

Parametric Estimation of Affine Deformations without Correspondences

Csaba Domokos

Department of Image Processing and Computer Graphics
University of Szeged

September 2010

Supervisor: Dr. Zoltán Kató

A DISSERTATION SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
OF THE UNIVERSITY OF SZEGED



University of Szeged
Doctoral School in Computer Science

Dedication

To my wife and my mother
In memory of my father, László Domokos

Contents

Preface	xvii
List of Notations Used	xix
1 Introduction	1
1.1 Summary by Chapters	2
1.2 Summary by Results	4
2 Image Registration: Review and Problem Statements	7
2.1 State of the Art	8
2.2 Binary Image Registration	11
2.2.1 Related Approaches	12
2.3 Realignment of Deformed Object Fragments	13
2.3.1 Related Approaches	14
2.4 Contributions	16
2.5 Basic Solution	17
2.6 Registration vs. Matching	18
2.7 Invariance vs. Covariance	18
3 Parametric Estimation of Affine Deformations of Binary Images: Polynomial Solution	19
3.1 Estimation of Affine Transformations	19
3.1.1 Construction of the Polynomial System	21
3.1.2 Discussion	24
3.2 Numerical Implementation	25
3.2.1 Fuzzy Object Representation	27
3.3 Estimation in the Presence of Noise	28
3.3.1 Geometric Noise	29
3.4 Experimental Results	32
3.4.1 Comparison with Previous Approaches	34
3.4.2 Robustness	35
3.4.3 Real Images	37

3.4.4	Registration of Hip Prosthesis X-ray Images	37
3.5	Summary	39
4	Affine Shape Alignment Using Covariant Gaussian Densities	41
4.1	The Parameter Estimation of Affine Transformations	41
4.1.1	Construction of Covariant Functions	42
4.1.2	Linear Estimation of Affine Parameters	45
4.1.3	Compound Objects	46
4.1.4	Choosing the Integration Domain	47
4.2	Discussion	49
4.2.1	Relation to Metric-Based Approaches	49
4.2.2	Relation to Canonical Representation-based Methods	51
4.2.3	Moment-based Interpretation	52
4.3	Numerical implementation	53
4.3.1	Single Density	53
4.3.2	Compound Objects	54
4.4	Experimental Results	61
4.4.1	Single Density	61
4.4.2	Compound Objects	62
4.4.3	Robustness	64
4.4.4	Real Images	67
4.4.5	Registration of Traffic Signs	67
4.5	Summary	69
5	Realignment of Deformed Object Fragments	71
5.1	Realignment of Deformed Shape Fragments	71
5.1.1	Relation between One Pair of Shapes	72
5.1.2	Solving for all Shapes Simultaneously	73
5.1.3	Choice of ω Functions	74
5.1.4	Affine Transformations	75
5.2	Numerical implementation	76
5.2.1	Algorithmic Solution and Complexity	77
5.3	Experimental results	78
5.3.1	Robustness	80
5.3.2	Numerical Stability	82
5.3.3	Real Images	83
5.4	Summary	88
6	Conclusions	89

Appendices	91
Appendix A Summary in English	91
A.1 Summary by Chapters	92
A.2 Key Points of the Thesis	93
Appendix B Summary in Hungarian	95
B.1. A fejezetek áttekintése	96
B.2. Az eredmények tézisszerű összefoglalása	97

List of Figures

1.1	An example of binary image registration.	2
2.1	An example of realignment deformed shape fragments.	13
3.1	Sign ambiguity of the Jacobian determinant.	20
3.2	The effect of the ω functions.	22
3.3	Observations corrupted by additive Gaussian noise on the coordinates.	29
3.4	Registration results for synthetic image pairs. The first two columns show the <i>template</i> and its affine distorted <i>observation</i> to be matched, while the other columns contain the registration result of each given method. The <i>template</i> and its <i>registered observation</i> are overlaid in such a way that overlapping pixels are depicted in black, while non-overlapping ones are shown in light or dark grey, respectively.	33
3.5	The robustness of the proposed polynomial approach in the case of incomplete objects was evaluated on images where we randomly removed 5%, 10%, and 20% of the foreground pixels ((b)-(d)) of the original image (a). The registration results are shown as overlaid contours of the <i>registered</i> shape.	36
3.6	Registration results on real images. For each image pair, the first two rows contain the <i>template</i> and <i>observation</i> with overlaid contours of the segmented silhouettes, the third row shows the difference between the <i>registered</i> shapes and in the last row the evaluated error measure δ is given. Note that this value is related to the overlapping area, hence it also depends on segmentation errors as segmented regions will never match perfectly in practice.	38
3.7	Registration of hip prosthesis X-ray images. Each image pair of a given patient was taken over a period of time. The overlaid contour in the second row shows the aligned contour of the corresponding image in the first row. For each pair, we also computed the δ measure.	39

- 4.1 The Gaussian density function fitted over the binary shape yields a consistent colouring. (a) The original binary image. (b) A 3D plot of the Gaussian density function over the binary shape. (c) The Gaussian density represented as a greyscale image. 43
- 4.2 The Mahalanobis distance represented as a *covariant function*. Contours have been placed over the original greylevel images for easier evaluation. (a) The Mahalanobis distance over the original image in Fig. 4.1 and (b) over the transformed image. The transformation was a $1.5\times$ shearing along the x -axis. 44
- 4.3 Gaussian PDFs fitted over a compound shape yield a consistent colouring. (a) Original shape. (b) A 3D plot of the Gaussian PDFs over the elliptic domain with $r = 2$. (c) Gaussian densities represented as a greyscale image. The white contour defines component boundaries. . . . 47
- 4.4 Canonical representation of a *template* and its *observation*. 50
- 4.5 The effect of the applied ω s on the image in Fig. 4.1. Contour lines have been placed over the original greylevel images for easier evaluation. 54
- 4.6 The integrals were computed over an elliptic domain. (a) An example of the domain (bold lines) and the integrated function (dashed lines), which is the i^{th} *covariant function*. (b) In order to simplify the integral, the coordinate system should be translated to the mean of the i^{th} object and its covariance matrix should be diagonalized. 56
- 4.7 The effect of the applied ω s on the compound shape shown in Fig. 4.3. 59
- 4.8 Registration result. (a) Distorted *observation* of the sample image in Fig. 4.1 (rotated, scaled and sheared). (b) Registration result achieved by our Single density [16] method [16]. The image is obtained by applying the recovered inverse transformation to the *observation*. 62
- 4.9 Some registration results provided by our method (MPDF with finite sums over a grid [17]), where 10%, 50% and 90% of the foreground pixels of the *observation* are missing. The images in the last row are overlaid in such a way that overlapping pixels are depicted in grey, while non-overlapping ones are shown black, respectively. 63
- 4.10 Some typical registration results on the synthetic data set. The *template* and corresponding *observation* are shown in the first row and second row, respectively. The other rows contain the registration results provided by the method of Kannala *et al.* [57], Suk and Flusser [95], and our method (MPDF with an effective numerical scheme [19]). The resulting images in the last row are overlaid in such a way that overlapping pixels are depicted in grey, while non-overlapping ones are shown black, respectively. 65

4.11	Sample <i>observations</i> for testing robustness. In (c) the true affine contours are highlighted in red.	66
4.12	Registration results on real images. Top: the images used as <i>templates</i> . Bottom: the corresponding <i>observations</i> with the overlaid contour of the registration results.	67
4.13	Registration results on traffic signs. Top: the images used as <i>templates</i> . Bottom: the corresponding <i>observations</i> with the overlaid contour of the registration results	68
5.1	Sample results on 2D synthetic images.	79
5.2	Sample results on 3D synthetic images.	80
5.3	The evaluated error measures in ascending order on the synthetic dataset, where the y -axes have a logarithmic scale. The median of error measures are also shown by a horizontal grid.	81
5.4	Sample <i>observations</i> with various degradations.	81
5.5	Median of error measures achieved by the proposed method on the 2D and 3D synthetic datasets in the case of various types of error.	82
5.6	Examples of <i>observations</i> for various strengths of deformations (in the rows, with an average κ value of 100, 110 and 120, respectively) and different numbers of fragments (in the columns).	83
5.7	The evaluated error measures in ascending order for 1000 randomly generated test cases with different numbers of fragments and measures of distortion. In the last row, there are the histograms of average distortion (κ) for various strengths of deformations.	84
5.8	Solutions of the Tangram puzzle (the average alignment runtime of an image was about 50 sec.) with a rigid-body transformation. Top: <i>observations</i> were taken by a digital camera. Middle: solutions found in the Tangram manual. Bottom: the scanned <i>template</i> silhouettes with overlaid contours of aligned fragments.	85
5.9	Solutions of the Tangram puzzle, where an affine model is used. Top: <i>observations</i> were taken by a digital camera. Middle: solutions found in the Tangram manual. Bottom: the scanned <i>template</i> silhouettes with overlaid contours of aligned fragments.	86
5.10	Bone fracture reduction. The CPU time was 15 sec. for these 1 megavoxel CT volumes.	87

List of Tables

1.1	The connection between the thesis points and the corresponding publications.	4
3.1	Registration results of the proposed method on the benchmark dataset containing 49 282 synthetic observations of 56 shapes. There was no solution in 5.47% of the test cases.	34
3.2	Registration results (medians of error measures) of the proposed method on 32 724 filled shapes and 16 558 line drawings.	34
3.3	Median of error measures versus resolution of the <i>observation</i>	34
3.4	Median of error measures on 1686 randomly selected images using some related approaches and the proposed algorithm.	35
3.5	Median of error measures versus σ of the noise on 1377 randomly selected images.	35
3.6	Median of error measures versus the ratio of removed foreground pixels on 1289 randomly selected images.	36
3.7	Median of error measures versus the size of occlusion on 794 randomly selected images.	36
4.1	Registration results (medians of error measures) on 1000 images and the runtimes of the method proposed by Kannala <i>et al.</i> [57] and of our Single density method.	62
4.2	Median of error measures and runtimes provided by the method of Kannala <i>et al.</i> [57], Suk and Flusser [95], and our Single density [16] and MPDF with finite sums over a grid method [17] in the case when 0%, . . . , 90% of the foreground pixels of the <i>observation</i> are missing.	63
4.3	The median of error measures and runtimes on 1435 images provided by the method of Kannala <i>et al.</i> [57], Suk and Flusser [95], and our method (MPDF with an effective numerical scheme [19]).	64
4.4	Median of error measures vs. various types of errors provided by by the method of Kannala <i>et al.</i> [57], Suk and Flusser [95], and our method (MPDF with an effective numerical scheme [19]).	66

5.1	Mean, standard deviation and median of error measures achieved by the proposed method on the 2D and 3D synthetic datasets.	80
A.1	The connection between the thesis points and the corresponding publications.	94
B.1.	A tézispontokhoz kapcsolódó publikációk.	98

List of Algorithms

1	Pseudo-code of binary image registration using a polynomial system of equations.	26
2	Pseudo-code of binary image registration using a single density.	53
3	Pseudo-code of binary image registration using MPDF with an effective numerical scheme.	60
4	Coeffs() computes C_t and C_o for Algorithm 3.	60
5	Pseudo-code of realignment of deformed fragments.	78

Preface

Image registration is the process of aligning images of a scene, and the aim is to compensate for image differences introduced by different poses. In many studies, multiple images are acquired from objects at different times, and often with different imaging modalities. The alignment process places the images in a common frame of reference that enables the fusion of information in the images and the determination of scene changes. Of course there are a large number of application areas which demand image registration. Image registration is a critical component of many remote sensing, medical, and industrial image analysis systems and it is one of the main challenging problems in image processing. In the past two decades great progress has been made in medicine, resulting in a tremendous increase in the utilization of various modalities, and more can be expected in the future.

In many applications the variability of image features is so complex that the only feasible way to register such images is to reduce them to a binary representation and solve the registration problem in that context. Therefore binary registration is an important problem in itself for many complex image analysis tasks. While the registration of greyscale and colour images has been extensively investigated, the alignment of binary shapes has received less attention. In real situations, the obtained images are related by a projective transformation. Although the projective transformation is nonlinear, it can often be successfully modelled by an affine transformation, which is linear. Owing to its linear property, the affine transformation is of great importance in image registration. In this dissertation we will investigate and address the problem of the parametric estimation of affine deformations on binary images.

The first of my **acknowledgments** goes to my supervisor, Dr. Zoltán Kató, who always motivated me and gave me invaluable advice. The work we did together was sometimes hard, but I believe that he has always guided my research studies with skill and wisdom. I would especially like to thank my wife Erika, who has always had faith in me and was by my side even in the most difficult times. I wish to thank my mother Erzsébet, and my brother László for all the love they have given me. I am very grateful to my friends (Laci, HP, Ájven, Robi, Csabi, Szabi and Madár) who have always helped

and understood me. I am grateful to Prof. Joseph M. Francos, who inspired me at the start of my research studies in this field, to Dr. Endre Szabó for fruitful discussions about hip prosthesis images from a radiological point of view, and to Prof. András Palkó and to Prof. Endre Varga, MD. for providing the X-ray and CT images. My thanks goes to all my colleagues at the Institute of Informatics (including László Nyúl, Kálmán Palágyi, Attila Tanács, Péter Balázs, Károly Dévényi and Márk Nyári) who helped me by giving me good advice when needed. I would also like to express my gratitude to David P. Curley for scrutinizing and correcting this thesis from a linguistic point of view.

This work was partially supported by the Doctoral School of the University of Szeged, Foundation for Education and Research in Informatics, Mecenatúra Scholarship Agency, Hungarian Scientific Research Grant OTKA K75637.

Csaba Domokos, September 2010.

List of Notations Used

\mathbf{x}, \mathbf{y}	points of the <i>template</i> and <i>observation</i> , respectively
n	dimension
$\mathbb{R}^n, \mathbb{P}^n$	n -dimensional real and projective space, respectively
$\mathbf{x} = [x_1, x_2, \dots, x_n, 1]^T$	points with homogeneous coordinates ($\mathbf{x} \in \mathbb{P}^n$)
$\mathbf{x} = [x_1, x_2, \dots, x_n]^T$	points with inhomogeneous coordinates ($\mathbf{x} \in \mathbb{R}^n$)
\mathbf{A}	affine transformation
(\mathbf{A}, \mathbf{t})	affine transformation in the inhomogeneous case
$ \mathbf{A} $	Jacobian determinant of the transformation \mathbf{A}
a_{kj}, q_{kj}	element of the k^{th} row and j^{th} column of \mathbf{A} and \mathbf{A}^{-1} , respectively
\mathbf{A}^T	transpose of \mathbf{A}
$\mathbf{A}_{j,:}^{-1}$	j^{th} row of \mathbf{A}^{-1}
\mathbf{C}^+	pseudo-inverse of \mathbf{C}
$\mathbf{C}_{1..3,1}$	first three elements of the first column of \mathbf{C}
$\mathbf{C}_{j,1..c}$	first c elements of the j^{th} row of \mathbf{C}
\mathbf{I}	identity matrix
$\text{diag}(a_1, \dots, a_n)$	diagonal matrix with a_1, \dots, a_n
$f, g : \mathbb{P}^n \rightarrow \mathbb{R}$	<i>covariant functions</i> under the transformation \mathbf{A}
ω_p	function acting on the point coordinates
ω_c	function acting on the <i>covariant functions</i>
$\mathbb{1}_t, \mathbb{1}_o$	the characteristic function of the <i>template</i> and <i>observation</i> , respectively
$\mathbb{L}_t, \mathbb{L}_o$	labelling function of the <i>template</i> and <i>observation</i> , respectively
\mathcal{F}	foreground region of a shape
$\mathcal{F}_t, \mathcal{F}_o$	foreground region of the <i>template</i> and <i>observation</i> , respectively
F	set of the foreground pixels of \mathcal{F}
F_t, F_o, F_r	set of the foreground pixels of the <i>template</i> , <i>observation</i> and <i>registered image</i> , respectively

$\binom{n}{k} = \frac{n!}{k!(n-k)!}$	binomial coefficient
p_k, r_k	polynoms
$\deg(p_k)$	degree of polynome p_k
$\omega^{(k)}(\mathbf{x})$	k^{th} coordinate of $\omega(\mathbf{x})$
\mathbf{y}^*	points of the noisy <i>observation</i>
\mathcal{F}_o^*	foreground region of the noisy <i>observation</i>
$\varepsilon^*(\mathbf{y}^*)$	noise function
$\varepsilon_1^*(\mathbf{y}^*)'_{y_2^*}$	partial derivative of the second variable of ε_1^*
ϵ	average misalignent in pixel between the <i>template</i> and <i>registered</i> image
δ	absolute difference
\triangle	symmetric difference of two sets
$\tilde{\mathbf{A}}$	estimated affine transformation by our algorithm
$(\tilde{\mathbf{A}}, \tilde{\mathbf{t}})$	estimated affine transformation by our algorithm in the inhomogeneous case
$\mathbf{p}_i, \mathbf{p}'_i$	i^{th} foreground pixel of the <i>template</i> and <i>observation</i> , respectively
$\mathbf{p}_i^{(k)}$	k^{th} coordinate of \mathbf{p}_i
M, N	the number of foreground pixels of the <i>template</i> and <i>observation</i> , respectively
ℓ	number of shapes
μ, μ'	mean values; centre of the <i>template</i> and <i>observation</i> , respectively
Σ, Σ'	covariance matrices of the points of the <i>template</i> and <i>observation</i> , respectively
μ_i, Σ_i	mean vector and covariance matrix of value of the i^{th} shape, respectively
$\mu^{(k)}$	k^{th} coordinate of μ
$p(\mathbf{x}), s(\mathbf{y})$	Gaussian probability density functions of the <i>template</i> and <i>observation</i> , respectively
P, S	<i>covariant functions</i> based on the Mahalanobis distance
\mathcal{P}, \mathcal{S}	<i>covariant functions</i> based on the unnormalized $p(\mathbf{x})$ and $s(\mathbf{y})$, respectively
$\mathcal{P}_i, \mathcal{S}_i$	<i>covariant functions</i> for the i^{th} pair of shapes
$\mathcal{P}_0, \mathcal{S}_0$	<i>covariant functions</i> for the overall shape
$\mathbf{N}_t, \mathbf{N}_o$	normalizing transformations of the <i>template</i> and <i>observation</i> , respectively
$(\mathbf{N}_t, \mathbf{t}_t), (\mathbf{N}_o, \mathbf{t}_o)$	normalizing transformations of the <i>template</i> and <i>observation</i> , respectively, in the inhomogenous case

$\mathcal{D}_t, \mathcal{D}_o$	corresponding integral domains synthetically generated over the <i>template</i> and <i>observation</i> , respectively
$\mathcal{D}_i, \mathcal{D}'_i$	i^{th} object fragment on the <i>template</i> and <i>observation</i> , respectively
\mathcal{D}'_i	set of foreground pixels of the i^{th} object fragment
\mathbb{N}_0	$\mathbb{N} \cup \{0\}$
\mathbb{Z}^2	2-dimensional discrete set
\mathbb{R}_0^+	non-negative real numbers
$\lceil \cdot \rceil$	upper integer values
(\mathcal{F}, d)	metric space
$(\mathbb{E}, d_{\mathbb{E}})$	Euclidean space
$\text{Iso}(\mathcal{F}_t, d_t)$	set of isometries
$\varphi : (\mathcal{F}_t, d_t) \rightarrow (\mathbb{E}, d_{\mathbb{E}})$	embedding in the <i>template</i> to Euclidean space

Chapter 1

Introduction

*'To discover something is to see what everyone sees and,
think what no one has thought before.'*

Albert Szent-Györgyi

Image registration is a crucial step in almost every image processing application where images of different views or sensors of an object need to be compared or combined. Typical application areas include visual inspection [105], object recognition [7], target tracking in video sequences [91], monitoring land usage on satellite images [29], super resolution [33], image mosaicking [37], and medical image analysis [47]. In a general setting, one is looking for a transformation which aligns two images such that one image (called the *observation*) becomes similar to the second one (called the *template*). Due to the large number of possible transformations, because of the different imaging conditions, there is a huge range of variations of the object signature. In fact, each *observation* is an element of the orbit of the transformations applied to the *template*. Hence the problem is inherently *ill-defined* unless this variability is taken into account. The registration process itself is an important problem, and is regarded as one of the main areas of image processing, hence there is a great deal about it in the literature [13; 47; 66; 88; 115].

In the simplest case Euclidean and similarity deformations are assumed. In many medical applications (e.g. bone fracture examination) these kinds of transformation are considered because of the biomechanical properties of the objects. In real situations, the images obtained are related by a projective transformation (also called planar homography). Although the projective transformation is nonlinear, it can often be successfully modelled by an affine transformation, which is linear. Owing to its linear property, the affine transformation is of great importance in image registration. When radiometric degradation is also expected, but the segmentation of the images is available, then *binary image registration* could be a valid alternative. When highly nonlinear radiometric distortion is considered, then the images deformed by a linear transformation can still be registered. Fig. 1.1 shows an example of the binary registration problem.

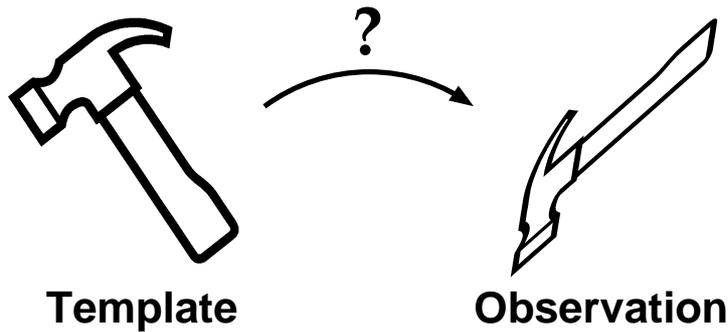


Figure 1.1: An example of binary image registration.

This dissertation summarizes the author's research results in binary image registration, where a given transformation between the shapes is a globally linear or nonlinear transformation that is made from by different linear transformations. Binary registration is an important problem in itself for many complex image analysis tasks. In many situations the variability of image features is so wide that the only feasible way to register such images is to reduce them to a binary representation and solve the registration problem in that context. On the other hand, a linear transformation is often considered as it has nice properties and it is a good approximation to a more complex nonlinear deformation (*e.g.* a projective transformation).

1.1 Summary by Chapters

In Chapter 2 we give a brief overview of the area of binary image registration. This review includes the state-of-the-art image registration methods and the best, relevant binary image registration techniques. This chapter will also present the preliminary notations used in the image registration field and the key ideas behind the proposed methods that will be used in the subsequent chapters.

The next two chapters address the problem of the parameter estimation of a global affine transformation of binary shapes. These chapters present novel methods which provide accurate and computationally simple solutions to the affine registration of shapes. Chapter 3 shows how the binary registration problem can be formulated as the solution of a system of polynomial equations obtained by integrating a set of polynomial functions over the shape domains. This novel method provides a direct solution without established correspondences or optimization. Moreover, the robustness of the resulting algorithm in the presence of i.i.d. Gaussian noise on the point coordinates and also segmentation errors is demonstrated. Comparative tests on partially occluded shapes reveal, however, that other *area-based* state-of-the-art methods cannot cope with occlusion either. Its performance on real images is demonstrated and the method

is applied to align pairs of hip prosthesis X-ray images. The advantage of the proposed solution is that it has a linear time complexity, is fast, easy to implement, works without established correspondences and it provides an exact solution regardless of the magnitude of the transformation.

The main difficulty with binary images is that they do not contain radiometric information; just the foreground pixel coordinates are available for the registration algorithm. Chapter 4 will show that in spite of the absent radiometric information, the registration problem can still be formulated as the solution of a linear system of equations, where all the available geometric information is used. The basic idea is to generate a pair of *covariant functions* that are related by the unknown transformation. The main contribution is the construction of these relations between shapes without establishing correspondences of any kind. The unknown transformation parameters are then found from the *least-squares* solution of an overdetermined system of equations. When the images contain compound shapes, an elegant and robust solution is proposed, where linear equations are constructed by integrating nonlinear functions over corresponding domains derived from compound shapes. This method is more robust and numerically more efficient than the previous approach, when images contain just a single object. The performance of the proposed methods were tested on a large synthetic dataset as well as on real images. The resulting algorithms are fast and provides a direct solution without establishing correspondences.

In Chapter 5 we propose a general framework to solve the realignment problem of deformed shape fragments without fixing the dimension of the input images (generally $n = 2, 3$). Given a *template* image with a set of shapes, and their affine distorted versions on the *observation* image, we would like establish a geometric correspondence between these images. This chapter introduces our registration problem followed by the basic ideas of the general solution with the assumption that only the overall segmentation of the *template* is known, *i.e.* its partitioning is hidden. The proposed method directly provides the parameters of the alignment by solving a polynomial system of equations. We also conduct an extended analysis of the numerical stability of the proposed algorithm. The performance and robustness of our method was investigated using large 2D and 3D synthetic benchmark datasets as well as in different real applications contexts (*e.g.* 2D Tangram, bone fracture alignment). In contrast to classical solutions based on landmark extraction and correspondences, the proposed solution finds the aligning transformations without any additional information. The main advantages are that the proposed method does not require correspondence, it is quite fast and easy to implement.

In Chapter 6 we draw some pertinent conclusion and discuss some interesting directions for future research.

1.2 Summary by Results

In the following a list of the most important results of the dissertation is given. Table 1.1 shows which thesis point is described in which publication by the author.

	[21]	[20]	[16]	[17]	[19]	[18]
I.	•	•				
II.			•	•	•	
III.						•

Table 1.1: The connection between the thesis points and the corresponding publications.

- I.) The author addresses the problem of the estimation of affine transformations for aligning a known 2D shape and its distorted *observation*. A novel approach is proposed where the exact transformation is obtained as the solution of a polynomial system of equations. The method was tested on synthetic as well as on real images and its robustness in the presence of segmentation errors and additive geometric noise was also demonstrated. The author successfully applied the method to the registration of hip prosthesis X-ray images. The advantage of the proposed solution is that it has a linear time complexity, is fast, easy to implement, works without established correspondences, and it provides an exact solution regardless of the magnitude of the transformation.
- II.) The author proposes a novel approach for the estimation of 2D affine transformations for aligning a planar shape and its distorted *observation*. The exact transformation is obtained as a least-squares solution of a linear system of equations constructed by fitting Gaussian densities to the shapes which preserve the effect of the unknown transformation. In the case of compound shapes, the author also presents a robust and efficient numerical scheme and achieves a near real-time performance. The proposed method was tested on synthetic as well as on real images. The robustness in the case of missing pixels, boundary errors, and modelling error is also demonstrated. The proposed method does not require point correspondences nor the solving of a complex optimization problem. It has a linear time complexity and provides an exact solution regardless of the magnitude of the deformation.
- III.) The author considers the problem of realigning broken objects without correspondences, where the segmentation of the overall *template* is known, but the segmentation of the object parts is unknown. The author applies linear transformations between the object fragments and presents the method by using 2D and 3D affine transformations. The basic idea is to construct and solve a polynomial

system of equations that provides the unknown parameters of the alignment. Here he quantitatively evaluated the proposed algorithm on a large synthetic dataset containing 2D and 3D images. The results show that the method performs fairly well and is robust against segmentation errors. He also presents the results of experiments on 2D real images as well as on volumetric medical images applied to surgical planning.

Chapter 2

Image Registration: Review and Problem Statements

In this chapter we will give a brief review of the state-of-the-art registration methods and we introduce some preliminary notations and ideas for the solution of the problem. Essentially, registration algorithms fall into two main categories: *Feature-based* and *Area-based* methods.

The *Feature-based* methods, as outlined in [7; 39; 71; 112], seek to establish point correspondences between two images. To do this, they extract several easily detectable features (like the intersection of lines and corners) from the images and then they use these points to compute the best transformation based on a similarity metric, or the aligning transformation is recovered directly as a solution of a system of equations constructed from the given correspondences based on these features. Therefore, to make this approach feasible, the correspondence problem must first be solved. Unfortunately, the solution to this problem is far from trivial and it usually relies on the assumption that the deformation is close to the identity (*i.e.* the size of the deformation is limited) and assumption that features provide strong contextual evidence for matching landmark points. Other approaches based on “continuous landmarks”, such as curves representing the boundaries of objects, usually lead to a complex, non-convex optimization problem that requires computationally expensive algorithms to solve it. Searching for the best transformation usually requires an iterative algorithm like the Iterative Closest Point (ICP) algorithm [8; 25]. The main drawback of these methods is that an optimization procedure has a high computational cost. Radiometric information plays a crucial role in establishing correspondences, hence in the case where radiometric information is not available (*e.g.* binary images) or it has been distorted by a highly nonlinear transformation, then the correspondence problem itself becomes quite challenging. The main advantage of these methods is that as long as a sufficient number of point matches are available, one can usually find an optimal aligning transformation, which suggests that these algorithms are less sensitive to occlusions.

Area-based (or featureless) methods [1; 16–18; 20; 21; 29; 41; 46; 57; 68; 95] treat the problem without attempting to detect well-defined points. Instead, the problem is solved by computing global descriptors [29] or invariants of the objects [46; 57], or by estimating the transformation parameters directly from image intensity values over corresponding regions [68] or defining a cost function based on a similarity metric and finding the solution via a complex nonlinear optimization procedure [44]. These methods are sometimes called correlation-like methods because they use a rectangular window to get some preliminary information about the distortion. They search for the position in the *observation* where the matching of the two windows is the best and then look for a good alignment between the windows in the *template* and in the *observation*. When the distortion is small, mutual information [58] as a similarity measure is often used. The drawback of this family of methods is the high computational cost and the restricted range of distortions (as with strong deformations, they may get stuck in a local minimum).

2.1 State of the Art

Several techniques have been proposed to address the affine registration problem. By thresholding the magnitude of the Fourier Transform of the images, Zhang *et al.* [114] construct affine invariant features, which are insensitive to noise, in order to establish a point correspondence. Several Fourier domain-based methods [54; 63] represent images in a coordinate system in which the affine transformation is reduced to an anisotropic scaling factor, which can be computed using cross-correlation methods. A novel one-element voxel attribute, the distance intensity (DI) is defined in [35]. This feature encodes spatial information at a global level, and the distance of the voxel to its closest object boundary, together with the original intensity information. Then the registration is obtained by exploiting mutual information as a similarity measure on the DI feature space. Govindu and Shekar [38] develop a framework that uses the statistical distribution of geometric properties of image contours to estimate the transformation parameters. The main advantages of these methods is that they do not need point correspondences across views and images may also differ by the overall level of illumination.

An affine registration algorithm for matching 2D feature points is presented in [49], which recovers both the aligning affine transformation as well as the unknown correspondences. The algorithm consists of two steps. First, the general affine case is reduced to the orthogonal case, then the unknown rotation is computed using the roots of a low-degree polynomial with complex coefficients. Another direct approach [84] extends the given pattern to a set of affine covariant versions, each having slightly different information, and then extracts features for registration from each of them separately.

In [74] the transformation is parameterized at different scales, using a decomposition of the deformation vector field over a sequence of nested (multiresolution) subspaces. A novel segment-based shape matching algorithm is presented in [72] which avoids problems associated with purely global or local methods. This approach generalizes the idea of finding a point-to-point correspondence between two shapes to that of finding a segment-to-segment correspondence.

In [29], an image registration algorithm based on affine moments is proposed. First, some representative regions are extracted which are matched based on the similarity of their moments. Then point correspondences are established as the centres of the region pairs and the transformation is recovered in the classical way by solving a system of equations constructed from the point correspondences. Although the resulting system is also linear, the solution is inherently less precise as in each equation only a part of the available information can be used. Flusser and Suk [28] propose a novel image normalization process in order to obtain unknown affine transformations based on affine moment invariants (AMI). Moments and invariants also provide an efficient tool for recovering linear deformations [95].

In many applications, registration appears as the problem of aligning a set of points. In [53], a robust approach is proposed, where each point set is represented by a mixture of spherical Gaussians and the point set registration is treated as a problem of aligning the two mixtures. For this purpose, the authors derived a closed-form expression for the L_2 -distance between two Gaussian mixtures, which in turn leads to a computationally efficient registration algorithm. Coherent Point Drift (CPD) [75] is a probabilistic approach for the non-rigid registration of point sets. CPD simultaneously recovers the non-rigid transformation and the correspondence between the point sets without making any prior assumption on the transformation model, except that of motion coherence. The registration is treated as a Maximum Likelihood estimation problem with a motion coherence constraint over the velocity field such that one point set moves coherently to align with the second set.

An energy function describing the interactions between the images is then minimized under a set of constraints, ensuring that the transformation maintains the topology in the deformed image. Manay *et al.* [67] explore an optimization framework for computing the shape distance and shape matching from integral invariants, which are employed for robustness to high-frequency noise. Shape warping by the computation of an optimal reparameterization allows this method to accommodate large localized changes such as occlusions and configuration changes. In [48], a method for identifying silhouettes from a given set of Radon projections is presented. The authors study how the Radon transform changes when a given 2D function is subjected to a rotation, scaling, translation, and reflection. Using these properties, the parameters of the aligning transformation are expressed in terms of the Radon transform.

Most of the existing approaches assume a linear transformation (rigid-body, similarity, affine) between the images, but in many applications nonlinear deformations [113] (e.g. projective, polynomial, elastic) need to be considered. An important class of nonlinear transformations is the plane-to-plane homography which aligns two images of the same planar object taken from different views. Lepetit and Fua proposed a method [64] for keypoint recognition on greyscale images. The main idea is to find keypoints during a training phase where a projectively different image set of target objects is used. Although the recognition of keypoints becomes very fast, the training phase is very time-consuming. In [52], planar homography is computed in the Fourier domain from a starting affine estimation using the shape contours. In [109], the concept of a characteristic line is employed to show some useful properties of a planar homography matrix, which is related to Euler angles of the planar pattern.

In [39], Guo *et al.* propose a method for registering shapes that have undergone diffeomorphic distortions¹, where simulated annealing is used to estimate point correspondences between the boundary points of the shapes. A Brownian motion model in the group of diffeomorphisms was introduced in [78]. The authors exploit a prior for warps based on a simple invariance principle under warping. An estimation based on this prior guarantees an invertible, source-destination symmetric, and warp-invariant warp. The maximum-likelihood warp is then computed via a Partial Differential Equation scheme. The authors of [89] use a Markov Random Field model to solve the registration problem. The deformation is described by a field of discrete variables, representing displacements of (blocks of) pixels. As finding the exact maximum a posteriori is intractable, a linear programming relaxation technique is used. In [101], the registration problem is formulated as a probabilistic inference using a generative model and the expectation-maximization algorithm. The authors define a data-driven technique which makes use of shape features. This gives a hybrid algorithm which combines generative and discriminative models. The measure of similarity is defined in terms of the degree of transformation required. The shapes are represented by sparse-point or continuous-contour representations depending on the form of the data. Klein *et al.* presented a stochastic gradient descent optimization method with adaptive step size prediction [59]. This method employs a stochastic subsampling technique to accelerate the optimization process. The selection mechanism for the method's free parameters takes into account the chosen similarity measure, the transformation model, and the image content, in order to estimate proper values for the most important cases.

¹A *diffeomorphism* is a bijective map such that both the map and its inverse are differentiable.

2.2 Binary Image Registration

In many situations the variability of image features is so complex that the only feasible way to register such images is to reduce them to a binary representation and solve the registration problem in that context [90]. X-ray images are good examples as they usually exhibit highly nonlinear radiometric distortions [22; 26], making registration hard to solve. Therefore *binary registration* is an important problem in itself for many complex image analysis tasks. While the registration of greyscale or colour images has been extensively investigated [3; 59; 63; 64; 78; 89], the alignment of binary shapes [7; 11; 20; 52; 95; 96; 109] has received less attention.

Radiometric information plays a crucial role in establishing correspondences. *Landmark-based* methods rely on the availability of rich radiometric information, *i.e.* they usually match local brightness patterns around salient points [65]. Although there are some time-consuming methods available to cope with brightness changes across image pairs [56] and there are robust keypoint detectors like SIFT [65] and SURF [5], such image degradations are difficult to handle. In many cases, however, such information may not be available, thus their use is limited in binary registration, since the correspondence problem becomes quite challenging. One can only use geometric information, but invariant geometric features (*e.g.* corners, junctions) might be difficult to extract (a circular shape, for instance). While these issues make classical brightness-based features unreliable, the segmentation of such images can be straightforward or readily available within a particular application. Therefore a valid alternative is to solve the registration problem using a binary representation (*i.e.* segmentation) of the images.

A recent approach for the binary registration of images was presented in [90]. Actually, the method addresses the registration of images taken under very different lighting conditions or in different seasons. Hence it is not possible to directly measure an invariant image feature. To overcome this difficulty, the authors extract edges from the images and compute some statistics of the edges, which is then used as a similarity metric for matching features. Although we are dealing with a different problem, this approach demonstrates the importance of the registration of binary images.

Belongie *et al.* proposed a novel approach for shape matching [7]. The method first searches for point correspondences between the two objects, then estimates the transformation using a generic thin plate spline model. The point matches are established using a novel similarity metric, called the *shape context*, which involves constructing a log-polar histogram of surrounding edge pixels. The advantage compared to traditional *landmark-based* approaches is that landmarks need not be salient points and radiometric information is not required. In essence, the method can be regarded as matching two points sets, each of them being a dense sample from the corresponding shape boundary. Obviously, there is no guarantee that point pairs will exactly correspond

because of the sampling procedure. However, having a dense sample will certainly keep mismatch error to a minimum. The correspondences are simply established by solving a linear assignment problem, which requires time-consuming optimization methods. For example, the complexity of the Hungarian method adopted in [7] is $\mathcal{O}(N^3)$, where N is the number of foreground pixels.

If perfect greylevel images were available *without any radiometric distortion*, the estimation of an aligning transformation would be reduced to finding the solution of a linear system of equations [40]. In real applications, however, such a strict requirement cannot be satisfied. Nevertheless, in the following chapters, we will show that registration can be solved without making use of any intensity information.

2.2.1 Related Approaches

Here, we describe some related binary image registration approaches. In the following chapters, we will compare our proposed methods to these methods. Probably the most closely related approach to ours is the binary registration algorithm proposed by Kannala *et al.* [57]. The fundamental difference is that the algorithm described in [57] constructs a system of equations by basically looking at the images at 3 different scales. Although the resulting system is linear, the solution is inherently less precise as in each equation the algorithm can only utilize part of the available information. But, our approach constructs the kind of equations that use all the available information in the images.

Flusser *et al.* presented an image registration algorithm based on affine moments [29]. First they extract some representative regions and compute their moments, then the regions from the *template* and *observation* are matched based on the similarity of their moments. After, point correspondences are established as the centres of the region pairs and the transformation is recovered in a classical way by solving a system of equations constructed from the point correspondences. While both methods make use of moments, the fundamental difference is that our method provides a direct solution without any point correspondences.

