Statistical Methods for Point Pattern Matching

Bin Luo

B.Eng., M.Eng.

Submitted for the degree of Doctor of Philosophy

Department of Computer Science

THE UNIVERSITY of York

July 30, 2001

Abstract

Point pattern matching is a topic of pivotal importance in computer vision. It's applications can be found in many areas such as pose estimation, stereo matching, three-dimensional reconstruction and motion tracking. There are two kinds of point pattern matching technique. Alignment aims to estimate the transformation parameters. Correspondence is concerned with point labelling. In this thesis, we aim to develop a unified statistical framework to solve the point set alignment and correspondence problems.

The starting point is to develop a robust corner detector which extracts features which can be used to represent salient image structure. We choose corners as the feature points for matching in this thesis since they are the dominant ones in many types of images. Based on a magneto-static analogy of images, we demonstrate that corners are located at the saddle points of the magnitude of the vector potential. We also describe a template-based method for locating the saddle-points.

The second contribution is to cast the problem of point-set alignment via Procrustes analysis is cast into a maximum likelihood framework using the EM algorithm. The aim is to improve the robustness of Procrustes alignment to noise and clutter. By constructing a Gaussian mixture model over the missing correspondences between individual points, we show how alignment can be realised by applying singular value decomposition to a weighted point correlation matrix. The method can be used to match unlabelled point-sets of different size.

We extend the method to non-rigid transformations using point distribution models. The point distribution models can then be aligned to incomplete point-sets when prior knowledge concerning point correspondence is unknown. Since this is a problem posed in terms of hidden variables (the correspondences), we

couch the recovery of maximum likelihood alignment parameters using the apparatus of the EM algorithm.

Thirdly, we show how point correspondence matching may be realized using a relational graph matching process. We make two novel contributions. Commencing from a probability distribution for matching errors, we show how the problem of graph-matching can be posed as maximum likelihood estimation using the apparatus of the EM algorithm. Our second contribution is to cast the recovery of correspondence matches between the graph-nodes in a matrix framework. Here we demonstrate that the method offers comparable performance advantages over more computationally demanding methods.

Finally, a unified matching framework is developed which integrates both point set alignment and correspondence. The utility measure underpinning the work is the cross-entropy between probability distributions for alignment and correspondence assignment errors. We show how alignment parameters and correspondence probabilities can be located using dual singular value decompositions.

Contents

List of Figures

List of Tables

Symbols

G_σ	Gaussian kernal of width σ
$\vec{E}$	Canny edge-map
$\vec{A}$	Vector potential
$\mathcal{H}$	Hessian matrix
H	Gaussian curvature
K	Mean-curvature/constant
$\mathcal{F}$	Fourier transform
$\mathcal{F}^{-1}$	Inverse Fourier transform
Φ	Transformation parameter matrix(Also for eigenvector matrix)
$\Phi^{(n)}$	Transformation parameter matrix at the nth iteration
Θ	Rotation matrix
$\mathcal{D}$	Data point index set
$\mathcal{M}$	Model point index set
$\vec{u}_i$	Data point position vector, $\vec{u}_i = (x_i, y_i)^T$
$\vec{v}_j$	Model point position vector, $\vec{v}_j = (x_j, y_j)^T$
$\mathbf{w}$	Data point set,$\mathbf{w} = \vec{u}_i, i \in \mathcal{D}$
$\mathbf{z}$	Model point set,$\mathbf{z} = \vec{v}_j, j \in \mathcal{M}$
D	Data point position matrix, $D = (\,\vec{u}_1 \mid \quad \vec{u}_2 \mid \quad .... \quad \mid \vec{u}_{\mid D \mid}\,)$
	Also for data graph adjacency matrix in chapter 5
M	Model point position matrix, $M = (\,\vec{v}_1 \mid \quad \vec{v}_2 \mid \quad .... \quad \mid \vec{v}_{\mid M \mid}\,)$
	Also for model graph adjacency matrix in chapter 5
P_e	Correspondence error probability
V_D	Set of data graph nodes
E_D	Set of data graph edges
G_D	Data graph, $G_D = (V_D, E_D)$

V_M	Set of model graph nodes
E_M	Set of model graph edges
G_M	Model graph, $G_M = (V_M, E_M)$
$\mathcal{L}$	Complete likelihood function
$\mathcal{E}$	Error measure
U	Right eigenvector of singular value decomposition(SVD)
V	Left eigenvector of SVD
Δ	Singular value diagonal matrix of SVD
$P_{i,j}$	*a posteriori* probabilities that data point i is in alignment with model point j
$Q_{i,j}$	*a posteriori* probabilities that data point i is in correspondence with model point j
W	Weight matrix
ζ	Weight matrix
μ_D	Centroid of data point set
μ_M	Centroid of model point set
Σ_D	Variance-covariance matrix of the data point set
Σ_M	Variance-covariance matrix of the model point set
λ_i	ith eigenvalue of a covariance matrix
$\vec{\phi}^{\lambda_i}$	Eigenvector corresponding to ith eigenvalue λ_i
$\vec{r}$	Modal coefficient vector of the PDM model
ψ_i^{λ}	Displacement vector of landmark i corresponding to eigenvector λ
Δ_j	Displacement matrix of the jth landmark
S	Correspondence matrix

Acknowledgement

I feel very fortunate to have had Professor Edwin R. Hancock as my thesis supervisor. I am impressed by his attitude to his research, his knowledge of computer vision, his kind advise and support to me and his students. My thanks also go to Dr Richard Wilson for his assessments of my work and helpful discussions.

My financial support comes from an Overseas Research Studentship, an Overseas Studentship of the University of York and a living bursary from Edwin's research group without which this work would have been impossible.

I would like to thank my colleagues at the University of York and Anhui University for their understanding and support. I have so many friends especially in China and in England. It is impossible for me to forget their friendship, encouragement and support during the course of my study.

I would like to thank my parents for the support which they have given me in my education over the decades.

My beloved wife and daughter must of course be thanked for their understanding and support during my period of studying in England.

Declaration

I declare that all the work in this thesis is solely my own except where attributed and cited to another author. Some of the material in the following chapters has been previously published, a full list of the publications is listed in the next page.

List of Publications

The following are the list of publications that have been produced during the course of my Ph.D. research.

2002

- B. Luo, E.R. Hancock, Matching point-sets using Procrustes alignment and the EM algorithm. In *Image Vision Computing* Vol.20, pages 377-396. 2002
- B. Luo, R.C. Wilson, E.R. Hancock, Eigenspaces for Graphs. To appear in *International Journal of Image and Graphics, 2002*
- B. Luo, R.C. Wilson, E.R. Hancock, Spectral Feature Vectors for Graph Clustering To appear in the Proceedings of *9th International workshop on Structural and Syntactic Pattern Recognition and 4th International workshop on Statistical Techniques in Pattern Recognition(S+SSPR2002)* August 6-9, Windsor, Canada, 2002
- B. Luo, R.C. Wilson, E.R. Hancock, Graph Spectral Approach for Learning View Structure To appear in the Proceedings of *16th International Conference on Pattern Recognition(ICPR2002)* August 11-15, Quebec, Canada, 2002
- B. Luo, R.C. Wilson, E.R. Hancock, The Independent and Principal Components of Graph Spectra To appear in the Proceedings of *16th International Conference on Pattern Recognition(ICPR2002)* August 11-15, Quebec, Canada, 2002
- B. Luo, R.C. Wilson, E.R. Hancock, Eigenspaces for Graphs. In the Proceedings of *Fifth Asian Conference on Computer Vision(ACCV2002)*

January 22-25, Melbourne, Australia, Pages 487-492, 2002

Awarded the conference best paper award.

- B. Luo, E.R. Hancock, Structural Graph Matching using the EM Algorithm and Singular Value Decomposition. In the Proceedings of *Fifth Asian Conference on Computer Vision(ACCV2002)* January 22-25, Melbourne, Australia, Pages 75-80, 2002

- B. Luo, R.C. Wilson, E.R. Hancock, Spectral Embedding of Graphs. To appear in the Proceedings of *The Seventh Computer Vision Winter Workshop(CVWW2002)* February 4-7, Bad Aussee, Austria, 2002

2001

- B. Luo, E.R. Hancock, Structural graph matching using the EM algorithm and singular value decomposition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.23, No.10, pages 1120-1136. 2001.

- B. Luo, A. Robles-Kelly, A. Torsello, R.C. Wilson, E.R. Hancock, A Probabilistic Framework for Graph Clustering. In the proceedings of *IEEE Computer Vision and Pattern Recognition(CVPR2001)*, I:912-919, 2001.

- B. Luo, A.R. Kelly, A. Torsell, R.C. Wilson and E.R. Hancock, Learning shape categories by clustering shock trees, In the proceedings of *International conference on image processing (ICIP2001)* ,October 7-10, pages 672-675, 2001

- B. Luo, E.R. Hancock, A Robust Eigendecomposition Framework for Inexact Graph Matching. In the proceedings of *11th International Conference on Image Analysis and Processing(ICIAP2001)*, September 26-28, pages 465-470. 2001.

- B. Luo, E.R. Hancock, Relational Constraints for Point Distribution Models, In *Springer Lecture Notes in Computer Science* 2124, pages 646-656, 2001.

- B. Luo, A.R. Kelly, A. Torsell, R.C. Wilson and E.R. Hancock, Discovering Shape Categories by Clustering Shock Trees, In *Springer Lecture Notes in Computer Science* 2124, pages 151-160, 2001.

- B. Luo, A.R. Kelly, A. Torsell, R.C. Wilson and E.R. Hancock, Clustering Shock Trees, In the proceedings of *The 3rd workshop on graph-based representations in Pattern Recognition (GbR2001)*, May 23-25, 217-228, 2001

- B. Luo, E.R. Hancock, Augmenting Point Distribution Models with Relational Constraints, In the Proceedings of *12th Scandinavian Conference on Image Analysis (SCIA2001)*, June 11-14, 41-48, 2001

2000

- B. Luo, E.R. Hancock, Alignment and correspondence using singular value decomposition. In *Lecture Notes in Computer Science*, Vol. 1876, pages 226-235. 2000.

- B. Luo, E.R. Hancock, Symbolic graph matching using the EM algorithm and singular value decomposition. In *International Conference on Pattern Recognition (ICPR2000)*, pages 141-144. 2000.

1999

- B. Luo, A.D.J. Cross and E.R. Hancock, Corner Detection via Topographic Analysis of Vector-Potential. In *Pattern Recognition Letters*, Vol.20, pages 635-650. 1999.

- B. Luo, E.R. Hancock, Feature Matching with Procrustes Alignment and Graph Editing. In *Seventh International Conference on Image Processing and Its Applications(IEE)*, pages 72-76. 1999.

- B. Luo, E.R. Hancock, Matching Point-sets Using Procrustes Alignment and the EM Algorithm. In *British Machine Vision Conference (BMVC1999)*, pages 43-52. 1999.

- B. Luo, E.R. Hancock, Procrustes Alignment with the EM Algorithm. In *8th International Conference on Computer Analysis of Images and Patterns (CAIP1999)*, pages 623-631. 1999.

1998

- B. Luo, A.D.J. Cross and E.R. Hancock, Corner Detection via Topographic Analysis of Vector Potential. In *British Machine Vision Conference (BMVC1998)*, pages 567-577. 1998.

- B. Luo, A.D.J. Cross and E.R. Hancock, Corner Detection Using Vector Potential. In *International Conference on Pattern Recognition (ICPR1998)*, pages 1018-1021. 1998.

Chapter 1

Introduction

1.1 The Problem

A fundamental problem of object recognition is to match a known object model to an image which contains that object. There are two kinds of techniques that have been developed in the computer vision community to solve this problem. The first of them is the alignment technique which aims to recover the transformation between the object model and the object image. The transformation parameters can then be used to deform the model so as to fit the object image under a predefined error model. The best fit is achieved when the error measurement is minimized. Multimodel image alignment such as integrating structural information from CT, MRI or PET are important practical examples of the application of image alignment. Another example is to align a model elicited from non pictorial information(e.g. a cartographic map) to aerial images suspected to contain an instance of that model.

The second matching technique is correspondence assignment. This aims to recover one-to-one correspondences between the feature sets extracted from both model and data images. There are many applications that rely on this kind

of technique. Examples include motion analysis which extracts the movement parameters from the matched features and stereopsis which aims to recover the depth or shape from the disparities between the feature sets in pairs of images.

Another issue addressed in this thesis is that of extracting features from both model and data images. Although a large number of techniques have been proposed, a method of extracting reliable features which is robust to severe contamination is always required.

1.2 Goals

The ultimate goal of this thesis is to develop an useful matching algorithm which can robustly find feature correspondences between two feature point sets and recover the transformation parameters simultaneously. To achieve this, we consider the following five subgoals.

- Robust feature extraction: We shall concentrate on developing a corner detector which is robust to noise present in the images.

- Robust alignment of two point patterns with similarity deformation: Here an alignment algorithm which is robust to noise and clutter is desired. The algorithm should use as little prior correspondence information as possible.

- Robust alignment of two point patterns with non-rigid deformation: The aim is to develop an algorithm which is able to fit a point distribution model to a given point set.

- Robust correspondence assignment algorithm: Structural information is explored to establish matching between the two point patterns while placing minimal reliance on prior alignment information.

- A statistical framework for the unification of both alignment and correspondence: By understanding the relationship between alignment and correspondence in point pattern matching, we aim to develop a unified iterative algorithm which simultaneously recovers the transformation parameters and correspondence matches.

1.3 Thesis Overview

After defining the problem domain and the goals of the thesis in Chapter 1, we shall give a brief review of the related literatures in Chapter 2. The review covers the literature on corner detection, point pattern alignment, deformable models and graph matching.

Based on a topographic representation of images, Chapter 3 presents a corner detector which converts the problem of corner detection into extracting saddle points in a vector potential field. We describe a template-based method for localising the saddle-points.

Chapter 4 discusses the development of an iterative point pattern alignment method which deals with point sets undergoing a Euclidean or similarity transformations. The method performs traditional Procustes alignment using the Expectation-Maximisation(EM) algorithm.

In Chapter 5, correspondence assignment is considered as a graph matching problem which exploits the structural relationship between the points in both model and data point sets. A mixture model for graph matching and an EM algorithm for updating the matches are introduced.

A unified algorithm for alignment and correspondence is described in Chapter 6. The algorithm developed in the previous chapters are combined to develop a matching algorithm which interleaves the estimation of alignment parameters

and correspondence matches. Here the hope is that by combining the two steps, the recovered matches will be more robust to noise and clutter. The problem of aligning two point sets with complex deformations is considered. The alignment is achieved by using non-rigid deformation represented using the point distribution model(PDM) introduced by Cootes et al(Cootes et al., 1995).

Chapter 7 gives a brief conclusion of the advantages and shortcomings of the thesis and some aspects which we think can be further explored based on the thesis.

Throughout the thesis, after describing the theoretical development of the algorithms, we present the experiment results on synthetic data to reveal the sensitivity measurement and on real images to show the potential applications of the algorithms.

Chapter 2

Literature review

In this chapter, we will review the literature which is related to the research presented in the thesis. The review covers the pre-processing step - feature point detection, point pattern alignment techniques, deformable models and graph matching methods.

2.1 Point feature detection

Although there are different kinds of point features which can be extracted from images, the most important point features are corners. Line junctions are another kind of point feature which can be extracted from pre-detected line patterns (Matas and Kittler, 1993; Deschenes and Ziou, 2000). The strategies for detecting different line junctions, such as T,L,X and Y junctions, are somewhat similar to the ones for detecting corners. Point features can also be extracted from the middle point of line segments (Wilson and Hancock, 1997) or regions. In this section we will concentrate on corner detection.

Corners are important dominant points in digital images. In many computer vision tasks, such as image registration(Yang and Cohen, 1999), image matching

(Costabile et al., 1985), object recognition (Liu and Srinath, 1990b; Han and Jang, 1990) and motion analysis (Dreschler and Nigel, 1992), accurate corner detection is essential. Broadly speaking, there are two corner-detection strategies adopted in literature. The first of these is based on the analysis of pre-segmented contours, while the second is based on the differential analysis of the raw gray-scale image. However, in both cases it is the rate of change of contour angle that is used to characterise corner features.

In the case of contour-based corner detection from pre-segmented contours there are three processing steps. Firstly, the image is segmented into contour features. Secondly, boundaries of the object in the image are extracted and chain coded. Finally, algorithms are developed for identifying corners in the chain-codes. A common technique is to search for corners at the intersection points or junction points between straight line segments (Xie et al., 1993). In fact, the use of chain codes to provide a digital characterisation of corners abound in the literature (Freeman and Davis, 1977; Beus and Tiu, 1987; Koplowitz and Plante, 1995; Rosenfeld and Johmston, 1973; Rosenfeld and Weszka, 1975). A good review is provided by Liu (Liu and Srinath, 1990a). However, it must be stressed that the weakness in the contour-based method of corner detection is the prior availability of a reliable image segmentation.

Gray-scale corner detection algorithms can be divided into two groups. Template based corner detectors (Rangarajan et al., 1989; Mehrotra et al., 1990) exploit the similarity between a given template of specific orientation for each image sub-window. Because multiple orientation templates are used, the technique is computationally expensive. Gradient-based corner detectors (Kitchen and Rosenfeld, 1982; Singh, 1990; Deriche and Giraudon, 1993; Noble, 1988; Wang and Brady, 1995; Harris and Stephens, 1988), on the other hand, rely on measuring the curvature of an edge that passes through a given image neighbour-

hood. The strength of the corner response depends on both the edge strength and the rate of change of edge direction. Gradient-based corner detection techniques are more likely to respond to noise than their contour-based counterparts, and often perform quite poorly.

Focusing in more detail on gradient-based corner detection, Wang and Brady (Wang and Brady, 1995) describe a curvature based corner detector which measures the strength of corners in terms of the so-called "total curvature". The total curvature is proportional to the second derivative of the image intensity along the edge tangent direction, and is inversely proportional to the edge strength. The method offers the attractive feature of exploiting an explicit measure of curvature as well as providing a degree of false corner suppression. However, one of its weaknesses is that it can not control false responses when significant image noise is present. The popular Plessey operator(Harris and Stephens, 1988) is another curvature based corner detector which is based only on the first derivatives of the image intensity. Although it is quite robust with respect to noise, it can only perform well on 'L'-junctions or right-angle corners. The localization of other junction types is poor. The SUSAN corner detector (Smith and Brady, 1997), on the other hand, performs a local template comparison. A degree of robustness is achieved by computing template correlation statistics for a circular mask centred on the corner candidates. In particular, it examines the number of pixels within the mask which have a similar brightness to the template. The number of agreement counts is taken as a vote for the corner hypothesis. High voting pixels are taken to have a significant corner strength. The method also performs a non-maximum suppression test to reject false-positives.

2.2 Point pattern matching

The problem of point pattern alignment has attracted sustained interest in both the vision and statistics communities for several decades (Kendall, 1984; Werman and Weinshall, 1995; Sprinzak and Werman, 1994; McReynolds and Lowe, 1996; Torr and Murray, 1997). The simplest way of aligning point-sets is to recover the transformation that minimises the mean-squared distance under a predefined transformation group. There are numerous examples of such methods in the computer vision literature. The method has been applied to Euclidean, affine (Werman and Weinshall, 1995; Shapiro et al., 1995) and perspective transformations (McReynolds and Lowe, 1996). Moreover, several authors have used robust statistics to overcome measurement noise and contamination (Torr and Murray, 1997; Shapiro and Brady, 1995). In the statistics literature, an elegant formal framework for alignment is provided by Procrustes normalisation (Dryden and Mardia, 1998). For the rigid transformation the Procrustes alignment is a simple three-step process. It commences by using the mean point positions to center the two sets. Next, the point-sets are scaled using their variance. Finally, the rotation is recovered by performing singular value decomposition on the inner product of the point-position matrices so as to maximise the correlation or overlap of the two point-sets. This process has been generalised to transformations that can be represented on projective manifolds by Kendall (Kendall, 1984). Unfortunately, the conventional Procrustes methods have two restrictions. One of them is that the points in the data and model sets must be labelled *a priori*. Another limitation is that the number of points in the data and model sets must be identical.

Alignment can therefore be regarded as a template matching method. In other words, it overlooks information concerning the relational arrangements of the points and is not invariant under the relevant transformation group. It is for

these reasons that the problem of correspondence has attracted more attention in the computer vision literature. Here Ullman (Ullman, 1976) was one of the first to recognise the importance of exploiting rigidity constraints in the correspondence matching of point-sets. Recently, several authors have drawn inspiration from Ullman's ideas in developing general purpose correspondence matching algorithms using the Gaussian weighted proximity matrix. For instance Scott and Longuet-Higgins (Scott and Longuet-Higgins, 1991) locate correspondences by finding a singular value decomposition of the inter-image proximity matrix. Shapiro and Brady (Shapiro and Brady, 1995; Shapiro et al., 1995), on the other hand, match by comparing the modal eigenstructure of the intra-image proximity matrix. In fact these two ideas provide some of the basic groundwork on which the deformable shape models of Cootes *et al* (Cootes et al., 1995) and Sclaroff and Pentland (Sclaroff and Pentland, 1995) build. This work on the coordinate proximity matrix is closely akin to that of Umeyama (Umeyama, 1988) who shows how point-sets abstracted in a structural manner using weighted adjacency graphs can be matched using an eigen-decomposition method. These ideas have been extended to accommodate parameterised transformations (Umeyama, 1991) which can be applied to the matching of articulated objects (Umeyama, 1993). More recently, there have been several attempts at modeling the structural deformation of point-sets. For instance, Amit and Kong (Amit and Kong, 1996) have used a graph-based representation (graphical templates) to model deforming two-dimensional shapes in medical images. Lades *et al* (Lades et al., 1993) have used a dynamic mesh to model intensity-based appearance in images. Finally, several authors have attempted to formalise the use of rigid body constraints. Examples include the work of Morgera and Cheong (Morgera and Cheong, 1995) and that of Hu and Ahuja (Hu and Ahuja, 1994) who have combined geometric, rigidity and disparity constraints.

In a recent paper Cross and Hancock (Cross and Hancock, 1998) have observed that the dichotomy between alignment and correspondence is an artificial one. The reason for this is as follows. Broadly speaking the aim of point pattern matching is to recover the transformation between image and model co-ordinate systems. In order to estimate the transformation parameters a set of correspondence matches between features in the two co-ordinate systems is required. In other words, the feature points must be labelled. Posed in this way there is a basic chicken-and-egg problem. Before good correspondences can be estimated, there need to be reasonable bounds on the transformational geometry. Yet this geometry is, after all, the ultimate goal of computation. This problem is usually overcome by invoking constraints to bootstrap the estimation of feasible correspondence matches. If reliable correspondences are not available, then a robust fitting method must be employed (Shapiro and Brady, 1995; Shapiro and Brady, 1992). This involves removing rogue correspondences through outlier rejection.

To overcome this dichotomy, Cross and Hancock (Cross and Hancock, 1998) developed a dual-step EM algorithm. Here the problem of alignment parameter recovery was posed as one of maximum likelihood estimation. The point correspondences were treated as missing data. In the expectation step of the algorithm the current set of alignment errors were used to compute correspondence probabilities. In the maximisation step of the algorithm the correspondence probabilities were used to weight contributions to an expected log-likelihood function for the required alignment parameters. The expectation and maximisation steps were interleaved and iterated to convergence. In addition, the algorithm uses constraints on the consistent pattern of correspondences provided by a Delaunay triangulation of the point-sets. The probability of correspondence match is used to gate contributions to the expected log-likelihood function for parameter estimation. Unfortunately, due to the high computational requirement, the algorithm can

only handle relatively small point sets. In addition, the method places the role of correspondence and alignment on an asymmetric footing. Structural correspondences are regarded as high-level constraints that constrain the alignment process.

2.3 Deformable models

The alignment method outlined previously can only accommodate the rigid point set deformations which apply under Euclidean, affine and perspective transformations. A more flexible types of deformations are non-rigid in nature. Non-rigid effects are the domain of deformable models.

Deformable models can be broadly divided into two categories: free-form deformable models and parametric deformable models. In the first class, no global structure of the model is used. The model is only constrained by local smoothness and continuity (Kass et al., 1988; Terzopoulos et al., 1988; Moshfeghi et al., 1994; Cohen et al., 1992). Because there is no domain-specific structure in the model, it can represent an arbitrary shape as long as the continuity and smoothness constraints are satisfied.

In the second class, a parameterized deformable template is used when prior information of the object shape and its model variations are available. The idea is that through an analysis of the geometric structure of an object, a parameterized template can be constructed which assembles the main geometrical shape structure of the object. Parameterized deformable template methods fall into three categories. The first method represents the deformable model as a collection of parameterized curves. Examples here include the work of Yuille(Yuille et al., 1992) and Lakshmanan(Lakshmanan and Grimmer, 1996). The second method represents the deformable template as the image of a prototype template

under a parametric mapping(Jain et al., 1996). The final subclass is the point distribution model(PDM) (Cootes et al., 1994) which will be described in more detail in the next subsection.

Deformable models have applications in many kinds of computer vision tasks. Examples include image segmentation(Yezzi et al., 1997; Cohen, 1991; Miller et al., 1991; McInerney and Terzopoulos, 1995; Singh et al., 1993), tracking (Leymarie and Levine, 1993; Wang and Lee, 1994; Delanges et al., 1995), matching(Moshfeghi, 1991; Moshfeghi et al., 1994; Gueziec and Ayache, 1994; Feldmar and Ayache, 1996; Thirion, 1994; Larvallee and Szeliski, 1994), recognition(Jolly et al., 1996; Ahmad et al., 1997; Delibasis and Undrill, 1994; Schwarzinger et al., 1995), registration(Davatzikos, 1997; Davatzikos, 1996) and motion analysis(Park et al., 1996; Gwydir et al., 1994; Terzopoulos et al., 1988).

In this thesis, we focus on the third class of deformable models, i.e., the point distribution models.

2.3.1 Point Distribution Models

Point distribution models(PDM's) are statistical models of shape and shape variation generated from sets of examples. Given a set of training images, landmark points for the shapes in the images are detected and labeled. The landmarks in training shapes should be consistently labelled. Each shape can then be described by a vector which represents the concatenated coordinates of the complete set of the landmarks. The mean shape is calculated from the average of the set of shape vectors. After calculating the covariance about the mean, the eigenvectors of the covariance matrix are used to represent the modes of variation for the deformable model. Only those eigenvectors corresponding to the largest eigenvalues are chosen as shape modes. Any shape in the training set can be approximated using

the mean shape and a weighted sum of the eigenvectors. The weights are treated as shape parameters.

The model may be extended to capture the statistics of grey scale level appearance. To do this local grey scale profile models are built for each landmark point. Similarly, mean grey scale profiles for each point are calculated. A principal component analysis(PCA) of the covariance matrix of the deviations is used to characterise the main modes of grey-scale variation. A vector of parameters for the leading eigenvalues are calculated. They are called local grey level model parameters.

Point distribution models can be fitted to a new image containing the shape. Firstly, a region of the image around each point is examined to find a suggested displacement which will lead the point move to a better position in the image. Secondly, the adjustments to the pose and scale and to the shape parameters of the PDM of each point are calculated using these displacements. Finally, the model parameters are updated to fit the model to the image. Unfortunately, the algorithm can only be used when there are no clutter point present, that is to say there are the same number of points in the data point set and the model point sets. In addition, the points must be labelled in advance. Finally, potentially useful structural information is overlooked. This information may prove useful in the reject of outliers or clutter.

2.4 Graph Matching

As identified in the previous section, the structure of point sets can provide important information which can be used to constrain correspondences. This involves abstracting the point sets using a neighbourhood graph such as a nearest neighbour graph, a Delauney graph or a Gabrial graph. For these reason, in this

section we review the literatures on graph matching.

Some of the pioneering work on graph matching was undertaken in the early 1970's by Barrow and Burstall (Barrow and Popplestone, 1971) and by Fischler and Enschlager (Fischler and Elschlager, 1973). These two studies provided proof of concept for the use of relational structures in high-level pictorial object recognition. Over the intervening three decades, there has been a sustained research activity. Broadly speaking the work reported in the literature can be divided into three areas.

The first of these is concerned with defining a measure of relational similarity. Much of the early work here was undertaken in the structural pattern recognition literature. For instance, Shapiro and Haralick (Shapiro and Haralick, 1985) showed how inexact structural representations could be compared by counting consistent subgraphs. This similarity measure was refined by Eshera and Fu (Eshera and Fu, 1986) and by Sanfeliu and Fu (Sanfeliu and Fu, 1983) who showed how the concept of string edit distance could be extended to graphical structures. The formal basis of graph edit distance has recently been extended by Bunke and his coworkers (Bunke and Shearer, 1998; Bunke, 1999) who have shown, among other things, that the edit distance is related to the size of the maximum common subgraph. Tirthapura, Sharvit, Klein and Kimia have shown how the classical Levenshtein distance can be used to match shock-graphs representing 2D skeletal shapes (Tirthapura et al., 1998).

Much of the work described above adopts a heuristic or goal directed approach to measuring graph similarity. The second issue addressed in our literature survey is that of how to develop more principled statistical measures of similarity. This endeavour involves the modeling of the processes of structural error present in the graph-matching problem. Wong and You (Wong and You, 1985) made one of the first contributions here by defining an entropy measure for structural

graph-matching. Boyer and Kak (Boyer and Kak, 1988) also adopted an information theoretic approach, but worked instead with attribute relations. Using a probabilistic relaxation framework Christmas, Kittler and Petrou (Christmas et al., 1995) have developed a statistical model for pairwise attribute relations. Working in the purely structural domain, Wilson and Hancock (Wilson and Hancock, 1997) have derived probability distributions for the relational errors that occur when there is significant graph corruption. More recently Cross and Hancock (Cross and Hancock, 1998) have developed a variant of the EM algorithm in which the structural error model of Wilson and Hancock (Wilson and Hancock, 1997) is used to improve the alignment of triangulated point-sets under perspective geometry. Myers, Wilson and Hancock(Myers et al., 2000) have shown that the computation of the distribution of correspondence errors can be rendered efficient using local edit distance.

The third issue is that of optimisation. Here there have been several attempts to use both continuous and discrete optimisation methods to locate optimal graph matches. Turning our attention first to discrete optimisation methods, there have been several attempts to apply techniques such as simulated annealing (Herault et al., 1990), genetic search (Cross et al., 1997) and tabu search (Williams et al., 1999) to the graph matching problem. However, continuous optimisation methods provide attractive alternatives since their convergence properties are usually better understood than their discrete counterparts. However, the main difficulty associated with mapping a discretely defined search problem onto a continuous optimisation method is that of embedding. There are several ways in which this embedding can be effected for the problem of graph matching. The most straightforward of these is to pose the graph-matching problem as that of recovering a permutation matrix which preserves edge or adjacency structure. For instance, Kosowsky and Yuille have cast the problem into a statistical physics

setting and have recovered a continuous representation of the permutation matrix using mean-field update equations (Yuille et al., 1994). Gold and Rangarajan (Gold and Rangarajan, 1996) have exploited the stochastic properties of Sinkhorn matrices to recover the matches using a soft-assign update algorithm. Umeyama (Umeyama, 1988) takes a more conventional least-squares approach and shows how an eigendecomposition method can be used to recover the permutation matrix. An alternative representation has recently been developed by Pelillo (Pellilo, 1999) which involves embedding the association graph in a continuous space. Matches are found by using the replicator equations of evolutionary game-theory to locate the maximal clique of the association graph, i.e. the maximum common subgraph, of the two graphs being matched. This method has subsequently also been applied to shock-graph matching (Pelillo et al., 1999).

Closely related to this work on recovering permutation structure by continuous embedding is the literature on spectral graph theory. This is a term applied to a family of techniques that aim to characterise the global structural properties of graphs using the eigenvalues and eigenvectors of the adjacency matrix (Chung, 1997). In the computer vision literature there have been a number of attempts to use spectral properties for graph-matching, object recognition and image segmentation. Umeyama has an eigendecomposition method that matches graphs of the same size (Umeyama, 1988). Borrowing ideas from structural chemistry, Scott and Longuet-Higgins were among the first to use spectral methods for correspondence analysis (Scott and Longuet-Higgins, 1991). They showed how to recover correspondences via singular value decomposition on the point association matrix between different images. In keeping more closely with the spirit of spectral graph theory, yet seemingly unaware of the related literature, Shapiro and Brady (Shapiro and Brady, 1992) developed an extension of the Scott and Longuet-Higgins method, in which point sets are matched by comparing the eigenvectors

of the point proximity matrix. Here the proximity matrix is constructed by computing the Gaussian weighted distance between points. The eigen-vectors of the proximity matrices can be viewed as the basis vectors of an orthogonal transformation on the original point identities. In other words, the components of the eigenvectors represent mixing angles for the transformed points. Matching between different point-sets is effected by comparing the pattern of eigenvectors in different images. Shapiro and Brady's method can be viewed as operating in the attribute domain rather than the structural domain. Horaud and Sossa(Horaud and Sossa, 1995) have adopted a purely structural approach to the recognition of line-drawings. Their representation is based on the immanantal polynomials for the Laplacian matrix of the line-connectivity graph. By comparing the coefficients of the polynomials, they are able to index into a large data-base of line-drawings. In another application involving indexing into large data-bases, Sengupta and Boyer(Sengupta and Boyer, 1998) have used property matrix spectra to characterise line-patterns. Various attribute representations are suggested and compared. Shokoufandeh, Dickinson and Siddiqi (Shokoufandeh et al., 1999) have shown how graphs can be encoded using local topological spectra for shape recognition from large data-bases.

Although formally elegant, the main limitation of these matrix methods is their inability to cope with graphs of different sizes. This means that they can not be used when significant levels of structural corruption are present.

2.5 Conclusions

In this thesis, our main interest lies with the matching of point patterns using statistical ideas. Although there exist many methods dealing with alignment and correspondence problems in the computer vision community, most of them oper-

ate with an artificial dichotomy. In this thesis, we will propose a unified statistical framework to solve the alignment and correspondence problems simultaneously.

Extracting raw image features is a crucial step for point pattern matching. Although many different point features can be used to fulfill our research, eg. corners, junctions, terminations, dominant points and middle points of line segments, we only concentrate in this thesis on corner features since they are common features in real images and well explored. The weakness of contour-based corner detectors is that they strongly depend on prior availability of reliable image segmentation. While robust and accurate segmentation itself is not an easy task, gray-scale corner detectors do not need a pre-segmentation step but they are still suffer from sensitivity to noise. One way to realise the problem is to invoke the scale space technique. By introducing the vector potential representation of the Canny edge map, we aim to develop a corner detector which is more robust to noise.

The point alignment process usually requires some labelled point features to estimate the transformations between the point sets under consideration. Unfortunately, the stable labelling of the feature points itself is a very difficult task. Conventional Procrustes alignment technique is useful when the points are labelled and also the numbers of points in each point sets are the same. We make the observation that by treating deformation parameters as the goal of computations and the unknown correspondences as the missing data, then the EM algorithm can be used to solve the alignment problem. The result is an iterative weighted Procrustes alignment method which uses the singular value decomposition(SVD) to find the alignment parameters.

Graph models can be used to represent the relational structure of the images under consideration. Graph matching has proved to be a powerful tool to find the correspondences between point sets. Even though some of the algorithms are

robust to structural disturbance, few of them can deal with graphs with different number of nodes. A graph matching algorithm which can tolerate structural error must be able to cope with different sized graphs and more importantly it must be computationally efficient. By casting the correspondence problem in a matrix framework, the method can be rendered more efficient by using the technique of singular value decomposition.

By noting that a robust correspondence assignment will benefit the alignment process and a good alignment is itself helpful for allocating the correspondences between point sets, a unified framework for pursuing alignment and correspondence simultaneously is proposed. This method uses the cross-entropy to measure the alignment error and correspondence error.

Our next interest is in the alignment of point sets under non-rigid deformations. While Procrustes alignment and its iterative variant can deal with rigid transformation quite well, they are not suitable to cope with non-rigid deformations. Although the point distribution model(PDM) is a well developed method, it has the shortcomings of requiring labelled points and point sets of identical size. In common with most other deformation models, it overlooks the relational arrangement of the points. Based on these observations, we will develop an iterative PDM fitting algorithm by embedding relational constraints in the EM algorithm.

Chapter 3

Feature Detection – Corner Detection via Topographic Analysis of Vector Potential

As described earlier, image feature points can be used for the purpose of matching. The starting point of our research in this thesis is to develop an algorithm which extracts dominant feature points which can be used to match two images. In particular, we concentrate on detecting corners. We aim to develop an algorithm that is robust to image noise so that it can deal with real world problems.

With this aim in mind, in this chapter we present a new corner detection method which exploits both template based and gradient based concepts. We appeal to gradient-based image representation which has already been shown to provide a convenient topographic representation for edge and symmetry features (Cross and Hancock, 1997; Cross and Hancock, 1999). The representation commences from the Canny edge-map, i.e. the directional gradient of the image intensity. By rotating the edge-gradient vectors we compute a field of edge-tangents. The tangent-vectors are smoothed by performing an averaging process in which

the vectors are weighted according to the inverse of their distance from the image point in question. This weighting function is suggested by magnetostatics. For this reason we refer to the resulting gradient field as the vector potential. According to this representation, local maxima of the magnitude vector-field which exhibit directional consistency (i.e. ridges) correspond to edges. Directionally consistent local minima (i.e. ravines) are symmetry lines. In this chapter we extend this topographic picture to include corners. These are saddle structures where directionally consistent ridges and ravines intersect.

It is important to contrast our method of analysis with Haralick's topographic primal sketch (Haralick et al., 1983). Whereas Haralick focuses on the topographic structure of grey-scale features (i.e. a scalar image representation), we analyze the topography of a vector field representation. The main advantage is that we are able to exploit directional consistency in the localisation of topographic structure and, hence, improve the robustness of feature detection. With the topographic representation to hand, the main practical problem that confronts us is the localisation of the saddle-points. This is more difficult than localising ridges and ravines since we are concerned with identifying point features rather than contour features. In the case of ridges and ravines we can exploit constraints on compatible continuity or directionality. In the case of saddle points the constraints are more subtle, since we are seeking locations which are consistent with being the junctions between saddle-ridges and saddle-valleys.

Based on this observation we develop topographic tests for the consistent saddle-structure in the vector potential. This is effectively a template based method. We search for consistent valley structure in the direction of the vector potential and consistent ridge structure in the orthogonal direction. In other words, the directional template characterise local saddle-structure as the intersection of ridge (i.e. edge) and ravine (i.e. symmetry) structures. Moreover,

computation is simplified since the template is fixed to be in the direction of the vector-potential. In this way we avoid explicit computation of directional second derivatives of the image intensity. As we will demonstrate experimentally, this offers advantages in terms of improved noise sensitivity.

The outline of this chapter is as follows. In Section 3.1, we review the vector-potential representation. Section 3.2 introduces the topographic representation of features in the vector potential. In Section 3.3, we confront some of the practical difficulties associated with saddle localisation. Real-world experimental examples are presented in Section 3.4. Questions of algorithm sensitivity and comparison are addressed in Section 3.5. Finally, Section 3.6 provides some conclusions and identifies avenues for future investigation.

3.1 Image representation using vector potential

In this section, we review the feature-representation recently reported by Cross and Hancock (Cross and Hancock, 1997). The starting point is to compute the Canny edge map (Canny, 1986). Accordingly, we commence by convolving the raw image I with a Gaussian kernel of width σ. The kernel takes the following form

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp\Big[-\frac{x^2+y^2}{2\sigma^2}\Big] \tag{3.1}$$

With the filtered image to hand, the Canny edge map is recovered by computing the gradient

$$\vec{E} = \nabla G_\sigma * I \tag{3.2}$$

In order to compute a vector field representation of the edge-map, we will need to introduce an auxiliary z dimension to the original $x-y$ co-ordinate system of the plane image. In this augmented co-ordinate system, the components of the

edge-map are confined to the image plane. In other words, the edge-vector at the point $(x, y, 0)$ on the input image plane is given by

$$\vec{E}(x, y, 0) = \begin{pmatrix} \frac{\partial G_\sigma * I(x,y)}{\partial x} \\ \frac{\partial G_\sigma * I(x,y)}{\partial y} \\ 0 \end{pmatrix} \tag{3.3}$$

For an ideal step-edge, the resulting image gradient will be directed along the boundary normal. A more convenient representation is the edge tangent vectors which flow along the object boundary. Accordingly, we re-direct the edge-vectors so that they are tangential to the original planar shape by computing the cross-product with the normal to the image plane $\hat{z} = (0, 0, 1)^T$. The tangent vector at the point $(x, y, 0)$ on the input image plane is defined to be

$$\vec{j}(x, y, 0) = \hat{z} \wedge \nabla G_\sigma * I(x, y) \tag{3.4}$$

To be more explicit, in terms of its components the tangent vector is given by

$$\vec{j}(x, y, 0) = \begin{pmatrix} -\frac{\partial G_\sigma * I(x,y)}{\partial y} \\ \frac{\partial G_\sigma * I(x,y)}{\partial x} \\ 0 \end{pmatrix} \tag{3.5}$$

In the previously reported work of Cross and Hancock the key idea underlying the image representation was to characterise edges and symmetry lines using topographic structures in the edge-tangent field. Edges corresponded to locations where the tangent vectors re-enforce one-another. In other words, the boundaries are identified as local maxima of the tangent field. Symmetry points are those at which there is cancellation between diametrically opposed tangent vectors. Axes of symmetry are lines of local minimum in the tangent field. At the level of fine detail, intensity ravines give rise to local symmetry axes.

