Matching Remote Sensing Images

by

Sherif H. AbdelSayed, B.A.Sc.

A thesis submitted to the
School of Graduate Studies and Research
in partial fulfilment of the requirements
for the degree

Master of Applied Science

Ottawa-Carleton Institute for Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
University of Ottawa
September 1995

ISBN 0-612-15581-1

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

I hereby declare that I am the sole author of this thesis.

I authorize the University of Ottawa to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Sherif H. AbdelSayed

I further authorize the University of Ottawa to reproduce this thesis by photocopying or other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Sherif H. AbdelSayed

ABSTRACT

Image analysis plays a crucial role in many computer vision applications in which images of the same scene with different geometrical orientations need to be compared for further processing.

This thesis describes the design and implementation of a model-based vision system for the recognition of aerial images. The main objective is to register two remote sensing images taken at different times. First, some distinctive features are extracted and matched then, these matched features are used as marking points in defining a geometric mapping function. Once registered, the reference image can be used as an aid to automatic interpretation and as a framework for detecting changes between successive images.

A two stage matching procedure is used for this task. In the first part, corners are extracted and matched in both images and an initial estimation of the mapping function is computed. This initial function is then used in the second part to estimate the parameters of a global mapping function for the entire image.

The process ends when all the extracted features in one image are either mapped to features in the other image, or rejected if no match could be found.

ACKNOWLEDGMENT

I wish to express my sincere gratitude to my supervisor, Dr. Dan Ionescu, for his constant guidance, encouragement, and support throughout my research.

I would like to thank the faculty and staff of the Department of Electrical Engineering, University of Ottawa, for their kindness and support.

Above all I would like to thank my family in law for their constant support and especially my wife, Sally, and our new baby, George Junior, for their love and caring.

Contents

List of Tables

List of Figures

Notation

Various symbols, superscripts, subscripts, and abbreviations used frequently in this thesis are summarized below. All notation is fully defined where it first arises in the text.

Symbols

$\mathcal{E}(\ )$	The mathematical expectation.
$G(u), F(u)$	Scoring Function.
K	Cornerity factor.
C_i	Control point or order i.
$r_i(t)$	Spline polynomial of order i
$x(i,j)$	The state values of the image.
$y(i,j)$	The observed values of the image.
$\mathbf{Y}$	Observation matrix.
R	Rotation Matrix.
T	Translation matrix.
H	Registration transformation function.
$\mathbf{I}$	Identity matrix
M_{pq}	Normalized Moment of order $(p+q)$.

$P(\ )$	Probability distribution.
A_i	Normalized invariant moment of order i.
r	Coefficient of correlation.
C_r	Contrast Ratio.

Greek Letters

γ/Γ	Multiplicative observation noise
μ_{pq}	Central moment of order (p,q)
σ_x^2	Statistical variance of x
Σ_a	Estimation derivative
θ	Rotational angle
Δ	Gradient
α	Scaling ratio
Ψ_{mv}	Minimum Variance Covariance

Special Symbols

$\bar{x}$	The mean of the value x
$\hat{x}$	The estimate of the value x

Acronyms and Definitions

2-D	Two Dimensional
3-D	Three Dimensional
AR	Auto-regressive
SPOT	System Probatoire d'Observation de la Terre
CCRS	Canadian Center for Remote Sensing
SAR	Synthetic Aperture Radar.
HRVS	High Resolution Visible Sensors
HYPER	Hypothesis Predicted and Evaluated Recursively
ARG	Attributed Relational Graphs
RPE	Recursive Parameter Estimation
DRO	Duda Road Operator
ICP	Iterative Closest Point
GCP	Ground Control Points
MMSE	Minimum Mean Square Error
LSE	Least Square Error
SNR	Signal to Noise Ratio

Chapter 1

Introduction

Image analysis is in general a complex task which requires different processing stages. A prime cause of this complexity is the presence of texture that causes difficulties for the low level processes such as edge detection and image matching. The capabilities required for such processes vary with the application and the complexity of the image. For many applications, a complete analysis is unnecessary and cost prohibitive.

However, many important decisions such as measuring the similarity between regions, detecting changes or performing mapping of two similar images and locating objects of interest in aerial images may be made based on the information extracted from these images. This created the need to have the images registered and matched for meaningful results without having to go for exhaustive analysis.

Image matching can be regarded as a process of comparing two images, in order to detect the differences (mainly rotation, translation and scaling) between those images. Systems where image registration and matching are key stages include

matching target with real-time images of the scene for target recognition, surface recognition such as monitoring global land usage using satellite images, matching stereo images to recover shapes for autonomous navigation, robot vision, dynamic biomedical diagnosis and aligning images from different modalities for diagnosis.

Matching different images involves recognizing patterns of features within an image description, guided by a *priori* models. In other words, given a pictorial description of a region or a scene. we can determine which region in another image is similar. The main problem in matching images lies in recognizing an instance of a known model pattern. A pattern can be seen as a set of locally defined primitives, where each of which can be located independently of the rest by examining small regions of the image. The matching task includes locating the pattern and finding a geometrical registration function that does the job of superimposing the instance and the model. There are mainly two approaches to deal with the matching problem. The first, known as the interest operator, extracts selected primitives(e.g., edges, corners, distinctive patterns, etc) separately form the two images and tries to match them [12],[15]. The second, builds on the notion that the brightness patterns are the same or at least similar in a small neighborhood of each image point and tries to use correlation matching techniques along epipolar lines, using either row images or warped images to find the corresponding points.

This thesis describes a framework for a recognition system applied to aerial images without human intervention. To achieve this task, a number of stages which include preprocessing, filtering, enhancement, segmentation and matching are applied. For this purpose, SPOT [1] images provided by the Canadian Center for Remote Sensing (CCRS) in Ottawa, Ontario are used.

[1]System Probatoire d'Observation de la Terre

1.1 Motivation

Extensive work was been performed in the past in the area of image matching for remote sensing applications. The main target for that was to replace the human operator by an automated system.

In the last two decades several researchers have exploited non-geometric feature properties in the form of semantic labels or order relation [2],[3],[4]. Many algorithms were introduced to cope with images subjected to geometrical deformation such as translation, few algorithms to cope with scaling and fewer with rotation.

With the launch of LANDSAT-1 in July 1972, and the subsequent launches of LANDSAT-2 in January 1975, and LANDSAT-3 in March 1978 remotely sensed imagery over wide areas of the world became readily available. The multi-spectral sensors on board the satellite provides simultaneous images in four spectral bands in the visible and infra-red regions to a ground resolution of 79 meters.

Currently, a satellite with a spatial resolution of 10 meters is the French satellite (SPOT). It uses High Resolution Visible (HRV) sensors (known as PLA). This has been launched in a nearly polar sun-synchronous orbit of about 900 km., in order to give repetitive coverage: every 18 days the satellite passes over the same place on the earth [23]. With this amount of information in hand, a computer assisted analysis of aerial images is needed for the various processing stages.

This thesis addresses the design and implementation of an automated recognition system that is capable of extracting predetermined features and performing matching between two images using these features.

The following section explains the approach followed in the implementation of

this system applied in aerial images.

1.2 Approach

In order to achieve the best performance, the recognition system is broken down to 7 major stages in which each of these steps is responsible for a specific task. These stages are described as follows;

- For most aerial images the contrast may not be high, first linear features and line intersections are detected and extracted from the gray scale images.
- In the next step, these features are modeled as B-splines. These are characterized by a set of parameters (control points) from which a curve can be totally generated. The B-splines are piecewise polynomial functions that can provide local approximations of contours of shapes using a small number of parameters. Modeling the image using the splines results in reducing the number of points that represent the boundaries. This in turn limits the storage requirement of the model. It will also smoothen the coarsely digitized contours and makes it reliable in reducing some of the noise that is embedded in the boundaries. This technique of modeling has been widely used in computer graphics, recognition of parts from boundaries, and shape analysis and synthesis. The control points that represent the boundary serve as descriptors of the features in the image. This approach was selected for its simplicity, limited storage requirements, and the invariance to the parameters for translation, rotation and scaling of the image. The invariance property simplifies the hypothesis generation and verification operations.

- By comparing the control points of the scene segments with the control points of different models representing segments of the reference image, the result is a list of hypothesized line segments.

- In order to reduce the computing time, a two stage matching process is used:

    1. In the first part, line intersections (corners) that are extracted from the feature extraction stage are compared and matched (since number of corners are small compared to other line features in a given image). This is obtained by measuring the correlation coefficient for a circular window around these corners. From these matched points, an initial registering transformation is calculated.

    2. The second part is to classify the line segments transforming by them using the previously computed transformation function. Each time a segment is matched, a new transformation function is calculated and used in the next step to match or reject new segments. A recursive least square parameter estimation technique is used to update the estimation of the transformation.

- The verification operation starts with the line segment that has the highest likelihood of being present. If so the presence is confirmed and its corresponding segment is removed from the scene. The verification is resumed with the next hypothesized segment until all line segments forming the scene are identified. Figure 1.1 outline the basic framework for the matching system

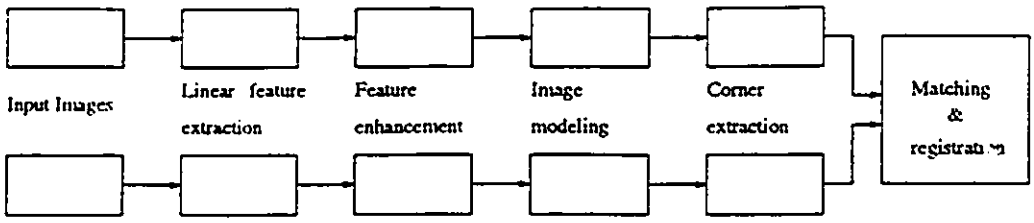


Figure 1.1: A block diagram of image recognition framework.

The general approach for the matching system can be summarized in the following steps:

1. Image Acquisition: Acquire a 2-D SPOT-PLA image The image is stored in raster sun format.

2. Preprocessing: The two-dimensional image is digitized, filtered, enhanced, thresholded, and stored as a 512x512 binary image.
   Three different steps are to be used in this step:

   (a) Application of a detection algorithm: the Duda Road Operator (DRO) has been found to offer the best results.

   (b) Enhancing the edge image using local transformation based on the mean and variance of the intensity maps and the intensity of the input image.

   (c) Thresholding the image with a variable threshold value computed through the histogram of the gray level image.

3. Image representation using the B-spline to convert the smooth curves into approximated line segments.

4. Modeling: Calculate the control points for line segment. This is the last operation in the learning mode. In the recognition mode we add the following steps.

5. Hypotheses Generation: Compare the control points of the scene with those of a reference image [2] The result of this step is a list of hypothesized objects.

6. Classification: Eliminate or add segments that do not satisfy predefined criteria from a list of the extracted segments. The criteria are based on a least square error between the matched segments. This step is iterative, meaning that each time a new segment is matched, it is added to the list of matched points and a new transformation is calculated with a recursive least square technique.

7. Verification: Confirm or reject the presence of a given hypothesis after every iteration using a quality measure. In this step a quality measure is used to test the validity of the match through a measurement of length of the identified model segments as a percentage of the total model length. This can take the value of 1, which is the maximum value obtained whenever the model is perfectly and completely identified in the scene. This factor decreases in the absence of some parts of the image or non-rigid distortions.

[2]The reference image is an image that is taken for the same view and is going through the same process as the scene.

1.3 Organization of this thesis

This chapter gave a brief introduction of the work described in this thesis. Chapter 2 covers prior approaches of known matching methods that tried to deal with the problem of registration of deformed images, in particular in the field of remote sensing. Chapter 3 describes the algorithm used in modeling shapes and extracting features in arbitrary scenes. Chapter 4 describes image modeling using the B-Spline algorithm. The two stage matching procedure and the algorithm used for registering the two images are described together with the mathematical derivation in chapter 5. Chapter 6 describes the software implementation and the experimental results. Some data structures as well as recursive functions are employed in feature generation and extraction. The registration and matching modules of the program are described. The optimized search strategy that was used is also described. Finally, conclusion shows some lessons learned in developing new approaches suggestions for the direction of future research.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, we review some approaches and systems developed to deal with matching and registration of images subjected to geometrical deformation. The techniques used in these systems were model-based, that is, object models were used to guide the matching process [1]. It was found that the matching processes vary according to the type of features used [36]. Statistical pattern classification techniques are usually employed with global feature-based systems. Local feature-based systems generally use the hypothesize-verify paradigm. They recognize the scene elements in two phases: in the first phase, hypotheses are generated, in the second phase, the hypotheses are tested and their validity is established. Almost all the vision systems that address geometrical deformation use local features since global features are not invariant under some types of deformation (e.g elastic deformation). Examples of

[1] Models are descriptions of the features, they can be based on either global features such as area, perimeter,centroid, etc., or on local feature such as local boundary characteristics.

local features are: longest segment of polygon approximation [30], holes, convex and concave corners [22], sub-templates[23], fixed length boundary segments [24], local lines and circular edge fragments[3]. Hypotheses are generated either by a Hough transform[3],[23] or by distance measures defined for feature pairs.

Matching can be conducted in several ways including: least squares distance between sequence of points, relaxation [28], search in interpretation space [3], sophisticated indexing of model features [30], growing clusters around matched privileged features [22],[14]. In the next section, we review some characteristics of these systems.

Davis [19] found approximate matches (together with registration) of pieces of shapes (simple closed polygons). The deformation included translation, rotation and a range of scale changes. First, he assigned figures of merit to pairwise-matched angles. Then, a relaxation method was used to prune matching. The results showed that the method was robust, but it was very difficult to implement for complex shapes.

Bolles and Cain [22] used local features (holes, corners, and clusters of adjacent features) to create the object models. The cluster of features was obtained by locating the most significant feature in the image and "growing" it by adding neighboring features. The matching process involved creating an association graph. Each node of the graph corresponded to a match (matches represent scene features similar to model features). Hypotheses were generated by finding the largest connected sub-graph (maximal clique). Hypotheses were tested by checking for object boundary consistency. This technique is not completely general since it assumes that the model objects have sharp local features close to each other, and that the cluster of local features must be visible for the part to be recognized.

Furthermore, the method relies on algorithms for solving the NP-complete maximal clique problem, so their performance is likely to degrade significantly as the number of features increase[22].

Ballard [3] generalized the Hough transform to recognize and locate the non-analytic shapes, whereby, a feature in "image space" is mapped into a locus of possible registrations in the "dual parameter space". The parameter space is tiled and represented by an array of counters. Each feature point votes for registrations by incrementing counters in its dual locus. A global minimum count identifies a best registration. The method is efficient when feature properties include orientation, size, and location. However, with the presence of noise, this method becomes complicated due to the discrete nature of the tiling.

Ayache and Faugeras [17] developed a recognition system called HYPER (Hypothesis Predicted and Evaluated Recursively). In this system, objects are modeled by their polygonal approximations, the angles between successive sides and salient sides. Hypotheses are generated by assigning salient sides of the models to compatible segments of the scene. The best hypotheses are verified by iteratively matching remaining model segments and updating the transformation matrices until a high quality measure of similarity is reached or enough hypotheses have been evaluated. Estimation of the affine transformation that maps some models onto the corresponding scene features is made recursively through the application of a Kalman filter. The efficiency of this method depends on the correct choice of privileged sides and the visibility of the corresponding privileged sides in the scene.

Grimson and Lozano-Perez [3] can locate entities in a chirped scene from sparse position and orientation data measurements. The extracted entities are modeled by local measurements of 3D positions and surface normal. Models consist of polyhe-

dral entities represented by their planar faces. The information about these faces (such as equations) and the relations between faces (such as distance) are also computed. Sparse range or tactile data of 3D objects are used as scene features. The matching process consists of two steps. First, an interpretation tree is generated by associating sides of the model with sensed points. Each path of the tree corresponds to an interpretation that is consistent with local geometric constraints. Interpretations inconsistent with local constraints are verified by a transformation test. An interpretation is accepted if it can be used to obtain a transformation that would place each sensed point on an object surface. After fast selection of an interpretation. the second step determines object position and validates the interpretation. To handle extraneous data points in a cluttered scene, sensed points can be ignored if no remaining side of a model is consistent with the interpretation and if enough points have been interpreted. To limit the size of the combinatorial search, only the complete interpretations in the tree are kept. This sometimes leads to identification errors.

Turney, Mudge, and Volz [23] used boundary curves for matching. They divided the model, or template, of length n pixels into $\frac{n}{2}$ sub-templates, each of length h. Each sub-template of these models was compared against the entire image boundary. To improve the system efficiency, the matching of a sub-template against the boundary was performed by a least-squares fit in θ_s space, where θ represented tangent slope and s represents the arc length along the curve. This method worked well for overlapping objects, but it was slow since the number of sub-templates is large.

Grosky and Mehrota [30] used a feature index organized as a search tree. In the recognition phase, a feature from the analyzed portion of the scene data was taken as a test feature. The feature index was searched to find the best matching model feature. The model/location list associated with the model feature was used

to hypothesize the identities and location in the image of the possible objects. The hypotheses verification was done by finding matching points in the scene data for selected points on the transformed boundary of a hypothesized object. The selected points belong to privileged strings. Each privileged string starts at a sharp vertex. The method was not robust since the sharpness of vertices can change and result in the selection of incorrect privileged strings and produce low accuracy of recognition.

Kalvin et al. [32] generated footprints [2] of boundary subsections . The footprint of a subsection is hashed to select a set of candidate models having similar footprints. Each of the candidate models was matched against the boundary segment under consideration to select the best candidate and to decide its location. The process of footprint generation was computationally complex and the number of candidate models found by hashing is usually large in case of noisy and complex scenes.

Koch and Ragasam [32] used special vertices of the polygon to detect and locate the model in the scene. Polygon moment calculation led to estimates of the dissimilarity between the scene and model corners, and determined the model corner location in the scene.

Bhanu and Faugeras [28] used the segment lengths, slope of the segment, the angle between two segments, and the inter-vertex distance as features. A shape measure indicated the goodness of a match between a model and a scene segment. A stochastic labeling scheme was then used to label each of the model segments as either a scene segment or NIL (no match). This method was computationally intensive and required a good estimate of the initial assignment of the labels for the convergence of the algorithm and validity.

[2]The footprint of an object is an n-dimensional curve derived from the boundary of that object where this curve is dependent on a starting point of the boundary.

Price and Reddy[18] developed a complex symbolic matching and registration system. They used 10 feature properties including line segments. In the matching process, every model segment was compared to every scene segment. If the inter-segment angle and the length are within certain thresholds, the model segment is said to be compatible with the scene segment. Then, their orientation difference is stored in an array known as a disparity array. Since segments of an object were arranged sequentially along the object contour, segments between the model and the scene would likely to be matched in sequence. The longest consecutive sequence of matched segments between the model and the scene correspond to the longest compatible consecutive diagonal entries of the disparity array that have similar orientation differences. A transformation that aligns the model segments with the matched scene segments was evaluated. Applying this transformation to the model segments, disparity values based on the segment positions and orientations were updated and stored in the disparity array. The final matches between the model and the scene segments were decided by finding the longest compatible consecutive diagonal entries of the new disparity array. Price's procedure was simple, but not computationally efficient. The technique was also sensitive to scale variations since the features used are scale-dependent.

Bhanu and Ming [28] suggested an approach similar to Price's but they used a different matching process. The matching process first applied the K-mean clustering algorithm iteratively on the disparity array until the optimal number of clusters was found. It then checked for the elements of each cluster that were in sequential order, and found the sequences. Several heuristics were included to determine the sequences. The process then clustered the sequences averages using the same clustering algorithm. The cluster that contains the largest number of sequences decided the final matches between the model and the scene segments. A confidence value defined as the ratio of the cumulative length of the segments to the local length

of the model segments was evaluated to verify the final match. This approach was computationally expensive due to the iterative nature of the matching algorithm

Maitre and Wu developed a method using dynamic programming to register and match images subjected to elastic deformation. Their method made use of a combination of an autoregressive (AR) modeling of the deformation working at the pixel level. Dynamic programming provided several sequences of the observations that can be used as candidates for the AR model. This method was limited to translations and was modified later [36] to cope with limited rotations. It also needed a set of ordered comparable sequence of primitives to be extracted from the image before performing the registration.

Besl and McKey [16] used free form curves and surfaces as well as point sets to handle six degrees of freedom and compute the global registration function. Their method worked as follows; given 3D data in a sensor coordinate system and a shape data set in a model coordinate system, they estimated the optimal rotation, translation and scaling that aligned, the model shape and the data shape minimizing the distance between them and thereby allowing the equivalence of the shapes via a mean-square distance metric. This procedure is not useful if only a subset of the data point shape corresponds to the model shape or a subset of it, i.e, the number of points in the data set of the scene image must be equal to the number of points in the data set of the model image. This limitation is avoided by modifying this procedure[20] to accept a subset of the data and yet can achieve a good registration results.

Kittler [8] described a method of matching and recognizing aerial road network images based on road network models. He used a list of line segments of images

obtained from a preprocessing stage as an input, the output is the correspondences between the image line segments and the model line segments. Kittler used attributed relational graphs (ARG) to describe the model and the scene . An ARG consists of a set of nodes, each node representing a line segment, and attributed relations between nodes. The task of matching was to find the best correspondences between the model and the scene. The correspondences were found by using relaxation labeling algorithm which optimized a criterion of similarity. This method was reported to match subgraphs of the image road structure to a map road model covering an area 10 times larger than the area imaged by the sensor. However, the image distortion due to perspective imaging geometry must corrected in a preprocessing stage.