Another class of related methods consists of object matching based on image moments [46; 95]. Suk and Flusser [95] construct affine normalized images by employing image moments. An affine transformation is decomposed into elementary transformations, and then they are successively eliminated by central and complex moments. The aligning transformation of two objects is then obtained by affine normalizing (see Section 4.2.1) both images. This approach works well on synthetic images and on real images. We should add that the method described in [95] allows mirroring too, which is excluded in our model. In [46], Heikkilä constructs affine descriptors using higher order moments and moment invariants. Similar to the previous method, the transfor-

mation parameters are eliminated one by one. However, in many cases this may result in increased registration error as erroneous parameter values are fixed and propagated towards the computation of subsequent parameters.

2.3 Realignment of Deformed Object Fragments

Suppose we have the following general problem. Given a *template* image with an object, and linear (affine) distorted versions of its broken fragments on the *observation* image, we would like establish a geometric correspondence between these images (see Fig. 2.1). The overall distortion is a global nonlinear transformation with the following constraints:

- The object parts are distinct.
- All fragments of the *template* are available.
- Each of them is subject to a *different* affine deformation.

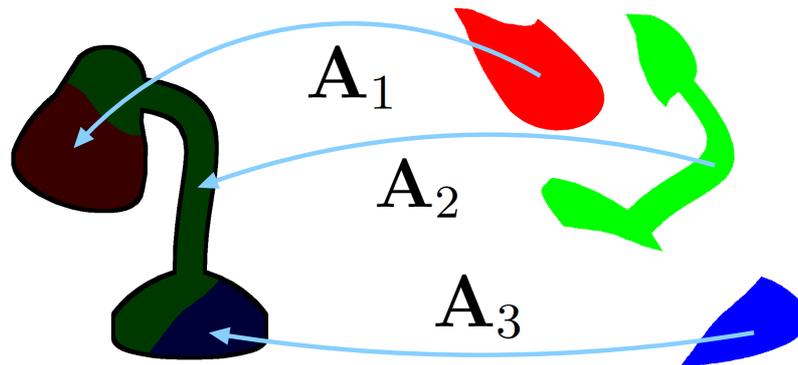


Figure 2.1: An example of realignment deformed shape fragments.

This is also known as the *puzzle* problem, which is not only interesting from a theoretical point of view [15; 62; 80; 110], but also arises in many application domains like archaeology [2; 51; 55; 70] and medical imaging [14; 47], e.g. bone fracture reduction [23; 82; 111]. Our goal is to reassemble the complete *template* object from its deformed parts. In general, most authors assume that fragments could be aligned by rigid-body transformations, since the purpose of these methods is to reassemble broken parts of the original object.

Although classical approaches may account for a *template* object by incorporating a set of constraints to improve the overall performance, they are usually applied to problems where a *template* is not available e.g. in archaeology [15; 51]. But, there are many applications where a *template* object is available. Often in industrial applications 3D models of manufactured parts can easily be produced. In this thesis, we will assume

that the overall segmentation of the *template* is unknown, *i.e.* its partitioning is hidden. In medical imaging an *atlas* can be used or, by taking advantage of the symmetry of the human body, the intact bone can provide a *template* for bone fracture reduction (see Section 5.3.3).

2.3.1 Related Approaches

In the past few decades, the problem of reassembling broken objects in an automatic way has gained increasing importance, and is mainly motivated by archaeology and medical applications. However, this problem is closely related to several challenging problems such as surface analysis, shape matching and object alignment.

Most of the existing solutions to the *puzzle* problem [15; 51; 62; 70; 80; 100] consist of matching fragment-pairs to find neighbours, which are then reassembled by a rigid body transformation. A general method for the automatic reassembly of broken 3D objects is presented in [51]. This method consists of a graph-cuts based segmentation for identifying potential fracture surfaces, and a feature-based global registration, which is based on integral invariants, for the pairwise matching of fragments. The local registration of multiple fragments is based on forward search techniques and surface consistency, where ICP [8] algorithm is also used. In [62], Kong and Kimia propose a 2D curve-matching technique based on the geometric features of puzzle pieces. The solution is obtained by a recursive grouping of triples using a best-first search strategy. The method can be extended to 3D fragments scanned by a laser range finder, where a pair of ridges are matched using a generalization of the 2D curve-matching approach. In [70], the rather high computational complexity of curve matching is reduced by adopting a multiscale technique. Papaioannou *et al.* addressed the problem of 3D object reconstruction by just using the surface geometry of fragments, without assuming anything about the final model to be reconstructed [80]. The basic idea behind the method is that the best fit of two 3D fragments is likely to occur at their relative overlaps, which minimizes the point-by-point distance between the mutually visible faces of the fragments. The matched pieces are then glued via a rigid-body transformation. In [100], a multiple-feature approach for determining matches between small fragments of archaeological artifacts is presented. The authors introduced a set of feature descriptors that are based not only on colour and shape, but also normal maps, which are easy to acquire. An efficient procedure for reassembling planar objects that have been broken into a large number (more than 1000) of irregular fragments is described in [15]. The procedure [15] compares the curvature-encoded fragment outlines, at progressively increasing scales of resolution, using an incremental dynamic programming sequence-matching algorithm. The method, proposed in [2], is based on the information extracted from the outlines and from the colour contents of the

fragments, without relying on any knowledge of the final image.

In [14], the problem of the computer vision-guided reconstruction of a fractured human mandible from a CT image sequence exhibiting multiple broken fragments is addressed. In the first phase of the proposed solution, the opposable fracture surfaces are identified using the Maximum Weight Graph Matching algorithm. Then identified pairs of opposable fracture surfaces are registered in the second phase using the ICP algorithm. In [36], a new algorithm for automatically solving jigsaw puzzles is presented. The algorithm [36] is able to solve more difficult puzzles and can handle puzzles in which pieces border more than four neighbours. A fast, robust, and accurate method for reassembling fragmented digital images is described in [32] and implemented by discrete Circular Harmonic expansions based on sampling theory. The computational efficiency of the algorithm [32] is given by the combined use of the correlation implemented by fast Fourier transforms for the location detection, and computation of the mutual angle by exploiting self-steerability properties of Circular Harmonic functions. In [102], a coordinate independent representation for the crack curves is developed and a new robust matching algorithm is proposed which finds matching pieces even when some brittle pieces are missing.

A related problem is the *partial matching* of shapes [11; 100], which addresses a particularly challenging issue of classical shape matching, where two shapes are in general dissimilar, but have several similar parts. In this context, our problem would require finding a partial matching between the *template* and each fragment of the *observation*. Current approaches are usually based on the Laplace-Beltrami framework [86; 87], but Bronstein *et al.* [11] proposed an elegant, ICP-based solution for the partial matching problem, where a non-rigid transformation is performed between 3D objects. In [11] the objective function is modified, where the dissimilarity, partiality and regularity are also controlled. The method solves the partial matching problem for one object. However, by successively applying this method, the problem can be solved for each shape fragment. Nevertheless, it requires precise parameter adjustment for each different shape, and considering the rather high computational complexity of these algorithms, the solution is far from optimal for our problem.

Another related problem, mainly found in medical imaging, is the *piecewise approximation* of nonlinear deformations by locally linear transformations. In [81], the distortion is modelled as a locally affine but globally smooth transformation, which accounts for local and global variations in image intensities. The classical solution [24] consists of identifying point correspondences based on salient points between the images and then either a time-consuming optimization procedure or the solution of a system of equations provides the parameters of the unknown deformation.

2.4 Contributions

In the following chapters we will propose novel methods which provide an accurate and computationally simple solution for the affine registration of planar shapes. The main difficulty with binary images is that they do not contain radiometric information; only the foreground pixel coordinates are available for the registration algorithm. We will show how the binary registration problem can be formulated as the solution of a system of polynomial/linear equations obtained by integrating a set of nonlinear functions over the shape domains. The proposed methods (see chapters 3-5) provide a direct solution without established correspondences or optimization. The basic idea is to generate enough relations that are related by the unknown transformation. Moreover, the robustness of the resulting algorithm in the presence of i.i.d. Gaussian noise on the point coordinates as well as segmentation errors are also demonstrated. But, being *area-based*, they are sensitive to occlusions. Comparative tests on partially occluded shapes reveal, however, that other *area-based* state-of-the-art methods cannot cope with occlusion either. Next, we demonstrate how well they work on real images and in various medical applications (e.g. registration of a hip prosthesis X-ray image, registration of traffic signs, bone fracture alignment). The main advantage of the proposed solution is that it has a linear time complexity, is fast, easy to implement, works without established correspondences and it provides an exact solution irrespective of the magnitude of the transformation.

The parametric estimation of two-dimensional affine transformations between two *grey-level* images was addressed by Hagege and Francos in [40–42], which provides an accurate and computationally simple solution that avoids both the correspondence problem as well as the need for optimization. The original problem was reformulated as an *equivalent linear parameter estimation* one having a *unique* and *exact* solution. However, the method relies on the availability of rich radiometric information, which is clearly not available in the binary case.

The methodology adopted here to solve the realignment problem is similar in spirit to the affine registration methods of [20] (see Chapter 3) and [43]. However, neither of these addresses the *puzzle* problem. In [43], it is assumed that radiometric information is available. Using this information, a linear system of equations is constructed, then its solution provides the parameters of the aligning transformations. Although the partitioning of the *template* is not available, this method [43] could be used here, but it requires rich radiometric information which is generally not available in the binary case. In [20], an elegant solution is presented to recover affine deformations between 2D shapes. The method [20] is also not able to solve the *puzzle* problem because the deformation is nonlinear. Moreover there is no direct correspondence between the *template* and its observed fragments.

2.5 Basic Solution

Here we will introduce our preliminary notations and also formulate the basic solution of the problem. From now on we shall just consider affine transformations. To this end, let us denote the homogeneous coordinates of the n dimensional *template* and *observation* points by $\mathbf{x} = [x_1, x_2, \dots, x_n, 1]^T$, $\mathbf{y} = [y_1, y_2, \dots, y_n, 1]^T \in \mathbb{P}^n$, respectively. The *identity relation* between the shapes is then as follows [18; 20; 21]

$$\mathbf{y} = \mathbf{A}\mathbf{x} \quad \Leftrightarrow \quad \mathbf{x} = \mathbf{A}^{-1}\mathbf{y}, \quad (2.1)$$

where \mathbf{A} is the unknown affine transformation that we want to recover [18; 20; 21]:

$$\mathbf{A} = \begin{bmatrix} a_{11} & \dots & a_{1n} & a_{1(n+1)} \\ \vdots & \ddots & \vdots & \vdots \\ a_{n1} & \dots & a_{nn} & a_{n(n+1)} \\ 0 & \dots & 0 & 1 \end{bmatrix}, \quad \text{and} \quad \mathbf{A}^{-1} = \begin{bmatrix} q_{11} & \dots & q_{1n} & q_{1(n+1)} \\ \vdots & \ddots & \vdots & \vdots \\ q_{n1} & \dots & q_{nn} & q_{n(n+1)} \\ 0 & \dots & 0 & 1 \end{bmatrix}.$$

Note that \mathbf{A}^{-1} always exists and it is also an affine transformation since \mathbf{A} is affine. Classical landmark-based approaches would now identify at least $n(n+1)/2$ point pairs $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{m \geq n(n+1)/2}$ then solve the system of linear equations represented by Eq. (2.1). However, here we are interested in a direct approach that does not involve solving the correspondence problem.

When we observe some image features (e.g. grey-levels of the pixels [42]) that are invariant under this transformation, we can define an additional relation [16; 17; 19]

$$f(\mathbf{x}) = g(\mathbf{A}\mathbf{x}) = g(\mathbf{y}), \quad (2.2)$$

where $f, g : \mathbb{P}^n \rightarrow \mathbb{R}$ are *covariant functions* under the transformation \mathbf{A} , defined on these observed features.

The above relations are still valid when a matrix function acts on both sides of Eq. (2.1) [20; 21; 42] (see Fig. 3.2) and Eq. (2.2) [16; 17; 19]. Indeed, for a properly chosen $\omega_p : \mathbb{P}^n \rightarrow \mathbb{R}$ and $\omega_c : \mathbb{R} \rightarrow \mathbb{R}$, we get

$$\omega_p(\mathbf{y}) = \omega_p(\mathbf{A}\mathbf{x}), \quad \text{and} \quad (2.3)$$

$$\omega_c(g(\mathbf{y})) = \omega_c(g(\mathbf{A}\mathbf{x})) = \omega_c(f(\mathbf{x})). \quad (2.4)$$

Starting from either Eq. (2.3) or Eq. (2.4), we can generate as many linearly independent equations as needed by making use of the nonlinear ω_p and ω_c functions, respectively. There is a fundamental difference between the above two equations, though. The nonlinear function ω_p acts directly on the point coordinates, and hence on the unknown

parameters of \mathbf{A} , and results in a *nonlinear* system of equations [20; 21]; in contrast ω_c acts on the *covariant functions* f and g that give rise to a *linear* system of equations [16; 17; 42]. Note that these equations do not contain any new information; they simply impose new linearly independent constraints required for a unique solution.

In addition, let concrete shapes be represented by their characteristic function $\mathbb{1} : \mathbb{P}^n \rightarrow \{0, 1\}$, where 0 and 1 correspond to the background and foreground, respectively. If we denote the *template* by $\mathbb{1}_t$ and the *observation* by $\mathbb{1}_o$, then Eq. (2.1) implies that

$$\mathbb{1}_t(\mathbf{x}) = \mathbb{1}_o(\mathbf{Ax}) = \mathbb{1}_o(\mathbf{y}) . \quad (2.5)$$

2.6 Registration vs. Matching

There is a fundamental difference between the problem of registration and *shape matching*. In either case, we fix the family of possible transformations. In the case of matching, we need to determine whether the two objects are from the same class or not. It is enough to ask whether there exists a transformation which aligns the objects (*i.e.* whether they are on the same orbit of the fixed transformation class), but the aligning transformation is not of interest. However, in the registration problem we always assume that there exists a transformation which aligns the objects and we need to estimate its parameters. This explains why the multiple object matching algorithm makes use of *invariants*, ignoring the effect of the unknown transformation, and why *covariance* is used to solve the registration problem.

2.7 Invariance vs. Covariance

Moment invariants [50; 104] have been extensively studied as they provide a powerful tool for shape matching. Essentially, invariants are functions immune to the action of a particular deformation. There is a well-established theory on affine invariants [28], but invariants of higher order deformations are hard to construct. Not long ago, important results on the existence of projective moment invariants [30], as well as on generalized invariants called *Implicit Moment Invariants* [27], were reported. However, we are not interested in constructing invariants as, being immune to deformations, they do not provide constraints on the actual transformation parameters. Instead, we shall use *covariant* functions that vary with the transformation, hence constraining its parameters. Indeed, *invariance* and *covariance* play a complementary role: While invariants identify a shape regardless of its deformation, covariants identify the actual deformation.

Chapter 3

Parametric Estimation of Affine Deformations of Binary Images: Polynomial Solution

In this chapter we focus on the estimation of affine transformations by aligning a known shape and its distorted *observation*. The classical way to solve this registration problem is to find correspondences between the shapes and then compute the transformation parameters from these established correspondences. Here, we propose a novel approach where the exact transformation is obtained as the solution of a polynomial system of equations. Our method has been tested on synthetic as well as on real images, and its robustness in the presence of segmentation errors and additive geometric noise was also demonstrated and analyzed. The method was successfully applied in the registration of hip prosthesis X-ray images as well as for real images. The advantage of our solution is that it has a linear time complexity, is fast, easy to implement, works without established correspondences and provides an exact solution regardless of the magnitude of the transformations.

3.1 Estimation of Affine Transformations

Here we will use the notations introduced in Section 2.5. We will start with the *identity relation* defined by Eq. (2.1). Classical landmark-based approaches would identify at least $n + 1$ point pairs $(\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{m \geq n+1})$ and solve the system of linear equations in Eq. (2.1). However, we are interested in a *direct solution* without solving the correspondence problem. For this reason, we will take the Lebesgue integral of both sides of the *identity relation* [20; 21]

$$\int_{\mathbb{P}^n} \mathbf{x} \, d\mathbf{x} = \frac{1}{|\mathbf{A}|} \int_{\mathbb{P}^n} \mathbf{A}^{-1} \mathbf{y} \, d\mathbf{y} , \quad (3.1)$$

where the integral transformation

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}, \quad d\mathbf{x} = |\mathbf{A}^{-1}|d\mathbf{y} = \frac{d\mathbf{y}}{|\mathbf{A}|}$$

has been applied. Although we write these integrals in \mathbb{P}^n , they are equivalent to the corresponding Lebesgue integrals in \mathbb{R}^n (*i.e.* integration is actually performed in the corresponding Cartesian coordinate system). This is because by using homogeneous coordinates, the real plane \mathbb{R}^n is mapped to the $w = 1$ plane in real projective space \mathbb{P}^n and affine transformations do not alter the homogeneous component w . One can therefore safely assume that it is always 1 and ignore it.

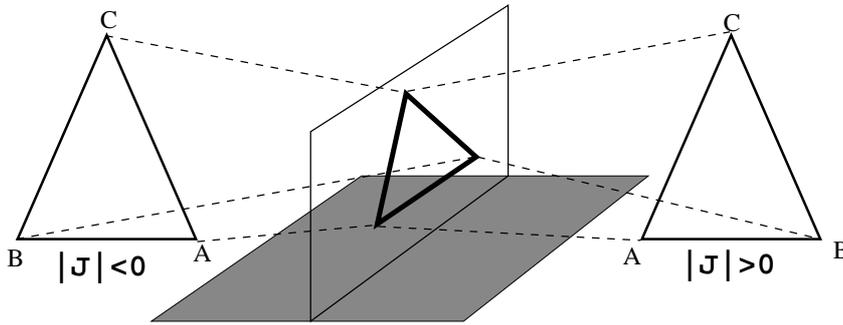


Figure 3.1: Sign ambiguity of the Jacobian determinant.

The determinant $|\mathbf{A}|$ is the Jacobian which corresponds to the measure of the transformation. Hence, it can be evaluated by integrating [20; 21]

$$\int_{\mathbb{P}^n} \mathbb{1}_t(\mathbf{x}) d\mathbf{x} = \frac{1}{|\mathbf{A}|} \int_{\mathbb{P}^n} \mathbb{1}_o(\mathbf{y}) d\mathbf{y} \quad \Leftrightarrow \quad |\mathbf{A}| = \frac{\int_{\mathbb{P}^n} \mathbb{1}_o(\mathbf{y}) d\mathbf{y}}{\int_{\mathbb{P}^n} \mathbb{1}_t(\mathbf{x}) d\mathbf{x}} .$$

Since the characteristic functions only take values from $\{0, 1\}$, we can further simplify the above integrals by making use of the relation [20; 21]:

$$\int_{\mathbb{P}^n} \mathbb{1}_t(\mathbf{x}) d\mathbf{x} \equiv \int_{\mathcal{F}_t} d\mathbf{x} ,$$

where the finite domain \mathcal{F}_t consists of the *template* foreground regions: $\mathcal{F}_t = \{\mathbf{x} \in \mathbb{P}^n | \mathbb{1}_t(\mathbf{x}) = 1\}$. Similarly, we can restrict the integral of $\mathbb{1}_o(\mathbf{y})$ to the *observation* foreground regions \mathcal{F}_o . Therefore evaluating the integrals yields the *area* of the foreground regions. From this point of view, the measure of the transformation $|\mathbf{A}|$ corresponds to the ratio of the area of the *observation* and *template* shapes [20; 21]

$$|\mathbf{A}| = \frac{\int_{\mathcal{F}_o} d\mathbf{y}}{\int_{\mathcal{F}_t} d\mathbf{x}} , \tag{3.2}$$

which can be directly computed from the input images. The sign ambiguity of the

determinant is also easily eliminated. A negative Jacobian would mean that the transformation is not orientation-preserving, *i.e.* the flipping of coordinates is allowed (see Fig. 3.1). In practice, however, physical constraints will usually prevent such a transformation, hence we can assume that $|\mathbf{A}|$ is always positive.

Now multiplying Eq. (3.1) by Eq. (2.5), we get a finite integral equation [20; 21]:

$$\begin{aligned} \int_{\mathbb{P}^n} \mathbf{x} \mathbb{1}_t(\mathbf{x}) \, d\mathbf{x} &= \frac{1}{|\mathbf{A}|} \int_{\mathbb{P}^n} \mathbf{A}^{-1} \mathbf{y} \mathbb{1}_o(\mathbf{y}) \, d\mathbf{y} \\ &\Updownarrow \\ \int_{\mathcal{F}_t} \mathbf{x} \, d\mathbf{x} &= \frac{1}{|\mathbf{A}|} \int_{\mathcal{F}_o} \mathbf{A}^{-1} \mathbf{y} \, d\mathbf{y} . \end{aligned} \quad (3.3)$$

This equation implies that the finite domains \mathcal{F}_t and \mathcal{F}_o are also related via $\mathcal{F}_o = \mathbf{A}\mathcal{F}_t$; *i.e.*, we match the shapes as a whole instead of point to point correspondences. In fact, Eq. (3.3) is a linear system of n equations for $k = 1, \dots, n$ [20; 21]

$$|\mathbf{A}| \int_{\mathcal{F}_t} x_k \, d\mathbf{x} = q_{k1} \int_{\mathcal{F}_o} y_1 \, d\mathbf{y} + \dots + q_{kn} \int_{\mathcal{F}_o} y_n \, d\mathbf{y} + q_{k(n+1)} \int_{\mathcal{F}_o} d\mathbf{y} .$$

It is clear that both sides of the equation as well as the Jacobian can be easily computed from the input shapes. Unfortunately, n equations alone are not enough to solve for $n(n+1)$ unknowns.

3.1.1 Construction of the Polynomial System

We will use Eq. (2.3) and get the following integral equation from Eq. (3.3) [20; 21]

$$\int_{\mathcal{F}_t} \omega(\mathbf{x}) \, d\mathbf{x} = \frac{1}{|\mathbf{A}|} \int_{\mathcal{F}_o} \omega(\mathbf{A}^{-1} \mathbf{y}) \, d\mathbf{y} . \quad (3.4)$$

The basic idea behind this approach is to generate enough linearly independent equations by making use of nonlinear ω functions. Note, however, that the generated equations contain no new information; they simply impose new linearly independent constraints. Indeed, from a geometric point of view, Eq. (3.3) simply matches the centre of mass of the *template* and *observation*, while the new equations of Eq. (3.4) match the centre of mass of the shapes obtained by the nonlinear ω transformations (see Fig. 3.2). Since the ω s are also applied to the unknowns, the resulting equations will be nonlinear. The simplest nonlinear system is a low order polynomial system, thus we need to find an ω such that Eq. (3.4) is a polynomial. The following proposition states that this can be achieved when ω is a polynome [20].



$$\omega(\mathbf{x}) = \mathbf{x} \quad \omega(\mathbf{x}) = [x_1^2, x_2^2, 1]^T \quad \omega(\mathbf{x}) = [x_1^3, x_2^3, 1]^T$$

Figure 3.2: The effect of the ω functions.

Proposition 3.1 *Let $\omega : \mathbb{P}^n \rightarrow \mathbb{P}^n$ and $\mathbf{x} \in \mathbb{P}^n$ ($n \in \mathbb{N}$). If the k^{th} coordinate of $\omega(\mathbf{x})$, denoted by $\omega^{(k)}(\mathbf{x}) = p_k$ is a real n -variate polynome, $1 \leq k \leq n$, then applying ω in Eq. (3.4) results in a polynomial system of equations of degree $\deg(p_k)$ at most.*

Proof

Let k be a fixed, arbitrary integer such that $1 \leq k \leq n$. We shall assume that $\omega^{(k)}(\mathbf{x})$ is polynomial, i.e. there exists an n -variate real polynome p_k with $\deg(p_k) \geq 1$, such that

$$\omega^{(k)}(\mathbf{x}) = p_k(x_1, \dots, x_n) = \sum_{i=1}^{u_k} s_i (\mathbf{A}_{1,:}^{-1} \mathbf{y})^{\alpha_{i1}} \dots (\mathbf{A}_{n,:}^{-1} \mathbf{y})^{\alpha_{in}} , \quad (3.5)$$

where $u_k = \binom{\deg(p_k)+n}{\deg(p_k)}$, and $\mathbf{A}_{j,:}^{-1}$ denotes the j^{th} row of \mathbf{A}^{-1} . One term of Eq. (3.5) can be expanded by making use of the *Multinomial theorem* [73]. For a given i and for all $1 \leq j \leq n$, we have

$$\begin{aligned} & (\mathbf{A}_{j,:}^{-1} \mathbf{y})^{\alpha_{ij}} = (q_{j1}y_1 + \dots + q_{jn}y_n + q_{j(n+1)})^{\alpha_{ij}} \\ &= \sum_{\substack{\beta_{ij1}, \dots, \beta_{ij(n+1)} \in \mathbb{N}_0 \\ \beta_{ij1} + \dots + \beta_{ij(n+1)} = \alpha_{ij}}} \frac{\alpha_{ij}!}{\beta_{ij1}! \dots \beta_{ij(n+1)}!} q_{j1}^{\beta_{ij1}} \dots q_{jn}^{\beta_{ijn}} q_{j(n+1)}^{\beta_{ij(n+1)}} y_1^{\beta_{ij1}} \dots y_n^{\beta_{ijn}} , \end{aligned}$$

hence we get an $(n + 1)$ -variate real polynome. In fact, we should compute the sum of the product of n pieces of the $(n + 1)$ - variate polynome in Eq. (3.5). Let $m = n(n + 1)$ and let us treat these products as an m -variate polynome. Moreover, the sum of m -variate polynoms is also an m -variate polynome. Integrating and noting the previous

result, we can rewrite Eq. (3.5) as

$$\int_{\mathcal{F}_o} \sum_{i=1}^{u_k} s_i (\mathbf{A}_{j,:}^{-1} \mathbf{y})^{\alpha_{i1}} \dots (\mathbf{A}_{n,:}^{-1} \mathbf{y})^{\alpha_{in}} d\mathbf{y} \equiv \int_{\mathcal{F}_o} \sum_{i=1}^{v_k} t_i q_1^{\gamma_{i1}} \dots q_m^{\gamma_{im}} y_1^{\delta_{i1}} \dots y_n^{\delta_{in}} d\mathbf{y},$$

where $v_k = \binom{\deg(p_k)+m}{\deg(p_k)}$. It is obvious from the above equation that the system of equations is of degree $\deg(p_k)$ at most. Furthermore, by making use of the basic properties of the Lebesgue integral, we get

$$\begin{aligned} \int_{\mathcal{F}_o} \sum_{i=1}^{v_k} t_i q_1^{\gamma_{i1}} \dots q_m^{\gamma_{im}} y_1^{\delta_{i1}} \dots y_n^{\delta_{in}} d\mathbf{y} &= \sum_{i=1}^{v_k} \int_{\mathcal{F}_o} t_i q_1^{\gamma_{i1}} \dots q_m^{\gamma_{im}} y_1^{\delta_{i1}} \dots y_n^{\delta_{in}} d\mathbf{y} \\ &= \sum_{i=1}^{v_k} t_i q_1^{\gamma_{i1}} \dots q_m^{\gamma_{im}} \int_{\mathcal{F}_o} y_1^{\delta_{i1}} \dots y_n^{\delta_{in}} d\mathbf{y} \\ &= \sum_{i=1}^{v_k} w_i q_1^{\gamma_{i1}} \dots q_m^{\gamma_{im}}. \end{aligned}$$

The last term is indeed a real polynome r_k with variables q_1, \dots, q_m yielding

$$\int_{\mathcal{F}_t} \omega^{(k)}(\mathbf{x}) d\mathbf{x} \equiv r_k(q_1, \dots, q_m).$$

Hence the system of equations is indeed polynomial, which completes the proof. \square

From now on we will consider the 2-dimensional case (*i.e.* $n = 2$), where $\mathbf{x} = [x_1, x_2, 1]^T$ and $\mathbf{y} = [y_1, y_2, 1]^T \in \mathbb{P}^2$ denote the homogeneous coordinates of the *template* and *observation* points, respectively (as stated in Section 2.5). It is thus apparent that the class of x^l ($l \in \mathbb{N}_0$) functions is an ideal choice for ω . Hence, we have the following polynomial equations for $k = 1, 2$ [20; 21]:

$$|\mathbf{A}| \int_{\mathcal{F}_t} x_k^l d\mathbf{x} = \sum_{i=1}^l \binom{l}{i} \sum_{j=0}^i \binom{i}{j} q_{k1}^{l-i} q_{k2}^{i-j} q_{k3}^j \int_{\mathcal{F}_o} y_1^{l-i} y_2^{i-j} d\mathbf{y} \quad l = 1, 2, 3. \quad (3.6)$$

This system of equations contains six polynomial equations of order three at most, which is enough to solve for all the unknowns. Actually, we have two separate systems for $k = 1, 2$, as shown in equations (3.7)–(3.9) [20; 21] below:

$$|\mathbf{A}| \int_{\mathcal{F}_t} x_k d\mathbf{x} = q_{k1} \int_{\mathcal{F}_o} y_1 d\mathbf{y} + q_{k2} \int_{\mathcal{F}_o} y_2 d\mathbf{y} + q_{k3} \int_{\mathcal{F}_o} d\mathbf{y}, \quad (3.7)$$

$$\begin{aligned} |\mathbf{A}| \int_{\mathcal{F}_t} x_k^2 d\mathbf{x} &= q_{k1}^2 \int_{\mathcal{F}_o} y_1^2 d\mathbf{y} + q_{k2}^2 \int_{\mathcal{F}_o} y_2^2 d\mathbf{y} + q_{k3}^2 \int_{\mathcal{F}_o} d\mathbf{y} + 2q_{k1}q_{k2} \int_{\mathcal{F}_o} y_1 y_2 d\mathbf{y} \\ &\quad + 2q_{k1}q_{k3} \int_{\mathcal{F}_o} y_1 d\mathbf{y} + 2q_{k2}q_{k3} \int_{\mathcal{F}_o} y_2 d\mathbf{y}, \end{aligned} \quad (3.8)$$

$$\begin{aligned}
 |\mathbf{A}| \int_{\mathcal{F}_t} x_k^3 \, d\mathbf{x} &= q_{k1}^3 \int_{\mathcal{F}_o} y_1^3 \, d\mathbf{y} + q_{k2}^3 \int_{\mathcal{F}_o} y_2^3 \, d\mathbf{y} + q_{k3}^3 \int_{\mathcal{F}_o} d\mathbf{y} + 3q_{k1}^2 q_{k2} \int_{\mathcal{F}_o} y_1^2 y_2 \, d\mathbf{y} \\
 &+ 3q_{k1}^2 q_{k3} \int_{\mathcal{F}_o} y_1^2 \, d\mathbf{y} + 3q_{k2}^2 q_{k3} \int_{\mathcal{F}_o} y_2^2 \, d\mathbf{y} + 3q_{k1} q_{k2}^2 \int_{\mathcal{F}_o} y_1 y_2^2 \, d\mathbf{y} \\
 &+ 3q_{k1} q_{k3}^2 \int_{\mathcal{F}_o} y_1 \, d\mathbf{y} + 3q_{k2} q_{k3}^2 \int_{\mathcal{F}_o} y_2 \, d\mathbf{y} \\
 &+ 6q_{k1} q_{k2} q_{k3} \int_{\mathcal{F}_o} y_1 y_2 \, d\mathbf{y} . \tag{3.9}
 \end{aligned}$$

3.1.2 Discussion

However, we may have several possible solutions for each unknown q_{ki} due to the cubic polynomial terms. Out of these potential solutions, we can select the right one by dropping the complex roots and choosing the transformation whose determinant matches the Jacobian computed via Eq. (3.2).

Note that an exact solution always exists whenever Eq. (2.1) is satisfied. In practice, however, a solution may not exist due to discretization errors or noise on the point coordinates. We can always check for the existence of a solution by computing the resultant of the system, which is a second order polynome. Still, the solution is not unique (but it exists!) when the shape is *affine symmetric*. An object is affine symmetric, or it has affine symmetry, when there exists an affine transformation that is invariant when applied on it, and it is not the identity transformation.

We note that this method could also be extended to higher dimensions at the price of adding higher order polynoms or restricting the space of admissible transformations. For example, in many medical applications 3-dimensional volume images are used. Hence, we should add a fourth-order equation and solve three systems. This is obviously more difficult and higher powers may compromise numerical stability. In [98], using the same idea as that presented here, a 3D binary registration method was given to find affine transformations by constructing and solving an overdetermined system of polynomial equations.

Another possibility is to use ω s of mixed coordinates (e.g. $\omega(\mathbf{x}) = [x_1 x_2, x_2, 1]^T$). This will generate new equations without increasing their order, but we lose the benefit of being able to handle the coordinates separately, i.e. we have to solve one large polynomial system of equations instead of three smaller ones. In order to retain this advantage (i.e. we have to solve the system of equations separately) we should only make use of univariate polynoms for ω s instead of mixed polynoms. Unfortunately, this makes this method numerically unfeasible, as there is no guarantee that we can apply a direct solver in a higher dimension to get a solution of the polynomial system of equations defined by Eq. (3.4).

Although the derivations are not moment-based *per se*, it is interesting to see how the resulting Eq. (3.6) is related to moments. *Image moments* and invariants [31] were

first introduced by Hu [50] for 2D pattern analysis. Since then, they have become one of the most popular region-based descriptors because any shape can be reconstructed from an infinite set of moments [34]. Traditional two dimensional $(i + j)^{\text{th}}$ order moments of a function $\varrho : \mathbb{R}^2 \rightarrow \mathbb{R}$ are defined as [97]

$$m_{ij}(\varrho) = \int_{\mathbb{R}^2} \varrho(\mathbf{x}) x_1^i x_2^j \, d\mathbf{x} ,$$

where $i, j \in \mathbb{N}_0$. When ϱ is an image function, then these moments are also referred to as image moments. In the binary case, where objects are represented by their silhouette, ϱ becomes a characteristic function producing [97]

$$m_{ij}(\mathcal{F}) = \int_{\mathcal{F}} x_1^i x_2^j \, d\mathbf{x} \quad (3.10)$$

with $\mathcal{F} = \{\mathbf{x} \in \mathbb{R}^2 : \varrho(\mathbf{x}) = 1\}$. This is often called the *shape* or *geometric* moment as it only uses polynomials of the coordinates. Generally, orthogonal moments, such as Legendre [34] or Zernike moments [99], are numerically more stable than regular moments. We should remark, however, that orthogonal moments can be expressed by regular moments. Similar to Legendre or Zernike moments, our functions could also be expressed in terms of *shape moments* whenever the ω functions adopted are polynomials. In this sense, our functions could be considered as shape moments. When ω is not a polynome, then its Taylor expansion results in an approximating polynome, which in turn yields an *infinite* sum of shape moments. Independently of the choice of ω , it can only be expressed in terms of shape moments by expanding it as a Taylor series. This result corresponds to similar findings reported in [30; 104] in the context of projective invariants. What we propose in this chapter is another approach, which, starting from the *identity relation* in Eq. (2.1), an elegant framework is constructed to generate an arbitrary set of linearly independent equations.

3.2 Numerical Implementation

Although we have constructed our equations in the continuum, in practice we only have a limited precision digital image. This means that the integrals, which are in fact the coefficients of the unknowns in equations (3.7)–(3.9), can only be *approximated* by a discrete sum over the foreground pixels and this introduces an inherent, although negligible error into our computation. In [60; 61], the effect of such errors on moment computation was analyzed and a number-theoretical estimation of quantization errors was derived. The continuous domains \mathcal{F}_t and \mathcal{F}_o are represented as finite sets of

foreground pixels [20]

$$\mathcal{F}_t \approx F_t = \{\mathbf{p}_i\}_{i=1}^M \quad \text{and} \quad \mathcal{F}_o \approx F_o = \{\mathbf{p}'_i\}_{i=1}^M,$$

where M and N are the number of foreground pixels of the *template* and *observation*, respectively. Next, the integrals are *approximated* by a finite sum over these foreground pixels [20]

$$\int_{\mathcal{F}_t} x_k \, d\mathbf{x} \approx \sum_{i=1}^M \mathbf{p}_i^{(k)} \quad \text{and} \quad \int_{\mathcal{F}_o} y_k \, d\mathbf{y} \approx \sum_{i=1}^N \mathbf{p}'_i^{(k)}, \quad k = 1, 2, \quad (3.11)$$

where $\mathbf{p}_i = [\mathbf{p}_i^{(1)}, \mathbf{p}_i^{(2)}]$ and $\mathbf{p}'_i = [\mathbf{p}'_i^{(1)}, \mathbf{p}'_i^{(2)}]$ stand for the i^{th} foreground pixel of the *template* and *observation*, respectively. The Jacobian in Eq. (3.2) reduces to [20]

$$|\mathbf{A}| = \frac{N}{M}. \quad (3.12)$$

Naturally, the resolution of the images affects the precision of these approximations. As the mesh size tends to zero, the finite sums better approximate the integrals. Therefore, our method performs better on higher resolution images. Experiments show that images with size 500×500 provide sufficiently accurate approximations.

Algorithm 1 summarizes the steps of our registration method [20]. It is clear that the solution is obtained in a single pass without any loop or optimization. Even though we have to solve a polynomial system, the complexity of this step is constant and, most importantly, independent of the image size. Matlab is quite efficient at solving our system, but other packages are also available like *PHCpack* [106–108].

Algorithm 1: Pseudo-code of binary image registration using a polynomial system of equations.

Input : *Template* ($\mathbb{1}_t$) and *observation* ($\mathbb{1}_o$) shapes as binary images

Output: Estimated affine transformation $\tilde{\mathbf{A}}$

- 1 Estimate the Jacobian $|\mathbf{A}|$ using Eq. (3.12).
 - 2 Evaluate the integrals in equations (3.7)–(3.9) to get the coefficients of the unknowns.
 - 3 Solve the system of equations (3.7)–(3.9) using a standard solver (e.g. Matlab).
 - 4 Choose the correct transformation based on the Jacobian to get $\tilde{\mathbf{A}}^{-1}$.
-

The images need to be scanned only once, and the integrals in equations (3.7)–(3.9) as well as the Jacobian can be evaluated during this scan. This step takes $c_1(M + N)$ time, where $M + N$ is the size of the input images. Once the system has been constructed, the rest of the algorithm runs in constant (c_2) time, independently of the input size. Thus the overall time complexity of the method is $c_1(M + N) + c_2$,

i.e. $\mathcal{O}(M + N)$. However, our experiments tell us that $c_1(M + N) \ll c_2$, in practice, hence the actual running time is dominated by the constant c_2 (see Table 3.3).

