



Shape Representation and Registration using Implicit Functions

A dissertation submitted by **Mohammad Rouhani** at
Universitat Autònoma de Barcelona to fulfil the de-
gree of **Doctor en Informàtica**.

Bellaterra, June 12, 2012

Director: **Dr. Angel D. Sappa**
Universitat Autònoma de Barcelona
Dept. Informàtica & Computer Vision Center



This document was typeset by the author using $\text{\LaTeX} 2_{\epsilon}$.

The research described in this book was carried out at the Computer Vision Center, Universitat Autònoma de Barcelona.

Copyright © 2012 by Mohammad Rouhani. All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the author.

ISBN 84-922529-8-7

Printed by Ediciones Gráficas Rey, S.L.

To my father

Abstract

Shape representation and registration are two important problems in computer vision and graphics. Representing the given cloud of points through an implicit function provides a higher level information describing the data. This representation can be more compact more robust to noise and outliers, hence it can be exploited in different computer vision application. In the first part of this thesis implicit shape representations, including both implicit B-spline and polynomial, are tackled. First, an approximation of a geometric distance is proposed to measure the closeness of the given cloud of points and the implicit surface. The analysis of the proposed distance shows an accurate estimation with smooth behavior. The distance by itself is used in a RANSAC based quadratic fitting method. Moreover, since the gradient information of the distance with respect to the surface parameters can be analytically computed, it is used in Levenberg-Marquadt algorithm to refine the surface parameters. In a different approach, an algebraic fitting method is used to represent an object through implicit B-splines. The outcome is a smooth flexible surface and can be represented in different level from coarse to fine. This property has been exploited to solve the registration problem in the second part of the thesis. In the proposed registration technique the model set is replaced with an implicit representation provided in the first part; then, the point-to-point registration is converted to a point-to-model one in a higher level. This registration error can benefit from different distance estimations to speed up the registration process even without need of correspondence search. Finally, the non-rigid registration problem is tackled through a quadratic distance approximation that is based on the curvature information of the model set. This approximation is used in a free form deformation model to update its control lattice. Then it is shown how an accurate distance approximation can benefit non-rigid registration problem.

Resum

Les representacions de forma i registre són dos problemes importants tant en la visió per computador com en els gràfics. La representació d'un núvol de punts a través d'una funció implícita proporciona major nivell d'informació alhora de descriure les dades. Aquesta representació pot ser més compacta, més robusta al soroll i als *outlier*, pel que pot ser explotada diferents aplicacions de visió per computador. La primera part d'aquesta tesi aborda representacions de forma implícites que inclouen tant la representació mitjanant *B-splines* i polinomis. Primer es proposa una aproximació per mesurar la distància geomètrica entre un núvol de punts i una superfície implícita. L'anàlisi de la distància proposada mostra una estimació acurada amb un comportament suau. Aquesta distància és usada en un algorisme d'ajustament quadràtic basat en RANSAC. A més a més, atès que la informació de gradient de la distància respecte els paràmetres de la superfície pot ser calculat analíticament, els paràmetres de la superfície poden ser refinats utilitzant l'algorisme de Levenberg-Marquadt. Seguint un enfocament diferent, un algorisme d'ajustament algebraic es pot utilitzar per representar un objecte a través de *B-splines* implícites. El resultat és una superfície suau i flexible que pot ser representada en diferents nivells de detall. Aquesta propietat ha estat explotada per solucionar el problema de registració a la segona part de la tesi. En el mètode de registració proposat, el model és substituït amb la representació implícita proposada en la primera part, i després la registració punt a punt és converteix en una registració punt a model en un nivell superior d'abstracció. Aquesta representació es pot beneficiar de diferents distàncies per accelerar el procés de registració sense haver de cercar correspondències. Finalment, el problema de registre de models no rígids és abordat mitjanant d'una aproximació de la distància quadràtica que està basada en la informació de la curvatura del conjunt de models. Aquesta aproximació s'utilitza en un model *Free Form Deformation* (FFD) per actualitzar la seva xarxa de control. Després és mostra com una aproximació acurada de la distància pot beneficiar el problema de registració no-rígida.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Objective	5
1.3	Thesis Outline	7
2	State of the art	11
2.1	Representation	11
2.1.1	Solution Space	12
2.1.2	Fitting Methodology	15
2.2	Registration	20
2.2.1	Deformation Model	21
2.2.2	Registration Distance	26
2.2.3	Optimization Method	32
3	Geometric Fitting	35
3.1	Problem Formulation	35
3.2	Approximated Residual Error	37
3.3	Implicit Polynomial Fitting	40
3.4	Experimental Results	41
3.4.1	Residual Error Approximation	42
3.4.2	Implicit Polynomial Fitting	44
3.5	Conclusions	49
4	Algebraic Fitting	53
4.1	Problem Formulation	54
4.2	Relaxed 3L-IP Fitting	58
4.2.1	Relaxing additional constrains	59
4.2.2	Convergence criteria	60
4.3	3L-IBS Fitting Method	61
4.3.1	3L-IBS	61
4.3.2	Regularization	62
4.3.3	Additional constraints	65
4.3.4	Efficient 3L-IBS Algorithm	67
4.4	Experimental Results	71
4.4.1	Relaxed 3L-IP Fitting Results	71

4.4.2	3L-IBS Fitting Results	75
4.5	Conclusion	81
5	Rigid Registration	85
5.1	Introduction	85
5.2	Proposed Approach	88
5.2.1	Distance Formulation	89
5.2.2	Distance Optimization	89
5.3	Experimental Results and Comparisons	92
5.3.1	Registration using IPs	92
5.3.2	Registration using IBSSs	96
5.4	Conclusions and Future Work	97
6	Non-Rigid Registration	105
6.1	Introduction	105
6.2	Non-rigid Registration	107
6.2.1	Registration Error (SD)	107
6.2.2	Deformation Space (FFD)	108
6.2.3	SD-FFD: A Novel Non-Rigid Registration	110
6.3	Experimental Results	111
6.4	Conclusions	114
7	Conclusions	117
7.1	Summary	117
7.2	Contributions	119
7.3	Future work	120
	Bibliography	123