Since the raw gradient vectors are likely to be noisy Cross and Hancock develop a means of smoothing the tangent field so that we can perform the required

topographic analysis. To realise this goal, they appeal to magneto-statics to develop a means of smoothing the tangent field. Accordingly they compute an analogue of the vector-potential by regarding the edge tangents as a field of elementary currents. The vector potential is found by integrating over volume and weighting the contributing currents according to inverse distance. In other words, the vector potential at the point $\vec{r} = (x, y, z)^T$ in the augmented space in which the original image plane is embedded is

$$\vec{A}(x, y, z) = \mu \int_{V'} \frac{\vec{j}(x', y', z')}{|\vec{r} - \vec{r}'|} dV' \tag{3.6}$$

where $\vec{r}' = (x', y', z')^T$ and μ is the permeability constant which we set equal to unity. Since the contributing edge tangent vectors (or currents) are distributed only on the image plane, the volume integral reduces to an area integral over the image plane. As a result, the components of the vector potential are as follows

$$\vec{A}(x, y, z) = \begin{pmatrix} -\int\int \frac{\partial G_\sigma * I(x', y')}{\partial y'} \frac{1}{\sqrt{(x-x')^2+(y-y')^2+z^2}} dx'dy' \\ \int\int \frac{\partial G_\sigma * I(x', y')}{\partial x'} \frac{1}{\sqrt{(x-x')^2+(y-y')^2+z^2}} dx'dy' \\ 0 \end{pmatrix} \tag{3.7}$$

The structure of the vector-potential deserves further comment. In the first instance, the components are confined to the $x - y$ plane for all values of the auxiliary co-ordinate z. However, as we move away from the image plane the role of this auxiliary dimension is to average the generating currents over an increasingly large area of the original image plane. In other words, the role of the auxiliary z-dimension is to allow volume integration of the edge-tangent vectors. By sampling the vector-potential for various $x - y$ planes at increasing sampling height z above the image plane, a scale-space representation is induced. Cross and Hancock exploit this property to produce a fine-to-coarse image representation as they sample the vector potential at increasing sampling heights above the physical image plane.

In order to develop the appropriate differential operators for feature characterisation from the vector-potential Cross and Hancock take the magneto-static analogy one step further and have appealed to the geometry of the associated magnetic field. According to magneto-statics the magnetic field is the curl of the vector potential. It is important to stress that because it is less computationally tractable than the vector-potential, the magnetic field is never used directly in their image representation. The role of the magnetic field is to provide an auxiliary representation. The geometry of the magnetic field allows us to understand the differential structure of the vector-potential. According to their representation of image structure, symmetry lines follow the local minima of the vector-potential. In other words, they connect image points where there is strong cancellation edge tangent vectors associated with symmetrically placed object boundaries. By contrast, edge contours follow the local maxima of the vector potential. According to their representation, the edge lines connect points where there is strong directional re-enforcement between edge tangent-vectors. Symmetry lines can be interpreted as locations where the magnetic field is perpendicular to the sampling image plane. Edges are locations where field lines are tangential to the relevant sampling plane. When viewed from perspective of the differential structure of the vector potential, symmetry lines are locations where the component of the curl in the image plane vanishes, i.e. $\hat{z} \wedge \nabla \wedge \vec{A}(x, y, z) = 0$; edges are locations where the transverse component of the divergence vanishes, i.e. $\nabla.(\hat{z} \wedge \vec{A}(x, y, z)) = 0$.

The novel contribution in this chapter stems from the observation that corners, or points of locally maximum boundary curvature, can be viewed as edge-locations where there is a local symmetry axis associated with a rapid change in boundary direction. Corners therefore correspond to locations where both the edge and symmetry conditions are simultaneously satisfied. From a topographic viewpoint,

corners are located where boundary lines and symmetry lines meet. In other words, we are interested in locating points where there is a local maximum of the magnitude of the vector potential in one direction and a local minimum in the orthogonal direction. As a result corner detection can be treated as saddle point detection.

3.2 Topographic Representation

In the previous section, we established that corners are saddle-points in the magnitude of the vector-potential. We therefore focus on the analysis of the scalar quantity

$$g(x,y)_z = |\vec{A}(x,y,z)| \tag{3.8}$$

The topographic structure of the vector potential can be characterised using the Hessian matrix

$$\mathcal{H}_g = \begin{pmatrix} g_{xx} & g_{xy} \\ g_{xy} & g_{yy} \end{pmatrix} \tag{3.9}$$

where the second derivatives are given by

$$\begin{aligned} g_{xx} &= \frac{\partial^2 g(x,y)_z}{\partial^2 x} \\ g_{xy} &= \frac{\partial^2 g(x,y)_z}{\partial x \partial y} \\ g_{yy} &= \frac{\partial^2 g(x,y)_z}{\partial^2 y} \end{aligned}$$

The eigen-structure of the Hessian matrix can be used to gauge the curvature of the surface. The two eigen-values of $\mathcal{H}$ are the maximum curvature k_1 and minimum curvature k_2. The orthogonal eigen-vectors of $\mathcal{H}$ are known as the principal curvature directions. The mean-curvature (K) of the surface is found by averaging the maximum and minimum curvatures. Finally, the Gaussian curvature (H)

is equal to the product of the two eigenvalues. As a result,

$$H = k_1 k_2 \tag{3.10}$$

and

$$K = \frac{k_1 + k_2}{2} \tag{3.11}$$

Class	Symbol	K	H	Region-type
Dome	D	-	+	Elliptic
Ridge	R	-	0	Parabolic
Saddle ridge	SR	-	-	Hyperbolic
Plane	P	0	0	Hyperbolic
Saddle-point	S	0	-	Hyperbolic
Cup	C	+	+	Elliptic
Valley	V	+	0	Parabolic
Saddle-valley	SV	+	-	Hyperbolic

Table 3.1: Curvature classes.

The signs and zeros of the mean and Gaussian curvatures can be used to categorise the local surface geometry into a number of distinct topographic classes. These classes are summarised in Table 1. In this chapter, we are interested in the saddle-structures which are labelled as hyperbolic features in the table. These features are characterised by the condition $H < 0$. In particular we are interested in points that are consistent with being the intersections of edge and symmetry lines, i.e. in the intersections of saddle-ridges and saddle valleys. The joint condition for the intersections is

$$K \neq 0 \ \wedge \ H < 0 \tag{3.12}$$

By searching for the intersection of consistent saddle ridges, we overcome some of the problems of localising saddle-points, for which $K = 0$ and $H < 0$. This can prove difficult since there are no constraints from the directionality of the desired feature. In particular, we mitigate this difficulty and realise the corner localisation process using templates to search for the junctions between symmetry lines and edge-lines.

3.3 Implementation

In this section we describe two aspects of the implementation of our corner detector. The first of these is the means by which we compute the vector potential. The computation is realised using fast Fourier transforms. The second implementational detail concerns the practical means by which we locate saddle structures.

3.3.1 Computing the vector potential

Key to our implementation is the fact that the volume integrals appearing in the definition of the vector-potential (equation 3.7) can be replaced by spatial-convolutions with a sampling height-dependant filter. Specifically, we invoke the Fourier duality between convolution in the spatial domain and multiplication in the frequency domain. In this way the discretised version of the vector-potential can be computed using just three 2D Fourier transforms and a pair of frequency-domain multiplications.

Our basic goal is to compute the vector potential at a given sampling height above the image plane. The two dimensional integrals appearing in the definition of vector-potential can be discretised to give the following x and y components

$$A_x(x, y, z) = -\sum_{x'} \sum_{y'} \frac{\partial G_\sigma * I(x', y')}{\partial y'} \frac{1}{\sqrt{(x - x')^2 + (y - y')^2 + z^2}} \qquad (3.13)$$

$$A_y(x,y,z) = \sum_{x'} \sum_{y'} \frac{\partial G_\sigma * I(x',y')}{\partial x'} \frac{1}{\sqrt{(x-x')^2 + (y-y')^2 + z^2}} \tag{3.14}$$

The double summation can be replaced by a convolution with a composite filter. For instance, the x-component of the vector potential is as follows

$$A_x(x,y,z) = -(V_x(\sigma,z) * I)(x,y) \tag{3.15}$$

We exploit the commutative properties of convolution to compute the composite filter $V_x(\sigma,z)$. The filter is found by convolving the appropriate directional derivative of the Gaussian with the inverse Euclidean distance operator, i.e.

$$V_x(\sigma,z)(x,y) = \sum_{x'} \sum_{y'} \frac{\partial G_\sigma(x',y'))}{\partial y'} \frac{1}{\sqrt{(x-x')^2 + (y-y')^2 + z^2}} \tag{3.16}$$

For practical reasons, we would like to realise the computation of the components of the vector potential using fast Fourier transforms. Our basic strategy is to exploit the Fourier duality between convolution in the spatial domain and multiplication in the frequency domain. Schematically, we utilise the identity

$$A_x = \mathcal{F}^{-1}\Big[\mathcal{F}[V_x(\sigma,z)] \times \mathcal{F}[I]\Big] \tag{3.17}$$

to compute each of the components of the vector potential in turn. In this way the vector-potential can be obtained using three separate Fourier transform operations. The first of these involves computing the Fourier transform of the raw image $\mathcal{F}[I]$. Two separate weighted spatial frequency distributions are then constructed by multiplying the components of the image Fourier transform with the Fourier representation for each of the composite filters in turn, i.e. we compute $\mathcal{F}[V_x(\sigma,z)]$ and $\mathcal{F}[V_y(\sigma,z)]$. Finally, the two spatial components of the vector-potential are obtained by inverse Fourier transformation of the two weighted frequency distributions.

3.3.2 Localizing saddles

Based on the results presented in Section 3.2, we make the following observations concerning the topographic structure of the vector potential in the proximity of corners

- There is a local minimum of the magnitude of the vector potential in the direction of the vector-potential.
- There is a local maxima of the magnitude of the vector potential in the orthogonal direction.
- At the locations of corners, the magnitude of vector potential along both the contour and its orthogonal direction changes rapidly.
- At the locations of corners, the magnitude of the Gaussian curvature is significant.

Based on the first two observations we search for saddle-points that are consistent when viewed from a finite support neighbourhood. In practice we localise consistent topographic structure using a simplified form of template convolution. Our template tests for orthogonal maxima and minima using directional second-derivatives and subsequent non-maximum suppression and non-minimum suppression tests. The saddle points are corner candidates. Because of image noise and other imperfections, the points detected by our saddle-template are not always the locations of true corners in the image. To overcome this problem we can appeal to the directional consistency of the derivatives of the vector potential to refine the corner estimates. The aim is to search for orthogonal ridge and ravine structures. To meet this goal, we used directional second derivative operators to compute a corner "strength" measure. This measure captures the

directional variations of the vector potential along the contour direction and in the orthogonal direction.

To be more formal, suppose that $V(x,y,z) = \partial^2/\partial\vec{A}_{\|}^2$ is the second derivative of the magnitude of the vector potential in the direction of the vector potential. The magnitude of this quantity will take on a maximum value when there is a local minimum or valley structure in the magnitude of the vector-potential. Further suppose that $R(x,y,z) = \partial^2/\partial\vec{A}_{\perp}^2$ is the second derivative of the magnitude of the vector-potential in the direction perpendicular to the local orientation of the vector-field. The magnitude of this quantity exhibits a local maxima when there is a ridge structure or local maxima in the vector potential. Using these two operators, we search for corners by computing the following corner strength measure which gauges consistent saddle structure.

$$C = |V(x,y,z)| \times |R(x,y,z)| \tag{3.18}$$

The measure is an approximation to the Gaussian curvature. It is large in value when there are orthogonal ridges and valleys in the magnitude of the vector-potential. Corners are selected by thresholding this aggregate measure of corner strength.

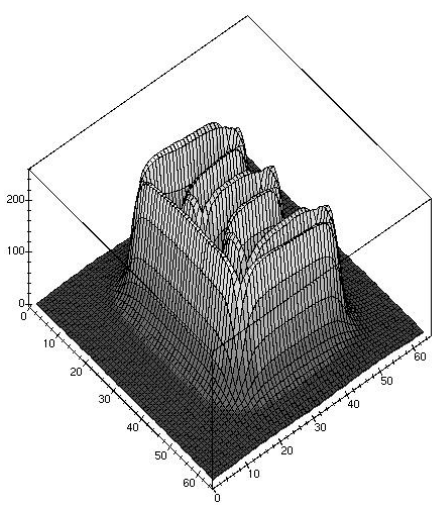
3.4 Experiments

In this section, we provide some experimental evaluation of the corner detection algorithm. The experimental work is divided into two parts. We commence with some examples on binary imagery to illustrate some of the properties of the representation. Next we furnish real world examples. To illustrate the properties of our vector potential representation and corner detection algorithm, we use an simple binary image of "E". Figure 3.1a is the magnitude of the vector potential

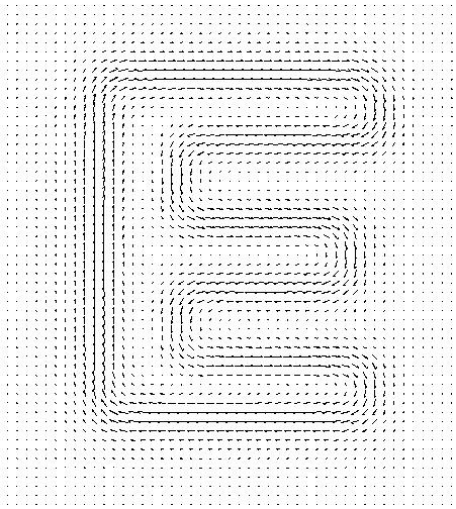
for the binary image, Figures 3.1b is the direction of the vector potential. Figure 3.1c shows the detected corners. For this simple image, the results are all correct. The magnitude of the vector potential is displayed as a function of the sampling height z in Figure 3.1a to emphasise topographic structure. Here the saddle-structure associated with the corners is clear. The ridge and ravine structure of the edge and symmetry lines is also evident. In Figure 3.1b we display the vectorial representation of $\vec{A}(x, y, 0)$. The main feature to note from this figure is that the direction of the vector potential changes rapidly at the corner locations.

As explained earlier, we can endow our image representation with a scale-space dimension by sampling the vector potential at increasing sampling heights above the image plane. In Figures 3.2 and 3.3, we provide some qualitative examples of this scale-space sampling. In each case the left-hand figure is an elevation map showing the magnitude of the vector-potential while the right-hand figure is the vector-field. The main feature to note from these examples is that as the scale or sampling height is increased, then so the saddle structures become shallower.

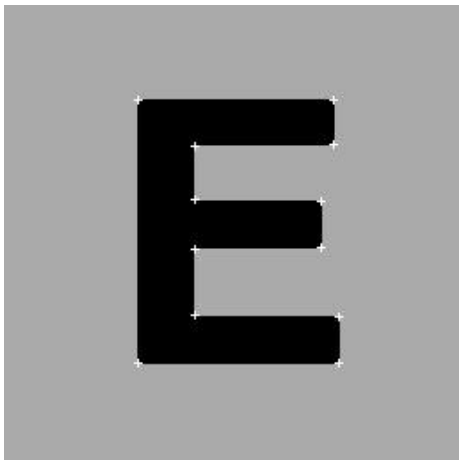
We now turn our attention to real-world scenes. To provide some comparison, we have provided some experimentation with the SUSAN corner detector (Smith and Brady, 1997). Figure 3.4a is the original INRIA office image. In Figure 3.4b we show the result of applying the algorithm reported in this chapter, while Figure 3.4c shows the result of applying the SUSAN corner detector. The results obtained with our algorithm are generally cleaner, and there are fewer false positives. There are also some interesting qualitative differences in the detected corners. For instance in our algorithm, the meeting of the line-like horizontal bars and thicker vertical bars of the window are detected as single junctions. In the case of SUSAN, double corners are returned. The result of our algorithm is more perceptually intuitive and may prove more useful for higher level matching problems.



(a) Magnitude

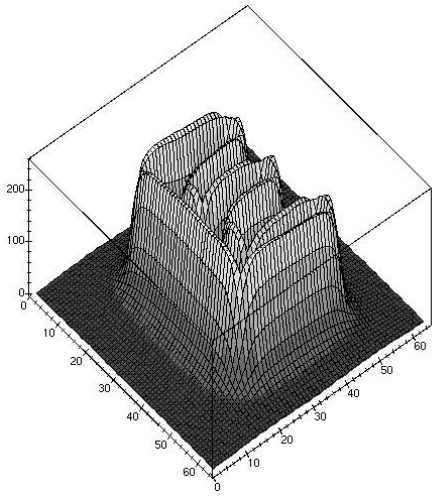


(b) Direction

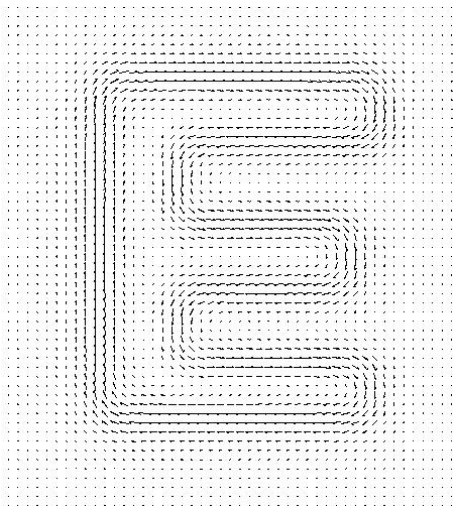


(c) Detected corners

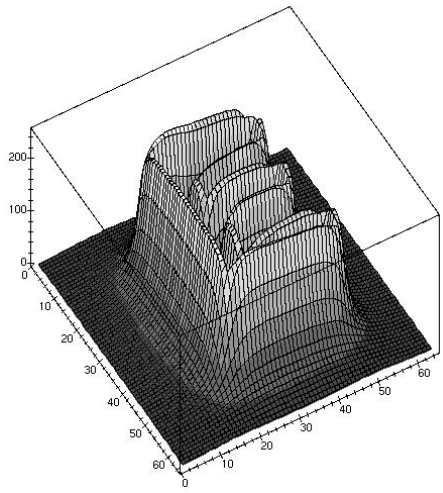
Figure 3.1: Topographic representation and corner detection.



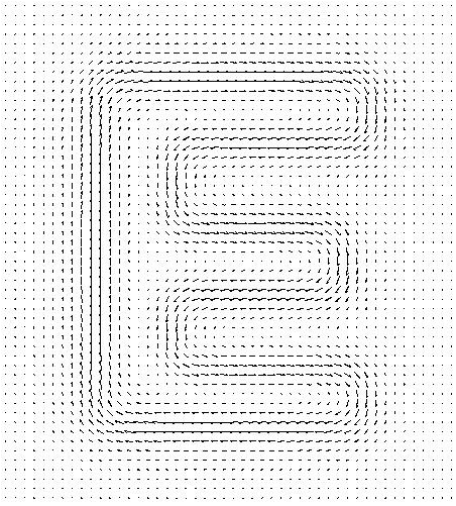
(a) Magnitude at sampling height z = 0



(b) Direction at sampling height z = 0

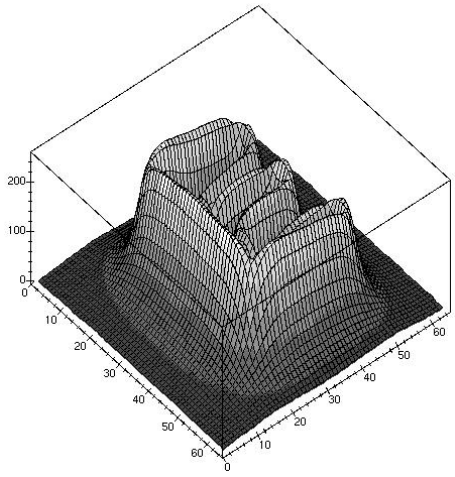


(c) Magnitude at sampling height z = 1

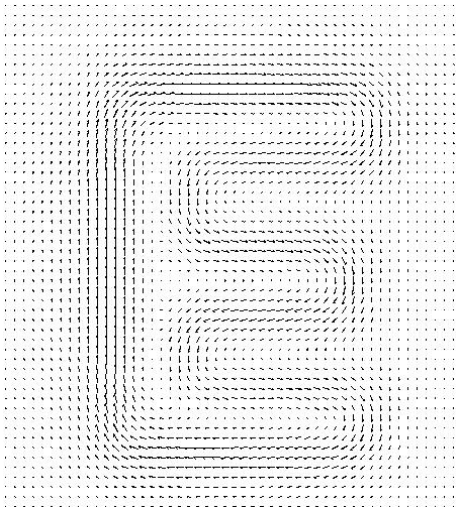


(d) Direction at sampling height z = 1

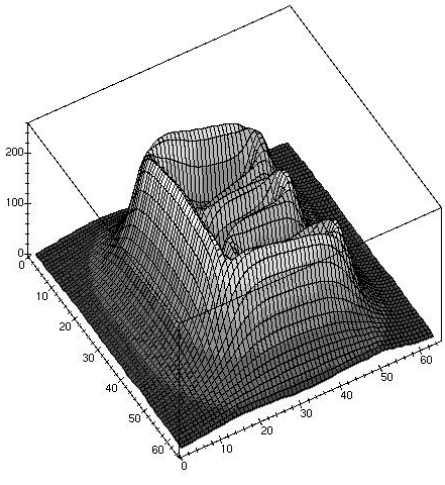
Figure 3.2: Topographic representation at different sampling heights.



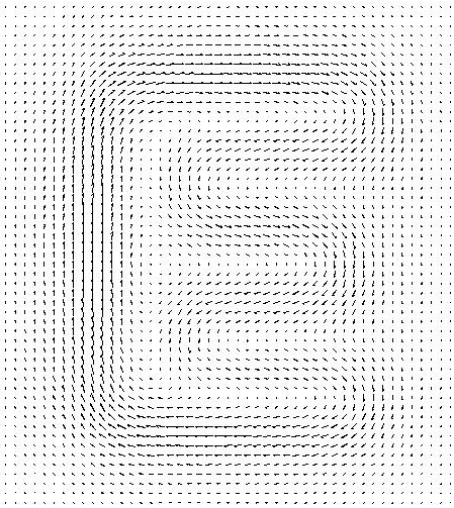
(a) Magnitude at sampling height z = 2



(b) Direction at sampling height z = 2



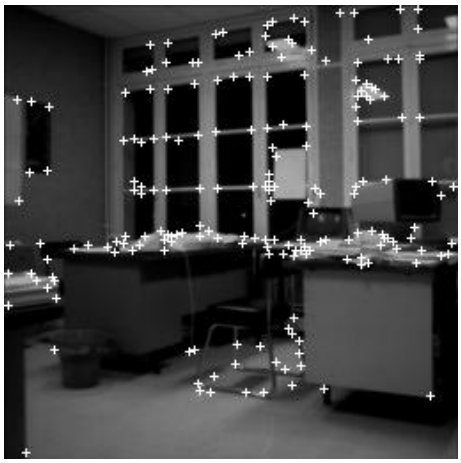
(c) Magnitude at sampling height z = 3



(d) Direction at sampling height z = 3

Figure 3.3: Topographic representation at different sampling heights(cont.).

(a) Original image



(b) Corners from new method

(c) Corners from SUSAN

Figure 3.4: Corner detection results.

(a) z = 0 (b) z = 0

(c) z = 1 (d) z = 1

(e) z = 2 (f) z = 2

Figure 3.5: Corner detection at different sampling heights.

Finally, we provide some examples of the scale-space detection of corners in the INRIA office scene. Figure 3.5 shows the results of corner detection at a number of different sampling heights. The left column of the figure shows the magnitude of the vector potential, while the right-column shows the detected corners superimposed on the original image. As we move from the top row of the figure to the bottom row, the sampling height z of the vector potential increases. As the sampling height increases, then so only the dominant corners remain. However, the majority of the significant corners persist over the full set of sampling heights.

3.5 Performance analysis

Our final piece of experimental work is aimed at measuring the noise sensitivity of our corner detector and comparing it with some of the alternative corner detectors reviewed in the introductory section of this chapter. The specific algorithms used in this comparison are the SUSAN corner detector (Smith and Brady, 1997), Wang and Brady's corner detector (Wang and Brady, 1995) and the Plessey corner detector (Harris and Stephens, 1988). To realise this comparison, we have generated synthetic images of cog-wheels (see Fig. 3.6) and have added salt-and-pepper noise with known proportion. By increasing the number of spikes on the circumference of the cog, we can systematically reduce the opening angle of the corners. The synthetic figure provides ground-truth data in which the number of target corners is known. We focus on two aspects of the noise sensitivity of our corner detector. The first of these is the scale-dependance of the corner detection process. The second is the error rate for false positives and true positives.

We commence our evaluation by measuring the accuracy of the corner detection process as a function of sampling height (i.e. spatial scale) and corner

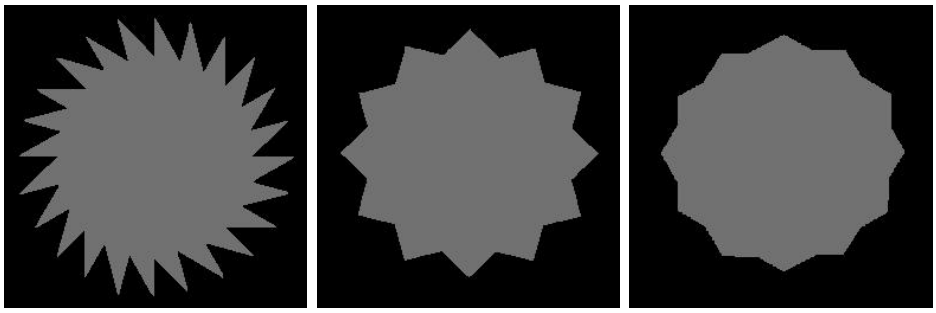


Figure 3.6: Cog-wheel test images.

opening angle. Figure 3.7 shows the fraction of correctly detected corners as a function of sampling height z. The different curves are for different opening angles. The main conclusion that can be drawn from this plot is that our corner detector degrades with increasing sampling height. There is also an indication that we encounter difficulties with small opening angle corners.

In order to understand in qualitative way the angle systematics involved in corner detection, we have generated a series of synthetic examples. The resulting plots of the vector-potential magnitude are shown in Figure 3.8. As the angle increases then so the depth of the saddle decreases. At small opening angles the width of the saddle becomes very narrow and hence difficult to localise.

The second aspect of our sensitivity study is to provide some comparison with the alternative corner detectors under conditions of controlled noise. The plots in Figure 3.9 show the probability of both true positives, i.e. the fraction of genuine corners that are correctly detected, and the probability of false positives, i.e. the fraction of detected corners which are false alarms. The probabilities of true positives and false positives are plotted as a function of the probability or fraction of added noise. In each case, the solid curve is the result of the proposed algorithm, the long dashed curve is for the SUSAN corner detector, the dashed curve is for Wang and Brady's corner detector, and the dotted line is for the

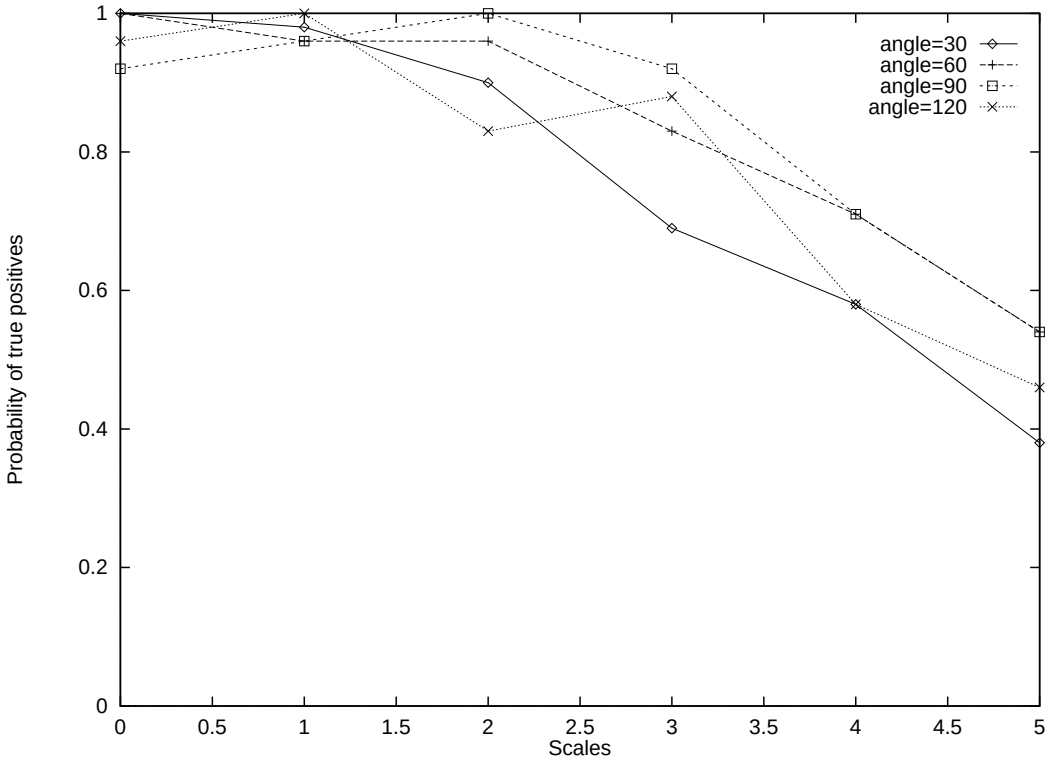
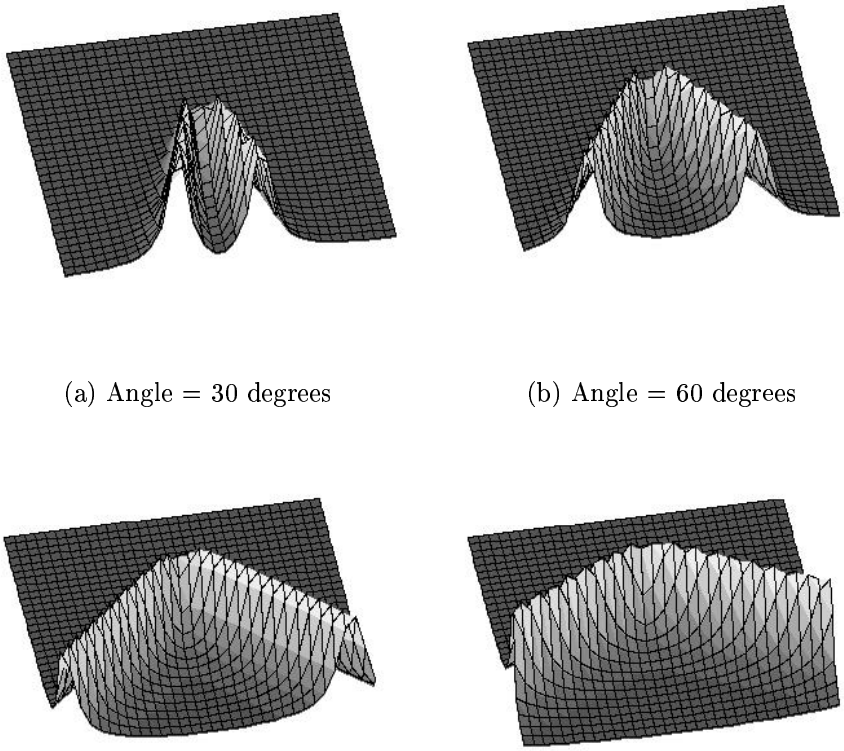


Figure 3.7: Comparison for different sampling heights.

Plessey corner detector. From the plot of Figure 3.9a, we can draw the conclusion that the proposed algorithm consistently outperforms the alternatives in the sense that it has a higher probability of true positives for small portions of added noise. When more noise added to the image, the performance of our method is almost identical with that of the Wang and Brady, and the Plessey corner detectors. Our method always outperforms SUSAN in terms of its probability of true positives.

The results shown in Figure 3.9b provide more conclusive support for our proposed corner detector. Here we see that the probability of false positives for our proposed method is consistently significantly lower than that for the alternative corner detectors. The only exception occurs at low noise-levels where the Plessey corner detector offers slightly better performance.

Next, we aim to compare the performance of the corner detectors for different opening angles and different sampling heights. We choose a small opening angle



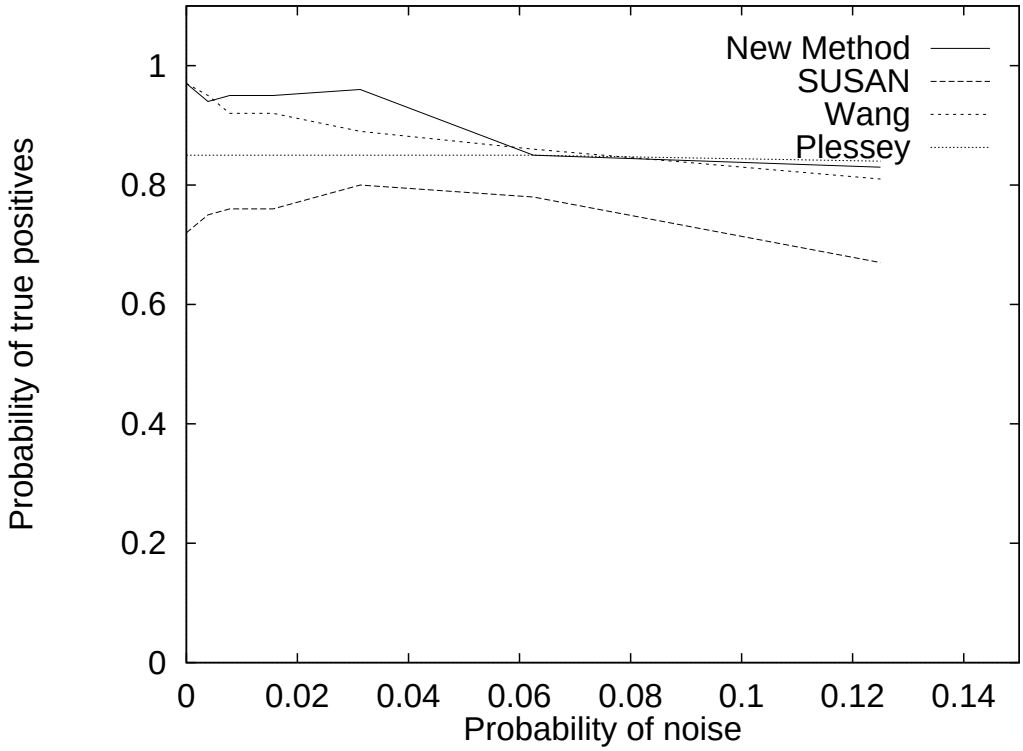
(a) Angle = 30 degrees

(b) Angle = 60 degrees

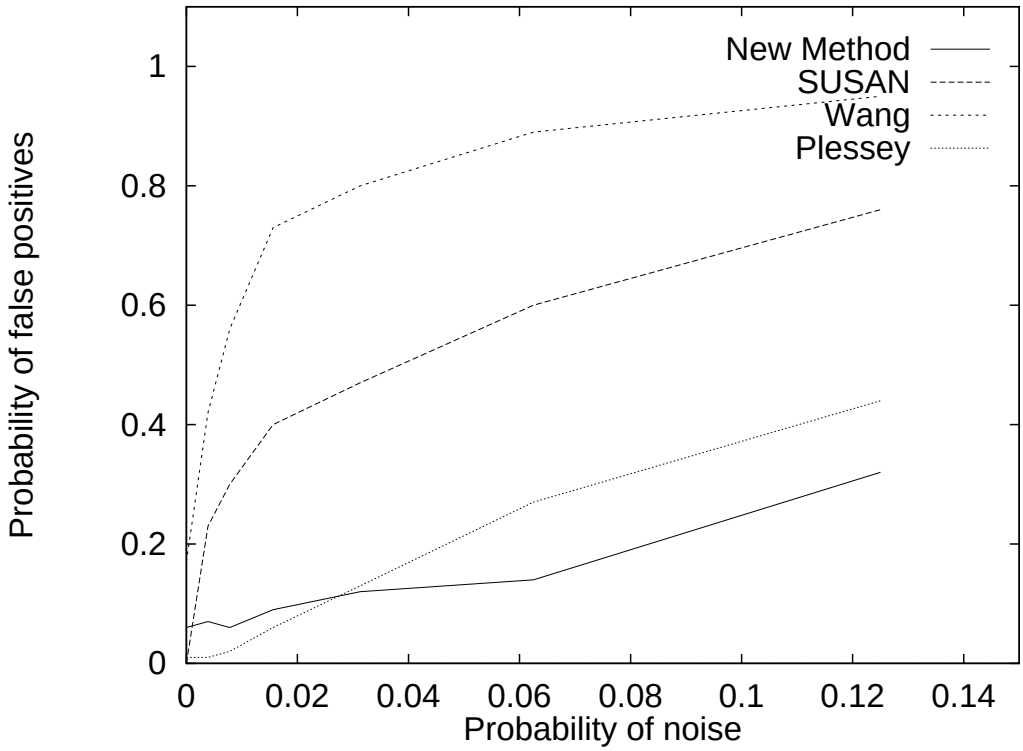
(c) Angle = 90 degrees

(d) Angle = 120 degrees

Figure 3.8: Saddle structures for different opening angles.



(a) True Positives



(b) False Positives

Figure 3.9: Noise sensitivity for various methods.

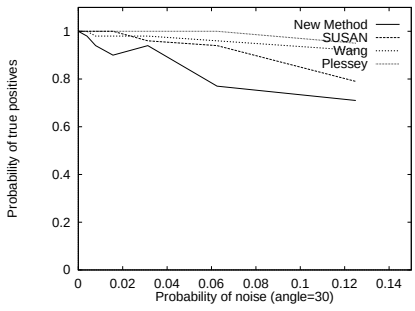
of 30 degrees, a medium opening angle of 90 degrees and a large opening angle of 120 degrees. From Figure 3.10, we draw the conclusion that, when measured in terms of the fraction of true positives, our method performs a little worse than that of the other corner detectors for small opening angles, a little better than the others at medium opening angles, and significantly better than the alternatives at large opening angles.

Although the proposed corner detection method performs relatively better for noisy images, it can only be used in the case where corners are the dominant features. Automaticly extracted corners are used in some kind of images in this thesis. While for other images in which corners are not easy to detect automaticly or other features are more important, we set the features by hand to facilitate following processes.

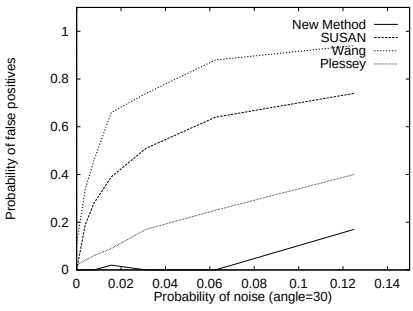
3.6 Conclusions

In this chapter we have presented a corner detection method that is based on the topographic analysis of a vector-potential image representation. According to our representation, corners are saddle-points where saddle-ridges and saddle-valleys intersect. Experimental results reveal the method offers performance advantages over the SUSAN corner detector, Wang and Brady's corner detector and the Plessey corner detector. The most striking of these is to offer better control over false positives.

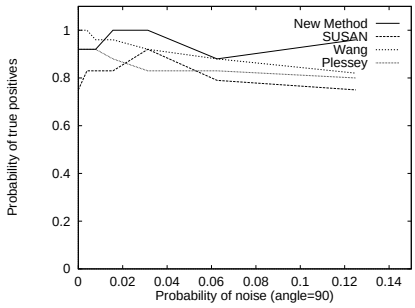
There are a number of ways in which the ideas presented in this chapter could be developed. We clearly have a means of computing a curvature scale-space. In this respect our work is similar to that of Brady and Asada (Asada and Brady, 1986) and Mokhtarian and Mackworth (Mokhtarian and Mackworth, 1992). Our next step is to investigate how the new corner representation can be used for



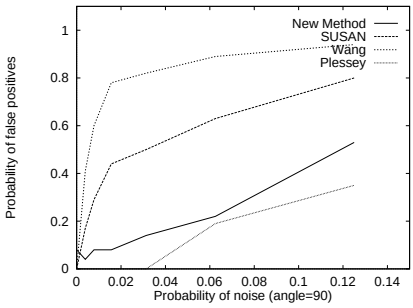
(a) True Positives, Opening angle = 30 degrees



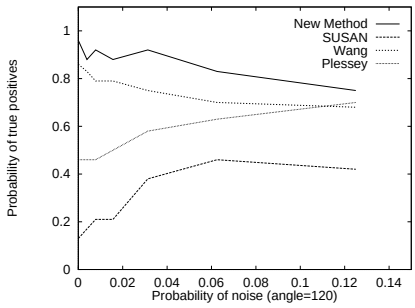
(b) False Positives, Opening angle = 30 degrees



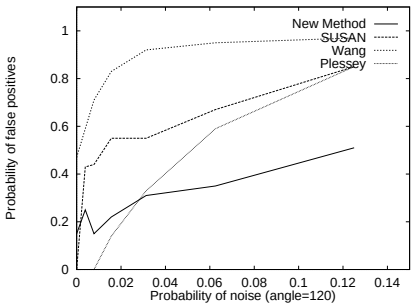
(c) True Positives, Opening angle = 90 degrees



(d) False Positives, Opening angle = 90 degrees



(e) True Positives, Opening angle = 120 degrees



(f) False Positives, Opening angle = 120 degrees

Figure 3.10: Noise sensitivity for various opening angles.

shape matching and recognition. In the next chapter, the corners extracted using the above corner detector will be used to match images through an iterative Procrustes alignment method.

Chapter 4

Iterative Procrustes Alignment

Having corner features to hand, the image alignment problem is converted to one of point set alignment which needs far less computation than if grey-scale convolution is attempted. In this chapter, by introducing a weight function to the squared error measure, we present an iterative Procrustes alignment method which aligns two point sets where rigid deformation exists between them.