3.2.1 Fuzzy Object Representation

There are many ways of discretizing a continuous-valued function. The easiest way to make a discrete image is by sampling the continuous function at uniform grid positions. This approach, known as Gaussian centre point digitization, could be applied in the above problem. An alternative is to perform a fuzzy discretization of the image [93; 97].

A discrete fuzzy subset [93; 97] F of a *reference set* $X \subset \mathbb{Z}^2$ is a set of ordered pairs

$$F = \{((i, j), \mu_F(i, j)) \mid (i, j) \in X\},$$

where $\mu_F : X \rightarrow [0, 1]$ is the *membership function* of F in X . The fuzzy membership function may be defined in various ways; its values reflect the levels of membership of the pixels to the object. One useful way to define the membership function on a reference set in the case where it is an image plane is to assign a value to each image element (pixel) that is proportional to its coverage by the imaged object. In this way, partial memberships (values strictly between 0 and 1) are assigned to the pixels on the boundary of the discrete object.

In the discrete formulation the geometric moments of order $i + j$ of a discrete fuzzy set F can be used, which are defined by [92]

$$m_{i,j}(F) = \sum_{\mathbf{p} \in X} \mu_F(\mathbf{p}) p_1^i p_2^j, \quad (3.13)$$

where $\mathbf{p} = [p_1, p_2]^T$. This equation can be used to estimate the geometric moments of a continuous 2D shape. Asymptotic error bounds for moments of order 2 at most, derived in [92], show that moment estimates calculated from a fuzzy object representation can provide a considerable increase in precision compared to estimates computed from a crisp representation, for the same spatial resolution.

If F is a fuzzy representation of \mathcal{F} , it follows that $m_{i,j}(\mathcal{F}) \approx m_{i,j}(F)$. Thus, similar to Eq. (3.11), by using Eq. (3.10) and Eq. (3.13) the integrals in Eq. (3.6) can be approximated by [97]

$$\int_{\mathcal{F}_t} x_k^l dx \approx \sum_{\mathbf{p} \in X_t} \mu_{F_t}(\mathbf{p}) p_k^l \quad \text{and} \quad \int_{\mathcal{F}_o} y_1^{l-i} y_2^{i-j} dy \approx \sum_{\mathbf{p}' \in X_o} \mu_{F_o}(\mathbf{p}') p_1'^{l-i} p_2'^{i-j},$$

where X_t and X_o are the reference sets (discrete domains) of the (fuzzy) *template* and (fuzzy) *observation* image, respectively. As in Eq. (3.12), the Jacobian can be

approximated by [97]

$$|\mathbf{A}| = \frac{m_{00}(\mathcal{F}_o)}{m_{00}(\mathcal{F}_t)} \approx \frac{m_{00}(F_o)}{m_{00}(F_t)} = \frac{\sum_{\mathbf{p}' \in X_o} \mu_{F_o}(\mathbf{p}')}{\sum_{\mathbf{p} \in X_t} \mu_{F_t}(\mathbf{p})}.$$

The approximating discrete system of polynomial equations can now be produced by inserting these approximations into Eq. (3.6) [97]:

$$|\mathbf{A}| \sum_{\mathbf{p} \in X_t} \mu_{F_t}(\mathbf{p}) p_k^l = \sum_{i=0}^l \binom{l}{i} \sum_{j=0}^i \binom{i}{j} q_{k1}^{l-i} q_{k2}^{i-j} q_{k3}^j \sum_{\mathbf{p}' \in X_o} \mu_{F_o}(\mathbf{p}') p_1'^{l-i} p_2'^{i-j}.$$

Obviously, the spatial resolution of the images affects the precision of this approximation. However, sufficient spatial resolution may be unavailable in real applications or, as expected in the case of 3D applications, it may lead to overly large amounts of data that cannot be successfully processed. Nevertheless, it was shown in [92] that increasing the number of grey levels representing pixel coverage by a factor n^2 provides asymptotically the same increase in precision as an n times increase of spatial resolution. Therefore the suggested approach, utilizing increased membership resolution, is a very powerful way of compensating for an insufficient spatial resolution, while still preserving the desired precision of moment estimates.

It was observed that the experimental data also confirm the theoretical results [97], *i.e.* that the use of fuzzy shape representations enhances the registration process compared to the binary case. This effect can be viewed as the fuzzy representation “increasing” the resolution of the object around its border. It also implies that registration based on a fuzzy border representation may work for lower image resolutions too, when the binary approach becomes unstable.

3.3 Estimation in the Presence of Noise

There are two types of noise which can affect a binary image. One is “radiometric”, *i.e.* pixels may randomly take a *foreground* or *background* colour. This is typically present in the form of a *salt and pepper* noise, which can be efficiently removed by appropriate morphological filtering. The second type manifests itself as an uncertainty in the point coordinates, *i.e.* it is a type of *geometric* noise. More precisely, the observed point coordinates may differ from the true coordinates by a random distance, as shown in Fig. 3.3. This will definitely undermine the identity relation in Eq. (2.1) and introduce an error into our system. In the following section, we will analyze this case and compute the average error caused by *geometric* noise on the *observation*.

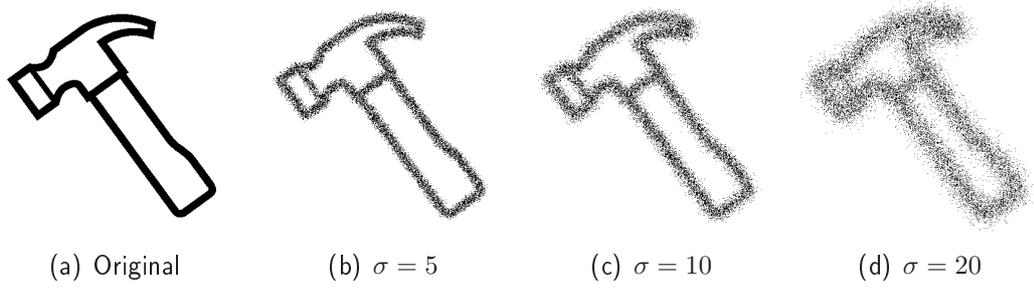


Figure 3.3: Observations corrupted by additive Gaussian noise on the coordinates.

3.3.1 Geometric Noise

It is reasonable to assume that the observed point coordinates are around the true ones, hence we will consider an i.i.d. additive Gaussian noise model on the *observation* coordinates. The *identity relation* Eq. (2.1) thus becomes [20]

$$\mathbf{y}^* = \mathbf{y} + \varepsilon(\mathbf{y}) = \mathbf{A}\mathbf{x} + \varepsilon^*(\mathbf{y}^*) \quad \Leftrightarrow \quad \mathbf{x} = \mathbf{A}^{-1}(\mathbf{y}^* - \varepsilon^*(\mathbf{y}^*)) ,$$

where $\varepsilon(\mathbf{y}) \equiv \varepsilon^*(\mathbf{y}^*) = [\varepsilon_1^*(\mathbf{y}^*), \varepsilon_2^*(\mathbf{y}^*), 0]^T$ is the noise function which gives a random translation at each point $\mathbf{y}^* = [y_1^*, y_2^*, 1]^T$. We shall assume that ε_1^* and ε_2^* are independent and normally distributed with 0 means and variances σ_1 and σ_2 , respectively. Actually, $\varepsilon^*(\mathbf{y}^*)$ is a two dimensional sample from, 0 the mean Gaussian distribution with a diagonal covariance matrix $\text{diag}(\sigma_1, \sigma_2)$. Thus Eq. (3.3) becomes [20]

$$\int_{\mathcal{F}_t} \omega(\mathbf{x}) \, d\mathbf{x} = \frac{1}{|\mathbf{A}|} \int_{\mathcal{F}_o^*} \omega(\mathbf{A}^{-1}(\mathbf{y}^* - \varepsilon^*(\mathbf{y}^*))) J(\mathbf{y}^*) \, d\mathbf{y}^* ,$$

where $\mathcal{F}_o^* = \{\mathbf{y}^* \in \mathbb{P}^2 | \mathbb{1}_o(\mathbf{y}^*) = 1\}$ and the integral transformation

$$\mathbf{x} = \mathbf{A}^{-1}(\mathbf{y}^* - \varepsilon^*(\mathbf{y}^*)) , \quad d\mathbf{x} = J(\mathbf{y}^*) d\mathbf{y}^* / |\mathbf{A}|$$

has been applied. From here on we will omit the integration domains unless they are ambiguous. In fact, the Jacobian becomes $J(\mathbf{y}^*)/|\mathbf{A}|$ in the noisy case with [20]

$$J(\mathbf{y}^*) = (1 - \varepsilon_1^*(\mathbf{y}^*)'_{y_1^*} - \varepsilon_2^*(\mathbf{y}^*)'_{y_2^*} + |\nabla \varepsilon^*(\mathbf{y}^*)|)$$

depending on the actual noise. Here $\varepsilon_1^*(\mathbf{y}^*)'_{y_2^*}$ denotes the partial derivative of the second variable (y_2^*) and [20]

$$|\nabla \varepsilon^*(\mathbf{y}^*)| = \left| \begin{array}{cc} \varepsilon_1^*(\mathbf{y}^*)'_{y_1^*} & \varepsilon_1^*(\mathbf{y}^*)'_{y_2^*} \\ \varepsilon_2^*(\mathbf{y}^*)'_{y_1^*} & \varepsilon_2^*(\mathbf{y}^*)'_{y_2^*} \end{array} \right| .$$

It is clear that the integrals of ε_i^* give the expected value, which is always 0 in our case. Furthermore, since in practice we always work with discrete pixel coordinates, the partial derivatives of ε^* can be approximated via finite differences, e.g. by central differences with $h \in \mathbb{N}$ [20]

$$\begin{aligned}\varepsilon_1^*(\mathbf{y}^*)'_{y_1^*} &\approx \frac{1}{2h} \left(\varepsilon^*([y_1^* + h, y_2^*, 1]^T) - \varepsilon^*([y_1^* - h, y_2^*, 1]^T) \right), \\ \varepsilon_1^*(\mathbf{y}^*)'_{y_2^*} &\approx \frac{1}{2h} \left(\varepsilon^*([y_1^*, y_2^* + h, 1]^T) - \varepsilon^*([y_1^*, y_2^* - h, 1]^T) \right).\end{aligned}$$

Therefore the integral of these derivatives will also be approximated by the integral of the finite differences, which thus equals 0.

Computation of $|\mathbf{A}|$

Since the true coordinates \mathbf{y} are unknown on a noisy *observation*, we have to integrate using $\mathbf{y} = \mathbf{y}^* - \varepsilon^*(\mathbf{y}^*)$. Thus the numerator of Eq. (3.2) becomes [20]

$$\int_{\mathcal{F}_o} d\mathbf{y} = \int_{\mathcal{F}_o^*} (1 - |\nabla \varepsilon^*(\mathbf{y}^*)|) d\mathbf{y}^* = \int_{\mathcal{F}_o^*} d\mathbf{y}^*,$$

because $\int_{\mathcal{F}_o^*} |\nabla \varepsilon^*(\mathbf{y}^*)| d\mathbf{y}^* = 0$ according to our previous deductions. Therefore Eq. (3.2) remains valid in the noisy case.

$$\omega(\mathbf{x}) = \mathbf{x}$$

The right hand side of Eq. (3.7) has to be evaluated on the noisy *observation*. Thus denoting $\mathbf{v} := \mathbf{A}^{-1}(\mathbf{y}^* - \varepsilon^*(\mathbf{y}^*))$, i.e.

$$v_k = q_{k1}(y_1^* - \varepsilon_1^*(\mathbf{y}^*)) + q_{k2}(y_2^* - \varepsilon_2^*(\mathbf{y}^*)) + q_{k3};$$

we get for $k = 1, 2$ [20]

$$\begin{aligned}\int_{\mathcal{F}_o} \mathbf{A}_{k,:}^{-1} \mathbf{y} d\mathbf{y} &= \int_{\mathcal{F}_o^*} v_k d\mathbf{y}^* - \int_{\mathcal{F}_o^*} v_k \varepsilon_1^*(\mathbf{y}^*)'_{y_1^*} d\mathbf{y}^* \\ &\quad - \int_{\mathcal{F}_o^*} v_k \varepsilon_2^*(\mathbf{y}^*)'_{y_2^*} d\mathbf{y}^* + \int_{\mathcal{F}_o^*} v_k |\nabla \varepsilon^*(\mathbf{y}^*)| d\mathbf{y}^*.\end{aligned}$$

It is not hard to see that every term, except the first one, equals 0. For example [20]

$$\int_{\mathcal{F}_o^*} v_k \varepsilon_1^*(\mathbf{y}^*)'_{y_1^*} d\mathbf{y}^* = \int_{\mathcal{F}_o^*} \left(q_{k1}(y_1^* - \varepsilon_1^*(\mathbf{y}^*)) + q_{k2}(y_2^* - \varepsilon_2^*(\mathbf{y}^*)) + q_{k3} \right) \varepsilon_1^*(\mathbf{y}^*)'_{y_1^*} d\mathbf{y}^*.$$

Evaluating the coefficient of q_{k1} yields [20]

$$\int_{\mathcal{F}_o^*} (y_1^* - \varepsilon_1^*(\mathbf{y}^*)) \varepsilon_1^*(\mathbf{y}^*)'_{y_1^*} d\mathbf{y}^* = \int_{\mathcal{F}_o^*} y_1^* \varepsilon_1^*(\mathbf{y}^*)'_{y_1^*} d\mathbf{y}^* - \int_{\mathcal{F}_o^*} \varepsilon_1^*(\mathbf{y}^*) \varepsilon_1^*(\mathbf{y}^*)'_{y_1^*} d\mathbf{y}^* .$$

Using finite differences for $\varepsilon_1^*(\mathbf{y}^*)'_{y_1^*}$ and iterating the integrals, we get for the first term [20]

$$\begin{aligned} \frac{1}{2h} \int y_1^* \left(\int \varepsilon_1^*([y_1^* + h, y_2^*, 1]^T) dy_2^* \right) dy_1^* - \frac{1}{2h} \int y_1^* \left(\int \varepsilon_1^*([y_1^* - h, y_2^*, 1]^T) dy_2^* \right) dy_1^* \\ = \frac{1}{2h} \int y_1^* 0 dy_1^* - \frac{1}{2h} \int y_1^* 0 dy_1^* = 0 , \end{aligned}$$

since ε_1 is a 0-mean Gaussian for which $\int \varepsilon_1 = 0$. The remaining terms can be evaluated in a similar way. Next, the right hand side of Eq. (3.7) in the case of a noisy *observation* is [20]:

$$\int_{\mathcal{F}_o^*} v_k d\mathbf{y}^* = q_{k1} \int_{\mathcal{F}_o^*} y_1^* d\mathbf{y}^* + q_{k2} \int_{\mathcal{F}_o^*} y_2^* d\mathbf{y}^* + q_{k3} \int_{\mathcal{F}_o^*} d\mathbf{y}^* .$$

Hence Eq. (3.7) remains valid. This is not surprising as the equation matches the centre of mass of the *template* and *observation* shapes. When the *observation* coordinates are corrupted by a 0-mean additive noise factor, the centre of mass does not change as the noise components integrate to 0.

$$\omega(\mathbf{x}) = [x_1^2, x_2^2, 1]^T$$

Using the noisy *observation*, Eq. (3.8) becomes [20]

$$|\mathbf{A}| \int_{\mathcal{F}_t} x_k^2 d\mathbf{x} = \int_{\mathcal{F}_o^*} \left(\mathbf{A}_k^{-1}(\mathbf{y}^* - \varepsilon^*(\mathbf{y}^*)) \right)^2 d\mathbf{y}^* .$$

It is apparent that both first and second order noise statistics will appear in the above equation. While first order statistics vanish, second-order moments ($\int \varepsilon_k^*(\mathbf{y}^*)^2 = \sigma_k^2, k = 1, 2$) will affect the equation. It is straightforward to show, using similar considerations as in Section 3.3.1, that in the noisy case Eq. (3.8) becomes [20]

$$|\mathbf{A}| \int_{\mathcal{F}_t} x_k^2 d\mathbf{x} = \int_{\mathcal{F}_o} (\mathbf{A}_k^{-1}(\mathbf{y}))^2 d\mathbf{y} + q_{k1}^2 \sigma_1^2 + q_{k2}^2 \sigma_2^2 .$$

Thus the error introduced by the noisy *observation* depends on the noise variances σ_1^2 and σ_2^2 , as well as on the strength of the unknown transformation \mathbf{A} .

$$\omega(\mathbf{x}) = [x_1^3, x_2^3, 1]^T$$

In this case, the system will include third order noise statistics. Fortunately, any symmetric distribution will have a third central moment of zero, thus using similar considerations as before, the noisy Eq. (3.9) is [20]

$$|\mathbf{A}| \int_{\mathcal{F}_t} x_k^3 \, d\mathbf{x} = \int_{\mathcal{F}_o} (\mathbf{A}_k^{-1}(\mathbf{y}))^3 \, d\mathbf{y} + 3q_{k1}^2 q_{k3} \sigma_1^2 + 3q_{k2}^2 q_{k3} \sigma_2^2 .$$

Summary

Overall, we can say that the error caused by i.i.d. additive Gaussian noise on the point coordinates of the *observation* is as follows [20]:

Equation	Error term
$ \mathbf{A} $: Eq. (3.2)	0
$\omega(\mathbf{x}) = \mathbf{x}$: Eq. (3.7)	0
$\omega(\mathbf{x}) = [x_1^2, x_2^2, 1]^T$: Eq. (3.8)	$q_{k1}^2 \sigma_1^2 + q_{k2}^2 \sigma_2^2$
$\omega(\mathbf{x}) = [x_1^3, x_2^3, 1]^T$: Eq. (3.9)	$3q_{k3}(q_{k1}^2 \sigma_1^2 + q_{k2}^2 \sigma_2^2)$

An experimental analysis of the noisy case is presented in Section 3.4.2. Our findings suggest that the proposed algorithm can cope with noisy observations as high as $\sigma_1 = \sigma_2 = 10$ noise levels.

3.4 Experimental Results

The proposed algorithm was tested on a large database of binary images of size 1000×1000 . The dataset consisted of 56 different shapes and their transformed versions, giving a total of over 50 000 images. The affine transformations we applied were randomly composed of $0^\circ, 10^\circ, \dots, 350^\circ$ rotations; $0, 0.4, \dots, 1.2$ shearings; $0.5, 0.7, \dots, 1.9$ scalings, and $-20, 0, 20$ translations along both axes. Some typical examples of these images can be seen in Fig. 3.4. The original shape of each was then used as the *template* image and the transformed images were used as the *observation*. The proposed algorithm was implemented in Matlab 7.2 and ran on a Pentium IV 3.2 GHz under the Linux operating system. The average runtime was around 1 second including the computation of the integrals and the solving of the polynomial system.

In order to quantitatively evaluate the registration results, we defined two kinds of error measures. The first one (denoted by ϵ) measures the distance between the true \mathbf{A} and the estimated $\tilde{\mathbf{A}}$ transformation obtained by our algorithm, based on all the *template pixels* \mathbf{p} . Intuitively, ϵ tells us the average transformation error per pixel. The second measures the absolute difference (denoted by δ) between the *observation* and

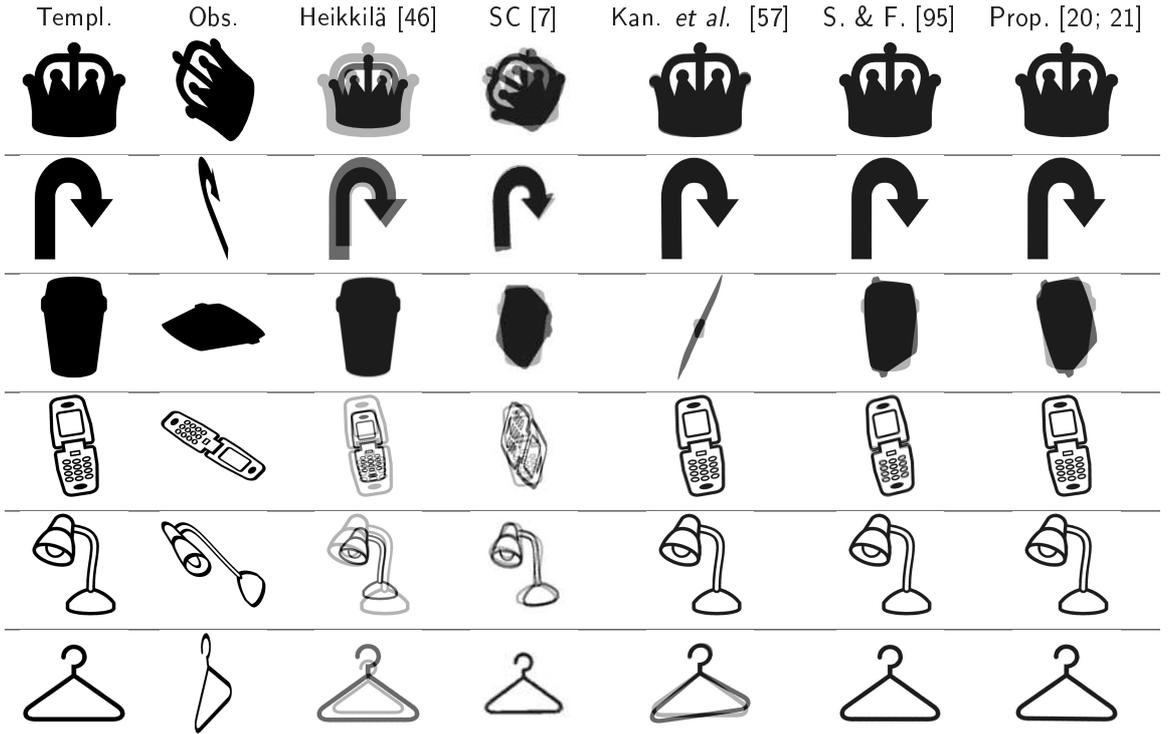


Figure 3.4: Registration results for synthetic image pairs. The first two columns show the *template* and its affine distorted *observation* to be matched, while the other columns contain the registration result of each given method. The *template* and its *registered observation* are overlaid in such a way that overlapping pixels are depicted in black, while non-overlapping ones are shown in light or dark grey, respectively.

the *registered* image [20].

$$\epsilon = \frac{1}{|F_t|} \sum_{\mathbf{p} \in F_t} \|(\mathbf{A} - \hat{\mathbf{A}})\mathbf{p}\|, \quad \text{and} \quad \delta = \frac{|F_r \Delta F_o|}{|F_r| + |F_o|} \cdot 100\%, \quad (3.14)$$

where Δ means the symmetric difference, while F_r denotes the set of pixels of the *registered* shape. Note that ϵ can only be used when the true transformation \mathbf{A} is also known, while δ can always be computed. But, ϵ gives a better characterization of the transformation error as it directly evaluates the mistransformation. δ only tells us the percentage of non-overlapping area between the *observation* and *registered* shape. Hence the value of δ also depends on the compactness and topology of the shapes. The performance of our algorithm on the benchmark dataset was evaluated based on these two measures. A summary of these results is given in Table 3.1 Another important indicator is the number of test cases where the algorithm finds no solution. Although all the applied transformations were affine, it is possible that an *observation* is not on the orbit of its *template*. The reason is mainly due to the high numerical error in the coefficients caused by a combination of strong deformation error and discretization error. The compactness of the shapes also affects the amount of discretization error.

Table 3.1: Registration results of the proposed method on the benchmark dataset containing 49 282 synthetic observations of 56 shapes. There was no solution in 5.47% of the test cases.

	ϵ (pixel)	δ (%)	CPU time (sec.)
Median	0.51	0.15	0.98
Mean	36.98	3.36	0.94
Variance	154.18	12.55	0.2

Table 3.2: Registration results (medians of error measures) of the proposed method on 32 724 filled shapes and 16 558 line drawings.

	ϵ (pixel)	δ (%)	CPU time (sec.)	Unsolved (%)
Filled	0.49	0.06	1	4.03
Line drawings	0.55	0.63	0.95	8.32

Obviously, such errors arise only around the boundaries so shapes with a longer contour will have a higher discretization error. Our database contains 32 724 filled shapes and 16 558 line drawings. Table 3.2 shows a comparison of the registration quality with these shapes. It is clear that the registration of line drawings (*i.e.* shapes with longer contours) is slightly less accurate. The resolution of the images is also important. As already mentioned in Section 3.2, the discrete sums will better approximate the integrals at higher resolutions. Fortunately, the time complexity of our method is linear, hence increasing the resolution will not increase the CPU time much. Table 3.3 shows the error and CPU time as a function of the resolution.

Table 3.3: Median of error measures versus resolution of the *observation*.

	ϵ (pixel)	δ (%)	CPU time (sec.)
100%	0.71	0.09	1.04
50%	14.68	0.74	0.9
25%	21.54	0.81	0.85
12.5%	25.39	4.91	0.81

3.4.1 Comparison with Previous Approaches

Now we will compare the registration results obtained by the proposed method with some of the most important binary registration approaches. Where an implementation was available, we evaluated quantitatively the performance of our algorithm with respect to these methods. For this purpose, we used 1686 images randomly chosen from our database. We ran the demo software [6] provided by the authors on our dataset. Although the method uses regularized thin-plate splines as a flexible class

of transformations, it is possible to set the regularization parameter to get affine behaviour: $\beta_{init}=500$. Other parameters were also set empirically to their optimal value (number of iterations $n_{iter}=15$; annealing rate $r=5$). Due to its high complexity, we had to reduce the size of the images by a factor of 0.2. In spite of this, as can be seen in Table 3.4, the CPU time was still over 20 seconds. We obtained the Matlab implementation from the authors of [57], [95], and [46] and conducted a comparative test. The results presented in Table 3.4 and Fig. 3.4 show that our method outperforms these approaches in terms of both quality and CPU time. In the case of that described in [95], the registration quality is almost the same, so it is fair to say that both methods give accurate registrations but our algorithm runs faster. Yet, our method is clearly better than the method described in [95] in terms of robustness, as it will transpire in the next section.

Table 3.4: Median of error measures on 1686 randomly selected images using some related approaches and the proposed algorithm.

	ϵ (pixel)	δ (%)	CPU time (sec.)
Heikkilä [46]	86.35	39.03	1.15
Shape context [7]	–	27.17	24.79
Kan. <i>et al.</i> [57]	8.89	9.7	32.45
S. & F. [95]	0.51	0.19	5.62
Proposed [20; 21]	0.5	0.15	0.93

3.4.2 Robustness

In Section 3.3, we calculated the errors caused by noisy observations in our polynomial system equations (3.7)–(3.9). Now we will experimentally test the robustness of the proposed method against i.i.d. Gaussian geometric noise. To do this, we used 1377 randomly selected images taken from our benchmark database. Here i.i.d. Gaussian noise with $\sigma = 1, 2, 5, 10, 15, 20$ was added to the extracted pixel coordinates of each *observation* and the registration algorithm was used as input for the point list of this noisy *observation* and the original *template*. To evaluate registration quality, the recovered transformation was applied to the original (*i.e.* noiseless) *observation* and the usual error measures ϵ and δ were computed. It is apparent from Table 3.5 that the proposed algorithm provides good solutions for noise as high as $\sigma = 10$. We also

Table 3.5: Median of error measures versus σ of the noise on 1377 randomly selected images.

σ	1	2	5	10	15	20
ϵ (pixel)	0.51	0.53	0.73	2.42	5.86	11.91
δ (%)	0.19	0.27	0.79	2.72	6.24	10.64

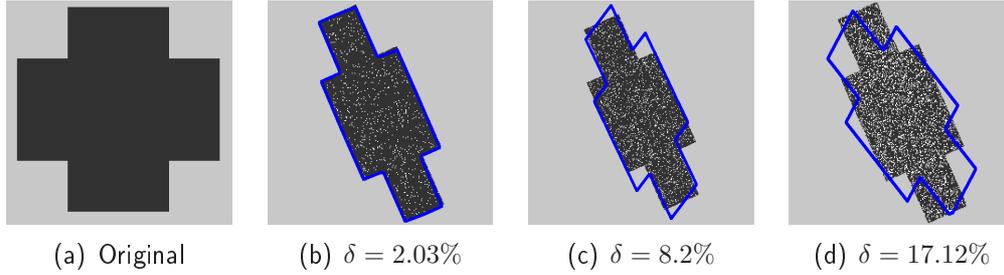


Figure 3.5: The robustness of the proposed polynomial approach in the case of incomplete objects was evaluated on images where we randomly removed 5%, 10%, and 20% of the foreground pixels ((b)-(d)) of the original image (a). The registration results are shown as overlaid contours of the *registered* shape.

examined the robustness of the proposed approach in the case of incomplete objects. For this purpose, we used 1289 randomly selected images taken from our benchmark database where we randomly removed 5%, 10%, 15%, and 20% of the foreground pixels before registration (see Fig. 3.5). Needless to say, such incomplete observations will introduce errors in the original system of equations (3.7)–(3.9). Table 3.6 shows that our method is quite robust, while the error rate of other state-of-the-art methods increases considerably even for missing pixels as low as 5%.

Table 3.6: Median of error measures versus the ratio of removed foreground pixels on 1289 randomly selected images.

	Heikkilä [46]		Kan. <i>et al.</i> [57]		S. & F. [95]		Proposed [20; 21]	
	ϵ	δ	ϵ	δ	ϵ	δ	ϵ	δ
5%	76.29	40.5	8.69	9.42	8.09	4.44	1.95	1.93
10%	75	40.64	12.07	10.82	16.12	8.78	3.56	3.78
15%	71.86	41.05	13.55	12.54	24.73	12.98	9.25	10.96
20%	70.38	39.63	14.42	14.41	33.09	17.7	8.86	10.9

However, we should add that all of these methods are less robust against the same amount of occlusion (*i.e.* when missing pixels are not uniformly distributed over the whole region). In Table 3.7, we list the results on 794 randomly selected images with occlusions of size 2.5%, 5% and 10% of the input shape area. Of course, even relatively small occlusions yield a rather high error rate for both the proposed method and other

Table 3.7: Median of error measures versus the size of occlusion on 794 randomly selected images.

	Heikkilä [46]		Kan. <i>et al.</i> [57]		S. & F. [95]		Proposed [20; 21]	
	ϵ	δ	ϵ	δ	ϵ	δ	ϵ	δ
2.5%	120.36	53.77	76.49	55.77	42.61	23.91	47.44	38.12
5%	153.38	55.65	137.34	68.68	91.88	37.16	162.34	51.1
10%	215.81	60.98	227.63	79.75	175.32	47.65	251.4	58.67

state-of-the-art methods. This is because they rely on quantities obtained by integrating over the whole object area. Thus large missing parts might drastically change these quantities and result in false registrations. Nevertheless, in many application areas, images can be taken under controlled conditions which guarantee that observations are not occluded (e.g. medical imaging and industrial inspection).

3.4.3 Real Images

The performance of our method was also evaluated on real images. Fig. 3.6 shows some examples of these images with overlaid contours. For segmentation, we used classical thresholding as well as active contours [10]. The main challenges are the segmentation errors (e.g. see the fifth image in Fig. 3.6) and a slight projective distortion between the image pairs. Overall, when reasonably good segmentations are available and the true transformation is sufficiently close to an affine one, then our method performs quite well, as shown by the δ error values and difference images displayed for each image pair.

3.4.4 Registration of Hip Prosthesis X-ray Images

Hip replacement [22; 45] is a surgical procedure in which the hip joint is replaced by a prosthetic implant. In the short term, post-operative, infection is a major concern. An inflammatory process causes bone resorption and a subsequent loosening or fracture often requires revision surgery. In current practice, clinicians assess loosening by inspecting a number of post-operative X-ray images of the patient's hip joint, taken over a period of time. Obviously such an analysis requires the registration of X-ray images, as shown in Fig. 3.7. Even a visual inspection can benefit from registration as clinically significant prosthesis movement can be very small [22; 45].

There are two main challenges in registering hip X-ray images: One is the highly nonlinear radiometric distortion [26], which makes any greylevel-based method unstable. Fortunately, the segmentation of the prosthetic implant is quite straightforward [79], so binary registration is a valid alternative here. From here on we will use active contours [10] to segment the implant. The second problem is that the true transformation is a projective one, which also depends on the position of the implant in 3D space. In fact, there is a rigid-body transformation in 3D space between the implants, which becomes a projective mapping between the X-ray images. Happily, the affine assumption is a good approximation here as the X-ray images are taken in a well-defined *standard position* of the patient's leg. Some registration results are shown in Fig. 3.7.

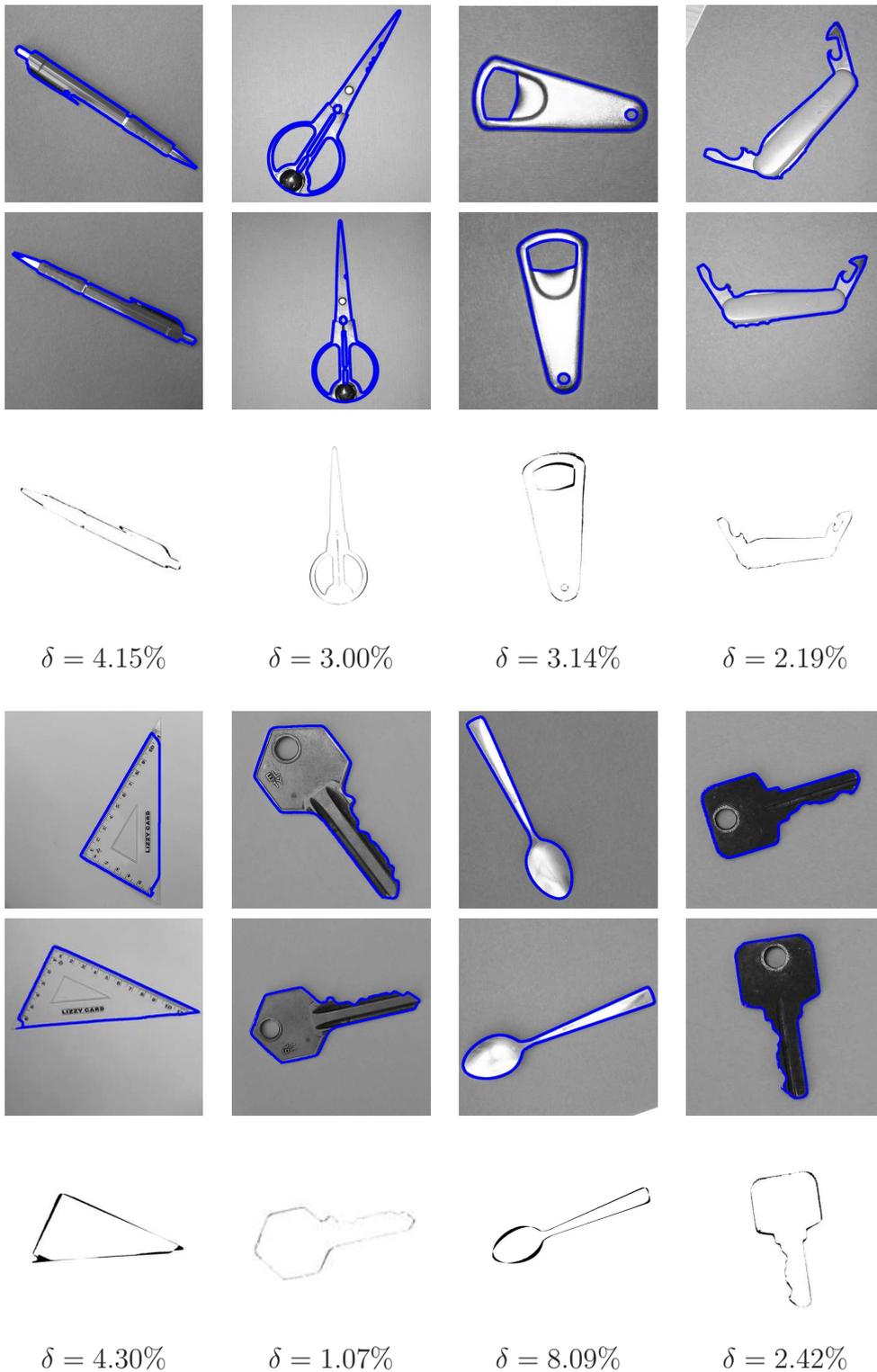


Figure 3.6: Registration results on real images. For each image pair, the first two rows contain the *template* and *observation* with overlaid contours of the segmented silhouettes, the third row shows the difference between the *registered* shapes and in the last row the evaluated error measure δ is given. Note that this value is related to the overlapping area, hence it also depends on segmentation errors as segmented regions will never match perfectly in practice.

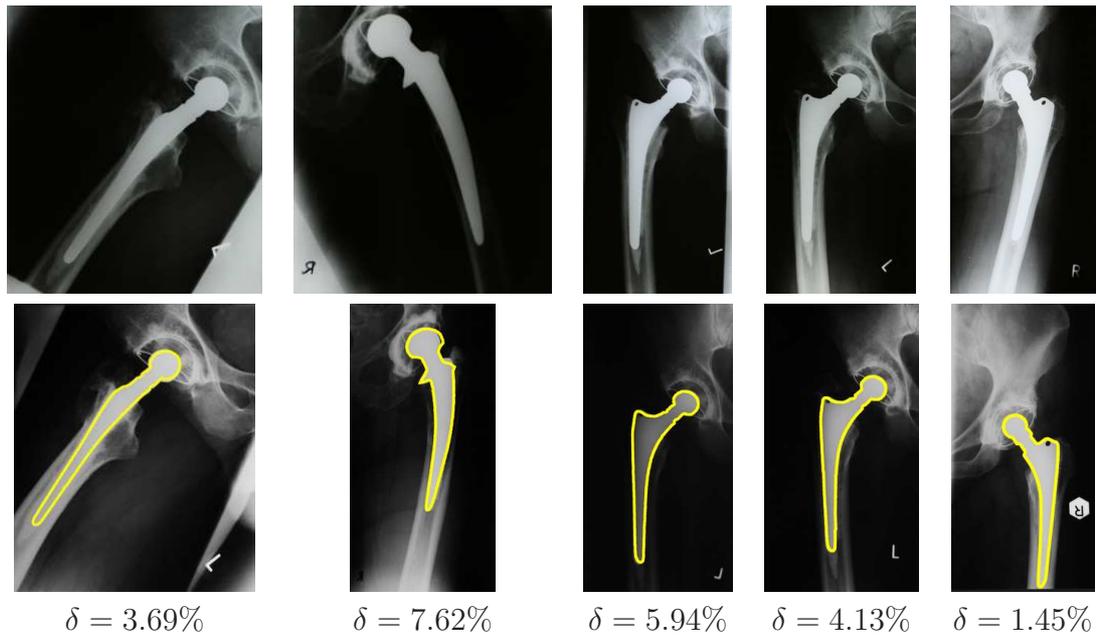


Figure 3.7: Registration of hip prosthesis X-ray images. Each image pair of a given patient was taken over a period of time. The overlaid contour in the second row shows the aligned contour of the corresponding image in the first row. For each pair, we also computed the δ measure.