Gaves and Quegen [9] used template matching to match map features to Synthetic Aperture Radar (SAR) images. The performance of this method using cross-correlation (which has no knowledge of speckle) was compared with a correlation measure, based on Klomogorov-Smirnov test of fit, which took speckle into consideration. The experimental results showed good performance using their method. However, neither of these approaches performed well on real data. This was due to the exponential speckle model being inappropriate for linear features in real data.

2.2 Conclusion

This section presented some of the systems that deal with the matching and registration of images in the presence of geometrical deformations. Only a representative samples were reviewed, since this problem is very popular and some of the other solutions are not very different from those reviewed here. The computers used vary from small PCs to powerful vector and massively parallel computers, so no conclusive

comparisons can be drawn between the various performance parameters listed.

Chapter 3

Image Analysis

3.1 Introduction

This chapter describes the techniques used in scene analysis, image enhancement and boundary extraction. These techniques extract meaningful information as a set of simple primitives, then describe it according to the features of those primitives and the relationships between them. One way of performing this task is by using grammatical syntax that consists of a set of rules defining the allowed relationships in grammatical form. Unfortunately, successful applications of grammatical scene analysis is very limited, and often generating a grammatical syntax to describe 2-D and 3-D scenes is difficult, when shapes get more complicated. The other way is using non-grammatical scene analysis which concentrates on 3-D scenes that consist of polyhedral objects. Describing those scenes consists of examining the properties the relationships between the edges, polygons and vertices of different shapes composing the scene. Complex shapes have been successfully interpreted in this way.

However, two major difficulties are encountered in designing a framework for analyzing the aerial images using this approach. The first difficulty is encountered in remote sensing applications, because the images are taken from a moving source (satellite, aircraft, etc.) where there might be substantial variations in the environmental illumination. Such situations may result in low-contrast images, specular reflections, shadows due to clouds etc. The second difficulty results from the amount of information contained in the images received. For instance, in order to process one image of 2000 × 2000 pixels, over 40 MB of memory is needed. This raises the need to extract only the useful information from the image. For these reasons, the following initial steps must be performed:

- Defining and generating for each object in the image, a set of primitive features that provide a significant and reliable image description.
- Converting the gray scale image information into a binary image.

The remainder of this chapter is structured as follows: Section 3.2 states the different processing steps involved in shape analysis procedure and describes the order in which they are executed. Section 3.4 includes the data processing necessary for primitive generation.

3.2 Object Shape Analysis

Object shape analysis involves two steps: shape description and shape recognition. First, we require a convenient description using the gray scale image. This is referred to as primitive feature generation. Three main approaches are known for shape description[47]:

1. Scaler transform techniques.

2. Global techniques.

3. Local techniques.

Scaler transform techniques

This class of techniques is based on the evaluation of some scaler measurements such as moments from the brightness function of the gray scale image. However, sometimes it is difficult to relate higher order moments to shapes[47].

Global techniques

This class of techniques examines both the interior and the boundary of the objects in the image. Morphological operations are amongst the most popular of the global techniques. They are based on the thinning of the object using iterative hit-and-miss transformation until the skeleton is obtained. The main drawback in these operations is that the computational effort is fairly complex[40] and a small amount of noise can severely alter the form of the skeleton[41]. Moreover, the execution time is proportional to the object area, and in some cases even to its square[39].

Local techniques

This class deals with the representation of objects in the images either in the frequency domain or in the spatial domain. The frequency domain boundary representation consists of computing the Fourier transformation of the boundary. The boundary can be expressed as the complex function $x(t) + jy(t)$ where $x(t)$ and $y(t)$ are the coordinates of the boundary point and t is the length parameter. The Fourier transform of the contour r will be the complex function $X(s)+jY(S)$ where X and Y are the Fourier transform of x, y. The major disadvantage is the difficulty to describe local information[47]. On the other hand describing images in the spatial

domain is more suited for aerial image analysis, since the type of deformation that is usually encountered is geometrical deformation.
When using spatial domain representation two situations may be encountered:

- The boundary is a collection of straight segments
- The boundary has curved edges

3.3 Primitives

Primitives are measurements that are made in an image or a region of the image. Since our main concern is the shape of different objects in the image we will compare measurements of the geometry of these objects.

1. Elementary primitives: each primitive is a pixel and its information content is low. Such primitives have been used in temporal tracking[19] to ensure highly accurate matching.
2. Combined primitives: these primitives are associations of pixels, windows of given size (e.g 5×5). They carry more local information than the previous ones.
3. Complex primitives: these are a composition of shapes that can be characterized using some geometric entities such as;
   - Vertices.
   - Lines.
   - Circular arc.

- Significant curvature points
- General Curved Boundaries.

Proper choice of primitives requires careful consideration. The more structured the primitive, the smaller the linkage problem. The result is faster matching and registration. For a given application, only primitives pertinent to the pre-registration stage must be chosen.

3.3.1 Feature Selection

The task of selecting the primitives used in the matching process involves the following steps:

- Eliminating the least useful information until a tractable set remains. Any of the primitives described in the previous section can be used depending upon the application and the nature of the image that consideration. For instance, linear segments and their intersections can be used as features to describe many man-made structures such as airports, roads and buildings in an aerial image. Such a description can then be used to match the scene with a map which contains similar descriptions [44].
- Selecting features that are invariant under rotation, translation and scaling. This means that the primitives used for shape description should be invariant under different types of uncorrected variations. Dominant points (Control points) along curves such as corners, intersections, inflection points, points of high curvature and points along discontinuity[32] are some examples of features that satisfy such criteria.

3.4 Feature Extraction

At this stage, two types of primitives are chosen as features to be extracted from the aerial image. The first feature is the line contours that compose the image. The second feature is the intersection between these line segments. Figure 3.1 shows an example of roads and intersections as seen by aerial imaging system. In sections 3.4.3 and 3.4.4 we describe in detail the steps performed in order to extract these features.

3.4.1 Boundary Extraction

The simplest procedure to extract a distinct geometrical features such as lines or regions in an image is to separate them from their background. This can be done directly using thresholding techniques. If the background lighting is fairly uniform, separation can be achieved by simply thresholding the image at a particular intensity level. Two procedures can be applied to convert the image into a binary form:

1. For images that have high contrast, threshold can be achieved by choosing a global threshold value. However, in most of the aerial images the lighting conditions are not uniformly distributed.

2. A Local enhancement is used to improve the image quality [42], in order to deal with the variation in lighting conditions and image granularity. This is followed by an adaptive thresholding algorithm to complete the process [42].

Figure 3.1: Roads as seen by SPOT Image

3.4.2 Image Enhancement

The objective of image enhancement is to change the image to highlight the features of interest. In most cases, this implies improving the visual appearance of the image. Instead of using histograms, local enhancement is used to improve the image quality[42]. The input gray scale image was enhanced using a local transformation based on the mean and variance of the intensity maps and the intensity of the input image $f(x,y)$ to create a new image $g(x,y)$ by performing the following transformation at each pixel location (x,y);

$$g(x,y) = A(x,y).[f(x,y) - m(x,y)] + m(x,y) \tag{3.1}$$

where

$$A(x,y) = k.\frac{M}{\sigma(x,y)} \tag{3.2}$$

where $m(x,y)$ and $\sigma(x,y)$ are the gray-level mean and standard deviation computed in a neighborhood centered at (x,y), M is the global mean of $f(x,y)$, and k is a constant in the range. Application to the local gain factor $A(x,y)$ to the difference between $f(x,y)$ and the local mean amplifies local variations. Since $A(x,y)$ is inversely proportional to the standard deviation of the intensity, regions with low contrast receive larger gain. Experiments showed that adding a fraction of the local mean and restricting the variations of $A(x,y)$ between two limits A_{min},A_{max} is often desirable in order to balance large excursions of intensity in isolated regions[42].In Figure 3.2 and Figure 3.3 we compare the edge image before and after local enhancement and the histogram in both cases.

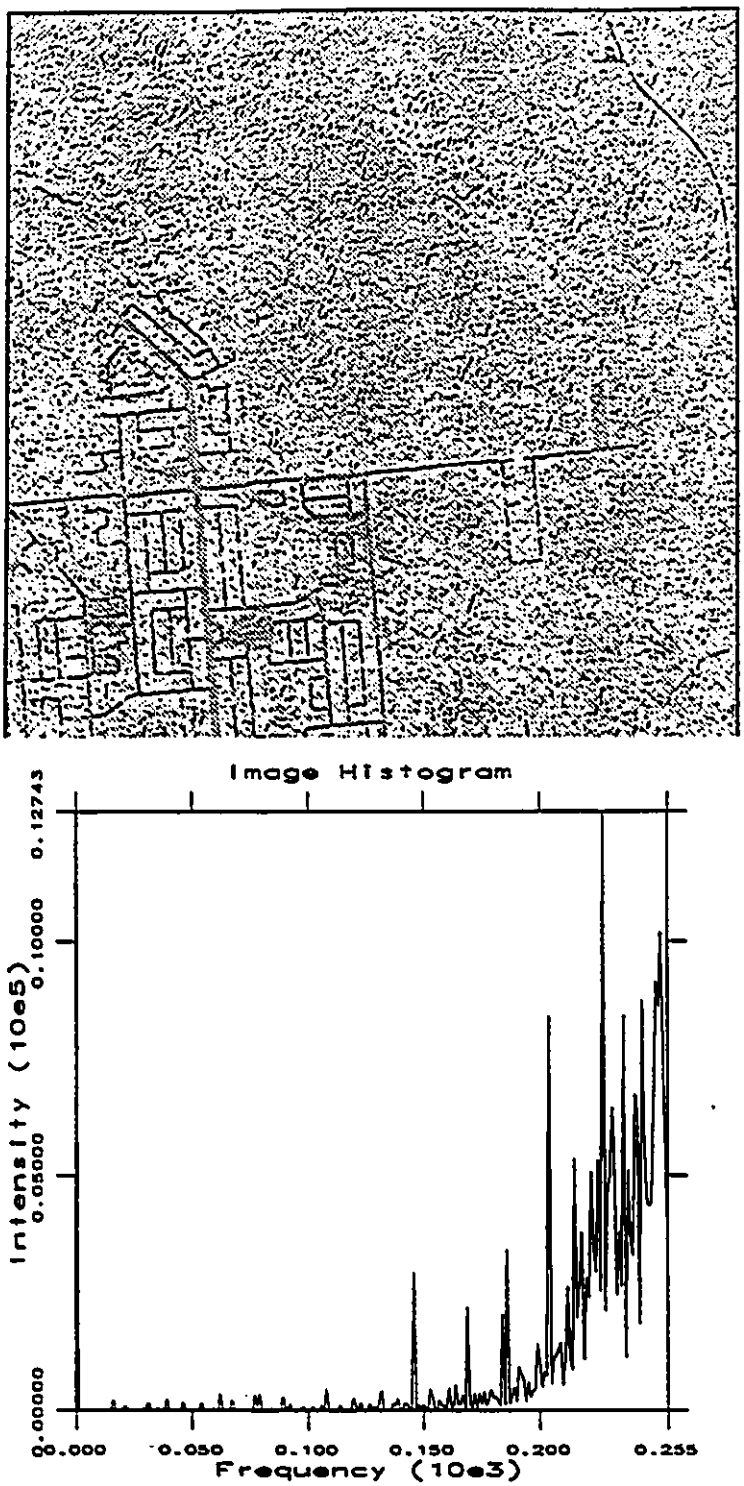


Figure 3.2: Edge image before local enhancement.

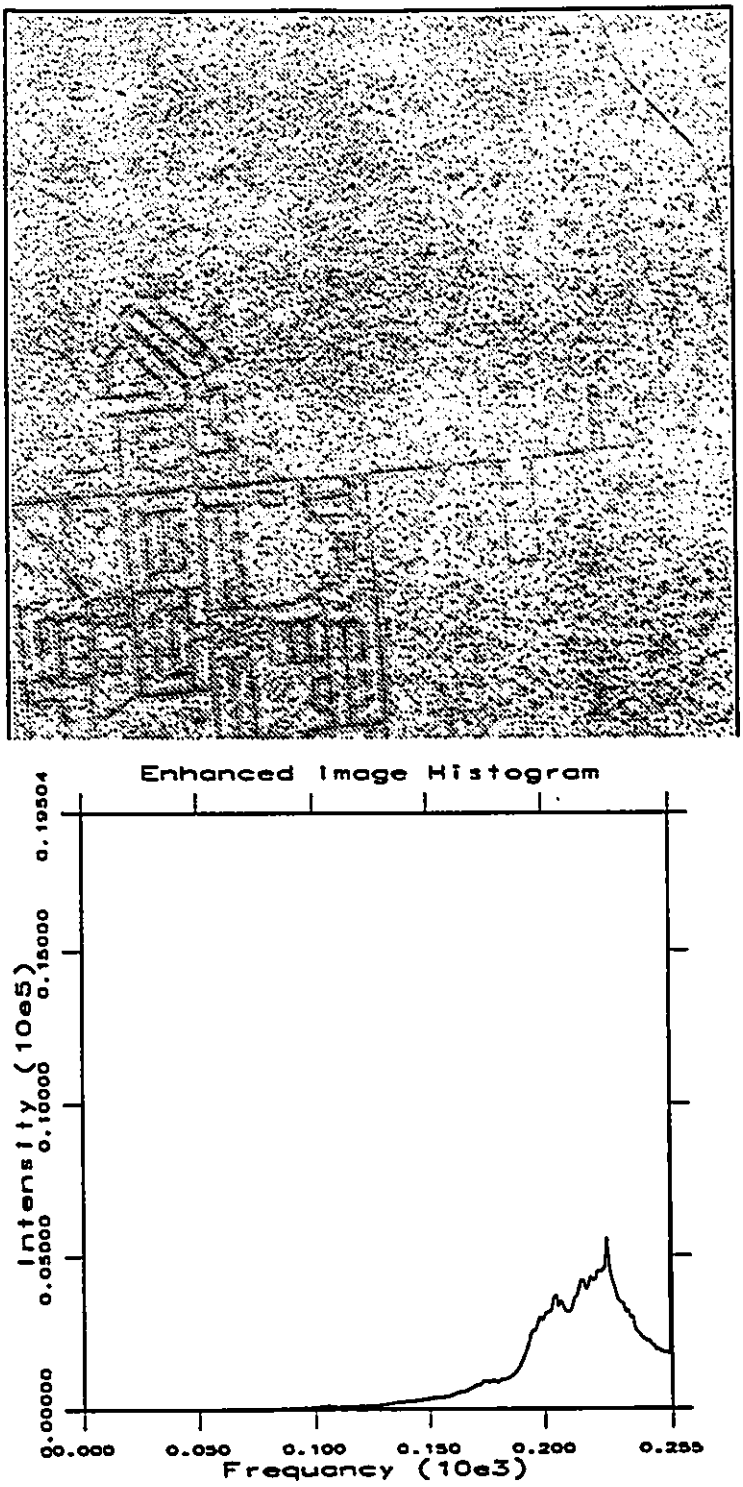
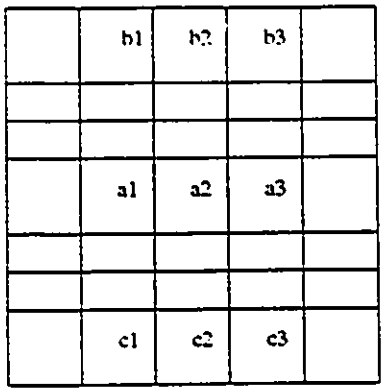


Figure 3.3: Edge image after local enhancement.

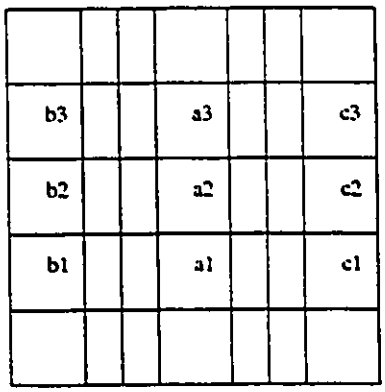
3.4.3 Road Extraction

Roads are linear structures of essentially constant width which, in general, are locally constant in intensity in the along-track direction, while showing significant contrast with adjacent terrain. Several methods were used to detect linear features in aerial images. Zhou and Chellapa [21] used autoregressive model to detect linear features via their edges. Others [43], [30] used generic edge detector to extract linear features. However, experiments showed [48] that such generic edge detectors are time consuming, and the output image contains unneccessary redundant information.
A specific interpretation of this road model is embodied in the Duda Road Operator (DRO). This operator was suggested originally by R. O Duda. Fischler et al. [48] has compared this operator with other edge detectors for road detection[45],[20]. The DRO is a filter designed to delineate the road center. It offers many desirable features over edge detectors. A group of 4 masks is used in order to detect the roads in the image. The masks labeled (1) and (2) as shown in Figure 3.4 measure the uniformity of intensity along a potential road track (a_1, a_2, a_3), and the contrast of this potential track with the adjacent terrain $(b_1, b_2, b_3; c_1, c_2, c_3)$. The terrain intensity is sampled at one-pixel offset from the assumed track to allow for slight variations in road width. This best reflects the road width for $SPOT - PLA$ data at 6.5 meter resolution, and ignores the highly variable mixed pixel adjacent to the roadway. The corresponding horizontal template is a simple 90° rotation. A scoring function $G(u)$ is designed to ignore slight variations in intensity along the road and to penalize all significant variations equally. In a similar fashion, another scoring function $F(u)$ rewards all large contrasts equally.

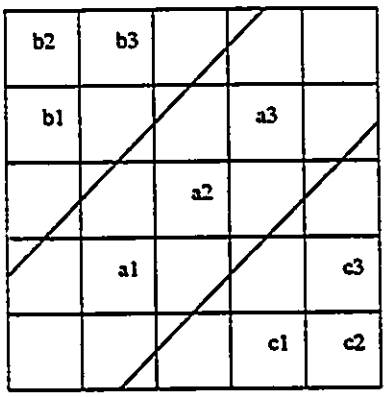
Figure 3.5 shows the function $F(u)$ depicted as a solid line is used for roads with low intensities. $F(-u)$ is used for roads with high intensities. A Symmetric



Mask to detect Horizontal line segments

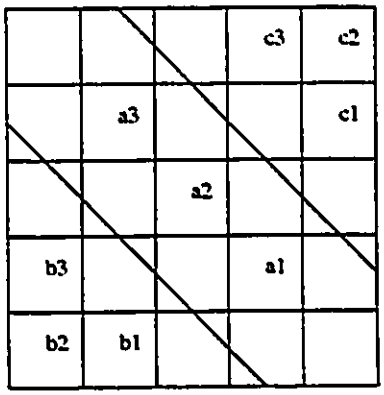


Mask to detect Vertical line segments



Mask to detect right diagonal line segments

(1)



Mask to detect left diagonal line segments

(2)

Figure 3.4: Templates for Duda Road Operator. (1) Mask to detect horizontal Road segments. (2) Mask to detect diagonal Road segments.