List of Tables

3.1	Surfaces fitted by using different metrics; Rad: radius; C: center; AEr: Accumulated Error according to the used metric; CPU time in sec.	44
3.2	Parameters set up.	45
3.3	Synthetic data set: AFE corresponding to the illustrations presented in Fig. 3.9.	48
3.4	Data set from AIM@SHAPE: AFE corresponding to the illustrations presented in Fig. 3.11.	49
4.1	Accumulated fitting errors to compare the results obtained by different approaches (3L: the 3L algorithm [11]; GO: the Gradient One [133]; MM: Min-Max [45]; MV: Min-Var [45]; PA: Proposed Approach; GA: Geometric Approach [2]).	73
4.2	Size of solution spaces and accumulated fitting errors (AFEs) to compare the results obtained by different approaches (3L: 3L algorithm [11]; GO: Gradient One [133]; PA: Proposed Approach.	76
4.3	Low resolution 3D data sets; (POL: the size of polynomial basis; DAT: number of data points; LAT: size of control lattice; ACT: size of active control lattice.	80
4.4	CPU times (in sec.) spent for the optimization and accumulated fitting errors (AFEs) to compare the results obtained by different approaches (3L: the 3L algorithm [11]; GO: the Gradient One [133]; PA: Proposed Approach.	80
4.5	High resolution 3D data sets; (DAT: number of data points; LAT: size of control lattice; ACT: size of active control lattice; CPU: the CPU time (in sec.) spent for optimization).	81
5.1	Comparisons of registration results for 2D cases (ICP: Iterative Closest Point [10]; GMM: Gaussian Mixture Models [61] ; DT: Distance Transform [34]; GF: Gradient Flow [156]; DA: Distance Approximation [99]; PA: Proposed Approach).	95
5.2	Number of iterations of different registration methods for 2D cases.	95
5.3	Comparisons of registration results for 3D cases (ICP: Iterative Closest Point [10]; GMM: Gaussian Mixture Models [61] ; DT: Distance Transform [34]; GF: Gradient Flow [156]; DA: Distance Approximation [99]; PA: Proposed Approach).	95
5.4	Number of iterations of different registration methods for 2D cases.	96

5.5	Points in data and model sets of the 3D examples [52] presented in Fig. 5.11.	97
5.6	Comparisons of IBS registration results for 3D cases (ICP: Iterative Closest Point [10]; GMM: Gaussian Mixture Models [61]; DT: Distance Transform [34]; GF: Gradient Flow [156]; DA: Distance Approximation [99]; PA: Proposed Approach).	98
5.7	Number of iterations of different registration methods for 2D cases.	98
6.1	Points in data and model sets of the 3D examples [52] presented in Fig. 6.1 and Fig. 6.3.	112
6.2	Comparisons of non-rigid shape registration algorithms.	114

List of Figures

1.1	Different representations: (a) 3D cloud of points; (b) triangulated mesh; (c) implicit functions.	2
1.2	Different resolutions in implicit representation (the regularization parameter decreases from left-to-right and top-to-bottom).	4
1.3	Using an interface for point sets registration: (a) Initial position of data (+) and model (o) sets; (b) Data set (+) and model set represented by an IBS; (c) Registration result of data set (+) and the IBS; (d) The same result but represented by using the model set (o) and transformed data set (+).	5
1.4	Nonrigid registration of two cloud of points; the data set (pink) moves toward the model set (dark blue) while the rigidity must be controlled during the registration; triangular mesh is used to highlight the difference.	6
2.1	Fitting the <i>Bunny</i> data set containing 817 points; (<i>left</i>) result by using an implicit polynomial [2]; (<i>right</i>) result by using IBS (in both cases the object triangular mesh is also depicted to highlight the accuracy of the results). . . .	13
2.2	Basis functions construct the solution space: (<i>left</i>) RBF ; (<i>right</i>) IBS. . . .	14
2.3	Orthogonal distance from a given point p computed by means of the iterative approach [2].	17
2.4	An algebraic criterion [134] is used to approximate the distance between a set of points and the zero set.	19
2.5	FFD can deform the whole region/space through moving its control lattice. . .	23
2.6	δ -coordinates are used to encode local geometric information (illustration from [128]).	25
2.7	Shape transformation using variational implicit functions (illustration from [139]).	25
2.8	Spin images projects local neighboring points on the tangent plane (illustration from [63]).	27
2.9	Volume integral descriptor can be used to find the feature points (illustration from [41]).	27
2.10	Distance field of a 2D bunny point set and its derivatives along two coordinate axes.	29
2.11	(<i>left</i>) A local quadratic approximation of the distance field using the curvature information. (<i>right</i>) A 3D presepective of this local approximation (illustration from [99]).	30