3.5 Summary

In this chapter we presented a novel approach for planar shape alignment. The fundamental difference compared to classical image registration algorithms is that our model works without any landmark, feature detection or optimization, adopting a novel idea where the transformation is obtained as a solution of a set of polynomial equations. Even though it uses all the available information in the input images, there is no need for established correspondences. We also investigated the proposed method in the presence of additive geometric noise. Our algorithm is simple to implement and runs quite fast, almost independently of the image size. The experimental results show that the proposed method provides good alignment on both real and synthetic images. Moreover, it is robust in the case of noisy observations. Lastly, comparative tests showed that both the efficiency and accuracy of our model are competitive with those of the state-of-the-art methods, or outperform them.

Chapter 4

Affine Shape Alignment Using Covariant Gaussian Densities

In this chapter, we propose a novel approach for the estimation of 2D affine transformations by aligning a planar shape and its distorted *observation*. The exact transformation is got from a least-squares solution of a linear system of equations constructed by fitting Gaussian densities to the shapes which preserve the effect of the unknown transformation. In the case of compound shapes, we also propose a robust and efficient numerical scheme for achieving near real-time performance. The method was tested on synthetic as well as on real images. Its robustness in the case of missing pixels, boundary errors, and modelling errors was also verified. With our method point correspondences are not required, nor the solution of a complex optimization problem. It has linear time complexity and yields an exact solution regardless of the size of the deformation.

4.1 The Parameter Estimation of Affine Transformations

In essence, we make use of the notation defined in Section 2.5, but here we need to use inhomogeneous representations of the coordinates, *i.e.* $\mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$ and $\mathbf{y} = [y_1, y_2, \dots, y_n]^T \in \mathbb{R}^n$. Hence the *identity relation* Eq. (2.1) becomes

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{t} \quad \Leftrightarrow \quad \mathbf{x} = \mathbf{A}^{-1}(\mathbf{y} - \mathbf{t}) = \mathbf{A}^{-1}\mathbf{y} - \mathbf{A}^{-1}\mathbf{t}, \quad (4.1)$$

where $(\mathbf{A}, \mathbf{t}) \in (\mathbb{R}^{n \times n} \times \mathbb{R}^{n \times 1})$ is the unknown affine transformation that we want to recover. Similar to Eq. (2.5), the following equality holds between the characteristic functions ($\mathbb{1}_t$ and $\mathbb{1}_o$):

$$\mathbb{1}_t(\mathbf{x}) = \mathbb{1}_o(\mathbf{A}\mathbf{x} + \mathbf{t}) = \mathbb{1}_o(\mathbf{y}). \quad (4.2)$$

The crucial step of the proposed approach is to construct a pair of *covariant functions* satisfying Eq. (2.2). Once these functions have been defined, we can adopt a direct method taken from [16; 17; 19; 42] to solve for the unknown transformation (\mathbf{A}, \mathbf{t}) . When greylevel images are considered, the image functions themselves serve as appropriate covariant functions [42]. Unfortunately, the construction of such functions for binary images is quite a challenging task due to the lack of radiometric information. Hence, these functions must be based only on available geometric information. Now we will show how these kinds of functions can be constructed.

4.1.1 Construction of Covariant Functions

We know that the *template* and *observation* are equivalent to within an affine transformation (this is stated in Eq. (4.1)), and we do not need to represent shapes. Therefore we can safely consider the points of the *template* as a sample from a bivariate normally distributed random variable denoted by $X \sim N(\mu, \Sigma)$ with probability density function (PDF) [16; 17; 19]

$$p(\mathbf{x}) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right).$$

We make use of the Gaussian probability density function, which is over the shape, to construct a *covariant function pair*. We should add that the shape is not represented or modelled by a Gaussian; just the mean and covariance value of the points are estimated. It is well known that applying any linear transformation to X also results in a normal random variable $Y = \mathbf{A}X + \mathbf{t}$ with parameters

$$X \xrightarrow{(\mathbf{A}, \mathbf{t})} Y \sim N(\mu', \Sigma') = N(\mathbf{A}\mu + \mathbf{t}, \mathbf{A}\Sigma\mathbf{A}^T). \quad (4.3)$$

Obviously, Eq. (4.3) is only valid when \mathbf{A} is positive definite. In our case, (\mathbf{A}, \mathbf{t}) is an affine transformation, thus \mathbf{A} is clearly non-singular. However, a negative determinant would mean that the transformation is not orientation-preserving (see Fig. 3.1). In practice, however, such transformations are usually excluded by physical constraints, hence we shall assume that $|\mathbf{A}|$ is always positive definite. The parameters of the probability densities $N(\mu, \Sigma)$ and $N(\mu', \Sigma')$ can be easily estimated as the sample mean and covariance values (*i.e.* the mean and covariance of the point coordinates). From a geometric point of view, the mean values μ and μ' represent the centre of mass of the *template* and *observation*, respectively, while Σ and Σ' capture the orientation and eccentricity of the shapes. Fig. 4.1 shows a binary shape and the associated Gaussian density calculated for it.

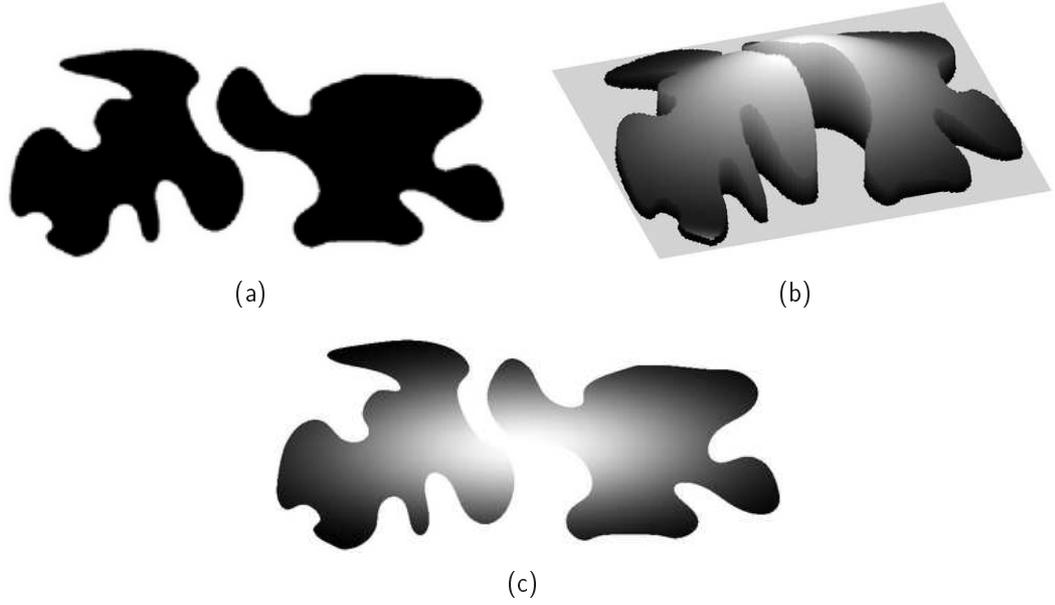


Figure 4.1: The Gaussian density function fitted over the binary shape yields a consistent colouring. (a) The original binary image. (b) A 3D plot of the Gaussian density function over the binary shape. (c) The Gaussian density represented as a greyscale image.

Now let us examine the relationship between the Gaussian density functions p and s computed from the *template* and *observation*, respectively. First [16; 17; 19]

$$s(\mathbf{y}) = \frac{1}{2\pi\sqrt{|\Sigma'|}} \exp\left(-\frac{1}{2}(\mathbf{y} - \mu')^T \Sigma'^{-1}(\mathbf{y} - \mu')\right).$$

and using $(\mathbf{y} - \mu') = (\mathbf{A}\mathbf{x} + \mathbf{t} - (\mathbf{A}\mu + \mathbf{t})) = (\mathbf{A}\mathbf{x} - \mathbf{A}\mu)$ we get

$$\begin{aligned} & \frac{1}{2\pi\sqrt{|\mathbf{A}\Sigma\mathbf{A}^T|}} \exp\left(-\frac{1}{2}(\mathbf{A}\mathbf{x} - \mathbf{A}\mu)^T \mathbf{A}^{-T} \Sigma^{-1} \mathbf{A}^{-1}(\mathbf{A}\mathbf{x} - \mathbf{A}\mu)\right) \\ &= \frac{1}{2\pi|\mathbf{A}|\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \underbrace{\mathbf{A}^T \mathbf{A}^{-T}}_{\mathbf{I}} \Sigma^{-1} \underbrace{\mathbf{A}^{-1} \mathbf{A}}_{\mathbf{I}}(\mathbf{x} - \mu)\right) \\ &= |\mathbf{A}|^{-1} \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right) = \frac{1}{|\mathbf{A}|} p(\mathbf{x}), \end{aligned} \quad (4.4)$$

where $|\mathbf{A}|$ can easily be derived from $\mathbf{A}\Sigma\mathbf{A}^T = \Sigma'$ since [16; 17; 19]

$$|\mathbf{A}||\Sigma||\mathbf{A}^T| = |\Sigma'|, \quad \text{hence} \quad |\mathbf{A}| = \sqrt{\frac{|\Sigma'|}{|\Sigma|}}. \quad (4.5)$$

It is clear from Eq. (4.4) that p and s are *covariant*. However, we can further simplify Eq. (4.4) by back substituting $|\mathbf{A}|$ into the equations and making the necessary equivalence conversions, yielding [16; 17; 19]

$$(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu) = (\mathbf{y} - \mu')^T \Sigma'^{-1}(\mathbf{y} - \mu'). \quad (4.6)$$

In fact, we get the *Mahalanobis distance*, which defines a metric over the *template* and another transformed metric over the *observation*. We then define our *covariant functions* $P, S : \mathbb{R}^n \rightarrow \mathbb{R}$ by [16; 17; 19]

$$P(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \quad \text{and} \quad S(\mathbf{y}) = (\mathbf{y} - \boldsymbol{\mu}')^T \boldsymbol{\Sigma}'^{-1} (\mathbf{y} - \boldsymbol{\mu}'). \quad (4.7)$$

With Eq. (4.6) and Eq. (4.7), we get the desired relation defined by Eq. (2.2) [16; 17; 19]

$$P(\mathbf{x}) = S(\mathbf{A}\mathbf{x} + \mathbf{t}) \quad \Leftrightarrow \quad P(\mathbf{A}^{-1}(\mathbf{y} - \mathbf{t})) = S(\mathbf{y}). \quad (4.8)$$

Note that both $P(\mathbf{x})$ and $S(\mathbf{y})$ can be computed directly from the input images. Fig. 4.2 shows an example of these functions fitted over a binary shape and its distorted observation.

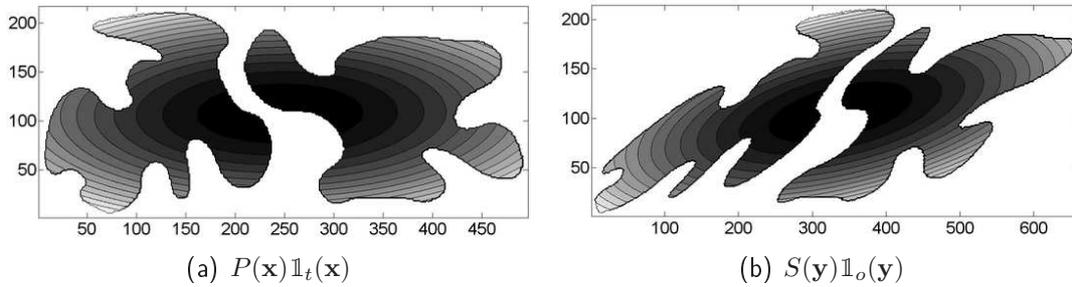


Figure 4.2: The Mahalanobis distance represented as a *covariant function*. Contours have been placed over the original greylevel images for easier evaluation. (a) The Mahalanobis distance over the original image in Fig. 4.1 and (b) over the transformed image. The transformation was a $1.5\times$ shearing along the x -axis.

It is well known that the normalizing constant $1/(2\pi\sqrt{|\boldsymbol{\Sigma}|})$ in the density functions ensures that the integral of the PDF evaluates to 1. It is also the maximum value of the density function, which is inversely proportional to the area of the shape. This dependence on the shape size may cause numerical instabilities in certain circumstances, hence we could define our *covariant functions* $\mathcal{P}, \mathcal{S} : \mathbb{R}^n \rightarrow \mathbb{R}$ as the unnormalized densities [17; 19]

$$\begin{aligned} \mathcal{P}(\mathbf{x}) &= 2\pi\sqrt{|\boldsymbol{\Sigma}|}p(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right), \quad \text{and} \\ \mathcal{S}(\mathbf{y}) &= 2\pi\sqrt{|\boldsymbol{\Sigma}'|}s(\mathbf{y}) = \exp\left(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu}')^T \boldsymbol{\Sigma}'^{-1} (\mathbf{y} - \boldsymbol{\mu}')\right). \end{aligned} \quad (4.9)$$

Since the covariance matrices and mean vectors can be computed from the images, both \mathcal{P} and \mathcal{S} are also obtained directly from the input shapes. Easy to see that they are *covariant*, and satisfy Eq. (2.2) [16; 17; 19]:

$$\mathcal{P}(\mathbf{x}) = \mathcal{S}(\mathbf{A}\mathbf{x} + \mathbf{t}) \quad \Leftrightarrow \quad \mathcal{P}(\mathbf{A}^{-1}(\mathbf{y} - \mathbf{t})) = \mathcal{S}(\mathbf{y}). \quad (4.10)$$

4.1.2 Linear Estimation of Affine Parameters

Since point correspondences are not available, we cannot construct a system directly from Eq. (4.1) or Eq. (4.8). We know, however, that the finite domains of the *template* and *observation*, $\mathcal{F}_t = \{\mathbf{x} \in \mathbb{R}^n | \mathbb{1}_t(\mathbf{x}) = 1\}$ and $\mathcal{F}_o = \{\mathbf{y} \in \mathbb{R}^n | \mathbb{1}_o(\mathbf{y}) = 1\}$, respectively, are related by (\mathbf{A}, \mathbf{t}) : $\mathbf{A}\mathcal{F}_t + \mathbf{t} = \mathcal{F}_o$ [16]. Therefore multiplying Eq. (4.1) by Eq. (4.8) we can integrate out the individual point correspondences [16; 17; 19]

$$\int_{\mathcal{F}_t} \mathbf{x}P(\mathbf{x}) d\mathbf{x} = |\mathbf{A}|^{-1} \int_{\mathcal{F}_o} \mathbf{A}^{-1}(\mathbf{y} - \mathbf{t})S(\mathbf{y}) d\mathbf{y} ,$$

where we have used the integral transformation

$$\mathbf{x} = \mathbf{A}^{-1}(\mathbf{y} - \mathbf{t}) , \quad d\mathbf{x} = |\mathbf{A}|^{-1}d\mathbf{y} .$$

Note that the Jacobian $|\mathbf{A}|$ can be readily computed from the input images using Eq. (4.5). In order to generate more linearly independent equations, we will adopt suitable nonlinear functions $\omega : \mathbb{R} \rightarrow \mathbb{R}$ and generate new equations according to Eq. (2.4) [16; 17; 19]:

$$\int_{\mathcal{F}_t} \mathbf{x}\omega(P(\mathbf{x})) d\mathbf{x} = |\mathbf{A}|^{-1} \int_{\mathcal{F}_o} \mathbf{A}^{-1}(\mathbf{y} - \mathbf{t})\omega(S(\mathbf{y})) d\mathbf{y} . \quad (4.11)$$

Letting q_{ki} denote the elements of \mathbf{A}^{-1} and $-\mathbf{A}^{-1}\mathbf{t}$, [16; 17; 19]

$$\mathbf{A}^{-1} = \begin{bmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & \ddots & \vdots \\ q_{n1} & \cdots & q_{nn} \end{bmatrix} \quad \text{and} \quad -\mathbf{A}^{-1}\mathbf{t} = \begin{bmatrix} q_{1(n+1)} \\ \vdots \\ q_{n(n+1)} \end{bmatrix} ,$$

we can expand the above integrals and get the following linear system for $k = 1, \dots, n$ [16; 17; 19]

$$|\mathbf{A}| \int_{\mathcal{F}_t} x_k \omega(P(\mathbf{x})) d\mathbf{x} = \sum_{i=1}^n q_{ki} \int_{\mathcal{F}_o} y_i \omega(S(\mathbf{y})) d\mathbf{y} + q_{k(n+1)} \int_{\mathcal{F}_o} \omega(S(\mathbf{y})) d\mathbf{y} .$$

Adopting a set of linearly independent functions $\{\omega_i\}_{i=1}^m$, we can rewrite the system in matrix form [16; 17; 19]

$$\begin{bmatrix} \int_{\mathcal{F}_o} y_1 \omega_1(S(\mathbf{y})) d\mathbf{y} & \cdots & \int_{\mathcal{F}_o} y_n \omega_1(S(\mathbf{y})) d\mathbf{y} & \int_{\mathcal{F}_o} \omega_1(S(\mathbf{y})) d\mathbf{y} \\ \vdots & \ddots & \vdots & \vdots \\ \int_{\mathcal{F}_o} y_1 \omega_m(S(\mathbf{y})) d\mathbf{y} & \cdots & \int_{\mathcal{F}_o} y_n \omega_m(S(\mathbf{y})) d\mathbf{y} & \int_{\mathcal{F}_o} \omega_m(S(\mathbf{y})) d\mathbf{y} \end{bmatrix} \times \begin{bmatrix} q_{k1} \\ \vdots \\ q_{k(n+1)} \end{bmatrix}$$

$$= |\mathbf{A}| \begin{bmatrix} \int_{\mathcal{F}_t} x_k \omega_1(P(\mathbf{x})) \, d\mathbf{x} \\ \vdots \\ \int_{\mathcal{F}_t} x_k \omega_m(P(\mathbf{x})) \, d\mathbf{x} \end{bmatrix}, \quad k = 1, \dots, n. \quad (4.12)$$

The solution of this linear system provides the parameters of the transformation. If $m > 3$ then the system is overdetermined and the result is obtained from a least-squares solution. Note that, independently of the number of systems, the matrix coefficients need to be computed only once. Hence the complexity of the algorithm depends linearly on the size of the shapes.

4.1.3 Compound Objects

Image analysis often deals with the matching of objects composed of several parts, yielding a group of disjoint shapes when segmented. Thus, assuming that the *template* object consists of $\ell \geq 2$ disjoint shapes, each component has exactly one corresponding shape on the *observation*, *i.e.* there exists a bijective mapping between the *template* and *observation* components under the transformation (\mathbf{A}, \mathbf{t}) . As a consequence, we can construct *covariant functions* $\mathcal{P}_i(\mathbf{x})$, $\mathcal{S}_i(\mathbf{y})$ for each pair of shapes similar to Eq. (4.9) that satisfy Eq. (4.10) [17; 19]:

$$\begin{aligned} \mathcal{P}_i(\mathbf{x}) &= \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_i)^T \Sigma_i^{-1}(\mathbf{x} - \mu_i)\right) \quad \text{and} \\ \mathcal{S}_i(\mathbf{y}) &= \exp\left(-\frac{1}{2}(\mathbf{y} - \mu'_i)^T \Sigma'_i{}^{-1}(\mathbf{y} - \mu'_i)\right), \end{aligned}$$

where Σ_i, Σ'_i and μ_i, μ'_i are the covariance matrices and mean vectors of the i^{th} shape on the *template* and *observation*, respectively. Furthermore, the overall shape (*i.e.* the whole foreground region) also gives rise to a pair of covariant functions $\mathcal{P}_0(\mathbf{x})$, $\mathcal{S}_0(\mathbf{y})$. Thus we have $\ell + 1$ relations. If the correspondence between the components were known, then we could simply construct a system of $\ell + 1$ equations and solve for the unknowns. As such a matching is usually not known, we will sum these relations, yielding [17; 19]

$$P(\mathbf{x}) \equiv \sum_{i=0}^{\ell} \mathcal{P}_i(\mathbf{x}) = \sum_{i=0}^{\ell} \mathcal{S}_i(\mathbf{y}) \equiv S(\mathbf{y}), \quad (4.13)$$

which can be readily used in Eq. (4.11).

Note that these sums are mixtures of unnormalized Gaussian densities, which can also be interpreted as a consistent colouring of the *template* and *observation*, respectively (see Fig. 4.3). By consistent colouring, we mean that these functions preserve the effect of the unknown transformation. Moreover, these functions can be constructed exactly and uniquely from the object points alone without knowing the aligning transformation. As a result, we can transform the original binary images into greylevel ones,

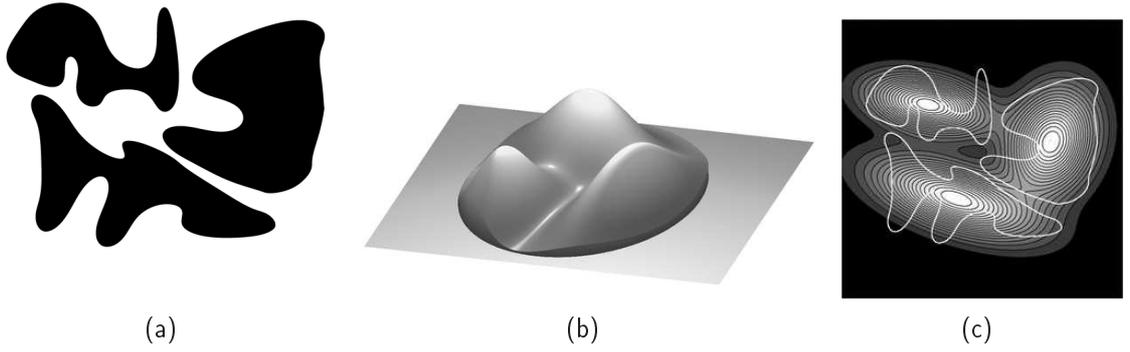


Figure 4.3: Gaussian PDFs fitted over a compound shape yield a consistent colouring. (a) Original shape. (b) A 3D plot of the Gaussian PDFs over the elliptic domain with $r = 2$. (c) Gaussian densities represented as a greyscale image. The white contour defines component boundaries.

where the corresponding pixels have exactly the same grey value.

Another way to generate new equations is to make use of $\mathcal{P}_i(\mathbf{x})$ and $\mathcal{S}_i(\mathbf{y})$ as *covariant function* instead of Eq. (4.13) in Eq. (4.11). Since the matching between the shapes is not known, we should sum the relations, defined in Eq. (4.11), for all $i = 0, \dots, \ell$ yielding [17; 19]

$$\begin{aligned} \sum_{i=0}^{\ell} \int_{\mathcal{F}_t} \mathbf{x} \omega(\mathcal{P}_i(\mathbf{x})) \, d\mathbf{x} &= \int_{\mathcal{F}_t} \mathbf{x} \sum_{i=0}^{\ell} \omega(\mathcal{P}_i(\mathbf{x})) \, d\mathbf{x} \\ &= |\mathbf{A}|^{-1} \int_{\mathcal{F}_o} \mathbf{A}^{-1}(\mathbf{y} - \mathbf{t}) \sum_{i=1}^m \omega(\mathcal{S}_i(\mathbf{y})) \, d\mathbf{y}. \end{aligned} \quad (4.14)$$

Although our approach also constructs a mixture of Gaussians, it is important to note that we do not represent shapes while the method outlined in [53] basically recovers the underlying continuous shape from the discrete point set using a method similar to Kernel Density Estimation.

4.1.4 Choosing the Integration Domain

A trivial choice for the domains in our integral equation Eq. (4.11) is the foreground regions \mathcal{F}_t and \mathcal{F}_o [16]. Since the parameters of the transformation are estimated by integrating over the segmented domains, this approach works well as long as we have a near-perfect segmentation. As it happens, this rarely occurs in practice [17; 19]. Therefore a clear disadvantage of this approach is that any segmentation error will inherently result in erroneous integrals and cause a misalignment. Furthermore, even if the segmentation is perfect, the precision of these domains is always compromised by the discretization error. However, image analysis is often concerned with the matching of objects composed of several parts, yielding a group of disjoint shapes. The topology

of such compound shapes will not change upon applying the affine group. From here on, we will develop a robust method for defining the corresponding integration domains by making use of the statistics of compound shapes.

We need to eliminate the segmentation domains \mathcal{F}_t and \mathcal{F}_o . Our goal is to select appropriate domains \mathcal{D}_t and \mathcal{D}_o that have the following properties [19]

1. They are related by the unknown transformation $\mathbf{A}\mathcal{D}_t + \mathbf{t} = \mathcal{D}_o$.
2. The integrands are rich enough (*i.e.* have characteristic patterns) within the selected domains.

The key idea is to make use of the statistics of the whole *template* and *observation* objects. Actually, the overall shape (*i.e.* the whole foreground region) of the *template* and *observation* also gives rise to a pair of covariant Gaussian densities $p(\mathbf{x})$ and $s(\mathbf{y})$. Since the equidensity contours of these PDFs are ellipsoids centered at the mean, it is natural to choose a corresponding pair of such ellipses as the integration domain. Simplifying Eq. (4.4), we get the well-known Mahalanobis distance, which defines a metric invariant under the unknown transformation (\mathbf{A}, \mathbf{t}) . Corresponding domains can then be found by selecting points whose Mahalanobis distance is less than r^2 from the mean [19]:

$$\begin{aligned}\mathcal{D}_t &= \{\mathbf{x} \in \mathbb{R}^n | (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \leq r^2\}, \\ \mathcal{D}_o &= \{\mathbf{y} \in \mathbb{R}^n | (\mathbf{y} - \boldsymbol{\mu}')^T \boldsymbol{\Sigma}'^{-1} (\mathbf{y} - \boldsymbol{\mu}') \leq r^2\}.\end{aligned}$$

To satisfy property 2, we may choose an ellipse according to the *two sigma rule* (*i.e.* $r = 2$), which guarantees that about 95% of the values lie within the enclosed ellipsoid (see Fig. 4.3). Our experiments showed that good alignments can be achieved with settings that range from $r = 1$ to $r = 3$. Another advantage is that these domains are analytical, which permits a quite efficient numerical implementation scheme, as outlined in Section 4.3.

In summary, all we need to construct a system of linear equations are the means and covariance values of the input shapes. Based on these statistics, we can choose the integration domains and construct appropriate covariant functions to get a system similar to Eq. (4.12) [17; 19]:

$$\begin{bmatrix} \int_{\mathcal{F}_o} y_1 \sum_{i=1}^{\ell} \omega_1(\mathcal{S}_i(\mathbf{y})) \, d\mathbf{y} & \cdots & \int_{\mathcal{F}_o} y_n \sum_{i=1}^{\ell} \omega_1(\mathcal{S}_i(\mathbf{y})) \, d\mathbf{y} & \int_{\mathcal{F}_o} \sum_{i=1}^{\ell} \omega_1(\mathcal{S}_i(\mathbf{y})) \, d\mathbf{y} \\ \vdots & \ddots & \vdots & \vdots \\ \int_{\mathcal{F}_o} y_1 \sum_{i=1}^{\ell} \omega_m(\mathcal{S}_i(\mathbf{y})) \, d\mathbf{y} & \cdots & \int_{\mathcal{F}_o} y_n \sum_{i=1}^{\ell} \omega_m(\mathcal{S}_i(\mathbf{y})) \, d\mathbf{y} & \int_{\mathcal{F}_o} \sum_{i=1}^{\ell} \omega_m(\mathcal{S}_i(\mathbf{y})) \, d\mathbf{y} \end{bmatrix} \times$$

$$\begin{bmatrix} q_{k1} \\ \vdots \\ q_{k(n+1)} \end{bmatrix} = |\mathbf{A}| \begin{bmatrix} \int_{\mathcal{F}_t} x_k \sum_{i=1}^{\ell} \omega_1(\mathcal{P}_i(\mathbf{x})) \, d\mathbf{x} \\ \vdots \\ \int_{\mathcal{F}_t} x_k \sum_{i=1}^{\ell} \omega_m(\mathcal{P}_i(\mathbf{x})) \, d\mathbf{x} \end{bmatrix}, \quad k = 1, \dots, n. \quad (4.15)$$

We note that $\int_{\mathcal{D}_t} \mathcal{P}_0(\mathbf{x}) \, d\mathbf{x} = 0$ and $\int_{\mathcal{D}_o} \mathcal{S}_0(\mathbf{y}) \, d\mathbf{y} = 0$ since $\mathcal{P}_0(\mathbf{x})$ and $\mathcal{S}_0(\mathbf{y})$ are symmetric over \mathcal{D}_t and \mathcal{D}_o , respectively. Thus $i = 0$ is ignored from the sums in Eq. (4.15).

4.2 Discussion

Here, we analyze the relation between the proposed approach and related techniques. We give two alternative interpretations of the proposed framework, thereby relating our approach to other state-of-the-art approaches of shape alignment. First, we analyze the method and show its relation to the metric based framework proposed by Bronstein *et al.* [12]. Applying this framework we are able to recover the non-rigid deformation of shapes. In the given framework, the problem is examined from a metric geometry point of view. Then its connection with classical moment-based approaches is discussed.

4.2.1 Relation to Metric-Based Approaches

Bronstein *et al.* proposed a generic framework for non-rigid shape matching in [12; 85], where the problem is studied from the perspective of metric geometry. The basic idea is to construct a so-called *canonical representation* of the original shape. Making use of this representation, they are able to either detect the symmetry of non-rigid shapes [85] or solve a non-rigid object recognition problem [12]. The mathematical background of this approach is based on measure theory. In the following we will investigate the similarities between our approach and the framework presented by Bronstein *et al.*

Following [12], let the *template* and *observation* be modelled as metric spaces (\mathcal{F}_t, d_t) and (\mathcal{F}_o, d_o) , where $d_t, d_o : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_0^+$ [19]

$$d_t(\mathbf{u}, \mathbf{v}) = (\mathbf{u} - \mathbf{v})^T \Sigma^{-1} (\mathbf{u} - \mathbf{v}) \quad \text{and} \quad d_o(\mathbf{u}, \mathbf{v}) = (\mathbf{u} - \mathbf{v})^T \Sigma'^{-1} (\mathbf{u} - \mathbf{v})$$

are the Mahalanobis distances constructed on the *template* and *observation*, respectively. A metric like these is often called an *extrinsic* metric [85]. Note that our Gaussians \mathcal{P} and \mathcal{S} are uniquely determined by the respective Mahalanobis distances in their exponents, hence the above metrics are equivalent to our covariant functions in Eq. (4.9). Now we will show that (\mathcal{F}_t, d_t) and (\mathcal{F}_o, d_o) are *isometric*, i.e. there exists a bijective homomorphism $\psi : (\mathcal{F}_t, d_t) \rightarrow (\mathcal{F}_o, d_o)$. Indeed, the map $\psi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is given by the unknown affine transformation (\mathbf{A}, \mathbf{t}) : $\psi(\mathbf{u}) = \mathbf{A}\mathbf{u} + \mathbf{t}$. Here ψ is clearly

bijjective since \mathbf{A} is nonsingular, and it is also easily seen that ψ is a homomorphism [19]:

$$\begin{aligned}
 d_t(\mathbf{u}, \mathbf{v}) &= (\mathbf{u} - \mathbf{v})^T \Sigma^{-1} (\mathbf{u} - \mathbf{v}) \\
 &= (\mathbf{A}^{-1} \psi(\mathbf{u}) - \mathbf{A}^{-1} \mathbf{t} - \mathbf{A}^{-1} \psi(\mathbf{v}) + \mathbf{A}^{-1} \mathbf{t})^T \\
 &\quad \mathbf{A}^{-T} \Sigma'^{-1} \mathbf{A}^{-1} \\
 &\quad (\mathbf{A}^{-1} \psi(\mathbf{u}) - \mathbf{A}^{-1} \mathbf{t} - \mathbf{A}^{-1} \psi(\mathbf{v}) + \mathbf{A}^{-1} \mathbf{t}) \\
 &= (\psi(\mathbf{u}) - \psi(\mathbf{v}))^T \Sigma'^{-1} (\psi(\mathbf{u}) - \psi(\mathbf{v})) \\
 &= d_o(\psi(\mathbf{u}), \psi(\mathbf{v})) .
 \end{aligned}$$

We should remark that the only possible isometries are affine transformations, denoted by $\text{Iso}(\mathcal{F}_t, d_t)$. This is obvious, due to the properties of the Mahalanobis distance (equidistant points lie on an ellipse, which can be obtained from an arbitrary ellipse by applying an affine transformation). Thus \mathcal{F}_t and any $\mathcal{F} \subset \mathbb{R}^2$ are extrinsically isometric [85], if there exists a $\psi \in \text{Iso}(\mathcal{F}_t, d_t)$, such that $\psi(\mathcal{F}_t) = \mathcal{F}$. This means that \mathcal{F} can be obtained from \mathcal{F}_t by applying an affine transformation. In this case \mathcal{F}_t and \mathcal{F} are called *congruent* [85].

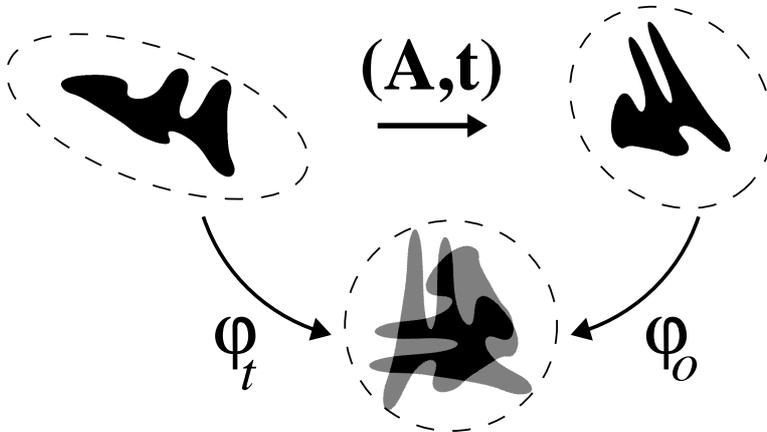


Figure 4.4: Canonical representation of a *template* and its *observation*.

For shape recognition [12; 46; 95], the *canonical form* of the shapes is used, because this representation is uniquely determined up to a rigid motion by the shape regardless of its deformation (see Fig. 4.4). Therefore, if two shapes are identical but deformed, then their *canonical forms* are related by a simple rigid body transformation, which is easy to verify [4; 103] once the *canonical representations* of the shapes are available. In the framework of Bronstein *et al.* [12], the *canonical form* is obtained by embedding the shapes into the Euclidean space $(\mathbb{E}, d_{\mathbb{E}})$, where the only possible isometries are rigid motions and the ordinary Euclidean metric $d_{\mathbb{E}}(\mathbf{u}, \mathbf{v}) = (\mathbf{u} - \mathbf{v})^T (\mathbf{u} - \mathbf{v})$ is used.

In the affine case, the embedding of the *template* $\varphi : (\mathcal{F}_t, d_t) \rightarrow (\mathbb{E}, d_{\mathbb{E}})$ is given by $\varphi(\mathbf{u}) = \mathbf{R}^T \mathbf{u}$, where \mathbf{R} is obtained by factorizing the inverse covariance matrix Σ^{-1} of \mathcal{F}_t [19]:

$$\Sigma^{-1} = \mathbf{U}\mathbf{V}\mathbf{U}^T = \mathbf{U}\sqrt{\mathbf{V}}\sqrt{\mathbf{V}}\mathbf{U}^T = (\mathbf{U}\sqrt{\mathbf{V}})(\mathbf{U}\sqrt{\mathbf{V}})^T = \mathbf{R}\mathbf{R}^T, \quad (4.16)$$

where \mathbf{V} is diagonal and \mathbf{U} could be either orthogonal or upper triangular, depending on whether we use an *SVD* or *LR* decomposition [94]. Note that the factorization of Σ^{-1} is not unique, but this ambiguity only introduces different rigid transformations within the Euclidean space. Making use of $\mathbf{u} = \mathbf{R}^{-T}\varphi(\mathbf{u})$, it can readily be seen that φ is a homomorphism, *i.e.* $d_t = d_{\mathbb{E}}(\varphi \times \varphi)$ [19]:

$$\begin{aligned} d_t(\mathbf{u}, \mathbf{v}) &= (\mathbf{u} - \mathbf{v})^T \Sigma^{-1} (\mathbf{u} - \mathbf{v}) \\ &= (\mathbf{R}^{-T}\varphi(\mathbf{u}) - \mathbf{R}^{-T}\varphi(\mathbf{v}))^T \mathbf{R}\mathbf{R}^T (\mathbf{R}^{-T}\varphi(\mathbf{u}) - \mathbf{R}^{-T}\varphi(\mathbf{v})) \\ &= (\varphi(\mathbf{u}) - \varphi(\mathbf{v}))^T \mathbf{R}^{-1}\mathbf{R}\mathbf{R}^T \mathbf{R}^{-T} (\varphi(\mathbf{u}) - \varphi(\mathbf{v})) \\ &= (\varphi(\mathbf{u}) - \varphi(\mathbf{v}))^T (\varphi(\mathbf{u}) - \varphi(\mathbf{v})) \\ &= d_{\mathbb{E}}(\varphi(\mathbf{u}), \varphi(\mathbf{v})). \end{aligned}$$

Since Σ^{-1} is positive definite, \mathbf{R}^T is nonsingular, hence φ is injective because $\varphi(\mathbf{u})$ has a unique solution for all \mathbf{u} . Furthermore $\varphi|_{\mathcal{F}_t} = \varphi_t \in \text{Iso}(\mathcal{F}_t, d_t)$, *i.e.* φ_t is bijective over \mathcal{F}_t and $\mathcal{F}_c = \varphi_t(\mathcal{F}_t)$ is called the *canonical form* of the shape \mathcal{F}_t (see Fig. 4.4).

While recognition deals with the question of whether two shapes are identical under a certain class of deformations, registration seeks to recover an aligning transformation between two shapes that is known to be identical but deformed. Obviously these are closely related problems, thus solving one will implicitly solve the other (see Section 2.6). In particular, recognition via *canonical representation* will implicitly solve the alignment problem of the shapes because (\mathbf{A}, \mathbf{t}) is got via $\varphi_o^{-1} \circ \rho \circ \varphi_t$, where ρ is the rigid body transformation between the *canonical forms*. This is exploited in [46; 95].