Point pattern matching is key to a number of problems in image analysis and computer vision. The problem is to find matches between two or more point-sets when there are geometric distortions, point measurement errors and contamination present. The problem has found concrete applications in shape analysis(Cootes et al., 1995), stereo reconstruction(Shapiro and Brady, 1992) and motion analysis(Shapiro et al., 1995; Torr and Murray, 1997). Broadly speaking, the problem can be addressed as either alignment or correspondence. Alignment is concerned with transforming the different point-sets onto one-another so as to minimise a measure of error or aggregate distance between corresponding points. The method aims to recover the parameters of a predefined geometric transformation. Correspondence on the other hand is a labelling problem. The method aims to exploit relational constraints concerning the arrangement of points to

find a consistent pattern of matches.

The framework presented by Cross and Hancock(Cross and Hancock, 1998) is a powerful one which allows unlabelled point-sets of different sizes to be matched. However, much of the potential of the method remains unexplored. The idea underpinning this chapter is to use the framework to explore how Procrustes alignment can be enhanced using the apparatus of the EM algorithm. There are two identifiable shortcomings with the standard Procrustes method. Firstly, it requires point sets of identical size. Secondly, it requires the points to be labelled, i.e. the correspondences to be known in advance. To overcome these two problems we develop an iterative Procrustes alignment method. The EM algorithm provides a natural framework for our study since it provides a framework for addressing maximum likelihood estimation problems which involve the possibility of hidden or missing data. Specifically, it provides a natural way of representing the unknown correspondences between individual points. By weighting the correspondences using the *a posteriori* alignment probabilities, we overcome the two problems noted above.

The algorithm has two steps. In the maximisation step of the algorithm, we align the points so that they minimise the weighted Procrustes distance. In the expectation step the positional residuals are used to estimate correspondence matching probabilities used in the weighting process. These two processes are interleaved and iterated to convergence.

The resulting algorithm has a number of similarities with alternatives already reported in the literature. For instance, when the alignment errors are assumed to follow a Gaussian distribution, then our *a posteriori* correspondence probabilities are exponential functions of the squared inter-image distance between points. Scott and Longuet-Higgins have shown how correspondences can be recovered via singular value decomposition of a matrix of correspondence weights computed in

this way. Although the architecture of our method has much in common with the Procrustes soft-assign method of Rangarajan, Chui and Bookstein(Rangarajan et al., 1997), it is couched in a statistical rather than an optimization framework. Finally, we note that although our analysis here is restricted to Euclidean transformations, it has wider reaching implications. Firstly, the recovery of point-pattern matches under similarity transform may provide a good starting point for more detailed affine or perspective matching. Secondly, the EM framework can be extended to Procrustes analysis over more complex transformation groups.

4.1 Point-set representation

Our goal is to recover the parameters of a geometric transformation $\Phi^{(n)}$ that best maps a set of image feature points $\mathbf{w}$ onto their counterparts in a model $\mathbf{z}$. In order to do this, we represent each point in the image data set by a position vector $\vec{u}_i = (x_i, y_i)^T$ where i is the point index. We will assume that all these points lie on a single plane in the image. In the interests of brevity we will denote the entire set of image points by $\mathbf{w} = \{\vec{u}_i, \forall i \in \mathcal{D}\}$ where $\mathcal{D}$ is the point index-set. The corresponding fiducial points constituting the model are similarly represented by $\mathbf{z} = \{\vec{v}_j, \forall j \in \mathcal{M}\}$ where $\mathcal{M}$ denotes the index-set for the model feature-points $\vec{v}_j$.

Later on we will show how the two point-sets can be aligned using singular value decomposition. In order to establish the required matrix representation of the alignment process, we construct two co-ordinate matrices from the point position vectors. The data-points are represented by the following matrix whose columns are the co-ordinate position vectors

$$D = (\, \vec{u}_1| \quad \vec{u}_2| \quad ....| \quad \vec{u}_{|\mathcal{D}|}\,) \tag{4.1}$$

The corresponding point-position matrix for the model is

$$M = (\ \vec{v}_1| \quad \vec{v}_2| \quad ....| \quad \vec{v}_{|\mathcal{M}|}\ ) \tag{4.2}$$

When the geometric transformation $\Phi^{(n)}$ is applied to the set of data-points, then a revised point position matrix

$$D^{(n)} = (\ \vec{u}_1^{(n)}| \quad \vec{u}_2^{(n)}| \quad ....| \quad \vec{u}_{|\mathcal{D}|}^{(n)}\ ) \tag{4.3}$$

is obtained.

4.2 Orthogonal Procrustes method for point-set alignment

Orthogonal Procrustes method aims to find the optimal rotation matrix between two point sets which can best align one to the other. Suppose the data point set D and the model point set M are defined as described in last section. The orthogonal Procrustes method aims to recover the general rotation matrix R which minimizes the quantity $\|D - RM\|^2$ subject to the constraint $R^T R = I_p$ where p is the dimension of the data and I_p is the $p \times p$ identity matrix.

Since the rotation matrix R is orthogonal, we have

$$\|D - RM\|^2 = Tr[D^T D] + Tr[M^T M] - 2Tr[DM^T R] \tag{4.4}$$

It is clear that the minimisation of $\|D - RM\|^2$ is equivalent to maximisation of the term $Tr[DM^T R]$.

The maximization of $Tr(DM^T R)$ is achieved by applying the Kristof's inequality(Kristof, 1970). If Δ is a diagonal matrix with nonnegative entries, and T is orthogonal, Kristof's inequality states that

$$Tr(T\Delta) \leq Tr(\Delta) \tag{4.5}$$

with equality if and only if $T = I$, with I an identity matrix.

Suppose that the SVD of DM^T is $U\Delta V^T$, with orthogonal matrices U and V, and diagonal matrix Δ. Using the invariance of the trace function under cyclic permutation and drawing on Kristof's inequality, we have

$$\begin{aligned} Tr(DM^T R) &= Tr(U\Delta V^T R) \\ &= Tr(V^T RU\Delta) \\ &\leq Tr(\Delta) \end{aligned} \tag{4.6}$$

We can show that $V^T RU$ is orthogonal if R is orthogonal. The maximum is achieved when $V^T RU = I$. Therefore, the optimal general rotation matrix R for aligning the two point sets is

$$R = VU^T \tag{4.7}$$

4.3 EM Algorithm for point-set alignment

The aim in this section is to show how the Procrustes alignment of the two point-sets can be realised using the EM algorithm. The ultimate goal of the alignment process is to identify point-to-point correspondences and the transformation parameters between the data and the model. Moreover, we are interested in the case where there are significant structural differences between the two-point sets due to the addition of noise or the occlusion and drop-out of certain feature-points.

The EM algorithm provides a natural framework for recovering the required correspondences. The method is concerned with finding maximum likelihood solutions to problems posed in terms of missing or hidden data. In the alignment problem it is the correspondences which are missing and the transformation parameters that need to be recovered. The utility measure underpinning the method is the expected log-likelihood function. Under the assumption that the positional

errors between the aligned point-sets are Gaussian, then the maximum likelihood problem becomes one of minimising a weighted squared error measure or Procrustes distance. The weights used to control the different positional errors are in fact the *a posteriori* probabilities of the point correspondences. The EM algorithm iterates between two interleaved computational steps. In the expectation step the *a posteriori* correspondence probabilities are estimated from the current position errors by applying the Bayes formula to the Gaussian distribution functions. In the maximisation step the alignment parameters are estimated so as to maximise the expected log-likelihood function. This is equivalent to minimisation of the weighted error measure. Here we realise the maximisation step by adopting a matrix representation of the point-sets together with their putative correspondence probabilities and applying singular value decomposition to recover the alignment parameters. In practice, we iterate the Procrustes alignment on a weighted correspondence matrix.

4.3.1 Mixture Model

The idea underpinning the EM algorithm is to construct a mixture model over the hidden data to explain the distribution of the observed data. The ultimate goal is the set of maximum likelihood parameters Φ which explain the observed distribution of data. In our alignment problem, the observed data are the position vectors belonging to the set $\mathbf{w}$. The parameters are the translation, rotation and scaling required by the Procrustes alignment of the point-sets.

The method commences by assuming that the different observations are independent of one-another. As a result we can factorise the joint conditional likelihood of the data over the individual point position vectors, i.e.

$$p(\mathbf{w}|\Phi) = \prod_{i \in \mathcal{D}} p(\vec{u}_i|\Phi) \tag{4.8}$$

The next step is to focus on the probability distribution $p(\vec{u}_i|\Phi)$. Here we assume that the observed data-point positions have arisen from the model-points via a measurement process. However, the original model point is hidden from us. We must therefore entertain the possibility that each data point may have originated via measurement error from any of the model-points. This situation is expressed probabilistically by constructing a mixture model over the set of hidden model-data associations or correspondences. As a result, we write

$$p(\vec{u}_i|\Phi) = \sum_{j \in \mathcal{M}} p(\vec{u}_i|\vec{v}_j, \Phi)\pi_{i,j} \tag{4.9}$$

where $p(\vec{u}_i|\vec{v}_j, \Phi)$ is the probability distribution for the data-point position measurement or observation $\vec{u}_i$ to have originated from the model-point $\vec{v}_j$ under the set of alignment parameters Φ. The quantity $\pi_{i,j}$ is the mixing proportion required for the model-point $\vec{v}_j$ in explaining the observation $\vec{u}_i$.

With these ingredients, the complete likelihood function that has to be maximised is

$$\mathcal{L} = \prod_{i \in \mathcal{D}} \sum_{j \in \mathcal{M}} p(\vec{u}_i|\vec{v}_j, \Phi)\pi_{i,j} \tag{4.10}$$

The idea underpinning the EM algorithm is to accommodate the hidden data, by re-couching the maximisation of the likelihood function in terms of the expected log-likelihood function. It was Dempster, Laird and Rubin (Dempster et al., 1977) who originally showed that maximising the expected value of the log-likelihood function under hidden or missing data, was equivalent to maximising the following quantity

$$Q(\Phi^{(n+1)}|\Phi^{(n)}) = \sum_{i \in \mathcal{D}} \sum_{j \in \mathcal{M}} P(\vec{v}_j|\vec{u}_i, \Phi^{(n)}) \ln p(\vec{u}_i|\vec{v}_j, \Phi^{(n+1)}) \tag{4.11}$$

According to this viewpoint the *a posteriori* probabilities available at iteration n of the algorithm are used to compute the expectation value of the log-likelihoods of the missing data at iteration $n+1$.

4.3.2 Maximisation

To develop a useful alignment algorithm we require a model for the measurement process. Here we assume that the observed position vectors, i.e. $\vec{u}_i$ are derived from the model points through a Gaussian error process. Suppose that the revised estimate of the position of the point $\vec{u}_i$ under the set of alignment parameters $\Phi^{(n)}$ is $\vec{u}_i^{(n)}$. According to our Gaussian model of the alignment errors,

$$p(\vec{u}_i|\vec{v}_j,\Phi^{(n)}) = \frac{1}{2\pi\sqrt{|\Sigma|}}\exp\Big[-\frac{1}{2}(\vec{v}_j-\vec{u}_i^{(n)})^T\Sigma^{-1}(\vec{v}_j-\vec{u}_i^{(n)})\Big] \tag{4.12}$$

where Σ is the variance-covariance matrix for the point measurement errors. Here we assume that the position errors are isotropic, in other words the errors in the x and y directions are identical and uncorrelated. As a result we write $\Sigma = \sigma^2 I_2$ where I_2 is the 2x2 identity matrix and σ^2 is the isotropic noise variance for the point positions. With this model, maximisation of the expected log-likelihood function $Q(\Phi^{(n+1)}|\Phi^{(n)})$ reduces to minimising the weighted squared error measure

$$\mathcal{E} = \sum_{i\in\mathcal{D}}\sum_{j\in\mathcal{M}} w_{i,j}^{(n)}(\vec{v}_j-\vec{u}_i^{(n+1)})^T(\vec{v}_j-\vec{u}_i^{(n+1)}) \tag{4.13}$$

where we have used the shorthand notation $w_{i,j}^{(n)}$ to denote the *a posteriori* correspondence probability $P(\vec{v}_j|\vec{u}_i,\Phi^{(n)})$.

We would like to recover the maximum likelihood alignment parameters by applying Procrustes normalisation to the two point-sets. This involves performing singular value decomposition of a point-correspondence matrix. In order to develop the necessary formalism, we rewrite the weighted squared error criterion using a matrix representation. Suppose that $W^{(n)}$ is the $|\mathcal{D}|\times|\mathcal{M}|$ data-responsibility matrix whose elements are the *a posteriori* correspondence probabilities $w_{i,j}^{(n)} = P(\vec{v}_j|\vec{u}_i,\Phi^{(n)})$. With this notation the quantity $\mathcal{E}$ can be expressed in the following matrix form

$$\mathcal{E} = Tr[M^T M] - 2Tr[D^{(n+1)} W^{(n)} M^T] + Tr[D^{(n+1)T} D^{(n+1)}] \tag{4.14}$$

Since the first and third terms of this expression do not depend on the alignment of the point-sets we can turn our attention to maximising the quantity

$$\mathcal{F} = Tr[D^{(n+1)} W^{(n)} M^T] \tag{4.15}$$

where $D^{(n+1)}$ is the revised matrix of point-positions which we aim to estimate via minimisation of $\mathcal{E}$. This quantity can be thought of as a weighted measure of overlap or correlation between the point-sets under the current alignment estimate. It is worth pausing to consider its relationship with their measures exploited elsewhere in the literature on point pattern matching. The quantity $Tr[DM^T]$ is simply the standard measure of overlap that is minimised in the work on least-squares alignment (Umeyama, 1991). In Procrustes analysis singular value decomposition is performed on the matrix DM^T to recover the rotation matrix. However, the ordering of the points in the two co-ordinate matrices D and M must be identical and the two point-sets must be of the same size. In other words, the points must be labelled in advance. By contrast in the case of the weighted measure of overlap DWM^T, the ordering is no longer critical since the elements of the weighting matrix W represent correspondence probabilities between the columns of the matrix D and those of the matrix M. Moreover the matrices M and D may have different numbers of columns, i.e. the point-sets may be of different sizes. Finally, it is important to note that matrix W, is just the inter-image proximity matrix used by Scott and Longuet-Higgins (Scott and Longuet-Higgins, 1991) in their work on recovering correspondences by singular value decomposition. So, the utility measure delivered by the EM algorithm plays a synergistic role. The role of the inter-image point proximity matrix W

is to encode correspondence information and to weight the computation of the correlation between the un-ordered columns of the point position matrices.

The quantity $\mathcal{F}$ can be maximised by performing a singular value decomposition. The procedure is as follows. The matrix $D^{(n)}W^{(n)}M^T$ is factorised into a product of three new matrices U, V and Δ, where Δ is a diagonal matrix whose elements are either zero or positive, and U and V are orthogonal matrices. The factorisation is as follows

$$D^{(n)}W^{(n)}M^T = U\Delta V^T \tag{4.16}$$

The matrices U and V define a rotation matrix Θ which aligns the principal component directions of the point-sets M and D. The rotation matrix is equal to

$$\Theta = VU^T \tag{4.17}$$

With the rotation matrix to hand we can find the Procrustes alignment which maximises the correlation of the two point sets. The procedure is to first bring the centroids of the two point-sets into correspondence. Next, the data points are scaled so that they have the same variance as those of the model. Finally, the scaled and translated point-sets are rotated so that their correlation is maximised.

To be more formal the centroids of the two point-sets are

$$\mu_D^{(n)} = \frac{\sum_{i\in\mathcal{D}} \vec{u}_i^{(n)}}{|\mathcal{D}|} \tag{4.18}$$

and

$$\mu_M = \frac{\sum_{j\in\mathcal{M}} \vec{v}_j}{|\mathcal{M}|} \tag{4.19}$$

The corresponding covariance matrices are

$$\Sigma_D^{(n)} = \frac{\sum_{i\in\mathcal{D}} (\vec{u}_i^{(n)} - \mu_D^{(n)})(\vec{u}_i^{(n)} - \mu_D^{(n)})^T}{|\mathcal{D}|} \tag{4.20}$$

and

$$\Sigma_M = \frac{\sum_{j \in \mathcal{M}} (\vec{v}_j - \mu_M)(\vec{v}_j - \mu_M)^T}{|\mathcal{M}|} \tag{4.21}$$

With these ingredients the update equation for re-aligning the data-points is

$$\vec{u}_i^{(n+1)} = \mu_M + \sqrt{\frac{Tr\Sigma_M}{Tr\Sigma_D}} V U^T (\vec{u}_i^{(n)} - \mu_D^{(n)}) \tag{4.22}$$

4.3.3 Expectation

In the expectation step of the EM algorithm the *a posteriori* probabilities of the missing data (i.e. the model-graph measurement vectors, $\vec{v}_j$) are updated by substituting the revised parameter vector into the conditional measurement distribution. Using the Bayes rule, we can re-write the *a posteriori* measurement probabilities in terms of the components of the corresponding conditional measurement densities

$$P(\vec{v}_j | \vec{u}_i, \Phi^{(n+1)}) = \frac{\alpha_j^{(n)} p(\vec{u}_i | \vec{v}_j, \Phi^{(n)})}{\sum_{j' \in \mathcal{M}} \alpha_{j'}^{(n)} p(\vec{u}_i | \vec{v}_{j'}, \Phi^{(n)})} \tag{4.23}$$

The mixing proportions are computed by averaging the *a posteriori* probabilities over the set of data-points, i.e.

$$\alpha_j^{(n)} = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} P(\vec{v}_j | \vec{u}_i, \Phi^{(n)}) \tag{4.24}$$

4.4 Experiments

The data used in our study is furnished by randomly generated point-sets. We have added two types of noise to the point-sets. Firstly, we have added Gaussian measurement errors to the positions of the points. The position errors are isotropic and of zero mean. The parameter of the noise process is the standard

deviation of the measurement error. The second type of noise is structural error. Here we have added controlled fractions of clutter points at random locations.

4.4.1 Iterative behaviour and effect of noise and initialisation

We commence by studying the effect of differences in rotation and scale on the rate of convergence. The alignment error is defined as the average distance between corresponding feature points. Figures 4.1 and 4.2 show the alignment error as a function of iteration number for the new matching method. The different curves are different initial rotations and scale-differences. From the plots we can see that the iterative Procrustes alignment algorithm converges within 8 iterations for a wide range of angles and scalings.

Next we turn our attention to the effects of various types of error. We measure the effectiveness of the various algorithms used in our study by comparing the final alignment and correspondence errors. The alignment error is the root-mean square distance between the final point position and the corresponding ground-truth point position. For the experiments with alignment error we provide three performance curves. The curve marked SP is the result obtained with the conventional Procrustes method if the points are labelled, i.e. the correspondences are known in advance. The curve labelled WP is the initial alignment obtained after one iteration of the weighted Procrustes method. The curve labelled IP is the result obtained after 10 iterations of the weighted Procrustes method. The difference between the curves labelled WP and IP indicates the improvements obtained by iterative application of our new method. We deem correspondences to be correct if the maximum *a posteriori* correspondence probability returns the label of the ground-truth point. The correspondence error is the fraction of points which

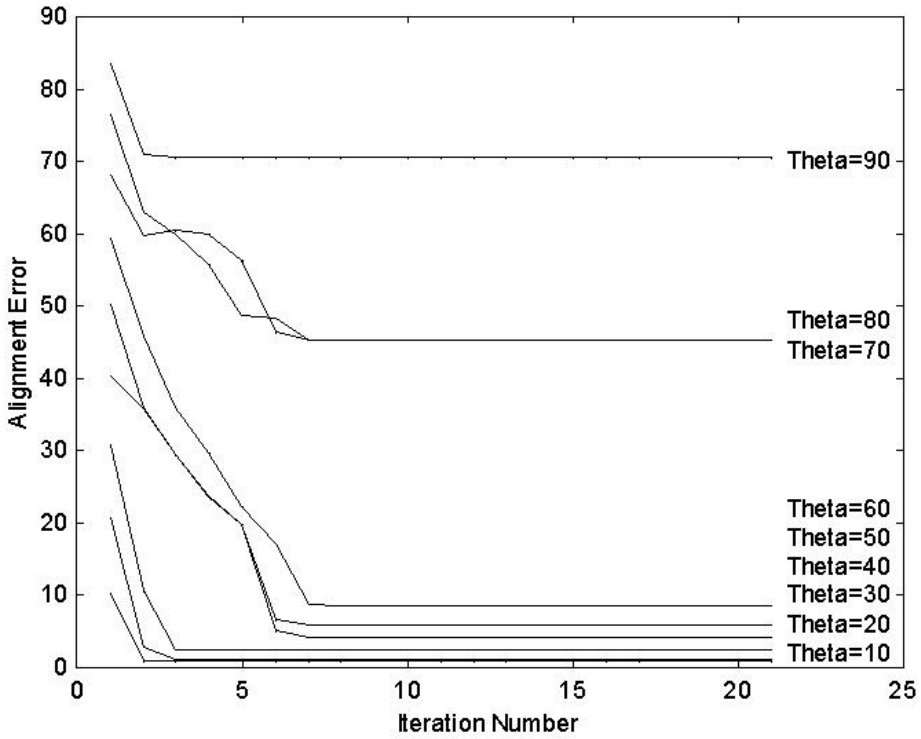


Figure 4.1: Convergence rate for rotated patterns.

are incorrectly labelled in this way. To give some idea of the benefits of iteration, we also report the correspondence results obtained with our new method after a single iteration (the curves marked "WP") and after 10 iterations (the curves marked "IP"). However, since the standard Procrustes method requires correspondences as input, the alternative algorithm is the Scott and Longuet-Higgins method. The curves for this method are labelled SLH.

We commence by investigating the effect of point-position errors. Here we have added random Gaussian jitter to the point-sets. The parameter of the noise process is the standard deviation of the Gaussian. Figure 4.3 shows the alignment error as a function of the standard deviation. Here, as expected, there is a clear linear relationship between the RMS alignment error and the standard deviation of the added Gaussian noise. There is little to distinguish the iterative

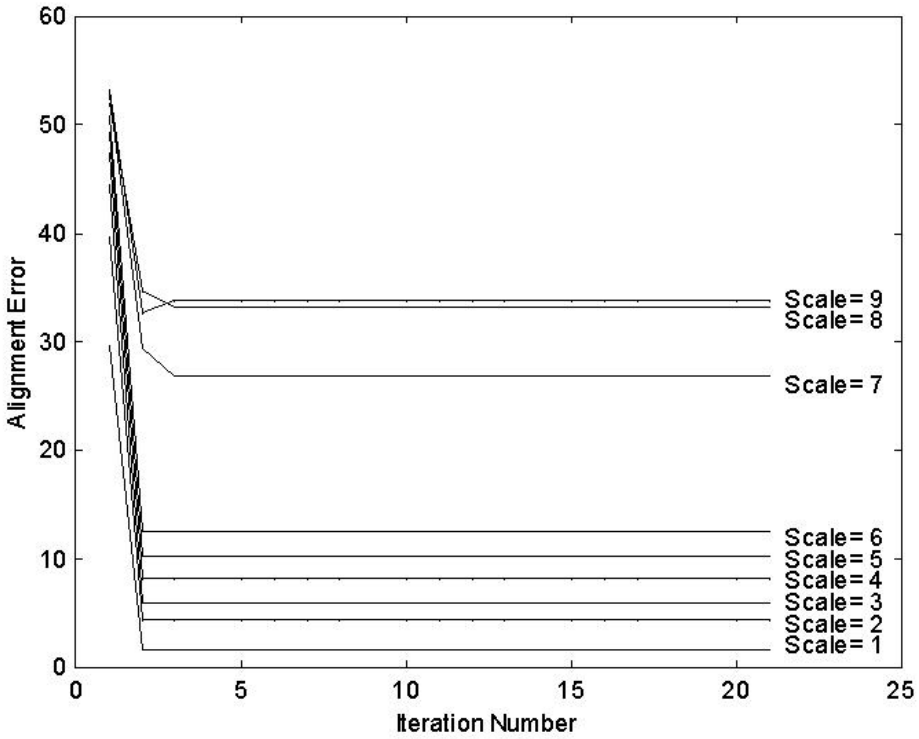


Figure 4.2: Convergence rate for scaled patterns.

weighted, single-step weighted and conventional Procrustes methods. When we turn our attention to the correspondence error shown in Figure 4.4 a different pattern emerges. Figure 4.4 shows that both of the iterative Procrustes method and Scott and Longuet-Higgins method return good correspondence result if the position error is small.

In Figures 4.5 and 4.6 we turn our attention to the effect of clutter. Here we have added a controlled fraction of contaminating points at random locations to the data-point set. Again there is no significant difference in the alignment errors returned by the three variants of the Procrustes method. In the case of correspondence error the iterative weighted Procrustes method outperforms the single-step weighted method. However, the Scott and Longuet-Higgins method returns the best results.

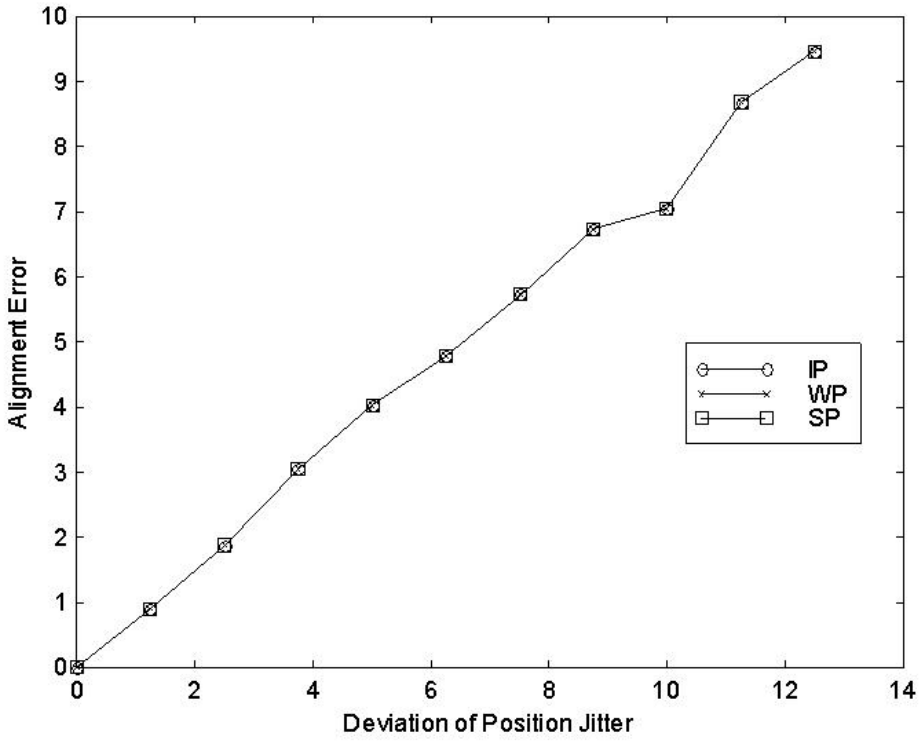


Figure 4.3: Alignment error as a function of point position error.

Figures 4.7 and 4.8 show the effect of initial rotation. The final alignment error is shown in Figure 4.7. Comparing the results obtained with a single iteration of the weighted Procrustes method and 10 iterations, there is a marked improvement. The new iterative method does not begin to fail until the rotation angle exceeds 50 degrees. The correspondence errors are shown in Figure 4.8. Here there is again a significant improvement after 10 iterations. It is interesting to note that the Scott and Longuet-Higgins algorithm fails when the rotation angle exceeds 20 degrees. This inability to cope with large rotations is a well documented problem with this method.

In Figures 4.9 and 4.10 we repeat the experiments under differences in scale. Again (see Figure 4.9), commencing from unknown correspondence the iterative method improves the alignment error. In the case of correspondence error (Fig-

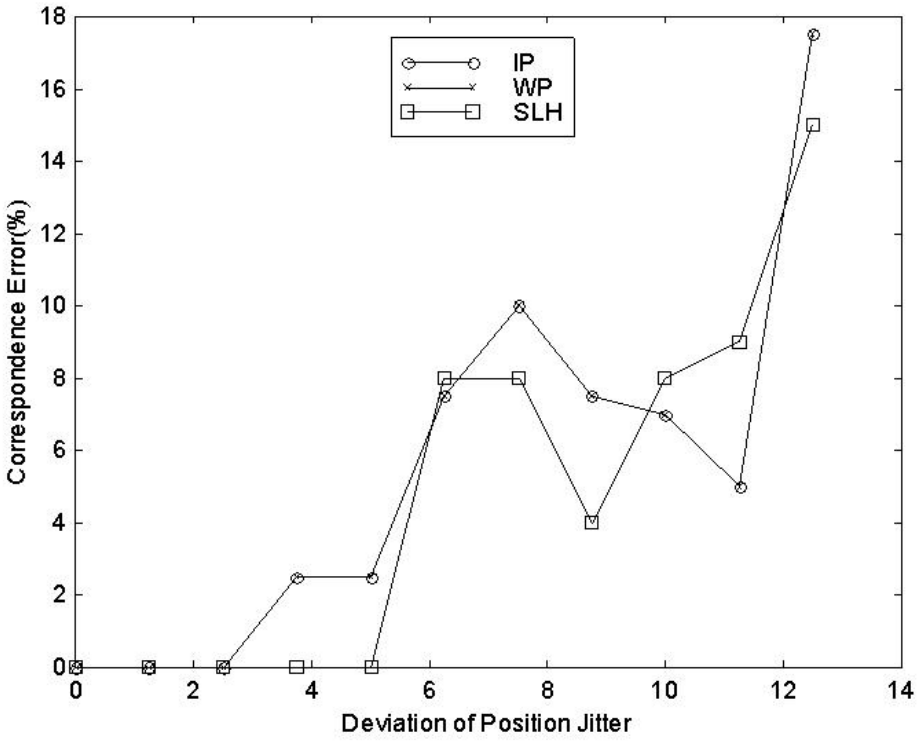


Figure 4.4: Correspondence error as a function of point position error.

ure 4.10), our method has similar performance with that of Scott and Longuet-Higgins.

The bar-plots shown in Figures 4.11 and 4.12 show the effects of mixing the different types of error described above. In each case the left-hand plot contains no clutter (i.e. the point-sets are of the same size) while the right-hand plot contains clutter (i.e. the point-sets are of different size). The main point to note from these two plots is that the new iterative weighted Procrustes method offers the best combination of alignment and correspondence accuracy.

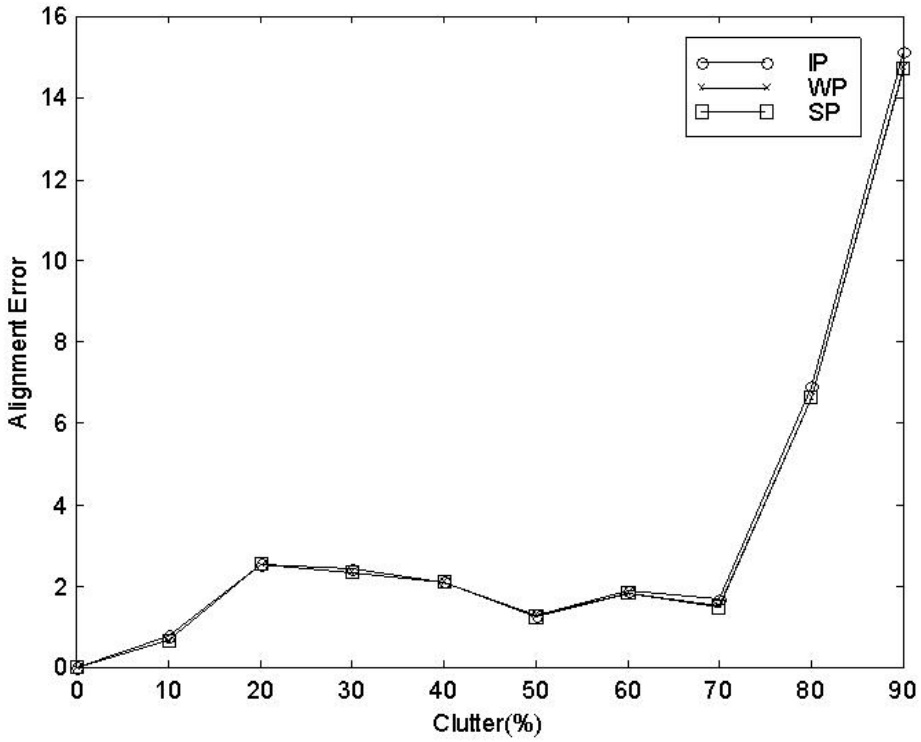


Figure 4.5: Alignment error as a function of percentage of clutter.

4.4.2 Real-world Examples

Finally, we provide some examples on real-world data. We use pairs of wide-baseline uncalibrated stereo images and a sequence of CMU/VASC model house images to test the proposed algorithm. The point-features are corners detected using the recently reported method of Luo, Cross and Hancock(Luo et al., 1999).

We commence with some examples of the matching results obtained with two images of beams in the loft of our Lab. Figure 4.13 shows the data used in an experiment in which there is a small shift in position, scale and rotation. Panels a) and b) show the raw images with the features super-imposed. Figure 4.14 shows the results of aligning these images with our iterative Procrustes method. In this figure panel a) shows the initial alignment. Panel b), c) and d) show the

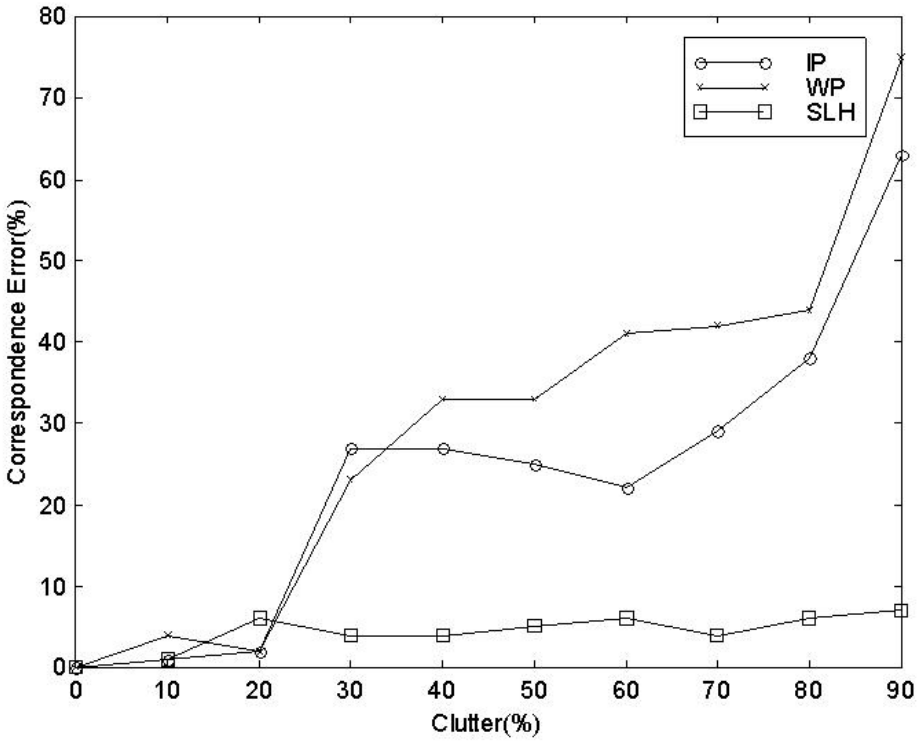


Figure 4.6: Correspondence error as a function of percentage of clutter.

alignments obtained after 1, 2 and 10 iterations. The final match is good.

Figure 4.15 shows a second example in which there are significant affine distortions and perspective effects due to the different depths of the features in the scene. Again panels a) and b) show the raw images. In panel c) we show the initial alignment, while panel d) shows the alignment obtained after 10 iterations. Despite the departures from Euclidean geometry the alignments are satisfactory.

Our next experiment is to test our iterative Procrustes alignment method on the CMU/VASC model-house images. We use a sequence of 10 images. For the sake of clarity, we only show the first six images in the thesis. They are shown in figure 4.16. The points marked on the images are corners extracted by using the corner detector which was developed in Chapter 3.

In figure 4.17, the initial pose differences are displayed by overlapping the

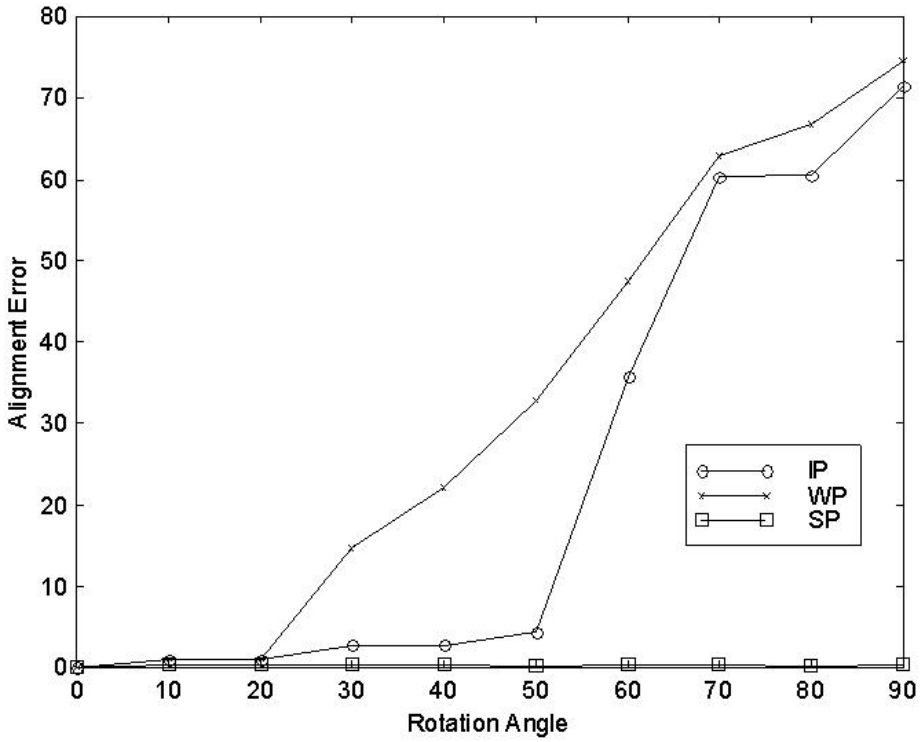


Figure 4.7: Alignment error as a function of rotation angle.

Delaunay triangulations of the feature points in the first image on the subsequent grey-scale images.

Figure 4.18 shows the alignment results using the iterative Procrustes method. There is very small alignment error for the first two images. The algorithm fails to align the images from the 5th image of the sequence. The reason can be found in the figures form 4.19 to 4.23 and table 4.2 where there are too many false correspondences found from the 5th image.

House index	0	1	2	3	4	5	6	7	8	9
IP	-	0.85	2.37	13.2	14.1	22.9	21.9	40.0	43.5	46.2

Table 4.1: Alignment errors of the IP method.

Next we give two tables to summarise the house image alignment and corre-

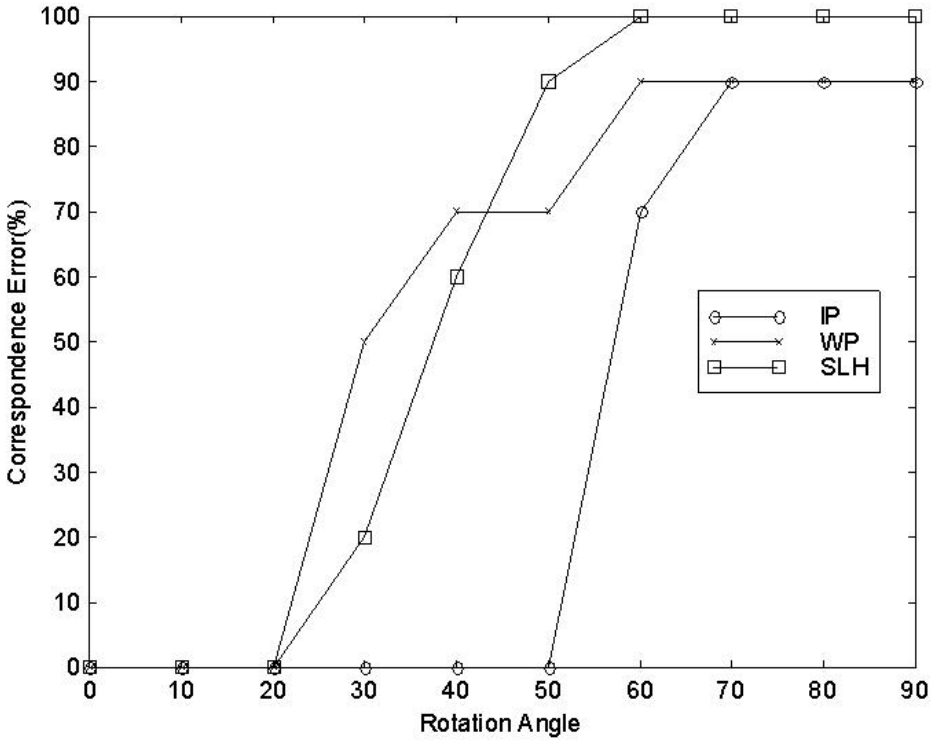


Figure 4.8: Correspondence error as a function of rotation angle.

spondence results. Table 4.1 shows the alignment error and Table 4.2 summarises the correspondence allocating results. In table 4.2, the number of correspondences which are found correctly, falsely and unmatched are listed.