3.1	Simplex used for estimating the geometric distance: (a) 2D case; (b) 3D case.	37
3.2	Contour of constant distance for: (a) Orthogonal distance; (b) algebraic distance; (c) [134]; (d) proposed distance estimation.	38
3.3	Convergence criteria defined as the deviation between the IP normal and the local normal is each point.	42
3.4	Estimated normal vectors for raw data points by using different approaches. (a) PGD approach. (b) Based on PCA in a local neighborhood. (c) Proposed approach (THI direction).	42
3.5	Accumulated error by using different approaches as a function of noise standard deviation.	43
3.6	Fitting a set of points from an ellipse. (a) Without noise: Algebraic (dotted line) and proposed method (solid line). (b) Noisy data case: Algebraic (dotted line) misses the elliptic structure, while the proposed approach (solid line) reaches a good result.	45
3.7	2D contours fitted by fifth (1st and 2nd rows) and sixth (3rd, 4th and 5th rows) degree IPs, results from: (a) the 3L algorithm; (b) [134]; (c) proposed approach; (d) [2], which is used as ground truths. AFE shows the accumulated fitting error respectively. The fourth row shows a case where [2] stops due to the maximum iteration criterion.	46
3.8	Fitting two concentric ellipses. (a) Result from the 3L algorithm. (b) Result from the proposed approach.	47
3.9	Synthetic data sets fitted with the proposed approach.	47
3.10	Solid surface representing a fourth degree IP; wire frame is used to visualize given data points. (a) IP obtained from the 3L algorithm. (b) Result from the proposed approach (note the similarity between wire frame and the surface from the computed IP).	48
3.11	Data set from AIM@SHAPE fitted with the proposed approach.	50
3.12	(a) Fitting with a rough initialization. (b), (c) and (d) First, second and third iterations respectively.	50
3.13	Parameter evolution of Fig. 3.12 along 25 iterations: (a) IP coefficient values; (b) AFE; (c) accumulated angle used as convergence criteria.	51
4.1	The proposed efficient 3L-IBS method for fitting; (left) three level sets; (right) the active control points.	54
4.2	Computing the normal vector: (left) using the principal component analysis; (right) using the neighboring triangle orientations in the triangular mesh.	55
4.3	Fitting a set of 2D points: (a) cat silhouette [127] (b) data set and additional offsets; (c) the fitting result by using a degree 30th implicit polynomial (496 coefficients) [2]; (d) the fitting result from the proposed algorithm using a 30×30 IBS (391 active coefficients).	56
4.4	(a) Level sets: original data (Γ_0), outer offset ($\Gamma_{-\delta}$) and inner offset ($\Gamma_{+\delta}$). (b) A 3D illustration of the original 3L algorithm. (c) The best fitting polynomial obtained by [2], showing that the values in the inner and outer sets should be relaxed.	59

4.5	(a) Coarse approximation used as an initialization of the proposed approach ($\delta = 10$ percent of the object size, AFE= 0.1989). (b) Fitting result obtained with the proposed approach (four iterations, AFE=0.1288).	60
4.6	Cubic B-spline basis functions: (top) 2D basis functions made out of blending functions; (bottom) a tensor product 3D basis functions defined on the unit square.	63
4.7	Illustrations of the fitted IBS with different regularization parameters (μ in (4.22)); resulting zero sets are highlighted.	64
4.8	Positional constraints can be applied for the proposed 3L-IBS method. (left) the original point and fitted IBS; (right) the constrained IBS passing through the three user-defined points.	66
4.9	Computationally efficient: (left) the active control points are highlighted; (right) the efficient algorithm considers only a narrow strips around the active control points.	67
4.10	The images $\log(H)$ of the regularization matrix H for a cubic control lattice of size $N = 15$; (left) when all the control points are considered (3375×3375); (right) when the proposed efficient algorithm is used (889×889)	68
4.11	The efficient 3L-IBS method to reconstruct the <i>Bunny</i> data set containing only 817 points; (a) the offset of <i>Bunny</i> ($\delta = 1\%$); (b)-(d) different settings for the regularization parameter $\lambda = 1, 10^2, 10^3$ respectively.	70
4.12	Complexities of the 3L-IBS and its efficient version for <i>Bunny</i> data set containing $10k$ points: (left) the size of the optimization problems; (right) the CPU-time spent for finding the optimal control lattice.	70
4.13	(a) Results from the 3L algorithm. (b) Results from the proposed approach. (c) Results from [2] used as ground truths.	72
4.14	(left) Non-uniform synthetic noisy data from an ellipsoid (127 points), fitted with the 3L algorithm ($\delta = 10\%$, AFE=4.3376), the proposed approach (AFE=3.1234) and [2] (AFE=3.2441). (right) 3D data points obtained with a structured light camera from a sphere; outer mesh correspond to the result obtained with the 3L algorithm (AFE=242.6154) while inner spheres show the results from the proposed approach (AFE=85.1591) and the ground truth (AFE=85.4815).	73
4.15	(a) Best result from the 3L algorithm obtained by setting $\delta = 4$ percent of the object size (AFE=0.1285). Results after incorporating: (b) two positional constraints (AFE=0.1179); (c) three positional constraints (AFE=0.1068); and (d) four positional constraints (AFE=0.1105).	74
4.16	Non-uniform sampling and open boundary cases: (left) Results from the 3L algorithm; (middle) Results from the proposed approach; (right) Results from an orthogonal distance based approach [2], non-linear fitting.	75
4.17	(a) The result obtained by the Gradient-One algorithm [133]. (b) An enlargement showing how the fitting algorithm ignores the positional constraint, the gradient vectors have a similar length and orientation though.	76
4.18	(1st col) original 2D data sets. Results obtained by: (2nd col) 3L-IP [2]; (3rd col) Gradient-one algorithm; (4th col) the proposed 3L-IBS.	77