4.2.2 Relation to Canonical Representation-based Methods

The *canonical representation* is often used in registration, since the aligning transformation can be estimated from the normalization. Suk and Flusser [95] proposed an image moments-based approach to compute the *normalized shape*. Then making use of the normalization, the transformation between the input shapes can be estimated. Originally, this kind of *normalization* was used for object matching, since the objects, being from the same class, should produce the same or equivalent *normalized image* (*e.g.* two images in the Euclidean space are equivalent when there is a rigid motion between them).

The embedding of an image into Euclidean space is not unique, as we noted in Section 4.2.1. There exists a rigid-body transformation between the *canonical forms* of the same object, hence we cannot use φ_t, φ_o directly to estimate the transformation between the shapes, where φ_t and φ_o denote the embedding of the *template* and *observation* into Euclidean space, respectively. The affine transformation between the shapes can be expressed as $\psi = \varphi_o^{-1} \circ \rho \circ \varphi_t$, where ρ is a rigid motion. The method, presented in [46], aligns affine distorted images by tracing back the problem to estimating the angle of rotation. In fact, this method [46] computes the *canonical representation* of the shapes, *i.e.* it eliminates the other parts of the transformation except the rotation. Nevertheless, this is numerically less feasible, since a small error in the estimated rotation could result in a large error after denormalization.

Our method differs from the *canonical representation-based methods*, where the fundamental difference is that our approach does not recover *canonical shapes*. Instead of embedding the shapes into an Euclidean space, we apply nonlinear ω functions to the metrics (d_t, d_o) to get any number of *isometric* spaces. This is easily achieved since $\omega \circ d_t = \omega \circ d_o(\psi \times \psi)$, thus $\psi : (\mathcal{F}_t, \omega(d_t)) \rightarrow (\mathcal{F}_o, \omega(d_o))$ will also be an *isometry*. Then our proposed method estimates the transformation parameters by using these relations of generated metric spaces.

4.2.3 Moment-based Interpretation

Image moments and invariants were discussed in Section 3.1.2. In this sense, we can identify first order *function moments* of the form

$$m_{10}(\mathcal{F}_t) = \int_{\mathcal{F}_t} x_1 \varrho(\mathbf{x}) \, d\mathbf{x}$$

in Eq. (4.15). A fundamental question is what kind of ϱ function could be used instead of the characteristic function in order to solve the registration problem. As we pointed out, we need *covariant* functions. While invariants are invariant actions of the affine group, *covariant* functions vary with the actual transformation, hence provide constraints on the unknown parameters. Indeed, from Eq. (2.4) we get $\varrho = \omega \circ f$, where ω can be any function that satisfies Eq. (2.4). It is clear that higher order moments should be avoided in order to keep our equations *linear*. Instead, a set of linearly independent functions $\{\omega_i\}_{i=1}^m$ will be adopted to generate appropriate moments. Theoretically any ω could be used, but we will show below that *power functions* are computationally favourable.

4.3 Numerical implementation

We proposed two different approaches to construct our linear system of equations. First, when the object has only one part, the foreground regions of the objects could only be used in Eq. (4.12) for the integration domain. In this case, the *covariant functions* are constructed by making use of Mahalanobis distance, defined over the objects, and we have to integrate over the shape domains (see Section 4.3.1). In the second case, when we have compound objects we were able to choose an elliptic domain for integration. Furthermore the *covariant functions* are defined by a mixture of Gaussians. Here, we have two possible ways to compute the integrals (see Section 4.3.2). We can approximate them by finite sums over a grid with sufficient resolution. However, we will also provide an efficient numerical scheme in order to evaluate the integrals by closed forms.

4.3.1 Single Density

As we said earlier, our equations were constructed in the continuum, but in practice we only have a limited precision digital image. Hence the integrals over the domains \mathcal{F}_t and \mathcal{F}_o can only be *approximated* by a discrete sum over the foreground pixels. Evidently, the resolution of the images affects the precision of this approximation. As the mesh size tends to zero, the finite sums better approximate the integral. Therefore our method should perform better on higher resolution images.

Theoretically any function could be applied to construct the system defined in Eq. (4.12). In practice, however the registration result depends on the set of ω because of the inherent errors due to discretization. In our experiments, we found that the following set of functions provides good results (see Fig. 4.5): x , $\cos(x)$, $\cos(2x)$, $\sin(x)$ and $\sin(2x)$.

The steps of the proposed method are shown in Algorithm 2. Note that there is no

Algorithm 2: Pseudo-code of binary image registration using a single density.

Input : *Template* ($\mathbb{1}_t$) and *observation* ($\mathbb{1}_o$) shapes as binary images

Output: Estimated affine transformation ($\tilde{\mathbf{A}}, \tilde{\mathbf{t}}$)

- 1 Compute the sample means μ, μ' and covariances Σ, Σ' from the points of the foreground objects
 - 2 Construct the *covariant functions* using Eq. (4.7):

$$P, S : \mathbb{R}^n \rightarrow \mathbb{R}, \mathbf{x} \mapsto (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)$$
 - 3 Choose a set of $\{\omega_i\}_{i=1}^m$ functions, $m \geq 3$
 - 4 Estimate the Jacobian $|\mathbf{A}|$ using Eq. (4.5)
 - 5 Construct the system of linear equations in Eq. (4.12)
 - 6 Solve the system (with least-squares when $m > 3$) to get $(\tilde{\mathbf{A}}^{-1}, -\tilde{\mathbf{A}}^{-1} \tilde{\mathbf{t}})$
-

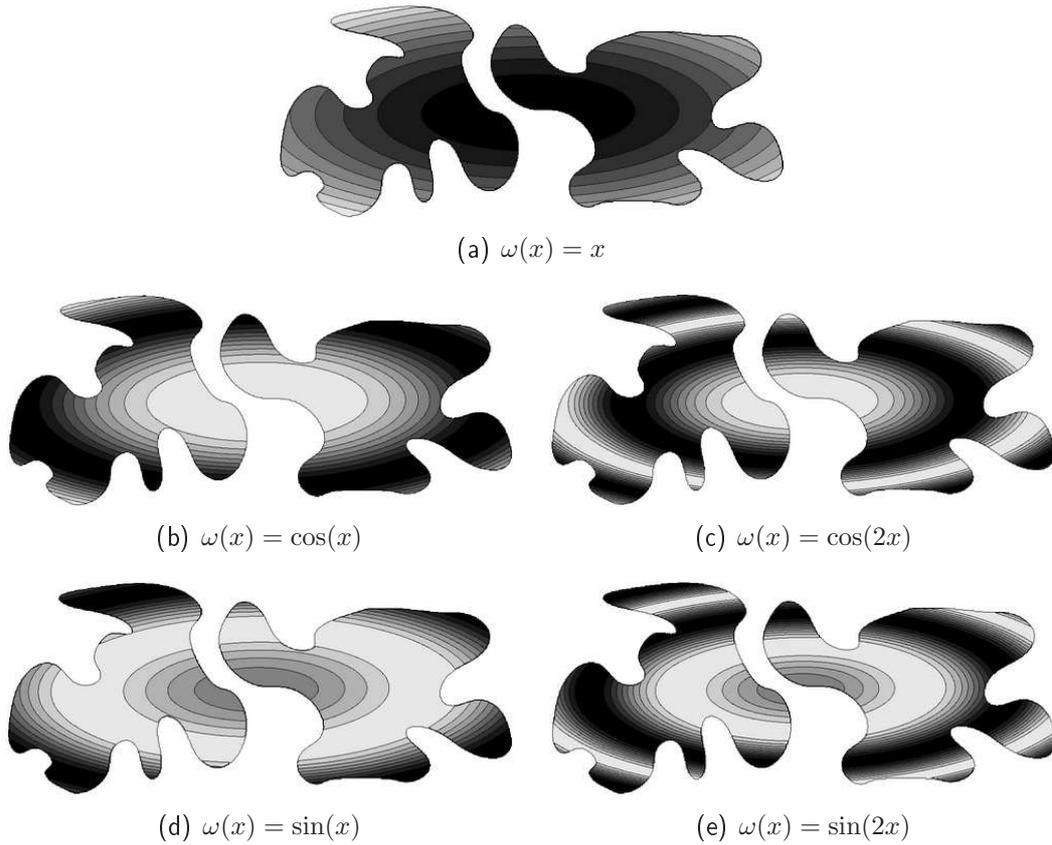


Figure 4.5: The effect of the applied ω s on the image in Fig. 4.1. Contour lines have been placed over the original greylevel images for easier evaluation.

iterative step. The proposed algorithm is very simple and the solution is provided in a single pass. It is evident from the formulation of Algorithm 2 that its time complexity is $\mathcal{O}(M + N)$, where $M + N$ is the number of foreground pixels. Due to its linear time complexity, the proposed algorithm runs quite fast on large images, so we need not compromise quality when the CPU time is a critical issue.

4.3.2 Compound Objects

If we have a compound object, we could make use of a mixture of Gaussians over an elliptic domain (see Section 4.1.4) to construct our linear system of equations in Eq. (4.15). An important numerical issue is the large numeric errors caused by greatly varying pixel coordinates. A standard technique to minimize this error is to normalize the *template* and *observation* to $[-1, 1] \times [-1, 1]$. The normalization consists of a translation of the origin to the centre of the shape followed by an appropriate scaling. If $s_k = \max_{\forall \mathbf{x} \in \mathcal{F}_t} (||x_k - \mu^{(k)}||)$, then

$$\mathbf{N}_t = \begin{bmatrix} 1/s_1 & 0 \\ 0 & 1/s_2 \end{bmatrix}, \quad \text{and} \quad \mathbf{t}_t = \begin{bmatrix} -\mu^{(1)}/s_1 \\ -\mu^{(2)}/s_2 \end{bmatrix}, \quad (4.17)$$

where the coordinates of the mean vector are $\mu = [\mu^{(1)}, \mu^{(2)}]^T$. Then the normalization of the *observation* \mathbf{N}_o is computed similarly. Note that the normalized mean and covariance of the fitted Gaussians $\hat{\mu}$ and $\hat{\Sigma}$ can also be computed from the unnormalized parameter. Here $\hat{\mu} = 0$ because of the normalization and $\hat{\Sigma} = \mathbf{N}_t \Sigma \mathbf{N}_t^T$; furthermore [19]

$$\hat{\mu}_i = \mathbf{N}_t \mu_i - \mathbf{N}_t \mu, \quad \text{and} \quad \hat{\Sigma}_i = \mathbf{N}_t \Sigma_i \mathbf{N}_t^T.$$

Applying the normalizing transformation $(\mathbf{N}_t, \mathbf{t}_t)$ to the *template* and $(\mathbf{N}_o, \mathbf{t}_o)$ to the *observation*, the algorithm will recover the transformation $(\hat{\mathbf{A}}, \hat{\mathbf{t}})$ which aligns the normalized shapes. Then the original transformation (\mathbf{A}, \mathbf{t}) is recovered by unnormalizing $(\hat{\mathbf{A}}, \hat{\mathbf{t}})$ [19]:

$$\mathbf{A} = \mathbf{N}_o^{-1} \hat{\mathbf{A}} \mathbf{N}_t, \quad \text{and} \quad \mathbf{t} = \mathbf{N}_o^{-1} \hat{\mathbf{A}} \mathbf{t}_t + \mathbf{N}_o^{-1} \hat{\mathbf{t}} - \mathbf{t}_o. \quad (4.18)$$

Furthermore, the normalization has to be taken into account in our equations as it affects the integral measure. Therefore, when normalizing the images, the left and right hand sides of the equations in Eq. (4.15) have to be multiplied by $|\mathbf{N}_t|$ and $|\mathbf{N}_o|$, respectively, and the Jacobian of the transformation is [19]

$$|\hat{\mathbf{A}}| = \frac{|\mathbf{N}_o|}{|\mathbf{N}_t|} |\mathbf{A}|.$$

Computing the Integrals

One way to compute the coefficients (*i.e.* the integrals) in our system of equations (Eq. (4.15)) is to approximate them by finite sums over a grid with sufficient resolution. Nevertheless, in contrast to Section 4.3.1, here the integration domains are ellipses chosen synthetically, hence the resolution of the image does not influence the precision. We can choose the precise resolution that we need, independently of the image resolution. In our experiments a 1500×1500 grid gave satisfactory results [17].

From here on we will develop an efficient numerical scheme to compute the integrals, which is another way of computing over the elliptic domains \mathcal{F}_t and \mathcal{F}_o defined in Section 4.1.4 in the two-dimensional case (*i.e.* $n = 2$). To do this, let us choose power functions x^l ($l \in \mathbb{Q}$) as the applied ω s (see Fig. 4.7). One term of the integral from Eq. (4.14) over the normalized *template* domain $\mathcal{D}_t = \{\mathbf{x} \in \mathbb{R}^2 | \mathbf{x}^T \Sigma^{-1} \mathbf{x} \leq r^2\}$ is computed as follows [19]

$$\int_{\mathcal{D}_t} \mathbf{x} \omega(\mathcal{P}_i(\mathbf{x})) \, d\mathbf{x} = \int_{\mathcal{D}_t} \mathbf{x} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_i)^T \Sigma_i^{-1} (\mathbf{x} - \mu_i)\right)^l \, d\mathbf{x}. \quad (4.19)$$

The computation for \mathcal{F}_o is the same as this with obvious substitutions. We can further simplify the integrand in Eq. (4.19) by translating the origin of the coordinate system

to μ_i and diagonalizing [19]

$$\Sigma_i^{-1} = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{bmatrix}.$$

The usual way to diagonalize a covariance matrix is by spectral decomposition. Unfortunately, it is inherently ambiguous as the applied rotation can be either α or $\alpha + \pi$, resulting in two different orientations of the coordinate system. In order to have a unique decomposition, we will diagonalize Σ_i^{-1} by a *shear transformation* \mathbf{G}_i : $\Sigma_i^{-1} = \mathbf{G}_i^T \mathbf{D}_i \mathbf{G}_i$, where [19]

$$\mathbf{G}_i = \begin{bmatrix} 1 & \frac{\sigma_{12}}{\sigma_{11}} \\ 0 & 1 \end{bmatrix}, \quad \text{and} \quad \mathbf{D}_i = \begin{bmatrix} \sigma_{11} & 0 \\ 0 & \frac{|\Sigma_i^{-1}|}{\sigma_{11}} \end{bmatrix} = \begin{bmatrix} d_{11} & 0 \\ 0 & d_{22} \end{bmatrix}. \quad (4.20)$$

Actually, the diagonalization of Σ_i^{-1} results in a coordinate transformation (see Fig. 4.6), which in turn gives the following integral transformation in Eq. (4.19) [19]:

$$\mathbf{z} = \mathbf{G}_i(\mathbf{x} - \mu_i) \Rightarrow \mathbf{x} = \mathbf{G}_i^{-1}\mathbf{z} + \mu_i, \quad \text{and} \quad d\mathbf{x} = |\mathbf{G}_i|^{-1} d\mathbf{z}. \quad (4.21)$$

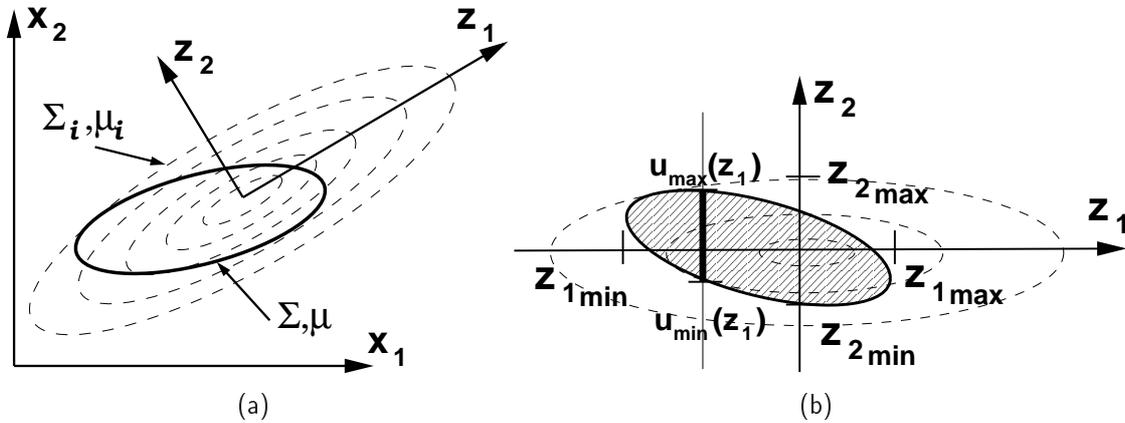


Figure 4.6: The integrals were computed over an elliptic domain. (a) An example of the domain (bold lines) and the integrated function (dashed lines), which is the i^{th} *covariant function*. (b) In order to simplify the integral, the coordinate system should be translated to the mean of the i^{th} object and its covariance matrix should be diagonalized.

Furthermore, let [19]

$$\hat{\mu} = \mathbf{G}_i \mu_i, \quad \text{and} \quad \hat{\Sigma}^{-1} = \mathbf{G}_i^{-T} \Sigma_i^{-1} \mathbf{G}_i^{-1} = \begin{bmatrix} \hat{\sigma}_{11} & \hat{\sigma}_{12} \\ \hat{\sigma}_{12} & \hat{\sigma}_{22} \end{bmatrix},$$

where $\hat{\boldsymbol{\mu}} = [\hat{\mu}^{(1)}, \hat{\mu}^{(2)}]^T$, yielding [19]

$$\mathbf{x} = \mathbf{G}_i^{-1} \mathbf{z} + \mathbf{G}_i^{-1} \mathbf{G}_i \boldsymbol{\mu}_i = \mathbf{G}_i^{-1} (\mathbf{z} - \hat{\boldsymbol{\mu}}) .$$

The integration domain \mathcal{D}_t in the new coordinate system becomes

$$\hat{\mathcal{D}}_t = \{ \mathbf{z} | (\mathbf{z} - \hat{\boldsymbol{\mu}})^T \hat{\boldsymbol{\Sigma}}^{-1} (\mathbf{z} - \hat{\boldsymbol{\mu}}) \leq r^2 \} ,$$

because using $\boldsymbol{\Sigma}^{-1} = \mathbf{G}_i^T \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{G}_i$ we get [19]

$$\mathbf{x}^T \boldsymbol{\Sigma}^{-1} \mathbf{x} = (\mathbf{z} - \hat{\boldsymbol{\mu}})^T \underbrace{\mathbf{G}_i^{-T} \mathbf{G}_i^T}_{\mathbf{I}} \hat{\boldsymbol{\Sigma}}^{-1} \underbrace{\mathbf{G}_i \mathbf{G}_i^{-1}}_{\mathbf{I}} (\mathbf{z} - \hat{\boldsymbol{\mu}}) = (\mathbf{z} - \hat{\boldsymbol{\mu}})^T \hat{\boldsymbol{\Sigma}}^{-1} (\mathbf{z} - \hat{\boldsymbol{\mu}}) . \quad (4.22)$$

Similarly, for $(\mathbf{x} - \boldsymbol{\mu}_i)$ we get [19]

$$(\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) = \mathbf{z}^T \mathbf{G}_i^{-T} \mathbf{G}_i^T \mathbf{D}_i \mathbf{G}_i \mathbf{G}_i^{-1} \mathbf{z} = \mathbf{z}^T \mathbf{D}_i \mathbf{z} .$$

Upon combining the above results, Eq. (4.19) takes the following form in the new coordinate system [19]:

$$\int_{\mathcal{D}_t} \mathbf{x} \omega(\mathcal{P}_i(\mathbf{x})) \, d\mathbf{x} = \int_{\hat{\mathcal{D}}_t} \mathbf{G}_i^{-1} (\mathbf{z} - \hat{\boldsymbol{\mu}}) \exp\left(-\frac{l}{2} \mathbf{z}^T \mathbf{D}_i \mathbf{z}\right) \, d\mathbf{z} . \quad (4.23)$$

$$\mathbf{G}_i^{-1} (\mathbf{z} - \hat{\boldsymbol{\mu}}) = \begin{bmatrix} 1 & -\frac{\sigma_{12}}{\sigma_{11}} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_1 - \hat{\mu}^{(1)} \\ z_2 - \hat{\mu}^{(2)} \end{bmatrix} = \begin{bmatrix} z_1 - \frac{\sigma_{12}}{\sigma_{11}} z_2 + \frac{\sigma_{12}}{\sigma_{11}} \hat{\mu}^{(2)} - \hat{\mu}^{(1)} \\ z_2 - \hat{\mu}^{(2)} \end{bmatrix}$$

and making use of the basic properties of integrals, it is quite apparent that we only need to compute the following types of integrals [19]:

$$c_k \int_{\hat{\mathcal{D}}_t} z_k \exp\left(-\frac{l}{2} \mathbf{z}^T \mathbf{D}_i \mathbf{z}\right) \, d\mathbf{z} , \quad k = 1, 2 , \quad \text{and} \quad c_3 \int_{\hat{\mathcal{D}}_t} \exp\left(-\frac{l}{2} \mathbf{z}^T \mathbf{D}_i \mathbf{z}\right) \, d\mathbf{z} , \quad (4.25)$$

where

$$c_1 \in \{1, 0\} , \quad c_2 \in \left\{ -\frac{\sigma_{12}}{\sigma_{11}}, 1 \right\} , \quad \text{and} \quad c_3 \in \left\{ \frac{\sigma_{12}}{\sigma_{11}} \hat{\mu}^{(2)} - \hat{\mu}^{(1)}, -\hat{\mu}^{(2)} \right\}$$

are the coefficients of z_1 , z_2 , and the constants from Eq. (4.24). Now we will derive a closed-form formula for computing the above integrals with $k = 1$. First, let us write explicitly the double integrals in Eq. (4.25). The bounds for the first variable z_1 are $z_{1_{\min}}$ and $z_{1_{\max}}$ (see Fig. 4.6 as well) [19]

$$z_{1_{\min, \max}} = \frac{|\hat{\boldsymbol{\Sigma}}^{-1} \hat{\mu}^{(1)} \mp \sqrt{|\hat{\boldsymbol{\Sigma}}^{-1}| \hat{\sigma}_{22} r^2}}{|\hat{\boldsymbol{\Sigma}}^{-1}|} = \hat{\mu}^{(1)} \mp \sqrt{\frac{\hat{\sigma}_{22} r^2}{|\hat{\boldsymbol{\Sigma}}^{-1}|}} .$$

The bounds for the second variable z_2 are functions of z_1 and correspond to a vertical slice of the ellipse $\widehat{\mathcal{D}}_t$ at z_1 :

$$u_{\min, \max}(z_1) = \widehat{\mu}^{(2)} - \frac{\widehat{\sigma}_{12}(z_1 - \widehat{\mu}^{(1)}) \pm \sqrt{\widehat{\sigma}_{22}r^2 - |\widehat{\Sigma}^{-1}|(z_1 - \widehat{\mu}^{(1)})^2}}{\widehat{\sigma}_{22}}.$$

Now the first integral from Eq. (4.25) can be rewritten as

$$\begin{aligned} c_1 \int_{z_1^{\min}}^{z_1^{\max}} z_1 \int_{u_{\min}(z_1)}^{u_{\max}(z_1)} \exp\left(-\frac{l}{2} \mathbf{z}^T \mathbf{D}_i \mathbf{z}\right) dz_2 dz_1 \\ = \frac{c_1 \sqrt{\pi}}{2C_1} \int_{z_1^{\min}}^{z_1^{\max}} z_1 \exp\left(-\frac{ld_{11}z_1^2}{2}\right) \left(\operatorname{erf}(C_1 u_{\max}(z_1)) - \operatorname{erf}(C_1 u_{\min}(z_1))\right) dz_1, \end{aligned} \quad (4.26)$$

where $C_1 = \sqrt{nd_{22}/2}$ and

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

The erf function can be efficiently approximated by its Taylor sum. We have thus reduced the original double integral to a single integral, which is simple and fast to compute by a standard quadrature formula.

We can derive similar formulas for the other two integrals in Eq. (4.25) with $C_2 = \sqrt{nd_{11}/2}$. The only difference is that we change the order of integration variables and the associated bounds are now [19]

$$z_{2\min, \max} = \widehat{\mu}_2 \mp \sqrt{\frac{\widehat{\sigma}_{11}r^2}{|\widehat{\Sigma}^{-1}|}}, \quad \text{and}$$

$$v_{\min, \max}(z_2) = \widehat{\mu}^{(1)} - \frac{\widehat{\sigma}_{12}(z_2 - \widehat{\mu}^{(2)}) \pm \sqrt{\widehat{\sigma}_{11}r^2 - |\widehat{\Sigma}^{-1}|(z_2 - \widehat{\mu}^{(2)})^2}}{\widehat{\sigma}_{11}}.$$

Then we get the following equations:

$$\begin{aligned} c_2 \int_{z_2^{\min}}^{z_2^{\max}} z_2 \int_{v_{\min}(z_2)}^{v_{\max}(z_2)} \exp\left(-\frac{l}{2} \mathbf{z}^T \mathbf{D}_i \mathbf{z}\right) dz_1 dz_2 \\ = \frac{c_2 \sqrt{\pi}}{2C_2} \int_{z_2^{\min}}^{z_2^{\max}} z_2 \exp\left(-\frac{ld_{22}z_2^2}{2}\right) \left(\operatorname{erf}(C_2 v_{\max}(z_2)) - \operatorname{erf}(C_2 v_{\min}(z_2))\right) dz_2, \end{aligned} \quad (4.27)$$

$$\begin{aligned}
c_3 \int_{z_{2\min}}^{z_{2\max}} \int_{v_{\min}(z_2)}^{v_{\max}(z_2)} \exp\left(-\frac{l}{2} \mathbf{z}^T \mathbf{D}_i \mathbf{z}\right) dz_1 dz_2 \\
= \frac{c_3 \sqrt{\pi}}{2C_2} \int_{z_{1\min}}^{z_{1\max}} \exp\left(-\frac{ld_{22}z_2^2}{2}\right) \left(\operatorname{erf}(C_2 v_{\max}(z_2)) - \operatorname{erf}(C_2 v_{\min}(z_2))\right) dz_2. \quad (4.28)
\end{aligned}$$

Choice of ω Functions

The closed-form formulas in equations (4.26)–(4.28) require that the applied set of $\{\omega_i\}_{i=1}^m$ functions be of the form

$$\omega_i : \mathbb{R} \rightarrow \mathbb{R} : \omega_i(x) = x^{l_i} \quad \text{with } l_i \in \mathbb{R}$$

While other choices of ω_i are also possible (e.g. a trigonometric family was successfully applied in a previous study [16; 17]), only the above *power functions* allow a closed form computation of the integrals. A clear benefit of this numerical scheme is near-real time performance. We found empirically that the l^{th} power and l^{th} root functions with odd l , i.e. the set $\{x, x^3, x^5, \sqrt[3]{x}, \sqrt[5]{x}\}$ (see Fig. 4.7), produced satisfactory alignments in all of our test cases.

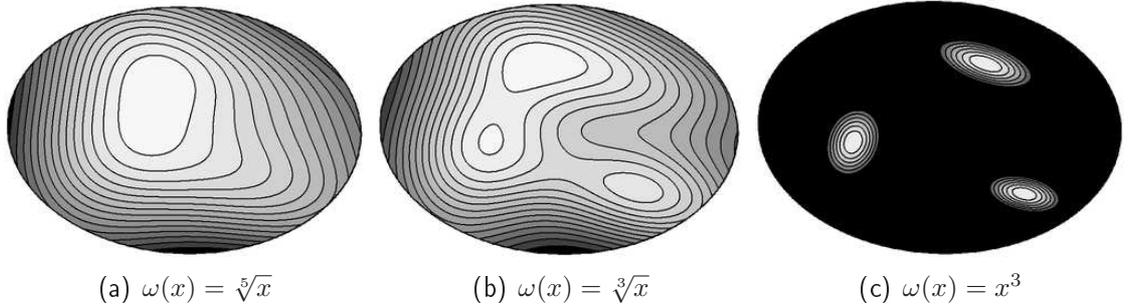


Figure 4.7: The effect of the applied ω s on the compound shape shown in Fig. 4.3.

The steps of the proposed algorithm are summarized in Algorithm 3. Note that the solution is obtained in a single pass without any loop or optimization. Although we have to compute the integrals and solve a linear system, the complexity of these steps is constant and, more importantly, it is independent of the image size. The images need to be scanned only once when computing the mean values and covariance matrices of the Gaussian densities. Once these parameters have been computed, the rest of the algorithm runs in constant time, i.e. it runs independently of the input size.

Algorithm 3: Pseudo-code of binary image registration using MPDF with an effective numerical scheme.

Input : *Template* and *observation* are labelled images ($\mathbb{I}_t, \mathbb{I}_o$); ℓ = number of components

Output: Estimated transformation ($\tilde{\mathbf{A}}, \tilde{\mathbf{t}}$)

- 1 Choose a set of exponents $\{l_i \in \mathbb{Q} | i = 1 \dots, \ell\}$
- 2 Initialize $r \in [1, 3]$.
- 3 Normalize the input images by $(\mathbf{N}_t, \mathbf{t}_t)$ and $(\mathbf{N}_o, \mathbf{t}_o)$
/*Construct the system of Eq. (4.15): \mathbf{C}_o (respectively \mathbf{C}_t) denotes the coefficient matrix (respectively constants) on the left and right hand sides of the system. */
- 4 $\{\mathbf{C}_t, |\Sigma|\} \leftarrow \text{Coeffs}(\mathbb{I}_t, \ell, r, \{l_i\}_{i=1}^m, 2)$
- 5 $\{\mathbf{C}_o, |\Sigma'|\} \leftarrow \text{Coeffs}(\mathbb{I}_o, \ell, r, \{l_i\}_{i=1}^m, 3)$
- 6 $\mathbf{C}_t \leftarrow \mathbf{C}_t \sqrt{|\Sigma'|/|\Sigma|}$
- 7 **return** $\begin{bmatrix} \tilde{\mathbf{A}} & \tilde{\mathbf{t}} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{N}_t & \mathbf{t}_t \\ \mathbf{0} & 1 \end{bmatrix}^{-1} \begin{bmatrix} [\mathbf{C}_o^+ \cdot \mathbf{C}_{t_{1..3,1}}]^T \\ [\mathbf{C}_o^+ \cdot \mathbf{C}_{t_{1..3,2}}]^T \\ [0 \ 0 \ 1] \end{bmatrix} \begin{bmatrix} \mathbf{N}_o & \mathbf{t}_o \\ \mathbf{0} & 1 \end{bmatrix}^{-1}$.

Function *Coeffs* computes \mathbf{C}_t and \mathbf{C}_o for Algorithm 3.

Input : Labelled image; ℓ = number of components; r ; $\{l_i\}_{i=1}^m$; c = number of columns of the matrix \mathbf{C}

Output: $\mathbf{C}, |\Sigma|$

- 1 Set all element of $\mathbf{C} \in \mathbb{R}^{m \times c}$ to 0
- 2 Compute μ , and Σ over the whole foreground region
- 3 **for** $i \leftarrow 1$ **to** ℓ **do**
- 4 Compute μ_i and Σ_i of the i^{th} component
/*Diagonalize Σ_i^{-1} */
- 5 Compute $\mathbf{G}_i, \mathbf{D}_i$ based on Eq. (4.20)
- 6 $\tilde{\Sigma}^{-1} \leftarrow \mathbf{G}_i^{-T} \Sigma^{-1} \mathbf{G}_i^{-1}, \tilde{\mu} \leftarrow \mathbf{G}_i \mu_i$
/*Computing the coefficients */
- 7 **for** $j \leftarrow 1$ **to** m **do**
- 8 $l \leftarrow l_j$
- 9 $\mathbf{w} \leftarrow [\text{Eq. (4.26), Eq. (4.27), Eq. (4.28)}$
- 10 $\mathbf{C}_{j,1..c} \leftarrow \mathbf{C}_{j,1..c} + \mathbf{w}_{1..c}$
- 11 **end**
- 12 **end**
- 13 **return** $\{\mathbf{C}, |\Sigma|\}$

4.4 Experimental Results

Now we will analyze the performance of our proposed algorithms. We will also compare them with the related previous approaches. In order to compare registration results, we evaluated two kinds of error measures for each estimated transformation $(\tilde{\mathbf{A}}, \tilde{\mathbf{t}})$ similar to those in Eq. (3.14) (see Chapter 3). Since we know the applied transformation for each synthetic example, we can evaluate the distance (denoted by ϵ) between the transformed version of the *template* by (\mathbf{A}, \mathbf{t}) (*observation*) and that by $(\tilde{\mathbf{A}}, \tilde{\mathbf{t}})$ (*registered*) based on all the *template* points \mathbf{x} [17; 19]:

$$\epsilon = \frac{1}{|F_t|} \sum_{\mathbf{x} \in F_t} \|(\mathbf{A} - \tilde{\mathbf{A}})\mathbf{x} + \mathbf{t} - \tilde{\mathbf{t}}\|.$$

Another useful measure is the absolute difference (denoted by δ and defined in Eq. (3.14)) between the *observation* and the *registered* image. The smaller these numbers are, the better the matching is. Thus, these measures give a quantitative characterization of the difference between the true transformation (\mathbf{A}, \mathbf{t}) and the estimated $(\tilde{\mathbf{A}}, \tilde{\mathbf{t}})$.

4.4.1 Single Density

In order to evaluate the performance of this algorithm, we created an image dataset containing 1000 synthetically generated *observations* for 37 different binary shapes. The applied transformations were randomly composed of $0^\circ, 60^\circ, \dots, 240^\circ$ rotations; 0, 0.5, 1 shearings; 0, 0.5, \dots , 2 scalings, and 0, 20 translations along both axes. The algorithm was implemented in Matlab 7.2 and run on a SunFire V490 with 8192MB memory under Solaris 10. The average runtime was around 1.5 seconds per image of size 1000×1000 . Fig. 4.8 shows a registration result, where the true (\mathbf{A}) and estimated $(\tilde{\mathbf{A}})$ transformations were [16]

$$\mathbf{A} = \begin{bmatrix} \cos(\frac{\pi}{9}) & \sin(\frac{\pi}{9}) \\ -\sin(\frac{\pi}{9}) & \cos(\frac{\pi}{9}) \end{bmatrix} \begin{bmatrix} 1.2 & 0.3 \\ 1.2 & 0.8 \end{bmatrix} = \begin{bmatrix} 1.538 & 0.5555 \\ 0.7172 & 0.6491 \end{bmatrix},$$

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1.5266 & 0.5374 \\ 0.7116 & 0.6389 \end{bmatrix}.$$

The registration error was $\delta = 0.95\%$ and $\epsilon = 1.59$ pixels.

One of the most closely related approaches is the binary registration algorithm proposed by Kannala *et al.* [57]. We obtained the Matlab implementation from the authors and conducted a comparative test on our dataset. The results presented in Table 4.1 show that our method outperforms theirs in both quality and computing time. The proposed algorithm was implemented in Matlab 7.2 and all the tests were



Figure 4.8: Registration result. (a) Distorted *observation* of the sample image in Fig. 4.1 (rotated, scaled and sheared). (b) Registration result achieved by our Single density [16] method [16]. The image is obtained by applying the recovered inverse transformation to the *observation*.

	ϵ (pixel)	δ (%)	CPU time (sec.)
Kan. <i>et al.</i> [57]	16.61	13.41	100.87
Single density [16]	5.42	2.6	1.5

Table 4.1: Registration results (medians of error measures) on 1000 images and the runtimes of the method proposed by Kannala *et al.* [57] and of our Single density method.

run on a Pentium IV 3.2 GHz under the Linux operating system.

4.4.2 Compound Objects

First, we present the results provided by our method, where the integrals are approximated by finite sums (MPDF with finite sums over a grid [17]). In this case, we use Eq. (4.13) as *covariant functions* to construct our system of equations Eq. (4.11). Now, the applied transformations were randomly composed of $0^\circ, 10^\circ, \dots, 350^\circ$ rotations; $0, 0.4, \dots, 1.2$ shearings; $0.5, 0.7, \dots, 1.9$ scalings, and $-20, 0, 20$ translations along both axes. In order to quantitatively evaluate and compare our methods we generated a dataset containing \approx binary images of size $\approx 1400 \times 1400$. In practice, segmentation never produces perfect shapes. Therefore we also evaluated the robustness of the proposed approach (MPDF with finite sums over a grid [17]) when 10%, 20%, \dots , 90% of the foreground pixels are missing from the *observation*. The Evaluation results are summarized in Table 4.2 and some registration results are shown in Fig. 4.9. It is clear that our method provides good results up to as high as 50% removed pixels, and results for 90% are also acceptable. In general, our method will perform well as long as the first and second order statistics of shapes do not change dramatically. The proposed approach was also compared with some recent binary registration approaches [57; 95], including our previous method using a single density [16]. The method of Kannala *et al.* [57] is clearly outperformed in both quality, robustness and computing time. In addition the method of Suk and Flusser [95] achieves slightly better results at the

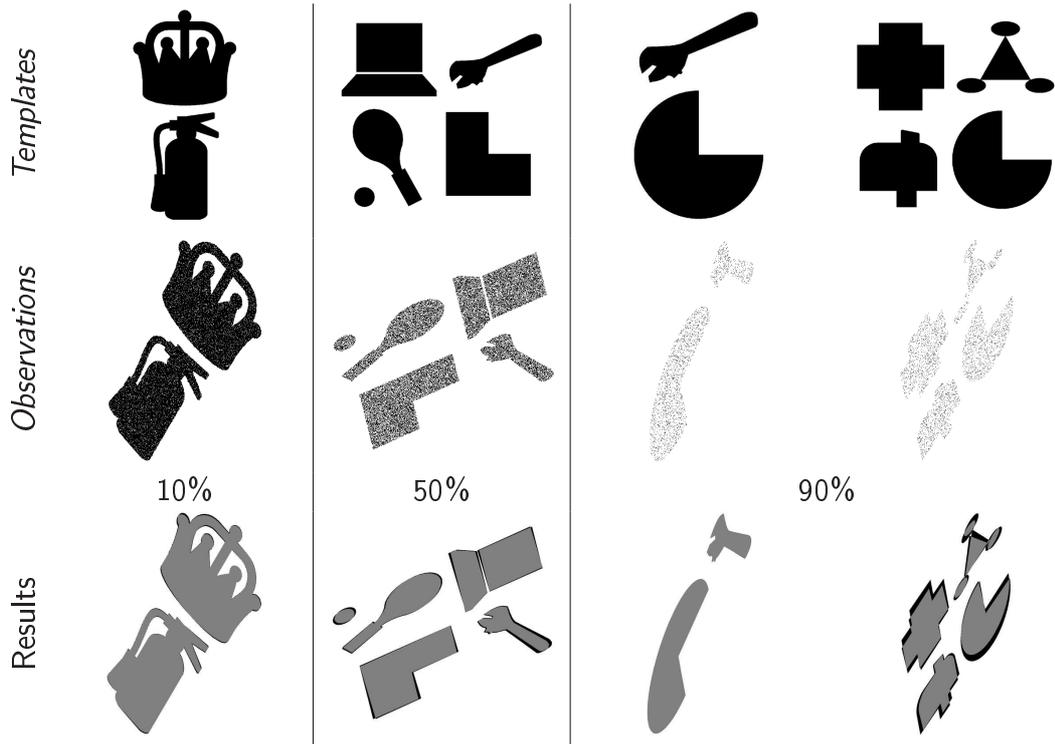


Figure 4.9: Some registration results provided by our method (MPDF with finite sums over a grid [17]), where 10%, 50% and 90% of the foreground pixels of the *observation* are missing. The images in the last row are overlaid in such a way that overlapping pixels are depicted in grey, while non-overlapping ones are shown black, respectively.

price of an increased complexity and computing time. In terms of robustness, however, our method is clearly superior to that in [95], which already performs poorly for 10% removed pixels and fails completely for 50%. Our previous method [16] performs quite well when there are no segmentation errors, but it also fails for 50% missing pixels.