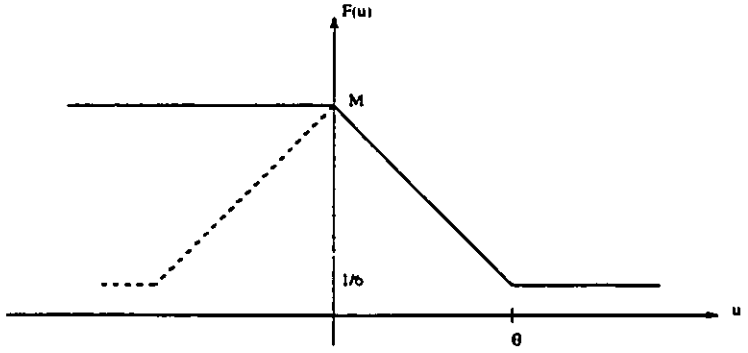


Figure 3.5: Road Edge scoring function

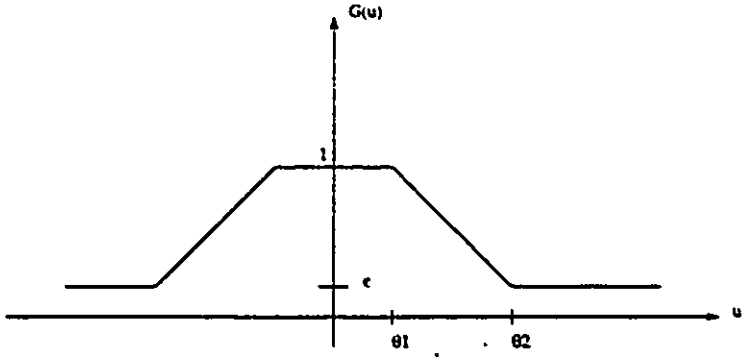


Figure 3.6: Road Uniformity scoring function

form of the function, shown by the dashed lines for negative values of u, is used when the road background contrast is unknown.

$$Score = \mid G(\mid a_1 - a_2 \mid) \times G(\mid a_2 - a_3 \mid) \sum_{i=1}^{3} F(a_i - b_i) + F(a_i - c_i) \mid \qquad (3.3)$$

where

$$G(u) = \begin{cases} 1 & 0 \leq u \leq \theta_1 \\ 1 - (1-\epsilon)\frac{u-\theta_1}{\theta_2-\theta_1} & \theta_1 \leq u < \theta_2 \\ \epsilon & u > \theta_2 \end{cases} \qquad (3.4)$$

$$F(u) = \begin{cases} M & u < 0 \\ M - \frac{M - \frac{1}{6}u}{\theta} & 0 \leq u < \theta \\ \frac{1}{6} & u > \theta \end{cases} \tag{3.5}$$

3.4.4 Corner Extraction

In spite of the robustness of the spline modeling of the image, control points may not correspond to corners on the real objects in the image, due to segmentation errors that may occur [47]. In order to obtain the real corners and intersections more processing of the boundaries is necessary. Several corner extraction methods are known. They can be broadly divided into two groups;
The first group works directly on grey-scale images. This includes template matching procedure proposed by Bretschi [7]. In this method a set of 8 templates being generated by successive 90° rotations of the first two shown.

$$\begin{bmatrix} -4 & 5 & 5 \\ -4 & 5 & 5 \\ -4 & -4 & -4 \end{bmatrix} \qquad \begin{bmatrix} 5 & 5 & 5 \\ -4 & 5 & -4 \\ -4 & -4 & -4 \end{bmatrix}$$

Ideally, this set of templates should be able to locate all corners and estimate their orientation within 22.5°. Unfortunately, corners vary to a great degree in a number of their characteristics, including in particular their degree of pointedness, internal angle and the intensity gradient at the boundary.
The second group consists of first extracting edges and searching for points having maximum curvature [45],[13] or performing piecewise linear approximation on the line segments and then searching for the line segment corners[21]. We adopted this approach because it is easy to implement and has good noise immunity. Define L_{is}

as the straight line segment that contains s chain links and ending on the node to which the link $\vec{c}_i$ is directed. The x and y components of L_{is} are given by:

$$X_{is} = \sum_{j=i-s+1}^{i} c_{jx} \tag{3.6}$$

$$Y_{is} = \sum_{j=i-s+1}^{i} c_{iy} \tag{3.7}$$

where c_{ix} is the x and y components of the segment link c_i. The angle that L_{is} makes with the $X - axis$ is given by

$$\theta_{is} = tan^{-1}(\frac{Y_{is}}{X_{is}}) \tag{3.8}$$

The incremental curvature δ_{is} is defined as twice the mean over two adjacent angular differences

$$\delta_{is} = \theta_{i+1,s} - \theta_{i-1,s} \tag{3.9}$$

s ranges from $5 - 13$ depending on the quantization error [10]. Davis [10] characterized a corner by three incremental curvature regions - two for which δ_{is} fluctuates within relatively narrow limits due to quantization noise, separated by a third region consisting of $s + 1$ nodes in which $\sum \delta_{is}$ equals a significant value. He defined the "cornerity" at the node i by

$$K_i = \sqrt{t_{i1}} \times \sum_{j=i}^{i+s} \delta_{js} \times \sqrt{t_{i2}} \tag{3.10}$$

where

$$t_{i1} = max(t : \delta_{i-v,s} \in (-\Delta, \Delta)), \qquad for 1 \leq v \leq t$$

$$t_{i2} = max(t : \delta_{i+s+v,s} \in (-\Delta, \Delta)), \qquad for 1 \leq v \leq t$$

and

$$\Delta = \frac{1}{s-1}$$

By scanning all line segments, all points p_i where $|K_i| \geq |K_j|$ for all j such that

$$|i - j| < s$$

as corner points.

3.5 Conclusion

In this chapter we have described the theory behind the techniques used to choose and extract linear features features and corners from an image. This features are used by the algorithm described in the next chapter in the image modeling. The next chapter reviews some image deformation types, algorithms to correct this kind of deformations and the algorithm we used to model aerial images subject to geometrical deformation. We also explain the usefulness of using such an algorithm.

Chapter 4

Image modeling

4.1 Introduction

In this chapter, some aspects of aerial image deformation are discussed then, an algorithm for modeling the images and the extracted features from chapter 3 is described. The merits of modeling the images using this algorithm is also discussed.

4.2 Geometric Deformation

There are numerous sources of geometric distortion in aerial images. These can be categorized into three groups;

1. **Spatial Distortion**

   These are distortions that can be modeled. These distortions are geometrically.

due to viewpoint change and sensor noise.

2. **Uncorrected Distortions**

   These are distortions that are difficult to model. They are often dependent on the scene and are often volumetric. Typical examples are lighting and atmospheric variations, shadows and volumetric sensor noise.

3. **Variations of interest**

   These are differences between two images which we would like to detect. Common examples include changes in the scene such as object movements, growths or deformations, and difference in sensor measurements when using sensors with varying sensitivities.

All variations can be further classified as either static/dynamic, internal/external and geometric/photometric [5],[8]. Some of these are well characterized, through understanding the viewing geometry. Examples include image skewing due to Earth rotation. Other causes of distortion are a more transient and random nature such as lighting and atmospheric changes. These are difficult to model because they depend on the characteristics of the physical objects in the scene.

Since image matching and registration deal with the removal of distortions and the detection of changes between images, it is important that the images are matched only with regards to the miss-registration source. Otherwise the changes of interest will be removed at the same time. Registration and matching of dissimilar images often has a special need to model the miss-registration source. In hierarchical search techniques described by Hall [26], for example, matching rules are selected which are more invariant to natural changes in scenery. In general, matching and registration of images obtained at different times or under different scene conditions is performed to extract changes in the scene.

4.3 Image Modeling With B-splines

B-splines were investigated as early as the nineteenth century by N. Lobachevsky (for more details see [36]). They were constructed as convolutions of certain probability distributions. Later, Schoenberg [31] used the B-splines for statistical data smoothing, and his paper started the modern theory of spline approximation. B-spline possesses some properties that makes it suitable for image modeling in general and the aerial image in particular.

- The B-spline is characterized by a set of parameters (control points) from which the curve can be totally generated.
- The B-spline is invariant under affine transformations. This means that the projected object curve onto the image space is a B-spline whose control points are related to the object control points through affine transformation.
- Any part of the image can be changed or modified independently of the rest of the image. This makes it easier to correct local distortions deformations in parts of the image without having to disturb the whole image.

4.4 Uniform B-spline Approximation

A digital boundary can be approximated with arbitrary accuracy by a line segment. For any curve, the approximation is exact when the number of segments is equal to the number of points in the boundary so that each pair of adjacent points defines a segment in the curve. In practice, the goal of spline approximation is to capture the "essence" of the boundary shape with the fewest possible control points [47]. A

common function shared by most spline approximation algorithms is the collinearity test that checks if points along a boundary portion are collinear with respect to a straight line. Collinearity is usually determined by the maximum perpendicular distance from a point of the boundary portion to the straight line. Consider a boundary portion between two points, A and C (see Figure 4.1), we compute the maximum perpendicular distance from the boundary portion to the straight line AC. If the distance is within tolerance, that boundary portion is approximated by the straight line segment AC. Otherwise, the point along the boundary portion that yields the maximum distance becomes a new break point, say B, and the boundary portion is approximated by two line segments AB and BC. The algorithm used to extract spline approximations is the uniform cubic B-spline algorithm.

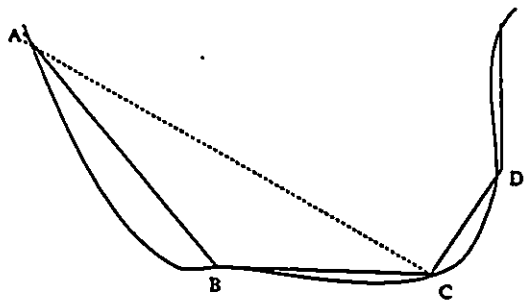


Figure 4.1: spline approximation concept.

The uniform B-spline are piecewise polynomial functions that provide local approximations to line contours using a small number of parameters(control points). These functions can represent both opened and closed curves that are C_{k-1} continuous. In cases of aerial images, both closed curves (regions) and open curves (open contours) can be used. However, in this thesis, since the extracted features are linear (roads, rivers, etc.,) only open curves will be modeled.

Traditional implementations of the uniform cubic B-spline algorithm [16], [4] are time consuming due to the following factors:

- If the initial position is chosen arbitrarily the number of control points will be large (worst case equals to the number of data points).
- The error norms used, either the maximum Euclidean distance or the integral square error, require extensive computation time.

In order to remedy the first problem, a rough, but very fast scan-along polygonal approximation [50] precedes the cubic B-spline algorithm. Vertices resulting from this scan along procedure serve as initial partition for the cubic B-spline algorithm. The second drawback is addressed by using area deviation per length as error norm [38]. This norm originally used for the scan along technique allows the cubic B-spline algorithm to have a better execution time.

A brief outline of the resulting cubic B-spline algorithm is given below:

1. Assign an arbitrary number of points along the boundary as the initial set of break points (called knots). The initial approximated lines is formed by joining the sequence of knots along the original boundary with straight lines. These points are determined from equation 4.1 :

$$t_i = \begin{cases} 0 & i \leq k \\ i-k+1 & k \leq i \leq n \\ n-k+2 & u > n \end{cases} \tag{4.1}$$

2. After generating the sequence of knots, the control points that generate the shape of the curves [4] are generated from the equation

$$P(u) = \sum_{i=0}^{n-1} V(i) B_{i,r}(u) \tag{4.2}$$

where $P(u) = [x(u), y(u)]$ is the continuous curve model composed of coupled pairs of x and y coordinate functions, $V(i) = [v_x(i), v_y(i)]$ is the ith control point and $B_{i,r}(u)$ is the rth order normalized B-spline basis function. For open curves, the normalized B-spline associated with the knots generated from the equations in 4.1 are given by the matrix in [16];

$$B_{i,r}(u) = \begin{pmatrix} 2 & 1 & & & \\ & 1 & 4 & 1 & \\ & & \vdots & & \\ & & 1 & 4 & 1 \\ & & & 1 & 2 \end{pmatrix} \tag{4.3}$$

Figure 4.2 shows a curve and the associated control points.

3. For each pair of adjacent control points, determine the points along the boundary portion that yield the maximum perpendicular distance to the straight line segment joined by the two break points. If the maximum perpendicular distance is greater than a given tolerance, that point becomes a new break point; i.e., the line segment is replaced by two line segments.

4. For each three consecutive adjacent points, say A, B, and C, compute the maximum perpendicular distance from the boundary portion between A and C to line AC. If the distance is within tolerance, knot B is removed. That is, line segments AB and BC are replaced by line segment AC. Note that each replacement is immediately tested for joining with the next line segment.

5. Repeat Steps (3) and (4) until an equilibrium is reached.

The number of points and their locations along the boundary varies if the boundary is at a different orientation or a different scale. To make the points obtained insensitive to the change of orientation and scale, an extra processing step is needed.

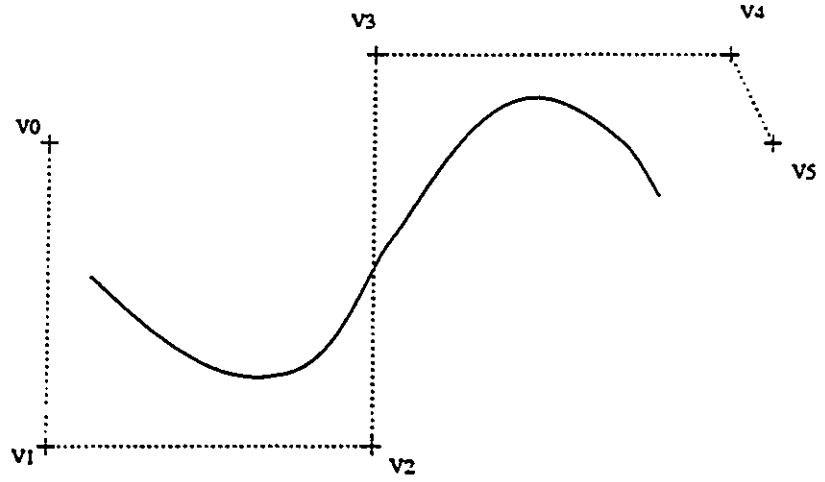


Figure 4.2: A curve defined by a sequence of control vertices, represented by the "+" sign, near which the curve passes. The dotted line connecting the control vertices form the control polygon.

4.4.1 Control point adjustment

Control point adjustment, also called piecewise approximation with variable joints or chord length is used to optimize the control points positions so that they are as close as possible to the real points on the curve. The problem of control points adjustment can be formulated as follows: given a contour P and integer number n, find the optimal position of each one of n points V_{i1}^{n} along P such that the n segment spline joining thesis points approximated the contour P with minimum error. Depending on the type of error norm employed, several techniques can be used for control point adjustment. When the maximum point-wise error norm (E_∞) is employed, control point adjustment comes down to a mathematical programming problem involving the minimization of a linear function which is subject to both linear and nonlinear quadratic constraints. Some dynamic programming methods have been proposed to solve this case [13]. On the other hand, control point adjustment in the case of the integral square error (E_2), is an analytical minimization problem [39]. Pavlidis in

[38] applied the Newton-Raphson method to solve this problem.
Both the mathematical programming and the minimization techniques are optimal. However, they are time consuming [13]. To reduce the computational time problem, Pavlidis proposed a fast control point adjustment method [41], it works for both, E_∞ and the (E_2) error norms.

This method finds the optimal position for the n control points by treating each control point separately. The algorithm keeps turning around the boundary from one control point to another. The position of each traversed control point is adjusted in order to minimize the total error norm. The algorithms keeps cycling around the boundary of each contour until no more control points adjustment can be made. During the adjustment procedure of the control point V_i, only the position of V_i will change while all the other $(n-1)$ control points remains fixed. Therefore only two segments V_{i-1}, V_i and V_i, V_{i+1} are affected by the control point adjustment of V_i and hence the only segment error norm which may be modified are E_i and E_{i+1}. When the algorithm converges $\frac{\partial E}{\partial V_i}$ is equal to zero for all V_i's. Therefore, the gradient of E is null and E reaches its minimum. However, this minimum can be local and the control point adjustment algorithm may generate suboptimal approximation [9].

4.5 Conclusion

For shape modeling, contour control points are desirable because they influence the contour model shape locally. In particular, the control point models based on B-spline are robust, since the knot selection, spline order and the spline model order can be easily adjusted. Furthermore, due to the invariance properties of the spline, the control point sequence of similar images with different geometrical deformation is equivalent to identical shape model transformation in the same order.

Chapter 5

Image Matching

Matching is the process of finding a correspondence between two images. It is used as a basic step for image registration and recognition. For aerial images the matching process is needed for a variety of applications such as change detection, map updating and image based navigation (for instance, two images are taken at different times of the day, in different seasons, and possibly sufficiently separated temporally that significant changes on the ground have taken place). Therefore, a robust matching procedure is needed in order to detect these changes.

5.1 Introduction

The previous chapter dealt with techniques used in the modeling of images, the derivation of the spline model, and finally the generation of the control points. In this chapter, a matching procedure is developed to perceive the correspondence between two images that are exposed to geometrical distortion. For that purpose, in section

5.5, features that were extracted from the images will be used in finding the initial correspondence, then in section 5.5.2, initial transformation will be calculated. In section 5.6, the parameter estimation equations will be derived and the registration function that aligned the two mismatched images will be computed.

5.2 Geometric Aspect of Image Formation

In most applications of remotely sensed imagery it is necessary to relate the image to other geocoded data such as maps or other reference images. Images acquired by LANDSAT for example, exhibit significant distortions so they cannot be used as references. These distortions can be categorized into two main groups. The first group is the spatial distortion which can be satisfactorily modeled usually by geometric transformation. This is called corrected distortions. The second group is volumetric distortion which can be corrected but it is difficult to model mathematically. Such distortion can be caused by lighting and atmospheric variations, shadows, and volumetric sensor noise. This type of distortion is beyond the scope of this thesis and will not be discussed. The distortion induced by geometric errors are corrected in two phases. In the first phase, the effect of the systematic errors are compensated (examples of systematic errors are earth rotation, mirror velocity and map projection). These errors are well understood and their effect can be modeled and removed by appropriate geometric transformation. A more complete discussion of the various distortions in remote sensing images is given by Bernstein [5] and Van Wie and Stein [51]. In the second phase, the remaining distortions are assumed to be the result of some random or unknown errors. Bernstein in [5] estimated this error from the location of ground control points (GCP). A ground control point is a physical point with a known location that is artificially inserted in the image.

Once all systematic distortions have been corrected, the residual distortion left are due to the random positional errors. Differences between the actual and the GCP locations can then be used to model this errors. The final result is then a complete transformation function that maps a pixel at (x, y) in the image into a point (u, v) in the other reference image or map. However, it is not always possible to use ground control points in the images. For this reason, a different approach is used, whereby, instead of using points that are artificially inserted in the image, some features that are detectable in the image and whose position are invariant under geometric transformation will be used. As explained earlier in chapter 3 in selecting features, corners are extracted as they maintain such properties.

5.2.1 Invariant Transformation

The underlying assumption of any matching algorithm is that there exists a mapping between the two sets of features which can be characterized by some invariant property. This raises the question of how to determine the position of an object visible in a single image. Solving the inverse perspective problem requires in addition to the image, supplementary information about the object because the perspective projective transformation is non invertible. Two situations may occur if information about the object position or the object shape are available: the first case is much easier to deal with and has a straightforward solution. In general the *a priori* information of the object position can be formulated into a linear or non linear constraints. A unique solution in that case exists. An example of that is described in [6]. The second case, is the case of the model based 3D position estimation, the object recognition and its reference to a specified model is required. Considering the *model* and the *scene* of the object, the position is determined by computing the perspective deformations the scene has been subject to. Unlike the first method,

the position of the object is determined indirectly using a least squares estimation. This case is discussed thoroughly in the next section.

The case of image registration using a model of the image is discussed in this section. Considering the model and the scene in the image, the registration parameters are determined by computing the geometric deformation of the scene. This method is indirect and uses least squares estimation. In section 5.3 we review the existing literature related to the subject. Section 5.5.2 presents the determination of initial matches and the constraints imposed on selecting the initial candidates for the matching process. Section 5.6 deals with the numerical solution to the registration parameters.

5.3 Model-Based Parameter Estimation

Model-based parameter estimation is a problem of fitting a model to experimental data (the model image). In this case the data interpretation process is divided into two tasks;

1. find the best match between the data in the image model and scene (classification problem).
2. compute the best values of parameters for the selected model (the parameter estimation problem).

The latter is determined by estimating the set of parameters which minimizes the squared distance between the set of feature points and the set of transformed points [6]. Generally speaking, these parameters consist of translation, scaling and rotational parameters.

According to Besl and McKay in [6], if d_i is the distance between the control points (x_i, y_i) and the transformed points (u_i, v_i) then we want to minimize:

$$\sum_{i=1}^{n} d_i^2 = \sum_{i=1}^{n}[(x_i - u_i)^2 + (y_i - v_i)^2] \tag{5.1}$$

where n is the number of control points. In general a global transformation that maps a set of points of one image to another set of points in the other image is described as

$$M_p = \alpha R S_r + T \tag{5.2}$$

R is the rotation component, T is the translation component and α is the scaling factor. Haralick and Shapiro in [27] defined these components as:

$$R = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.3}$$

the translation T is defined as;

$$T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{pmatrix} \tag{5.4}$$

and the scaling is given by

$$\alpha = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.5}$$

Therefore, the transformation H which maps the set of control points m into a set of points n is computed as follows; first a number of points in the scene and their corresponding points in the image model are matched. Then α, R and T are determined by minimizing the quadratic function in equation (5.1) which calculates the sum of the distances between the matched scene points and the image by the model scene transformation of their corresponding points in the model. In the next

section, the determination of the matched features will be described. Substituting equations (5.3),(5.4),(5.5) into equation (5.2), the transformation becomes

$$(u \quad v \quad 1) = (x \quad y \quad 1) \times \begin{pmatrix} \alpha\cos\theta & \alpha\sin\theta & 0 \\ -\alpha\sin\theta & \alpha\sin\theta & 0 \\ t_x & t_y & 1 \end{pmatrix} \tag{5.6}$$

or

$$\begin{aligned} u_i &= x_i\alpha\cos\theta - y_i\alpha\sin\theta + t_x \\ v_i &= x_i\alpha\sin\theta + y_i\alpha\cos\theta + t_x \end{aligned} \tag{5.7}$$

The above equation can be put in a general form

$$\begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ x_n \\ y_n \end{pmatrix} = \begin{pmatrix} u_1 & -v_1 & 1 & 0 \\ v_1 & u_1 & 0 & 1 \\ u_2 & -v_2 & 1 & 0 \\ v_2 & u_2 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ u_n & -v_n & 1 & 0 \\ v_n & u_n & 0 & 1 \end{pmatrix} \times \begin{pmatrix} \alpha\cos\theta \\ \alpha\cos\theta \\ t_x \\ t_y \end{pmatrix} \tag{5.8}$$

The model based parameter estimation involves two steps

1. model or scene feature matching
2. estimation of the model transformation by the least-squares minimization.