4.19	<i>(left)</i> The result obtained by the Gradient-one algorithm [133]. <i>(right)</i> An enlargement showing how the fitting algorithm ignores the positional constraint, the gradient vectors have a similar length and orientation though. . . .	78
4.20	<i>(1st col)</i> original 3D data sets. Results obtained by: <i>(2nd col)</i> 3L-IP [2]; <i>(3rd col)</i> Gradient-one algorithm; <i>(4th col)</i> the proposed 3L-IBS.	79
4.21	<i>(left)</i> Original high resolution 3D data sets. <i>(right)</i> Results obtained by the proposed 3L-IBS.	82
4.22	<i>(left)</i> The efficient 3L-IBS method for fitting a dense set of points. <i>(right)</i> Enlargement of a region of the surface.	83
4.23	<i>(left)</i> Triangular meshes of a face. <i>(right)</i> Optimal implicit surface obtained through our efficient 3L-IBS fitting method.	84
5.1	Using an interface for point sets registration: <i>(a)</i> Initial position of data (+) and model (o) sets; <i>(b)</i> Data set (+) and model set represented by an IBS; <i>(c)</i> Registration result of data set (+) and the IBS; <i>(d)</i> The same result but represented by using the model set (o) and transformed data set (+).	88
5.2	The level curves of the IBS is used to approximate the distance. In addition, the distance sensitivity with respect to small changes in rotation and translation can be approximated as well.	90
5.3	<i>(left)</i> Sensitivity of the distance with respect to small changes in rotation. <i>(right)</i> Sensitivity of the distance with respect to the translation along y axis.	91
5.4	Convergence region of the proposed approach with respect to $[\theta, t_x, t_y]$ for the 2D object and IBS in Fig. 5.10(<i>top</i>).	92
5.5	Initial positions of data sets and model sets for noisy (<i>top</i>) and partial overlap (<i>bottom</i>) examples registered with the different approaches.	93
5.6	Model sets and data sets containing different density of points. <i>(left)</i> Initial configurations. <i>(right)</i> Final results from the proposed approach.	94
5.7	Real data sets (from [52] and [51]) registered with the proposed approach and state of the art techniques. <i>(left)</i> Initial set up of the given data and model sets represented by means of triangular meshes to highlight details. <i>(middle)</i> IPs representing model sets and data points. <i>(right)</i> Results of the proposed registration approach represented through triangular meshes to make easier a visual evaluation.	100
5.8	Partial overlap cases. <i>(left)</i> Initial set up of data sets and model sets to be registered. <i>(middle)</i> IPs representing model sets and data points from the data sets. <i>(right)</i> Results from the proposed approach.	101
5.9	Evolution of ARE of different registration algorithms along 30 iterations.	101
5.10	<i>(a)</i> Initial positions of data (+) and model (o) sets. <i>(b)</i> Data (+) sets and IBSs representing the models. <i>(c)</i> Final results of registered data (+) sets and IBSs. <i>(d)</i> The same result but represented by using the model (o) sets and transformed data (+) sets with the proposed approach.	102
5.11	3D cases corresponding to real data sets registered with the proposed approach and state of the art techniques: <i>(left)</i> Initial positions of data (+) sets and model (o) sets. <i>(middle)</i> Data (+) sets and model sets represented by IBSs; <i>(right)</i> Final results of registered data (+) and model (o) sets with the proposed approach.	103

6.1	The proposed method (SD-FFD): (a) initial configuration of the data set (blue) and the model set (red); (b) result with $\lambda = 10^5$ (at iteration 15); (c) result with $\lambda = 1$ (at iteration 22); (d) the optimal FFD control lattice results in a very dense correspondence, not only on the boundary but over the whole space.	106
6.2	(a) Illustration of a point-to-point distance based approach (e.g., ICP [20]). (b) Local quadratic approximation [99] used in current work.	108
6.3	Registration results of 2D shapes ([52], [127]) using the proposed approach (SD-FFD) (a) initial configurations of data sets (blue) and model sets (red); (b) results with $\lambda = 10^5$; (c) results with $\lambda = 10^2$; (d) results with $\lambda = 1$; (e) the optimal FFD control lattice results in a very dense correspondence, not only on the boundary but over the whole space.	113
6.4	Registration results of 3D shapes using the proposed approach (SD-FFD). (a) initial configurations of data sets (green) and model sets (red); (b) results with $\lambda = 10^4$; (c) results with $\lambda = 10^2$; (d) Final registration results showing the blending of the two shapes (data & model).	115

Chapter 1

Introduction

Shape representation and registration are counted as two main problems in computer vision. The first one focuses on different ways to describe a given cloud of points through some parameters, while the second one is mainly about transforming two cloud of points in order to move them closer to each other. The cloud (set) of points can be picked up from a 2D boundary or 3D surface of an object. This process can be done by 3D scanners, stereo cameras or time of flight cameras. Meanwhile in this process, some registration technique might be required in order to put all clouds of points, obtained from different views, into a same cloud of points. Moreover, depending on the acquisition method, the obtained set of points can be disturbed by a percentage of noise or outliers. Then the main problem is how to describe this cloud of points through the optimal set of parameters. This latter is studied in shape representation.

These two problems, although somehow connected, have been separately studied in different communities like Computer Vision (CV), Computer Graphics (CG) and Computer-Aided Design (CAD). In this thesis we firstly study both problems deeply and then try to connect them together. The main concept that is common in both problems is the notion of "distance". In other words, for both in shape representation and registration problems a distance must be formulated to measure the closeness. In shape representation this closeness is considered between the given cloud of points and the curve/surface to be reconstructed, while in shape registration it is formulated to measure the closeness of the two given cloud of points. This fact has been exploited during this thesis as a clue to find a unification between these two problems.

1.1 Motivation

Object representation, in computer graphics and CAD, concerns with modeling and describing object through a set of parameters. Different varieties of tools, including NURBS and Bezier surfaces have been provided by these communities to design and deform the model of an object. In computer vision, on the other hand, object representation models are applied

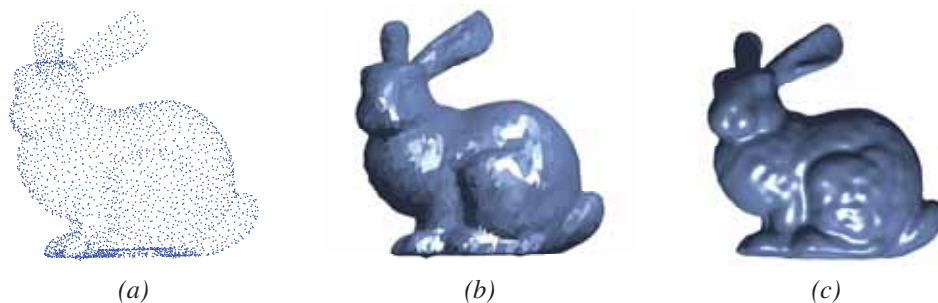


Figure 1.1: Different representations: (a) 3D cloud of points; (b) triangulated mesh; (c) implicit functions.

to describe and analyze a given shape/image. This image can be a 3D cloud of point, for instance. Then object representation techniques might be used to describe this cloud of points by a set of parameters. This problem is also referred to as surface fitting (surface reconstruction) and it might be applied to reduce the effect of noise and outlier. The final output will be a clean and compact surface that describe the object as a whole.