	Kan. <i>et al.</i> [57]		S. & F. [95]		Single density [16]		MPDF with finite sums over a grid [17]	
	29.79 sec.		3.91 sec.		0.48 sec.		4.65 sec.	
	ϵ	δ	ϵ	δ	ϵ	δ	δ	ϵ
0%	2.7	1.65	0.43	0.06	0.64	0.31	0.58	0.25
10%	3.65	2.35	19.25	9.91	82.65	35.11	2.64	1.55
50%	7.37	4.77	109.85	51.16	407.31	84.01	7.7	4.59
90%	26.44	14.39	258.34	92.11	748.95	100	23.66	13.23

Table 4.2: Median of error measures and runtimes provided by the method of Kannala *et al.* [57], Suk and Flusser [95], and our Single density [16] and MPDF with finite sums over a grid method [17] in the case when 0%, ..., 90% of the foreground pixels of the *observation* are missing.

Now we will analyze the performance of the proposed method (see Algorithm 3), where the integrals are computed in the closed form introduced in Section 4.3.2 (MPDF with an effective numerical scheme [19]). For a quantitative evaluation, we created

several benchmark databases of ≈ 1500 synthetically generated observations of 60 different compound shapes. Similar to the previous case, the transformations we applied were randomly composed of similar transformations as above. The resulting images are of size $\approx 1500 \times 1500$, and some typical examples can be seen in Fig. 4.10. The performance of our algorithm on the benchmark dataset was evaluated based on these measures. A summary of these results is given in Table 4.3.

	Runtime (sec.)	ϵ (pixel)	δ (%)
Kan. <i>et al.</i> [57]	28.64	2.68	1.64
S. & F. [95]	4.03	0.43	0.06
MPDF with an effective numerical scheme [19]	0.33	0.54	0.19

Table 4.3: The median of error measures and runtimes on 1435 images provided by the method of Kannala *et al.* [57], Suk and Flusser [95], and our method (MPDF with an effective numerical scheme [19]).

Our approach was also compared with some recent binary registration approaches [57; 95]. One approach that is closely related to ours is that proposed by Kannala *et al.* [57], which was analyzed in Section 3.4.1. This method constructs a linear system of equations by basically looking at the images at 3 different scales, hence the solution is inherently less precise as they can only use part of the available information in each equation. In contrast to [57], our approach always uses all the information available in the images. The method proposed by Suk and Flusser [95] computes normalization parameters based on image moments. The aligning transformation can then be directly computed from these parameters. We got the Matlab implementation from the authors of both methods and conducted a comparative test on our benchmark datasets. The results presented in Table 4.3 and Fig. 4.10 show that our method outperforms both approaches in terms of computing time. Furthermore, in contrast to [57], it provides almost perfect alignments, while the registration quality of [95] is slightly better but the price is a CPU time that is 10 times bigger. However, our method is clearly superior to that described in [95] in terms of robustness, as demonstrated in the next section.

4.4.3 Robustness

The robustness of the proposed algorithm was analyzed against missing pixels, boundary and modelling errors. Besides using real images inherently subject to such errors, we also conducted a systematic test on simulated data. In the first test case, 10%, ..., 90% of the foreground pixels were removed from the *observation* before registration to simulate *missing pixels*. In the second case, square-shaped regions of a total size 1%, ..., 5% of the shape were randomly added to or removed from the boundary of foreground

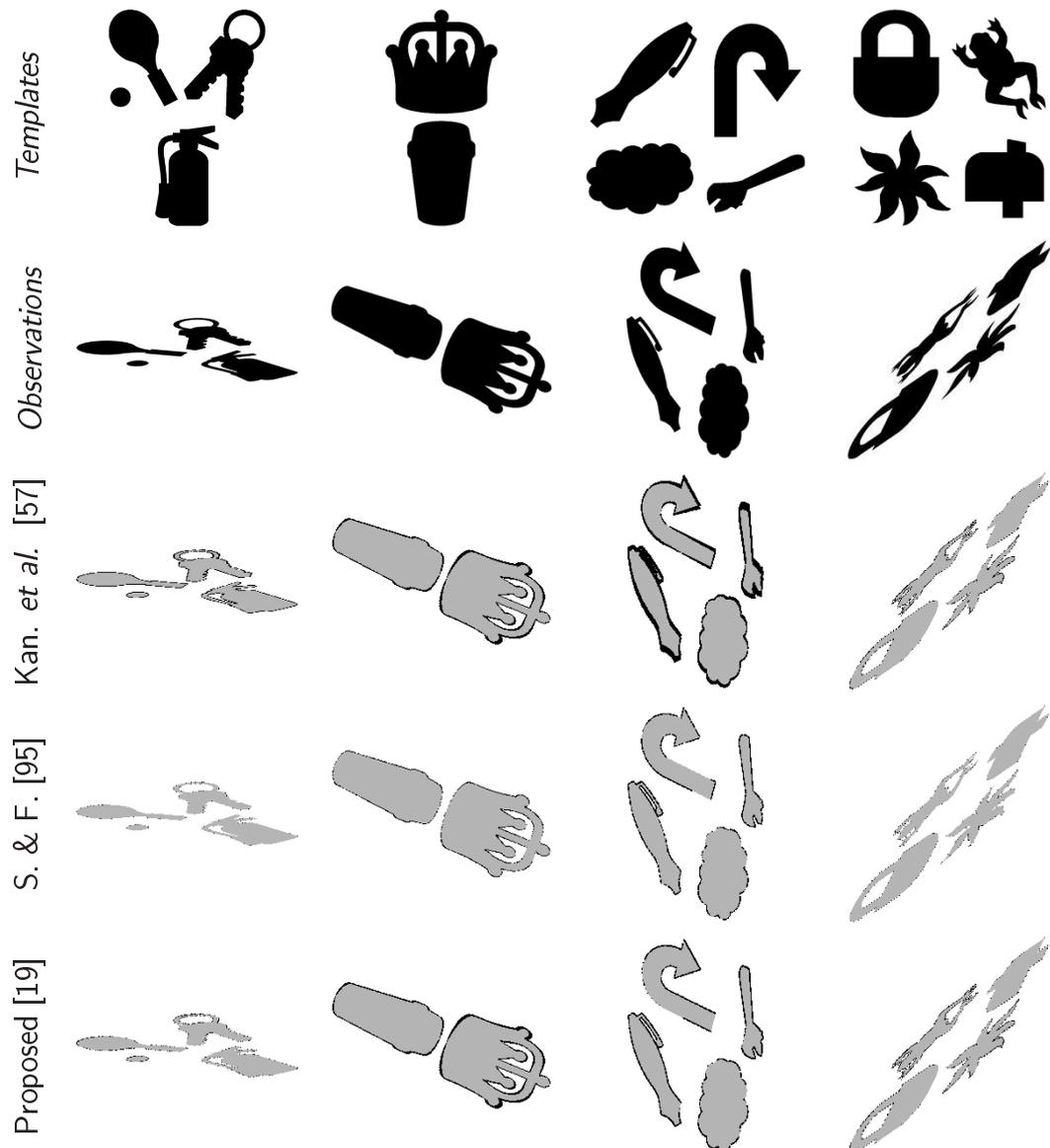


Figure 4.10: Some typical registration results on the synthetic data set. The *template* and corresponding *observation* are shown in the first row and second row, respectively. The other rows contain the registration results provided by the method of Kannala *et al.* [57], Suk and Flusser [95], and our method (MPDF with an effective numerical scheme [19]). The resulting images in the last row are overlaid in such a way that overlapping pixels are depicted in grey, while non-overlapping ones are shown black, respectively.

regions to simulate *boundary error*. Note that we do not include cases where erroneous foreground regions appear as disconnected regions, because such false regions can be efficiently removed by appropriate morphological filtering. We will therefore concentrate on cases where segmentation errors cannot be filtered out. Another issue is *modelling error*. Obviously, the method should fail for anything which is far from the affine case. It is, however, expected to be robust when an affine deformation model is a good

approximation of the true transformation. This will definitely undermine the *identity relation* in Eq. (4.1) and introduce an error into our system. Therefore our third test case consists in adding zero-mean Gaussian noise with deviation σ to each pixel coordinate (basically it is a random displacement), resulting in a nonlinear deformation. See the examples of such errors in Fig. 4.11.

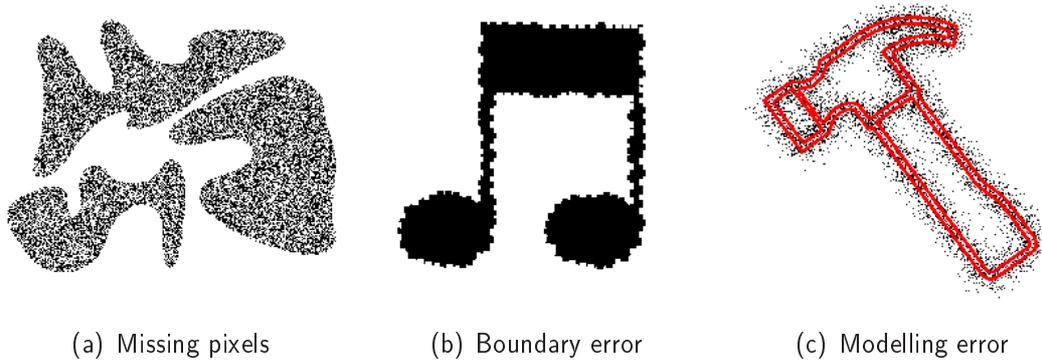


Figure 4.11: Sample *observations* for testing robustness. In (c) the true affine contours are highlighted in red.

Table 4.4 summarizes the achieved performance for the above three cases. Considering the fact that a $\delta < 6\%$ corresponds to a visually acceptable alignment, our method proved to be robust in all three cases, although at different levels. It can tolerate missing pixels up to as high as 50%, a 2.5% segmentation error, and a $\sigma = 2.5$

	Kan. <i>et al.</i> [57]		S. & F. [95]		Proposed [19]	
Missing pixels (% of removed pixels)						
%	ϵ	δ	ϵ	δ	ϵ	δ
10	3.66	2.38	19.28	9.86	2.16	1.26
50	7.65	4.71	110.23	50.54	6.06	3.67
90	26.22	14.75	258.12	92.2	18.33	10.04
Boundary errors (size of randomly added/removed squares in %)						
%	ϵ	δ	ϵ	δ	ϵ	δ
1	5.84	3.85	1.65	0.88	4.97	2.96
2.5	10.77	6.67	3.21	1.74	9.27	5.27
5	18.37	11.06	5.68	3.08	17.96	9.86
Modelling errors (σ of random displacements)						
σ	ϵ	δ	ϵ	δ	ϵ	δ
1	6.19	3.96	73.11	34.86	5.19	3.08
2.5	12.74	7.82	75.09	35.67	10	5.7
5	27.33	16.73	73.63	35.02	18.44	10.48

Table 4.4: Median of error measures vs. various types of errors provided by the method of Kannala *et al.* [57], Suk and Flusser [95], and our method (MPDF with an effective numerical scheme [19]).

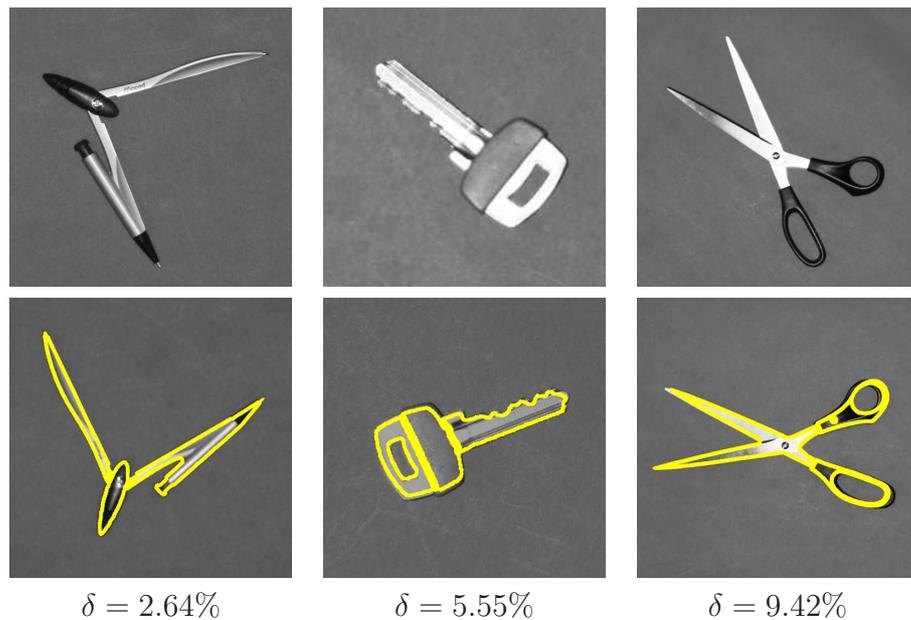


Figure 4.12: Registration results on real images. **Top:** the images used as *templates*. **Bottom:** the corresponding *observations* with the overlaid contour of the registration results.

modelling error. Note also that the results presented in [57] are consistently beaten by our approach in each test case, whereas the approach in [95] is quite sensitive to missing data and fails completely for modelling errors; but it is robust against segmentation errors. Now we should remark that occlusion yields a rather high error rate for both our method and the other state-of-the-art methods. This is because they rely on quantities obtained by integrating over the whole object area. Thus large missing parts will drastically change these quantities and result in false registrations. Nevertheless, in many application areas one can take images under controlled conditions which guarantee that observations are not occluded (e.g. in medical imaging and industrial inspection).

4.4.4 Real Images

The performance of our method was also evaluated on real images. Fig. 4.12 shows some examples of these images. The segmentation was performed via thresholding. The main challenges are segmentation errors and a slight projective distortion between the image pairs. In general, when reasonably good segmentations are available and the true transformation is close enough to an affine one then our method performs quite well, as is shown by the δ error values displayed below each image pair.

4.4.5 Registration of Traffic Signs

Nowadays, modern cars include many safety systems. Automatic traffic sign recognition is a major challenge of such intelligent systems, where one of the key tasks is the

real-time matching of an observed sign with its *template*. Fig. 4.13 shows typical registration results on such images. Here, we used classical thresholding and some morphological operations for segmentation, but automatic detection/segmentation is also possible [83]. Note that these images are inherently corrupted by several types of error. First, the true transformation is *projective*, but since signs have to be detected and recognized from a larger distance the affine model is a valid assumption here. Second, thresholding produces imperfect segmentations in the presence of specular reflections. Then, various surface errors result in missing data inside foreground regions (see the images in last column in Fig. 4.13). In spite of these problems, our method performs quite well when reasonably good segmentations are available and the true transformation is sufficiently close to an affine one.



Figure 4.13: Registration results on traffic signs. **Top:** the images used as *templates*. **Bottom:** the corresponding *observations* with the overlaid contour of the registration results

4.5 Summary

In this chapter, we presented a novel approach for planar shape alignment. The fundamental difference between classical image registration algorithms and ours is that our model works without any landmark, feature detection or optimization, incorporating a novel idea where the transformation is obtained as a solution of a set of linear equations. The complexity of the algorithm is linear and, by introducing an efficient numerical scheme, it is capable of registering images at near real-time speed. The experimental results demonstrate that our method provides good alignments on both real and synthetic images. Moreover, its robustness was demonstrated with missing pixels, boundary and modelling errors. In general, our method will perform well as long as the first and second order statistics of shapes do not change dramatically, hence its superiority can be fully exploited in applications where occlusion can be kept to a minimum. Comparative tests showed that our model is generally more efficient and accurate than the other state-of-the-art methods.

Chapter 5

Realignment of Deformed Object Fragments

This chapter addresses the problem of simultaneously estimating different linear deformations, resulting in a global nonlinear transformation between an original object and its broken fragments without correspondences. A general framework is proposed, where the solution of a polynomial system of equations directly provides the parameters of the alignment. We quantitatively evaluate the proposed algorithm on a large synthetic dataset containing 2D and 3D images, where linear (affine and rigid-body) transformations are considered. We then conduct an extended analysis of the numerical stability of the proposed algorithm. The results show that the method is robust against segmentation errors. After, we present experimental results on 2D real images as well as on volumetric medical images applied to surgical planning.

The problem itself was introduced in Section 2.3. Here, we make use of the mathematical results obtained in Chapter 3, where the registration method for one pair of shapes was proposed, and extend them to the case of several object fragments and transformations, which need to be solved simultaneously. First we need to introduce some notations and definitions, then we generalize our theory (introduced in Chapter 3) to that for several object fragments.

5.1 Realignment of Deformed Shape Fragments

We will use the notations defined in Section 2.5, *i.e.* we make use of homogeneous coordinates, and we will also introduce some new notations. Our purpose is to realign $2 \leq \ell \in \mathbb{N}$ shapes of the *observation* to their original position on the *template*. The transformation is nonlinear and is composed of ℓ linear transformations, where the i^{th}

transformation is denoted by

$$\mathbf{A}_i = \begin{bmatrix} a_{i11} & a_{i12} & \cdots & a_{i1(n+1)} \\ \vdots & & \ddots & \vdots \\ a_{in1} & a_{in2} & \cdots & a_{in(n+1)} \\ 0 & 0 & \cdots & 1 \end{bmatrix}.$$

The labelling of shapes on the input images is given by the functions $\mathbb{L}_t, \mathbb{L}_o : \mathbb{P}^n \rightarrow \{0, 1, \dots, \ell\}$, which assign a value 0 to the background. Furthermore $\mathcal{D}_i = \{\mathbf{x} \in \mathbb{P}^n | \mathbb{L}_t(\mathbf{x}) = i\}$ and $\mathcal{D}'_i = \{\mathbf{y} \in \mathbb{P}^n | \mathbb{L}_o(\mathbf{y}) = i\}$ denote the points of the i^{th} *template shape* and its distorted *observation*, respectively.

If shape correspondences were known, a pairwise alignment could be recovered by any standard binary registration method like that described in [20] and Chapter 3. Unfortunately, to find these correspondences would require solving a partial matching problem [11] between each *observation* shapes and the *template*, which is far from trivial. Therefore we are interested in a *direct solution* without identifying corresponding object-pairs.

We will assume that the input images contains ℓ shapes and that, furthermore, there exists a correspondence between them, *i.e.* there is a bijective map between these shapes. However, the correspondence is *unknown*, so it cannot be used directly. More precisely, \mathbb{L}_t is hidden, since the partitioning of the *template* is unknown, hence the characteristic function of the *template* object could only be used in that case.

5.1.1 Relation between One Pair of Shapes

Let us now consider the i^{th} fragment. The points of the i^{th} *template shape* and its distorted *observation* are related by the *identity relation* Eq. (2.1) [20]:

$$\mathbf{x} = \mathbf{A}_i \mathbf{y}. \quad (5.1)$$

Here \mathbf{A}_i denotes the transformation, realigning the i^{th} *observation* shape to its original position, but in the previous chapters it was the transformation which aligns the *template* to the *observation*. One way to recover \mathbf{A}_i is to establish point correspondences and then set up a system of equations using Eq. (5.1), whose solution provides the parameters of the unknown distortion. Since \mathcal{D}_i is unknown, finding correspondences is practically impossible.

Notice that the identity relation Eq. (5.1) remains valid when an arbitrary $\omega : \mathbb{P}^n \rightarrow \mathbb{R}$ function acts on both sides of Eq. (2.3) [18; 20]:

$$\omega(\mathbf{x}) = \omega(\mathbf{A}_i \mathbf{y}). \quad (5.2)$$

Integrating over the domain \mathcal{D}_i , we get [18; 20]

$$\int_{\mathcal{D}_i} \omega(\mathbf{x}) \, d\mathbf{x} = |\mathbf{A}_i| \int_{\mathcal{D}'_i} \omega(\mathbf{A}_i \mathbf{y}) \, d\mathbf{y} , \quad (5.3)$$

where the integral transformation

$$\mathbf{x} = \mathbf{A}_i \mathbf{y} , \quad d\mathbf{x} = |\mathbf{A}_i| d\mathbf{y}$$

has been applied and $|\mathbf{A}_i|$ is the Jacobian determinant.

Based on Eq. (5.3), we can construct as many equations as needed by making use of a set of linearly independent functions. Note that these equations do not contain any new information; they simply impose new, linearly independent constraints. The nonlinear function ω acts directly on the point coordinates and hence on the unknown parameters of \mathbf{A}_i resulting in a *nonlinear system of equations* [18; 20].

5.1.2 Solving for all Shapes Simultaneously

We know relations Eq. (5.3) between the i^{th} shape-pair, but neither the partitioning (*i.e.* the hidden labelling \mathbb{L}_t) of the *template* nor correspondences between the shapes is known. As the standard technique is to sum all the equations for all shape domains \mathcal{D}_i and solve the problem simultaneously, we estimate all the parameters in one system of equations. By making use of a set of $\{\omega_j\}_{j=1}^m$ functions in Eq. (5.3), we get [18]:

$$\sum_{i=1}^{\ell} \int_{\mathcal{D}_i} \omega_j(\mathbf{x}) \, d\mathbf{x} = \sum_{i=1}^{\ell} |\mathbf{A}_i| \int_{\mathcal{D}'_i} \omega_j(\mathbf{A}_i \mathbf{y}) \, d\mathbf{y} .$$

Let $\mathcal{F}_t := \cup_{i=1}^{\ell} \mathcal{D}_i$ is the shape domain corresponding to the whole *template*. Therefore the left hand side of the above equation can be written as [18]

$$\sum_{i=1}^{\ell} \int_{\mathcal{D}_i} \omega_j(\mathbf{x}) \, d\mathbf{x} = \int_{\cup_{i=1}^{\ell} \mathcal{D}_i} \omega_j(\mathbf{x}) \, d\mathbf{x} = \int_{\mathcal{F}_t} \omega_j(\mathbf{x}) \, d\mathbf{x} ,$$

which can be computed directly from the input image without knowing the partitioning \mathcal{D}_i . The resulting system of equations has $\ell n(n+1)$ unknowns [18]:

$$\int_{\mathcal{F}_t} \omega_j(\mathbf{x}) \, d\mathbf{x} = \sum_{i=1}^{\ell} |\mathbf{A}_i| \int_{\mathcal{D}'_i} \omega_j(\mathbf{A}_i \mathbf{y}) \, d\mathbf{y} \quad j = 1, \dots, m . \quad (5.4)$$

The solution of this system of equations provides all the unknown parameters of the overall deformation. Since each ω_j provides one equation, we need $m \geq \ell n(n+1)$ linearly independent functions to solve for ℓ linear transformations. In practice, $m >$

$\ell n(n+1)$, yielding an overdetermined system for which a *least-squares solution* is obtained.

5.1.3 Choice of ω Functions

Theoretically, any nonlinear function satisfying Eq. (5.2) could be used to construct the system of equations defined in Eq. (5.4). In practice, however, the solution is obtained via iterative least-squares minimization algorithms like the *Levenberg-Marquardt algorithm* [69], and requires a carefully chosen numerical scheme. Since the solver iteratively estimates the solution of the system of equations, it requires evaluating the integral terms in Eq. (5.4) at each iteration step, which is very time-consuming. Hence our aim is to choose an ω such that the integral terms can be computed easily. It is not hard to see that if ω is a polynome then these integrals will become precomputed constants. It was shown in Section 3.1.1 and in [20] that choosing a set of polynomial functions for ω will result in a polynomial system of equations, where these integrals become precomputed constants [20]

$$\omega(\mathbf{A}_i \mathbf{y}) = \sum_{k=1}^t p_k(\mathbf{A}_i) r_k(\mathbf{y}),$$

where $t \in \mathbb{N}$ is the finite number of terms, $p_k : \mathbb{P}^{n \times n} \rightarrow \mathbb{R}$ and $r_k : \mathbb{P}^n \rightarrow \mathbb{R}$ are polynomial too. In this case, the right hand side of Eq. (5.4) becomes a nonlinear system of equations instead of a system of integral equations (*i.e.* unknowns are emphasized in the integrals):

$$\int_{\mathcal{D}'_i} \omega(\mathbf{A}_i \mathbf{y}) \, d\mathbf{y} = \sum_{k=1}^t p_k(\mathbf{A}_i) \int_{\mathcal{D}'_i} r_k(\mathbf{y}) \, d\mathbf{y},$$

where the terms $\int_{\mathcal{D}'_i} r_k(\mathbf{y}) \, d\mathbf{y}$ can be readily precomputed from the images.

Furthermore, the simplest nonlinear system is a low order polynomial system. The time complexity of the algorithm is considerably decreased by applying a polynomial set for ω , avoiding scanning the image pixels for each iteration. Based on these findings, the following set of polynoms will be adopted [18]

$$\{\omega_j\}_{j=1}^m = \{\mathbf{x} \mapsto x_1^{u_1} \dots x_n^{u_n} \mid u_1, \dots, u_n \in \mathbb{N}_0, \sum_{i=1}^n u_i \leq d\}, \quad (5.5)$$

where $\omega_j : \mathbb{P}^n \rightarrow \mathbb{R}$, d is the maximum degree, and the number of polynoms is given by [18]

$$m = \frac{1}{n!} \prod_{i=1}^n (d+i).$$

5.1.4 Affine Transformations

We will apply the proposed framework to well-known classes of linear deformations: 2D and 3D affine and rigid-body transformations. 2D affine transformations are often used as a linear approximation of projective distortions. In general, it is important to choose the correct transformation model. When an object is broken into several parts, the fragments are generally distorted by different rigid-body transformations. A 3D rigid body transformation is important in many medical applications. In particular, when bony structures need to be aligned in CT volumes, this transformation should be performed because of the bio-mechanical properties of bones.

2D Affine Transformations

A 2D affine transformation has 6 parameters, hence we have $n(n+1)\ell = 6\ell$ unknowns. Based on Eq. (5.5), a set $\{\omega\}_{j=1}^m$ of polynoms is used to construct the system of equations. In order to obtain a sufficient number of equations, the maximum degree d has to be chosen such that [18]

$$m = \frac{(d+1)(d+2)}{2} \geq 6\ell \Rightarrow d \geq \left\lceil \frac{\sqrt{1+48\ell} - 3}{2} \right\rceil ,$$

where $\lceil \cdot \rceil$ denotes the upper integer values. Eq. (5.4) becomes [18]

$$\int_{\mathcal{F}_i} x_1^{u_1} x_2^{u_2} d\mathbf{x} = \sum_{i=1}^{\ell} |\mathbf{A}_i| \int_{\mathcal{D}'_i} (a_{i11}y_1 + a_{i12}y_2 + a_{i13})^{u_1} (a_{i21}y_1 + a_{i22}y_2 + a_{i23})^{u_2} d\mathbf{y} ,$$

where the Jacobian can be expressed as $|\mathbf{A}_i| = a_{i11}a_{i22} - a_{i12}a_{i21}$.

2D Rigid-body Transformations

This kind of transformation has only three parameters: a rotation α and translations t_1, t_2 along the two axes. Now a similar set of $\{\omega\}_{j=1}^m$ can be used as in the affine case, but we need fewer polynoms:

$$m = \frac{(d+1)(d+2)}{2} - 1 \geq 3\ell \Rightarrow d \geq \left\lceil \frac{\sqrt{9+24\ell} - 3}{2} \right\rceil .$$

Since a rigid-body transformation does not change the size of the objects, the Jacobian determinant equals 1, hence it is not included in the equations. This explains why m is 1 fewer than the affine case. Now, Eq. (5.4) becomes

$$\int_{\mathcal{D}} x_1^{u_1} x_2^{u_2} d\mathbf{x} = \sum_{i=1}^{\ell} \int_{\mathcal{D}'_i} \left\{ (y_1 \cos \alpha_i - y_2 \sin \alpha_i + t_{i1})^{u_1} (y_1 \sin \alpha_i + y_2 \cos \alpha_i + t_{i2})^{u_2} \right\} d\mathbf{y} ,$$

3D Affine Transformations

We shall also present our proposed method in the case of 3D affine transformations. The extension of the 2D case to 3D is rather straightforward. Here, the *template* parts undergo different 3D affine transformations and have a total of 12ℓ unknowns. Similar to the 2D case, we make use of Eq. (5.5) and we need to choose d such that [18]

$$m = \frac{(d+1)(d+2)(d+3)}{6} \geq 12\ell \quad \Rightarrow \quad d \geq \left\lceil \frac{c}{3} + \frac{1}{c} - 2 \right\rceil, \quad (5.6)$$

where $c = \sqrt[3]{3 \left(324\ell + \sqrt{(324\ell)^2 - 3} \right)}$. The Jacobian can be expressed in a similar way as in the 2D case.

3D Rigid-body Transformations

It has six degrees of freedom: $\alpha_1, \alpha_2, \alpha_3$ are the rotation angles and t_1, t_2, t_3 are the translations along the three coordinate axes. Again a similar set of $\{\omega\}_{j=1}^m$ can be used as in Eq. (5.6), but we need fewer polynomials [18]:

$$m = \frac{(d+1)(d+2)(d+3)}{6} - 1 \geq 6\ell \quad \Rightarrow \quad d \geq \left\lceil \frac{c}{3} + \frac{1}{c} - 2 \right\rceil,$$

where $c = \sqrt[3]{3 \left(27 + 162\ell + \sqrt{(27 + 162\ell)^2 - 3} \right)}$. The Jacobian determinant is 1, hence it is not included in the equations.

5.2 Numerical implementation

First of all, the coordinates of both images are *normalized* to the unit hyper-cube $[-0.5, 0.5]^n$, i.e. $\cup_{i=1}^{\ell} \mathcal{D}'_i \mapsto [-0.5, 0.5]^n$ and $\mathcal{F}_t \mapsto [-0.5, 0.5]^n$ in order to avoid numerical instability due to the high powers. This is achieved by translating the origin to the centre of the mass of the *template* and *observation* followed by an appropriate isotropic scaling with a common factor corresponding to the maximum size of the *template* and *observation*. Of course, the solution of the nonlinear system has to be unnormalized to get the right transformations between the original shapes. Denoting the normalizing transformations of the *template* and *observation* by $\mathbf{N}_t, \mathbf{N}_o$, respectively and the solutions by $\widehat{\mathbf{A}}_i$, the true transformation is thus obtained via $\widetilde{\mathbf{A}}_i = \mathbf{N}_t^{-1} \widehat{\mathbf{A}}_i \mathbf{N}_o$ for all $i = 1, \dots, \ell$.

Since a least-squares solution involves minimizing the algebraic error of Eq. (5.4), we expect an equal contribution from each equation in order to guarantee an unbiased error measure. This is achieved by normalizing the range of each ω_j to $[-1, 1]$. We

found experimentally that the transformations occurring during the least-squares minimization process do not transform the shapes out of a hyper-sphere with centre at the origin and a radius $\sqrt{n}/2$ (i.e. the circumscribed hyper-sphere of the unit hyper-cube). Thus normalization can be achieved by dividing the integrals in Eq. (5.4) by an appropriate constant c_j , corresponding to the maximal magnitude of the integral over this domain [18]:

$$c_j = \int_{\|\mathbf{x}\| \leq \frac{\sqrt{n}}{2}} |\omega_j(\mathbf{x})| \, d\mathbf{x}, \quad j = 1, \dots, m.$$

After summing up all the equations in Eq. (5.4), a given fragment influences the algebraic error of the system of equations with its size as weight. Obviously, the larger parts influence the algebraic error much more than the smaller one. Due to this, the bigger fragments will be much more accurately aligned. If we could normalize the terms of Eq. (5.7) corresponding to each fragment (e.g. its measures could be used), then the algebraic error would be better balanced and a precise alignment could be found, hence this drawback of the method could be eliminated. This is why, we should take into account the size of the fragments when constructing the system of equations. Unfortunately, this is not possible, since the partitioning of the *template* is unknown.

5.2.1 Algorithmic Solution and Complexity

In practice, only a limited precision digital image is available, thus the integrals can only be *approximated* by a discrete sum over the foreground pixels. The continuous domains \mathcal{F}_t and \mathcal{D}'_i are represented as finite sets of foreground pixels denoted by F_t and D'_i . Thus the discrete form of the normalized Eq. (5.4) becomes for all $j = 1, \dots, m$ [18]

$$\frac{1}{c_j} \sum_{\mathbf{x} \in F_t} \omega_j(\mathbf{N}_t \mathbf{x}) = \frac{1}{c_j} \sum_{i=1}^{\ell} |\mathbf{A}_i| \sum_{\mathbf{y} \in D'_i} \omega_j(\mathbf{N}_o \mathbf{A}_i \mathbf{y}). \quad (5.7)$$

The above system of equations is solved by iterative least-squares minimization using the *Levenberg-Marquardt algorithm* [69], which requires an evaluation of the equations for each iteration step. As we noted in Section 5.1.3, the time complexity of the algorithm will be considerably decreased if the sums can be precomputed, then the need to scan the image pixels for each iteration is avoided.

Since the *Levenberg-Marquardt* algorithm finds a local minimum, the solution depends on the choice of initialization. A better initialization may be available in a particular application, but finding it is application-dependent. We found experimentally that the following procedure provides a better initialization than the identity transformation. First, we find a solution of a larger system of equations, then we reduce the number of equations and continue the optimization process with the actual solution and the smaller system to obtain a better initialization than the identity transformation.

Hence, we look for a solution of the overdetermined system of equations, where the number of equations is at least twice as the number of unknowns. In our case, this can be achieved by increasing the maximum degree d by 2. With the new optimal solution we restart the solver, where the maximum degree d is increased by one and this results in a $1.5\times$ over-determined system of equations. Then this solution provides a good initialization for our system of equations, where the minimal d is used. For each step, the maximum number of iterations is 5000. This process also improves the quality of the alignment. The purpose of the experiments was to demonstrate the performance of our method when no better initialization is available. Note that for a medical application, the identity is a reasonable initialization choice due to physical constraints.

The simple pseudo-code of the algorithm is shown in Algorithm 5. Since a set of polynomial functions was applied to generate Eq. (5.7), the unknowns have been eliminated from the sums, as we noted in Section 5.1.3 [18; 20]. Hence the algorithm has a linear time complexity: the complexity of constructing the system Eq. (5.7) is $\mathcal{O}\left(\left(|F_t| + \sum_{i=1}^{\ell} |D'_i|\right) n^2 \ell\right)$; and the complexity of the solver itself is thus independent of the size of the input images.

Algorithm 5: Pseudo-code of realignment of deformed fragments.

Input : The binary *template* (F_t) and ℓ *observation* shapes ($D'_i, i = 1, \dots, \ell$)

Output: ℓ estimated linear transformations $\tilde{\mathbf{A}}_i$

- 1 Normalize the input coordinates by applying an appropriate similarity transformation \mathbf{N}_t and \mathbf{N}_o to $[-0.5, 0.5]^n$ so that the centre of mass becomes the origin.
 - 2 Initialize $\tilde{\mathbf{A}}_i$ with the identical transformation.
 - 3 $d_0 \leftarrow$ minimal number, which provides enough equations in Eq. (5.7) for all $\ell n(n+1)$ unknowns
 - 4 **for** $d \leftarrow 2$ **to** 0 **do**
 - 5 Choose the set of $\omega_j : \mathbb{P}^n \rightarrow \mathbb{R}$ polynomial functions based on Eq. (5.5) with the actual value of $d_0 + d$.
 - 6 Construct the (overdetermined) system of equations Eq. (5.7).
 - 7 Find a least-squares solution of the system using a *Levenberg-Marquardt* algorithm. The solver is initialized with the actual values of $\tilde{\mathbf{A}}_i$.
 - 8 **end**
 - 9 Unnormalize the solutions $\hat{\mathbf{A}}_i$ to get the parameters of the aligning transformation via $\tilde{\mathbf{A}}_i = \mathbf{N}_t^{-1} \hat{\mathbf{A}}_i \mathbf{N}_o$.
-

5.3 Experimental results

The proposed method was evaluated on 2D and 3D synthetic datasets. In the case of 2D transformations, the dataset consisted of 10 *template* objects. Synthetic observations were generated by first cutting each object into 2 parts in 4 different ways, resulting in

4 images for each *template*. Then 600 *observations* of size 700×700 were generated by applying randomly composed affine transformations to each of these images with the following parameter ranges: rotation angles of $[-\pi/4; \pi/4]$ and scaling factors from $[0.75; 1.25]$, skewing from $[-0.1; 0.1]$, and translations of $[-25; 25]$ along both axes. In the 3D case, 10 *template* volumes were randomly cut into 2 parts by a plane, such that the smaller part was at least 20% of the original volume. By cutting each volume in five different ways, 50 volume images are obtained. Then random 3D affine transformations with similar parameters as in the 2D case (the only difference being that translations were in the interval $[-10; 10]$) were used to generate a total of 200 3D *observations* of size $250 \times 250 \times 250$.