House index	0	1	2	3	4	5	6	7	8	9
Correct	-	22	11	11	9	4	3	2	2	0
False	-	0	1	4	3	9	9	10	10	8
Missed	-	8	18	15	18	17	18	18	18	22

Table 4.2: Correspondence allocation results of the IP method.

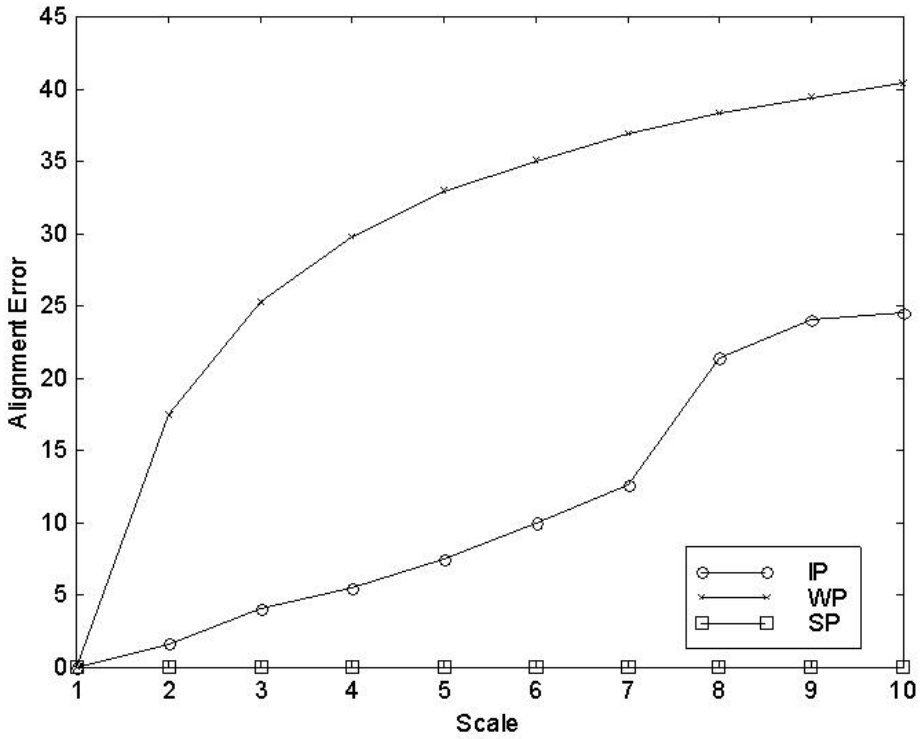


Figure 4.9: Alignment error as a function of scale level.

4.5 Conclusions

In conclusion, we have shown how the process of Procrustes alignment can be formulated as a maximum likelihood estimation problem using the EM algorithm. This interpretation leads to a new point-set similarity measure in which point correspondence probabilities weight the standard least-squares point overlap distance. In other words, our new measure of point-set similarity combines the ideas already developed by Scott and Lonquet-Higgins, and, Umeyama in a single statistical utility measure.

When viewed from the perspective of the conventional Procrustes alignment method, our new iterative weighted method offers two advantages. First, it does not require correspondences to be provided, i.e. the points to be labelled in

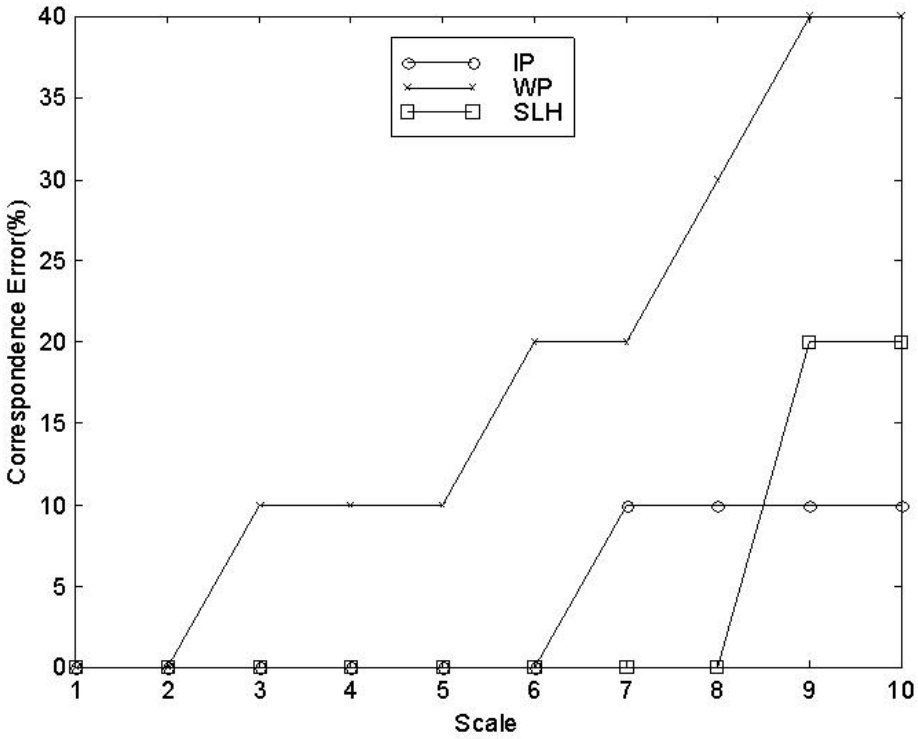
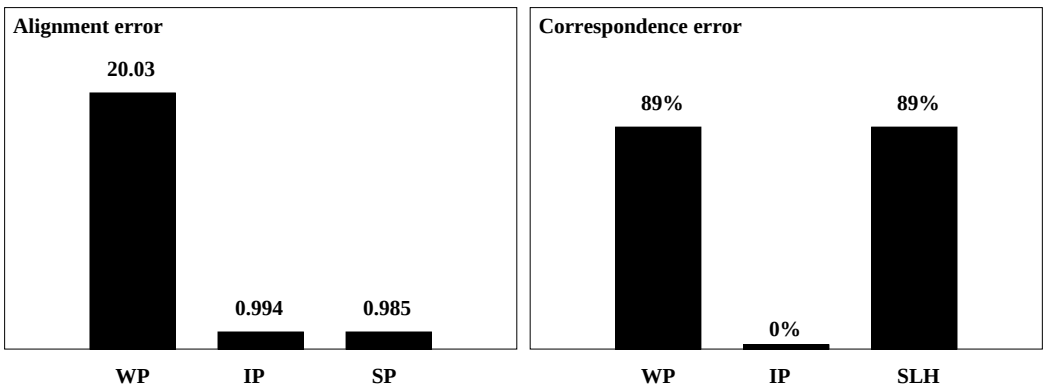
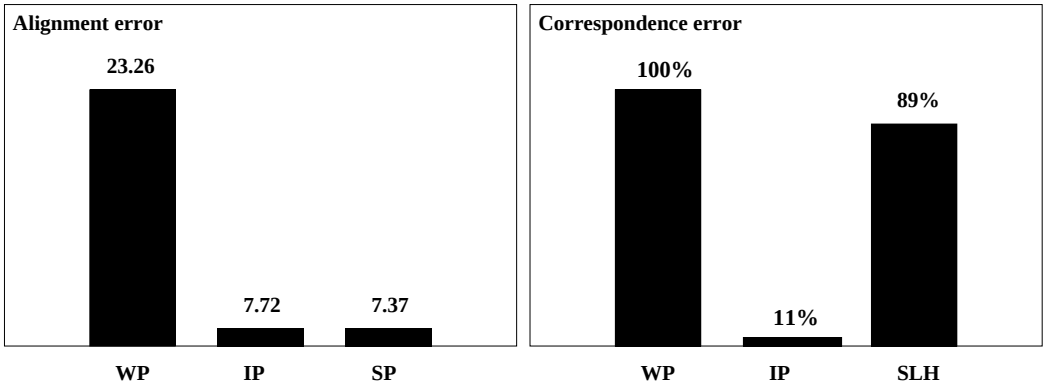


Figure 4.10: Correspondence error as a function of scale level.



(a) S=2,R=50°,P=0,C=0% (b) S=2,R=50°,P=0,C=10%

Figure 4.11: Comparison of the Weighted Procrustes, Iterative Procrutes and Standard Procrustes methods under scaling(S), rotation(R), position random error(P) and clutter(C).



(a) S=5,R=50°,P=10,C=0% (b) S=5,R=50°,P=10,C=10%

Figure 4.12: Comparison of the Weighted Procrustes, Iterative Procrutes and Standard Procrustes methods under scaling(S), rotation(R), position random error(P) and clutter(C).

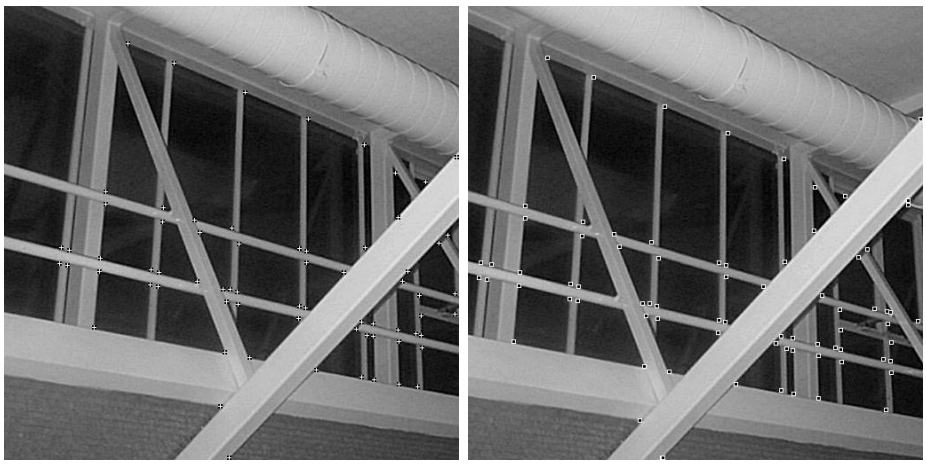
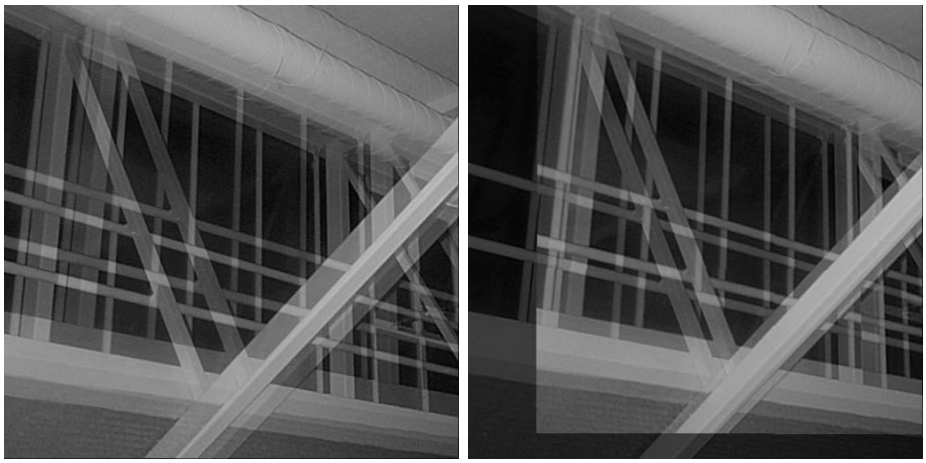


Figure 4.13: Ceiling images with features.

advance. Second, it allows us to align point-sets of different sizes.

There are clearly a number of ways in which the work reported in this chapter can be extended. The most obvious shortcoming of the work described here is that it confines its attention to similarity transformations. The Procrustes method has been extended to more complicated transformations and our new EM framework



(a) Overlapped original images

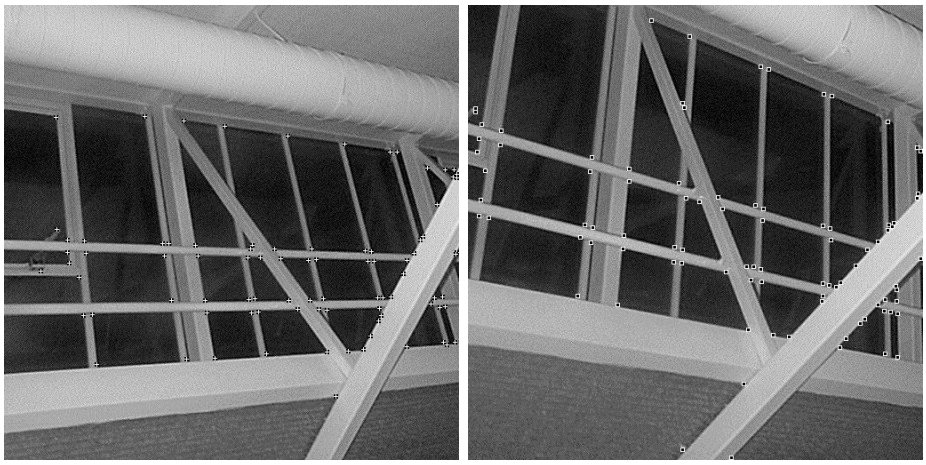
(b) 1 Iteration

(c) 2 Iterations

(d) 10 Iterations

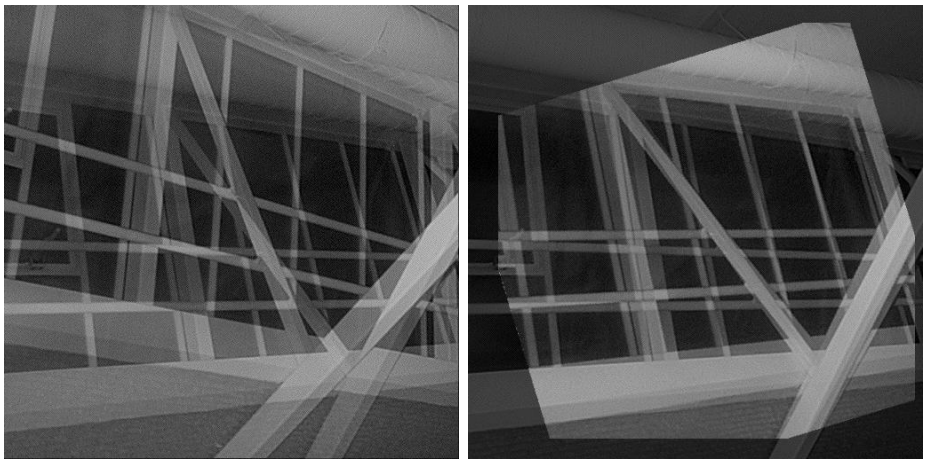
Figure 4.14: Ceiling image alignment results.

would generalise easily to these extensions.



(a) First image with features

(b) Second image with features



(c) Overlapped original images

(d) Final alignment

Figure 4.15: Another example of ceiling image alignment.

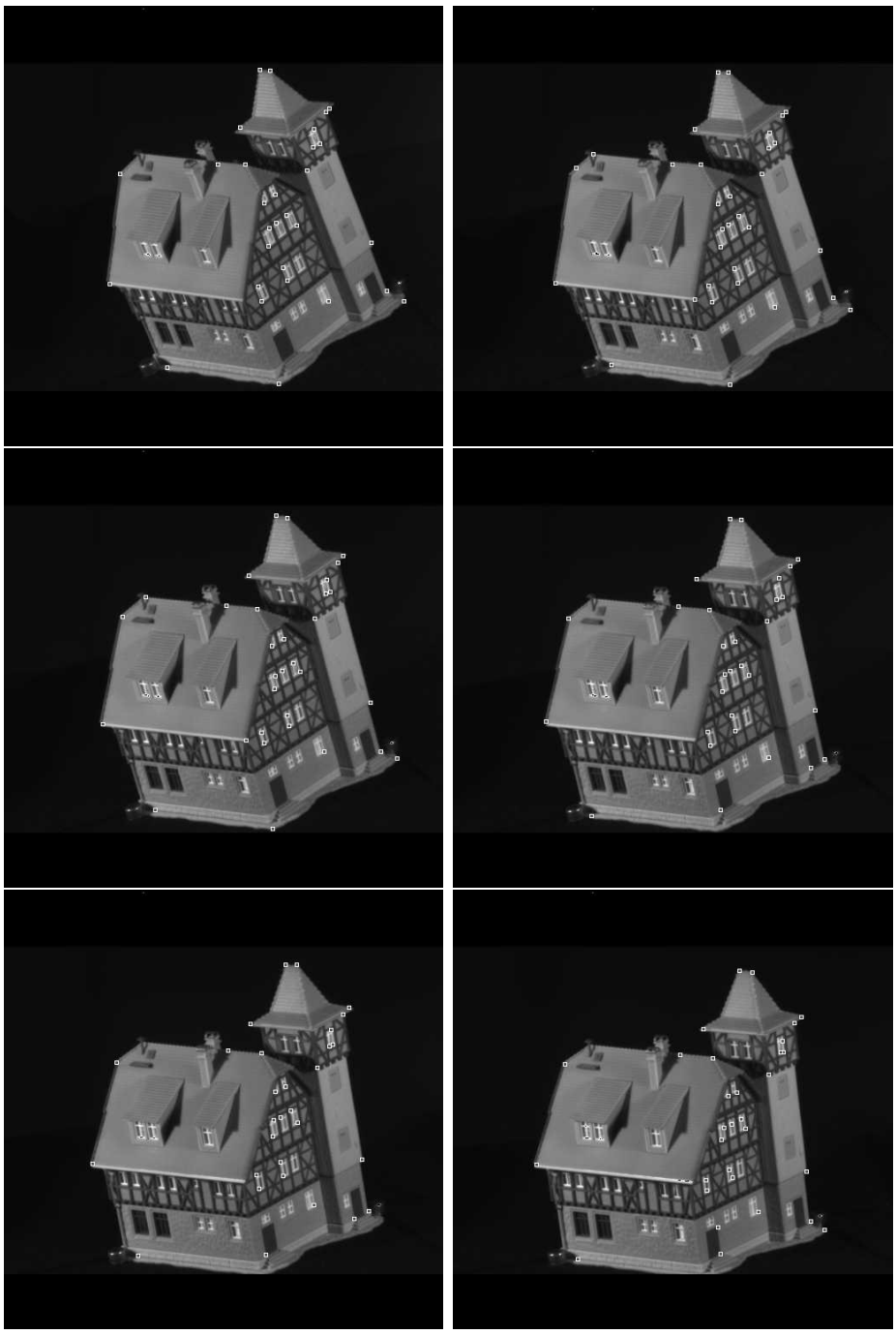


Figure 4.16: House images with feature points.

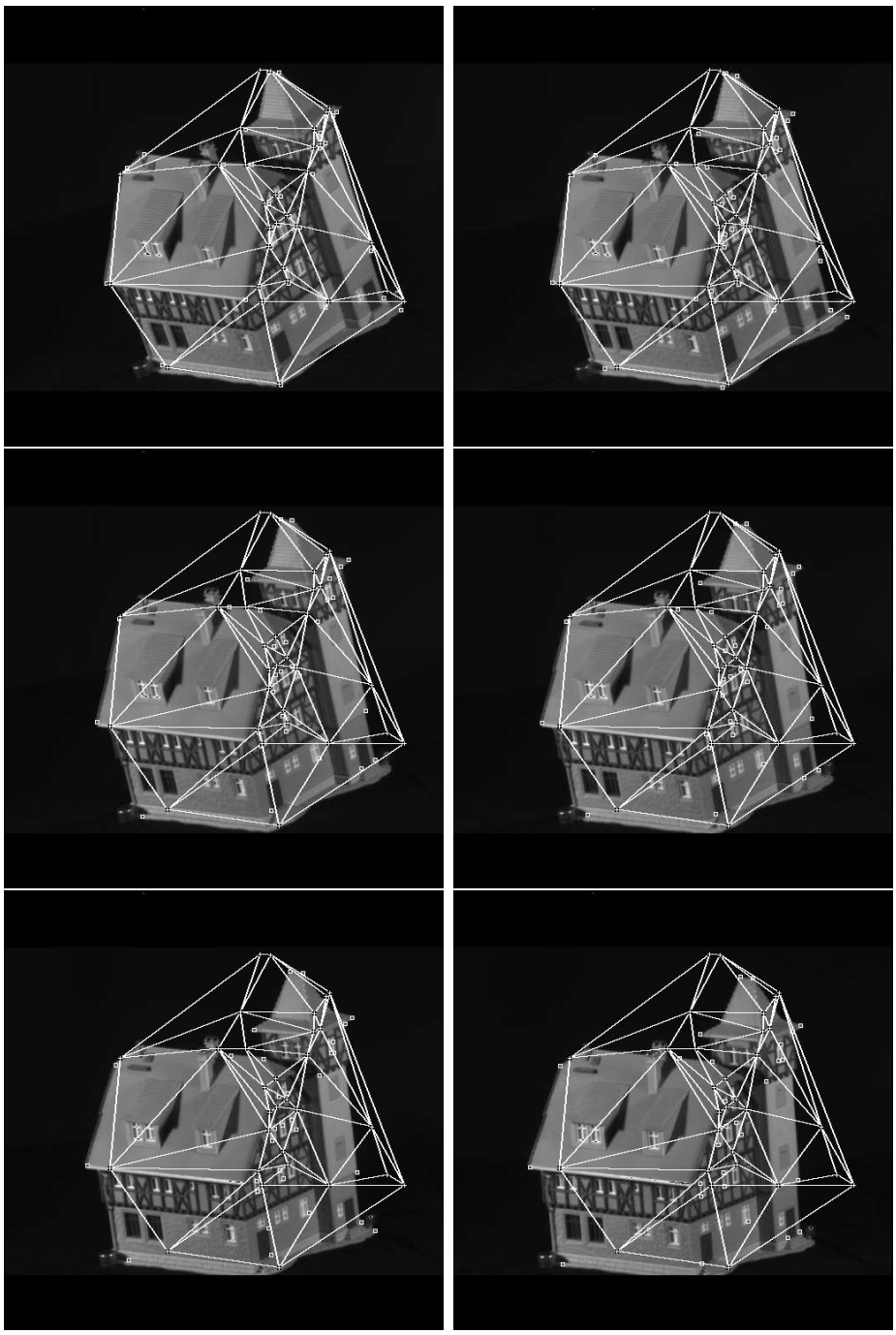


Figure 4.17: Initial pose differences.

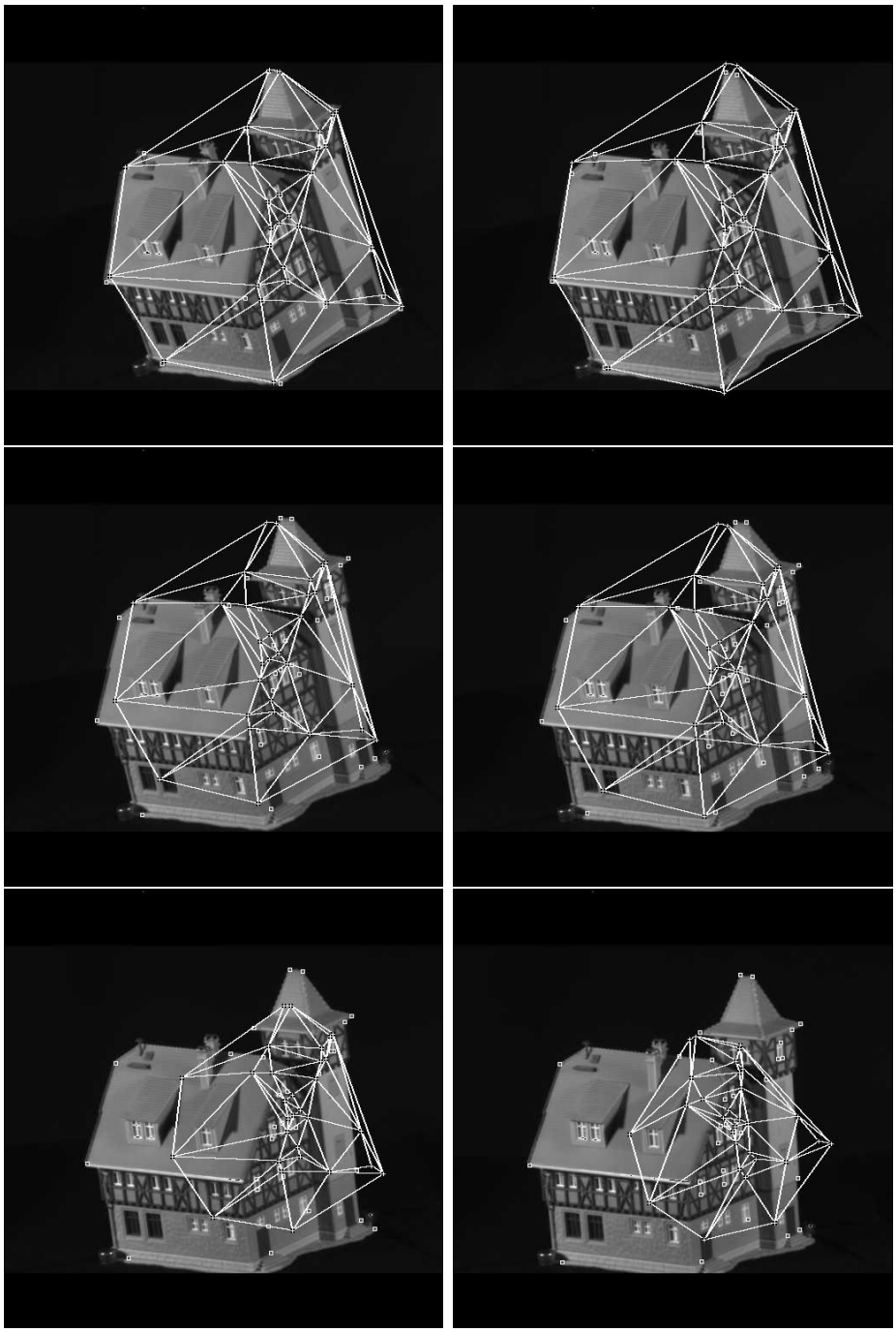


Figure 4.18: House image alignment results(IP).

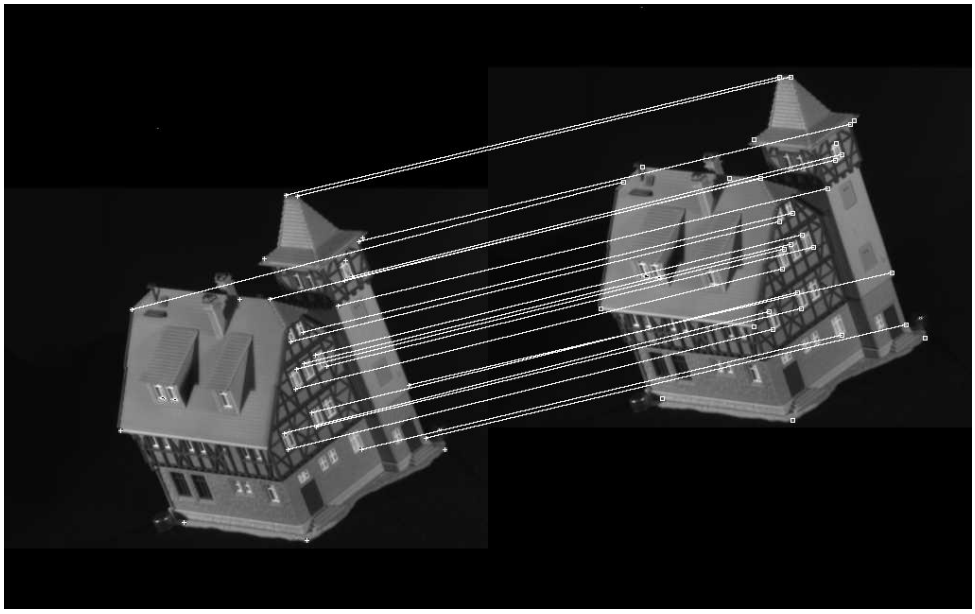


Figure 4.19: Correspondences between the first and the second images.

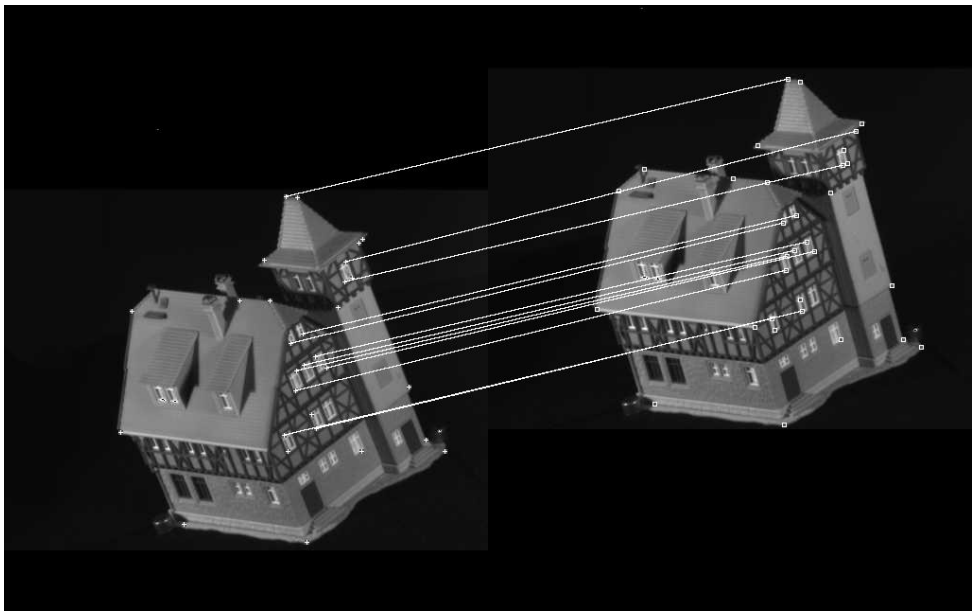


Figure 4.20: Correspondences between the first and the third images.

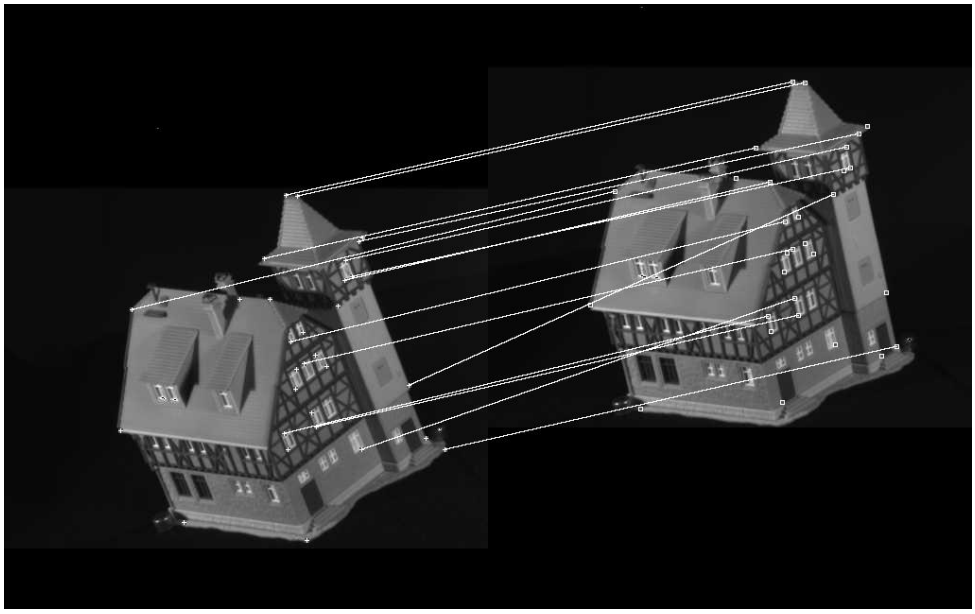


Figure 4.21: Correspondences between the first and the fourth images.

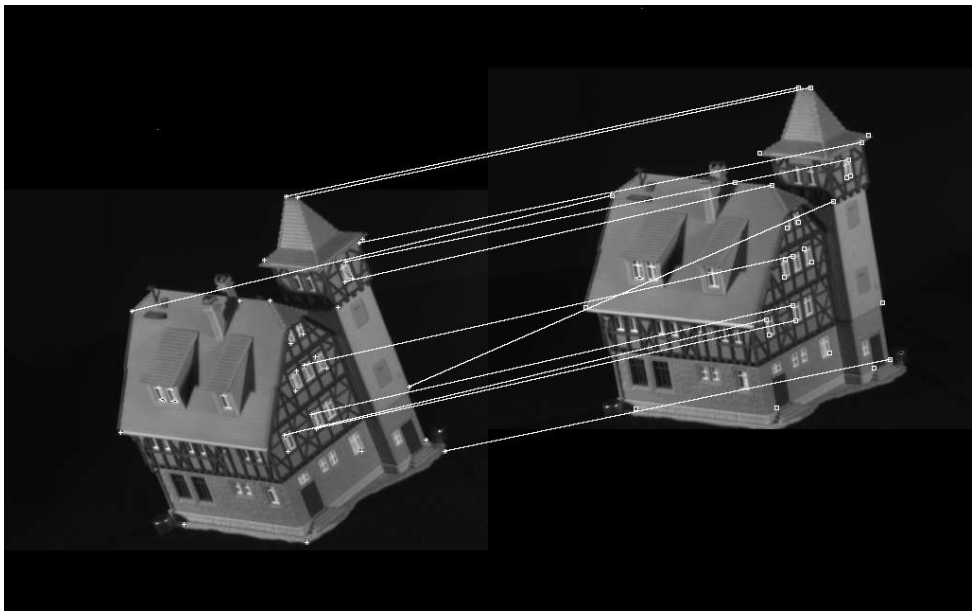


Figure 4.22: Correspondences between the first and the fifth images.

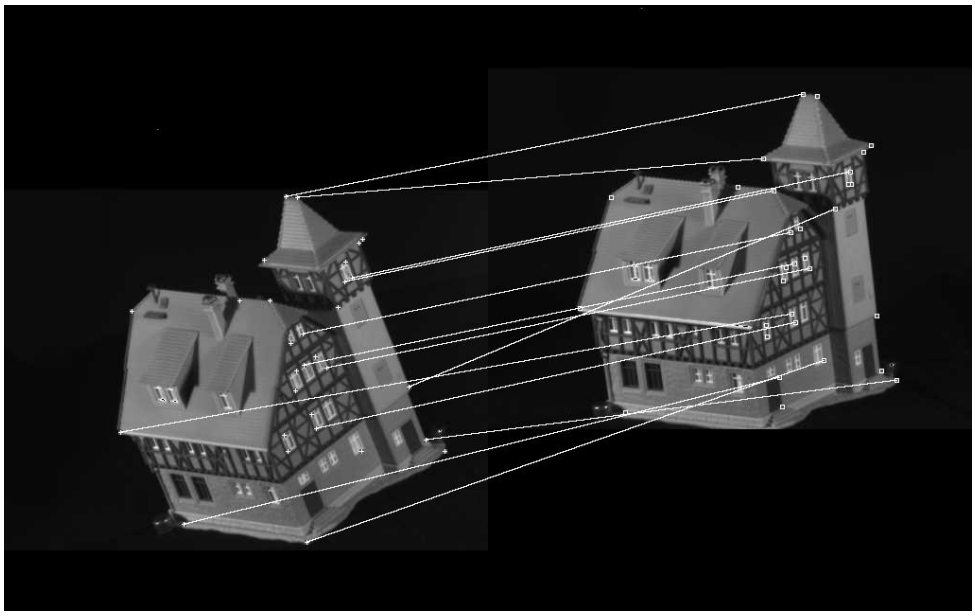


Figure 4.23: Correspondences between the first and the sixth images.

Chapter 5

Correspondence through graph matching

In the previous chapter we showed how the EM algorithm could be used for the purpose of iterative Procrustes analysis. In this chapter we consider how the method can be used to solve the correspondence allocating problem when point-sets are abstracted using neighbourhood graphs.

Graph-matching is a task of pivotal importance in high-level vision since it provides a means by which abstract pictorial descriptions can be matched to one-another. Unfortunately, since the process of eliciting graph structures from raw image data is a task of some fragility due to noise and the limited effectiveness of the available segmentation algorithms, graph-matching is invariably approached by inexact means (Shapiro and Haralick, 1985; Sanfeliu and Fu, 1983). Because of this many high-level matching techniques have weakened the role of structural information and have relied more heavily on the use of attribute relations. This is disappointing since structural graph representations provide abstractions that convey important visual invariances. It is for this reason that we return to the problem of structural graph-matching in this chapter. We make two contribu-

tions. First, we aim to render the process robust to structural error using the apparatus of the EM algorithm. Second, we cast the resulting statistical utility measure into a matrix setting and show how matching can be realised using singular value decomposition.

From the review of the literature presented in Chapter 2, we draw the following observations: Firstly, the use of principled probabilistic methods for gauging similarity has met with considerable success for inexact graph-matching. Secondly, although more computationally elegant, matrix methods for graph-matching have failed to cope with the realistic cases of inexact graph-matching or matching graphs of different size. Moreover, there has been little attempt to combine these two pieces of work.

Based on these observations our aim in this chapter is to cast the statistical matching of graphs into a matrix representation and to exploit singular value methods to efficiently recover correspondences. We commence by developing a likelihood function for the graph-matching problem. This treats the graph to be matched (the data-graph) as observed data and the set of correspondences with the available model (the model-graph) as hidden variables. Accordingly, we construct a mixture model over the set of correspondences between the nodes of the data-graph and those of the model-graph. We adopt a Bernoulli model for the probability distribution of the correspondence errors encountered in matching the data-graph to the model-graph. The existence or otherwise of correspondence errors is gauged using the edge-consistency of the pattern of matches.

With the likelihood function for the graph-matching problem to hand, we develop an algorithm for recovering the pattern of correspondence matches. Since the likelihood-function has a mixture-structure we use the expectation maximisation (EM) algorithm of Dempster, Laird and Rubin (Dempster et al., 1977) to iteratively estimate a set of assignment variables which indicate the state of

match between the two graphs. The EM algorithm provides a principled way for recovering maximum likelihood solutions to problems posed in terms of missing or hidden data. From a computational standpoint, the EM algorithm relies on interleaved iterative steps. In the maximisation step, a parameter are estimated so as to maximise the value of the expected log-likelihood function. The expectation step updates the *a posteriori* probabilities of the hidden variables, which in their turn are needed to weight contributions to the expected log-likelihood function.

By adopting a matrix representation, we show that when the distribution of correspondence errors is modeled using a Bernoulli distribution, then the expected log-likelihood function of the EM algorithm is related to the weighted product of the adjacency matrices. The weighting facilitated by the EM algorithm provides a means of excluding structural errors. With this principled similarity measure to hand, the maximisation step of the EM algorithm can be realised via singular value decomposition. Specifically, the diagonalisation of the weighted adjacency measure delivers correspondence matches. The resulting matching method can be applied to graphs of different size. Moreover, as we will demonstrate in our experiments, it can accommodate severe levels of structural corruption.

Finally, it is worth comparing our iterative matrix-based graph-matching method with the use of subgraph isomorphisms for matching. This is a classical approach. One of the best known algorithms for locating exact subgraph isomorphisms is that of Ullman (Ullman, 1976), which uses tree search with backtracking. For inexact graph-matching there are several extensions of the idea which include the edit-distance methods of Eshera and Fu (Eshera and Fu, 1986) and Sanfeliu and Fu (Sanfeliu and Fu, 1983) and the method developed by Bunke and Allerman (Bunke and Allermann, 1983). An important development is the association-graph method of Ambler *et al*(Ambler and et al., 1973). Here

subgraph isomorphisms which correspond to the maximum common subgraph are found by searching for the maximum clique of the association graph. The method has been used successfully for stereo correspondence by Horaud and Skordas (Horaud and Skordas, 1989). Moreover, Wilson and Hancock (Wilson and Hancock, 1997) have shown how the association graph can be combined with a Bayesian discrete relaxation process to match graphs which are subject to considerable structural corruption. Pelillo (Pelillo et al., 1999) has developed a probabilistic relaxation scheme for efficiently computing the size of the maximum common subgraph. Bunke has shown the relationship between the size of the maximum common subgraph and the graph edit distance (Bunke, 1999). Further work by Bunke and Messmer (Bunke and Messmer, 1997), has developed a technique in which the search for a maximum common subgraph can be rendered efficient using a subgraph decomposition and by tabulating the associated isomorphisms. The main similarities and difference between our EM algorithm and these subgraph isomorphism methods are as follows. The main difference is that we adopt a probabilistic framework and use optimisation rather than search. Hence, while our method facilitates evidence combination, it is prone to convergence to local optima. The similarity of the method resides in the fact that the matrix product of the correspondence probabilities with the adjacency matrix of the two graphs plays a role which is reminiscent of an association structure.

5.1 A Likelihood Function for Graph Matching

Our overall goal in this chapter is to develop a maximum likelihood framework for structural graph matching. In this section we develop the likelihood function underpinning our study. To commence we must define some notation. We use the notation $G = (V, E)$ to denote the graphs under match, where V is the set of

nodes and E is the set of edges. Our aim in matching is to associate nodes $V_D = \{x_1, x_2, ...x_{|V_D|}\}$ in a graph $G_D = (V_D, E_D)$ representing data to be matched against those from the set $V_M = \{y_1, y_2, ..., y_{|V_M|}\}$ in a graph $G_M = (V_M, E_M)$ representing an available model. Formally, the matching is represented by a function from the nodes in the data graph G_D to those in the model graph G_M. Suppose that the state of match between the two graphs is represented by the function $f : V_D \rightarrow V_M$ from the nodes of the data-graph to those of the model-graph. We will use latin letters to denote nodes from the data-graph and greek letters to denote nodes from the model-graph. Hence, the statement $f^{(n)}(a) = \alpha$ means that the node $a \in V_D$ is assigned the label or symbol $\alpha \in V_M$.