These two steps can be performed either successively or simultaneously. In the first case, the feature matching and the least-squares minimization steps are performed independently [34], [33], [15] and [37]. The above methods are based on classical techniques for parameter estimation [28], [1]. Therefore, they have no internal

mechanism to detect or reject gross errors. However, when a robust matching [52], [20], is employed, gross errors are rejected automatically [21]. In the second case least-squares minimization is performed as the matching is made. After each matching the new feature correspondence is taken into account to refine the pixel position. To initialize the algorithm, a minimum number of matching points are essential to permit a first estimation of the parameters. In [48] and [17] a study of the minimum number of point correspondences for parameter computation is carried out. Performing matching and least-squares minimization was first introduced by Fischler and Bolles [18]. The advantage of using a method called RANSAC (Random Sample Consensus) is that it permits the filtering of data which has a significant percentage of gross errors. The method presented by Faugeras et al., in [17] to solve the 2D parameter estimation problem is also based upon the same idea.

5.4 Model - Scene matching

The model-scene matching problem has been thoroughly studied during the last decade. Matching model and scene features is used considerably in model based object recognition. Since large number of publications is available in that subject, it is difficult to survey all existing techniques. In this thesis only some of those techniques will be reviewed.

Geometric hashing technique

This technique is based upon the principle that, assuming the linearity of the geometric transformation, the coordinates of the model feature (calculated in a frame related to the model) are equal to the coordinates of their corresponding scene features calculated in the corresponding scene frame [21]. This

principle permits the determination of the model and scene feature correspondences. This method is fast and easy to implement. However, if the image has rotational deformation this method can not be applied [53].

Distance measure technique

This technique is used as a matching segment technique. It calculates an error norm (or distance measure) between each scene segment and the corresponding segment in the model [37].

Graph Matching

The graph matching technique [10] consists of dividing the model contour into subparts. Using local evaluation function, for each element in the template, a subset of object elements that the template element may be associated with, is determined. Each association of a template element with a single matching object will correspond to a node in the association graph. Graph matching techniques have been widely used in model based pattern recognition, because of their suitability for programming and their structural nature that gives them position and orientation independence. Some graph matching algorithms are probabilistic [14].

Template matching

This is another class of methods which are widely used for matching images. Normally, no detailed radiometric information can be deduced from a given map feature about an other feature in a corresponding image. Template matching can, however, often predict that certain type of features can be detected (for example high curvature edges and line segments). These features are suitable candidates for matching. When a template is correlated with the image

the position of the feature is established where the maximum correlation measure occurs. The exact manner in which partial matching can affect feature detect-ability depends on the correlation measure being used.

5.5 Image Correspondence

As described earlier in chapters 3 and 4, primitives are defined and the image is represented and modeled, we can formulate the problem of image correspondence as: Which primitives extracted from one image correspond to which primitives in the other image, based on geometrical constraints. The approach to solve this problem is divided into two steps:

1. Find an initial correspondence between some distinct points in the scene and the model.
2. Calculate an initial transformation H based on this initial match, and use it as a starting point registration transformation between the scene and the image.

Major similarity measures that are used to find correspondence between images are the sum of absolute differences and the cross-correlation coefficient. The sum of the absolute differences is computationally fast[46], but the correlation coefficient measure is more accurate[29]. However, if the images are rotated with respect to each other, no similarity measure can be used. This is because, even though the centers of two windows correspond to each other, other points in the window will not correspond to each other, and low similarity measure might be obtained. As a matter a fact, similarity measure is not the only problem. When the two images are rotated,

it is impossible for two rectangular windows to contain the same part of the scene (except when two windows are rotated by multiple of 90° with respect to each other). To overcome this problem, a circular window is used. To perform the first step, we used the correlation coefficient technique, in which the location and orientation of a template or pattern is found in the image. This statistical measure has the property of measuring the correlation on an absolute scale ranging from $[-1, 1]$. The value measured by the correlation coefficient gives an indication of similarity between the image and the scene. This is useful in order to quantitatively measure the confidence or reliability in a match and to reduce the number of measurements needed when a pre-specified confidence is sufficient [24].

5.5.1 Assumptions and Constraints

For the geometrical constraints imposed, the following relations between the objects in the image and the scene are assumed;

- The Scaling factor from one image to the other, may not be known.
- It is possible for object a_i to have no possible model l_i
- It is possible for several objects to share a common model l_i
- Since no constraints is imposed on the geometry of the images i.e the image can be rotated and translated, the matching algorithm must be invariant under this geometrical transformation
- The images are assumed to have Gaussian noise of zero mean.

5.5.2 Determination of initial matching

The task of obtaining candidates for initial matching can be stated as follows: we are given two sets of points in the plane. The first set contains m points. The second set is similar to the first one, except that some of the points from the first set may be missing and some new points are not present in the first set. The second set contains n points. The primitives are denoted in one image by a_i where $i = 1, \cdots, m$ and call them objects. The primitives in the other image are denoted by l_j where $j = 1, \cdots, n$ and call them models.

To find the initial matching between points from the image and the scene, the correlation coefficient is used [42]. This gives an indication of how close the correspondence between the object a_i to the model l_j.

As mentioned earlier, corners will be used as initial points for matching since the number of corners in any given image is reasonably lower than the number of line segments. The initial matching is performed using a correlation window of a circular shape of diameter size $(2n + 1)$ centered at the point m_1 and by using a rectangular search area of size $(2du + 1) \times (2dv + 1)$ around this point in the second image. A correlation operation is then performed on a given window between point m_1 in the first image and all extracted points in the second image. This search window can be reduced only if *a priori* knowledge about the disparities between the matched points is known. Otherwise the whole image must be scanned to find the matched points. Figure 5.1, shows a diagram of a search window and mask used to find the initial matching points.

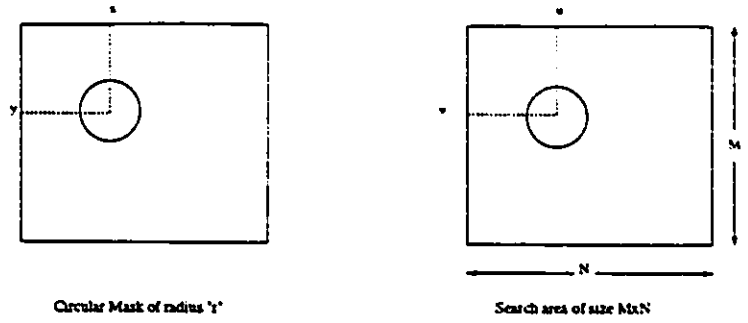


Figure 5.1: Correlation-coefficient using circular window

5.5.3 Invariant Moments

In order to satisfy the conditions assumed in section 5.5.1, moments are used to compute the initial matching as they possess the property of being invariant under geometric transformation such as rotation, translation and scaling.

The two-dimensional moment m of a digital image is defined by

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y) \tag{5.9}$$

where m_{pq} is $(p+q)th$ order moment of the image. The center of gravity (centroid) $(\bar{x},\bar{y})$ of f can be expressed in terms of the normalized moments M as;

$$\bar{x} = \frac{M_{10}(f)}{M_{00}(f)}$$
$$\bar{y} = \frac{M_{01}(f)}{M_{00}(f)}$$

Using the center of gravity, the central moment μ of order (p,q)

$$\mu_{pq} = \sum_x \sum_y = (x-\bar{x})^p (y-\bar{y})^q \tag{5.10}$$

is invariant with respect to the translation of the image f [29].

In general, more than one moment must be used in the decision making. This depends upon the constraints that are imposed on the matching procedure. One way

of measuring the similarity between the two windows is by computing the distance between two vectors of moments from the windows. The smaller the distance, the more similar the two windows [22]. This method, however, requires that the feature elements in the vectors be of the same scale. Moment of different orders do not have the same scale. The correlation between two sets of features is defined when the feature values in each set are of the same scale. When features of different scale are used in the computation of correlation, the feature with the largest scale will dominate the correlation value. To overcome this problem, the moments of all orders are normalized so that they would have the same scale. Hu [29] generalized the moment invariance to take into account the other geometrical transformations (rotation and scaling). To determine the similarity between two windows, eight normalized invariant moments from the two windows are used. These are: one zeroth-order, three second-order and four third-order normalized invariant moments. The normalized moment M of order $(p+q)$ for the images f with dimension $n \times n$ is defined as

$$M_{pq} = \sum_{x=1}^{n} \sum_{y=1}^{n} (\frac{x}{x'})^p (\frac{y}{y'})^q f(x,y) \tag{5.11}$$

where

$$\begin{aligned} x' &= \frac{1}{n} \sum_{x=1}^{n} x \\ &= \frac{n(n+1)}{2n} \\ &= \frac{n+1}{2} \end{aligned} \tag{5.12}$$

$$\begin{aligned} y' &= \frac{1}{n} \sum_{y=1}^{n} y \\ &= \frac{n+1}{2} \end{aligned} \tag{5.13}$$

Substituting into 5.11 we have

$$\begin{aligned} M_{pq} &= \sum_{x=1}^{n}\sum_{y=1}^{n}[\frac{x}{(n+1)/2}]^p[\frac{y}{(n+1)/2}]^q f(x,y) \\ &= [\frac{2}{(n+1)}]^{p+q}\sum_{x=1}^{n}\sum_{y=1}^{n}x^p y^q f(x,y) \\ &= [\frac{2}{(n+1)}]^{p+q}m_{pq} \end{aligned} \tag{5.14}$$

The above equation shows that normalized $(p+q)$th order moment M_{pq} is equal to the $(p+q)$ order moment multiplied by a scaling factor of $[\frac{2}{n+1}]^{p+q}$. To determine the normalized central moments, Hu in [29], replaced the term x by $\frac{x}{(n+1)/2}$ and y by $\frac{y}{(n+1)/2}$in equation 5.10.

$$\begin{aligned} u_{pq} &= \sum_{x=1}^{n}\sum_{y=1}^{n}[\frac{(x-\bar{x})}{(n+1)/2}]^p.[\frac{(y-\bar{y})}{(n+1)/2}]^q f(x,y) \\ &= [2/(n+1)]^{p+q}\sum_{x=1}^{n}\sum_{y=1}^{n}(x-\bar{x})^p(y-\bar{y})^q f(x,y) \\ &= [2/(n+1)]^{p+q}u_{p+q} \end{aligned}$$

Hu [29] also derived an expression for the second order and third order moments which are invariant with respect to the rotation and translation of image f;

$$a_1 = u_{20}+u_{02} \tag{5.15}$$

$$a_2 = (u_{20}+u_{02})^2+4u_{11}^2 \tag{5.16}$$

$$a_3 = (u_{30}+3u_{12})^2+(3u_{21}-u_{03})^2 \tag{5.17}$$

$$a_4 = (u_{30}+u_{12})^2+(u_{21}+u_{03})^2 \tag{5.18}$$

$$\begin{aligned} a_5 = {} & (u_{30}+3u_{12})(u_{30}+u_{12})[(u_{30}+u_{12})^2 \\ & -3(u_{21}+u_{30})^2] \\ & +(3u_{21}-u_{03})(u_{21}+u_{03})[3(u_{30}+u_{12})^2 \\ & -(u_{21}+u_{03})^2] \end{aligned} \tag{5.19}$$

$$
\begin{aligned}
a_6 &= (u_{20}-u_{02})[(u_{30}+u_{12})^2-(u_{21}+u_{03})^2] \\
&\quad 4u_11(u_{30}+u_{12})(u_{21}+u_{03}) \qquad (5.20)\\
a_7 &= (3u_{21}-u_{03})(u_{30}+u_{12})[(u_{30}+u_{12})^2 \\
&\quad -3(u_{21}+u_{03})^2] \\
&\quad -(u_{30}-3u_{12})(u_{21}+u_{30})[3(u_{30}+u_{12})^2 \\
&\quad -(u_{21}+u_{30})^2] \qquad (5.21)
\end{aligned}
$$

where $a_1, \cdots, a_7$ are the invariant moments. Thus the normalized invariant moments $A_1, \cdots, A_7$ are determined from the above equations [29]

$$A_0 = \sum_{x=1}^{n}\sum_{y=1}^{n} xy \qquad (5.22)$$

$$A_1 = [2/(n+1)]^2 a_1 \qquad (5.23)$$

$$A_2 = [2/(n+1)]^4 a_2 \qquad (5.24)$$

$$A_3 = [2/(n+1)]^6 a_3 \qquad (5.25)$$

$$A_4 = [2/(n+1)]^6 a_4 \qquad (5.26)$$

$$A_5 = [2/(n+1)]^{12} a_5 \qquad (5.27)$$

$$A_6 = [2/(n+1)]^8 a_6 \qquad (5.28)$$

$$A_7 = [2/(n+1)]^{12} a_7 \qquad (5.29)$$

By calculating a two sets of normalized invariant moments A and B from the windows in both images , then the similarity between the windows is determined by computing the cross-correlation factor:

$$r = \frac{\sum_{i=0}^{7} A_i B_i}{\sum_{i=0}^{7} A_i^2 \sum_{i=0}^{7} B_i^2} \qquad (5.30)$$

where A_i and B_i are the two sets of normalized invariant moments from the windows in the two images. The *Zero* order normalized invariant moment A_0 is equal to the sum of the intensity values in the window. The second and third order normalized

invariant moments are (A_1, A_2, A_3) and (A_4, A_5, A_6, A_7), respectively. The result of computing equation (5.30) ranges from -1 to 1 for two windows that are not similar in the model and the scene [6]. Thus, a global registration function is computed based on all matched points.

In practice, results from previous work [25], [35] to match points led to the following conditions:

- one to one match (proper match)
- one to many match (ambiguous match)
- one to none (missing points)

In order to solve this problem, we perform the following steps:

1. From the matched lists we pick up any arbitrary point that has one to one mapping in the scene.
2. Compute the initial transformation H_0.
3. Update the transformation function using a recursive technique.

The mathematical formulation of the above steps is performed in the next section.

5.6 Parameter Estimation Equations

In this section, the least-squares minimization problem is described. This problem is stated as follows; given n scene points and their corresponding model, find the

location for which the model points best fit the scene points. The solution of this problem can be classified into two group;

1. The geometrical methods.
2. The analytical methods.

Geometrical methods

This class of methods compute the position of the pixels by making consideration on the relative geometric position of the model and the scene points. Examples of such analysis are presented in [10], [48], and [12]. The advantages of using these technique is the possibility they offer to understand the inverse perspective problem [27].

Analytical methods

This class of methods use a well established numerical methods that exist for parameter estimation. One of the best work that surveys analytical techniques for that purpose is Haralick and Joo [33] in which three numerical techniques for least-squares estimation for position estimation are described .

The first method described in [33] employs a two stages least-squares fitting algorithm to find the best fit of a $3D$ points (model points) in a set of $2D$ points (scene points). This method is iterative; at each iteration, in the first stage. an optimal range of each point in the scene is calculated. This is based on the assumption that the rotation, translation and scaling which define the object position are known. In the second stage, the scene point ranges are supposed known and the optimal transformation parameters are calculated using the method presented in [34]. Faugeras in [17] used a parameter estimation technique (Kalman filter) to solve the minimization problem for

industrial parts recognition. This method was adopted in this thesis and a variation of the procedure is derived to fit in the problem of matching aerial images.

Joo and Haralik in [33] presented a second method to find the least squares solution by linearization of the least-squares function. This method is also iterative. The solution is refined at iteration using a linearized model. Faugeras [37] proposed the same technique using segment matching instead of point matching. Another class of methods presented in [33] employs more elaborated nonlinear regression techniques. This is done by minimizing an objective function that is more general than the sum of the squared residuals associated with the sample mean. Besl and McKay in [6] used a procedure based on the same idea. However, in order to achieve good results using this type of techniques, all points in the two images must be matched first. Otherwise, it may trap in a local minimum. In the next section the details for the second step will be presented.

Let $p_1, p_2, \cdots, p_n$ be the observed model points in the scene and $q_1, q_2, \cdots, q_n$ be the corresponding data points in the model. The point correspondence is made using the invariant moments introduced in section 5.5.3. Using equation (5.2) we have

$$q_i = \alpha(Rp_i) + T \tag{5.31}$$

where R, T, and α represent the rotation, translation and scaling components respectively.

It is clear that the above equation is valid only if the system is ideal i.e it does not contain any noise. In reality we have

$$q_i = \alpha(Rp_i) + T + \epsilon_i \tag{5.32}$$

where ϵ_i represents zero mean noise components such that

$$E(\epsilon) = 0 \tag{5.33}$$

Equation 5.31) is a linear system with four unknowns. It should be noted however, that ϵ_i is a random variable and that there is no way to compute R, T, and α using equation 5.31 directly. These parameters can only be estimated indirectly using the set of measured data q_m.

Estimation theory offers a large variety of estimators of which performance varies according to the problem. In this thesis, maximum likelihood estimator is used. This estimator maximizes the likelihood function

$$P(q_i^m|(R^*, T^*, \alpha^*)) \tag{5.34}$$

where P is the probability density of q_i^m given R, T, and α, q_i is the coordinate of pixel i. P is not a conditional probability, it is the evaluation of the probabilitv of having q_i^m assuming R, T, and α are true. The maximum likelihood estimator is generally biased. However, it has the following interesting asymptotic properties;

- When the number (m) of samples increases, the estimator is asymptotically efficient, i.e unbiased and of minimum variance.
- If the observation vector q_i^m is Gaussian, the maximum likelihood estimator comes to a least-squares estimator which is easier to estimate [28].

5.6.1 Estimation of Initial Transformation

To initialize the algorithm for the registration phase of the two image, an initial set of parameters is needed. Aldroubi and Unser in [49] assumed the initial set of

parameters to be zero. They used multi-resolution technique [49] to register the two images. In this technique, the registration function converged fast at coarse resolution and good results were obtained [49]. However, when the images are re-sampled at higher resolution, the convergence rate deteriorated. Also, in the case where only one level of resolution is used, setting the initial values to zero may not lead to a global minima and/or may take long time to converge. [49]. For this reason, a different approach was taken. A variation of the procedure described by Faugeras in [17] is used. Since the model position is defined by a transformation H as described in section 5.3, this transformation is described by a parameter vector $H = (k\cos\theta, k\sin\theta, t_x, t_y)$, such that the relation between a point (u, v) in the scene and an arbitrary point (x, y) in the reference image is given by the set of equations given in 5.7. Given a point in the scene and a corresponding point in the reference image, a hypothesis (i.e., a prediction of the position of the model in the scene) is generated by matching the privileged point of reference image to a compatible point of the scene. This compatibility is already determined in the initial matching stage. By finding the partial derivatives of equation 5.1 with respect to each coefficient of the coordinate transformation described by equations 5.7, the transformation parameters of H_0 can be obtained.

5.6.2 On Line Parameter Estimation

Based on the previous properties, equation (5.2) can be written as

$$Y_m = X_m H \tag{5.35}$$

where

$$Y_m = \begin{pmatrix} y(1) \\ y(2) \\ \vdots \\ y(m) \end{pmatrix}$$

$$X_m = \begin{pmatrix} x_1(1) & \cdots & x_n(1) \\ x_1(2) & & x_n(2) \\ \vdots & & \vdots \\ x_1(m) & \cdots & x_n(m) \end{pmatrix}$$

$$H = \begin{pmatrix} H_1 \\ H_2 \\ \vdots \\ H_n \end{pmatrix}$$

H is the transformation function, X is the set of points in the reference image and Y is the set of points in the scene. The subscript m is introduced since more than one equation in the above vector equation will be computed. Furthermore, denote H in equation 5.35 as $\hat{H}(m)$:

$$\hat{H}(m) = (X_m^t X_m)^{-1} X_m^t Y_m \tag{5.36}$$

If a new equation $(m+1)$th, is obtained as

$$Y(m+1) = [H_1 x_1(m+1), H_2 x_2(m+1), \cdots, H_n x_n(m+1)]$$

defining

$$x^t(m+1) = [x_1(m+1), x_2(m+1), \cdots, x_n(m+1)]$$

then we have

$$Y(m+1) = x^t(m+1)H \tag{5.37}$$

Thus the system of $(m+1)$ equations can be written as

$$Y_{m+1} = X_{m+1} H \tag{5.38}$$

in which

$$Y_{m+1} = \begin{pmatrix} y(1) \\ y(2) \\ \vdots \\ y(m) \\ \\ y(m+1) \end{pmatrix} \tag{5.39}$$

$$X_{m+1} = \begin{pmatrix} x_1(1) & \cdots & x_n(1) \\ x_1(2) & & x_n(2) \\ \vdots & & \vdots \\ x_1(m) & \cdots & x_n(m) \\ \\ x_1(m+1) & \cdots & x_n(m+1) \end{pmatrix} \tag{5.40}$$

and the new least square estimator is

$$\hat{H}(m+1) = (X_{m+1}^t X_{m+1})^{-1} X_{m+1}^t Y_{m+1} \tag{5.41}$$

Let $P(m) = (X_m^t X_m)^{-1}$,

from the matrix inversion lemma [2] $p(m+1)$ can be written in the form

$$\begin{aligned} p(m+1) &= [P(m)^{-1} + x(m+1)x^t(m+1)]^{-1} \\ &= P(m) - P(m)x(m+1)[1 + x^t(m+1)P(m)x(m+1)]^{-1} \\ &\quad x^t(m+1)P(m) \end{aligned} \tag{5.42}$$

From equation (5.41)

$$\hat{H}(m+1) = P(m+1)[X_m^t Y_m + x(m+1)Y(m+1)]$$

$$
\begin{aligned}
= \; & P(m)X_m^t Y_m - P(m)x(m+1)[1 + x^t(m+1) \\
& P(m)x(m+1)]^{-1}.x^t(m+1)P(m)X_m^t Y_m \\
& + P(m)x(m+1)Y(m+1)Y(m+1) - P(m)x(m+1) \\
& [1 + x^t(m+1)P(m)x(m+1)^{-1} \\
& x^t(m+1)P(m)x(m+1)Y(m+1)
\end{aligned}
$$

The last two terms can be arranged in the form of

$$
\begin{aligned}
& P(m)x(m+1)[1 + x^t(m+1)P(m)x(m+1)]^{-1} \\
\times & [1 + x^t(m+1)P(m)(m+1) - x^t(m+1)P(m)x(m+1)]Y(m+1) \\
= & \; P(m)x(m+1)[1 + x^t(m+1)P(m)x(m+1)]^{-1}Y(m+1)
\end{aligned}
$$

From equation (5.36) we have

$$
\hat{H}(m) = P(m)X_m^t Y_m
$$

Thus $\hat{H}(m+1)$ can be simplified to the form

$$
\begin{aligned}
\hat{H}(m+1) \; = \; & \hat{H}(m) + P(m)x(m+1)[1 + x^t(m+1)P(m)x(m+1)]^{-1} \\
& [Y(m+1) - x^t(m+1)\hat{H}(m)]
\end{aligned} \tag{5.43}
$$

The above equation can be carried out by the following recursive set of equations;

$$
\begin{aligned}
\hat{H}(m+1) \; = \; & \hat{H}(m) + \gamma(m+1)P(m)x(m+1)[Y(m+1) \\
& - x^t(m+1)\hat{H}(m)]
\end{aligned} \tag{5.44}
$$

$$
\begin{aligned}
P(m+1) \; = \; & P(m) - \gamma(m+1)P(m)x(m+1) \\
& x^t(m+1)P(m)
\end{aligned} \tag{5.45}
$$

where

$$
\gamma(m+1) = \frac{1}{[1 + x^t(m+1)P(m)x(m+1)]} \tag{5.46}
$$

The matrix $P(m)$ is a measure of the covariance of $\hat{H}(m)$ such that the covariance $\Psi = \sigma_v^2 P(m)$. In the next section an expression for the covariance will be derived.