Triangular meshes can be mentioned as widely-used tools in computer graphics that provide quite fast representation tools. They are very simple and can be easily rendered just by knowing the light and camera positions. Moreover, different textures can be easily projected in each triangle, and then over the whole object. Therefore, the final rendering of a triangular mesh may look quite realistic. Triangular meshes, though simple, require hundreds of thousands triangles for a high resolution model. This problem makes it costly specially from designing point of view. Variety of alternative representations have been proposed in CAD and CG to design and model an object with fewer parameters. Bezier surfaces and NURBS, for instance, provide flexible tool to model 3D surfaces through some control points. In designing applications, this set of control points can be defined by the user through some graphic interface. In computer vision, for surface reconstruction, the optimal control point can be found through some optimization process.

Figure 1.1 illustrates three different ways of representing a 3D object. The left figure corresponds to the point representation, containing 2899 3D points. Point representation is the most simple visualization tool in computer graphics and vision. Rendering a 3D cloud of points only requires the camera position and orientation plus the 3D coordinates of the points to figure out their projection on a 2D plane. Although a point is an abstract entity, a circular marker can be used instead to visualize it. Unfortunately the 2D projection (image) of the 3D cloud of point is the most ambiguous representation, since no lighting effect is provided by this tool.

Triangular mesh provides a richer alternative to represent a given 3D cloud of points. In addition to the point coordinate this representation requires the connectivity between the points. This information can be encoded in a table of triangles (or polygon in general) whose rows contain the indices of triangles. This mesh of triangles provides the local information about the surface that can be used to approximate the surface orientation (referred to as normal vector). Moreover, each single triangle is a geometric entity occupying a physical space

unlike a point in space. Hence, each triangle can be rendered independently. Assembling all the triangles together results in a visualization of the whole 3D object. Figure 1.1(*middle*) visualizes the 3D Bunny containing 2899 points and 5786 triangles. This representation does not provide any smooth (differentiable) geometry although it can be visualized as a smooth object by using some lighting tricks.

Implicit functions, on the other hand, provide a very strong tool to represent objects in 2D, 3D and even 4D (object morphing during the time). They can describe different objects with complex topology through their zero set (the points where the implicit function reaches zero). They can apply different functional space to maintain any continuity (e.g., belonging to C^∞ by using implicit polynomials). Implicit B-Splines (IBSs) and Radial Basis Functions (RBFs) are among the common representation tools to describe objects implicitly. This representation can be very compact like low degree implicit polynomial, or very flexible like a high resolution implicit B-spline.

Figure 1.1(*right*) illustrates a middle-resolution IBS (a lattice of $20 \times 20 \times 20$ of control value) to represent the cloud of 2899 points shown in the left. This representation is obtained by assembling cubic spline patches that result in a C^2 continuity. In addition, this IBS provides a more compact representation compared to the triangular mesh. Moreover, it should be mentioned that this IBS has 20^3 control parameters and only about 1600 of them are involved to represent the zero set. While, on the other hand, the triangular mesh in Fig. 1.1(*middle*) requires more than 26000 parameters for triangular mesh (2899×3 values as point coordinates plus 5786×3 indices to recover the connectivity). Therefore, the IBS provides a smooth representation with about 94% save in memory space in this typical example.

In addition to smoothness and compactness, implicit functions can easily represent the object in different levels through controlling the *regularization* parameter. This parameter control the rigidity of the implicit function over the space where object lies in. Higher regularization parameter imposes more rigidity force, and results in a coarser representation (see Fig. 1.2(*a*)). Lower regularization parameter, on the other hand, relaxes this force and results in a more varying zero set (see Fig. 1.2(*f*)). Changing this parameter from high to low produces in-between shapes as illustrated in Fig. 1.2 (*b*) – (*e*).

In the first part of this work implicit functions and distance metrics on these functions are studied to represent 2D boundaries and 3D objects. Implicit polynomial and implicit B-spline are the main two formats we apply to represent the object through the zero level set. On the other hand distance approximations in geometric and algebraic frameworks have been explored and some novel contributions have been proposed to the community.

In the second part of the thesis, the registration problem is tackled using the representation tools provided in the first part. Point set registration is another main problem in Computer Vision. It aims at finding the best set of transformation parameters to align two given sets of points. The first given set is referred to as model set (target) and the second one as data set (source). During the registration the data set moves toward the model set in order to minimize the distance between these two sets. Depending on the application the transformation must be restricted to some extends. For instance, in 3D scanners the transformation to assemble two different views of an object must be restricted to Euclidean (rigid-body) motion, while in other applications like medical imaging this transformation must be extended in order to capture the deformation of organs.

