1996.

[29] H. R. Tsai, S. J. Horng, S. S. Tsai, S. S. Lee, T. W. Kao and C. H. Chen, "Optimal speed-up parallel image template matching algorithms on processor arrays with a re-configurable bus system," Computer Vision and Image Understanding, vol. 71, no. 3, pp. 393-412, 1998.

[30] H. I. Avi-Itzhak, J. A. V. Mieghem and L. Rub, "Multiple subclass pattern recognition: A maxi-min correlation approach," IEEE transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 4, pp. 418-431, 1995.

[31] H. Penz, I. Bajla, K. J. Mayer and W. Krattenthaler, "High-speed template matching with point correlation in image pyramids," Proceedings of SPIE Diagnostic Imaging Technologies and Industrial Applications, vol. 3827, pp. 85-94, 1999.

[32] J. Mutch, D.G. Lowe, "Multiclass object recognition with sparse, localized features," Proceedings of IEEE Computer Society conference on Computer Vision and Pattern Recognition, vol. 1, pp. 11-18, 2006.

[33] W. Krattenthaler, K. J. Mayer and M. Zeiller, "Point correlation: a reduced-cost template matching technique," Proceedings of IEEE International Conference Image Processing, 1994. (ICIP-94), vol. 1, pp. 208-212, 1994.

[34] J. You and P. Bhattacharya, "A wavelet-based coarse-to-fine image matching scheme in a parallel virtual machine environment," IEEE Transactions on Image Processing, vol. 9, no. 9, pp. 1547-1559, 2000.

[35] X. Cai and P. Ye, "Feature point set image matching algorithm for satellite attitude determination," 1st International Symposium on Systems and Controls in Aerospace and Aeronautics, pp. 212-216, 2006.

[36] M. Z. Zhang and V. K. Asari, "An efficient multiplier-less architechture for 2-D convolution with quadrant symmetric kernels," The VLSI Journal, vol. 40, no. 4, pp. 490-502, 2007.

[37] M. Khosravi and R. W. Schafer, "Low complexity matching criteria for image/video applications," Proceedings of IEEE International Conference on Image Processing, 1994, vol. 3, pp. 776-780, 1994.

[38] M. J. Atallah, "Faster image template matching in the sum of the absolute value of differences measure," IEEE Transactions on Image Processing, vol. 10, no. 4, pp. 659-663, 2001.

[39] S. Umeyama, "An eigen-decomposition approach to weighted graph matching problems," IEEE transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 5, pp. 695-703, 2002.

[40] A. J. Fitch, A. Kadyrov, W. J. Christmas and J. Kittler, "Fast robust correlation," IEEE transactions on Image Processing, vol. 14, no. 8, pp. 1063-1073, 2005.

[41] T. Kawanishi, T. Kurozumi, K. Kashino and S. Takagi, "A fast template matching algorithm with adaptive skipping using inner-sub-templates' distances," Proceedings of the 17th International Conference on Pattern Recognition, 2004, vol. 3, pp. 654-657, 2004.

[42] L. D. Stefano, S. Mattoccia and F. Tombari, "Fast template matching using bounded partial correlation," Machine Vision and Aplications, vol. 13, no. 4, pp. 213-221, 2003.

[43] L. D. Stefano, S. Mattoccia and F. Tombari, "ZNCC-based template matching using bounded partial correlation," Pattern Recognition Letters, vol. 26, no. 14, pp. 2129-2134, 2005.

[44] L. D. Stefano and S. Mattoccia, "A sufficient condition based on the Cauchy-Schwarz inequality for efficient template matching," Proceedings of International Conference on Image Processing, 2003, vol. 1, pp. 269-272, 2003.

[45] N. D. H. Dowson, R. Bowden and T. Kadir, "Image template matching using mutual information and NP-windows," Proceedings of the 18th International Conference on Pattern Recognition, vol. 2, pp. 1186-1191, 2006.

[46] X. Zhang, L. Li, X. Zhu, Y. Shang and O. Yu, "A weighted least squares image matching based target tracking algorithm," Proceedings of the SPIE on 27th International Congress on High-Speed Photography and Photonics, vol. 6279, no. 1, pp. 62793-62798, 2007.

[47] G. Ravichandran and D. Casasent, "Advanced in-plane rotation-invariant correlation filters," IEEE transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 4, pp. 415-420, 1994.