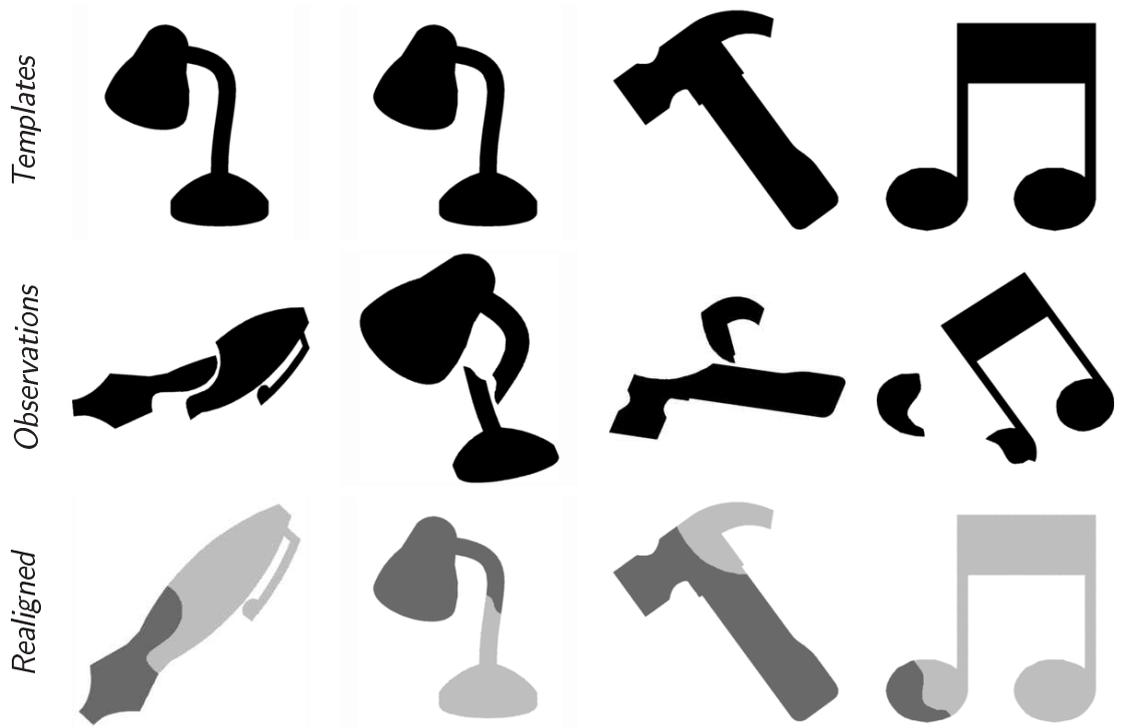


Figure 5.1: Sample results on 2D synthetic images.

For the evaluation of alignment results, we defined two kind of error measures. The first (denoted by ϵ) measures the average misalignment per pixels and the second is the estimated $\tilde{\mathbf{A}}_i$ transformation for each object. The second is the absolute difference (denoted by δ) between the *template* and the *realigned* image [18]:

$$\epsilon = \frac{1}{|F_o|} \sum_{\substack{\mathbf{p} \in D'_i \\ 1 \leq i \leq \ell}} \|(\mathbf{A}_i - \hat{\mathbf{A}}_i)\mathbf{p}\|, \quad \text{and} \quad \delta = \frac{|F_r \triangle F_t|}{|F_r| + |F_t|} \cdot 100\%,$$

where $F_o = \cup_{i=1}^{\ell} D'_i$ and F_r denote the set of pixels of the *observation* and *realigned* shape respectively. As a subjective evaluation measure, we found experimentally that

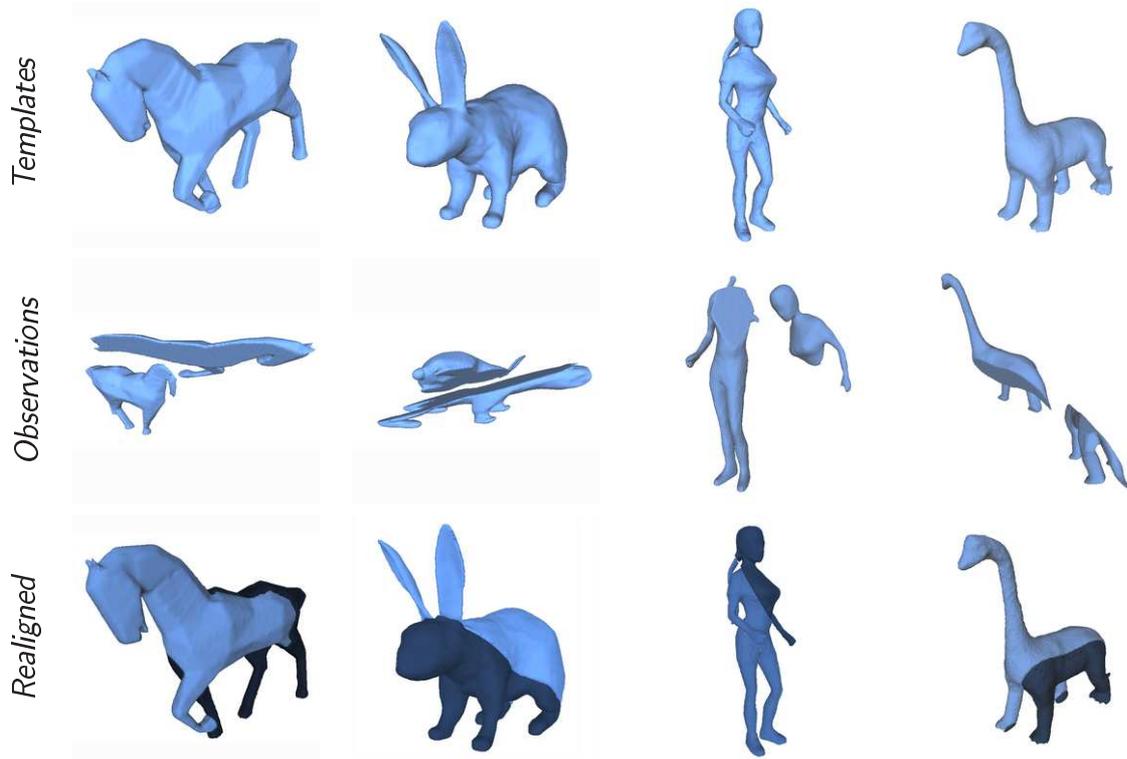


Figure 5.2: Sample results on 3D synthetic images.

	ϵ (pixel)			δ (%)		
	μ	σ	m	μ	σ	m
2D	15.62	46.33	0.11	0.36	1.65	0.13
3D	7.56	15.79	0.68	9.44	14.62	2.78

Table 5.1: Mean, standard deviation and median of error measures achieved by the proposed method on the 2D and 3D synthetic datasets.

a $\delta \leq 5\%$ in 2D and a $\delta \leq 10\%$ in 3D corresponds to a visually good alignment.

The proposed method was implemented in Matlab and ran under Linux with 3GHz CPU and 3GB RAM. A typical runtime was under 2.5 seconds for 2D and 40 seconds for 3D shapes. Some results are shown in Fig. 5.1 and Fig. 5.2. The quantitative results in Table 5.1 clearly show that the proposed method provides almost perfect alignments in both 2D and 3D. Fig. 5.3 shows the evaluated error measures (ϵ, δ) in the synthetic case in ascending order. It can be seen that there is no systematic error, but only a few test cases provide unacceptable results.

5.3.1 Robustness

In practice, segmentation never produces perfect shapes. Therefore, besides using various kinds of real images inherently subject to such errors, we also evaluated the

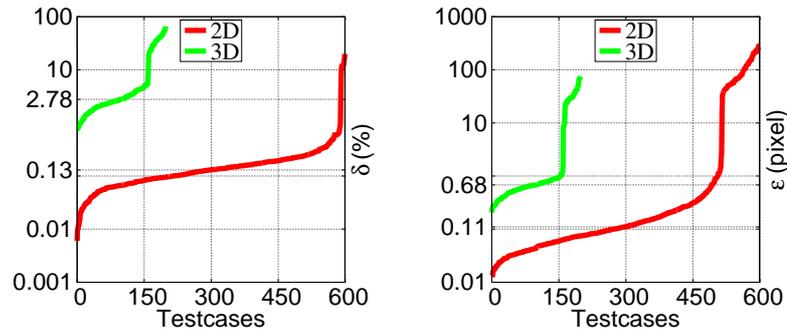


Figure 5.3: The evaluated error measures in ascending order on the synthetic dataset, where the y -axes have a logarithmic scale. The median of error measures are also shown by a horizontal grid.

robustness of the proposed approach against different types of synthetically generated segmentation errors. In our test cases 1%, ..., 10% of the foreground pixels were altered. In the first test case the missing pixels are considered, where some foreground pixels were removed from the *observation* before registration. In the second case, we occluded continuous square-shaped regions of size amounting to 1%, ..., 10% of the shape. Next, we randomly added or removed squares uniformly around the boundary. Note that we do not include cases where erroneous foreground regions appear as disconnected regions, because such false regions can be efficiently removed by appropriate morphological filtering. We therefore concentrate on cases where segmentation errors cannot be filtered out. See samples of such errors in Fig. 5.4.

Fig. 5.5 shows that our method is quite robust whenever errors are uniformly distributed over the whole shape (in the case of first and third type noise). However, it becomes less stable in the case of larger localized errors like occlusion. This is the usual behaviour of area-based methods because they rely on quantities obtained by integrating over the object area. Thus large missing parts could drastically change these quantities and result in false alignments. Nevertheless, in many application areas one can take images under controlled conditions which guarantee that observations are not occluded (e.g. medical imaging and industrial inspection).

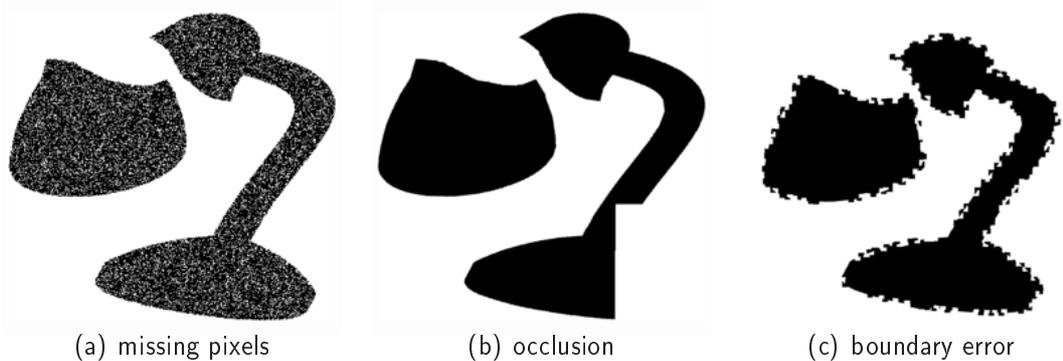


Figure 5.4: Sample *observations* with various degradations.

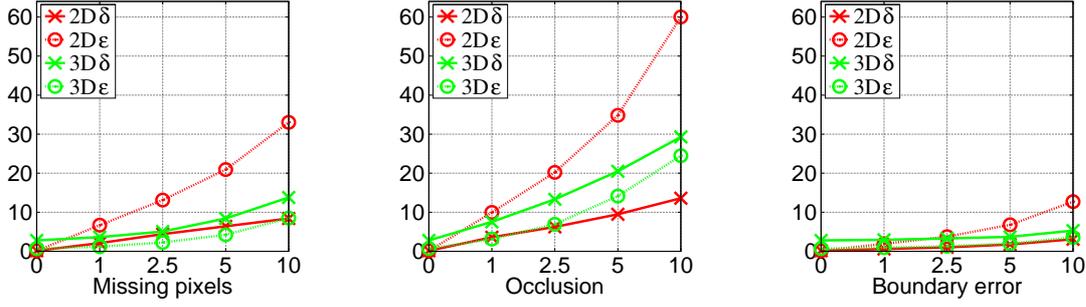


Figure 5.5: Median of error measures achieved by the proposed method on the 2D and 3D synthetic datasets in the case of various types of error.

5.3.2 Numerical Stability

The above experiments demonstrate the performance of the proposed method in the case of two fragments. Obviously, the more parts we have, the more equations are required, which affects numerical stability. In addition, more parts allow more affine transformations that may produce a stronger deformation. Now we will analyze the alignment quality in the case of more fragments and various strengths of deformations. To do this we will define the average distortion of a pixel as (*i.e.* as an ϵ measure):

$$\kappa = \frac{1}{|F_t|} \sum_{\substack{\mathbf{p} \in D_i \\ 1 \leq i \leq \ell}} \|(\mathbf{A}_i - \mathbf{I})\mathbf{p}\| ,$$

where $\mathbf{I} \in \mathbb{P}^{n \times n}$ is the identity matrix. κ is defined such that it clearly shows the strength of the overall deformation. Making use of this measure, we can easily classify the generated *observations* based on their strength of deformation. In order to analyze the numerical stability of our method, we will consider some *observations* that are generated with similar values of κ , then we will examine the results. Fig. 5.6 provides examples of average distorted objects.

In order to analyze the results obtained by the proposed algorithm in the case of various κ values, we generated three different 2D datasets with 2, 3, 4 and 5 object fragments, where the value of κ was constrained. The *observations* were generated by applying randomly composed affine transformations with the following parameter ranges: rotation angles of $[-\pi/8; \pi/8]$ and scaling factors from $[0.75; 1.25]$, skewing from $[-0.1; 0.1]$, and translations of $[-10; 10]$ along both axes. The generated transformations chosen were $|\kappa - k| \leq 5$, where $k \in \{60, 70, 80\}$. After applying the selected transformation, the fragments were translated until the image had no overlaid fragments. Due to this translation the average κ was 100, 110 and 120 in our three cases. Fig. 5.6 shows an example where the object has more fragments and the strength of deformation is close to the average value of κ .

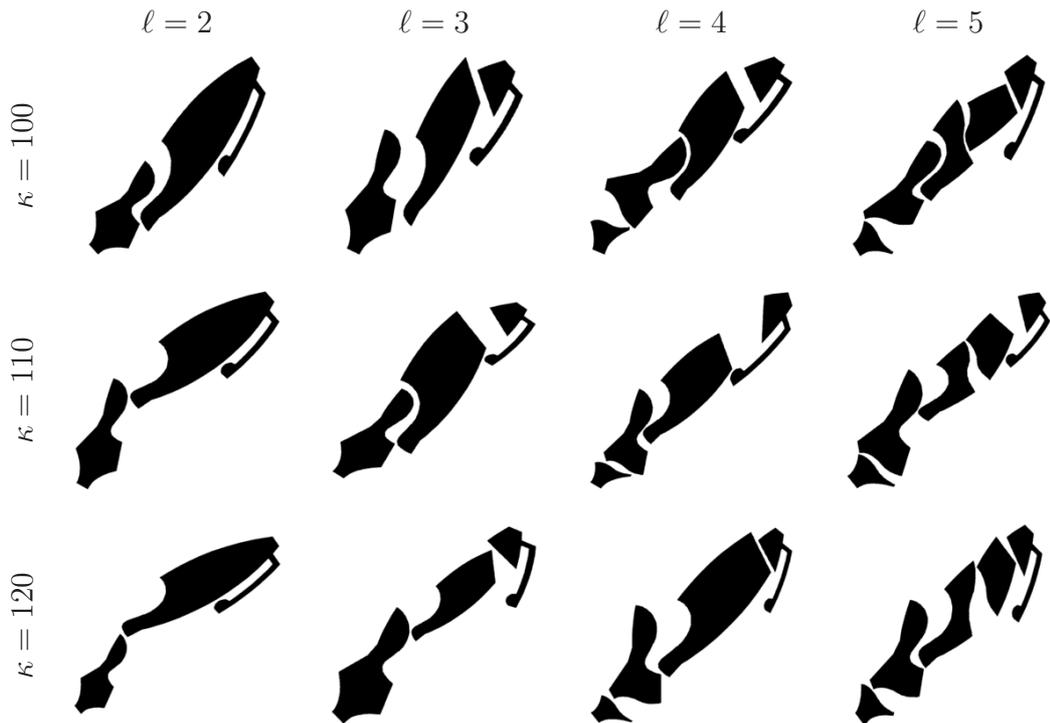


Figure 5.6: Examples of *observations* for various strengths of deformations (in the rows, with an average κ value of 100, 110 and 120, respectively) and different numbers of fragments (in the columns).

The graphs in Fig. 5.7 show that the fragment number affects numerical stability. A bigger number of fragments results in a more complex deformation, hence it makes the *puzzle problem* more difficult. We note that a completely random fragment-configuration corresponds to a complex deformation, for which a stable solution is difficult to obtain. And when pieces are in a relative order then a rather accurate solution can be obtained. Nevertheless, the quality of results obtained by the proposed method is less influenced by increasing the strength of the deformation. In the last row (on Fig. 5.7) there are histograms of κ values of the *observations* for our three different datasets.

Symmetry is an issue which can influence numerical stability. We note that both the shape and the adopted omega function are simultaneously symmetric, causing the corresponding equation to vanish. This is unlikely in practice but even if it occurs, our system is overdetermined so doing without a few equations is acceptable.

5.3.3 Real Images

We will now demonstrate our method for two possible real applications, where 2D and 3D rigid-body transformations are considered. First an interesting problem, solving the 2D Tangram problem is considered. We assume that planar tiles are approximately

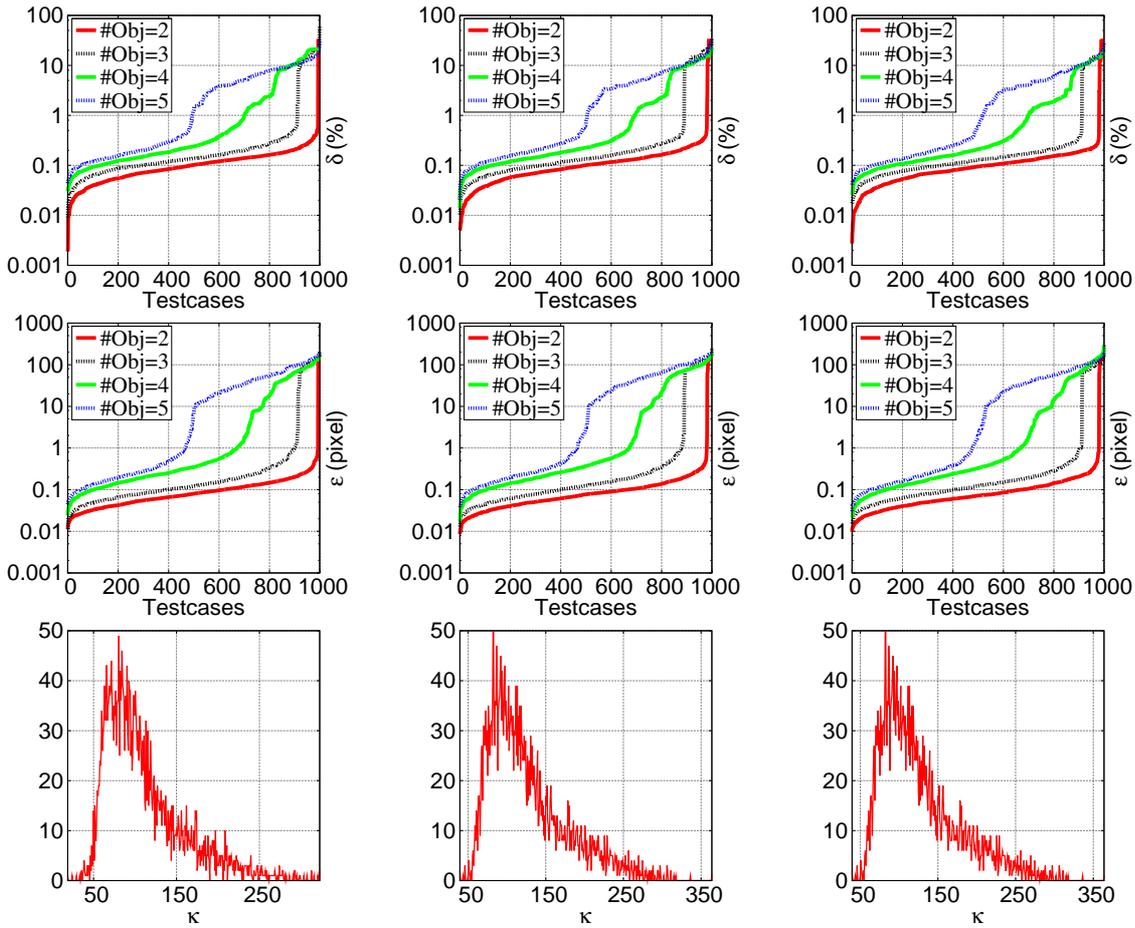


Figure 5.7: The evaluated error measures in ascending order for 1000 randomly generated test cases with different numbers of fragments and measures of distortion. In the last row, there are the histograms of average distortion (κ) for various strengths of deformations.

deformed by 2D rigid-body transformations. Then a medical application is presented, where we address the problem of realigning bone fractures. Because of the biomechanical properties of the bones, here we need to consider a 3D rigid-body deformation.

Solving the Tangram Puzzle

A Tangram is a dissection puzzle consisting of seven flat tiles (called *tans*), which are put together to form various shapes. The goal is to form a specific shape given only by its silhouette. Fig. 5.8 shows some examples of these shapes and the solutions found by our method. The images were taken with a digital camera, then thresholded. The resulting 2D shapes were realigned according to the *template*. Here the *templates* are scanned versions of the printed shapes found in the Tangram manual, which are only approximate silhouettes of the final tile configurations. The 2D rigid-body model is a good approximation of the actual transformation acting between the shapes. However, there is a similarity transform between the shapes. After applying a global scaling

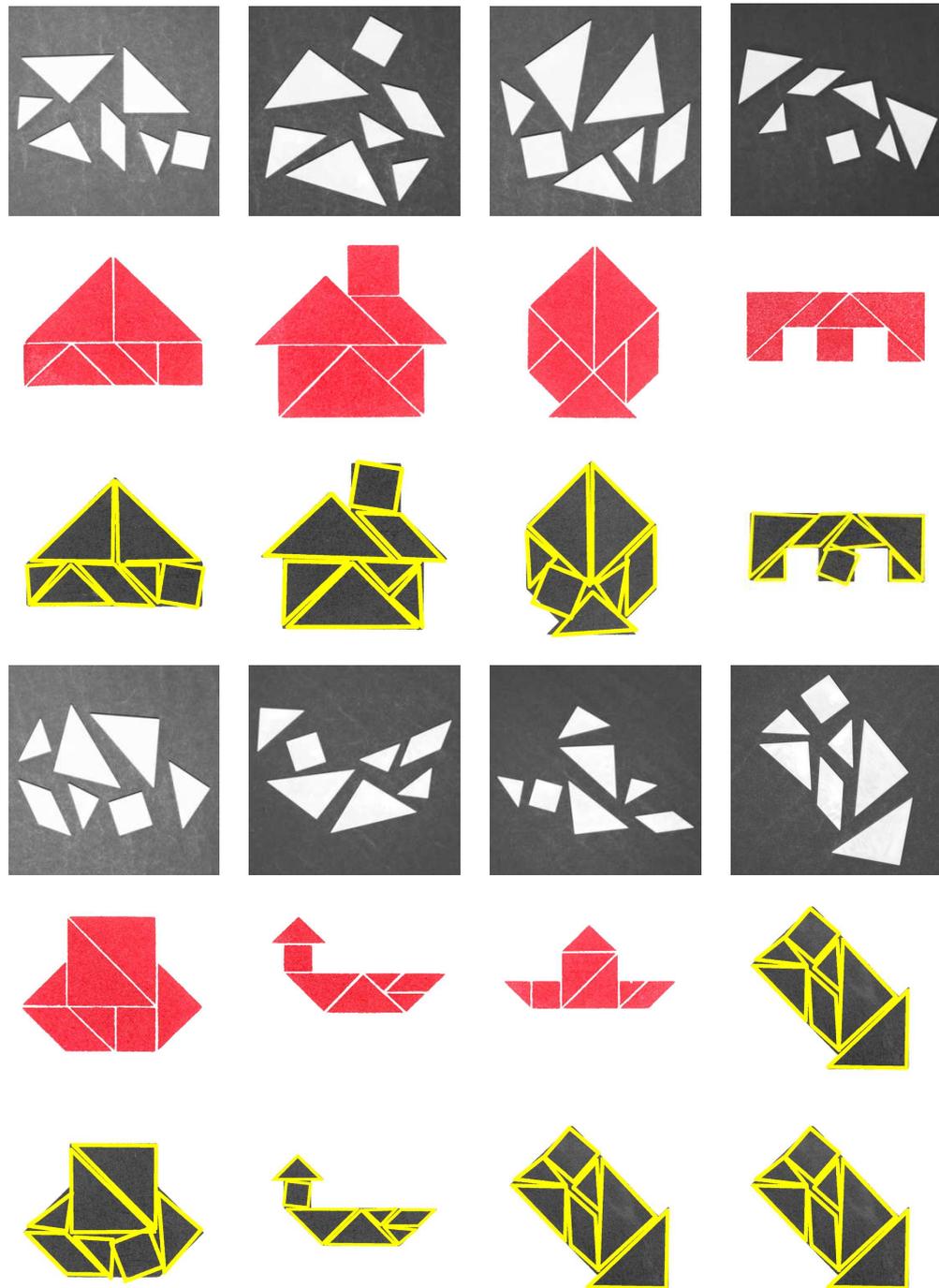


Figure 5.8: Solutions of the Tangram puzzle (the average alignment runtime of an image was about 50 sec.) with a rigid-body transformation. **Top:** *observations* were taken by a digital camera. **Middle:** solutions found in the Tangram manual. **Bottom:** the scanned *template* silhouettes with overlaid contours of aligned fragments.

factor to the input images, which is estimated based on the ratio of the foreground areas (*i.e.* the measure of the similarity transformation), we simply solve the problem by considering 2D rigid-body transformations between the shapes.

It is well known that the *Levenberg-Marquardt* algorithm finds a local minimum close

to the initialization settings. Finding a good initial configuration is largely application-dependent. For example, on these images a global optimization procedure like the Spectral Gradient Method (SPG) [9] provided a good initialization, while *Levenberg-Marquardt* gives a better solution than starting from the identity transform. SPG looks for the solution in a given interval. In our case, the constraints were $-\pi/2 \leq \alpha \leq \pi/2$ and $-0.5 \leq t_1, t_2 \leq 0.5$ for all fragments, after coordinate normalization.

Real images always have a segmentation error, but the proposed method still provides good results. When the distortion is more complex (e.g. the objects are rotated more than $\pi/2$), then our algorithm may provide geometrically incorrect results, since the system of equations is solved by a nonlinear optimization procedure. Next, we note that some tiles are slightly overlapping in Fig. 5.8. This is because overlaps are invisible to the system of equations. Nevertheless, overlaps could be prevented by checking the transformed fragments for each iteration, but this is a rather time-consuming procedure.

Needless to say, we could always apply an affine model as an approximation of the actual plane projective transformation acting between the shapes (see Fig. 5.9). But in general, the special transformation class has to be applied because in some cases, where the more general transformation class is considered, the method provides results that are algebraically correct, but in the geometrical sense they are completely wrong. For

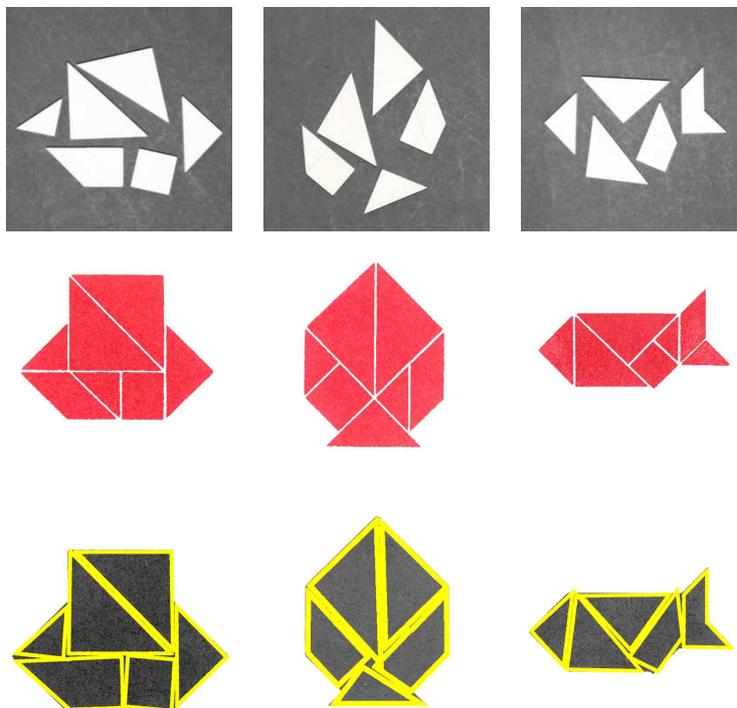


Figure 5.9: Solutions of the Tangram puzzle, where an affine model is used. **Top:** *observations* were taken by a digital camera. **Middle:** solutions found in the Tangram manual. **Bottom:** the scanned *template* silhouettes with overlaid contours of aligned fragments.

example the solver could provide some nearly singular transformations, which results in vanishing tiles, and obviously this cannot be a geometrically correct result. The δ error (*i.e.* overlap between the *template* and *realigned* image) is small, but in a geometrical sense this cannot occur.

Realigning Bone Fractures

Complex bone fracture reduction frequently requires surgical care, especially when the angulation or displacement of bone fragments are large. In such situations, computer-aided surgical planning [23] is done before the actual surgery takes place, which allows the doctor to gather more information about the dislocation of the fragments and to arrange and analyze the surgical implants to be inserted. A crucial part of such a system is the relocation of bone fragments to their original anatomic position. Since the input data is typically a volume CT image, this repositioning has to be performed in 3D space, which requires an expensive special 3D haptic device and quite a lot of manual work. Therefore automatic bone fracture reduction can save considerable time and provide experts with a rough alignment that can be manually fine-tuned according

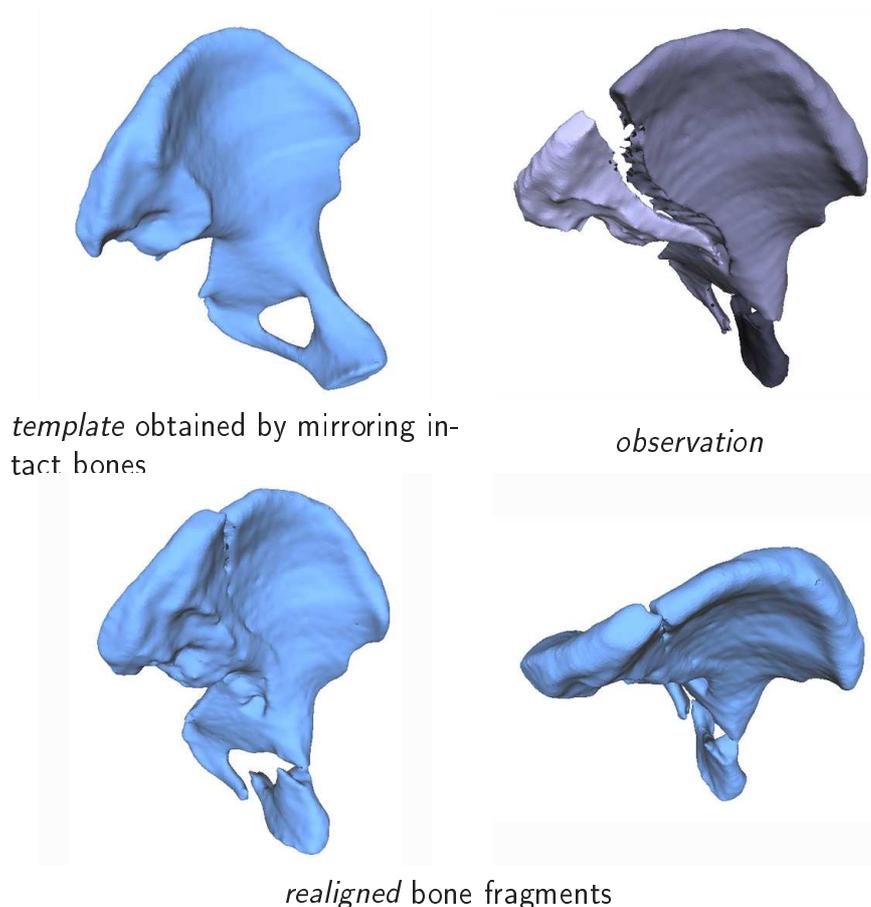


Figure 5.10: Bone fracture reduction. The CPU time was 15 sec. for these 1 megavoxel CT volumes.

to anatomic requirements.

Since surgical planning involves the bio-mechanical analysis of the bone with implants, only rigid-body transformations are allowed. In [23], a classical ICP algorithm is used to realign fractures. Winkelbach *et al.* [111] proposed an approach for estimating the relative transformations between fragments of a broken cylindrical structure by using well-known surface registration techniques like 2D depth correlation and the ICP algorithm. In [82], registration is solved by using a quadrature filter phase difference to estimate local displacements.

Now we shall apply our puzzle framework to reduce images of pelvic fractures using 3D rigid-body transformations. In the case of single side fractures, the *template* is simply obtained by mirroring intact bones of the patient. Fig. 5.10 shows a typical result for a pelvic fracture with three fragments. The main challenges are segmentation errors and, due to the variability of the human body, a slightly different *template*. In spite of these difficulties, the alignment of larger parts is quite accurate and only the small fragment has a noticeable alignment error. Since the error caused by a misplaced small piece is relatively low, the solver may not find the best transformation. However, as human verification and correction of the result is needed anyway in a real surgical planning system, these small errors are not critical and can be easily corrected manually.

5.4 Summary

A novel framework to solve shape realignment problem was proposed and applied in this chapter to 2D and 3D affine and rigid-body transformations. In contrast to classical solutions based on landmark extraction and correspondences, the proposed solution finds the aligning transformations without any additional information. Essentially, the method consists of constructing a polynomial system of equations whose solution directly provides the unknown parameters. The quantitative evaluations on both 2D and 3D synthetic datasets demonstrate the performance and robustness of the method and the results obtained on real images suggest that it can be applied in various application domains. Fig. 5.8 and Fig. 5.10 show that the results obtained by the proposed method are much more influenced by the number of fragments than the size of the input images. The 3D images have larger sizes than the 2D case, but the CPU time is almost the same in both cases. The main advantages are that the proposed method does not require correspondences, it is quite fast and easy to implement.

Chapter 6

Conclusions

Binary registration is an important problem in itself for many complex image analysis tasks. In many applications, when images have highly nonlinear radiometric distortion or the radiometric information is limited, but the segmentation of the objects are available, binary image registration could be a valid alternative. The classical way to solve the registration problem is to find point correspondences between the images, then the solution of a system of equations constructed from these point correspondences provides the unknown parameters of the aligning transformation. Nevertheless, finding reliable point correspondences on binary images is quite a challenge, hence we interested in a direct solution without solving the correspondence problem.

In this chapter we draw some conclusions about the methods proposed in the previous chapters and mention some possible directions of future research. We proposed several approaches for solving the problem of the parametric estimation of affine deformations of binary images. The fundamental difference between classical image registration algorithms and ours is that our approach works without any landmark, feature detection or optimization, adopting a novel idea where the transformation is obtained as a solution of a set of equations. First, we proposed a novel approach where the exact transformation is obtained as the solution of a polynomial system of equations. Next, we introduced a method where the exact transformation was obtained from a least-squares solution of a linear system of equations constructed by fitting Gaussian densities to the shapes which preserve the effect of the unknown transformation. In the case of compound shapes, we also proposed a robust and efficient numerical scheme for achieving near real-time performance. Even though these methods use all the available information in the input images, there is no need for established correspondences. Our algorithms have a linear time complexity; are simple to implement, and run quite fast; furthermore, they yield an exact solution regardless of the size of the deformation. The experimental results showed that the proposed methods provide good alignments on both real and synthetic images; moreover, they are robust in the case of noisy observations. We also addressed the problem of simultaneously estimating different linear

(affine and rigid-body) deformations, resulting in a global nonlinear transformation between the original object and its broken fragments without correspondences. In this thesis a general framework was presented, where the solution of a polynomial system of equations directly provides the parameters of the alignment. The main drawback of the methods outlined in chapter 3 to 5 is that the occlusion is not very well tolerated, but this is a true of all *area-based* methods.

Most likely, the methods introduced by Hagege and Francos [40–42] are the closest to the methods proposed in this dissertation, since they have a similar basic idea behind them. Hagege and Francos addressed the parametric estimation of two-dimensional affine transformations between two *grey-level* images in [40–42], which provides an accurate and computationally simple solution that avoids both the correspondence problem and the need for optimization. The original problem was reformulated as an *equivalent linear parameter estimation* one having a *unique* and *exact* solution. However, these methods rely on the availability of rich radiometric information, which is clearly not available in the binary case.

As we described in Section 3.2.1, by making use of a fuzzy representation of a digital image we were able to improve the performance of the proposed method. This is important, as we estimated the integrals by a finite sum, and this approximation clearly influences the accuracy. In [97], the efficiency of the fuzzy representation was analyzed. The proposed framework, presented in Chapter 3 and Chapter 5, could be extended to include diffeomorphic deformations. However, in the case of diffeomorphic deformations the unknown parameters of the deformation can be obtained via a least-squares solution, which makes the problem more difficult, since the Jacobian is not a simple constant in the nonlinear case. In [76] a novel approach is proposed to estimate the parameters of a diffeomorphism that aligns two binary images. The problem is traced back to the solution of a system of nonlinear equations which directly provides the parameters of the aligning transformation. The basic idea is similar to that described in Section 3. Naturally, recovering the parameters of planar homographies is a fundamental problem in computer vision with various applications. The method considered in [77] assumes a planar homography that aligns two binary images. Here the solution is also obtained by solving a system of nonlinear equations generated by integrating linearly independent functions over the domains determined by the shapes. These methods [76; 77] have the similar advantages to the method proposed here.

Appendix A

Summary in English

Image registration is a crucial step in almost every image processing application where images of different views or sensors of an object need to be compared or combined. Typical application areas include object recognition, target tracking in video sequences, monitoring land usage on satellite images, super resolution, image mosaicking, and medical image analysis. In a general setting, one is looking for a transformation which aligns two images such that one image (called the *observation*) becomes similar to the second one (called the *template*). Due to the large number of possible transformations, the problem is inherently *ill-defined* unless this variability is taken into account.

In many situations, the variability of image features is so complex that the only feasible way to register such images is to reduce them to a binary representation and solve the registration problem in that context. X-ray images are good examples as they usually exhibit highly nonlinear radiometric distortions, making registration hard to solve. If perfect greylevel images were available, the estimation of an aligning transformation could be reduced to solving a linear system of equations. In real applications, however, such a strict requirement cannot be satisfied. Nevertheless, registration can be solved without making use of any intensity information. In real situations, the images obtained are related by a projective transformation (also called planar homography). Although projective transformations are nonlinear, they can often be successfully modelled by an affine transformation, which is linear. Owing to its linear property, the affine transformation is of great importance in image registration.

The chief aim of this dissertation was to present several techniques developed by the author which can be used to estimate the parameters of the affine deformations of binary images. The realignment problem of deformed shape fragments is also considered, where the basic ideas of the registration methods is used. In contrast to the classical approaches, the methods presented here work without any landmark, feature detection or optimization and use all the available information in the input images. They adopt a novel idea where the aligning transformation is directly obtained from a solution of a polynomial or a linear system of equations. The main advantages of the proposed

algorithms are that they have a linear time complexity; are simple to implement, and run quite fast; furthermore, in the case of registration they yield an exact solution irrespective of the size of the deformation.