One of the goals in this chapter is to show how the two graphs can be matched using matrix factorisation methods. We therefore introduce some matrix notation to represent the graphs. To this end we define a $|V_D| \times |V_M|$ matching matrix $S^{(n)}$ whose elements are assignment variables which convey the following meaning

$$s_{a\alpha} = \begin{cases} 1 & \text{if } f(a) = \alpha \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

We represent the structure of the two graphs using a $|V_D| \times |V_D|$ adjacency matrix D for the data graph and a $|V_M| \times |V_M|$ adjacency matrix M for the model graph. The elements of the adjacency matrix for the data graph are defined as follows

$$D_{ab} = \begin{cases} 1 & \text{if } (a, b) \in E_D \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

while those for the model graph are defined to be

$$M_{\alpha\beta} = \begin{cases} 1 & \text{if } (\alpha, \beta) \in E_M \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

Since we are working with undirected graphs, the two adjacency matrices are symmetric, i.e. $D = D^T$ and $M = M^T$.

Having introduced the necessary formalism, we now proceed to develop our maximum likelihood framework for graph-matching. We seek the matrix of assignment variables that maximises the likelihood of the observed data-graph. Hence, we seek the matrix of assignment variables which satisfies the condition

$$S = \arg\max_{\hat{S}} P(G_D|\hat{S}) \tag{5.4}$$

Underpinning our model of the likelihood function is the idea that the correspondence matches assigned to the nodes of the data-graph are hidden variables which have arisen through a noisy observation process. In other words, we must entertain the possibility that any single node of the data-graph may be in correspondence with any of the nodes in the model-graph. To capture this feature of the graph-matching problem, we construct a mixture model over the set of possible correspondences. We follow the standard approach to constructing the likelihood function for a mixture distribution. This involves factorising the likelihood function over the observed data (i.e. the nodes of the data-graph) and summing over the hidden or unobserved variables (i.e. the corresponding nodes in the model-graph). As a result we write

$$P(G_D|S) = \prod_{a \in V_D} \sum_{\alpha \in V_M} p(x_a|y_\alpha, S) \tag{5.5}$$

where $p(x_a|y_\alpha, S)$ is the probability that data-graph node a is in correspondence with the model-graph node α under the matrix of assignment variables S.

In order to proceed, we require a model for the observation density $p(x_a|y_\alpha, S)$. We commence from the definition of conditional probability and write

$$P(x_a|y_\alpha, S) = \frac{P(x_a, y_\alpha, S)}{P(y_\alpha, S)} \tag{5.6}$$

Under the assumption that the observation density is factorial over the parameters

of the mixture model, i.e. the set of assignment variables, then we can write

$$P(x_a|y_\alpha, S) = \frac{\left\{\prod_{b\in V_D}\prod_{\beta\in V_M} P(s_{b\beta}|x_a, y_\alpha)\right\}P(x_a, y_\alpha)}{\left\{\prod_{b\in V_D}\prod_{\beta\in V_M} P(s_{b\beta}|y_\alpha)\right\}P(y_\alpha)} \tag{5.7}$$

After some re-arrangement using the definitions of conditional probability, we find that

$$P(x_a|y_\alpha, S) = \frac{\left\{\prod_{b\in V_D}\prod_{\beta\in V_M} \frac{P(x_a|y_\alpha, s_{b\beta})P(y_\alpha|s_{b\beta})P(s_{b\beta})}{P(x_a, y_\alpha)}\right\}P(x_a, y_\alpha)}{\left\{\prod_{b\in V_D}\prod_{\beta\in V_M} \frac{P(y_\alpha|s_{b\beta})P(s_{b\beta})}{P(y_\alpha)}\right\}P(y_\alpha)} \tag{5.8}$$

Cancelling the terms $P(y_\alpha|s_{b\beta})$ and $P(s_{b\beta})$ which appear under the products in the numerator and denominator, and collecting together terms, the above expression simplifies to

$$P(x_a|y_\alpha, S) = \left[\frac{1}{P(x_a|y_\alpha)}\right]^{|V_D|\times|V_M|-1} \prod_{b\in V_D}\prod_{\beta\in V_M} P(x_a|y_\alpha, s_{b\beta}) \tag{5.9}$$

If we further assume that the data-graph node x_a is conditionally dependant on the model-graph node y_α only in the presence of the correspondence matches S, then $P(x_a|y_\alpha) = P(x_a)$. Hence, we can write

$$P(x_a|y_\alpha, S) = B_a \prod_{b\in V_D}\prod_{\beta\in V_M} P(x_a|y_\alpha, s_{b\beta}) \tag{5.10}$$

where the constant

$$B_a = \left[\frac{1}{P(x_a)}\right]^{|V_D|\times|V_M|-1}$$

depends only on the identity of the data-graph node x_a.

Next we develop a model for the probability distribution for the observed set of correspondences between the nodes of the data and the model graphs given the current set of assignment parameters, i.e $P(x_a|y_\alpha, s_{b\beta})$. Our model draws on the recent work of Wilson and Hancock (Wilson and Hancock, 1997) and assumes that the observed data-graph nodes are derived from the model-graph nodes through a Bernoulli distribution. The parameter of this distribution is the probability

of correspondence error P_e. The idea behind this model is that the data-graph node x_a can emit a symbol y_α drawn from the set of model-graph nodes. The probability that this symbol is the correct correspondence is $1 - P_e$ while the probability that it is in error is P_e. To gauge the correctness of the emitted symbol, we check whether the nodes a and b of the data-graph are matched to a valid edge $(\alpha, \beta) \in E_M$ of the model-graph. To test for edge-consistency, we make use of the quantity $D_{ab}M_{\alpha\beta}s_{b\beta}$. This is unity if the label-assignment $f(b) = \beta$ can be made to node x_b, in such a way that the data-graph edge $(a, b) \in E_D$ is matched to an edge $(\alpha, \beta) \in E_M$ of the model-graph. When this condition is not met then the quantity is zero. In other words,

$$D_{ab}M_{\alpha\beta}s_{b\beta} = \begin{cases} 1 & \text{if } (a,b) \in E_D \text{ and } (\alpha,\beta) \in E_M \text{ and } f(b) = \beta \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

Using this switching property, the Bernoulli distribution becomes

$$P(x_a|y_\alpha, s_{b\beta}) = (1 - P_e)^{D_{ab}M_{\alpha\beta}s_{b\beta}} P_e^{1-D_{ab}M_{\alpha\beta}s_{b\beta}} \tag{5.12}$$

With the factorial assumption and the distribution rule to hand, the observation density becomes

$$P(x_a|y_\alpha, S) = B_a \prod_{b \in V_D} \prod_{\beta \in V_M} (1 - P_e)^{D_{ab}M_{\alpha\beta}s_{b\beta}} P_e^{1-D_{ab}M_{\alpha\beta}s_{b\beta}} \tag{5.13}$$

This expression is exponential in character. It can be re-written in as a natural exponential function

$$P(x_a|y_\alpha, S) = K_a \exp\left[\mu \sum_{b \in V_D} \sum_{\beta \in V_M} D_{ab}M_{\alpha\beta}s_{b\beta}\right] \tag{5.14}$$

where

$$\mu = \ln \frac{1 - P_e}{P_e} \tag{5.15}$$

and

$$K_a = P_e^{|V_D| \times |V_M|} B_a \tag{5.16}$$

Finally, the likelihood function becomes

$$P(G_D|G_M, S) = \prod_{a \in V_D} \sum_{\alpha \in V_M} K_a \exp\Big[\mu \sum_{b \in V_D} \sum_{\beta \in V_M} D_{ab} M_{\alpha\beta} s_{b\beta}\Big] \tag{5.17}$$

and the corresponding log-likelihood function for the assignment matrix is

$$\mathcal{L}(S) = \sum_{a \in V_D} \log\Big\{ \sum_{\alpha \in V_M} K_a \exp\Big[\mu \sum_{b \in V_D} \sum_{\beta \in V_M} D_{ab} M_{\alpha\beta} s_{b\beta}\Big]\Big\} \tag{5.18}$$

Unfortunately, because of the mixture structure the direct estimation of the matrix of assignment variables S from the log-likelihood function is not tractable in closed form. For this reason, in the next section we explain how the expectation-maximisation algorithm may be used instead.

5.2 Expectation-Maximisation

Having developed our computational model which poses the graph-matching problem in a maximum-likelihood framework, in this section we provide a concrete algorithm for recovering the parameters of the underlying mixture-model. We choose again to use the EM algorithm.

5.2.1 Expected log-likelihood function

The utility measure underpinning the EM algorithm is the conditional expected log-likelihood. The basic idea is to identify updated parameters that maximise their expected likelihood conditional upon the previously available iterates. This utility measure is frequently referred to as the incomplete data-likelihood. In our matching problem, the parameters are the discrete matching assignments to the nodes of the data-graph. Incompleteness originates from the fact that the matching configurations are not directly observable from the data. In other words, although we can observe the structure of the model and data graphs, the

matching-function f is hidden from us. The incomplete data-likelihood is obtained by weighting the individual contributions by the appropriate *a posteriori* matching probabilities. In a more general context, Dempster, Laird and Rubin (Dempster et al., 1977) observed that maximising a weighted log-likelihood function of this sort was equivalent to maximising the conditional expectation of the likelihood for a new parameter set given an old parameter set.

For our graph-matching problem, and from the Bayes theorem and the well-known development of the EM algorithm (Ripley, 1996; Bishop, 1995; McLachlan and K.E., 1988), maximisation of the expectation of the conditional likelihood is equivalent to maximising the weighted log-likelihood function

$$\Lambda(S^{(n+1)}|S^{(n)}) = \sum_{a \in V_D} \sum_{\alpha \in V_M} P(y_\alpha|x_a, S^{(n)}) \ln P(x_a|y_\alpha, S^{(n+1)}) \tag{5.19}$$

where $S^{(n)}$ indicates the matrix of assignment variables taken at iteration n of the EM algorithm. Hence, the *a posteriori* correspondence matching probabilities computed at iteration n, i.e. $P(y_\alpha|x_a, S^{(n)})$ are used to weight the iteration $n+1$ contributions to the log-likelihood function.

With the expected log-likelihood function to hand, the maximum-likelihood matrix of assignment variables is the one which satisfies the condition

$$S^{(n+1)} = \arg\max_{\hat{S}} \Lambda(\hat{S}|S^{(n)}) \tag{5.20}$$

One way to realise the update process is by parallel iterative local gradient ascent. In the next section we show how the expected log-likelihood function can be recast in a matrix framework. This allows us to realise the update procedure more efficiently using singular value decomposition.

5.2.2 Matrix Representation

To commence, we note that when the distribution function for the assignment variables is substituted from Equation (5.14) the expected log-likelihood function

becomes

$$\Lambda(S^{(n+1)}|S^{(n)}) = \sum_{a\in V_D}\sum_{b\in V_D}\sum_{\alpha\in V_M}\sum_{\beta\in V_M} Q_{a\alpha}^{(n)}\left\{\ln K_a + \mu D_{ab}M_{\alpha\beta}s_{b\beta}^{(n+1)}\right\} \tag{5.21}$$

where we have introduced the $|V_D| \times |V_M|$ matrix $Q^{(n)}$ whose elements $Q_{a\alpha}^{(n)} = P(y_\alpha|x_a, S^{(n)})$ are set equal to the *a posteriori* probability of correspondence match between the data-graph node a and the model-graph node α at iteration n of the EM algorithm.

The expression for the expected log-likelihood function $\Lambda(S^{(n+1)}|S^{(n)})$ simplifies if we note that the first term under the curly braces is simply proportional to the normalised probability mass over the state-space of the matching process. In other words, it contributes a constant amount

$$\sum_{a\in V_D}\sum_{\alpha\in V_M} Q_{a\alpha}^{(n)}\ln K_a = \sum_{a\in V_D}\ln K_a \tag{5.22}$$

Based on this observation, the critical quantity in determining the update direction for maximum likelihood matches is

$$\hat{\Lambda}(S^{(n+1)}|S^{(n)}) = \sum_{a\in V_D}\sum_{b\in V_D}\sum_{\alpha\in V_M}\sum_{\beta\in V_M} Q_{a\alpha}^{(n)}D_{ab}M_{\alpha\beta}s_{b\beta}^{(n+1)} \tag{5.23}$$

To write this component of the expected log-likelihood function in matrix notation, we group the indices as follows

$$\hat{\Lambda}(S^{(n+1)}|S^{(n)}) = \sum_{a\in V_D}\sum_{b\in V_D}\sum_{\alpha\in V_M}\sum_{\beta\in V_M} Q_{a\alpha}^{(n)}M_{\alpha\beta}(s_{\beta b}^{(n+1)})^T D_{ba}^T \tag{5.24}$$

Using the repeated index contraction for matrix products, this is equal to

$$\hat{\Lambda}(S^{(n+1)}|S^{(n)}) = Tr[Q^{(n)}M(S^{(n+1)})^T D^T] \tag{5.25}$$

where $Q^{(n)}$ is the matrix of correspondence probabilities. Finally, using the fact that the trace of a product of matrices is invariant under cyclic permutation of the matrix order

$$\hat{\Lambda}(S^{(n+1)}|S^{(n)}) = Tr[D^T Q^{(n)}M(S^{(n+1)})^T] \tag{5.26}$$

As a result we confine our attention to the quantity $Tr[D^TQ^{(n)}MS^T]$.
In Umeyama's (Umeyama, 1988) work the eigendecomposition method attempts to find the permutation matrix $\mathcal{P}$ which maximises $D^T\mathcal{P}M$. The utility measure used by the EM algorithm can be regarded as a weighted version of Umeyama's least squares criterion.

5.2.3 Maximisation

The maximisation step of the EM algorithm can be stated as that of recovering the set of correspondence indicators $S^{(n+1)}$ which satisfies the condition

$$S^{(n+1)} = \arg\max_{S} Tr[D^TQ^{(n)}MS^T] \tag{5.27}$$

In other words, the utility measure gauges the degree of correlation between the edge-sets of the two graphs under the weighted permutation structure induced by the correspondence probabilities.

To locate the updated set of correspondence indicators we use the extremum principal reported by Scott and Longuet-Higgins (Scott and Longuet-Higgins, 1991). Their result is as follows. Suppose that G is a positive definite $|V_D| \times |V_M|$ matrix. They have shown how the $|V_D| \times |V_M|$ orthogonal matrix R that maximises the quantity $Tr[GR^T]$ may be found by performing singular value decomposition. To do this they perform the matrix factorisation $G = V\Delta U^T$, where V is a $|V_D| \times |V_D|$ orthogonal matrix, U is a $|V_M| \times |V_M|$ orthogonal matrix and Δ is a $|V_D| \times |V_M|$ matrix whose off-diagonal elements $\Delta_{i,j} = 0$ if $i \neq j$ and whose "diagonal" elements $\Delta_{i,i}$ are non-zero. Suppose that E is the matrix obtained from Δ by making the diagonal elements $\Delta_{i,i}$ unity. The matrix R which maximises $Tr[GR^T]$ is $R = VEU^T$. This extremum principle may be applied to our graph matching problem if we make the substitution $G = D^TQ^{(n)}M$ and

perform the singular value decomposition $D^T Q^{(n)} M = V \Delta U^T$ to obtain R. This matrix satisfies the condition

$$R = \arg\max_{\hat{R}} Tr[D^T Q^{(n)} M \hat{R}^T] \tag{5.28}$$

Provided that the matrix $D^T Q^{(n)} M$ is positive-definite, then the elements of R are real.

Although this extremum principle is useful, it is not entirely suited to our needs. The reasons for this are that the elements of R can not be interpreted as probabilities since they are neither guaranteed to be positive, nor are they normalised. Furthermore, they can not be interpreted as assignment indicators since they are not binary in nature. To overcome these difficulties, we follow Scott and Longuet-Higgins by testing the elements of R to obtain a matrix of binary correspondence indicators $S^{(n+1)}$. If the element $R_{a,\alpha}$ is the maximum value for both the row and column that contains it, then the assignment indicator $s_{a,\alpha}^{(n+1)}$ is set to unity. Otherwise it is set to zero. As a result the updated set of correspondence indicators is

$$s_{a\alpha}^{(n+1)} = \begin{cases} 1 & \text{if } R_{a\alpha} = \arg\max_{b\beta} R_{b\beta} \\ 0 & \text{otherwise} \end{cases} \tag{5.29}$$

There are alternatives to this decision step. For instance, we could use the Sinkhorn normalisation idea of Gold and Rangarajan (Gold and Rangarajan, 1996) to preprocess the matrix, or we could apply a bipartite graph matching algorithm to the matrix R. The choice above is dictated by reasons of simplicity.

5.2.4 Expectation

In the expectation step of the EM algorithm, the *a posteriori* probabilities of the hidden data are computed from the component densities appearing in the

mixture-distribution. This is done by applying the Bayes theorem. At iteration $n+1$ we have

$$P(y_\alpha|x_a, S^{(n+1)}) = \frac{p(x_a|y_\alpha, S^{(n)})\pi_\alpha^{(n)}}{\sum_{\alpha' \in V_M} p(x_a|y_{\alpha'}, S^{(n)})\pi_{\alpha'}^{(n)}} \tag{5.30}$$

where

$$\pi_\alpha^{(n)} = \frac{1}{|V_D|} \sum_{a \in V_D} P(y_\alpha|x_a, S^{(n)}) \tag{5.31}$$

We can re-express the *a posteriori* probabilities using the indicator variables in the following manner

$$Q_{a\alpha}^{(n+1)} = \frac{K_a \exp[\mu \sum_{b \in V_D} \sum_{\beta \in V_M} D_{ab} M_{\alpha\beta} s_{b\beta}^{(n)}] \pi_\alpha^{(n)}}{\sum_{\alpha' \in V_M} K_a \exp[\mu \sum_{b \in V_D} \sum_{\beta \in V_M} D_{ab} M_{\alpha\beta} s_{b\beta}^{(n)}] \pi_{\alpha'}^{(n)}} \tag{5.32}$$

Since the constant K_a depends only on the index of the data graph node x_a it cancels between the numerator and the denominator.

At this point it is worth pointing out that there are alternative views of the E-step of the EM algorithm. For instance, in the graphical models literature, it is common to introduce a set of indicator variables to model the affinities between observed and missing data. These variables may be updated in an optimisation-step which uses mean-field annealing.

5.3 Soft Assign

Before we proceed to experiment with the new graph matching process, it is interesting to briefly review the standard quadratic formulation of the matching problem investigated by Simic (Simic, 1991), Sugarnathan *et al* (Suganthan et al., 1995) and Gold and Rangarajan (Gold and Rangarajan, 1996). Here the aim has been to deploy continuous optimisation methods such as the relatively heuristic graduated assignment (Gold and Rangarajan, 1996; Blake, 1987) or the more principled mean-field theory (Hoffmann and Buhmann, 1997; Yuille and

Kosowsky, 1994; Rangarajan et al., 1996; Yuille et al., 1994; Peterson and Soderberg, 1989) to update a set of assignment variables representing the matching process. Although there are many variants of the idea, the common feature of these algorithms is to commence from the quadratic cost function

$$\mathcal{E}_H = -\frac{1}{2}\sum_{a\in V_D}\sum_{\alpha\in V_M}\sum_{b\in V_D}\sum_{\beta\in V_M} D_{ab}M_{\alpha\beta}s_{a\alpha}s_{b\beta} \tag{5.33}$$

Performance has been demonstrated to be enhanced if additional, sometimes non-quadratic, terms are added. Examples include node self amplification term that encourages binary solutions (Gold and Rangarajan, 1996) and the addition of a logarithmic barrier entropy that convexifies the energy (Yuille and Kosowsky, 1994). To see the relationship between the graph-matching energy and the expected log-likelihood function we note that

$$\mathcal{E}_H = -\frac{1}{2}Tr[D^T S M S^T] \tag{5.34}$$

In other words, the weighting matrix Q is replaced by the assignment matrix S.

One of the simplest ways of updating the assignment variables is to use the soft-max ansatz of Bridle (Bridle, 1990). This ensures that the assignment variables remain constrained to lie within the range $[0, 1]$ by adopting the update rule

$$s_{a\alpha} \leftarrow \frac{\exp\Big[-\frac{1}{T}\frac{\partial \mathcal{E}_H}{\partial s_{a\alpha}}\Big]}{\sum_{\alpha'\in V_M}\exp\Big[-\frac{1}{T}\frac{\partial \mathcal{E}_H}{\partial s_{a\alpha'}}\Big]} \tag{5.35}$$

The temperature T is usually controlled using a slow exponential annealing schedule of the form suggested in (Gold and Rangarajan, 1996). For the quadratic graph-matching energy,

$$\frac{\partial \mathcal{E}_H}{\partial s_{b\beta}} = -\frac{1}{2}\sum_{a\in V_D}\sum_{\alpha\in V_M} D_{ab}M_{\alpha\beta}s_{a\alpha} \tag{5.36}$$

More recently Finch, Wilson and Hancock (Finch et al., 1997) have developed a more sophisticated soft-assign graph-matching algorithm which revolves around optimising the non-quadratic energy

$$\mathcal{E}_F = \sum_{a \in V_D} U_a \tag{5.37}$$

where

$$U_a = \frac{\sum_{\alpha \in V_M} H_{a\alpha} \exp[-\mu H_{a_\alpha}]}{\sum_{\alpha \in V_M} \exp[-\mu H_{a_\alpha}]} \tag{5.38}$$

and

$$H_{a\alpha} = \sum_{b \in V_D} \sum_{\beta \in V_M} D_{ab} M_{\alpha\beta} (1 - s_{b\beta}) \tag{5.39}$$

For this energy function the partial derivative is given by

$$\frac{\partial \mathcal{E}_F}{\partial s_{b\beta}} = - \sum_{a \in V_D} \sum_{\alpha \in V_M} D_{ab} M_{\alpha\beta} \Big[1 - \mu \Big(H_{a\alpha} - U_a \Big) \Big] Q_{a\alpha} \tag{5.40}$$

We will use these two soft-assign graph matching methods for the purposes of experimental comparison. However, it is important to stress that the soft-assign update process adopted here is very simplistic and leaves considerable scope for further refinement. For instance, in (Gold and Rangarajan, 1996) Sinkhorn matrices have been exploited to impose a permutation structure on the final solution.

5.4 Experiments

In this section of the chapter, we provide some experimental evaluation of the new graph-matching technique. There are two aspects to this study. We commence with a sensitivity study using synthetic data. The aim here is to evaluate how the new method performs under controlled structural corruption and to compare it with some alternatives reported elsewhere in the literature. The second part of the study evaluates the method on real-world data.

5.4.1 Sensitivity Study

Our sensitivity study is divided into two parts. First, we compare our method with some alternative methods for inexact graph-matching. These methods are capable of accommodating graphs of different size, but are not based on matrix factorisation. Here we investigate the effect of adding additional nodes to the graphs. The second class of methods used for comparison are those which rely on matrix factorisation techniques. These methods do not work when the graphs are of different size. Here we keep the graphs of fixed equal size and investigate the effect of corrupting the pattern of edges.

Inexact Graph Matching

We commence by studying the effect of controlled structural error on the graphs being matched. The graphs used in our study are the Delaunay triangulations of randomly generated point-sets. The effects of structural error are simulated by deleting a predefined fraction of randomly selected nodes and re-triangulating the remaining points.

We compare the performance of our new matching method with three alternatives. These are the dictionary-based relaxation scheme of Wilson and Hancock (Wilson and Hancock, 1997), the quadratic assignment method of Gold and Rangarajan (Gold and Rangarajan, 1996) and the non-quadratic graduated assignment method of Finch, Wilson and Hancock (Finch et al., 1998). Figure 5.1 compares the four algorithms. Here we show the fraction of correct correspondences as a function of the fraction of nodes deleted from the graphs. The main feature to note is that the new graph matching method delivers performance that is intermediate between the discrete relaxation method and the non-linear graduated assignment method. This is an interesting observation when we compare

the computational overheads associated with the three methods.

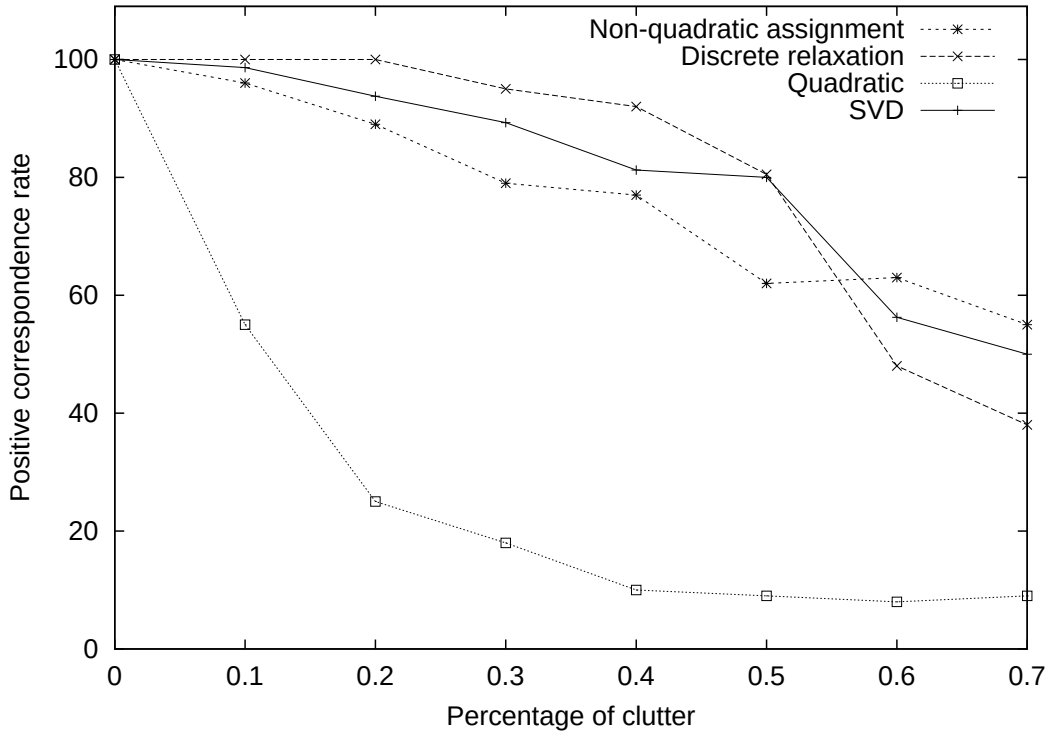


Figure 5.1: Sensitivity study for graphs of different size.

To provide some indication of the iterative properties of our new algorithm, Figure 5.2 shows the fraction of correct correspondences as a function of iteration number. The method takes approximately 15 iterations to converge. Moreover, there is significant improvement in each of the iterations. At this point it is worth pointing out the relative complexities of the different methods investigated here. The SVD can be computed in order $|V|^3$ per iteration, where $|V|$ is the number of nodes in the graph. The Wilson and Hancock method, on the other hand requires $|V|^2C^3$ computations per iteration for exact graph-matching and $4^C C|V|^2$ computations per iteration for inexact graph-matching, where C is the average degree of nodes of the graph. The quadratic assignment method has complexity of order $|V|^3$ per iteration. The Wilson and Hancock method also converges in about 10-15 iterations, while the quadratic assignment method takes

100's of iterations. In other words, our new SVD-based method is both accurate and efficient.

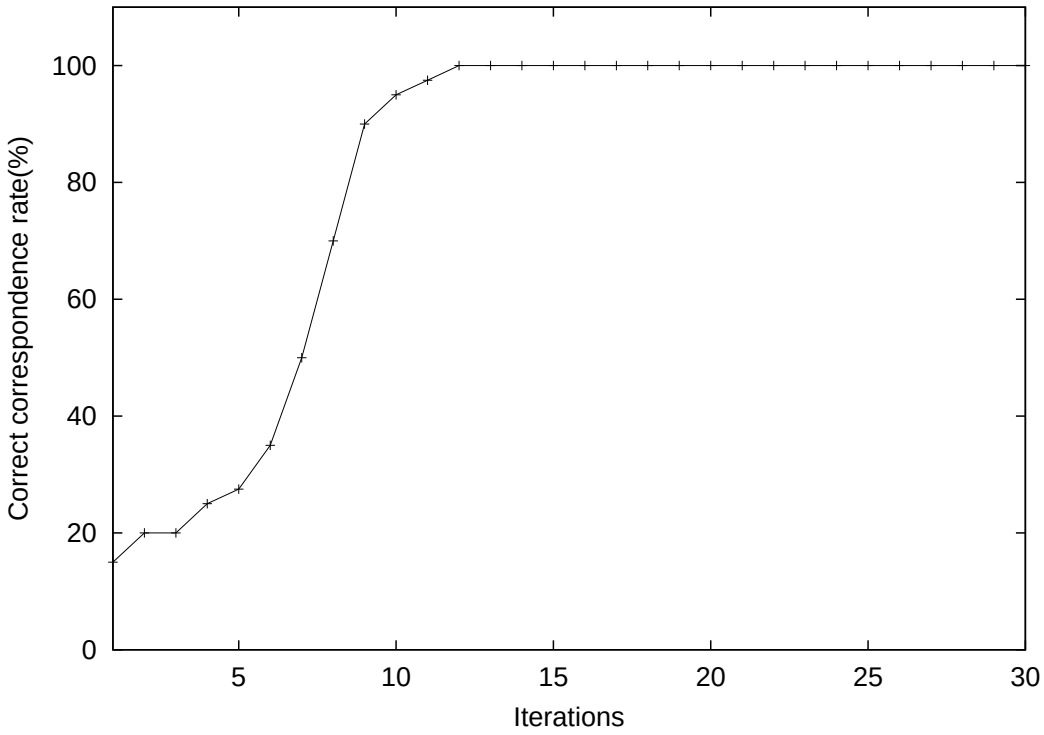


Figure 5.2: Matching convergence rate.

Factorisation Methods

In this subsection we provide comparison with two methods for weighted graph-matching which share with our own method the feature of relying on matrix factorisation. The methods selected for this comparison are

- Umeyama's weighted graph-matching method which seeks the permutation matrix $\mathcal{P}$ that minimises quantity $J(\mathcal{P}) = ||\mathcal{P}M - D||$ (Umeyama, 1988). The method performs the singular value decompositions $M = U_M \Delta_M U_M^T$ and $D = U_D \Delta_D U_D^T$, where the U's are orthogonal matrices and the Δ's are diagonal matrices. Once these factorisations have been performed, the required permutation matrix is $\mathcal{P} = U_D U_M^T$.

- Shapiro and Brady's (Shapiro and Brady, 1992) weighted graph-matching method which uses the modal structure of the two weighted adjacency matrices D and M. The modal structure of the two adjacency graphs is obtained by solving the eigenvalue equation $D\phi_l^D = \lambda_l \phi_l^D$ where λ_l is the l^{th} eigenvalue of the adjacency matrix D and ϕ_l^D is the corresponding eigenvector. The eigenvectors are ordered according to the size of the associated eigenvalues and are used as the columns of the modal matrix $\Phi_D = \left(\phi_1^D, \phi_2^D, \phi_3^D, ...\right)$. This procedure is repeated to construct a second modal matrix Φ_M for the model-graph adjacency matrix M. The column index of these two modal matrices refers to the order of the eigenvalues while the row-index is the index of the nodes in the graphs. Shapiro and Brady find correspondences by locating pairs of rows which have minimum distance, i.e.

$$s_{a,\alpha} = \begin{cases} 1 & \text{if } \alpha = \arg\min_{\alpha'} \sum_{l=1}^{N} ||\Phi_D(a,l) - \Phi_M(\alpha',l)||^2 \\ 0 & \text{otherwise} \end{cases} \tag{5.41}$$

These two methods rely on weighted adjacency matrices rather than the binary ones defined earlier. To conduct our experiments, we have generated random 2D point-sets. We use the positions of these points to generate the weights of the adjacency matrix. Suppose that $\vec{x}_\alpha^M$ and $\vec{x}_\beta^M$ represent the co-ordinate vectors associated with the nodes indexed α and β. The weight associated with the edge connecting the nodes is

$$M_{\alpha\beta} = \exp\left[-k||\vec{x}_\alpha^M - \vec{x}_\beta^M||^2\right] \tag{5.42}$$

These two methods are not effective when the graphs under study contain different numbers of nodes. To compare with our method we have therefore kept the number of points fixed and have added Gaussian errors to the point positions. The parameter of the noise process is the standard deviation of the

positional jitter. In our experiments, we express this parameter as a fraction of the average minimum distance between points (the relative standard deviation). It is important to stress that the methods compared here use different representations of the arrangement of the points. The Shapiro and Brady, and Umeyama methods use the weighted adjacency matrix. Our method, on the other hand, uses a binary adjacency matrix to represent the Delaunay triangulation of the points.

In Figure 5.3 we show the fraction of correct correspondences as a function of the relative standard deviation for our new method (bold solid curve), Umeyama's (Umeyama, 1988) method (faint solid curve) and the method of Shapiro and Brady (Shapiro and Brady, 1992) (dash-dotted curve). The main feature to note is that our method outperforms the two alternatives. There is little to distinguish the performance of the Shapiro and Brady (Shapiro and Brady, 1992), and Umeyama (Umeyama, 1988) methods. Both fail abruptly once the relative standard deviation exceeds 0.2, i.e. the noise standard deviation is greater than 20% of the average closest point distance. Our method, on the other hand, degrades almost linearly with the noise standard deviation. However, it must be stressed that the results are not completely comparable. In the case of Shapiro and Brady, and Umeyama (Umeyama, 1988), we are measuring the sensitivity of the method to noise on the entries of the weighted adjacency matrices. In the case of our method, we are measuring the sensitivity of the method to errors in the edge-sets of the graphs used for matching.

To show that the point-jitter does indeed result in significantly different adjacency matrices, in Figure 5.4 we show the fraction of edge differences as a function of the relative standard deviation of the positional jitter. The fraction of edge errors in the unweighted adjacency graphs is defined to be

$$F = \frac{\sum_{a=1}^{N}\sum_{b=1}^{N}|D_{ab} - \sum_{c=1}^{N} s_{ac}M_{cb}|}{\sum_{a=1}^{N}\sum_{b=1}^{N}|M_{ab}|} \tag{5.43}$$

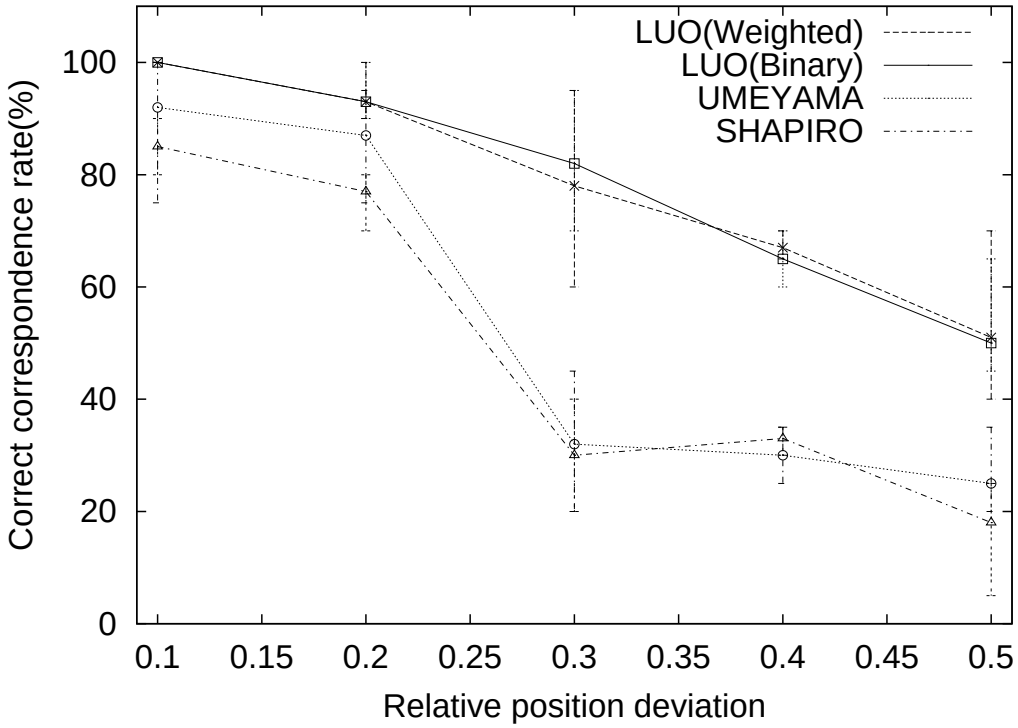


Figure 5.3: Comparison of the four eigendecomposition methods for graphs with the same number of nodes.

where N is the number of nodes being matched. From Figure 5.4 it is clear, that the fraction of edge errors is 50% when the relative standard deviation of the positional jitter is 0.4. In other words, our method is finding 70% of the correct correspondences, even when 50% of the entries in the data-graph adjacency matrix are in error.

Finally, we illustrate the results obtained when we apply our method to the weighted adjacency matrix rather than the binary adjacency matrix. The dot-dashed curve in Figure 5.3 shows the fraction of correct correspondences as a function of the relative standard deviation of the point-position jitter. The method performs considerably better than the Shapiro and Brady, and Umeyama methods. However, there is little to distinguish its performance from that obtained with the binary adjacency matrix.

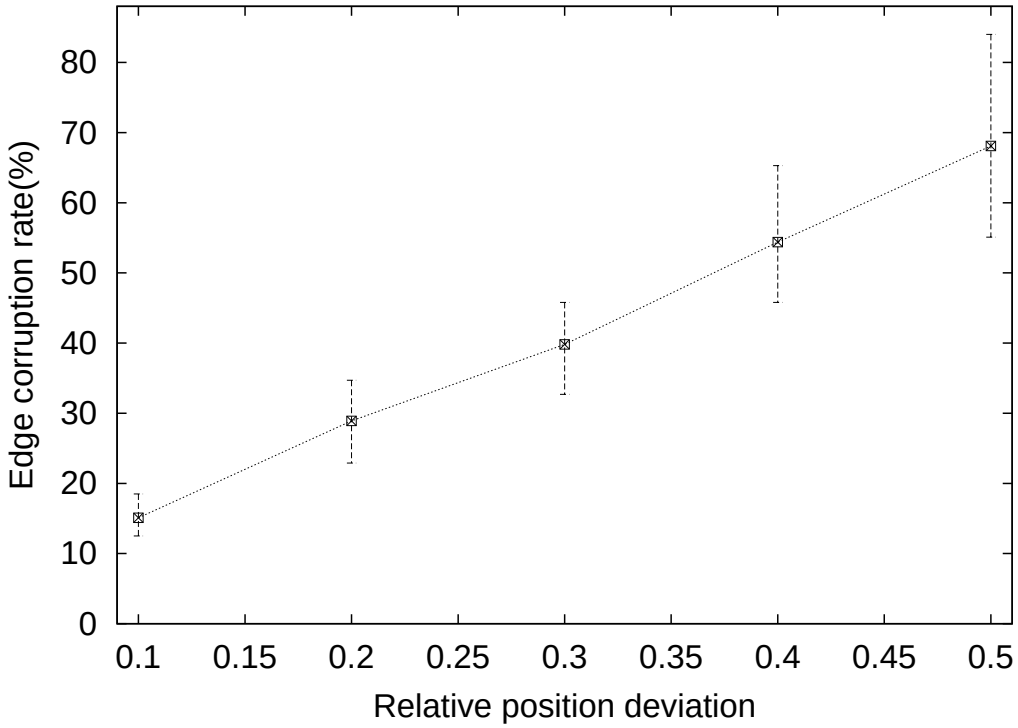


Figure 5.4: Fraction of edge errors as a function of the relative standard deviation of the point-position error.

5.4.2 Real-world data

We commence our real-world evaluation of the graph-matching method on images of indoor scenes. Here we are concerned with matching the Delaunay triangulations of corner-features. We use the corner detector recently reported by Luo, Cross and Hancock(Luo et al., 1999) to extract point features. Figure 5.5 shows two examples of the indoor images used in our study. Superimposed on the images are the detected corners and their associated Delaunay triangulations. The two images are taken from different viewpoints. There is rotation, scaling and perspective distortion present. Moreover, several of the objects in the scene are at different depths and move relative to one-another. As a result there are significant structural differences in the two Delaunay graphs. Figure 5.6 shows

the correspondences between the corners as lines between the two images. After checking by hand, the fraction of correct correspondences is 77%.

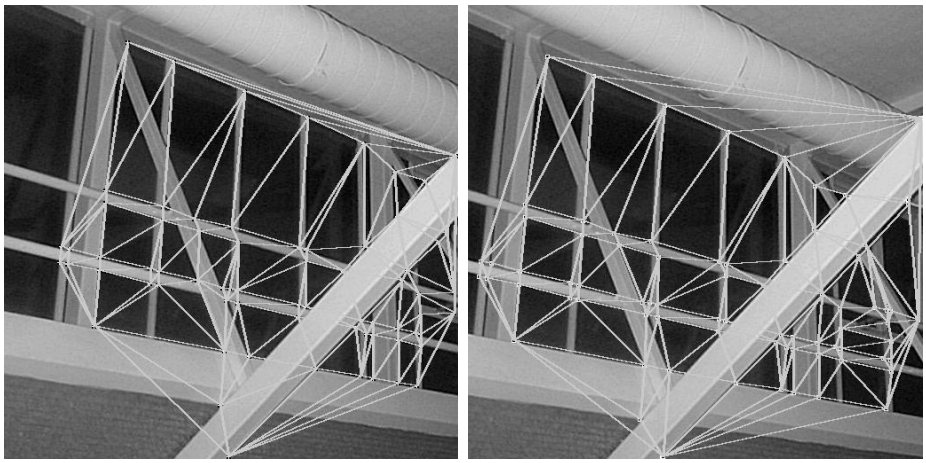


Figure 5.5: Ceiling images overlayed with Delaunay graphs.