5.6.3 Covariance Matrix Calculation

The covariance matrix Ψ corresponding to the estimate error $\hat{H} - H$ is

$$\begin{aligned}\Psi &= E\{(\hat{H} - H)(\hat{H} - H)^t\} \\ &= E\{[(X^tX)^{-1}X^t\epsilon][(X^tX)^{-1}X^t\epsilon]^t\} \\ &= (X^tX)^{-1}E\{\epsilon\epsilon^t\}X(X^tX)^{-1}\end{aligned}$$

Since the error may be weighted differently for each of the parameters to be measured, an extra weighting matrix is added. Faugeras in [17] assumed W be the weighting matrix and defined it as

$$\begin{pmatrix} w_i^2 & 0 \\ 0 & w_i^2 \end{pmatrix} \tag{5.47}$$

The values of w_i is chosen such that the ratio between the lengths of a line segment in the image and a corresponding line segment in the scene is below a threshold (typically 0.3)

Then the function to be minimized in equation 5.1 becomes

$$\begin{aligned}d &= \sum_{i=1}^{m} \epsilon_i^2 W = \epsilon^t W \epsilon \\ &= \sum_{i=1}(Y_i - X_i H)^t W (Y_i - X_i H)\end{aligned} \tag{5.48}$$

If $S = E[\epsilon\epsilon^t]$ then Ψ becomes

$$\Psi = (X^tWX)^{-1}X^tWRW^tX(X^tWX)^{-1} \tag{5.49}$$

Faugeras also assumed the weighting matrix W to be equal to the inverse of the rotation matrix R. This led the Covariance matrix Ψ to become

$$\Psi = (W = R^{-1}) = (X^tR^{-1}X)^{-1} = \Psi_{mv} \tag{5.50}$$

The error covariance matrix based on this assumption in equation 5.50 is the minimum error covariance matrix in the sense that for any other choice of the weighting matrices W we will have

$$\Psi_{mv} \leq \Psi$$

Deutsch showed in [11] that this is a positive definite matrix that is less or equals to Ψ if the difference $\Psi_{mv} - \Psi$ is non negative definite. The subscript mv in Ψ_{mv} denotes the minimum variance property. A complete derivation of this procedure can be found in [11]. Hence, the elements of the initial covariance matrix Ψ_{mv} is given by 4×4 elements below

$$\Psi_{mv} = \begin{pmatrix} \psi_{1,1} & \psi_{1,2} & 0 & 0 \\ \psi_{2,1} & \psi_{2,2} & 0 & 0 \\ 0 & 0 & \psi_{3,3} & 0 \\ 0 & 0 & 0 & \psi_{4,4} \end{pmatrix}$$

The elements of the above matrix are given by

$$\begin{aligned} \psi_{(1,1)} &= \alpha_0^2 \sin^2(\theta_0)\sigma_\theta^2 + \cos^2(\theta_0)\sigma_\alpha^2 \\ \psi_{(2,2)} &= \alpha_0^2 \cos^2(\theta_0)\sigma_\theta^2 + \sin^2(\theta_0)\sigma_\alpha^2 \\ \psi_{(1,2)} &= \psi_{0(2,1)} \\ &= \sin(\theta_0)\cos(\theta_0)(\sigma_\alpha^2 - \alpha_0^2\sigma_\theta^2) \\ \psi_{(3,3)} &= \sigma_x^2 \\ \psi_{(4,4)} &= \sigma_y^2 \end{aligned}$$

The rest of the elements in the matrix are zero. Ψ_0 is initialized for each hypothesis with respect to the error variance σ_α^2, σ_θ^2, σ_x^2 and σ_y^2 attached to the initial estimate α_0, θ_0. The results of equation 5.44 show that the new estimate is given by the old estimate plus a correction term. The matrix $P(m)$ in the correction term can be updated by the recursive formula in equations 5.44, 5.45).and 5.46. It is also clear that matrix inversion is eliminated (the term $[1 + x^t(m+1)P(m)x(m+1)]$ is a

scalar quantity) and therefore the computational efficiency is improved for updating the estimate $\hat{H}$.

Thus, to summarize, the following steps are used in order to estimate the parameters of $\hat{H}$

1. Calculate the initial Covariance from section 5.6.3 the first m points at P_0 and the initial transformation $\hat{H}_0$ from section 5.5.3.
2. Calculate $\gamma(m+1)$ from equation (5.46)
3. Calculate the new Covariance $P(m+1)$ from equation (5.45)
4. Update the transformation $\hat{H}$ in equation for the new covariance matrix from equation (5.44).

5.7 Conclusion

In this chapter a methodology of matching two images under geometrical deformation was described. First matching points were estimated from both images. Third order moments were used for that purpose since they possess the property of being invariant under rotational, translational and scaling geometrical deformations. Then, an initial transformation that maps the extracted features from one image to the other was estimated. This transformation was used as a starting guess to calculate global transformation. In order to achieve this, a recursive parameter estimation algorithm was derived under some predetermined assumptions. The advantage of using this approach is it estimates a local transformation that represents the deformation parameters, then, at every iteration it adds more features (line segments)

and updates the parameters in a recursive manner. In this way an estimation of the transformation parameters can be obtained.

Chapter 6

Tests and Results

6.1 Introduction

In this chapter the testing methodology followed in the validation of the algorithms is discussed. An evaluation of the robustness, effectiveness, and the limitations of the algorithms and the complete matching and registration is conducted. The tests seek to confirm that the capabilities of the algorithms are not severely affected by deformations in the image such as: changes in the orientation (rotation), or the position (translation), or magnification (scaling), and finally the absence of part of the image due to changes of the scene or any other uncontrollable conditions. The tests were conducted on a set of SPOT-PLA images provided by the Canadian Center of Remote Sensing (CCRC). Only the results and parameters of five sets of images which are used to validate the experiments are illustrated. Showing the results for a larger number of images would take too much space and will not contribute to a better understanding of the merits of the approach discussed in this thesis. All

the algorithms are implemented in C/Unix language. Each stage of the matching process is tested and validated starting by extracting the features and dominant points, and continuing with image modeling up to the last stage of image matching and registration. The tests will cover the following areas:

- First, the verification that the dominant points detected are always located in the same relative area of the object's boundary. That is the points are relatively stable under rotations and various scaling factors.
- The stability of the moment parameters is then checked. This is achieved by testing the algorithm under different changes in scaling and different degrees of rotations and translations.
- Finally, the registration algorithm is tested under different geometric orientations. Also, it is checked under different levels of Gaussian noise to test the robustness under noise variation.

The next section describes the results of the algorithms and operations used in the preprocessing phase. These algorithms have been applied to the images shown in Figures 6.2 to 6.5. Figure 6.1 shows Khoros graphics Environment (KGE) where all the algorithms implemented in this thesis were developed and integrated within.

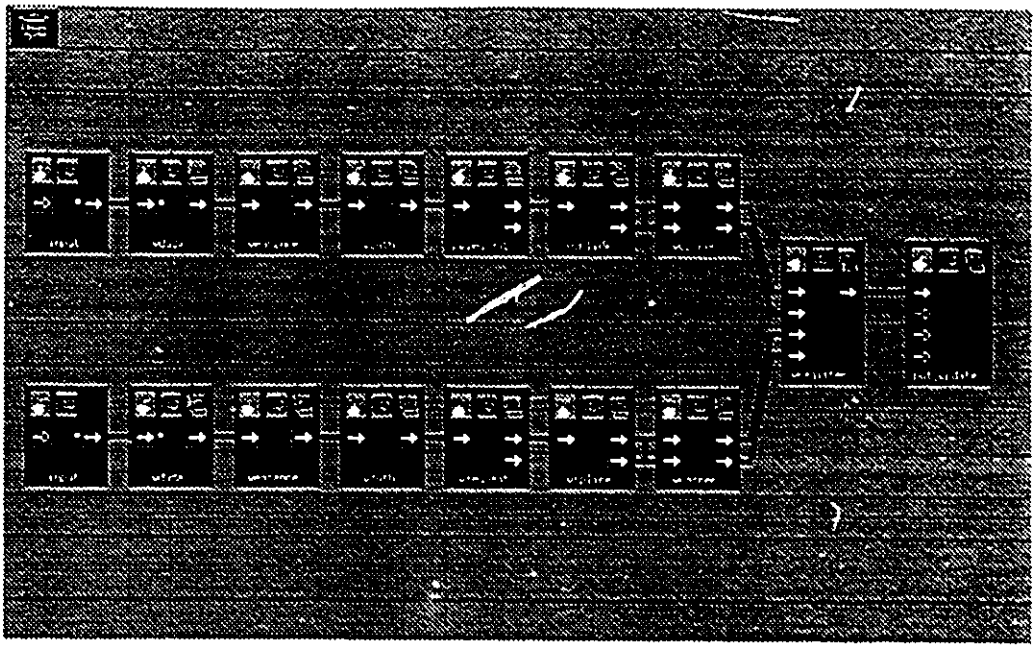


Figure 6.1: A snap shot of the KGE with the blocks that represent each stage for the matching process.

Figure 6.2: SPOT-PLA image of a section of Ottawa, Ontario

Figure 6.3: A different SPOT-PLA image of the same section of Ottawa, Ontario

6.2 Feature Extraction

This stage involves extracting a 512 × 512 pixels image from a 2000 × 2000 image supplied by the Canadian Center for Remote Sensing (CCRS). [1], where each pixel has an 8 bit gray scale value from 0 to 256. The next sections describe each step performed in this stage.

6.2.1 Linear Feature Extraction

In this step, The Duda road operator (DRO) is implemented and used in order to extract linear features from the image such as roads, rivers, etc. These features are extracted based on the assumption that they are the features of interest. The Duda

[1] All images are inverted during processing for printing purposes.

filter applies a scoring function for the detection of the linear features as described in equations 3.4 and 3.5 in chapter 3.This function is used in conjunction with four different masks in order to identify 4 directions: vertical, horizontal, right diagonal and left diagonal. The three parameters that affect the performance of the filter are:

- θ linear feature contrast threshold which takes the default value as 15.
- θ_1 lower intensity threshold along the linear feature. This takes the default value as 5.
- θ_2 higher intensity threshold along the linear feature. This also takes the default value as 15.

  Several tests were conducted with different values for those parameters and the best values were used as the default.

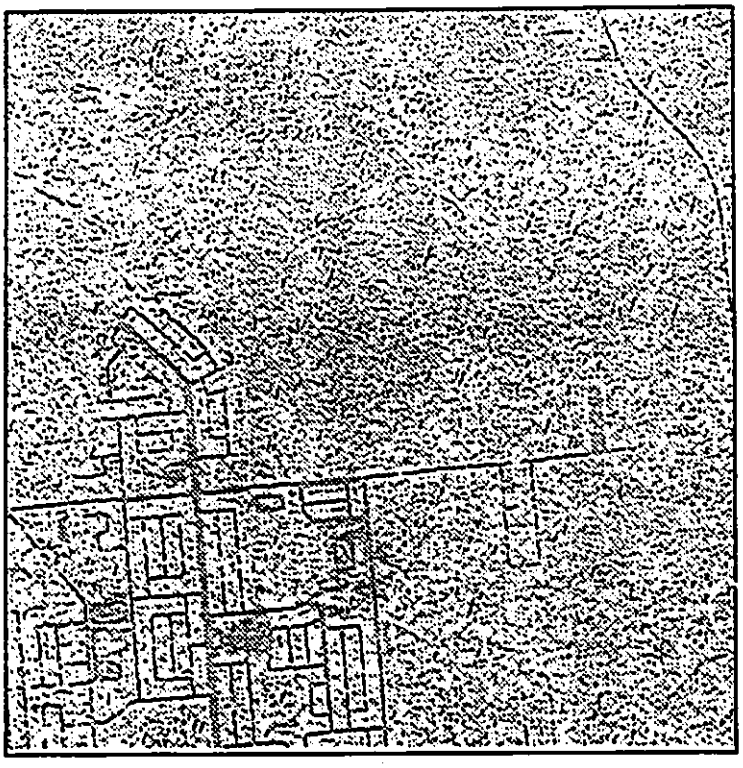


Figure 6.4: The result of the Duda filter on the original image

6.2.2 Image Enhancement

The objective of image enhancement is to highlight the features of interest. In most cases this implies improving the visual appearance of the image. Several advantages are gained from performing this stage;

- An increase in the dynamic range of the image.
- Approximating the image histogram to a bimodal shape so that a proper threshold value can be chosen.

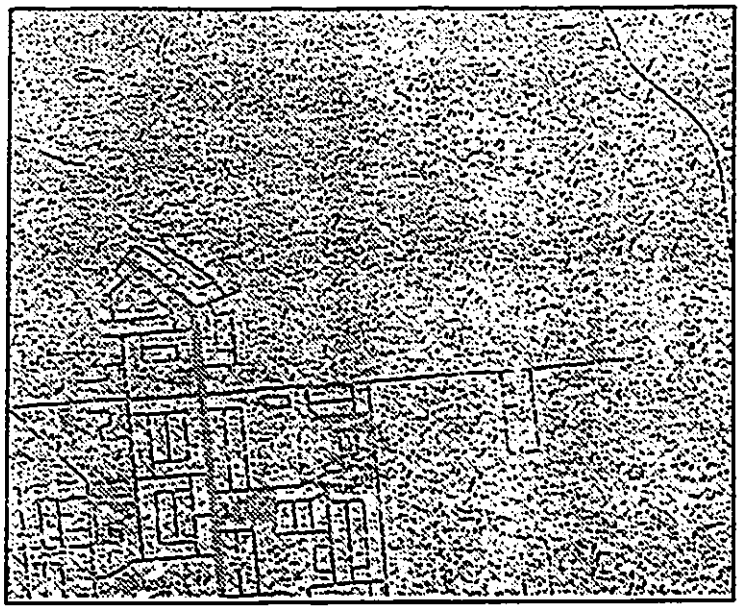
Figure 6.5-b shows histogram of the images before image enhancement while figure 6.5-d shows the histogram for the same image after the local enhancement.

From the enhancement step, the threshold value can be obtained The output form the thresholding step is shown in figure 6.2.2

6.2.3 Corner Detection

Since the time factor is considered in the process of image recognition, other features that are important and easy to extract are considered as well. The main assumption of choosing these features are:

1. They satisfy the invariance property
2. They are less in number compared with the line segments
3. They are easy to extract.



(a)

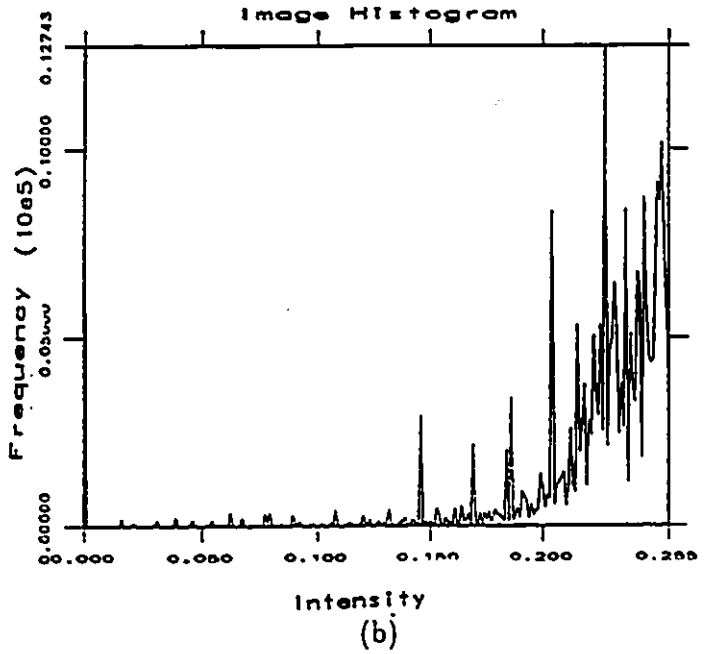
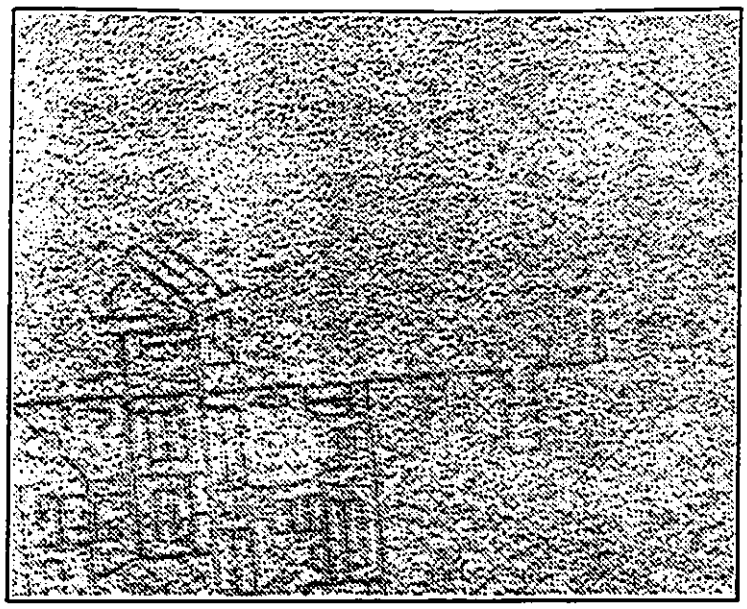
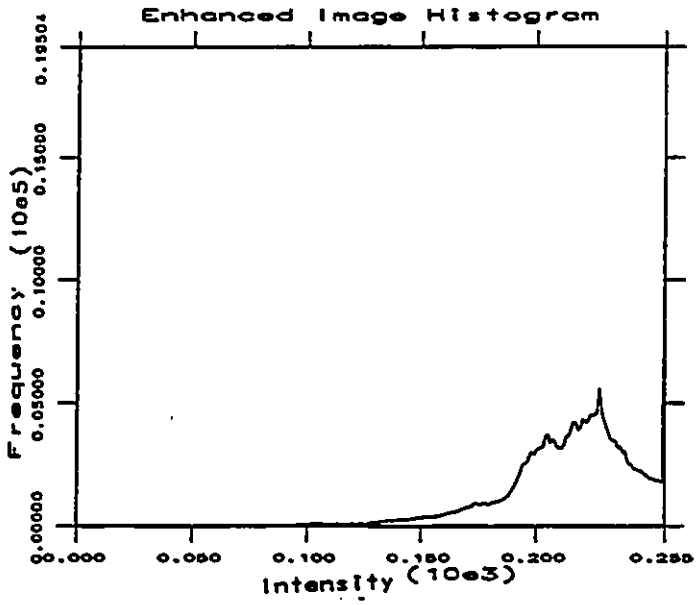


Image Histogram
Frequency (10e5)
0.12743
0.10000
0.00000
0.000
0.050
0.100
0.200
Intensity

(b)



(c)



(d)

Figure 6.5: Effect of local enhancement Applied to SPOT-PLA image a) Image before enhancement, b) The corresponding histogram , c) Image before enhancement. d) The corresponding histogram.