[48] B. John Oommen and G. Badr, "Breadth-first search strategies for trie-based syntactic pattern recognition," Pattern Analysis and Applications, vol. 10, no. 1, pp. 1-13, 2007.

[49] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles and H. C. Wolf, " Parametric correspondence and chamfer matching: Two new techniques for image matching," Proceedings of the 5th International Joint Conference on Artificial Intelligence, pp. 659-663, 1977.

[50] S. G. Mallat, "A theory for multi-resolution signal decomposition: The wavelet representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 7, pp. 674-693, 1989.

[51] G. Borgefors, "Hierarchical chamfer matching: a parametric edge matching algorithm," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 6, pp. 849-865, 1988.

[52] A. Ghafoor, R. N. Iqbal and S. Khan, "Robust image matching algorithm," Proceedings of the 4th EURASIP Conference Focused on Video/Image Processing and Multimedia Communications, vol. 1, pp. 155-160, 2003.

[53] J. You, W. Zhu, E. Pissaloux and H. Cohen, "Hierarchical image matching: A chamfer matching algorithm using interesting points," International Journal of Real-Time Imaging, vol. 1, pp. 245-259, 1995.

[54] C. T. Yang, S. Lai and L. Chang, "Hybrid image matching combining Hausdorff distance with normalized gradient matching," Pattern Recognition, Elsevier Science Inc., vol. 40, no. 4, pp. 1173-1181, 2007.

[55] J. Chen, P. Tan and T. Goh, "Refinement to the chamfer matching for a "center-on" fit," Image and Vision Computing NZ, pp. 360-365, 2003.

[56] A. Thayananthan, B. Stenger, P. H. S. Torr and R. Cipolla, "Shape context and chamfer matching in cluttered scenes," Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision And Pattern Recognition, vol. 1, pp. 127-133, 2003.

[57] D. H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," Patter Recognition, vol. 13, no. 2, pp. 111-122, 1987.

[58] A. Margalit and A. Rosenfeld, "Using probabilistic domain knowledge to reduce the expected computational cost of template matching," Computer Vision, Graphics and Image Processing, vol. 51, no. 3, pp. 219-234, 1990.

[59] A. Margalit and A. Rosenfeld, "Using feature probabilities to reduce the expected computational cost of template matching," Computer Vision, Graphics and Image Processing, vol. 52, no. 1, pp. 110-123, 1990.

[60] M. A. Sid-Ahmed, "Serial architectures for the implementation of 2-D digital filters and for template matching in digital images," IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 38, no. 5, pp. 853-857, 1990.

[61] S. Inglis and I. H. Witten, "Compression-based template matching," Proceedings of the Data Compression Conference, 1994, pp. 106-115, 1994.

[62] M. Khosravi and R. W. Schafer, "Template matching based on a grayscale hit-or-miss transform," IEEE Transactions on Image Processing, vol. 5, no. 6, pp. 1060-1066, 1996.

[63] L. Prasad and S. S. Iyengar, "High performance algorithms for object recognition problem by multi-resolution template matching," Proceedings of 7th International Conference on Tools Artificial Intelligence, 1995, vol. 5, no. 8, pp. 362-365, 1995.

[64] R. Reeves, K. Kubik, "Compressed domain image matching using symmetric convolution," Proceedings of IEEE Region 10 Annual Conference on Speech and Image Technologies for Computing and Telecommunication, vol. 2, pp. 543-546, 1997.

[65] M. Uenohara and T. Kanade, "Use of Fourier and Karhunen-Loeve decomposition for fast pattern matching with a large set of templates," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 8, pp. 891-898, 1997.

[66] M. Ni and S. E. Reichenbach, "Pattern matching by sequential subdivision of transformation space," Proceedings of the 17th International Conference on Pattern Recognition, 2004, vol. 2, pp. 145-148, 2004.

[67] F. Tombari, S. Mattoccia and L. D. Stefano, "Template matching based on the L_p norm using sufficient conditions with incremental approximations," IEEE International Conference on Video and Signal Based Surveillance, pp. 20-20, 2006.

[68] F. Essannouni, R. O. H. Thami, D. Aboutajdine and A. Salam, "Fast L4 template matching using frequency domain," Electronics letters, vol. 43, no. 9, pp. 507-508, 2007.