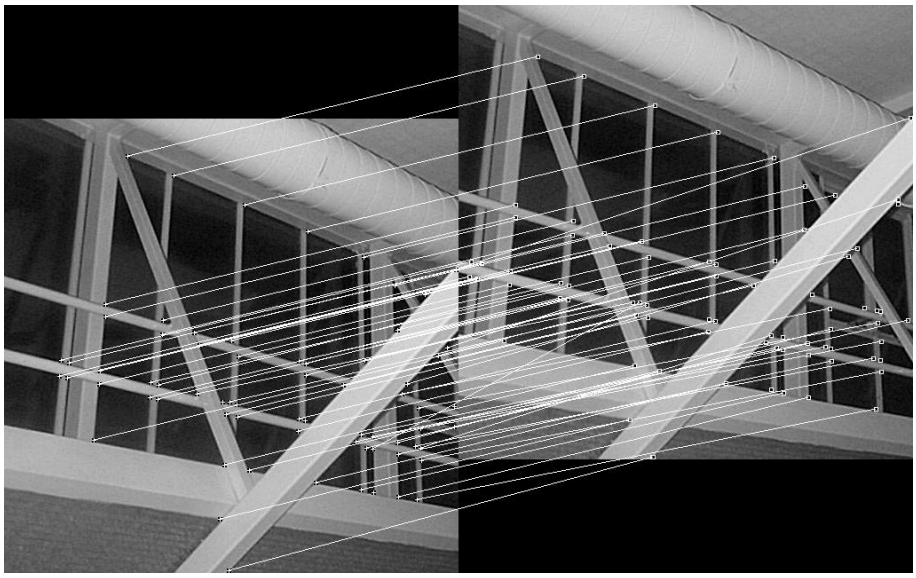


Figure 5.6: Correspondences of the ceiling images.

We repeat this set of experiments using images taken from the CMU/VASC model-house sequence. The images used in our study are shown in Figure 5.7 and correspond to different camera viewing directions. The detected corner features

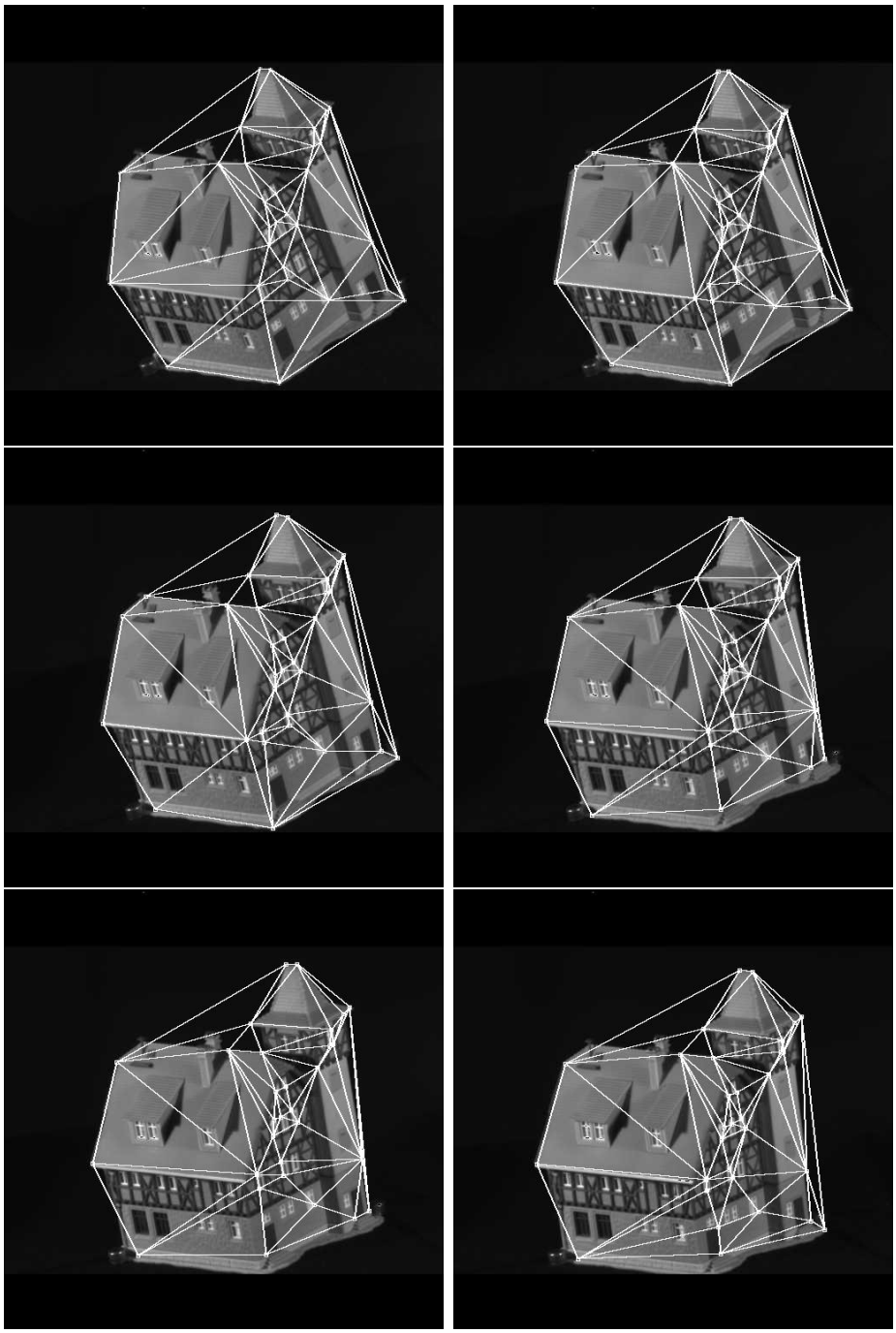


Figure 5.7: Delaunay graphs overlayed on the toy house images.

and their Delaunay triangulations are overlayed on the images. There are clearly significant structural differences in the graphs. Figures 5.8 to 5.12 show the results obtained when we match the first image to each of the subsequent images in the

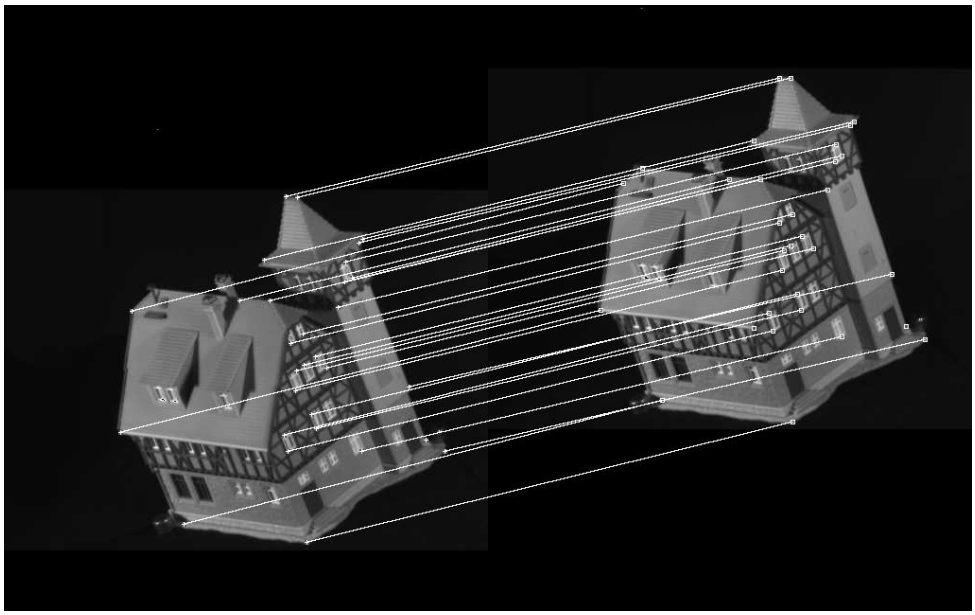


Figure 5.8: Correspondences between the first and the second images.

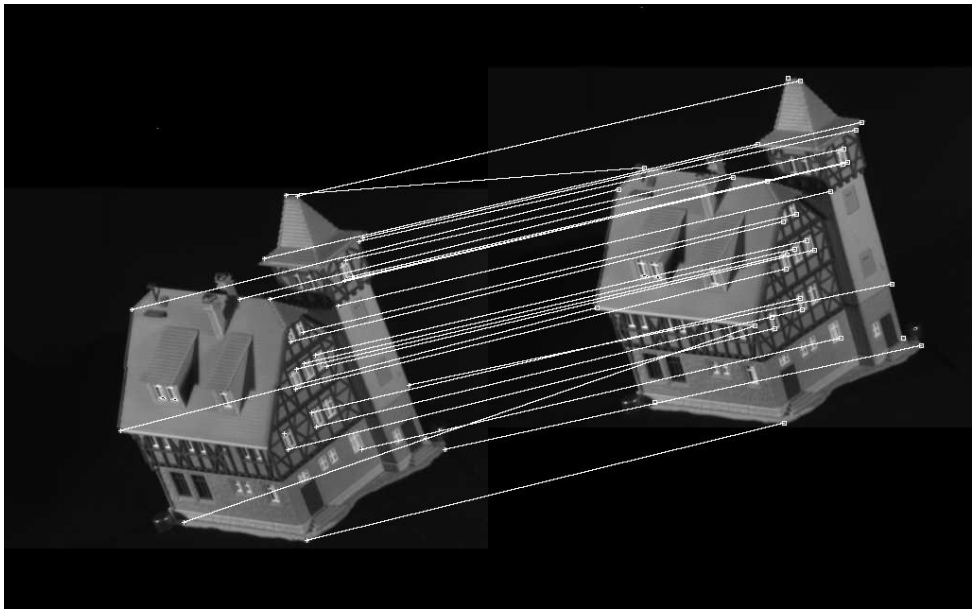


Figure 5.9: Correspondences between the first and the third images.

sequence. The results are summarised in Table 5.1. Here we list the number of corners that are in correct correspondence, the number of corners that are in error, and the number of corners for which there are no correspondences (i.e. there is no row and column maximum). The method breaks down after the 4th

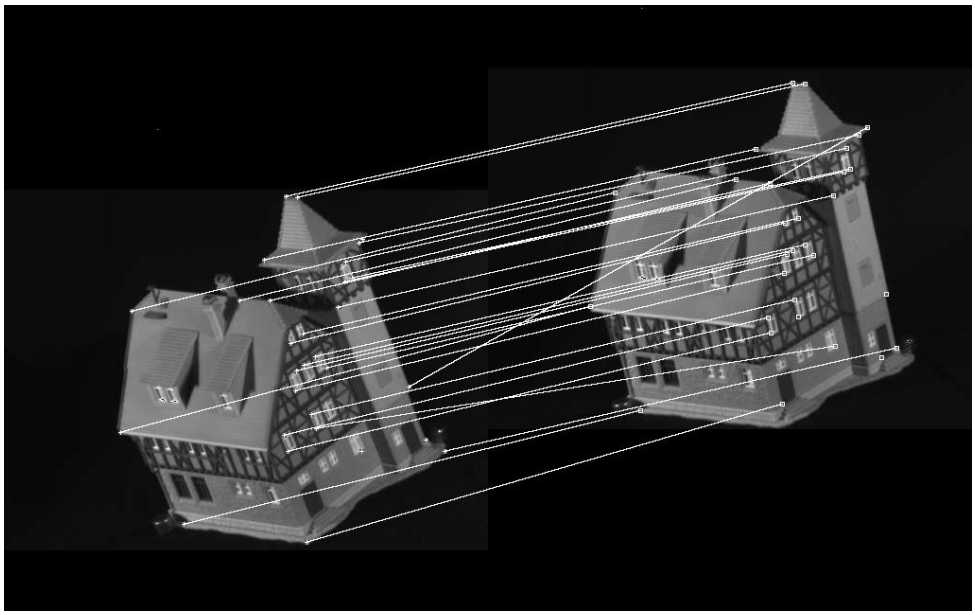


Figure 5.10: Correspondences between the first and the fourth images.

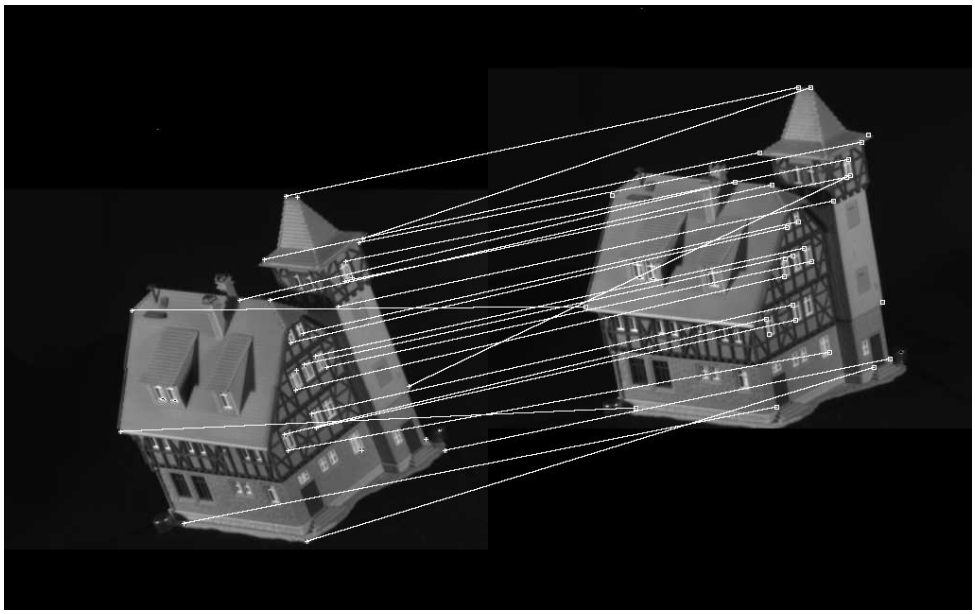


Figure 5.11: Correspondences between the first and the fifth images.

image in the sequence.

To provide some comparison, we have selected a pair of images which contain the same number of corner points (image 1 and image 5). Although the number of corners is the same, there are differences in the both identities of the

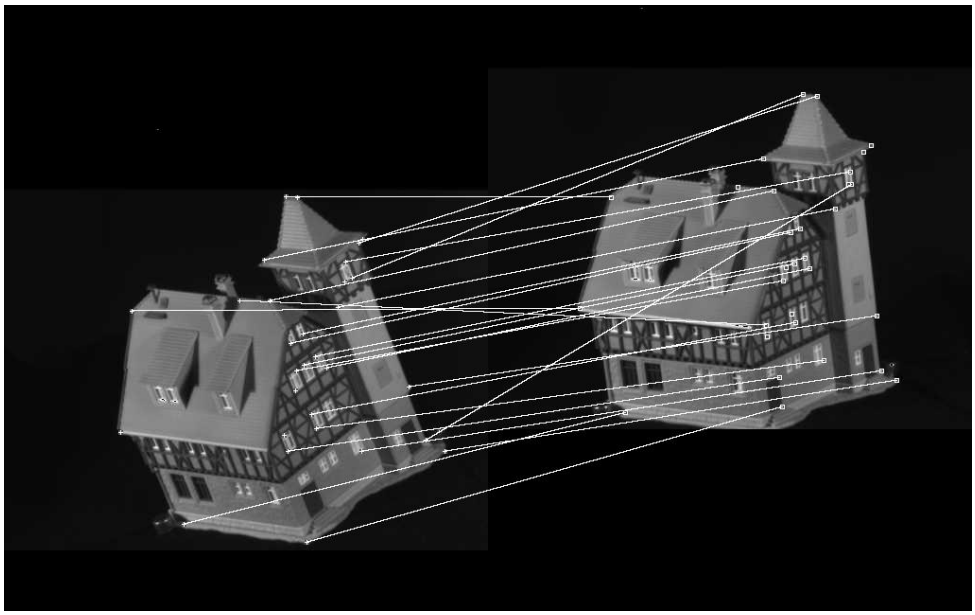


Figure 5.12: Correspondences between the first and the sixth images.

House index	0	1	2	3	4	5	6	7	8	9
Correct	-	29	26	24	17	13	11	5	3	0
False	-	0	2	3	8	11	12	15	19	24
Missed	-	1	2	3	5	6	7	10	8	6

Table 5.1: Correspondence allocation results of the GM method.

detected points and their structural arrangement. For these images we compare the matches returned by the unweighted and weighted versions of our algorithm (referred to as Luo), the method of Umeyama and the method of Shapiro and Brady. The results are shown in Figures 5.13 to 5.16 and the numbers of correct matches are summarised in Table 5.2. From these results it is clear that the new method returns considerably better matches.

Methods	Luo(Weighted)	Luo(Unweighted)	Umeyama	Shapiro
Correct	29	29	10	11
False	0	0	7	7
Missed	1	1	13	12

Table 5.2: Summary of the comparison of the matching algorithms.

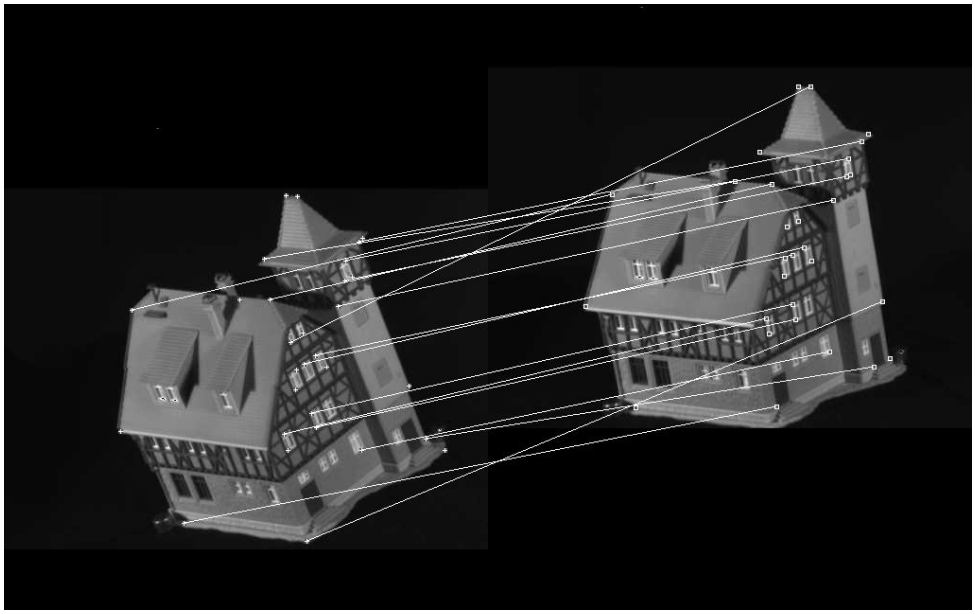


Figure 5.13: Correspondences from the Umeyama algorithm.

5.5 Conclusions

Our main contributions in this chapter are twofold. First, we have cast the problem of graph-matching into a maximum likelihood framework by constructing a mixture model over the set of hidden correspondences and adopting a Bernoulli model for the distribution of edge-matching errors. Second, we have used the apparatus of the EM algorithm to show how the problem of estimating the correspondence indicators may be cast into a compact matrix setting. This allows us to use singular value decomposition to estimate the correspondence indicators in the M-step. The result is an efficient algorithm that can be used to accurately

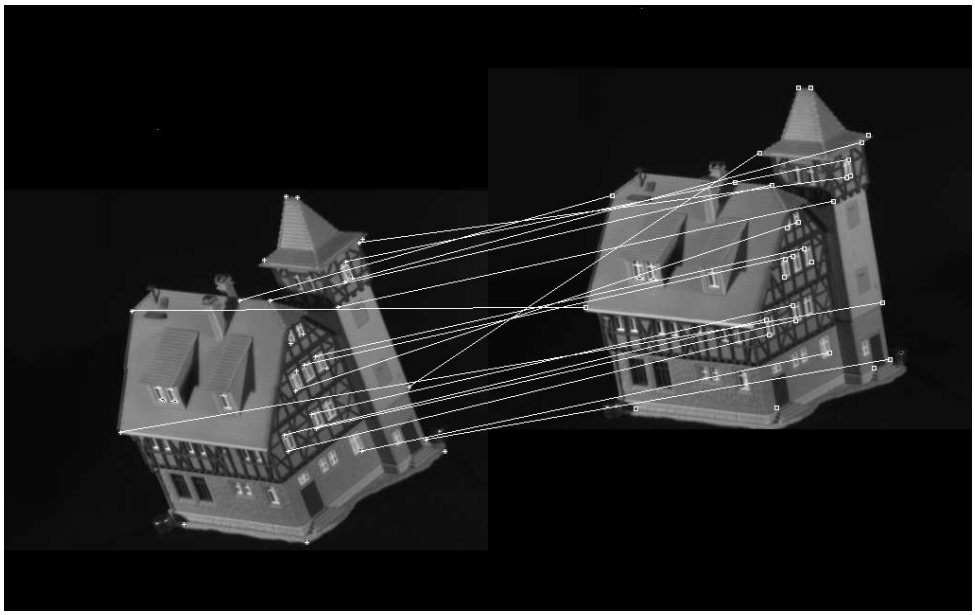


Figure 5.14: Correspondences from the Shapiro algorithm.

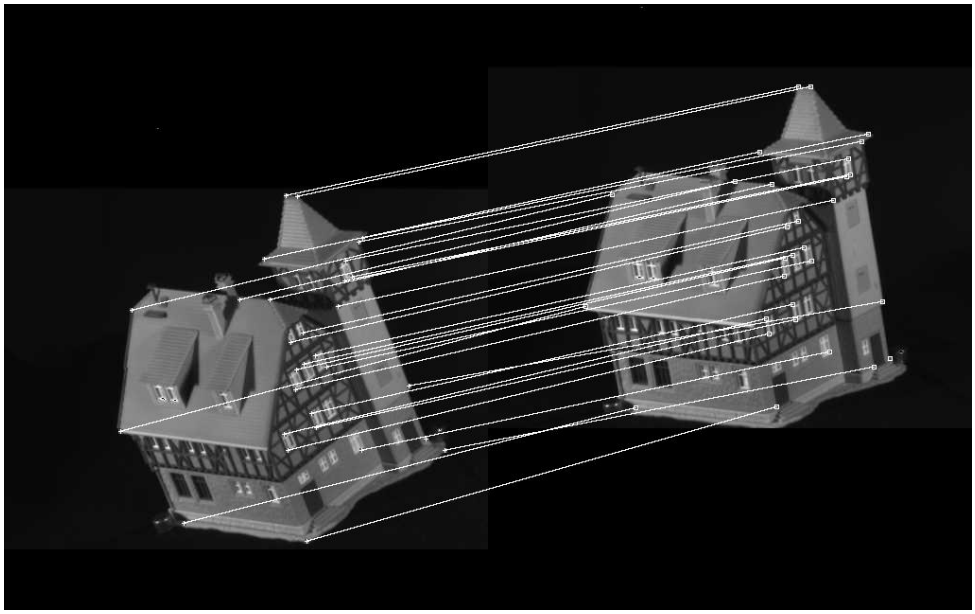


Figure 5.15: Correspondences from the unweighted variant of our algorithm.

match inexact graphs under considerable levels of structural corruption.

When viewed from the perspective of recent work on matrix-based graph-matching, the important contribution of this chapter is to show how point-sets of different sizes can be matched using singular value decomposition.

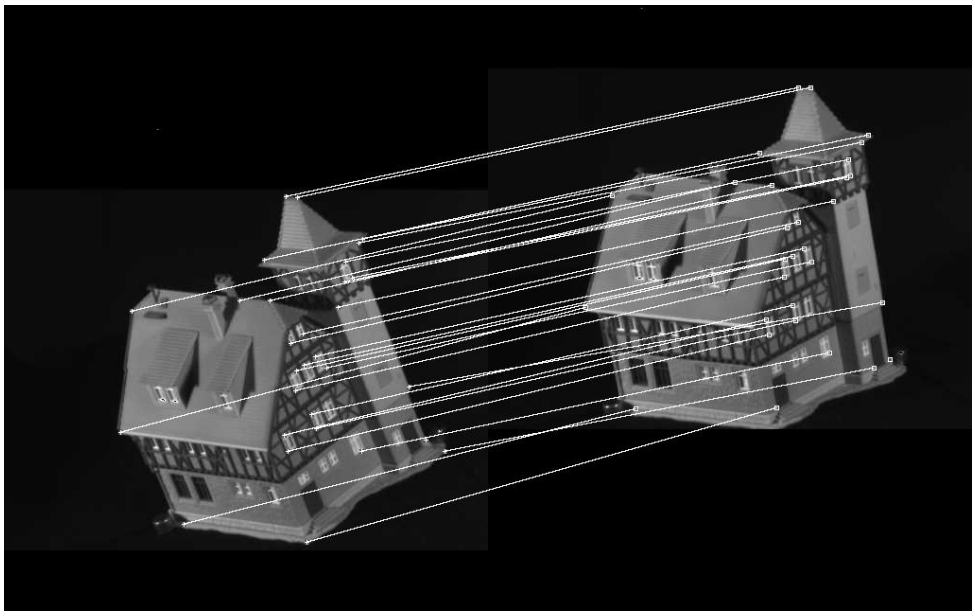


Figure 5.16: Correspondences from the weighted variant of our algorithm.

Chapter 6

Unification of alignment and correspondence

6.1 Introduction

In previous chapters of this thesis, we presented an iterative Procrustes alignment method and a graph matching method using the EM algorithm. In this chapter, we aim to develop a unified framework for point set alignment and correspondence assignment. The motivation for the work reported in this chapter is that the dichotomy normally drawn between the two processes overlooks considerable scope for synergistic interchange of information. In other words, there must always be bounds on alignment before correspondence analysis can be attempted, and vice versa. Our approach in developing the new point-pattern matching method was to embed constraints on the spatial arrangement of correspondences within an EM algorithm for alignment parameter recovery. This process has many features reminiscent of Jordan and Jacob's hierarchical mixture of experts algorithm (Jordan and Jacobs, 1994). The observation underpinning this chapter is that although the method proved effective it fails to put the alignment and correspon-

dence processes on a symmetric footing. Relational constraints were simply used to gate the contributions to the log-likelihood function for the alignment errors.

The idea underpinning this chapter is to provide a new framework for the maximum likelihood matching of point-sets which allows a symmetric linkage between alignment and correspondence. Specifically, we aim to realise interleaved iterative steps which communicate via an integrated utility measure. The utility measure is the cross-entropy between the probability distributions for alignment and correspondence.

We consider two matching applications. The first of these involves Procrustes alignment. By casting the cross-entropy in terms of matrices, we realise optimisation via dual singular value decompositions. The first of these transforms the point set positions so as to locate an alignment that maximises the weighted correlation between the point-set co-ordinates. The second singular value decomposition updates the set of correspondence probabilities that maximise the weighted correlation between the edge-sets of the adjacency graphs for the point-sets. These processes are interleaved and iterated to convergence.

In the second application we apply the new matching framework to the non-rigid alignment of point-sets which deform according to a point distribution model. This is a class of deformable shape model (Cohen and Cohen, 1993; Kass et al., 1988; Pentland and Sclaroff, 1991) developed by Cootes and Taylor (Cootes et al., 1995). The idea underpinning point distribution models is to learn the modes of variation of point-patterns by computing the eigenmodes of the co-variance matrix for a set of training examples. The eigenvectors of the co-variance matrix define directions in which the points can move with respect to the mean-pattern. Once trained in this way, a point distribution model can be fitted to data by estimating the proportions of each eigen-mode that minimise the distance between the data and the aligned model. When carefully trained, the

method can be used to model quite complex point deformations. Moreover, the modal deformations defined by the eigenvectors can be reconciled with plausible natural modes of shape variation.

While much effort has been expended in improving the training of PDM's, there are a number of shortcomings that have limited their effective matching to noisy point-data. Firstly, in order to estimate the modal displacements necessary to align a point-distribution model with data, the data points must be labelled. In other words, the correspondences between points in the model and the data must be known *a priori*. Secondly, the point-sets must be of the same size. As a result, point distribution models can not be aligned reliably to point-sets which are subject to contamination or dropout. Finally, information concerning the relational arrangement of the points is overlooked. Our new matching framework provides a natural way of overcoming these shortcomings.

Our aim in this chapter is to present a statistical framework for aligning point distribution models. We depart from the conventional treatment in three ways. First, we deal with unlabelled point-sets. That is to say we commence without knowledge of the correspondences between the model and the data to be matched. Our second contribution is to deal with the case in which the model and the data contain different numbers of points. This might be due to poor feature location or contamination by noise. Finally, we aim to exploit constraints on the relational arrangement of the points to improve the fitting process.

6.2 Dual Step Matching Algorithm

We characterise the matching problem in terms of separate probability distributions for alignment and correspondence. In the case of alignment, the distribution models the registration errors between the data point positions and their coun-

terparts in the model. The correspondence process on the other hand captures the consistency of the pattern of matching assignments to the graph representing the point-sets. The set of assignments is represented by the function $f : \mathcal{D} \rightarrow \mathcal{M}$. Suppose that $P_{i,j}^{(n)}$ is the probability that node i from the data graph is in alignment with node j from the model graph at iteration n. Similarly, $Q_{i,j}^{(n)}$ is the probability that node i is in correspondence with node j. Further suppose that $p_{i,j}^{(n)} = p(u_i|v_j^{(n)})$ is the probability distribution for the alignment error between the nodes i and j at iteration n. The distribution of the correspondence errors associated with the assignment function $f^{(n)}$ at iteration n is $q_{i,j}^{(n)}$. With these ingredients the utility measure which we aim to maximise in the dual alignment and correspondence steps is

$$\mathcal{E} = \sum_{i \in \mathcal{D}} \sum_{j \in \mathcal{M}} \left[Q_{i,j}^{(n)} \ln p_{i,j}^{(n+1)} + P_{i,j}^{(n)} \ln q_{i,j}^{(n+1)} \right] \tag{6.1}$$

In other words, the two processes interact via a symmetric expected log-likelihood function. The correspondence probabilities weight contributions to the expected log-likelihood function for the alignment errors, and vice-versa. In Cross and Hancock's work, they showed how the first term arises through the gating of the log-likelihood function of the EM algorithm (Cross and Hancock, 1998).

The aligned point positions and correspondence matches are recovered via the dual maximisation equations

$$M^{(n+1)} = \arg\max_{M} \sum_{i \in \mathcal{D}} \sum_{j \in \mathcal{M}} Q_{i,j}^{(n)} \ln p_{i,j}^{(n+1)} \tag{6.2}$$

and

$$f^{(n+1)} = \arg\max_{f} \sum_{i \in \mathcal{D}} \sum_{j \in \mathcal{M}} P_{i,j}^{(n)} \ln q_{i,j}^{(n+1)} \tag{6.3}$$

6.2.1 Correspondences

The correspondences are recovered via maximisation of the quantity

$$\mathcal{E}_c = \sum_{i \in \mathcal{D}} \sum_{j \in \mathcal{M}} P_{i,j}^{(n)} \ln q_{i,j}^{(n+1)} \tag{6.4}$$

In Chapter 5, we have shown that the probability distribution for the assignment variables is

$$q_{i,j}^{(n)} = K_a \exp\Big[\mu \sum_{i' \in \mathcal{D}} \sum_{j' \in \mathcal{M}} E_D(i,i') E_M(j,j') s_{i'j'}\Big] \tag{6.5}$$

where K_a and μ are constants. With this distribution to hand, the correspondence assignment step reduces to one of maximising the quantity

$$\mathcal{F}_c = \sum_{i \in \mathcal{D}} \sum_{i' \in \mathcal{D}} \sum_{j \in \mathcal{M}} \sum_{j' \in \mathcal{M}} P_{i,j}^{(n)} E_M(j,j') s_{j',i'}^{(n+1)T} E_D(i',i)^T \tag{6.6}$$

where $E_D(i,i')$ and $E_M(j,j')$ are the elements of the adjacency matrices for the data and model graphs. In more compact notation, the updated matrix of correspondence indicators $S^{(n+1)}$ satisfies the condition

$$S^{(n+1)} = \arg\max_{S} Tr[E_D^T P^{(n)} E_M S^{(n)T}] \tag{6.7}$$

where $P^{(n)}$ is a matrix whose elements are the alignment probability $P_{i,j}^{(n)}$. In other words, the utility measure gauges the degree of correlation between the edge-sets of the two graphs under the permutation structure induced by the alignment and correspondence probabilities. Using the method outlined in Chapter 5, we recover the matrix of assignment variables that maximises $\mathcal{F}_c$ by performing the singular value decomposition $E_D^T P^{(n)} E_M = V \Delta U^T$, where Δ is again a diagonal matrix and U and V are orthogonal matrices. The matrices U and V are used to compute a matrix $R^{(n+1)} = VEU^T$ which maximises $Tr(E_D^T P^{(n)} E_M S^T)$. The updated set of correspondence indicators is

$$s_{i,j}^{(n+1)} = \begin{cases} 1 & \text{if } R_{i,j} = \arg\max_{i'j'} R_{i',j'} \\ 0 & \text{otherwise.} \end{cases} \tag{6.8}$$

The *a posteriori* probabilities can be calculated by the expectation step. As presented in Chapter 5, it can be represented by

$$Q_{i,j}^{(n+1)} = \frac{q_{i,j}^{(n)} \pi_j^{(n)}}{\sum_{j' \in \mathcal{M}} q_{i,j'}^{(n)} \pi_{j'}^{(n)}} \tag{6.9}$$

where

$$\pi_j^{(n)} = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} Q_{i,j}^{(n)} \tag{6.10}$$

6.3 Rigid point set alignment

In this chapter, we tackle two kinds of point pattern matching problems. When the deformation is rigid, we use the Procrustes alignment technique to match the point sets. If there exists non-rigid deformation, the Procrustes alignment will fail to match the point sets. We will derive a new alignment method to match point sets with non-rigid deformations by appealing the point distribution models. For the sake of completeness, we summarise the Procrustes alignment method next while the details can be found in Chapter 4. The derivation of the new non-rigid alignment method is presented in the next section.

The alignment is achieved via maximisation of the quantity

$$\mathcal{E}_a = \sum_{i \in \mathcal{D}} \sum_{j \in \mathcal{M}} Q_{i,j}^{(n)} \ln p_{i,j}^{(n+1)} \tag{6.11}$$

By assuming the point sets alignment error to be Gaussian distributed, we have

$$p_{i,j}^{(n)} = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\Big[-\frac{1}{2}(\vec{u}_i - \vec{v}_j^{(n)})^T \Sigma^{-1} (\vec{u}_i - \vec{v}_j^{(n)})\Big] \tag{6.12}$$

where Σ is the covariance matrix for the alignment error. Here we assume that the position errors are isotropic, in other words the errors in the x and y directions are identical and uncorrelated. As a result we write $\Sigma = \sigma^2 I_2$ where I_2 is the 2x2 identity matrix and σ^2 is the isotropic noise variance for the point positions.

With this distribution to hand, our next step is to maximise the quantity

$$\mathcal{F}_a = \sum_{i \in \mathcal{D}} \sum_{j \in \mathcal{M}} Q_{i,j}^{(n)} (\vec{u}_i - \vec{v}_j^{(n+1)})^T (\vec{u}_i - \vec{v}_j^{(n+1)}) \tag{6.13}$$

where $Q_{i,j}^{(n)}$ is the probability of point i in correspondence with point j. Based on the description in Chapter 4, the matrix notation of the maximisation step is

$$M^{(n+1)} = \arg\max_{M} Tr[DQ^{(n)} M^{(n)T} \Phi] \tag{6.14}$$

.

Drawing on the EM Procrustes method described in Chapter 4, then if the singular value decomposition of $DQ^{(n)}M^{(n)T}$ is $U\Delta V^T$, the updated data point positions will be

$$\vec{u}_j^{(n+1)} = \mu_M + \sqrt{\frac{Tr\Sigma_D}{Tr\Sigma_M}} V U^T (\vec{u}_j^{(n)} - \mu_D^{(n)}), \tag{6.15}$$

where

$$\Sigma_D = \frac{\sum_{i \in \mathcal{D}} (\vec{u}_i^{(n)} - \mu_D^{(n)})(\vec{u}_i^{(n)} - \mu_D^{(n)})^T}{|\mathcal{D}|} \tag{6.16}$$

and

$$\Sigma_M = \frac{\sum_{j \in \mathcal{M}} (\vec{v}_j - \mu_M)(\vec{v}_j - \mu_M)^T}{|\mathcal{M}|} \tag{6.17}$$

The new *a posteriori* correspondence assignment probability can be calculated using the Bayes rule. The expectation of the assignment probability is

$$P_{i,j}^{(n+1)} = \frac{p_{i,j}^{(n)} \pi_j^{(n)}}{\sum_{j' \in \mathcal{M}} p_{i,j'}^{(n)} \pi_{j'}^{(n)}} \tag{6.18}$$

where

$$\pi_j^{(n)} = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} P_{i,j}^{(n)} \tag{6.19}$$

6.4 Non-rigid point sets alignment

In this section we consider how our unified framework for alignment and correspondence can be applied to non-rigid deformations. We focus on point distribution models.

6.4.1 Point Distribution Models

The point distribution model of Cootes and Taylor commences from a set training patterns. Each training pattern is a configuration of labelled point co-ordinates or landmarks. The patterns of landmark points are collected as the object in question undergoes representative changes in shape. To be more formal, each pattern of landmark points consists of M labelled points whose co-ordinates are represented by the set of position co-ordinates $\{\vec{x}_1, \vec{x}_2, ..., \vec{x}_M\} = \{(x_1, y_1), ...(x_M, y_M)\}$. Suppose that there are N patterns of landmark points. The t^{th} training pattern is represented using the long-vector of landmark co-ordinates $\vec{X}_t = (x_1, y_1, x_2, y_2, \cdots, x_M, y_M)^T$, where the subscripts of the co-ordinates are the landmark labels. For each training pattern the labelled landmarks are identically ordered. The mean landmark pattern is represented by the average long-vector of co-ordinates

$$\hat{X} = \frac{1}{N}\sum_{t=1}^{N}\vec{X}_t$$

The covariance matrix for the landmark positions is

$$T = \frac{1}{N}\sum_{t=1}^{N}(\vec{X}_t - \hat{X})(\vec{X}_t - \hat{X})^T \tag{6.20}$$

The eigenmodes of the landmark covariance matrix are used to construct the point-distribution model. First, the eigenvalues λ of the landmark covariance matrix are found by solving the eigenvalue equation

$$|T - \lambda I| = 0$$

where I is the $2M \times 2M$ identity matrix. The eigen-vector $\vec{\phi}^{\lambda_i}$ corresponding to the eigenvalue λ_i is found by solving the eigenvector equation $T\vec{\phi}^{\lambda_i} = \lambda_i\vec{\phi}^{\lambda_i}$. According to Cootes and Taylor, the landmark points are allowed to undergo displacements relative to the mean-shape in directions defined by the eigenvectors of the covariance matrix T. To compute the set of possible displacement

directions, the K most significant eigenvectors are ordered according to the magnitudes of their corresponding eigenvalues to form the matrix of column-vectors $\Phi = (\phi^{\lambda_1}|\phi^{\lambda_2}|...|\phi^{\lambda_K})$, where $\lambda_1, \lambda_2, ..., \lambda_K$ is the order of the magnitudes of the eigenvectors. The landmark points are allowed to move in a direction which is a linear combination of the eigenvectors. The updated landmark positions are given by

$$\vec{X}^{(n)} = \hat{X} + \Phi\vec{r}$$

where $\vec{r}$ is a vector of modal coefficients. This vector represents the free-parameters of the global shape-model.

6.4.2 Landmark displacements

The matrix formulation of the point-distribution model allows the global shape-deformation to be computed. However, in order to develop our EM-like matching method we will be interested in individual point displacements. We will focus our attention on the displacement vector for the landmark point indexed i produced by the eigenmode indexed λ. The two components of displacement are the elements of long-vector $\vec{\phi}^{\lambda}$ indexed $2i-1$ and $2i$. We denote the displacement vector by $\vec{\psi}_i^{\lambda} = (\phi_{2i-1}^{\lambda}, \phi_{2i}^{\lambda})^T$. For each landmark point the set of displacement vectors associated with the individual eigenmodes are concatenated to form a $2 \times K$ displacement matrix. For the j^{th} landmark, the displacement matrix is

$$\Delta_j = (\vec{\psi}_j^{\lambda_1}|\vec{\psi}_j^{\lambda_2}|...|\vec{\psi}_j^{\lambda_K})$$

The point-distribution model allows the landmark points to be displaced by a vector amount which is equal to a linear superposition of the displacement-vectors associated with the individual eigenmodes. To this end let $\vec{r} = (r_1, r_2, ..., r_K)^T$ represent a vector of modal superposition coefficients for the different eigenmodes.

The mean position of landmark j is

$$\hat{v_j} = \frac{1}{N}\sum_{t=1}^{N} \vec{v}_j^t$$

The aim in this section is to develop an iterative method for aligning the point distribution model. At iteration n of the algorithm we denote the aligned position of the landmark point j by the vector

$$\vec{v}_j^{(n)} = \hat{v_j} + \Delta_j \vec{r}^{(n)}$$

We wish to align the point distribution model represented in this way to a set of observed data-points. The data-points to be fitted are represented by an unlabelled set of $|\mathcal{D}|$ point position vectors $\mathbf{w} = \{\vec{u}_1, \vec{u}_2, ..., \vec{u}_{|\mathcal{D}|}\}$. The size of this point set may be different to the number of landmark points $|\mathcal{M}|$ used in the training of the point-distribution model. The free parameters that must be adjusted to align the landmark points with $\mathbf{w}$ are the modal coefficients $\vec{r}$.