Figure 6.6: The result of the dynamic thresholding.

Figures 6.7 and 6.8 shows that result of this step. These features (corners) will be used in a latter stage.

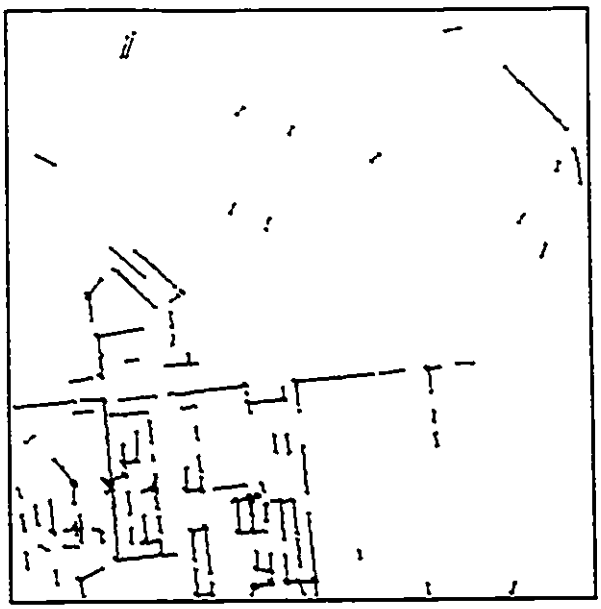


Figure 6.7: Corner detection for test image (1)

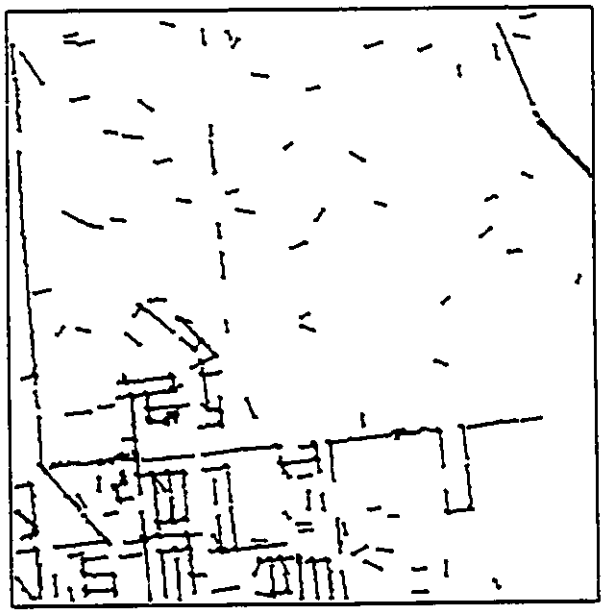


Figure 6.8: Corner detection For test image (2)

Table 6.1 shows the (x, y) positions of the extracted points from both images

Table 6.1: Extracted Corner points coordinates

No	Reference Image	Scene Image
1	(86, 370)	(126, 400)
2	(240, 482)	(396, 361)
3	(226, 463)	(139, 398)
4	(91, 405)	(24, 464)
5	(156, 416)	(229, 502)
6	(254, 493)	(234, 378)
7	(168, 414)	(23, 317)
8	(84, 337)	(84, 456)
9	(110, 349)	(237, 390)
10	(175, 504)	(64, 486)
11	(224, 449)	(268, 386)
12		(116, 454)
13		(111, 386)
14		(267, 373)
15		(183, 465)
16		(195, 464)
17		(107, 331)
18		(153, 451)
19		(84, 469)

6.3 B-spline modeling

In this step a cubic spline image model of order 3 is used to implement the algorithm described in chapter 4. Figures 6.9 and 6.10 show the result of this step when applied to the both test images as a part of the overall matching and registration process.

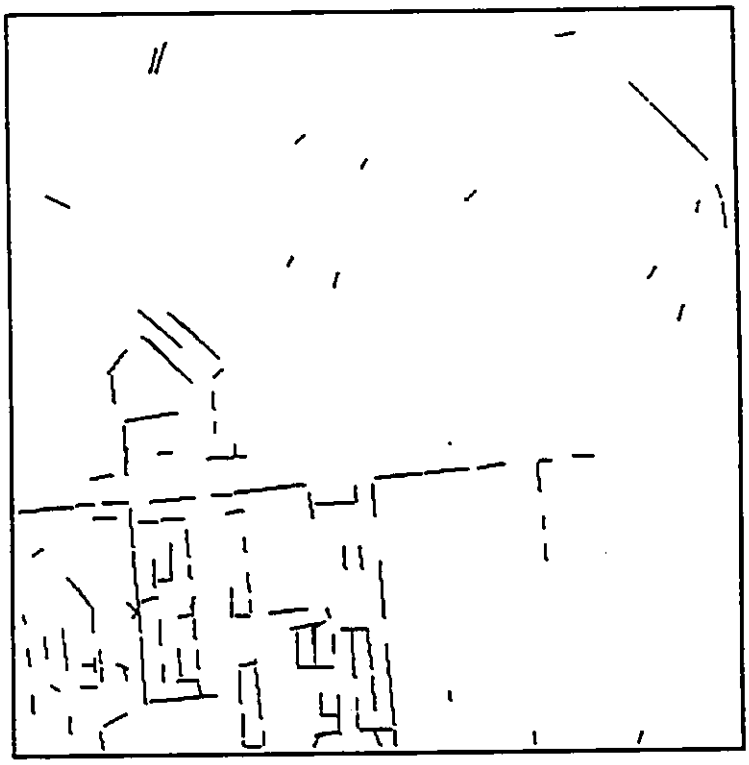


Figure 6.9: Modeling test image (1) using B-spline model

Figure 6.10: Modeling test image (2) using B-spline model

6.4 Matching with invariant Moment

To verify the validity of this stage, the two images in figures ?? and ?? are used. A number of 11 points were obtained from the image in figure ??, while 19 points were extracted from the image in figure ??. A circular templates of 16 pixel radius were taken centered at the selected points in each image. The initial matching is performed between the points in both images by computing the correlation coefficients using the algorithm described in section 5.5.3. The matched points are then used to determine the initial registration parameters as described in section 5.6.1. The result of the matching procedure is summarized in table 6.2. and the matched points are circled in figure 6.11.

No	Reference Image	Scene Image	Correlation Co.
1	(240, 482)	(396, 361)	0.9884
2	(91, 405)	(24 464)	0.9221
3	(156, 416)	(229, 502)	0.8409
4	(168, 414)	(23, 317)	0.9101
5	(110, 349)	(237, 390)	0.9084
6	(175, 504)	(64, 486)	0.9021
7	(254, 493)	(234, 378)	0.8999
8	(224, 449)	(268, 386)	0.9633

Table 6.2: Matching Points Coordinates.

6.4.1 Invariance to Scaling

To verify the invariance to scaling, the parameters of the image, were computed and compared for scaling of the image from 0.85 to 1.1. The results of these experi-

Figure 6.11: Matched points from both images

ments are given in table 6.4.1. The parameters in that table show that the moment parameters for the image stay almost unchanged under various scaling ratios.

Scale	M_1	M_2	M_3	M_4	M_5	M_6	M_7
0.80	1.50e-03	2.73e-08	2.23e+11	2.32e+12	1.66e+24	2.56e+08	-4.11e+23
0.85	1.50e-03	3.11e-08	1.98e+11	2.20e+12	1.42e+24	2.62e+08	-2.91e+23
0.90	1.49e-03	3.21e-08	1.74e+11	2.04e+12	1.17e+24	2.47e+08	-2.13e+23
0.95	1.47e-03	2.61e-08	1.84e+11	1.74e+12	9.17e+23	1.73e+08	-4.41e+23
1.00	1.48e-03	2.64e-08	1.38e+11	1.29e+12	5.05e+23	1.29e+08	-2.32e+23
1.05	1.48e-03	2.64e-08	9.95e+10	9.31e+11	2.62e+23	9.31e+07	-1.18e+23
1.10	1.48e-03	2.65e-08	7.02e+10	6.57e+11	1.31e+23	6.59e+07	-5.78e+22
1.15	1.48e-03	2.66e-08	4.90e+10	4.55e+11	6.27e+22	4.58e+07	-2.68e+22

Table 6.3: The Moment Parameters with Scaling for the test image. Each row corresponds to the parameters of a sub-part scaled by a given factor. The object is scaled by a given factor ranging from 0.97 to 0.86.

The diagram in figures 6.12 a) to g) represent the variation of the moments around the computed mean value.

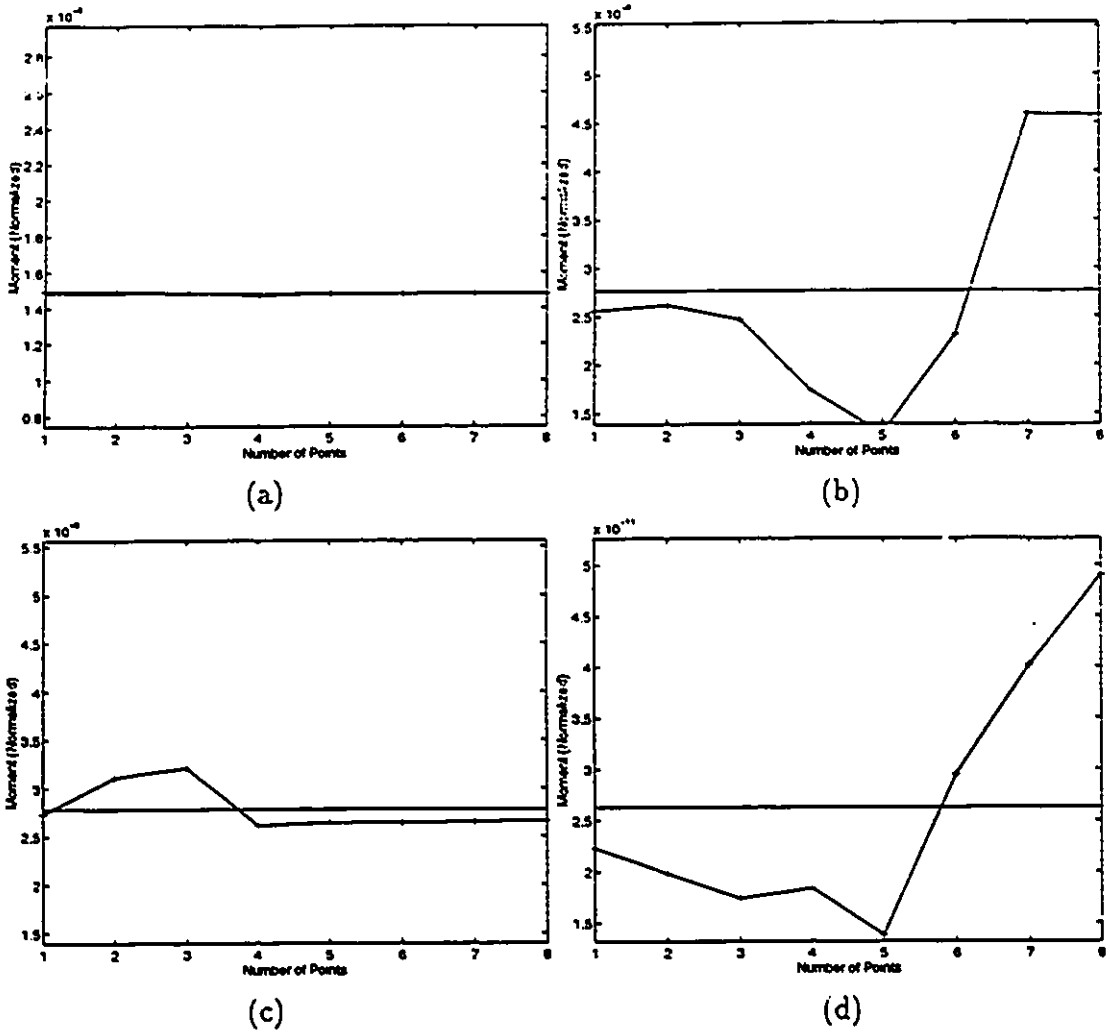
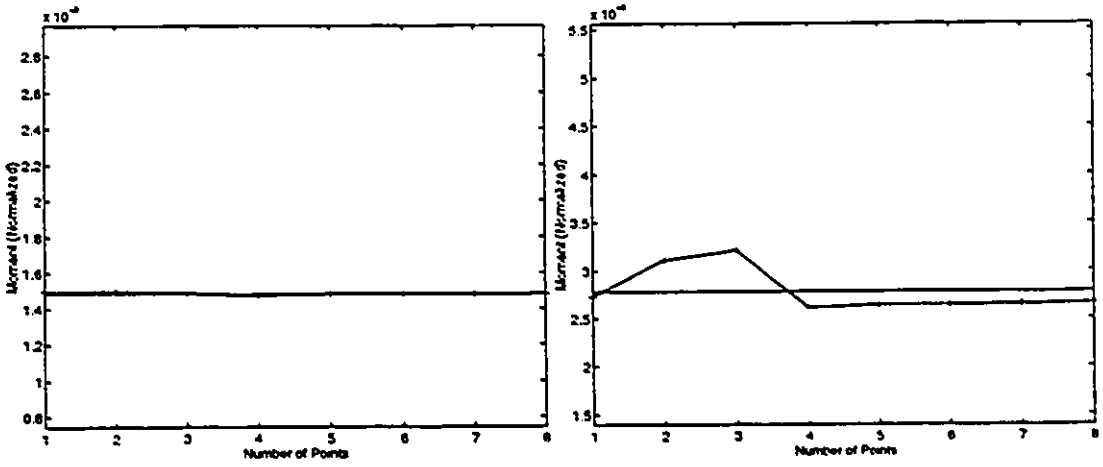
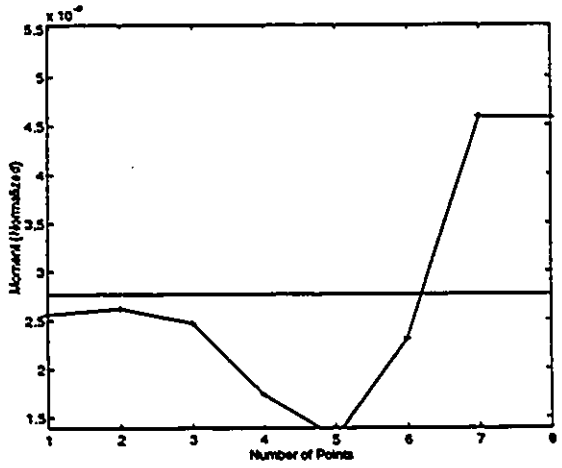


Figure 6.12: Evaluation of the invariance of moment parameters under Scaling. a) $M = 1$ b) $M = 2$ c) $M = 3$ d) $M = 4$



(e) (f)



(g)

Figure 6.13: Evaluation of the invariance of moment parameters under Scaling e) $M = 5$ f) $M = 6$ g) $M = 7$

6.4.2 Invariance to Rotation

The features in both the image and scene have been found to be stable under rotations varying from 0° to 360°. The results of this test are given in table 6.5. The diagrams in figures 6.14 show the variation of the moment parameters under different rotation values.

Rotation	M_1	M_2	M_3	M_4	M_5	M_6	M_7
0	9.85e-04	2.76e-09	4.92e-13	7.61e-14	1.27e-26	-3.51e-18	-1.64e-27
20	9.91e-04	2.78e-09	5.28e-13	7.88e-14	1.39e-26	-3.61e-18	-9.75e-27
40	9.91e-04	2.75e-09	4.64e-13	7.66e-14	1.15e-26	-3.38e-18	-1.35e-26
60	9.91e-04	2.84e-09	4.61e-13	7.75e-14	1.29e-26	-3.63e-18	-5.47e-27
80	9.91e-04	2.81e-09	5.07e-13	7.76e-14	1.35e-26	-3.66e-18	-8.93e-27
100	9.91e-04	2.79e-09	4.89e-13	7.73e-14	1.32e-26	-3.65e-18	-1.38e-26
120	9.91e-04	2.84e-09	4.14e-13	7.72e-14	1.22e-26	-3.65e-18	-8.92e-27
140	9.91e-04	2.82e-09	5.19e-13	8.08e-14	1.41e-26	-3.74e-18	-8.56e-27
160	9.91e-04	2.80e-09	5.25e-13	7.51e-14	1.25e-26	-3.47e-18	-5.22e-27
180	9.85e-04	2.76e-09	4.92e-13	7.61e-14	1.27e-26	-3.51e-18	-1.64e-27
200	9.91e-04	2.78e-09	5.28e-13	7.88e-14	1.39e-26	-3.61e-18	-9.75e-27
220	9.91e-04	2.75e-09	4.64e-13	7.66e-14	1.15e-26	-3.38e-18	-1.35e-26
240	9.91e-04	2.84e-09	4.61e-13	7.75e-14	1.29e-26	-3.63e-18	-5.47e-27
260	9.91e-04	2.81e-09	5.07e-13	7.76e-14	1.35e-26	-3.66e-18	-8.93e-27
280	9.91e-04	2.79e-09	4.89e-13	7.73e-14	1.32e-26	-3.65e-18	-1.38e-26
300	9.91e-04	2.84e-09	4.14e-13	7.72e-14	1.22e-26	-3.65e-18	-8.92e-27
320	9.91e-04	2.82e-09	5.19e-13	8.08e-14	1.41e-26	-3.74e-18	-8.56e-27
340	9.91e-04	2.80e-09	5.25e-13	7.51e-14	1.25e-26	-3.47e-18	-5.22e-27

Table 6.4: The Moment Parameters with Rotation for the test image. Each row corresponds to the parameters of the image rotated by 20°. The image is scaled by a factor ranging from 0 to 360

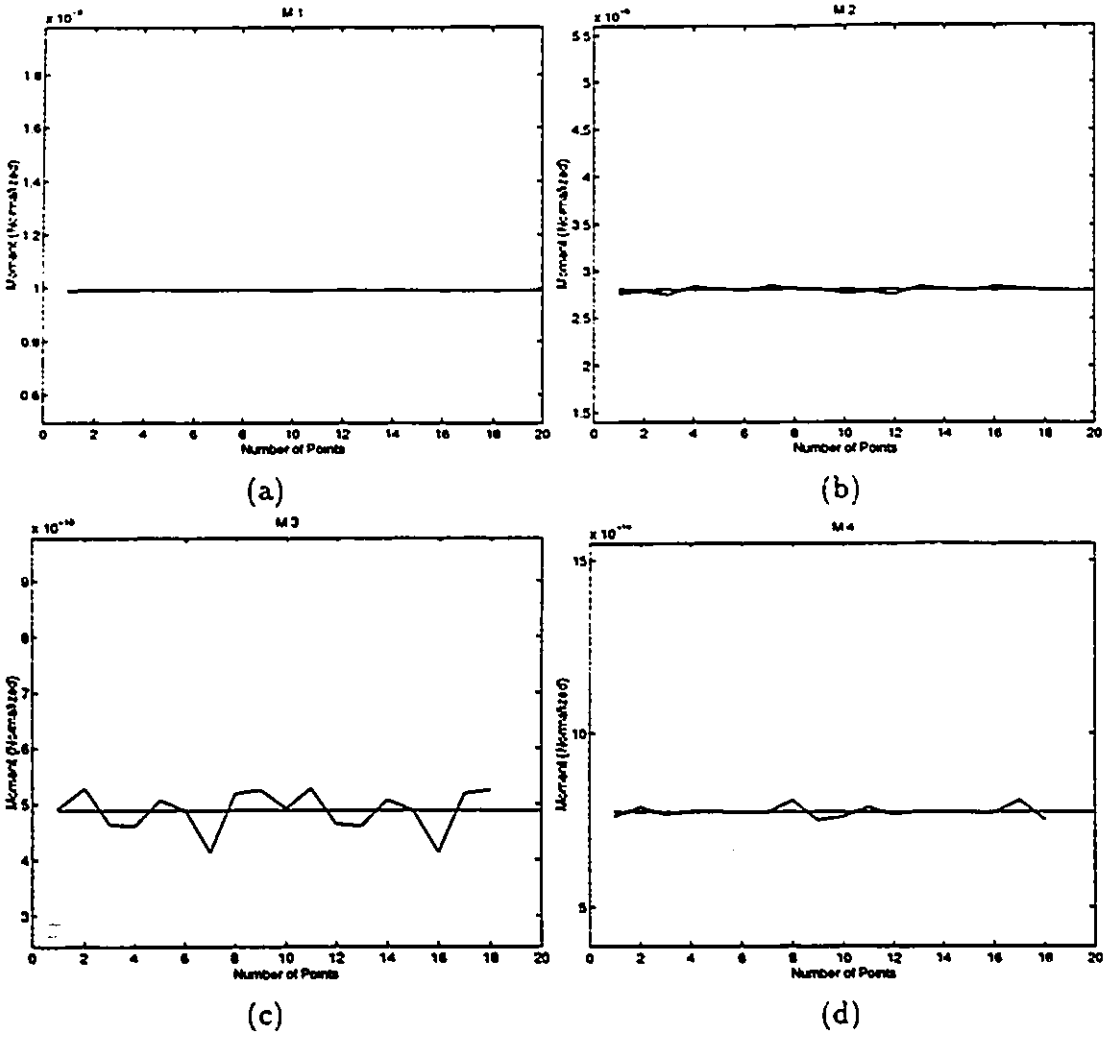


Figure 6.14: Evaluation of the invariance of moment parameters under Scaling. a) $M = 1$ b) $M = 2$ c) $M = 3$ d) $M = 4$