[69] Wei, S. D. Liu, S. W. Lai, S. Hong, "Fast template matching by applying Winner-update on Walsh-Hadamard domain," IEEE conference on Acoustic, Speech and Signal Processing, vol. 1, pp. 1029-1032, 2007.

[70] Y. Liang, R. M. Guest, M. C. Fairhurst and J. M. Potter, "Feature-based assessment of visuo-spatial neglect patients using hand-drawing tasks," Pattern Analysis and Applications, vol. 10, no. 4, pp. 361-374, 2007.

[71] R. Klettea and P. Zamperoni, "Measure of correspondence between binary patterns," Image and Vision Computing, vol. 5, no. 4, pp. 287-295, 1987.

[72] E. Persoon and K. S. Fu, "Shape discrimination using Fourier descriptors," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 3, pp. 388-397, 1987.

[73] H. Kauppinen, T. Seppanen and M. Pietikainen, "An experimental comparison of autoregressive and Fourier-based descriptors in 2d shape classification," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 2, pp. 201-207, 1995.

[74] C. T. Zahn and R. Z. Roskies "Fourier descriptors for plane closed curves," IEEE Transactions on Computers, vol. 21, no. 3, pp. 269-281, 1972.

[75] P. L. Rosin and S. Venkatesh, "Extracting natural scales using fourier descriptors" Pattern Recognition, vol. 26, no. 9, pp. 1383-1393, 1993.

[76] C. D. Stefano, F. Tortorella and M. Vento, "An entropy-based method for extracting robust binary templates," Machine Vision and Applications, vol. 8, no. 3, pp. 173-178, 1995.

[77] F. S. Cohen and W. Jin-Yinn, "Modeling image curves using invariant 3-D object curve models: a path to 3-D recognition and shape estimation from image contours," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 1, pp. 1-12, 1994.

[78] M. Baroni and G. Barletta, "Digital curvature estimation for left ventricular shape analysis," Image and Vision Computing, vol. 10, no. 7, pp. 485-494, 1992.

[79] J. R. Bennet and J. S. MacDonald, "On the measurement of curvature in a quantized environment," IEEE Transactions on Computers, vol. 24, no. 8, pp. 803-820, 1975.

[80] R. M. Cesar Jr. and L. da F. Costa, "Towards effective planar shape representation with multi-scale digital curvature analysis based on signal processing techniques," Pattern Recognition, vol. 29, no. 9, pp. 1559-1569, 1996.

[81] D. M. Wuescher and K. L. Boyer, "Robust contour decomposition using a constant curvature criterion," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 1, pp. 41-51, 1991.

[82] W. Richards, B. Dawson and D. Whittington, "Encoding contour shape by curvature extrema," Journal of the Optical Society of America, vol. 3, no. 9, pp. 1483-1491, 1986.

[83] A. Rattarangsi and R. T. Chin, "Scale-based detection of corners of planar curves," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 4, pp. 430-449, 1992.

[84] P. Meer, E. S. Baugher and A. Rosenfeld, "Extraction of trend lines and extrema from multi-scale curves," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 3, pp. 217-226, 1988.

[85] F. Mokhtarian and A. K. Mackworth, "Scale based description and recognition of planar curves and two-dimensional shapes," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 5, pp. 674-675, 1986.

[86] F. Mokhtarian and A. K. Mackworth, "A theory of multi-scale, curvature-based shape representation for planar curves," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 8, pp. 789-805, 1992.

[87] J. A. Garcia and J. F. Valdivia, "Representing planar curves by using a scale vector," Pattern Recognition Letters, vol. 15, no. 9, pp. 937-942, 1994.

[88] A. Bengtsson and J. O. Eklundh, "Shape representation by multi-scale contour approximation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 1, pp. 85-93, 1991.

[89] P. Maragos, "Pattern spectrum and multi-scale shape representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 7, pp. 701-716, 1989.

[90] S. Loncaric, "A survey of shape analysis techniques", Pattern Recognition, vol. 31, no. 8, pp. 983-1001, 1998.

[91] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 10, pp. 1615-1630, 2003.

[92] H. C. Huang and Y. P. Hung, "Adaptive early jump-out technique for fast motion estimation in video coding," Graphical Models and Image Processing, vol. 59, no. 6, pp. 388-394, 1997.

[93] A. M. Waxman, M. C. Seibert, A. Gove, D. A. Fay, A. M. Bernardon, C. Lazott, W. R. Steele and R. K. Cunningham, "Neural processing of targets in visible, multi-spectral IR and SAR imagery," Neural Networks, vol. 8, no. 7-8, pp. 1029-1051, 1995.