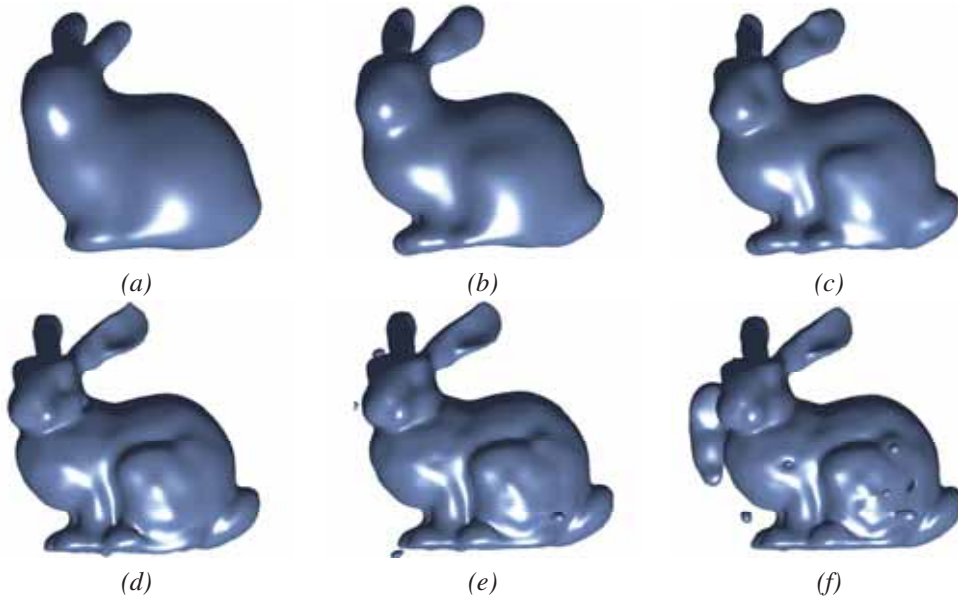


Figure 1.2: Different resolutions in implicit representation (the regularization parameter decreases from left-to-right and top-to-bottom).

Figure 1.3(a) illustrates data and model set as two cloud of points in 3D of the same object. The data set and model set contain N_d points and N_m points respectively. The first notion in the registration problem that needs to be defined first is the *distance* between the two sets of points. This distance provides a reliable criterion to both handle registration and measure the quality of the registration. In a very simple case, one can use the point-to-point distance as a registration criterion. Then, a complexity of $O(N_d N_m)$ must be spent to consider different combinations to find the closest points. Moreover, the obtained distance might not be reliable enough, specially when the point sets have different densities/distributions.

The notion of distance in registration has been considered in different frameworks like probabilistic models and distance fields. In this thesis this concept is studied using the implicit representations provided in the first part. Using implicit curves/surfaces to represent the model set, the distance is converted from a point-to-point type to a point-to-model one. Figure 1.3(b) illustrates the first step, when the model set is replaced with an implicit B-spline surface. This implicit surface, as studied in the first part of the thesis, facilitates the distance measurement.

We will study how adapting a fast algebraic criterion in the registration framework can improve the distance measurement. The provided distance is robust to noise due to the fitting stage, and behaves smoothly while we move a point in a local neighborhood. Moreover, since its gradient information can be analytically extracted, it can be minimized through any gradient-based optimization algorithm. The result of this point-to-model minimization stage provides an accurate configuration for the previous point-to-point registration problem. Figure 1.3(c) depicts the result of optimization stage that minimizes the distance between the

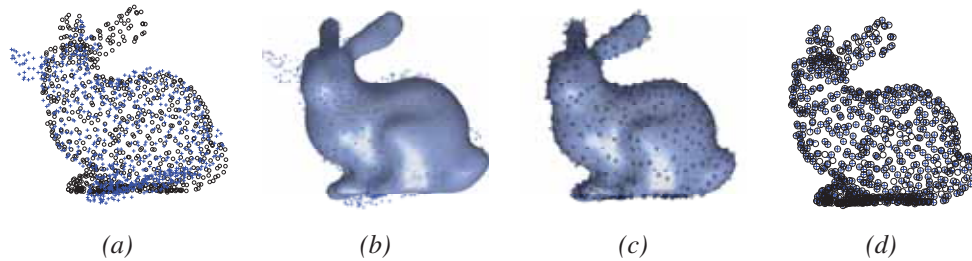


Figure 1.3: Using an interface for point sets registration: (a) Initial position of data (+) and model (o) sets; (b) Data set (+) and model set represented by an IBS; (c) Registration result of data set (+) and the IBS; (d) The same result but represented by using the model set (o) and transformed data set (+).

data set and the implicit surface. For the sake of visualization the original model set, already replaced with the implicit surface, is provided. Figure 1.3(d) highlights the fact that using an implicit surface as an interface results in the optimal registration of the given clouds of points.

In addition to the rigid registration, the non-rigid case has been casted as well. Non-rigid deformation includes those transformations that cannot be captured by rigid rotation and translation. Figure 1.4(a) shows two clouds of points, where the data cloud is a non-rigid deformation of the model cloud. For a better visualization these two set are represented through their triangular meshes. Figure 1.4(c) illustrates the result after rigid registration. It can be understood from this figure that the data set is still far from the model set. Iterating more rigid registration cannot vanish the distance between these two sets. Therefore, the solution space containing the transformation parameter should be extended to capture more deformation.

Deformation models, as studied in Chapter 2, include different types of transformations that can be classified as: intrinsic (e.g. Laplacian deformation) and extrinsic (e.g. Thin Plate Splines). The first category include those deformations only applied on the manifold (curve/surface), while the second one include more general deformations applied on the whole region (plane/space). In our work on non-rigid registration we apply Free Form Deformation (FFD) to model non-rigid transformations. Figures 1.4(d) – (f) illustrate how the distance between the two model and data sets decreases during the registration. Hence, allowing more transformations to be captured may result in a significant decrease in the registration error.