6.4.3 Alignment method

To develop a useful alignment algorithm we require a model for the measurement process. Here we assume that the observed position vectors, i.e. $\vec{u}_i$ are derived from the model points through a Gaussian error process. According to our Gaussian model of the alignment errors,

$$p(\vec{u}_i|\vec{v}_j, \vec{r}^{(n)}) = \frac{1}{2\pi\sqrt{|\Sigma|}} \cdot \exp\Big[-\frac{1}{2}(\vec{u}_i - \vec{v}_j^{(n)})^T\Sigma^{-1}(\vec{u}_i - \vec{v}_j^{(n)})\Big] \qquad (6.21)$$

where Σ is the variance-covariance matrix for the point measurement errors. Here we assume that the position errors are isotropic, in other words the errors in the x and y directions are identical and uncorrelated. As a result we write $\Sigma = \sigma^2 I_2$ where I_2 is the 2x2 identity matrix and σ^2 is the isotropic noise variance for the

point positions. With this model, the alignment step is concerned with minimising the weighted square error measure

$$\mathcal{E}_a = \sum_{i=1}^{|\mathcal{D}|} \sum_{j=1}^{|\mathcal{M}|} Q_{i,j}^{(n)} (\vec{u}_i - \vec{v}_j^{(n+1)})^T (\vec{u}_i - \vec{v}_j^{(n+1)}) \tag{6.22}$$

With the point-distribution model displacement process detailed earlier, the weighted squared-error becomes

$$\mathcal{E}_a = \sum_{i=1}^{|\mathcal{D}|} \sum_{j=1}^{|\mathcal{M}|} Q_{i,j}^{(n)} (\vec{u}_i - \hat{v}_j - \Delta_j \vec{r}^{(n+1)})^T (\vec{u}_i - \hat{v}_j - \Delta_j \vec{r}^{(n+1)}) \tag{6.23}$$

Our aim is to recover the vector of modal coefficients which minimize this weighted squared error. To do this we solve the system of saddle-point equations which results by setting

$$\frac{\partial \mathcal{E}_a}{\partial \vec{r}^{(n+1)}} = 0 \tag{6.24}$$

After applying the rules of matrix differentiation and simplifying the resulting saddle-point equations, the solution vector is

$$\vec{r}^{(n+1)} = (\sum_{j=1}^{|\mathcal{M}|} \Delta_j^T \Delta_j)^{-1} \{ \sum_{i=1}^{|\mathcal{D}|} \sum_{j=1}^{|\mathcal{M}|} Q_{i,j}^{(n)} \vec{u}_i^T \Delta_j - \sum_{j=1}^{|\mathcal{M}|} \hat{v}_j^T \Delta_j \} \tag{6.25}$$

Further simplification results if we note that the landmark covariance matrix is symmetric. Its individual eigenvectors are orthogonal to one another, i.e. $\vec{\phi}^{\alpha T} \vec{\phi}^{\beta} = 0$ if $\alpha \neq \beta$. As a result the modal co-efficient for the eigenmode indexed λ_k is

$$\vec{r}_k^{(n+1)} = \frac{\sum_{i=1}^{|\mathcal{D}|} \sum_{j=1}^{|\mathcal{M}|} Q_{i,j}^{(n)} \sum_{k=1}^{K} \vec{u}_i^T \vec{\psi}_i^{\lambda_k} - \sum_{j=1}^{|\mathcal{M}|} \sum_{k=1}^{K} \hat{v}_j^T \vec{\psi}_j^{\lambda_k}}{\vec{\phi}^{\lambda_k T} \vec{\phi}^{\lambda_k}} \tag{6.26}$$

Finally, we update the *a posteriori* alignment probabilities. This is done by substituting the revised parameter vector into the conditional measurement distribution. Using the Bayes rule, we can re-write the *a posteriori* alignment probabilities using the measurement density function

$$P_{i,j}^{(n+1)} = \frac{\pi_j^{(n)} p_{i,j}^{(n)}}{\sum_{j'=1}^{|\mathcal{M}|} \pi_{j'}^{(n)} p_{i,j'}^{(n)}} \tag{6.27}$$

where

$$\pi_j^{(n)} = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} P_{i,j}^{(n)} \tag{6.28}$$

Upon substituting the Gaussian distribution appearing in Equation (6.19), the revised alignment probabilities are related to the updated point positions in the following manner

$$P_{i,j}^{(n+1)} = \frac{\pi_j^{(n)} \exp(-\frac{1}{2\sigma^2}(\vec{u}_i - \vec{v}_j^{(n)})^T(\vec{u}_i - \vec{v}_j^{(n)})}{\sum_{j'=1}^{|\mathcal{M}|} \pi_{j'}^{(n)} \exp(-\frac{1}{2\sigma^2}(\vec{u}_i - \vec{v}_{j'}^{(n)})^T(\vec{u}_i - \vec{v}_{j'}^{(n)})} \tag{6.29}$$

6.5 Experiments

In this section, we provide some experimental evaluation of the new unified approach to correspondence and alignment. We again use the set of house images from the CMU/VASC image database. Although the graph matching methods are the same as described in Chapter 5, we use two kinds of alignment methods. The first of them is the iterative Procrustes alignment method. The details of the method can be found in Chapter 4. When dealing with non-rigid deformation problems, we provide another alignment method based on the point distribution model. Next we show the results by using iterative Procrustes alignment method under the unified framework.

6.5.1 Procrustes Alignment

Like in the previous chapters, we match the first house image to the other 9 house images in the sequence. Figure 6.1 shows the first six alignment results using the unified matching method described in this chapter. From left to right, and from top to bottom, the panels in Figure 6.1 respectively show the alignment of the first image to the second, third, fouth etc images in the house sequence. We display the results by placing the Delaunay graph of the feature points in

House index	0	1	2	3	4	5	6	7	8	9
UM	-	1.07	1.90	4.75	5.17	6.87	23.4	24.8	27.1	36.3

Table 6.1: Alignment errors of the UM method.

House index	0	1	2	3	4	5	6	7	8	9
Correct	-	29	28	29	29	26	7	4	4	5
False	-	0	0	0	0	2	17	26	20	16
Missed	-	1	2	1	1	2	6	0	6	9

Table 6.2: Correspondence allocation results of the UM method.

first image overlayed in its final configuration on the grey-scale images. Table 6.1 summarises the alignment errors by using the unified matching method(Refered to as UM). It is not difficult to see that the method breaks after the sixth images.

These results can be compared with the results from the iterative Procrustes alignment method(Figure 4.18 and Table 4.1). From Figure 6.2 we can draw the conclusion that the unified method outperforms the iterative Procrustes method with respect to improved alignment error.

Next we turn our attention to the effect of the unified method on correspondence assignment. In figure 6.3 to 6.7 we show the resulting correspondences with segments connecting the points in the first image with their correspondences in a sequential image in the house sequence. Table 6.2 summarises the results. We can see that from the 6th image there are significant number of false correspondences.

To conclude this subsection, we compare the unified method with the iterative Procrustes method and the matrix factorisation graph matching methods derived in the previous chapters. We compare three measures of performance. These are the fraction of correct correspondences, false correspondences and missed correspondences. From the performance curves in Figures 6.8, 6.9 and 6.10, we

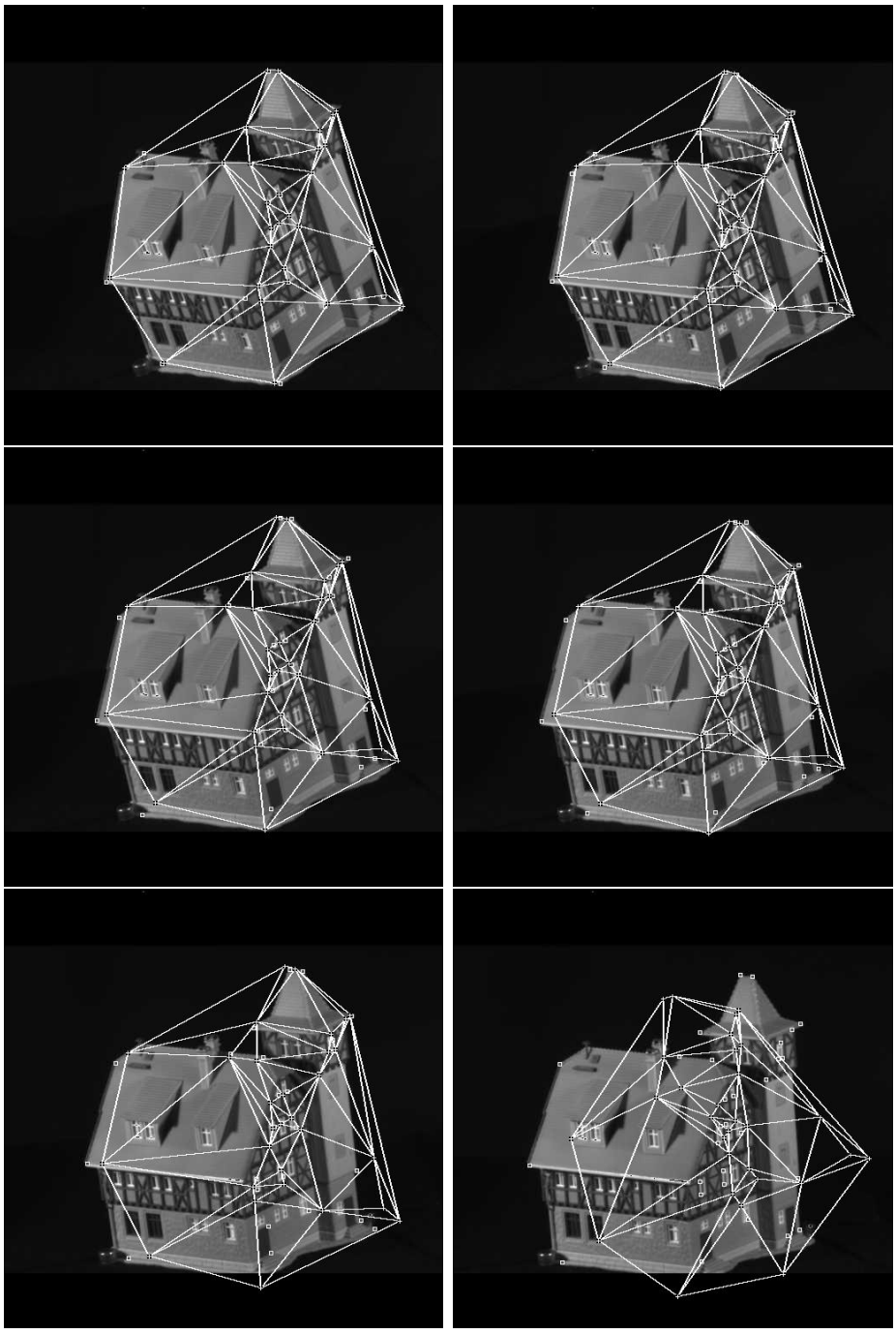


Figure 6.1: House image alignment results(UM).

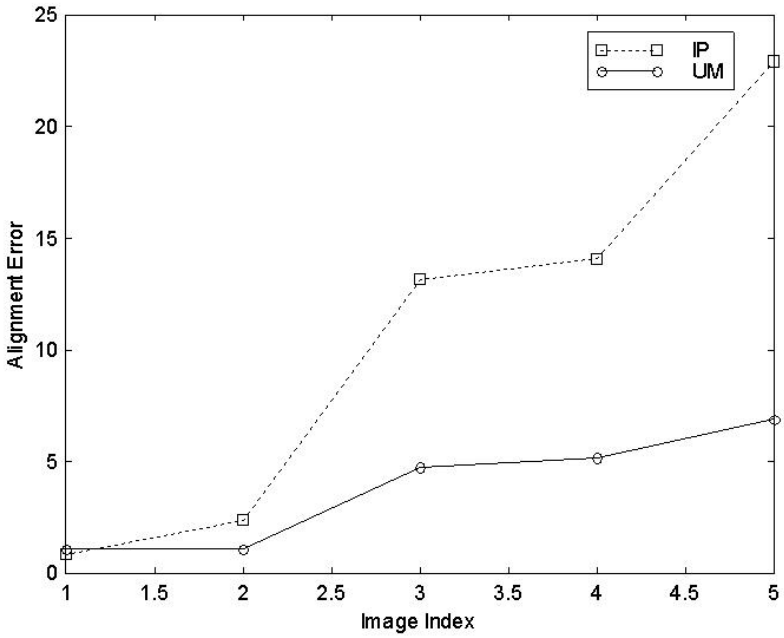


Figure 6.2: Comparison of the IP and UM methods – alignment error.

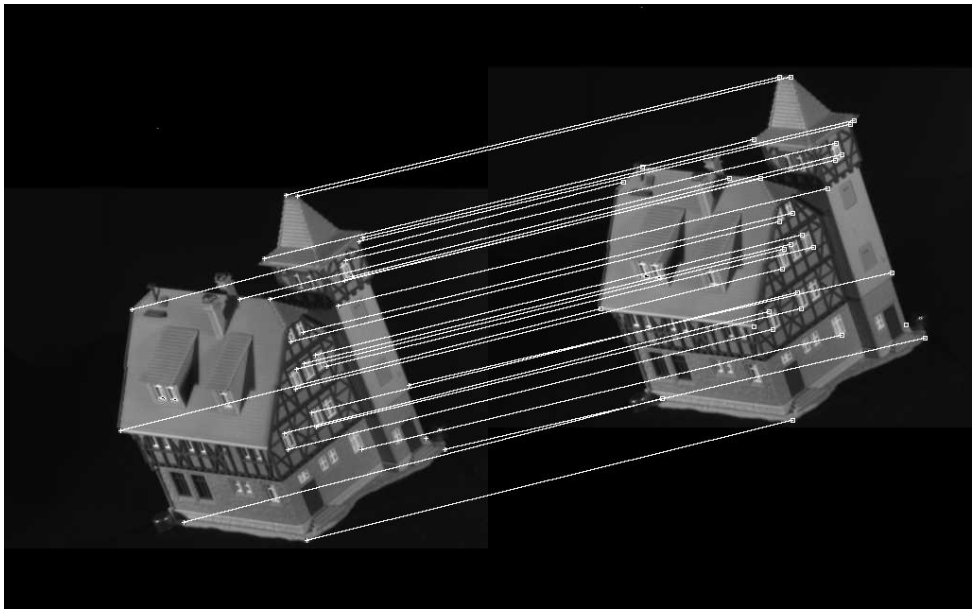


Figure 6.3: Correspondences between the first and the second images.

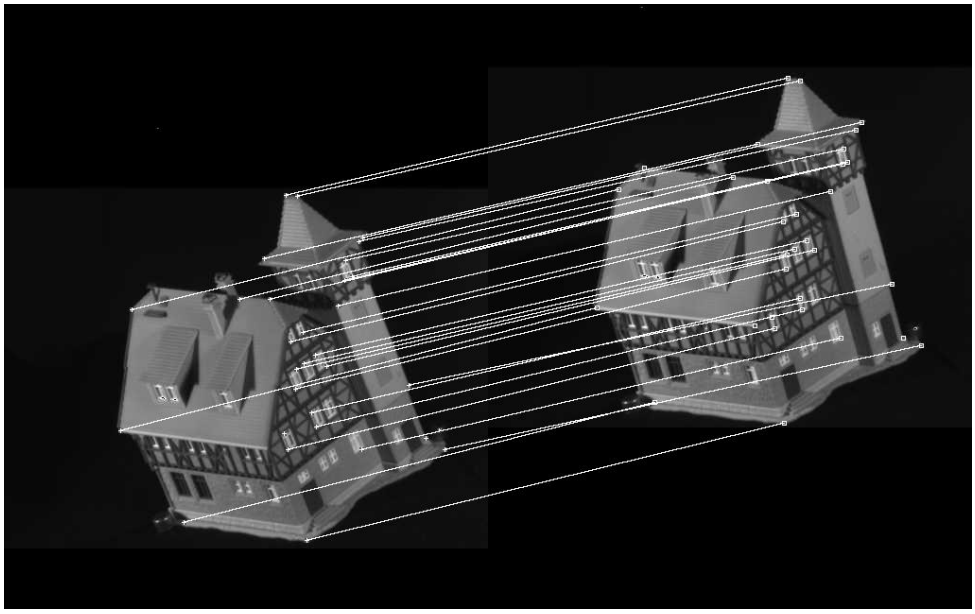


Figure 6.4: Correspondences between the first and the third images.

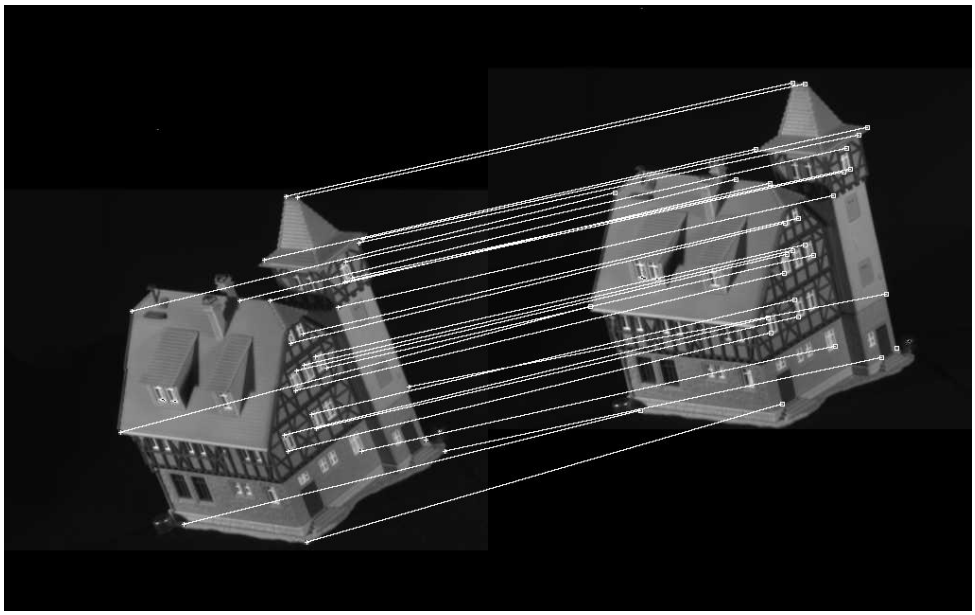


Figure 6.5: Correspondences between the first and the fourth images.

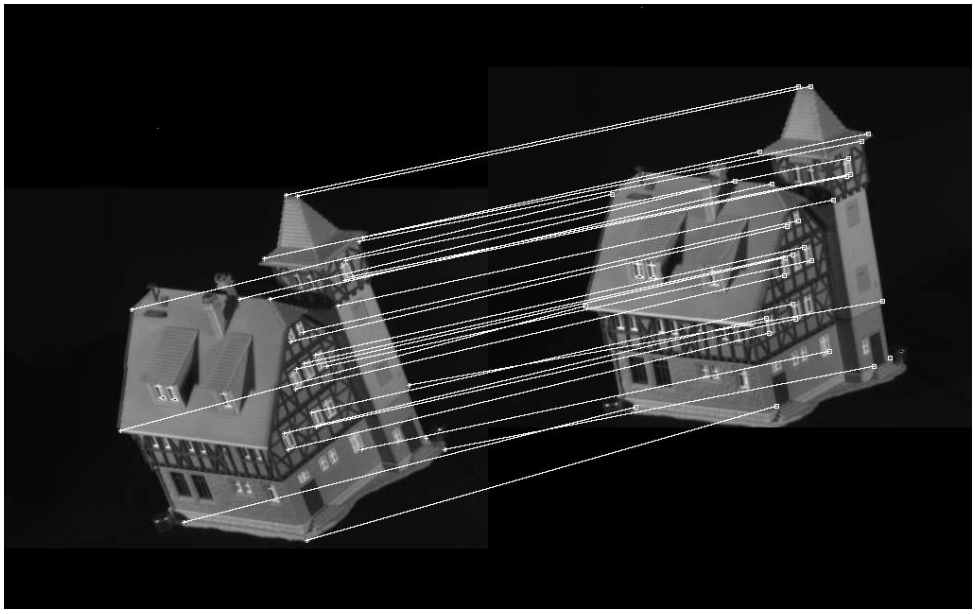


Figure 6.6: Correspondences between the first and the fifth images.

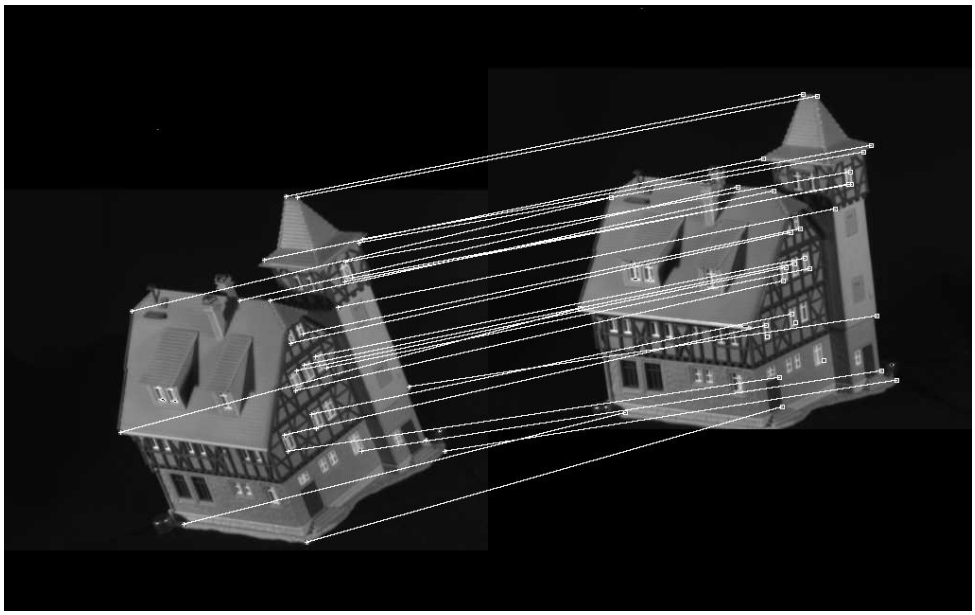


Figure 6.7: Correspondences between the first and the sixth images.

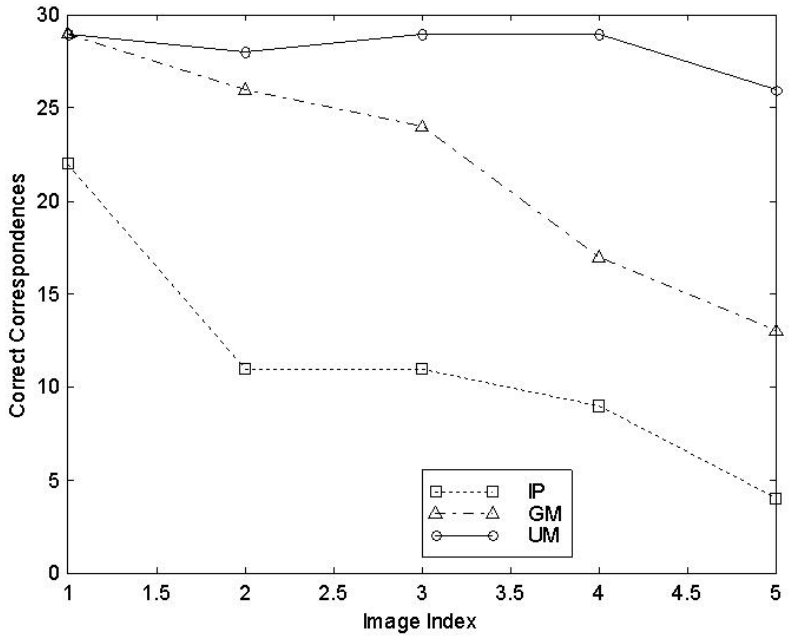


Figure 6.8: Comparison of the IP, GM and UM methods – correct correspondences.

conclude that the matrix factorisation graph matching method outperforms the iterative Procrustes method although it has relatively more false correspondences. While the unified matching method is the best in respect to all three performance measures.

6.5.2 Point Distribution Model fitting

We have experimented with our new point distribution model alignment method on an X-ray angiogram image sequence of a beating heart. Here the feature points are hand-labelled locations of maximum curvature on the outline of the heart. There are 16 feature points in each image. In total we have used 19 frames to train the point-distribution model. The mean shape is shown in Figure 6.11a superimposed on one of the images from the sequence; the different frames

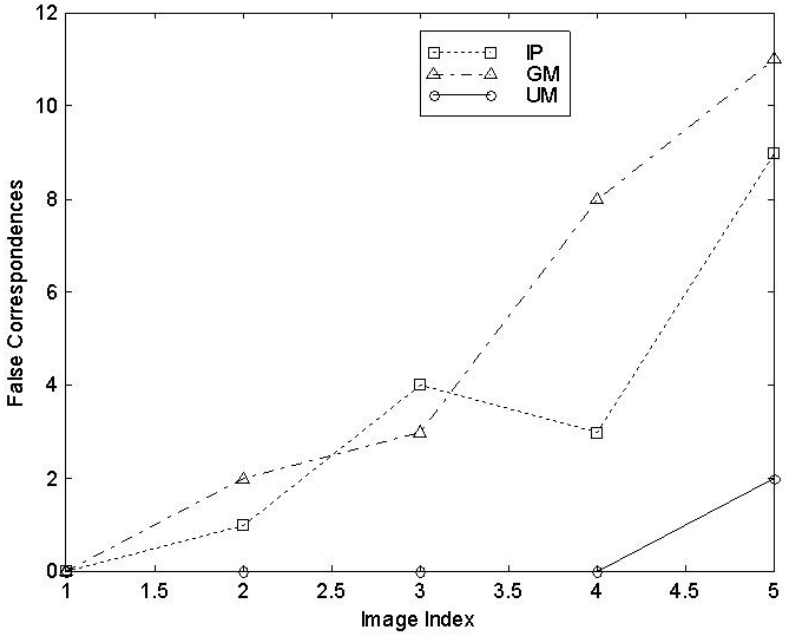


Figure 6.9: Comparison of the IP, GM and UM methods – false correspondences.

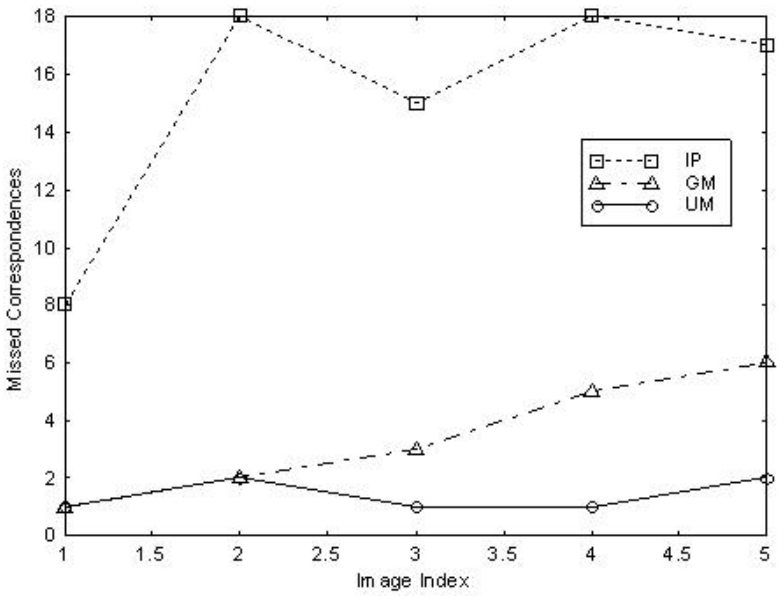


Figure 6.10: Comparison of the IP, GM and UM methods – missed correspondences.

used for training are shown in Figure 6.11b. In Figure 6.12 we show the modal displacements corresponding to the first 6 eigenmodes.

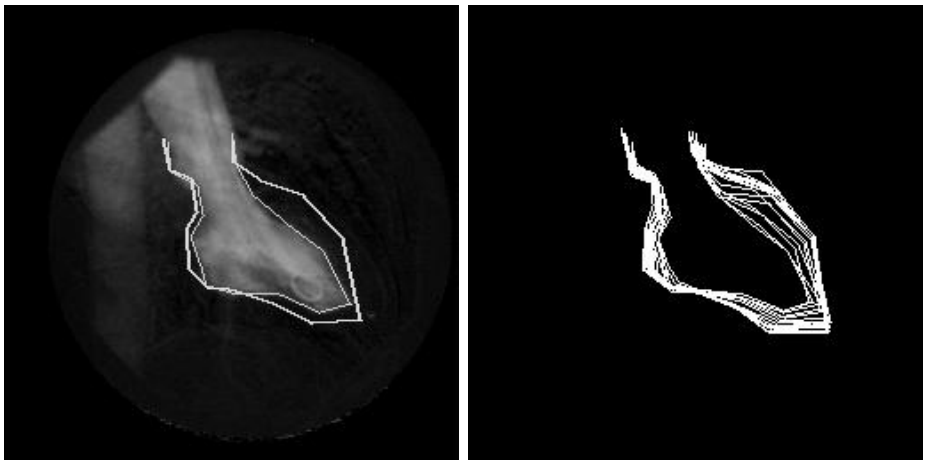


Figure 6.11: Overlapped mean shape and training images.

An example of the alignment of the PDM to an image which was not part of the training set is shown in Figure 6.13. The sequence shows the PDM iterating from the mean shape to the final alignment. The different panels in the figure show different iterations of the algorithm. The process converges in 10 iterations. In this example there is no initial correspondence assignment. The final set of correspondences obtained are shown in Figure 6.14. These are all correct.

In Figures 6.15 and 6.16 we illustrate the structural aspect of the matching process. Figures 6.15 shows the Delaunay graphs for the model point set image and another image from the sequence. The second image contains clutter. In Figure 6.16 we show the result obtained when we align the points in the model with those in the second image. Here we use the positions of the points from model to initialise the fitting of the point-distribution model. The crosses represent the initial positions of the model landmark-points. The lines represent the displacements of these points resulting from the alignment of the PDM. Notice that none of the feature-points is displaced in error so that it is brought into

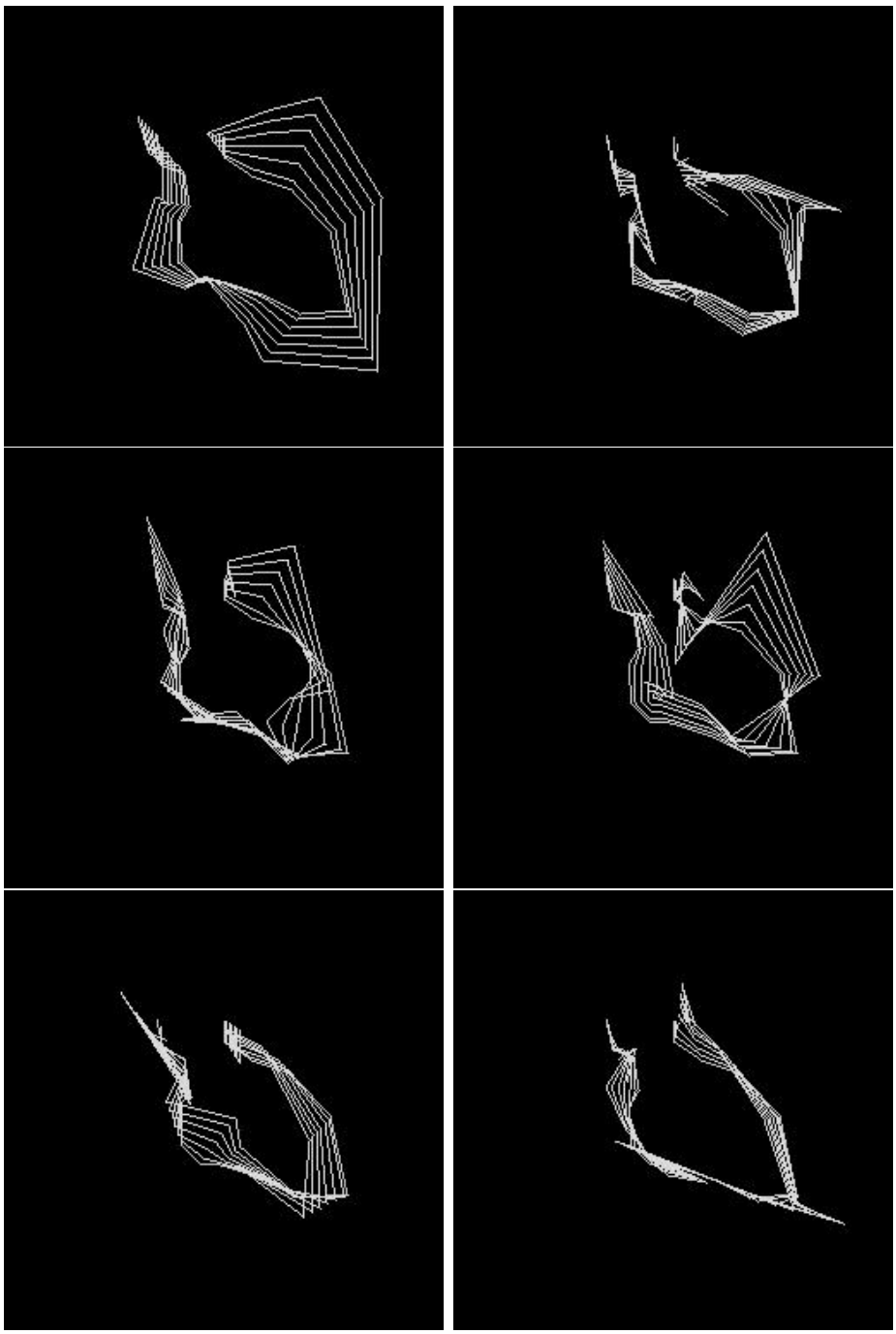


Figure 6.12: Eigenmodes of the training images.

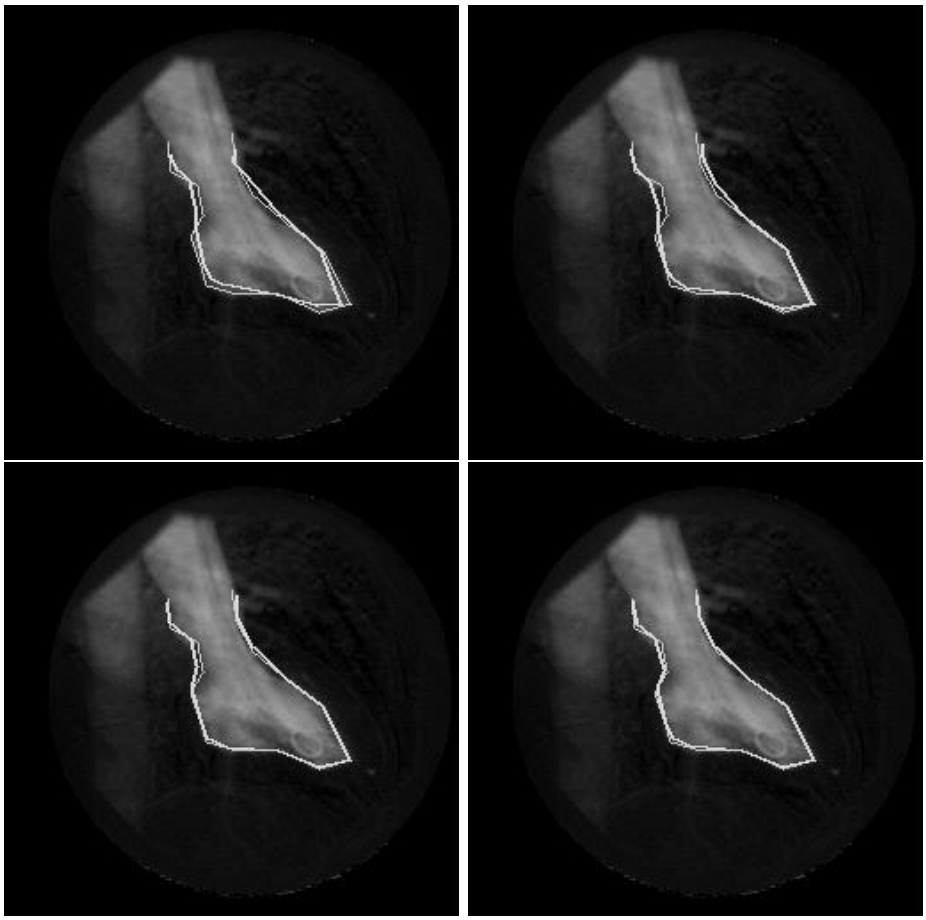


Figure 6.13: Alignment results of the heart images.

correspondence with the noise point.

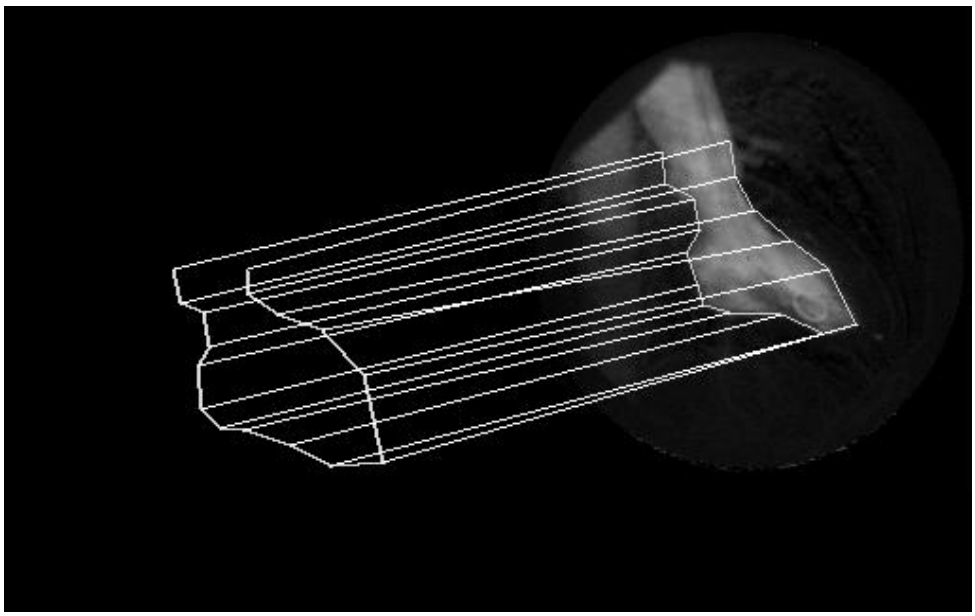


Figure 6.14: Correspondences of the heart images.

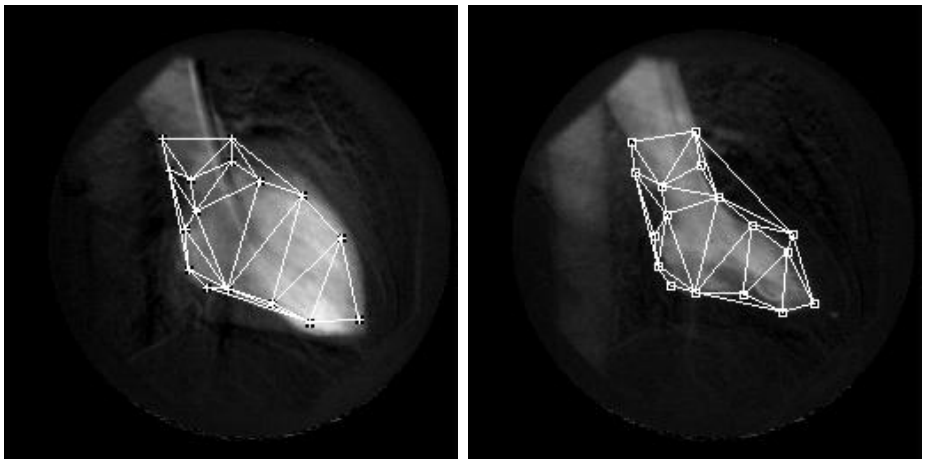


Figure 6.15: Graph structure of two point sets with clutter.

In Figure 6.17 and 6.18 we illustrate the effect of removing the adjacency graph from the point representation and the correspondence step from the matching process. To meet this goal we simply fit the point distribution model so as to

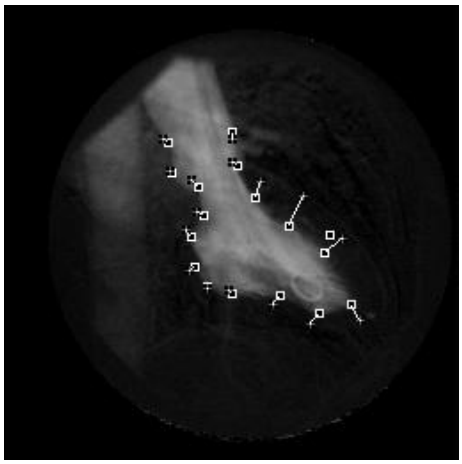


Figure 6.16: Correspondence result of two point sets with clutter.

minimise the quantity

$$\mathcal{E}_{EM} = \sum_{i=1}^{|\mathcal{D}|} \sum_{j=1}^{|\mathcal{M}|} \left[P_{i,j}^{(n)} \ln p_{i,j}^{(n+1)} \right] \tag{6.30}$$

This means that we use the EM algorithm to estimate the vector of modal parameters $\vec{r}$. Figure 6.17 shows the triangulated point-sets. The second point set contains a clutter point. When the EM algorithm is used to align the PDM, then this clutter point is matched in error to one of the landmark points. However, when our new unified matching framework is used to match the augmented PDM, then the clutter point finds to correspondence in the model.