[94] A. Katz and P. Thrift, "Hybrid neural network classifiers for automatic target detection," Expert Systems, vol. 10, no. 4, pp. 243-244, 1993.

[95] J. Sas and U. Markowska-Kaczmar, "Semi-supervised handwritten word segmentation using character samples similarity maximization and evolutionary algorithm," CISIM, vol. 1, no. 1, pp. 316-321, 2007.

[96] W. E. L. Grimson and D. P. Huttenlocher, "On the sensitivity of the Hough transform for object recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 3, pp. 255-274, 1990.

[97] R. T. Al-Zubi and D. I. Abu-Al-Nadi, "Automated personal identification system based on human iris analysis," Pattern Analysis and Applications, vol. 10, no. 2, pp. 147-164, 2007.

[98] F. Juric, M. Dhome, "Hyper-plane approximation for template matching," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 996-1000, 2002.

[99] R. Deriche and G. Giraudon, "A computational approach for corner and vertex detection," International Journal of Computer Vision, vol. 10, no. 2, pp. 101-124, 1993.

[100]C. H. Chen, J. S. Lee and Y. N. Sun, "Wavelet transformation for gray-level corner detection," Pattern Recognition, vol. 28, no. 6, pp. 853-861, 1995.

[101]U. Kothe, "Integrated edge and junction detection with the boundary tensor," Proceedings of 9th IEEE International Conference on Computer Vision, 2003, vol. 1, pp. 424-431, 2003.

[102]E. Sojka, "A new approach to detecting the corners in digital images," Proceedings of International Conference on Image Processing, vol. 3, pp. 445-448, 2003.

[103]K. J. Lee, Y. T. Kim, H. C. Myung, J. M. Kim and Z. Bien, "A corner matching algorithm with uncertainty handling capability," Proceedings of the Sixth IEEE International Conference on Fuzzy Systems, 1997, vol. 3, pp. 1469-1474, 1997.

[104]S. Messelodi, C. M. Modena and M. Zanin, "A Computer based vision system for the detection and classification of vehicles at urban road intersections," Pattern Analysis and Applications, vol. 8, no. 1-2, pp. 17-21, 2005.

[105]A. Kolcz, J. Alspector, M. Augusteijn and M. Zanin, "A line-oriented approach to word spotting in handwritten documents," Pattern Analysis and Applications, vol. 3, no. 2, pp. 153-168, 2000.

[106]D. G. Lowe, "Object recognition from local scale-invariant features," Proceedings of the 7th IEEE International Conference on Computer Vision, 1999, vol. 2, pp. 1150-1157, 1999.

[107]P. H. Gosselin, M. Cord and S. P. Foliquet, "Kernels on bags for multi-object database retrieval," Proceedings of the 6th ACM international conference on Image and video retrieval, pp. 226-231, 2007.

[108]S. Lyu, "Mercer kernels for object recognition with local features," Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 223-229, 2005.

[109]F. Saitoh, "Image template matching using pixels with local three median gray-levels," The Journal of the Institute of Image Electronics Engineers of Japan, vol. 33, no. 5, pp. 822-828, 2004.

[110] Y. Amit, D. Geman and K. Wilder, "Joint induction of shape features and tree classifiers" IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 11, pp. 1300-1305, 1997.

[111] S. Belongie and J. Malik, "Matching with shape context," IEEE Workshop on content-based access of image and video libraries, 2000.

[112] D. P. Huttenlocher, R. H. Lilien and C. F. Olson, "View-based recognition using an Eigen-space approximation to the Hausdorff measure," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 9, pp. 951-955, 1999.

[113]Y. H. Tseng and H. J. Lee, "Document image binarization by two-stage block extraction and background intensity determination," Pattern Analysis and Applications, vol. 11, no. 1, pp. 33-44, 2008.

[114]D. M. Gavrila and J. Giebel, "Virtual sample generation for template-based shape matching," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 676-681, 2001.

[115]F. Pernus, A. Leonardis and S. Kovacic, "Two-dimensional object recognition using multi-resolution non-uniform-preserving shape features," Pattern Recognition Letters, vol. 15, pp. 1071-1079, 1994.