1.2 Objective

In this thesis we tackle both representation and registration problems as well as their relationship. Our intention is to find different alternatives to represent a given cloud of points in order to obtain a higher level of object representation. This strategy results in a representation method that is robust to noise and outliers. In the second parts of the thesis we investigate

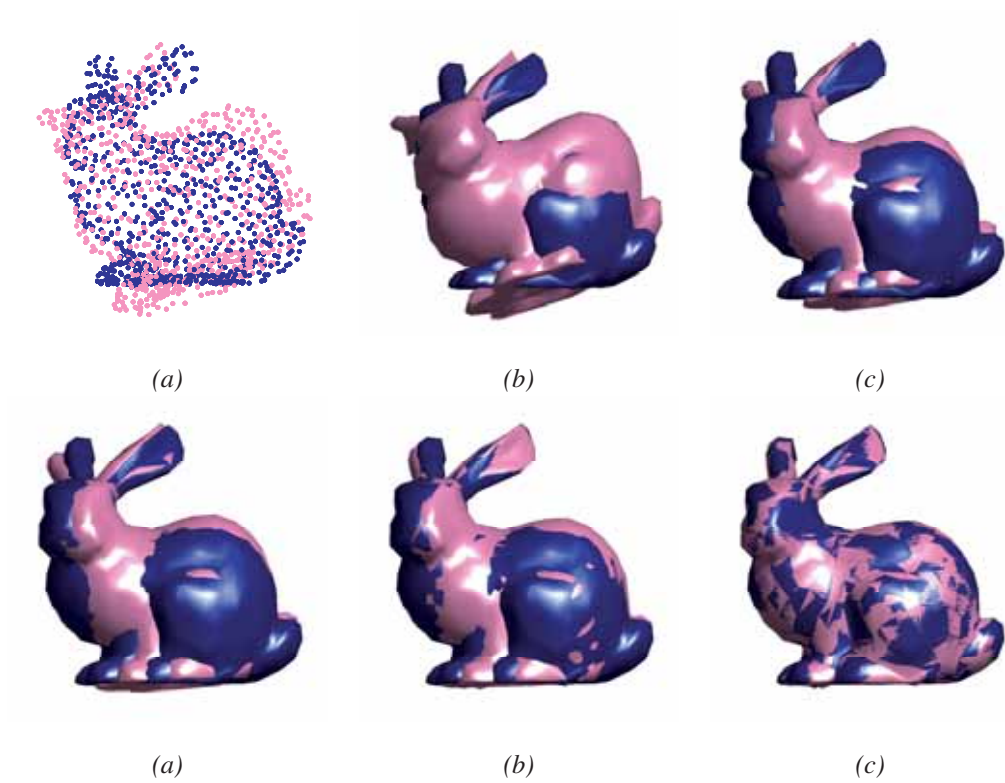


Figure 1.4: Nonrigid registration of two cloud of points; the data set (pink) moves toward the model set (dark blue) while the rigidity must be controlled during the registration; triangular mesh is used to highlight the difference.

how such a representation equipped with a distance estimation can benefit the registration problems. Our main objective in both part of the thesis can be summarized as follows:

- **Reviewing reconstruction techniques:** we study different reconstruction methods in both computer graphics and computer vision communities. It includes different representations from parametric to implicit, as well as different fitting techniques from algebraic to geometric ones.
- **Classifying registration methods:** a comprehensive review of the literature is targeted in order to study and classify different registration techniques. We study both rigid and non-rigid registration methods using different criteria for error measurement. These techniques vary from coarse to fine registration.
- **Flexible representation:** we aim at finding flexible representations that do not require any parametrization. For this purpose we explore implicit representations formed by polynomials and B-splines. Curve/surface reconstruction in this domain can be equipped with a fast algebraic fitting methodology.
- **Distance approximation:** working on the implicit functions can be facilitated by using some distance metric. We study different techniques for obtaining accurate distance approximation. Moreover, we study how these approximations can be accelerated while the accuracy is preserved.
- **Rigid registration:** in addition to reviewing state-of-the-art on registration methods, we try to find a robust method that does not rely on point level computation. For this purpose, we exploit implicit representations equipped with a fast distance estimation. Merging these two results in a method in a higher level.
- **Non-rigid registration:** as the last part of the thesis non-rigid registration is explored in order to expand the application of distance approximation on non-rigid registration. We use a quadratic distance approximation that incorporate curvature information. Applying this distance can accelerate the convergence of the registration method.

1.3 Thesis Outline

Chapter 2 of this thesis reviews state-of-the-art techniques on both shape representation and registration. In the first part different techniques on object representation are classified based on the solution space and the fitting methodology. Solution space contains the set of all possible parameters describing the object. It can be simple and contains the set of polynomial coefficients, or can be more elaborated like radial basis functions. Fitting methodology is the way the distance is defined. The distance provides a criterion to define the closeness of the given set of points to a curve/surface to be constructed. Then an optimization algorithm can be followed to find the best set of parameters in the solution space. The distance concept by itself contains two categories referred to as geometric and algebraic distance. The first one is quite accurate and measure the orthogonal distance between the point and surface, while the latter is just a rough estimation of the real distance but quite fast instead.

The second part of Chapter 2 reviews and classifies different registration techniques. Our classification includes different deformation models and registration distances. Based on the deformation model used during the registration, we classify state-of-the-art into affine group, extrinsic and intrinsic deformations. The affine group is the simplest type of deformation that includes rigid body motion, plus shearing and scaling. Non-rigid registration methods can use either extrinsic or intrinsic deformation. In the earlier the whole region/space where the object lies in can be deformed, while the latter deforms only the boundary/surface of the object. In addition to deformation model, the registration distance provides another criterion to classify state-of-the-art. We study different techniques including both coarse and fine registration techniques. Coarse registration methods use either some general statistics of the point set distribution (e.g., PCA) or use some point descriptors to match two sets of points (e.g., shape context). Fine registration methods runs more accurate registration after some initialization is provided. More detail on these techniques are explained and classified in Chapter 2.