A.1 Summary by Chapters

In Chapter 2 we give a brief overview of the area of binary image registration. This review includes the state-of-the-art image registration methods and the best, relevant binary image registration techniques. This chapter also presents the preliminary notations used in the image processing field and the key ideas behind the proposed methods that are used in the subsequent chapters.

The next two chapters address the problem of the parameter estimation of a global affine transformation of binary shapes. These chapters present novel methods which provide accurate and computationally simple solutions to the affine registration of shapes. Chapter 3 shows how the binary registration problem can be formulated as the solution of a system of polynomial equations obtained by integrating a set of polynomial functions over the shape domains. This novel method provides a direct solution without established correspondences or optimization. Moreover, the robustness of the resulting algorithm in the presence of i.i.d. Gaussian noise on the point coordinates and also segmentation errors is demonstrated. Comparative tests on partially occluded shapes reveal, however, that other *area-based* state-of-the-art methods cannot cope with occlusion either. Its performance on real images is demonstrated and the method is applied to align pairs of hip prosthesis X-ray images. The advantage of the proposed solution is that it has a linear time complexity, is fast, easy to implement, works without established correspondences and it provides an exact solution regardless of the magnitude of the transformation.

The main difficulty with binary images is that they do not contain radiometric information; just the foreground pixel coordinates are available for the registration algorithm. Chapter 4 shows that in spite of the absent radiometric information, the registration problem can still be formulated as the solution of a linear system of equations, where all the available geometric information is used. The basic idea is to generate a pair of *covariant functions* that are related by the unknown transformation. The main contribution is the construction of these relations between shapes without establishing correspondences of any kind. The unknown transformation parameters are then found from the *least-squares* solution of an overdetermined linear system of equations. When the images contain compound shapes, an elegant and robust solution is proposed, where linear equations are constructed by integrating nonlinear functions over corresponding domains derived from compound shapes. This method is more robust and numerically more efficient than the previous approach, when images contain just a single object.

The performance of the proposed method was tested on a large synthetic dataset as well as on real images. The resulting algorithm is fast and provides a direct solution without establishing correspondences.

In Chapter 5 we propose a general framework to solve the realignment problem of deformed shape fragments without fixing the dimension of the input images (generally $n = 2, 3$). Given a *template* image with a set of shapes, and their affine distorted versions on the *observation* image, we would like establish a geometric correspondence between these images. This chapter introduces our registration problem followed by the basic ideas of the general solution with the assumption that only the overall segmentation of the *template* is known, *i.e.* its partitioning is hidden. The proposed method directly provides the parameters of the alignment by solving a polynomial system of equations. We also conduct an extended analysis of the numerical stability of the proposed algorithm. The performance and robustness of our method are investigated using large 2D and 3D synthetic benchmark datasets and also in different real applications contexts (*e.g.* 2D Tangram, bone fracture alignment). In contrast to classical solutions based on landmark extraction and correspondences, the proposed solution finds the aligning transformations without any additional information. The main advantages are that the proposed method does not require correspondence, it is quite fast and easy to implement.

In Chapter 6 we draw some pertinent conclusions and discuss some interesting directions for future research.

A.2 Key Points of the Thesis

In the following a listing of the most important results of the dissertation is given. Table A.1 shows which thesis point is described in which publication by the author.

- I.) The author addresses the problem of the estimation of affine transformations for aligning a known 2D shape and its distorted observation. He proposes a novel approach where the exact transformation is obtained as the solution of a polynomial system of equations. He tested the proposed method on synthetic as well as on real images. The author demonstrated the robustness of the algorithm in the presence of segmentation errors and additive geometric noise too, then he successfully applied the method to the registration of hip prosthesis X-ray images.
- II.) The author proposes a novel approach for the estimation of 2D affine transformations for aligning a planar shape and its distorted observation. He shows how the exact transformation is obtained as a least-squares solution of a linear system of equations constructed by fitting Gaussian densities to the shapes which preserve the effect of the unknown transformation. In the case of compound shapes, the

author also presents a robust and efficient numerical scheme and achieves a near real-time performance. He tested the method on synthetic as well as on real images and demonstrated its robustness in the case of missing pixels, boundary errors, and modelling errors.

- III.) The author considers the problem of realigning broken objects without correspondences, where the segmentation of the overall *template* is known, but the segmentation of the object parts is unknown. The author applies linear transformations between the object fragments and presents the method by using 2D and 3D affine transformations. He shows that constructing and solving a polynomial system of equations provides the unknown parameters of the alignment. Here he quantitatively evaluated the proposed algorithm on a large synthetic dataset containing 2D and 3D images. He analyzes the numerical stability of the method as well as its robustness against segmentation errors. He also presents the results of experiments on 2D real images as well as on volumetric medical images applied to surgical planning.

	[21]	[20]	[16]	[17]	[19]	[18]
I.	•	•				
II.			•	•	•	
III.						•

Table A.1: The connection between the thesis points and the corresponding publications.

Appendix B

Summary in Hungarian

A képregisztráció majdnem minden képfeldolgozási alkalmazás fontos lépése, ahol az objektumok eltérő nézőpontból vagy különböző szenzorral készített képeit kell összehasonlítani esetleg kombinálni. Tipikus felhasználási területei közé tartozik az alakfelismerés, objektumok követése videószekvenciákon, műholdképek változásaink detektálása, szuperfelbontású képek, képmozaikozás, illetve az orvosi képfeldolgozás. Általánosan a feladat egy olyan transzformáció megkeresése, mely az egyik képet (*megfigyelés*) a második képhez illeszti (*sablon*). A lehetséges transzformációk nagy száma miatt a probléma *rosszul definiált*, hacsak nem vesszük figyelembe ezt a nagy változatosságot.

Sok esetben a képjellemzők változása annyira összetett, hogy az egyetlen járható út a képek regisztrálására, hogy a bináris reprezentációjuk alapján oldjuk meg a problémát. A röntgen képek esete jó példa arra, hogy egy erősen nemlineáris színtorzulás megnehezíti a regisztrációt. Ha a képek tökéletes szürkeárnyaltos változata adott lenne, akkor az illesztő transzformáció paramétereinek becslése visszavezethető lenne egy lineáris egyenletrendszer megoldására [40; 42]. Mindazonáltal valós alkalmazások során ilyen erős feltétel ritkán teljesül. Itt egy olyan regisztrációs eljárást mutatunk be, amely a kép intenzitásértékeinek felhasználása nélkül oldja meg a feladatot. A képek között valós esetben projektív transzformáció van (melyet síkhomográfiának is hívnak). Bár a projektív transzformációk nemlineárisak, gyakran affin transzformációkkal jól modellezhetők, melyek viszont már lineárisak. Emiatt az affin transzformációk fontos szerepet játszanak a képregisztrációban.

A disszertáció legfontosabb célja, hogy bemutassa a Szerző bináris képek affin transzformációinak paraméterbecslésére vonatkozó kutatási eredményeit. Széttört alakzatok helyreállításának problémájával is foglalkozunk, mely során a regisztrációnál megadott alapötleteket használjuk fel. Alapvető különbség a klasszikus módszerekhez képest, hogy az itt bemutatott módszerek jellemző pontok és képjellemzők detektálása, illetve optimalizálás nélkül működnek úgy, hogy az input képeken elérhető összes információt felhasználják. Egy olyan újfajta megközelítést használnak, ahol a transzformáció közvetlenül adódik egy polinom vagy egy lineáris egyenletrendszer megoldásaként. A

javasolt módszerek legnagyobb előnye, hogy lineáris időbonyolultságúak, könnyen implementálhatók, gyorsak, továbbá a deformáció erősségétől függetlenül megkapjuk a regisztráció pontos megoldását.

B.1. A fejezetek áttekintése

A 2. fejezetben egy rövid áttekintést adunk a bináris képregisztrációról. Ez az áttekintés a jelenlegi regisztrációs módszereket és a legfontosabb, releváns bináris regisztrációs technikákat tartalmazza. A fejezet bemutatja a regisztráció megoldása során alkalmazott jelöléseket, illetve a javasolt módszerek alapötletét, melyet a további fejezetekben használni fogunk.

A következő két fejezet bináris alakzatok globális affin transzformációinak paraméterbecslésével foglalkozik. Ezen fejezetek újfajta módszereket mutatnak be alakzatok affin regisztrációjára, melyek pontos és egyszerűen számítható megoldást adnak. A 3. fejezet bemutatja, hogyan fogalmazható át a bináris regisztráció problémája egy olyan polinom egyenletrendszer megoldására, melyet az alakzatok felett hatványfüggvények integrálása alapján írunk fel. Ez a módszer pontmegfeleltetés és optimalizálás nélkül, direkt megoldást ad. Továbbá szemléltetjük az algoritmus robusztusságát szegmentálási hiba, illetve a koordinátákon értelmezett additív geometriai zaj esetén is. Az összehasonlító tesztek mutatják, hogy a takarást a többi *terület alapú* módszer sem képes jól kezelni. Az algoritmus hatékonyságát valós képeken és csípőprotézisről készült röntgen képek regisztrációja során is szemléltetjük. A javasolt módszer előnye, hogy lineáris időbonyolultságú, gyors, könnyen implementálható, nem igényel megfeleltetéseket, továbbá a deformáció erősségétől függetlenül megkapjuk a pontos megoldást.

Bináris képek esetén a legnagyobb nehézséget az okozza, hogy nem tartalmaznak színinformációt, így a regisztrációs algoritmus csupán az előtérpixelek koordinátáját tudja felhasználni. A 4. fejezet bemutatja, hogy a színinformáció hiányának ellenére, a regisztrációs probléma átfogalmazható egy lineáris egyenletrendszer megoldására, ahol a képen elérhető összes információt felhasználjuk. Az alapötlet olyan *kovariáns függvények generálása*, melyek az ismeretlen transzformáció alapján kapcsolódnak egymáshoz. Egyik legfontosabb eredményünk, hogy képesek vagyunk megkonstruálni ilyen összefüggéseket az alakzatok közötti bármiféle megfeleltetés felhasználása nélkül. Az ismeretlen transzformáció paraméterei ezután egy túlhatározott lineáris egyenletrendszer legkisebb négyzetes értelemben vett megoldásaként adódnak. Ha a képek több részből álló alakzatokat tartalmaznak, akkor elegáns és robusztus megoldás adható, ahol a lineáris egyenletrendszert az összetett alakzat részei felett kiintegrált nemlineáris függvények alapján kapjuk meg. Ez a módszer robusztus és numerikusan is hatékonyabb, mint az előző megközelítés, ahol a képek csupán egyetlen részből álló objektumot tartalmaznak. A módszer hatékonyságát nagyméretű szintetikus generált adatbázison

és valós képeken is teszteltük. A megadott algoritmus gyors, továbbá direkt megoldást ad megfeleltetések felhasználása nélkül.

Az 5. fejezetben egy olyan általános keretrendszert mutatunk be, mely törött alakzatok visszaállításának problémáját oldja meg anélkül, hogy az input képek dimenzióját rögzíteni kellene (mely általában $n = 2, 3$). Legyen adott egy *sablon* kép alakzatok egy halmazával, illetve ezek affin transzformált változatai a *megfigyelés* képen, ekkor a képek közötti geometriai kapcsolatot szeretnénk meghatározni. A fejezet először bemutatja a regisztrációs problémát, majd pedig az általános megoldás alapötletét, mely során feltételezzük, hogy a *sablon* kép teljes szegmentálása nem ismert, azaz partíciónálása rejtett. A javasolt módszer direkt módon, egy polinom egyenletrendszer megoldásával adja meg az illesztési transzformáció paramétereit. A javasolt algoritmus numerikus stabilitását bővebben elemezzük. A módszerünk hatékonyságát és robusztusságát nagyméretű 2D és 3D szintetikus képeket tartalmazó adatbázison, illetve különböző valós alkalmazások esetén is teszteltük (pl. 2D Tangram, törött csontok visszaállítása). A klasszikus megoldásokkal szemben, melyek jellemző pontok kinyerésén vagy megfeleltetéseken alapulnak a javasolt módszer minden további információ felhasználása nélkül adja meg az illesztő transzformáció paramétereit. Legfontosabb előnyei közé tartozik, hogy nem igényel megfeleltetéseket, gyors és könnyen implementálható.

A 6. fejezetben a legfontosabb megállapításokat emeljük ki, illetve az érdekesebb jövőbeli kutatási irányokat elemezzük.

B.2. Az eredmények tézisszerű összefoglalása

Az alábbiakban három tézispontba rendezve foglaljuk össze a Szerző legfontosabb eredményeit. A B.1. táblázat mutatja, az egyes tézispontokhoz, mely publikációk kapcsolódnak.

- I.) A Szerző egy olyan módszert mutat be, mely egy ismert alakzat és annak deformált változatai között lévő affin transzformációk paramétereinek becslésére alkalmas. Egy olyan újfajta megközelítést javasol, ahol a pontos transzformáció egy polinom egyenletrendszer megoldásával megkapható. A módszert szintetikus és valós képeken is tesztelte. A Szerző bemutatta a módszer robusztusságát szegmentálási hiba és additív geometriai zaj esetén, majd pedig azt, hogy a módszer sikeresen alkalmazható csípőprotézisről készült röntgen képek regisztrációjára.
- II.) A Szerző egy olyan újfajta megközelítést mutat be, mely egy ismert alakzat és annak deformált változatai között lévő affin transzformációk paramétereinek becslésére alkalmas. A Szerző megmutatja, hogyan kapható meg a pontos transzformáció az alakzatok felett megkonstruált Gauss sűrűségfüggvények (melyek megőrzik az ismeretlen transzformáció hatását) alapján felírt lineáris egyenletrendszer

legkisebb négyzetes értelemben vett megoldásával. Több részből álló alakzatok esetén a Szerző egy robusztus és numerikusan hatékony megoldást mutat be, mellyel közel valós idejű hatékonyság érhető el. A módszert mind szintetikus, mind valós képeken tesztelte. A robusztusság mellett hiányzó pixelek, kontúr menti hiba és modellezési hiba esetén is szemléltette a módszer működését.

- III.) A Szerző széttört objektumok összeillesztésének problémáját vizsgálja pontmegfeleltetések felhasználása nélkül, ahol csak a teljes *sablon* kép szegmentálása ismert, az egyes alakzatoké nem. A Szerző a széttört objektumok között lineáris transzformációkat feltételez, és a módszert 2D és 3D affin transzformációk esetén mutatja be. Megmutatja, hogyan lehet megkonstruálni egy olyan polinom egyenletrendszert, melynek megoldása megadja az ismeretlen illesztési transzformáció paramétereit. A javasolt algoritmust nagyméretű szintetikus adatbázison tesztelte, mely 2D és 3D képeket is tartalmaz. A robusztusság és a módszer szegmentálási hibákra való érzékenysége mellett a módszer numerikus stabilitását is elemzi. Az eredményeket 2D valós képeken és egy 3D orvosi alkalmazás esetében (műtéti tervezés) is bemutatta.

	[21]	[20]	[16]	[17]	[19]	[18]
I.	•	•				
II.			•	•	•	
III.						•

B.1. táblázat. A tézispontokhoz kapcsolódó publikációk.

Bibliography

- [1] Pedro M. Q. Aguiar. Unsupervised simultaneous registration and exposure correction. In *Proceedings of International Conference on Image Processing*, pages 361–364, Atlanta, GA, USA, October 2006. IEEE.
- [2] Francesco Amigoni, Stefano Gazzani, and Simone Podico. A method for re-assembling fragments in image reconstruction. In *Proceedings of International Conference on Image Processing*, volume 2, pages 581–584, Barcelona, Spain, September 2003. IEEE.
- [3] Vincent Arsigny, Olivier Commowick, Nicholas Ayache, and Xavier Pennec. A fast and log-euclidean polyaffine framework for locally linear registration. *Journal of Mathematical Imaging and Vision*, 33(2):222–238, February 2009.
- [4] K. Somani Arun, Thomas S. Huang, and Steven D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, September 1987.
- [5] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, August 2008.
- [6] Serge Belongie, Jitendra Malik, and Jan Puzicha. Matching with shape contexts. software, April 2002. http://www.eecs.berkeley.edu/Research/Projects/CS/vision/shape/sc_digits.html.
- [7] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape context. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.
- [8] Paul J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

- [9] Ernesto G. Birgin, José Mario Martínez, and Marcos Raydan. SPG: Software for convex-constrained optimization. *ACM Transactions on Mathematical Software*, 27(3):340–349, September 2001.
- [10] Thomas Boudier. The snake plugin for imageJ. Software. <http://www.snv.jussieu.fr/~wboudier/softs/snake.html>.
- [11] Alex M. Bronstein and Michael M. Bronstein. Regularized partial matching of rigid shapes. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Proceedings of the European Conference on Computer Vision*, volume 5303 of *Lecture Notes in Computer Science*, pages 143–154, Marseille, France, October 2008. Springer.
- [12] Alexander M. Bronstein, Michael M. Bronstein, Ron Kimmel, Mona Mahmoudi, and Guillermo Sapiro. A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *International Journal of Computer Vision*, 89(2-3):266–286, September 2010.
- [13] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, December 1992.
- [14] Ananda S. Chowdhury, Suchendra M. Bhandarkar, Robert W. Robinson, and Jack C. Yu. Virtual multi-fracture craniofacial reconstruction using computer vision and graph matching. *Computerized Medical Imaging and Graphics*, 33(5):333–342, July 2009.
- [15] Helena Cristina da Gama Leitão and Jorge Stolfi. A multiscale method for the reassembly of two-dimensional fragmented objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1239–1251, September 2002.
- [16] Csaba Domokos and Zoltan Kato. Binary image registration using covariant Gaussian densities. In Aurélio Campilho and Mohamed Kamel, editors, *Proceedings of International Conference on Image Analysis and Recognition*, volume 5112 of *Lecture Notes in Computer Science*, pages 455–464, Póvoa de Varzim, Portugal, June 2008. Springer.
- [17] Csaba Domokos and Zoltan Kato. Affine alignment of compound objects: A direct approach. In *Proceedings of International Conference on Image Processing*, pages 169–172, Cairo, Egypt, November 2009. IEEE.
- [18] Csaba Domokos and Zoltan Kato. Affine puzzle: Realigning deformed object fragments without correspondences. In Kostas Daniilidis, Petros Maragos, and Nikos

- Paragios, editors, *Proceedings of European Conference on Computer Vision*, volume 6312 of *Lecture Notes in Computer Science*, pages 777–790, Heraklion, Crete, Greece, September 2010. Springer.
- [19] Csaba Domokos and Zoltan Kato. Affine shape alignment using covariant Gaussian densities: A direct solution. *IEEE Transactions on Image Processing*, 2010. Submitted.
- [20] Csaba Domokos and Zoltan Kato. Parametric estimation of affine deformations of planar shapes. *Pattern Recognition*, 43(3):569–578, March 2010.
- [21] Csaba Domokos, Zoltan Kato, and Joseph M. Francos. Parametric estimation of affine deformations of binary images. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, pages 889–892, Las Vegas, NV, USA, April 2008. IEEE.
- [22] M.R. Downing, P.E. Undrill, P. Ashcroft, D.W.L. Hukins, and J.D. Hutchison. Automated femoral measurement in total hip replacement radiographs. In *Proceedings of International Conference on Image Processing and Its Applications*, volume 2, pages 843–847, Dublin, Ireland, July 1997. IEEE.
- [23] Balázs Erdőhelyi and Endre Varga. Semi-automatic bone fracture reduction in surgical planning. In *Proceedings of the International Conference on Computer Assisted Radiology and Surgery*, volume 4 of *International Journal of Computer Assisted Radiology and Surgery*, pages S98–S99, Berlin, Germany, June 2009. Springer.
- [24] Jacques Feldmar and Nicholas Ayache. Rigid, affine and locally affine registration of free-form surfaces. *International Journal of Computer Vision*, 18(2):99–119, May 1996.
- [25] Andrew W. Fitzgibbon. Robust registration of 2D and 3D point sets. *Image and Vision Computing*, 21(13):1145–1153, December 2003.
- [26] Corneliu Florea, Constantin Vertan, and Laura Florea. Logarithmic model-based dynamic range enhancement of hip X-ray images. In Jacques Blanc-Talon, Wilfried Philips, Dan Popescu, and Paul Scheunders, editors, *Proceedings of International Conference on Advanced Concepts for Intelligent Vision Systems*, volume 4678 of *Lecture Notes in Computer Science*, pages 587–596, Delft, Netherlands, August 2007. Springer.
- [27] Jan Flusser, Jaroslav Kautsky, and Filip Šroubek. Implicit moment invariants. *International Journal of Computer Vision*, 86(1):72–86, January 2010.

- [28] Jan Flusser and Tomáš Suk. Pattern recognition by affine moment invariants. *Pattern Recognition*, 1(1):167–174, January 1993.
- [29] Jan Flusser and Tomáš Suk. A moment-based approach to registration of images with affine geometric distortion. *IEEE Transactions on Geoscience and Remote Sensing*, 32(2):382–387, March 1994.
- [30] Jan Flusser and Tomáš Suk. Projective moment invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1364–1367, October 2004.
- [31] Jan Flusser, Tomáš Suk, and Barbara Zitová. *Moments and Moment Invariants in Pattern Recognition*. A John Wiley & Sons, October 2009.
- [32] Massimo Fornasier and Domenico Toniolo. Fast, robust and efficient 2D pattern recognition for re-assembling fragmented images. *Pattern Recognition*, 38(11):2074–2087, November 2005.
- [33] Hassan Foroosh, Josiane B. Zerubia, and Marc Berthod. Extension of phase correlation to subpixel registration. *IEEE Transactions on Image Processing*, 11(3):188–200, March 2002.
- [34] Alban Foulonneau, Pierre Charbonnier, and Fabrice Heitz. Multi-reference shape priors for active contours. *International Journal of Computer Vision*, 81(1):68–81, January 2009.
- [35] Rui Gan and Albert C. S. Chung. Distance-intensity for image registration. In *Proceedings of Computer Vision for Biomedical Image Applications*, volume 3765 of *Lecture Notes in Computer Science*, pages 281–290, Beijing, China, October 2005. Springer.
- [36] David Goldberg, Christopher Malon, and Marshall Bern. A global approach to automatic solution of Jigsaw puzzles. *Computational Geometry*, 28(2–3):165–174, June 2004.
- [37] Arthur Ardeshir Goshtasby, George C. Stockman, and Carl V. Page. A region-based approach to digital image registration with subpixel accuracy. *IEEE Transactions on Geoscience and Remote Sensing*, GE-24(3):390–399, May 1986.
- [38] Venu Govindu and Chandra Shekhar. Alignment using distributions of local geometric properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):1031–1043, October 1999.
- [39] Hongyu Guo, Anand Rangarajan, Sarang C. Joshi, and Laurent Younes. Non-rigid registration of shapes via diffeomorphic point matching. In *Proceedings of*

- IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, volume 1, pages 924–927, Arlington, VA, USA, April 2004. IEEE.
- [40] Rami Hagege and Joseph M. Francos. Parametric estimation of two dimensional affine transformations. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 305–308, Montreal, May 2004. IEEE.
- [41] Rami Hagege and Joseph M. Francos. Parametric estimation of multi-dimensional affine transformations: An exact linear solution. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 861–864, Philadelphia, USA, March 2005. IEEE.
- [42] Rami Hagege and Joseph M. Francos. Linear estimation of sequences of multi-dimensional affine transformations. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 785–788, Toulouse, France, May 2006. IEEE.
- [43] Rami R. Hagege and Joseph M. Francos. Estimation of affine geometric transformations of several objects from global measurements. In *Proceedings of IEEE International Workshop on Multimedia Signal Processing*, pages 1–5, Rio de Janeiro, Brazil, October 2009.
- [44] Michael Sass Hansen, Mads Fogtmann Hansen, and Rasmus Larsen. Diffeomorphic statistical deformation models. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1–8, Rio de Janeiro, Brazil, October 2007.
- [45] Kevin Hardinge, Martyn L. Porter, Peter R. Jones, David W. L. Hukins, and Christopher J. Taylor. Measurement of hip prostheses using image analysis. The maxima hip technique. *Journal of Bone and Joint Surgery*, 73-B(5):724–728, September 1991.
- [46] Janne Heikkilä. Pattern matching with affine moment descriptors. *Pattern Recognition*, 37(9):1825–1834, September 2004.
- [47] Derek L. G. Hill, Philipp G. Batchelor, Mark Holden, and David J. Hawkes. Medical image registration. *Physics in Medicine and Biology*, 46(3):R1–R45, March 2001.
- [48] Fawaz Hjouj and David W. Kammler. Identification of reflected, scaled, translated, and rotated objects from their radon projections. *IEEE Transactions on Image Processing*, 17(3):301–310, March 2008.

- [49] Jeffrey Ho, Ming-Hsuan Yang, Anand Rangarajan, and Baba Vemuri. A new affine registration algorithm for matching 2D point sets. In *Proceedings of IEEE Workshop on Applications of Computer Vision*, pages 25–31, Austin, TX, USA, February 2007. IEEE.
- [50] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, February 1962.
- [51] Qi-Xing Huang, Simon Flöry, Natasha Gelfand, Michael Hofer, and Helmut Pottmann. Reassembling fractured objects by geometric matching. *ACM Transactions on Graphics*, 25(3):569–578, July 2006.
- [52] Paresh Jain and C. V. Jawahar. Homography estimation from planar contours. In *Proceedings of International Symposium on 3D Data Processing, Visualization and Transmission*, pages 877–884, Chapel Hill, NC, USA, June 2006.
- [53] Bing Jian and Baba C. Vemuri. A robust algorithm for point set registration using mixture of gaussians. In *Proceedings of IEEE International Conference on Computer Vision*, volume 2, pages 1246–1251, Beijing, China, October 2005. IEEE.
- [54] Alexander Kadyrov and Maria Petrou. Affine parameter estimation from the Trace transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1631–1645, October 2006.
- [55] Martin Kampel and Robert Sablatnig. 3D puzzling of archaeological fragments. In *Proceedings of Computer Vision Winter Workshop*, pages 31–40, Piran, Slovenia, February 2004.
- [56] Shunichi Kaneko, Yutaka Satoh, and Satoru Igarashi. Using selective correlation coefficient for robust image registration. *Pattern Recognition*, 36(5):1165–1173, May 2003.
- [57] Juho Kannala, Esa Rahtu, Janne Heikkilä, and Mikko Salo. A new method for affine registration of images and point sets. In *Proceedings of Scandinavian Conference on Image Analysis*, volume 3540 of *Lecture Notes in Computer Science*, pages 224–234, Joensuu, Finland, June 2005. Pattern Recognition Society of Finland, Springer.
- [58] Jeongtae Kim and Jeffrey A. Fessler. Intensity-based image registration using robust correlation coefficients. *IEEE Transactions on Medical Imaging*, 23(11):1430–1444, November 2004.

- [59] Stefan Klein, Josien P. Pluim, Marius Staring, and Max A. Viergever. Adaptive stochastic gradient descent optimisation for image registration. *International Journal of Computer Vision*, 81(3):227–239, March 2009.
- [60] Reinhard Klette and Joviša Žunić. Towards experimental studies of digital moment convergence. Technical Report CITR-TR-61, Computer Science Department of The University of Auckland CITR at Tamaki Campus, Auckland, New Zealand, June 2000.
- [61] Reinhard Klette and Joviša Žunić. Multigrid convergence of calculated features in image analysis. *Journal of Mathematical Imaging and Vision*, 13(3):173–191, December 2004.
- [62] Weixin Kong and Benjamin. B. Kimia. On solving 2D and 3D puzzles using curve matching. In *Proceedings of Computer Vision and Pattern Recognition*, volume 2, pages 583–590, Kauai, HI, USA, December 2001. IEEE.
- [63] M. Pawan Kumar, Sujit Kuthirummal, C. V. Jawahar, and P. J. Narayanan. Planar homography from Fourier domain representation. In *Proceedings of International Conference on Signal Processing and Communications*, pages 560–564, Bangalore, India, December 2004. IEEE.
- [64] Vincent Lepetit and Pascal Fua. Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1465–1479, September 2006.
- [65] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):1573–1405, November 2004.
- [66] J. B. Antoine Maintz and Max A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, March 1998.
- [67] Siddharth Manay, Daniel Cremers, Byung-Woo Hong, Anthony J. Yezzi Jr., and Stefano Soatto. Integral invariants for shape matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1602–1618, October 2006.
- [68] Steve Mann and Rosalind W. Picard. Video orbits of the projective group a simple approach to featureless estimation of parameters. *IEEE Transactions on Image Processing*, 6(9):1281–1295, September 1997.
- [69] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, June 1963.

- [70] Jonah C. McBride and Benjamin B. Kimia. Archaeological fragment reconstruction using curve-matching. In *Proceedings of International Conference on Computer Vision and Pattern Recognition Workshop*, volume 1, page 3, Madison, WI, USA, June 2003.
- [71] Robert A. McLaughlin, John Hipwell, David J. Hawkes, J. Alison Noble, James V. Byrne, and Tim Cox. A comparison of 2D-3D intensity-based registration and feature-based registration for neurointerventions. In *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention*, volume 2489 of *Lecture Notes in Computer Science*, pages 517–524, Tokyo, Japan, September 2002. Springer.
- [72] Graham McNeill and Sethu Vijayakumar. Hierarchical procrustes matching for shape retrieval. In Bob Werner, editor, *Proceedings of Computer Vision and Pattern Recognition*, volume 1, pages 885–894, New York, NY, USA, June 2006. IEEE.
- [73] Russell Merris. *Combinatorics*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, 2nd edition, August 2003. ISBN: 047126296X.
- [74] Oliver Musse, Fabrice Heitz, and Jean-Paul Armspach. Topology preserving deformable image matching using constrained hierarchical parametric models. *IEEE Transactions on Image Processing*, 10(7):1081–1093, July 2001.
- [75] Andriy Myronenko, Xubo Song, and Miguel Á. Carreira-Perpián. Non-rigid point set registration: Coherent point drift. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Proceedings of Conference on Neural Information Processing Systems*, pages 1009–1016, Vancouver, Canada, December 2006. MIT Press.
- [76] Jozsef Nemeth, Csaba Domokos, and Zoltan Kato. Nonlinear registration of binary shapes. In *Proceedings of International Conference on Image Processing*, pages 1101–1104, Cairo, Egypt, November 2009. IEEE.
- [77] Jozsef Nemeth, Csaba Domokos, and Zoltan Kato. Recovering planar homographies between 2D shapes. In *Proceedings of International Conference on Computer Vision*, pages 2170–2176, Kyoto, Japan, September 2009. IEEE.
- [78] Mads Nielsen, Peter Johansen, Andrew D. Jackson, Benny Lautrup, and Søren Hauberg. Brownian warps for non-rigid registration. *Journal of Mathematical Imaging and Vision*, 31(2–3):221–231, July 2008.

- [79] Alina Oprea and Constantin Vertan. A quantitative evaluation of the hip prosthesis segmentation quality in X-ray images. In *Proceedings of International Symposium on Signals, Circuits and Systems*, volume 1, pages 1–4, Iasi, Romania, July 2007. IEEE.
- [80] Georgios Papaioannou and Evaggelia-Aggeliki Karabassi. On the automatic assemblage of arbitrary broken solid artefacts. *Image and Vision Computing*, 21(5):401–412, May 2003.
- [81] Senthil Periaswamy and Hany Farid. Medical image registration with partial data. *Medical Image Analysis*, 10(3):452–464, June 2006.
- [82] Johanna Pettersson, Hans Knutsson, and Magnus Borga. Non-rigid registration for automatic fracture segmentation. In *Proceedings of International Conference on Image Processing*, pages 1185–1188, Atlanta, GA, USA, October 2006. IEEE.
- [83] Miguel S. Prieto and Alastair R. Allen. Using self-organising maps in the detection and recognition of road signs. *Image and Vision Computing*, 27(6):673–683, May 2009.
- [84] Esa Rahtu, Mikko Salo, Janne Heikkilä, and Jan Flusser. Generalized affine moment invariants for object recognition. In *Proceedings of International Conference on Pattern Recognition*, volume 2, pages 634–637, Hong Kong, China, August 2006. IEEE.
- [85] Dan Raviv, Alexander M. Bronstein, Michael M. Bronstein, and Ron Kimmel. Full and partial symmetries of non-rigid shapes. *International Journal of Computer Vision*, 89(1):19–39, August 2010.
- [86] Martin Reuter. Hierarchical shape segmentation and registration via topological features of Laplace-Beltrami eigenfunctions. *International Journal of Computer Vision*, 89(2):287–308, 2010. To appear.
- [87] Raif M. Rustamov. Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of Eurographics Symposium on Geometry Processing*, pages 225–233, Barcelona, Spain, July 2007. Eurographics, ACM SIGGRAPH.
- [88] Joaquim Salvi, Carles Matabosch, David Fofi, and Josep Forest. A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing*, 25(5):578–596, May 2007.

- [89] Alexander Shekhovtsov, Ivan Kovtun, and Václav Hlaváč. Efficient MRF deformation model for non-rigid image matching. *Computer Vision and Image Understanding*, 112(1):91–99, October 2008.
- [90] Katherine M. Simonson, Steven M. Drescher, and Franklin R. Tanner. A statistics-based approach to binary image registration with uncertainty analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):112–125, January 2007.
- [91] Maneesh Singh, Himanshu Arora, and Narendra Ahuja. Robust registration and tracking using kernel density correlation. In *Proceedings of IEEE Workshop on Image and Video Registration*, pages 174–182, Washington DC, USA, July 2004. IEEE.
- [92] Nataša Sladoje and Joakim Lindblad. Estimation of moments of digitized objects with fuzzy borders. In Fabio Roli and Sergio Vitulano, editors, *Proceedings of International Conference on Image Analysis and Processing*, volume 3617 of *Lecture Notes in Computer Science*, pages 188–195, Cagliari, Italy, September 2005. Springer.
- [93] Nataša Sladoje and Joakim Lindblad. High precision boundary length estimation by utilizing gray-level information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):357–363, February 2009.
- [94] J. Stoer and R. Bulirsch. *introduction to Numerical Analysis*, volume 12 of *Text in Applied mathematics*. Springer, third edition, 2002.
- [95] Tomáš Suk and Jan Flusser. Affine normalization of symmetric objects. In Jacques Blanc-Talon, Wilfried Philips, Dan Popescu, and Paul Scheunders, editors, *Proceedings of International Conference on Advanced Concepts for Intelligent Vision Systems*, volume 3708 of *Lecture Notes in Computer Science*, pages 100–107, Antwerp, Belgium, September 2005. Springer.
- [96] Hemant D. Tagare, David Groisser, and Oskar Skrinjar. Symmetric non-rigid registration: A geometric theory and some numerical techniques. *Journal of Mathematical Imaging and Vision*, 34(1):61–88, May 2009.
- [97] Attila Tanács, Csaba Domokos, Nataša Sladoje, Joakim Lindblad, and Zoltan Kato. Recovering affine deformations of fuzzy shapes. In Arnt-Børre Salberg, Jon Yngve Hardeberg, and Robert Jenssen, editors, *Proceedings of Scandinavian Conference on Image Analysis*, volume 5575 of *Lecture Notes in Computer Science*, pages 735–744, Oslo, Norway, June 2009. Springer.

- [98] Attila Tanács, Nataša Sladoje, Joakim Lindblad, and Zoltan Kato. Estimation of linear deformations of 3D objects. In *Proceedings of International Conference on Image Processing*, Hong Kong, China, September 2010. IEEE. Accepted.
- [99] Michael Reed Teague. Image analysis via the general theory of moments. *Journal of the Optical Society of America*, 70:920–930, August 1980.
- [100] Corey Toler-Franklin, Benedict Brown, Tim Weyrich, Thomas Funkhouser, and Szymon Rusinkiewicz. Multi-feature matching of fresco fragments. Technical Report TR-874-10, Department of Computer Science, Princeton University, Princeton, NJ, USA, May 2010.
- [101] Zhuowen Tu, Songfeng Zheng, and Alan Yuille. Shape matching and registration by data-driven EM. *Computer Vision and image Understanding*, 109(3):290–304, March 2008.
- [102] Göktürk Üçoluk and I. Hakki Toroslu. Automatic reconstruction of broken 3-D surface objects. *Computers & Graphics*, 23(4):573–582, August 1999.
- [103] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, April 1991.
- [104] L. van Gool, T. Moons, E. Pauwels, and A. Oosterlinck. Vision and Lie’s approach to invariance. *Image and Vision Computing*, 13:259–277, 1995.
- [105] Jose A. Ventura and Wenhua Wan. Accurate matching of two-dimensional shapes using the minimal tolerance zone error. *Image and Vision Computing*, 15(12):889–899, December 1997.
- [106] Jan Verschelde. Algorithm 795: PHCpack: a general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software*, 25(2):251–276, June 1999.
- [107] Jan Verschelde. PHCpack: a general-purpose solver for polynomial systems by homotopy continuation. software, July 1999.
- [108] Jan Verschelde, Pierre Verlinden, and Ronald Cools. Homotopies exploiting newton polytopes for solving sparse polynomial systems. *SIAM Journal on Numerical Analysis*, 31(3):915–930, June 1994.
- [109] Jianhua Wang and Yuncai Liu. Characteristic line of planar homography matrix and its applications in camera calibration. In *Proceedings of International Conference on Pattern Recognition*, volume 1, pages 147–150, Hong-kong, China, August 2006. IEEE.

- [110] Simon Winkelbach and Friedrich M. Wahl. Pairwise matching of 3D fragments using cluster trees. *International Journal of Computer Vision*, 78(1):1–13, June 2008.
- [111] Simon Winkelbach, Ralf Westphal, and Thomas Goesling. Pose estimation of cylindrical fragments for semi-automatic bone fracture reduction. In Bernd Michaelis and Gerald Krell, editors, *Proceedings of Annual Symposium of the German Association for Pattern Recognition*, volume 2781 of *Lecture Notes in Computer Science*, pages 556–573, Magdeburg, Germany, September 2003. Springer.
- [112] Stefan Wörz and Karl Rohr. Physics-based elastic registration using non-radial basis functions and including landmark localization uncertainties. *Computer Vision and Image Understanding*, 111(3):263–274, September 2008.
- [113] Lyubomir Zagorchev and Ardeshir Goshtasby. A comparative study of transformation functions for nonrigid image registration. *IEEE Transactions on Image Processing*, 15(3):529–538, March 2006.
- [114] Yani Zhang, Changyun Wen, and Ying Zhang. Recognition of symmetrical images using affine moment invariants in both frequency and spatial domains. *Pattern Analysis & Applications*, 5(3):316–325, August 2002.
- [115] Barbara Zitová and Jan Flusser. Image registration methods: A survey. *Image and Vision Computing*, 21(11):977–1000, October 2003.