[116]K. Mikolajczyk, A. Zisserman, and C. Schmid, "Shape recognition with edge-based features," Proceedings of British Machine Vision Conference, vol. 2, pp. 779-788, 2003.

[117]C. Fermuller and W. Kropatsch, "A syntactic approach to scale-space-based corner description," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 7, pp. 748-751, 1994.

[118]S. Belongie, J. Malik and J. Puzicha, "Shape matching and object recognition using shape contexts," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 4, pp. 509-522, 2002.

[119]O. Carmichael and M. Hebert "Shape-based recognition of wiry objects," Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 401-408, 2003.

[120]A. Berg, T. Berg and J. Malik, "Shape matching and object recognition using low distortion correspondences," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, vol. 1, pp. 26-33, 2005.

[121]S. Fumihiko, "Rotation invariant image template matching based on correlation of local contour form," The Journal of the Institute of Image Electronics Engineers of Japan, vol. 34, no. 6, pp. 753-759, 2005.

[122]W. Förstner, "A framework for low level feature extraction" Lecture Notes in Computer Science, vol. 801, pp. 383-394, 1994.

[123]C. D. Schrider, J. A. Skipper and D. W. Repperger, "Histogram-based template matching for object detection in images with varying contrast," Proceedings of the SPIE Image Processing: Algorithms and Systems V., vol. 6497, pp. 64970B, 2007.

[124]K. Graumen and T. Darrell, "The pyramid match kernel: Discriminative classification with set of image features", Proceedings of the Tenth IEEE International Conference on Computer Vision, 2005, vol. 2, pp. 1458-1465, 2005.

[125]M. Liu, T. X. Han and T. S. Huang, "Online appearance learning by template prediction," IEEE conference on Advanced Video and Signal Based Surveillance, pp. 236-241, 2005.

[126] Y. Hel-Or and H. Hel-Or, "Real-time pattern matching using projection kernels," IEEE Transactions on Pattern Analysis and machine Intelligence, vol. 27, no. 9, pp. 1430-1445, 2005.

[127] A. Thayananthan, R. Navaratnam, P. H. S. Torr and R. Cipolla, "Likelihood models for template matching using the PDF projection theorem," British *Machine Vision Conference*, 2004.

[128] T. Mita, T. Kaneko and O. Hori, "A probabilistic approach to fast and robust template matching and its application to object categorization," Proceedings of the 18th International Conference on Pattern Recognition, 2006, vol. 2, pp. 597-601, 2006.

[129]R. S. Choras, T. Andrysiak and M. Choras, "Integrated color, texture and shape information for content-based image retrieval," Pattern Analysis and Applications, vol. 10, no. 4, pp. 333-343, 2007.

[130]A. Schlapbach and H. Bunke, "A writer identification and verification system using HMM based recognizers," Pattern Analysis and Applications, vol. 10, no. 1, pp. 33-43, 2007.

[131]G. P. Penney, J. Weese, J. A. Little, P. Desmedt, D. L. G. Hill and D. J. Hawkes, "A comparison of similarity measures for use in 2-D-3-D medical image registration," IEEE Transactions on Medical Imaging, vol. 17, no. 4, pp. 586-595, 1998.

[132]D. Patel and T. J. Stonham, "Texture image classification and segmentation using RANK-order clustering," Proceedings of 11th International Conference on Image, Speech and Signal Analysis, vol. 3, pp. 92-95, 1992.

[133]J. You and H. A. Cohen, "An orientation and resolution independent texture classifier in segmentation of images of unknown rotation and scale," Proceedings of 11th IAPR International Conference on Image, Speech and Signal Analysis, vol. 3, pp. 49-52, 1992.

[134]L. Shang, "Image vectorization in digital image watermarking," Proceedings of the SPIE on Visual Communications and Image Processing 2003, vol. 5150, pp. 872-879, 2003.

[135] C. D. Bei and R. M. Gray, "An improved fast encoding algorithm for vector quantization," Journal of the American Society for Information Science and Technology, vol. 55, no. 1, pp. 81-87, 2004.

[136]H. Peng, F. Long and Z. Chi, "Document image recognition based on template matching of component block projections," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 9, pp. 1188- 1192, 2003.

[137]L. Kotoulas and I. Andreadis, "Real-time computation of Zernike moments," IEEE Transactions on Circuit and System for Video Technology, vol. 15, no. 6, pp. 801-809, 2005.