Next, we turn our attention to some of the quantitative properties of the method. In Figure 6.19 we show the final RMS alignment error on the number of eigenmodes (ordered according to the size of their respective eigenvalues) used in the alignment process. Here there is slow improvement once we use more than 30% of the eigenmodes in the fitting process.

In Figures 6.20a and b, we investigate the effect of clutter on the matching process. In Figure 6.20a we show the evolution of one of the fitted PDM parameters with iteration number. The dotted curve is the result obtained with

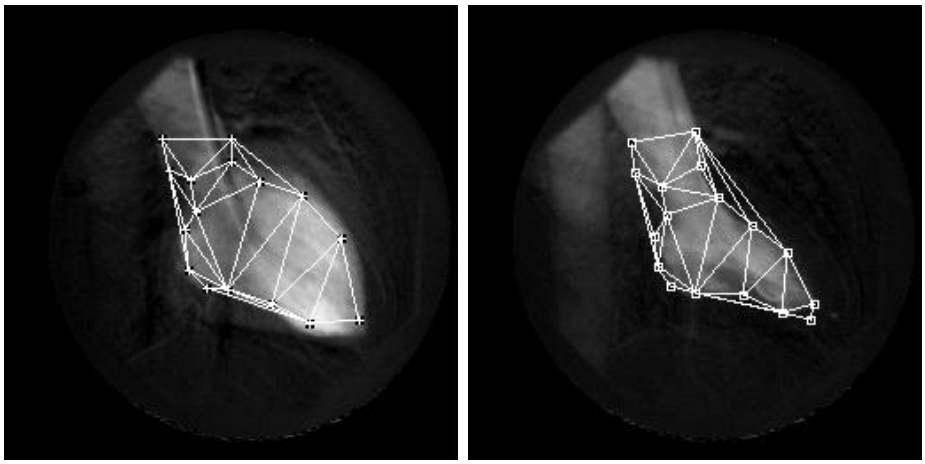


Figure 6.17: Graphs for algorithm comparison.

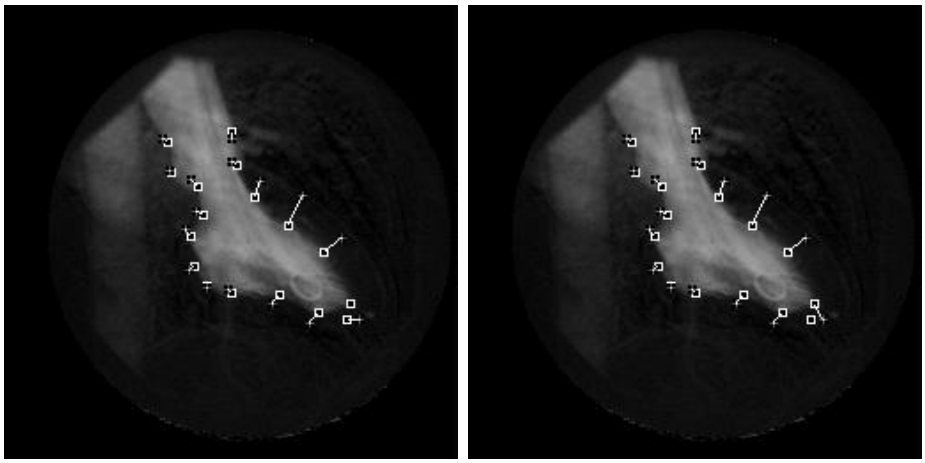


Figure 6.18: Comparison of EM-PDM and GRAPH-PDM.

the EM algorithm, while the solid curve is the result obtained when we use the new method reported in this chapter. The main feature to notice is that both methods converge to the same parameter values, but that the convergence of the graph-based method is faster. In Figure 6.20b we repeat this experiment when random noise points are added to the data. Here we should recover the same parameter values as in Figure 6.20a above. Both algorithms result in a significant parameter error. However, the graph-based method gives a final result which is

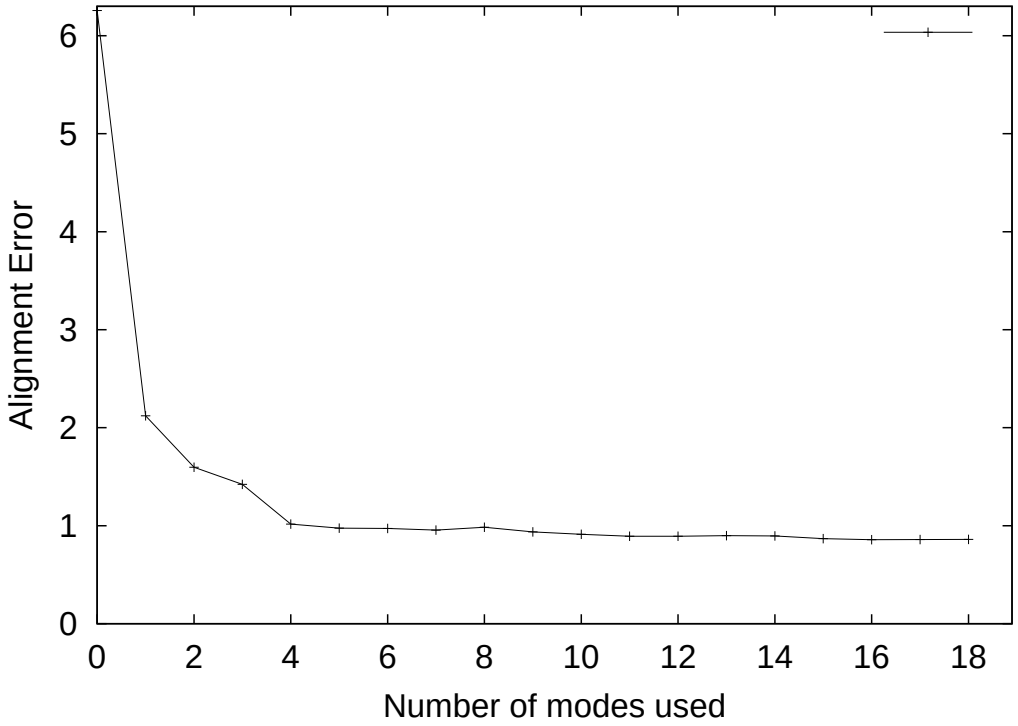


Figure 6.19: The effect of number of modes used to alignment error.

closer to the correct answer. Moreover, its convergence is again much faster than the EM method.

Next, we investigate the effect of added clutter on the alignment error. Figure 6.21 shows the alignment error as a function of iteration number for the EM algorithm and the graph-based method. In Figure 6.21a the fraction of added clutter is 6.25%. Here both methods converge to a result in which the alignment error is consistent with zero. However, the graph-based method converges at a much faster rate. Figure 6.21b repeats this experiment when the fraction of added clutter is 18.75%. Now both methods are subject to a substantial alignment error. However, in the case of the graph-based method this is smaller than that incurred by the EM method.

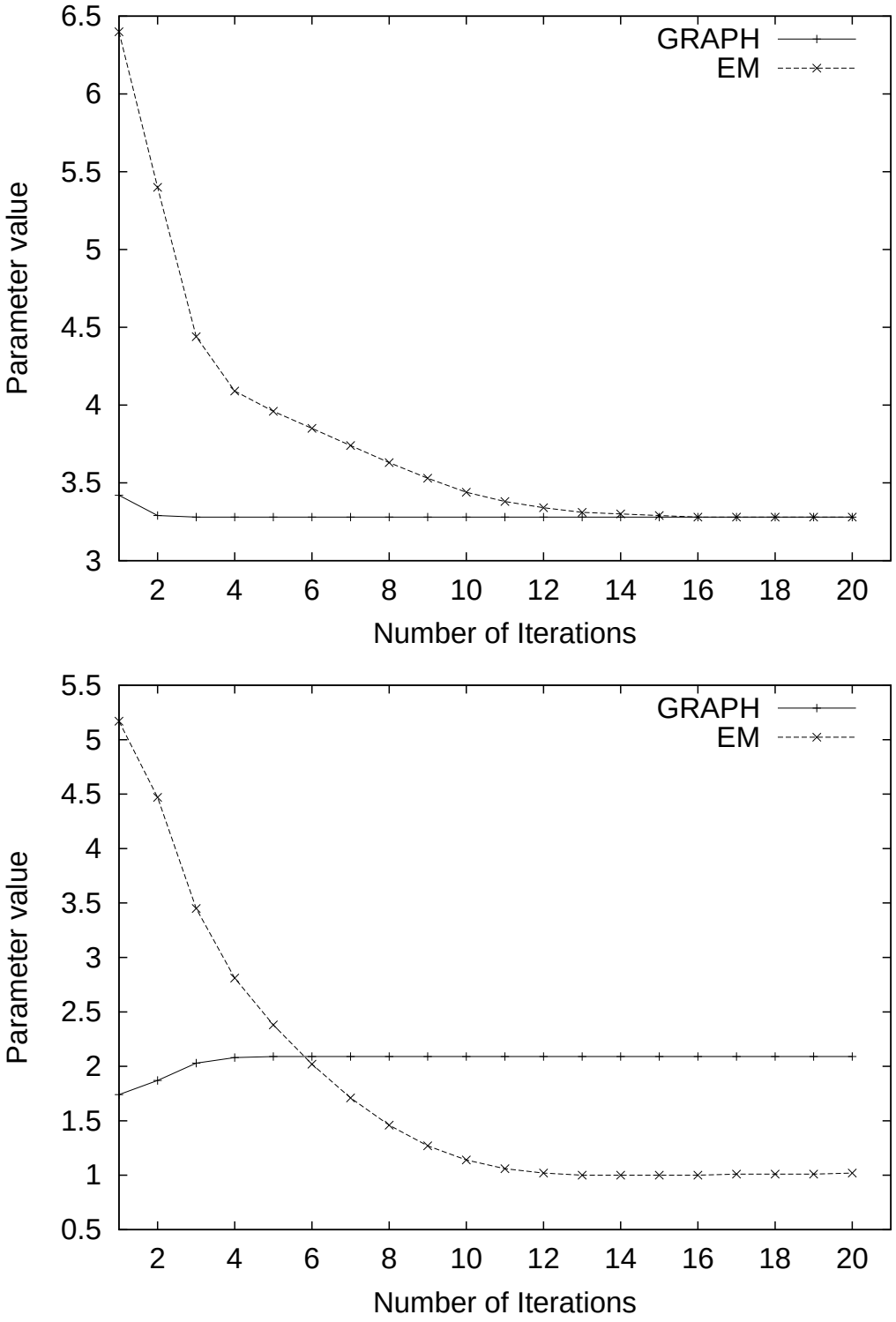


Figure 6.20: Comparison of PDM parameter value (a)Without clutter (b)With 18.75% clutter.

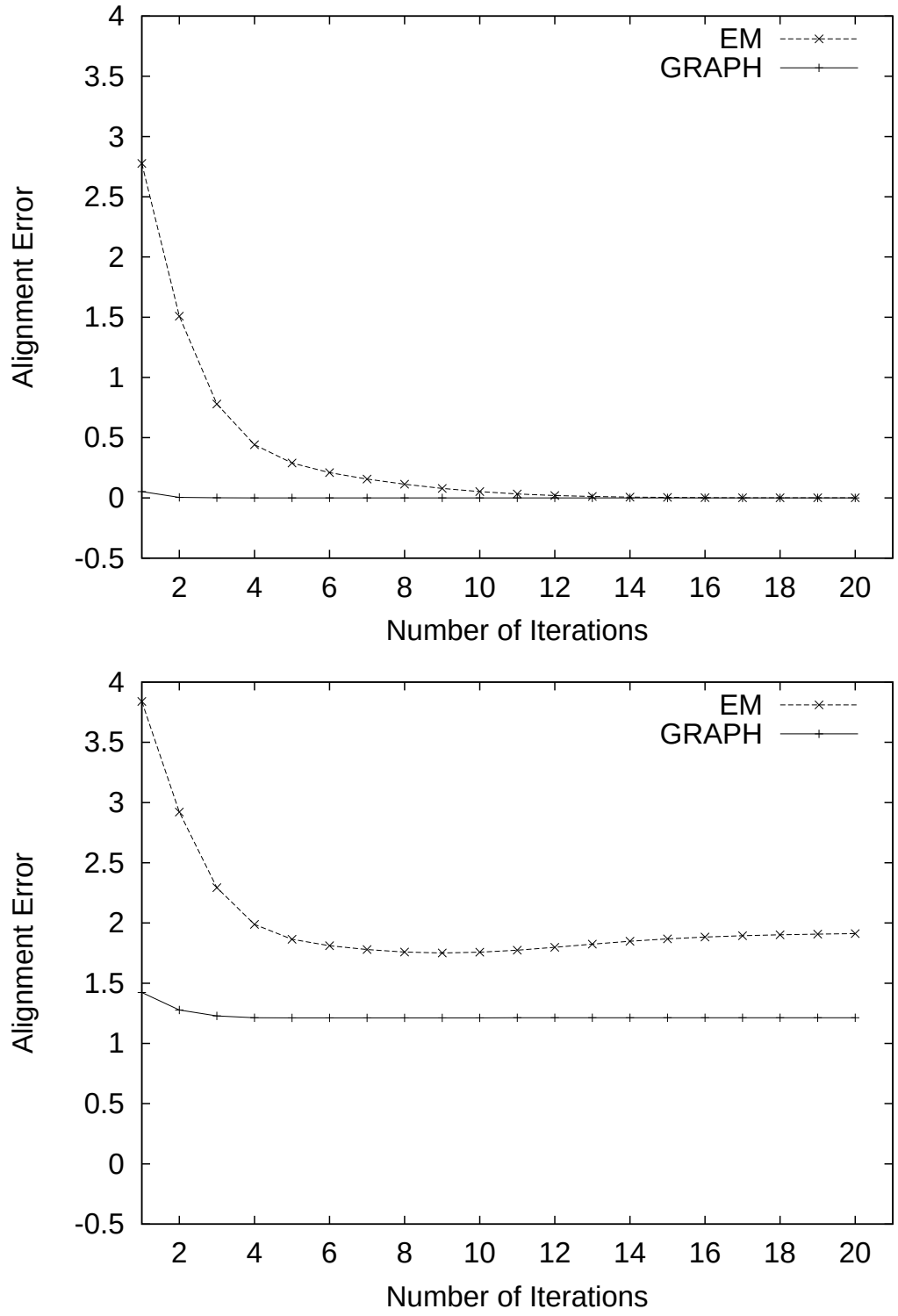


Figure 6.21: Comparison of the alignment error (a) 6.25% clutter (b) 18.75% clutter.

6.6 Conclusions

In conclusion, we have shown how the processes of point-set alignment and correspondence analysis can be unified using a symmetric entropy. For the problem of Procrustes alignment, we have shown that by drawing on a Gaussian model of point position errors and an exponential model of correspondence assignment errors, we are able to cast the two problems as maximisation of weighted correlation measures. In both cases the point matches can be recovered using singular value decomposition. Our new measures of point-set similarity naturally combine the ideas already developed by Scott and Longuet-Higgins and Umeyama in a single statistical utility measure. An experimental study reveals that the proposed method is outperforms that of the iterative Procrustes alignment method presented in Chapter 4 and the matrix factorisation graph matching method described in Chapter 5 in terms of its ability to recover alignment parameters and correspondence assignment.

In the case of point distribution model alignment, we have made two contributions. First, we show how the point distribution models can be augmented with point adjacency information. Second, we show how to fit the resulting model to noisy and unlabelled point-sets using a unified approach to correspondence and alignment. The method is shown to operate effectively when the landmark points are both unlabelled and subject to structural corruption. The method is both rapid to converge and robust to point-set contamination.

Chapter 7

Conclusions and future work

In this chapter, first of all, we summarize the main contributions of this thesis. These include corner detector through topographic analysis of the vector potential field representation, the iterative Procrustes alignment algorithm, the point distribution model fitting algorithm, the new matrix formalization for graph matching and the unified framework of point pattern alignment and correspondence. Second, we shall offer several possible suggestions for extensions and developments of the methods presented in this thesis.

7.1 Conclusions

The overall goal of this thesis was to pursue point pattern matching by applying statistical methods. We chose corners as the feature points in the images since we believe that corners are one of the most important image features and they are comparatively stable in real world images. Although many working algorithms exist, a robust and precise corner detector is always of interest. Based on the rich yet under exploited vector field representation of images suggested by Cross and Hancock(Cross and Hancock, 1999), our starting point is to develop a corner

detector by topographical analysis of the simulated vector potential field. We argue that corners are located at the points where saddle-ridges and saddle-valleys intersect. To support our argument, we have tested our algorithm with real indoor images and synthetic images. The results suggest that the proposed corner detector is superior to some existing corner detectors such as SUSAN(Smith and Brady, 1997), Wang and Brady's corner detector (Wang and Brady, 1995) and the Plessey corner detector (Harris and Stephens, 1988) in the terms of false positive suppression.

Procrustes analysis(Dryden and Mardia, 1998) can be used to match two point sets under particular transformations(Bookstein, 1997). The simplest one is the rigid transform. By factorising the cross correlation of the two feature sets under study using singular value decomposition, a general rotation matrix can be found which in turn can be used to align two point sets. Although it is proved very useful, the main drawback is that the two point sets must have same size and be labelled before hand. In solving real world problems, usually these two requirements can not be satisfied. To overcome the difficulties, we appeal to Cross and Hancock's EM framework(Cross and Hancock, 1998) for recovering transformation parameters.

By constructing a Gaussian mixture model over the unknown correspondences between the point sets, we arrived at a new statistical similarity measure. This is actually the weighted cross correlation of the two point sets. To effectively use SVD, we derived the matrix representation of the measure. The general rotation matrix can then be recovered by applying the SVD on the utility measure. We proposed an iterative Procrustes alignment algorithm. Real world examples were provided to show the effectiveness of the algorithm. A comparison with standard orthogonal Procrustes method shows the iterative Procrustes alignment algorithm is comparable with the standard Procrustes method in the sense of alignment

error. In our opinion, the new iterative method has two advantages. First of all it does not need correspondence information as prior information. Second, it can deal with point sets of different size.

Another approach to point pattern matching is correspondence assignment. We treat the point correspondence problem as graph matching problem. Graphs provide powerful tools for image presentation and have many concrete applications. Because of it's combinatorial nature, graph matching usually needs more computational effort. In this thesis, by constructing a mixture model over hidden correspondences and introducing a Bernoulli model for the structural error distribution, we arrived at a new likelihood function for graph matching. This likelihood function is maximised by using the apparatus of the EM algorithm. Another contribution to the topic of graph matching presented in this thesis is the matrix framework. By adopting the matrix representation, the maximisation can be effectively attained by using SVD. Real world image examples and synthetic examples are provided to illustrate the utility of the method. Comparing with other alternatives, our method delivers performance that is intermediate between the discrete relaxation method and the non-quadratic matching methods in the sense of recovering correct correspondences. While our method is computational less demanding.

As observed by Cross and Hancock(Cross and Hancock, 1998), the dichotomy between the alignment and correspondence processes is an artificial one. In Chapter 6 we combined the alignment and correspondence processes into a unified framework. Relational constraints were used to improve the recovery of alignment parameters. A unified utility measure was proposed based on the cross-entropy between probability distributions of alignment and correspondence errors. A dual step EM algorithm was used to match point sets under rigid and non-rigid transformations. Experimental results show that when we combine the alignment and

correspondence processes into a unified framework, smaller alignment error and more correct correspondences can be achieved.

After aligning two point sets with rigid deformation, what we have explored is how to align them with more complicated non-rigid deformations. Deformable models have proved very popular in recent years to model complicated variation of shapes. We took Cootes and Taylar's point distribution model(PDM)(Cootes et al., 1994) as a case study. Although the PDM is proved a useful model for describing shapes with non-rigid variations by using eigendecomposition of the covariance matrix of the point sets, it needs correspondences to be provided *a priori* and the point sets have to be the same size. These two drawbacks limit the use of the PDM. We described an algorithm to align the PDM to other deformed shapes under consideration. By constructing Gaussian mixture models over the unknown correspondences, we derived an update rule for the PDM parameters in the sense of maximum likelihood by applying the EM algorithm. The effectiveness of the algorithm is shown by aligning images form a sequence of X-ray angiograms of a beating heart.

In brief, we have made five contributions in this thesis. Firstly, we developed a unified framework for alignment and correspondence. The utility measure is the cross entropy using the Gaussian model of point sets alignment error and the exponential model of correspondence assignment error. Secondly, a matrix framework for point sets alignment is developed using the *a posteriori* correspondence probability weighted correlation between the two point sets. Thirdly, graph matching is achieved under a matrix framework with weighted correlation of adjacency matrix as the measure. Fourthly, a point distribution model is developed based on a series samples of X-ray angiogram images of heart beating images. Algorithms were developed for fitting the PDM model to a test pattern by using the proposed iterative Procrustes method and the unified alignment and corre-

spondence method. In addition, a corner detection algorithm is reported in the thesis to complete the process from low level feature points extraction to high level matching.

7.2 Future work

There are several obvious shortcomings of the methods proposed in the thesis which need further research effort.

In this thesis, we only use corners in the images to fulfill the complicated matching problem. While in real matching problem, important feature points do not always appear as corners. Other point features such as junctions, terminations and points derived from line features can also be used to match images. When we are able to robustly extract multi-features, the matching results will be more reliable.

In Chapter 4, we restrict our problem to be in the similarity transformation space since this is the easiest situation. When confronted with other more complicated transformations, such as those that arise from affine and perspective geometries, it is possible to extend our method by deriving the maximum likelihood solution for the different transformations accordingly by appealing statistical shape analysis theory. For instance, robust affine alignment method for point-sets could be derived using the method of point-set registration under affine transformations reported by Dryden(Dryden and Mardia, 1998). We also expect that our point alignment method could be enhanced to be able to cope with perspective transformations by using Lu, Mjolsness and Hager's exterior orientation estimation algorithm(Lu et al., 1996).

Content based image retrieval from large image databases is currently an active research area. Most research endeavor to find algorithms to recognise images

by using only feature attributes. This is disappointing since there are important semantic information in structural representation of images. The graph matching algorithm proposed in Chapter 5 could be used effectively to retrieve large scale structural databases. Both of Sengupta and Boyer's graph eigendecomposition model representation method(Sengupta and Boyer, 1998) and Horaud and Sossa's weighted graph indexing method(Horaud and Sossa, 1995) provided interesting examples.

Another possible application area of the point-set matching methods presented in this thesis is that of view based object recognition. The edge consistency measure of graphs (Equation 5.17) could be used as a pairwise graph distance measure. This graph distance measure could be used to cluster the images. The training images could be embedded into lower dimensions by using multidimensional scaling(MDS) techniques. Pose recovery and object recognition could be conducted by using a method which is similar with that of Murase and Nayar's parametric eigenspace analysis method(Murase and Nayar, 1994).

Bibliography

Ahmad, T., Taylor, C., Lanitis, A., and Cootes, T. F. (1997). Tracking and recognising hand gestures, using statistical shape models. *Image and Vision Computing*, 15(5):345–352.

Ambler, A. and et al. (1973). A versatile computer-controlled assembly system. In *Proc. 3rd Int. J. Conf. Arti. Intell.*, pages 298–307.

Amit, Y. and Kong, A. (1996). Graphical Templates for Model Registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:225–236.

Asada, H. and Brady, M. (1986). The curvature primal sketch. *IEEE Transactions on Pattern Analysis and Matchine Intelligence*, 8(1):2–14.

Barrow, H. and Popplestone, R. (1971). Relational descriptions in picture processing. *Machine Intelligence*, VI:377–396.

Beus, H. L. and Tiu, S. S. H. (1987). An improved corner detection algorithm based on chain-code plane-curves. *Pattern Recognition*, 20(20):291–296.

Bishop, C. (1995). *Neural networks for pattern recognition.* Oxford: Clarendon Press.

Blake, R. (1987). *Development of an incremental graph matching device*, volume 30 of *NATO ASI series*, pages 355–366. Spinger-Verlag, Berlin.

Bookstein, F. (1997). Shape And the Information in Medical Images: A Decade of the Morphometric Synthesis. *Computer Vision and Image Understanding*, 66(2):97–118.

Boyer, K. and Kak, A. (1988). Structural Stereopsis for 3D Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:144–166.

Bridle, J. (1990). Training stochastic model recognition algorithms can lead to maximum mutual information estimation of parameters. *Proc. Neural Information Processing Systems*, pages 211–217.

Bunke, H. and Allermann, G. (1983). Inexact graph matching for structural pattern recognition. *Pattern Recognition Letters*, 1:245–253.

Bunke, H. and Messmer, B. (1997). Recent advances in graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:169–203.

Bunke, H. and Shearer, K. (1998). A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19:255–259.

Bunke, H. (1999). Error correcting graph matching: On the influence of the underlying cost function. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:917–922.

Canny, J. F. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Matchine Intelligence*, 8(6):679–698.

Christmas, W., Kittler, J., and Petrou, M. (1995). Structural matching in computer vision using probabilistic relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):749–764.

Chung, F. (1997). *Spectral Graph Theory*. American Mathmatical Society Ed., CBMS series 92.

Cohen, I., Cohen, L. D., and Ayache, N. (1992). Using deformable surfaces to segment 3D images and infer differential structure. *CVGIP:Image Understanding*, 56(2):243–263.

Cohen, L. D. and Cohen, I. (1993). Finit element methods for active contour models and balloons for 2D and 3D images. *IEEE Transactions on Pattern Analysis and Machne Intellegence*, 15(11):1131–1147.

Cohen, L. D. (1991). On active contour models and ballons. *CVGIP:Image Understanding*, 52(2):211–218.

Cootes, T., Hill, A., Taylor, C., and Haslam, J. (1994). Use of active shape models for locating structure in medical images. *Image and Vision Computing*, 12(6):355–365.

Cootes, T., Taylor, C., Cooper, D., and Graham, J. (1995). Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59.

Costabile, M. F., Guerra, C., and Pieroni, G. (1985). Matching shapes: a case study in time varying images. *Computer Vision, Graphics and Image Processing*, 29:296–310.

Cross, A. D. J. and Hancock, E. R. (1997). Scale-space vector field for feature analysis. In *IEEE Computer Society Conferece on Computer Vision and Pattern Recognition*, pages 738–743, San Juan,Puerto Rico. IEEE Computer Society.

Cross, A. D. J. and Hancock, E. R. (1999). Scale-space vector field for symmetry detection. *Image and Vision Computing*, 17(5-6):337–345.

Cross, A. and Hancock, E. (1998). Graph Matching with a Dual-Step EM Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1236–1253.

Cross, A., Wilson, R., and Hancock, E. (1997). Inexact graph matching with genetic search. *Pattern Recognition*, 30(6):953–970.

Davatzikos, C. (1996). Spatial normalization of 3d brain images using deformable models. *Journal of Computer Assisted Tomography*, 20(4):656–665.

Davatzikos, C. (1997). Spatial transformation and registration of brain images using elastically deformable models. *Computer Vision and Image Understanding*, 66(2):207–222.

Delanges, P., Benois, J., and Barba, D. (1995). Active contours approach to object tracking in image sequences with complex background. *Pattern Recognition Letters*, 16(2):171–178.

Delibasis, K. and Undrill, P. (1994). Anatomical object recognition using deformable geometric-models. *Image and Vision Computing*, 12(7):423–433.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum-likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Soc. Ser. B (methodological)*, 39:1–38.

Deriche, R. and Giraudon, G. (1993). A computational approach for corner and vertex detection. *International Journal of Computer Vision*, 10:101–124.

Deschenes, F. and Ziou, D. (2000). Detection of line junctions and line terminations using curvilinear features. *Pattern Recognition Letters*, 21(6-7):637–649.

Dreschler, L. and Nigel, H. (1992). Volumetric model and 3D trajectory of a moving car derived from monocular TV-frame sequence of a street scene. *Proceedings of IJCAI*, pages 692–697.

Dryden, I. and Mardia, K. V. (1998). *Statistical Shape Analysis*. John Wiley & Son Ltd.

Eshera, M. and Fu, K. (1986). An image understanding system using attributed symbolic representation and inexact graph-matching. *Journal of the Association for Computing Machinery*, 8(5):604–618.

Feldmar, J. and Ayache, N. (1996). Rigid, Affine And Locally Affine Registration of Free-Form Surfaces. *International Journal of Computer Vision*, 18(2):99–119.

Finch, A., Wilson, R., and Hancock, E. (1997). Softening discrete relaxation. *M Jordan M. Mozer and T. Petsche, editors, Advances in Neural Information Processing Systems*, 9:438–444.

Finch, A., Wilson, R., and Hancock, E. (1998). An energy function and continuous edit process for graph matching. *Neural Computation*, 10(7):1873–1894.

Fischler, M. and Elschlager, R. (1973). The representation and matching of pictorical structures. *IEEE Transactions on Computers*, 22(1):67–92.

Freeman, H. and Davis, L. S. (1977). A corner finding algorithm for chain code curves. *IEEE Transactions on Computers*, 26:297–303.

Gold, S. and Rangarajan, A. (1996). A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):377–388.

Gueziec, A. and Ayache, N. (1994). Smoothing and matching of 3D space curves. *International Jouornal of Computer Vision*, 12(1):79–104.

Gwydir, S. H., Buettner, H. M., and Dunn, S. M. (June,1994). Non-rigid motion analysis and feature labelling of the growth cone. In *IEEE Workshop on Biomedical Image Analysis,Seattle,WA*, pages 80–87. Los Alamitors, CA:IEEE Computer Society Press.

Han, M. H. and Jang, D. (1990). The use of maximum curvature points for the recognition of partially occluded objects. *Pattern Recognition*, 23:21–33.

Haralick, R., Watson, L., and Laffey, T. (1983). The topographic primal sketch. *IJRR*, 2:50–72.

Harris, C. G. and Stephens, M. (1988). A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151.

Herault, L., Horaud, R., Veillon, F., and Niez, J. (1990). Symbolic image matching by simulated annealing. In *Proceedings of British Machine Vision Conference*, pages 319–324.

Hoffmann, T. and Buhmann, J. (1997). Pairwise data clustering with deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:671–680.

Horaud, R. and Skordas, T. (1989). Stereo correspondence through feature grouping and maximal cliques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1168–1180.

Horaud, R. and Sossa, H. (1995). Polyhedral object recognition by indexing. *Pattern Recognition*, 28(12):1855–1870.

Hu, X. and Ahuja, N. (1994). Matching point features with ordered geometrc, rigidity, and disparity constrains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:1041–1049.

Jain, A. K., Zhong, Y., and Lakshmanan, S. (1996). Object Matching Using Deformable Templates. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 18(3):267–278.

Jolly, M. P. D., Lakshmann, S., and Jain, A. K. (1996). Vehicle segmentation and classification using deformable templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(3):293–308.

Jordan, M. and Jacobs, R. (1994). Hierarchical mixture of experts and the EM algorithm. *Neural Computation*, 6:181–214.

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: active contour models. *International Journal of Computer Vision*, 1(4):321–331.

Kendall, D. G. (1984). Shape manifolds,Procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society*, 16(MAR):81–121.

Kitchen, L. and Rosenfeld, A. (1982). Gray-level corner detection. *Pattern Recognition Letters*, 1:95–102.

Koplowitz, J. and Plante, S. (1995). Corner detection for chain coded curves. *Pattern Recognition*, 28(6):843–852.

Kristof, W. (1970). A theorem on the trace of certain matrix products and some applications. *Journal of Mathematical Psychology*, 7:515–530.

Lades, M., Vorbruggen, J., Buhmann, J., Lange, J., von der Maalsburg, C., Wurtz, R., and W.Konen (1993). Distortion-invariant object-recognition in a dynamic link architecture. *IEEE Transactions on Computers*, 42:300–311.

Lakshmanan, S. and Grimmer, D. (1996). A deformable template approach to detecting straight edges in radar images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):438–443.

Larvallee, S. and Szeliski, R. (1994). Recovering the position and orientation of free-form objects from image contours using 3D distance maps. *IEEE Transactions on Medical Imaging*, 13(1):378–390.

Leymarie, F. and Levine, M. D. (1993). Tracking deformable objects in the plane using an active contour model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):617–634.

Liu, H. C. and Srinath, M. D. (1990a). Corner detection from chain-code. *Pattern Recognition*, 23(1-2):51–68.

Liu, H. C. and Srinath, M. D. (1990b). Partial classification using contour matching in distance transformation. *IEEE Transactions on Pattern Analysis and Matchine Intelligence*, 12:1072–1079.

Luo, B., Cross, A., and Hancock, E. (1999). Corner detection via topographic analysis of vector potential. *Pattern Recognition Letters*, 20:635–650.

Lu, C., Mjolsness, E., and Hager, G. (1996). Online computation of exterior orientation with application to hand-eye calibration. *Mathematical and Computer Modeling*, 24(5):121–143.

Matas, J. and Kittler, J. (1993). Junction detection using probabilistic relaxation. *Image and Vision Computing*, 11(4):197–202.

McInerney, T. and Terzopoulos, D. (1995). A dynamic finint element surface model for segmentation and tracking in multidimentional medical images

with application to cardiac 4D image analysis. *Computerized Medical Imageing and Graphics*, 19(1):69–83.

McLachlan, G. and K.E., B. (1988). *Mixture models: Inference and applications to clustering*. New York; Marcel Dekker.

McReynolds, D. and Lowe, D. (1996). Rigidity Checking of 3D Point Correspondences Under Perspective Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1174–1185.

Mehrotra, R., Nichani, S., and Ranganathan, N. (1990). Corner detection. *Pattern Recognition*, 23(11):1223–1233.

Miller, J. V., Breen, D. E., O'Bara, R. M., and Wozny, M. J. (1991). Geometrical deformed models: A method for extracting closed geometric models from volume data. *Computer Graphics*, 25(4):217–226.

Mokhtarian, F. and Mackworth, A. K. (1992). A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Transactions on Pattern Analysis and Matchine Intelligence*, 14(8):789–805.

Morgera, S. and Cheong, P. (1995). Rigid-Body Constrained Noisy Point Pattern-Matching. *IEEE Transactions on Image Processing*, 4(5):630–641.

Moshfeghi, M., Ranganath, S., and Nawyn, K. (1994). Three-dimentional elastic matching of volumes. *IEEE Transactions on Image Processing*, 3(2):128–138.

Moshfeghi, M. (1991). Elastic matching of multimodality medical images. *CVGIP:Graphical Models and Image Processing*, 53:271–282.

Murase, H. and Nayar, S. (1994). Illumination planning for object recognition using parametric eigenspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(12):1219–1227.

Myers, R., Wilson, R., and Hancock, E. (2000). Bayesian graph edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:628–635.

Noble, J. (1988). Finding corners. *Image and Vision Computing*, 6(2):121–128.

Park, J., Metaxas, D., Young, A. A., and Axel, L. (1996). Deformable models with parameter functions for cardiac motion analysis from tagged mri data. *IEEE Transactions on medical imaging*, 15(3):278–289.

Pelillo, M., Siddiqi, K., and Zucker, S. (1999). Matching hierarchical structures using association graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1105–1120.

Pellilo, M. (1999). Replicator equations, maximal cliques, and graph isomorphism. *Neural Computation*, 11(8):1933–1955.

Pentland, A. and Sclaroff, S. (1991). Closed-form solutions for physically based shape modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):715–729.

Peterson, C. and Soderberg, B. (1989). A new method for mapping optimisation problems. *International Journal of Neural Systems*, 1:2–33.

Rangarajan, A., Chui, H., and Bookstein, F. L. (1997). The softassign procrustes matching algorithm. In *Information Processing in Medical Imaging*, pages 29–42. Springer.

Rangarajan, A., Gold, S., and Mjolsness, E. (1996). A novel optimizing network architecture with applications. *Neural Computation*, 8:1041–1060.

Rangarajan, K., Shah, M., and Brackle, D. V. (1989). Optimal corner detection. *Computer Vision, Graphics and Image Processing*, 48:230–245.

Ripley, B. D. (1996). *Pattern recognition and neural networks.* Cambridge University Press, New York.

Rosenfeld, A. and Johmston, E. (1973). Angle detection on digital curves. *IEEE Transactions on Computers*, 22:875–878.

Rosenfeld, A. and Weszka, J. S. (1975). An improved method of angle detection on digital curves. *IEEE Transactions on Computers*, 24:940–941.

Sanfeliu, A. and Fu, K. (1983). A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions Systems, Man and Cybernetics*, 13(3):353–362.

Schwarzinger, M., Noll, D., and Vonseelen, W. (1995). Object recognition with constrained elastic models. *Mathematical and computer modelling*, 22(4-7):163–184.

Sclaroff, S. and Pentland, A. (1995). Modal Matching for Correspondence and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:545–661.

Scott, G. and Longuet-Higgins, H. (1991). An Algorithm for Associating the Features of 2 Images. *Proceedings of the Royal Society of London Series B-Biological*, 244(1309):21–26.

Sengupta, K. and Boyer, K. (1998). Modelbase partitioning using property matrix spectra. *Computer Vision and Image Understanding*, 70(2):177–196.

Shapiro, L. and Brady, J. M. (1992). Feature-based Correspondence - An Eigenvector Approach. *Image and Vision Computing*, 10:283–288.

Shapiro, L. and Brady, J. M. (1995). Rejecting Outliers and Estimating Errors in an Orthogonal-regression Framework. *Phil. Trans. Roy. Soc. A*, 350:403–439.

Shapiro, L. and Haralick, R. (1985). A metric for comparing relational descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(1):90–94.

Shapiro, L., Zisserman, A., and Brady, J. (1995). 3D motion recovery via affine epipolar geometry. *International Journal of Computer Vision*, 16(2):147–182.

Shokoufandeh, A., Dickinson, S., Siddiqi, K., and Zucker, S. (1999). Indexing using a spectral encoding of topological structure. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 491–497.

Simic, P. (1991). Constrained nets for graph matching and other quadratic assignment problems. *Neural Computation*, 3:268–281.

Singh, A., von Kurowski, and Chiu, M. Y. (1993). Cardiac MR image segmentation using deformable models. In *Biomedical Image Processing and Biomedical Visualization*, pages 8–28, Bellingham, WA:SPIE. Vol. 1905 SPIE Proc.

Singh, A. (1990). Gray level corner detection - A generalization and a robust real-time implementation. *Computer Vision, Graphics and Image Processing*, 51(1):54–69.

Smith, S. and Brady, J. (1997). SUSAN - a new approach to low level image processing. *Int. Journal of Computer Vision*, 23(1):45–78.

Sprinzak, J. and Werman, M. (1994). Affine Point Matching. *Pattern Recognition Letters*, 15(4):337–339.

Suganthan, P., Teoh, E., and Mital, D. (1995). Pattern-recognition by graph matching using the potts mft neural networks. *Pattern Recognition*, 28(7):997–1009.

Terzopoulos, D., Witkin, A., and Kass, M. (1988). Constraints on deformable models: Recovering 3D shape and nonrigid motion. *Artificial Intelligence*, 36(1):91–123.

Thirion, J. P. (June,1994). Extremal points: Definition and application to 3D image registration. In *Proc. Conf. Computer Vision and Pattern Recognition(CVPR'94)*, pages 496–501. Los Alamitos, CA:IEEE Computer Society Press.

Tirthapura, S., Sharvit, D., Klein, P., and Kimia, B. (1998). Indexing based on edit-distance matching of shape graphs. *Multimedia Storage And Archiving Systems III*, 3527:25–36.

Torr, P. and Murray, D. W. (1997). The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix. *International Journal of Computer Vision*, 24:271–300.

Ullman, J. (1976). An algorithm for subgraph isomorphism. *JACM*, 23(1):31–42.

Umeyama, S. (1988). An eigen decomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:695–703.

Umeyama, S. (1991). Least Squares Estimation of Transformation Parameters between Two Point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380.

Umeyama, S. (1993). Parameterised Point Pattern Matching and its Application to Recognition of Object Families. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:136–144.

Wang, H. and Brady, M. (1995). Real-time corner detection algorithm for motion estimation. *Image and Vision Computing*, 13(9):695–703.

Wang, Y. and Lee, O. (1994). A Feature Seeking and Tracking Image Sequence Representation Scheme. *Image Processing*, 3(5):610–624.

Werman, M. and Weinshall, D. (1995). Similarity and Affine Invariant Distances between 2D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):810–814.

Williams, M., Wilson, R., and Hancock, E. (1999). Deterministic search for relational graph matching. *Pattern Recognition*, 32(7):1255–1271.

Wilson, R. and Hancock, E. (1997). Structural matching by discrete relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):634–648.

Wong, A. and You, M. (1985). Entropy and distance of random graphs with application to structural pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:509–609.

Xie, X., Sudhakar, R., and Zhuang, H. (1993). Corner detection by a cost minimization approach. *Pattern Recognition*, 26(8):1235–1243.

Yang, Z. and Cohen, F. (1999). Image registration and object recognition using affine invariants and convex hulls. *IEEE Transactions on Image processing*, 8(7):934–946.

Yezzi, A., Kichenassamy, S., Kumar, A., Oliver, P., and Tannenbaum, A. (1997). A geometric snake model for segmentation of medical imagery. *IEEE transactions on medical imaging*, 16(2):199–209.

Yuille, A. L., Hallinan, P. W., and Cohen, D. S. (1992). Feature Extraction from Faces using Deformable Templates. *International Journal of Computer Vision*, 8(2):133–144.

Yuille, A. and Kosowsky, J. (1994). Statistical physics algorithms that converge. *Neural Computation*, 6:341–356.

Yuille, A., Stolorz, P., and Utans, J. (1994). Statistical Physics, Mixtures of Distributions, and the EM Algorithm. *Neural Computation*, 6:334–340.