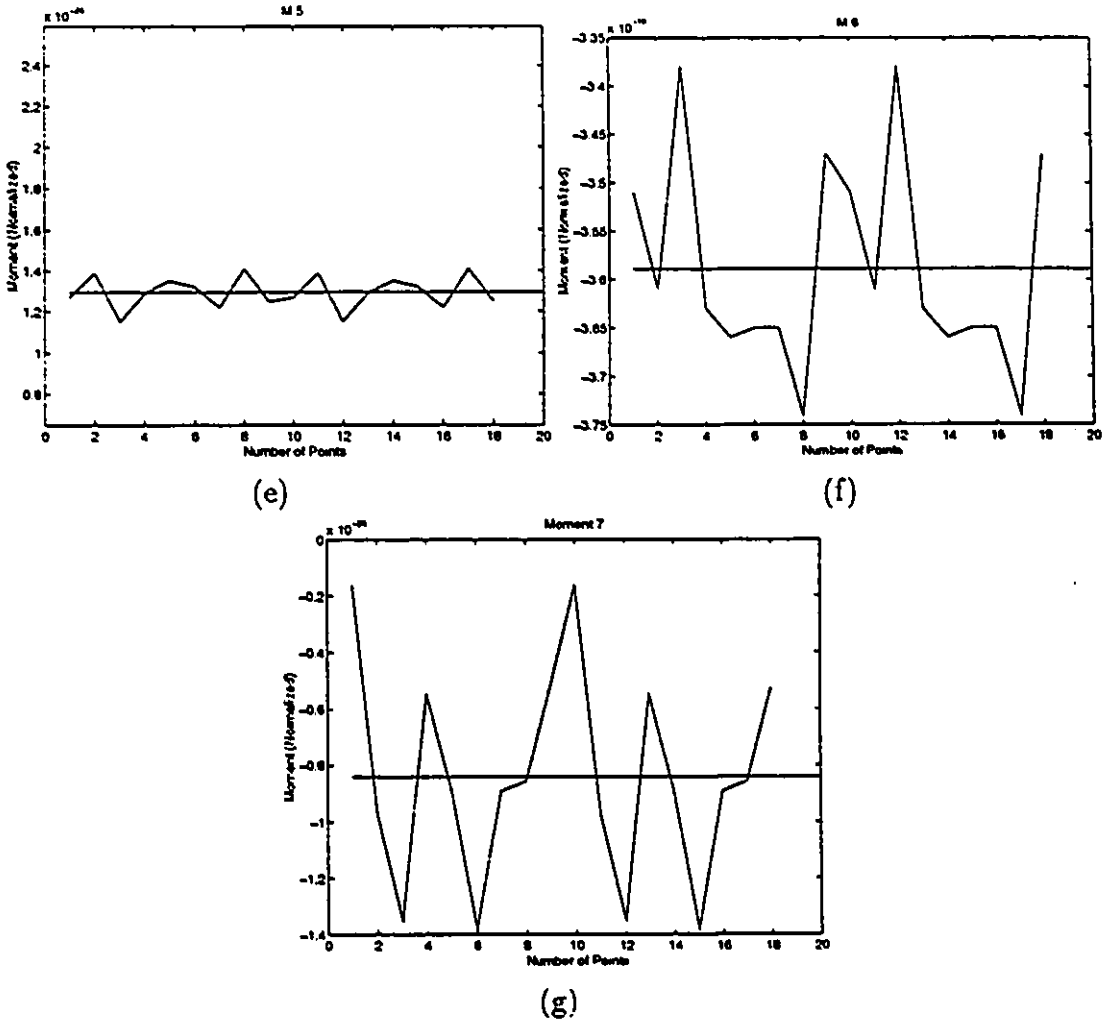


Figure 6.15: Evaluation of the invariance of moment parameters under Scaling.
e) $M = 5$ f) $M = 6$ g) $M = 7$

6.4.3 Parameter Estimation and Registration function

Several test images were generated by translating, scaling and rotating a standard reference image. The transformed images were then brought back into registration using the iterative parameter estimation procedure described in chapter 5. The behavior of the algorithm is analyzed by monitoring the evolution of the normalized quadratic error criterion of equation 5.1 in chapter 5 as a function of the number of iterations. The result of this test is given in figures 6.16 to 6.18. All parameters were initially set to the values computed from the initial matching stage. In this experiment, the test images were noise-free images, The parameters of the transformation function were imposed in order to check the convergence of the algorithm.

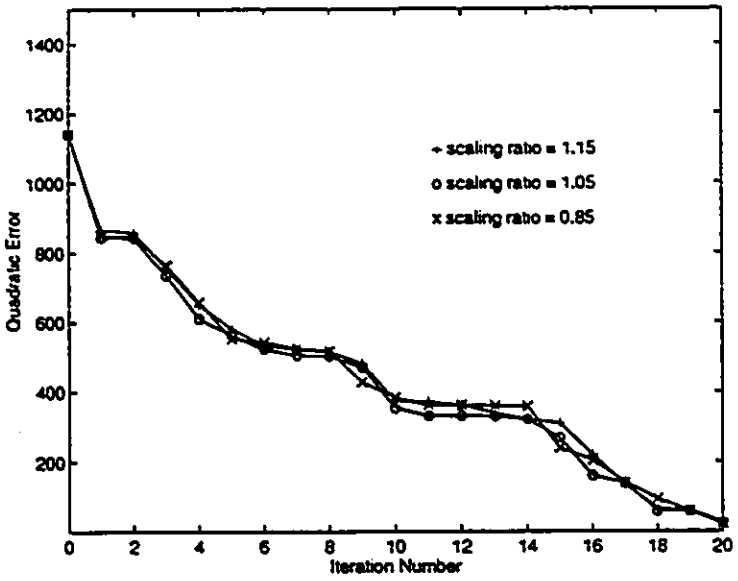


Figure 6.16: Evaluation of the quadratic error per pixel as a function of the number of iterations during the registration of the test image under scaling deformation

Next, the performance of the algorithm was investigated in the presence of noise. For this purpose, the same test images were used while adding Gaussian white noise with a known standard deviation to both the reference and the transformed images.

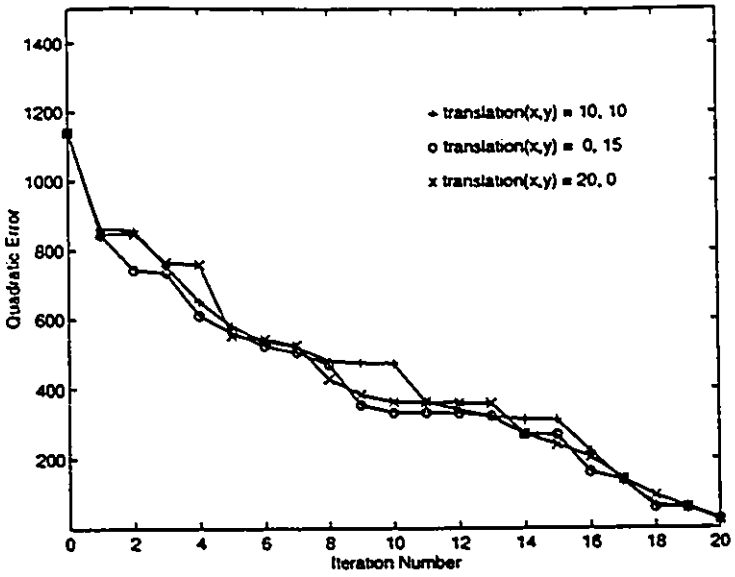


Figure 6.17: Evaluation of the quadratic error per pixel as a function of the number of iterations during the registration of the test image under translation deformation

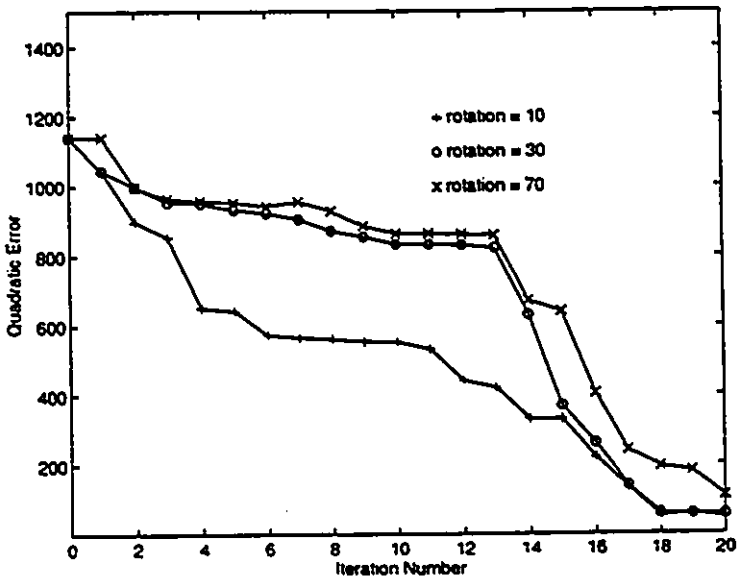


Figure 6.18: Evaluation of the quadratic error per pixel as a function of the number of iterations during the registration of the test image under rotation deformation

The results are summarized in table 6.5. The first observation shows that the parameter estimates become less accurate with the increasing level of noise then the noise-free images, which is to be expected. The graphs of the error as a function of iteration number is shown in figure 6.19.

SNR (dB)	Initial Estimates	Final Estimates	L.S. Error
$\sigma = 0(\rightarrow \infty)$	(12, 21)	(16, 21)	1.0884
$\sigma = 1(32.36dB)$	(23, 25)	(27, 37)	1.1221
$\sigma = 10(22.38dB)$	(33, 13)	(39, 17)	0.0084
$\sigma = 25(18.44dB)$	(35, 30)	(38, 35)	0.1001
$\sigma = 50(15.4dB)$	(47, 20)	(51, 34)	0.0084

Table 6.5: Results of the registration of the test image with various levels of noise.The actual parameter values are Translation $T = (15, 15)$ and $\theta = 30°$.

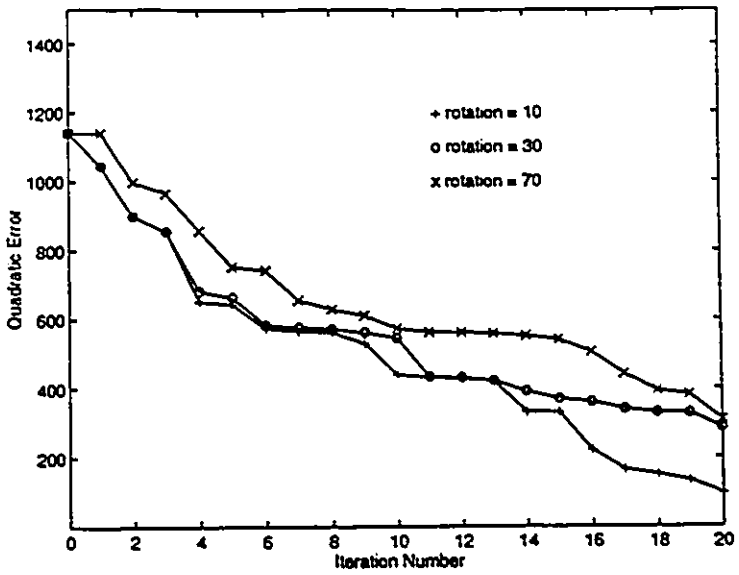


Figure 6.19: Evaluation of the quadratic error per pixel as a function of the number of iterations during the registration of the test image under various noise conditions.

Finally, the Algorithm was applied on aerial images with both transformation parameters and noise values unknown. The results of this test are shown in figure 6.20.

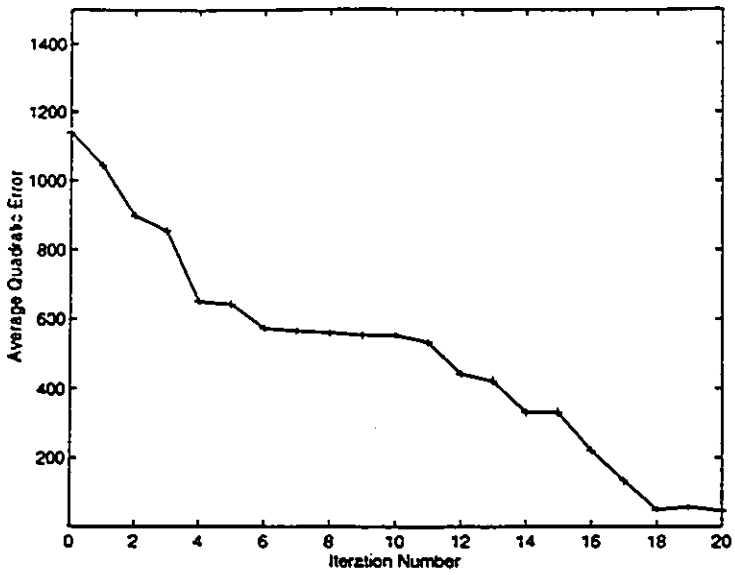


Figure 6.20: Evaluation of the quadratic error per pixel as a function of the number of iterations during the registration of the SPOT-PLA test image with unknown parameter values

6.4.4 Further Tests

The results from figure 6.21 to 6.33 were obtained for different images.

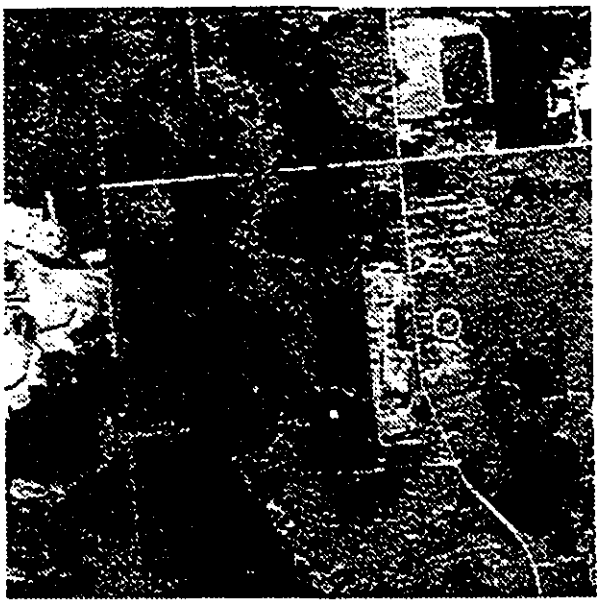


Figure 6.21: SPOT-PLA test no. 1 image 1

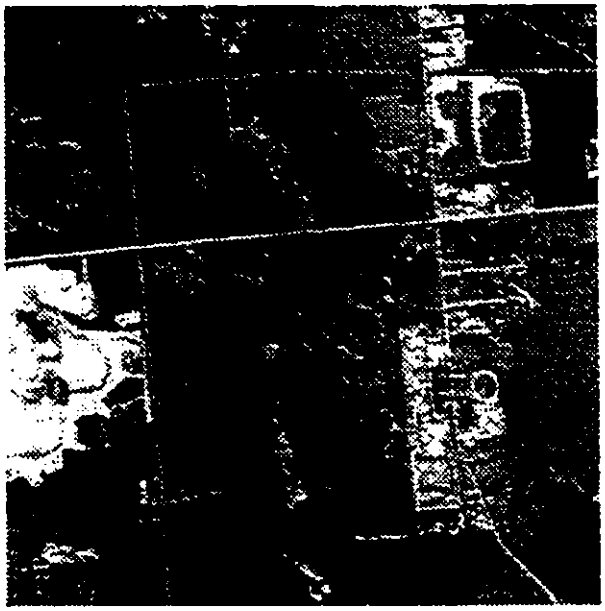


Figure 6.22: SPOT-PLA test no. 1 image 2

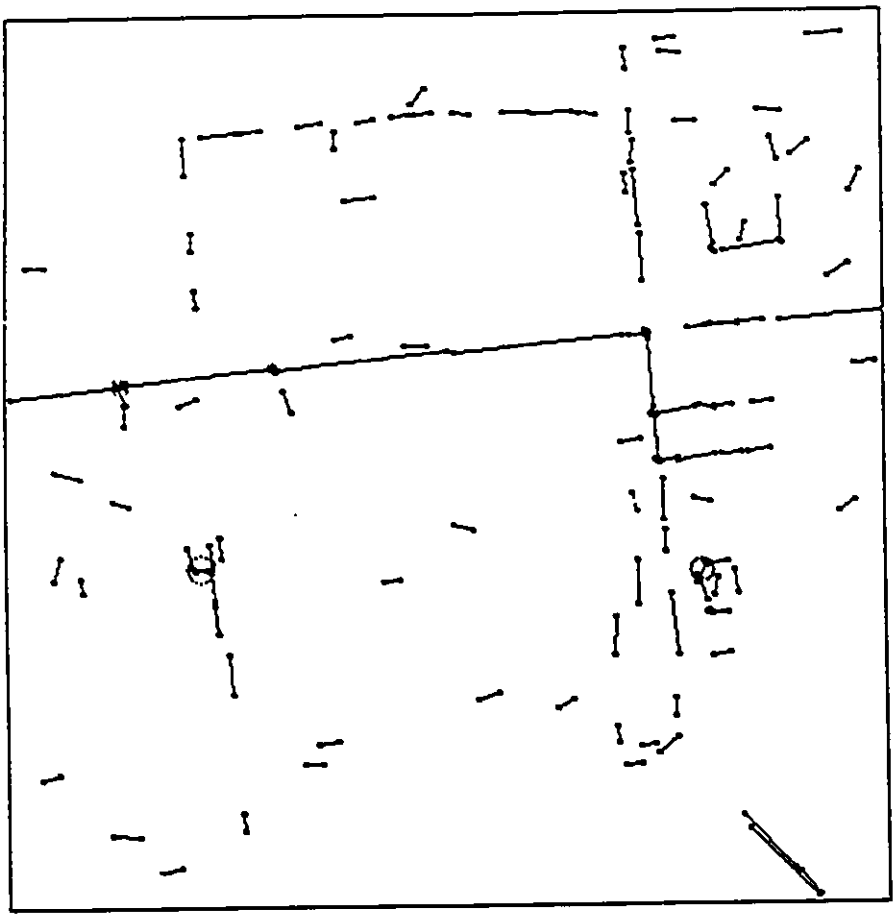


Figure 6.23: Result of the matching process of images (1) and (2) is shown above.

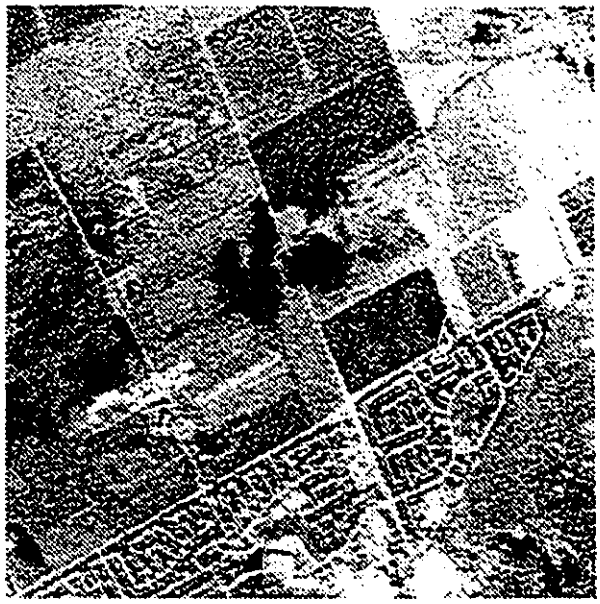


Figure 6.24: SPOT-PLA test no. 2 image 1

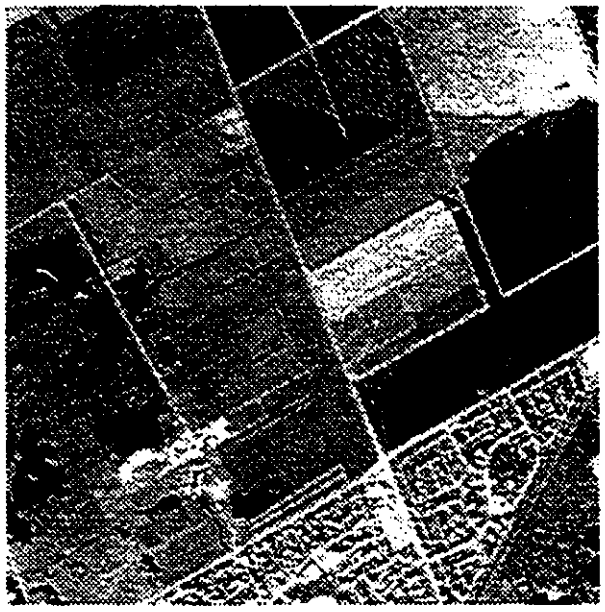


Figure 6.25: SPOT-PLA test no. 2 image 2

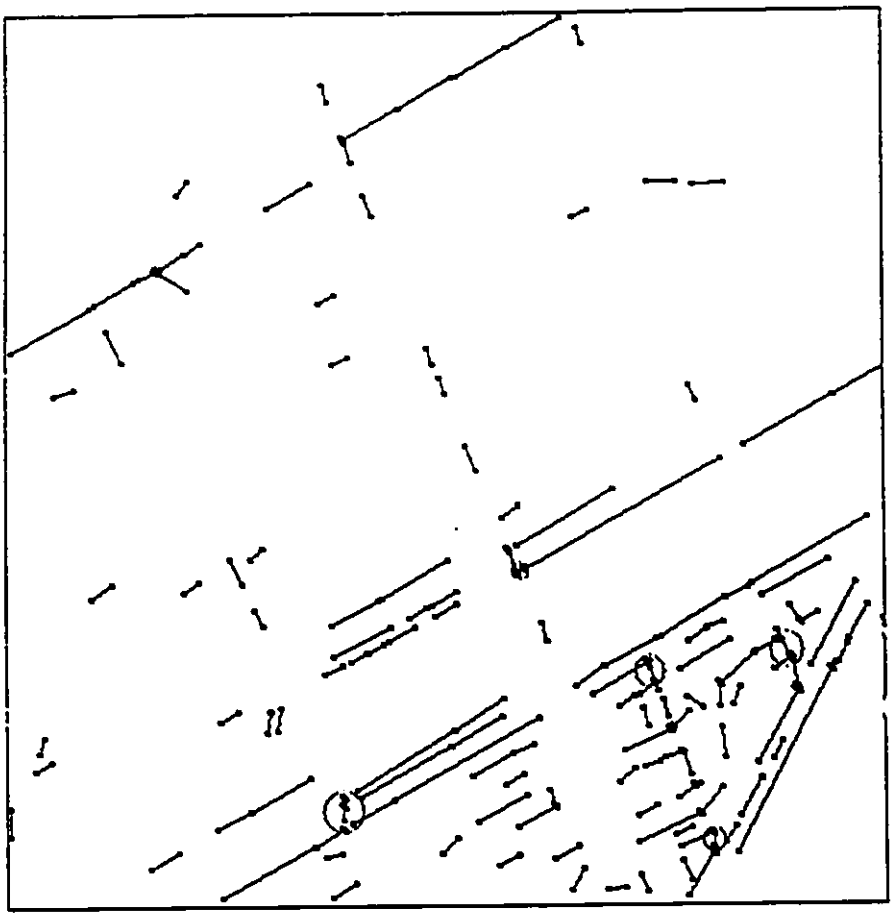


Figure 6.26: Result of the matching process of images (1) and (2) is shown above.