[138]C. Grassl, T. Zinsser and H. Niemann, "Efficient hyper-plane tracking by intelligent region selection," 6th IEEE Southwest Symposium on Image Analysis and Interpretation, pp. 51-55, 2004.

[139]F. Jurie and M. Dhome, "A simple and efficient template matching algorithm," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no.7, pp. 996-1000, 2002.

[140] Z. J. Zhang, S. B. Huang and Z. L. Shi, "Image matching through combined features of singular value and region," Proceedings of 2004 International Conference on Machine Learning and Cybernetics, vol. 6, pp. 3793-3797, 2004.

[141] B. C. Song, M. J. Kim and J. B. Ra, "A fast multi-resolution feature matching algorithm for exhaustive search in large image databases," IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, no. 5, pp. 673-678, 2001.

[142] Q. Li and B. Zhang, "Template matching based on image gray value," Proceedings of the SPIE Visual Communications and Image Processing 2005, vol. 5960, pp. 614-622, 2005.

[143]F. Tang and H. Tao, "Fast multi-scale template matching using binary features," IEEE international conference on Acoustic, Speech and Signal Processing, vol. 1, pp. 36-36, 2007.

[144]T. X. Han, V. Ramesh, Y. Zhu and T. S. Huang, "On optimizing template matching via performance characterization," Proceedings of the Tenth IEEE International Conference on Computer Vision 2005, vol.1, pp. 182-189, 2005.

[145]F. Ullah and S. Kaneko, "Using orientation codes for rotation-invariant template matching," Pattern Recognition Letters, vol. 37, no. 2, pp. 201-209, 2004.

[146]G. J. F. Banon and S. D. Faria, "Area-based matching algorithm assessment from satellite images," Proceedings of European Congress for Stereology and Image Analysis, vol. 20, no. 2, pp. 573-578, 2001.

[147]D. Hall, B. Leibe and B. Schiele, "Saliency of interest points under scale changes," Proceedings of the British Machine Vision Conference, pp. 646-655, 2002.

[148]D. G. Lowe, "Distinctive image features from scale invariant key-points," International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[149]K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," International Journal of Computer Vision, vol. 60, no. 1, pp. 63-86, 2004.

[150]C. Schmid, R. Mohr and C. Bauckhage, "Evaluation of interest point detectors," International Journal of Computer Vision, vol. 37, no. 2, pp. 151-172, 2004.

[151]M. Zuliani, C. Kenney and B. S. Manjunath, "A mathematical comparison of point detectors," Conference on Computer Vision and Pattern Recognition Workshop, vol. 11, pp. 172-172, 2004.

[152]F. Wu and H. Schweitzer, "Fast selection of linear features in image data," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 3, pp. 49-49, 2005.

[153]A. Harol, C. Lai, E. Pekalska and R. P. W. Duin," Pair-wise feature evaluation for constructing reduced representations," Pattern Analysis and Applications, vol. 10, no. 1, pp. 55-68, 2007.

[154]P. Sala, R. Sim, A. Shokoufandeh and S. Dickinson, "Landmark selection for vision-based navigation," IEEE Transactions on Robotics, vol. 22, no. 2, pp. 334-349, 2006.

[155]A. K. Jain, "Fundamentals of Digital Image Processing," Prentice Hall, Englewood Cliffs, NJ 07632, pp. 434-437, 1989.

[156]S. Haykin, Adaptive Filter Theory, 4th edition, chapter 5 and 9, 2002.

[157]I. Kunttu and L. Lepisto, "Shape-based retrieval of industrial surface defects using angular radius Fourier descriptors," IET Image Processing, vol. 2, no. 1, pp. 231–236, 2007.

[158]C. Samir, A. Srivastava and M. Daoudi, "Three-dimensional face recognition using shapes of facial curves," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 28, pp. 1858–1863, 2006.

[159]A. Jalil, A. Manzar, A. Zahoor and I. M. Qureshi, "Rotation-invariant features for texture image classification," IEEE International Conference on Engineering of Intelligent System, pp 22-23, April 2006.

[160] A. Hatabu, T. Miyazaki and I. Kuroda, "Optimization of decision-timing for early termination of SSDA-based block matching," Proceedings of the ICME'03, vol. 2, pp. 821-824, 2003.

[161] H. Wolfson and I. Rigoutsos, "Geometric Hashing: An overview," IEE Computer Science and Engineering, vol. 4, pp. 10-12, 19971.