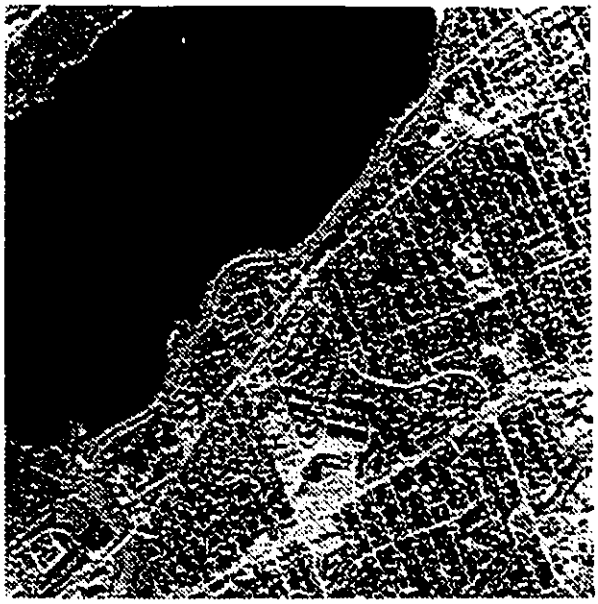


Figure 6.27: SPOT-PLA test no. 3 image 1

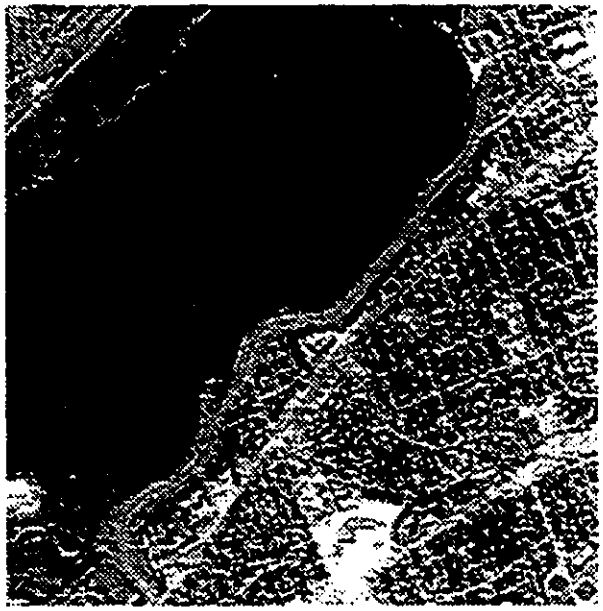


Figure 6.28: SPOT-PLA test no. 3 image 2

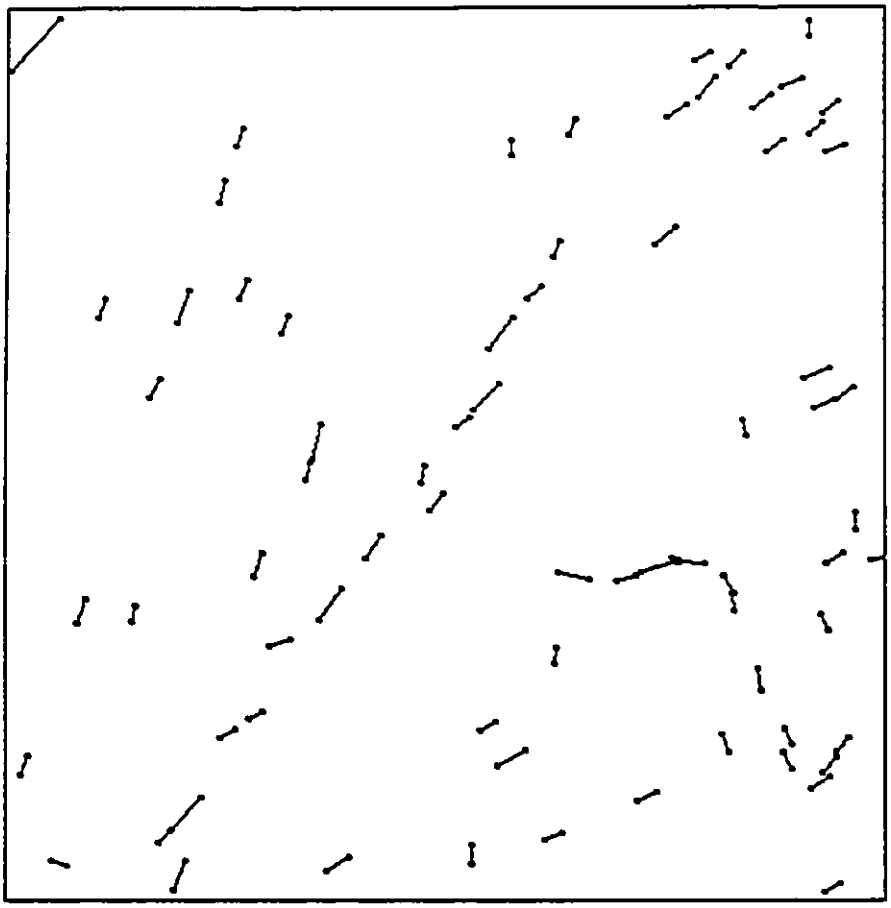


Figure 6.29: Result of the matching process of images (1) and (2) is shown above.

Figure 6.30: SPOT-PLA test no. 4 image 1

Figure 6.31: SPOT-PLA test no. 4 image 2

Figure 6.32: Result of the matching process of images (1) and (2) is shown above.

Figure 6.33: SPOT-PLA test no. 5 image 1

Figure 6.34: SPOT-PLA test no. 5 image 2

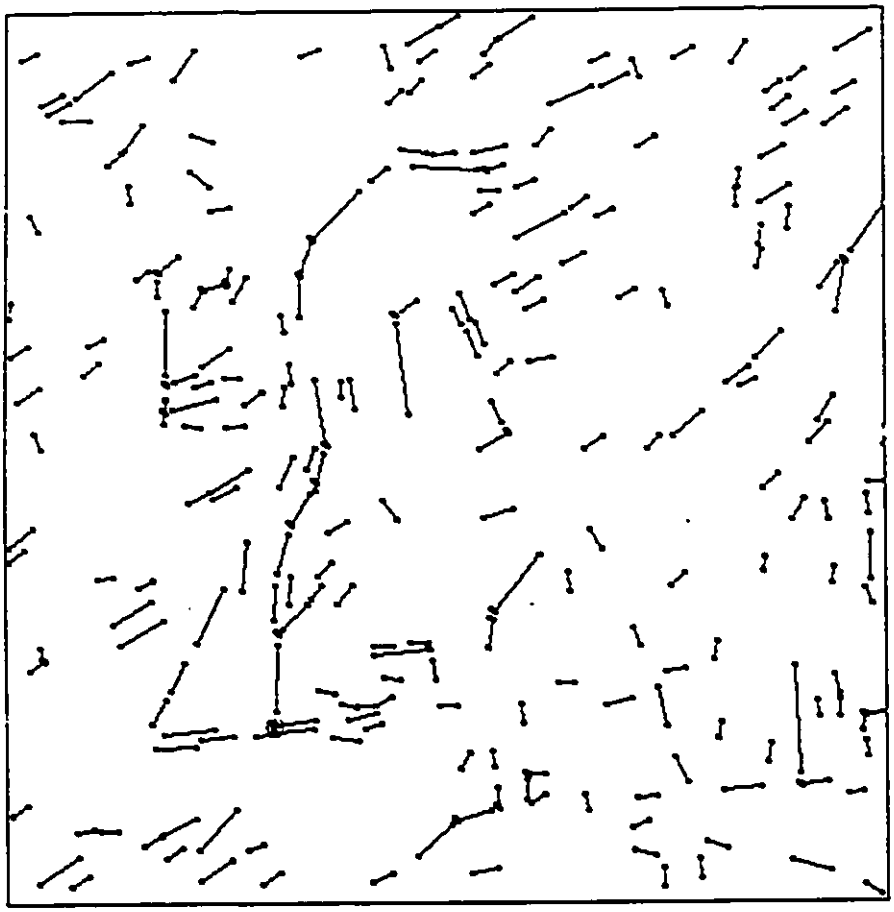


Figure 6.35: Result of the matching process of images (1) and (2) is shown above.

6.5 Conclusion

This chapter showed the results of the different phases of the recognition process and the invariance of the algorithms employed to translation, rotation and scaling deformation. The control points parameters were used and a spline model of order 4 was found to be adequate for the representation of the image. The usefulness of the methods described in the previous chapters was illustrated through an example of matching and registration of a scene that contained geometrically distorted images.

Chapter 7

Conclusion

Image matching is considered one of the most challenging problem in computer vision. It has received tremendous interest from a large number of researchers.

The work discussed in this thesis is an implementation of a generic image recognition system. The system was designed with a modular approach to ease its extension, and the addition of new methods and algorithms in the future. The design focused on pursuing efficient methods for feature extraction, representation and matching. The first objective of the project was to design and implement a system that can extract and model generic features in a number of situations determined by the combination of possible geometrical deformation and occlusion. A second objective was to have the system recognize and detect changes in a sequence of images for the same scene but taken at different times in a short and predictable period of time so that it can be applied in real time vision problems.

We managed to fulfill the objectives of the project, since the vision system can deal with images with different geometrical orientation. It worked efficiently for a

number of images under different orientations, translations and scaling transformations. The main original features of the system, and the contributions of the author to the solution of this problem are:

- The feature extraction and enhancement algorithms which simplify the matching process and make sure that only the desired features feed the matching phase. They also enable the system to deal with non-rigid shapes.

- The spline model used to describe the boundary and to build a model for the image is another original contribution of the author. Most of the previous work done to model shapes with spline was based only on closed shapes (e.g circular and polygonal shapes), since it is easier to model it mathematically. The spline parameters provide a number of advantages: an accurate description of shape using a small number of parameters, limited storage requirements, size and orientation invariance. .

- The modular approach used in the design and implementation helped in the maintenance and extension of the system, since it was relatively easy to make changes and add algorithms to the system. This approach made it convenient to experiment with several algorithms for the various stages of the recognition process.

- The matching was also enhanced by using a two stage matching process in order to speed up the matching phase. First the matching was performed on the corners that are extracted from the images, since they are fewer in number than the other extracted features. Then the second stage was performed to confirm on the matched features and to find the global change between the images through the computation of a global mapping transformation. This mapping transformation is computed using a recursive parameter estimation

algorithm.

7.1 Future Work

For future research the following subjects can improve the system to make it more generic, and might extend it to support more recognition algorithms:

- Symbolic processing of the shape to obtain a more intelligent sub-part decomposition. The symbolic processing of the shape will also enable the system to deal more effectively with 3-D objects and to make the pruning step in the recognition phase more effective.
- Improvement of the object modeling to make the system aware of the particular characteristics of non-rigid objects and varying shape (as in the problem of hand-writing recognition).
- Introduce powerful indexing methods in shape modeling so that the performance of the system does not degrade when the number of object models increases. One of the methods considered is structural indexing described in [46].
- Improve the hardware performance by implementing the algorithms on digital signal processors for real time applications.
- Introduce robust parallel algorithm in order to speed up the processing time.

Bibliography

[1] B. D. Anderson and J. D. Moor. *Optimal Filtering.* Prentice-Hall, 1979.

[2] K. J. Astrom and P. Eykhoff. System identification - a survay. *Automatica*, 7:123–161, 1971.

[3] D. H. Ballard. Generalization the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, July 1980.

[4] R. Bartels and B. A. Barsky. *An Introduction to Splines for use in Computer Graphics and Geometric Modeling.* Morgan Kaufmann Publishers, Inc, 1986.

[5] R. Bernstein. Image geometry and rectification. In *Manual of Remote Sensing*, chapter 21. American Society of Photogrammetry, 2ed edition, 1983.

[6] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Tras. Pattern and. Machine Intel.*, PAMI-14(2):239–256, February 1992.

[7] J. Bretschi. *Automated Inspection Systems for Industry.* IFS Publication Ltd. Bedford U.K, 1981.

[8] F. C.Billingsley. Data processing and reprocessing. In *Manual of Remote Sensing*, chapter 17. American Society of Photogrammetry, 2ed edition, 1983.

[9] F. Cohen and Jin-Yinn Wang. Part i: Modeling image curves using invariant 3-d object curve model - a path to 3d recognition and shape estimation form image contours. *IEEE Trans. Pattern Anal. Machine Intel.*, PAMI-16(1):1–12, January 1994.

[10] L. S. Davis. Shape matching using relaxation techniques. *IEEE Trans. Pattern Anal. Machine Intel.*, PAMI-1(1):60–72, January 1979.

[11] R. Deutsch. *Estimation Theory.* Prentice Hall, Englewood Cliffs, N. J., 1965.

[12] M. Dhome and M. Richtin. Determination of the attitude of 3d objects from a single prespective view. *IEEE Trans. Pattern Anal. Machine Intel.*, PAMI-11(12):1265–1278, Dec. 1989.

[13] J. G. Dunham. Optimum uniform piecewise linear approximation of planer curves. *IEEE Trans. Pattern Anal. Machine Intel.*, PAMI-8(1):67–75, January 1986.

[14] A. C. Sanderson F. C. A. Groen and J. F. Schlag. Symbol recognition in electrical diagrams using probabilistic graph matching. *Pattern Recognition Letters*, Vol. 3:343–350, 1985.

[15] T. L. Faber and E. M. Stokely. Orientation of 3d structure in medical images. *IEEE Trans. Pattern Anal. Machine Intel.*, PAMI-10(5):626–633, September 1988.

[16] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design - A practical Guide.* Academic Press, Inc, 1990.

[17] O. D. Faugeras and N. Ayache. Hyper: A new approach for the recognition and positioning of two-dimensional objects. *IEEE Trans. Pattern Anal. Machine Intel.*, PAMI-8(1):44–54, January 1986.

[18] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automatic cartography. *Communications of the ACM*, Vol. 24(6):381–395, June 1981.

[19] K. S. Fu. *Syntactic methods in Pattern Recognition*, chapter 4. Academic Press, 1974.

[20] A. Rosenfeld G. J. Vanderburg. Two stage template matching. *IEEE Trans. Systems Man., and Cybernetics*, SMC-2:104–107, 1977.

[21] D. M. Gavrila and F. C. A. Groen. 3d object recognition from 2d images using geometric hashing. Master's thesis, Computer Science., Vrije Universiteit, Amsterdam, The Netherlands, 1990.

[22] V. E. Giuliano, P. E. Jones, and G. E. Kimbelland B. Stein. Automatic patterrn recognition by gestalt method. *Inform. Contr.*, 4:332–345, 1961.

[23] Goddard Space Flight Centre, Greenbelt, Md. *Landsat Data Users Handbook, Document no. 76SD4258*, 1976.

[24] A. Goshtasby. Template matching in rotated images. *IEEE Trans. Pattern Anal. Machine Intel.*, PAMI-7(3):338–344, May 1985.

[25] A. Goshtasby and A. K. Jain. Automatic digital image registration. In *Proceedings in 8th Symposium Mach. Processing Remote Sensed Data*, pages 347–352, 1982.

[26] E. L. Hall. *Computer Image Processing and Recognition*. Academic Press, New York, 1979.

[27] R.M. Haralick and L.G. Shapiro. *Computer and Robot Vision*, volume 2. Addison-Wesley, 1992.

[28] T. C. Hsia. *System Identification.* Lexington Books, 1977.

[29] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Trans. Inform. Theory*, IT-18:179-187, 1962.

[30] M. Hueckel. An operator which locates edges in digitized pictures. *J. Assoc. Comput. Mach.*, 18(1):113–125, January 1971.

[31] I.Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. *Appl. Math.*, 4:45–54, 1946.

[32] A. K. Jain. *Fundamental of Digital Image Processing.* Prentice Hall, 1989.

[33] H. Joo and R. M. Haralick. 2d and 3d pose estimation. Technical Report CH2614-6, IEEE, 1988.

[34] S. Huang K. S. Arun and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Tras. Pattern and. Machine Intel.*, PAMI-9(5):698–700, September 1980.

[35] M. Merickel. 3d recognition: The registration problem. *Computer Vision, Graphics and Image Processing*, 8(42):206–219, 1988.

[36] M. Mortenson. *Geometric Modeling.* Wiley, 1985.

[37] Y.Liu O. D. Faugeras and T. S. Haung. Determination of camera location from 2d to 3d line and point correspondence. *IEEE Trans. Pattern Anal. Machine Intel.*, PAMI-12(1):28–37, January 1990.

[38] T. Pavlidis. Polygonal approximation by newton method. *IEEE Trans. Compute.*, 26(8):800–807, June 1977.

[39] T. Pavlidis. *Structural Pattern Recognition*, chapter 3. Springer-Verlag, 1977.

[40] T. Pavlidis. Algorithms for shape analysis of contours and waveforms. *IEEE Trans. Pattern Anal. Machine Intel.*, 2(4):311-318, July 1980.

[41] T. Pavlidis. *Algorithms for Graphics and Image Processing.* Computer Science Press, 1982.

[42] R. Woods R. C. Gonzalez. *Digital Image Processing*, chapter 6. Addison- Wesley Publishing, 1992.

[43] P. E. Hart R. O. Duda. *Pattern Classification and Scene Analysis.* Wiley, 1973.

[44] Gerard Medioni Ramakant Nevatia. Matching images using linear features. *IEEE Trans. Pattern Anal. Machine Intel.*, PAMI-6(6):677-687, November 1984.

[45] A. Rosenfeld and E. Jonhston. ngle detection on digital curves. *IEEE Trans. Compute.*, C-22:875-878, September 1973.

[46] F. Stein and G. Medioni. Structural indexing: efficient 2-d object recognition. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-14(12):1198-1204, 1992.

[47] H. S. Barid T. Pavlidis, S. Kahan. On the recognition of printed characters of any font and size. *IEEE Trans. Pattern Anal. Machine Intel*, 9(4):301-312, July 1980.

[48] M. A. Fischler J. M. Tenenbaum and H. C. Wolf. Detection of roads and linear structures in low-resolution aerial imagery using a multi source knowledge integration technique. *Computer Graphics and Image Processing*, 15(3):201-223, 1981.

[49] M. Unser and Akram Aldruobi. A multiresolution image registration procedure using spline pyramids. *Mathematical Imaging*, 166(2034):160-169, 1993.

[50] K. Wal and P. E. Danielsson. A fast method for polygonal approximation of digital curves. *Computer Graphics and Image Processing*, 28:220–227, 1984.

[51] P. Van Wie and M. Stein. A landsat digital image rectification system. *IEEE Trans. on Geoscience Electronics*, GE-15(3):130–137, 1977.

[52] R. Y. Wong and E. L. Hall. Scene matching with invariant moments. *Compute. Graphics Image Processing*, 8:16–24, 1978.

[53] R. Desiche Z. Zhang and O.Fuagras. A robust technique for matching two uncalibrated images through the recovery of unknown epipolar geometry. Technical Report 2273, Institute National De Recherche en Informatique et en Automatic (INRIA), May 1994.