Chapter 3 and 4 construct the first part of the thesis, which deals with shape representation. The main objective is to describe a given set of points through an implicit function. In our implementation two types of implicit functions are used: implicit polynomial as a simple representation, and implicit B-spline as a complex one. The parameter vector in the first case contains the coefficients of polynomial, and in the second one contains the control value of implicit B-spline. Then, different techniques including both geometric and algebraic approaches are hired to find the best parameters in the solution space. It should be mentioned that all the fitting techniques proposed in these chapters are applicable for both implicit B-spline and polynomial. So, our main focus is the fitting criterion used for finding the optimal parameters.

In Chapter 3 curve and surface fitting using a geometric distance is tackled. This distance is an approximation of the real distance of point to implicit function. In this approximation a simplex between the point and implicit function is constructed first. This simplex is a triangle in 2D or a tetrahedron in 3D whose edges continues along the axes from the point to the implicit function. Then, the height of this simplex is chosen as an approximation of the real distance. This approximation is exploited for both distance estimation and distance minimization. As the first set of experiments, this distance estimation is used in a RANSAC-based fitting method. In this framework, first a random set of points is chosen to interpolate the implicit polynomial; then, the distance of the whole cloud of points to this function is estimated using the proposed method. In the second set of experiments, this distance estimation is used in a more elaborate optimization method (e.g., LMA) that exploits the gradient information of the distance estimation.

Chapter 4 focuses on algebraic approaches for curve/surface fitting. Algebraic fitting methods are less accurate but quite fast in comparison with the geometric ones. The algebraic method we use, referred to as 3L algorithm, is based on the three levels comprising the data set and its inside and outside offsets. Then, the optimal parameters can be easily found as a linear least squares solution. In the first part of this chapter, we propose a technique to relax the constraints in the original 3L algorithm. During the relaxation, the orientation compatibility between the implicit function and the cloud of points is monitored. In the second part of this chapter, implicit B-splines are explored and used for algebraic fitting. A simple and fast

framework to reconstruct a flexible curve/surface is provided in the linear least squares form. We show how this flexibility can be controlled to explore different levels of detail from coarse to fine.

Chapter 5 and 6 construct the second part of the thesis, and focus on registration problem. The main objective is to transform two given clouds of points to obtain a unified set. Based on the transformation type, the method can be rigid or non-rigid registration that are studied in Chapter 5 and 6 respectively. The initial idea of these chapters is to focus on the concept of distance in registration problem. Unlike the two pervious chapters that deal with the distance between points and implicit functions, Chapter 5 and 6 consider the distance between two sets of points. One of the objective we follow in the second part of this thesis is to relate these two concepts of distance, and then using this relationship in the registration.

In Chapter 5 a novel approach to rigid registration is introduced. This method exploits implicit representations that are already presented in Chapter 3 and 4. In this algorithm one of the given cloud of point (model) is described first by an implicit function. Then, the second cloud of points (data) moves toward the zero set of this implicit function using a distance metric. This distance is an algebraic estimation of the real distance between points and the implicit functions; hence, it will be fast to compute the distance between the data and implicit function instead of the distance between two cloud of points. In other words, an implicit function is used as an interface to facilitate the distance estimation. All the formulation is presented in a non-linear least squares form. Moreover, in our experiments both implicit B-spline and polynomial are used as interface to represent the model set. This is to show the independence of the method to the specific solution space used for representation.

Chapter 6 considers non-rigid registration of two clouds of points. The non-rigid deformation is described by Free-Form Deformation (FFD) that is counted as an extrinsic deformation. The registration distance we use is a quadratic approximation of the real distance that incorporates curvature information in addition to orientation. This distance has been perviously proposed for rigid registration, but we extend it for capturing both rigid and non-rigid deformation by just using a single framework. Since the local curvature information is incorporated in the distance formulation, the convergence is faster compared to state-of-the-art. In Chapter 6 we present more detail about the formulation that is in the least squares form. Finally in Chapter 7 a conclusion of this thesis and our contribution in different stages is presented.

Chapter 2

State of the art

In this chapter the related works are reviewed comprising two main tracks: representation and registration. The first one includes the related work in both algebraic and geometric fitting methods as well as different representation spaces, while the latter concerns of rigid and non-rigid registration methods. Both categories are well studied and properly classified in order to provide a better understanding of related work, and facilitate the referencing in following chapters.

2.1 Representation

Object representation is an interesting and challenging problem in Computer Vision (CV), Computer Graphics (CG) and Computer Aided Design (CAD). In computer graphics different representations are used to provide a smooth description of the object to be rendered (e.g., MPU proposed by [95]). Representation model has been widely used in computer vision in several applications like object modeling [158], 3D object reconstruction [151], large-scale scene reconstruction [18], 3D object recognition using implicit functions [9], pose estimation and object identification [140] to mention a few.

In this section the most relevant work in representation models are reviewed and classified based on the solution space and the fitting methodology. The solution space determines the domain where the shape parameters are defined. Moving on this space will change the shape of zero set, and the objective of fitting is to find the best parameters. The goodness of fitting is defined through the distance function that is explained in the second subsection.

2.1.1 Solution Space

The main objective of fitting methods, in general, is to describe a given set of N points $\Gamma = \{\mathbf{p}_i\}_1^N$ through a parameter vector. The parameter vector consists of values that control the shape of the object. The point set can be extracted from an object's surface [77], or can be taken from the whole volume [76]. This cloud of points can be visualized through a triangular mesh. It contains a table of vertex coordinates and a table of triangle indices. The triangles can be easily rendered, and consequently the whole object can be visualized with lighting and other details. Despite the simplicity, triangular meshes need too much memory space, so it does not provide a compact representation.

Parametric representations are widely used in computer graphics as compact alternatives of triangular meshes. Bezier, B-splines and NURBS are different parametric models to visualize curves or surfaces. All these representations are smooth and linear with respect to their parameters. The main problem of these methods is the parametrization; in other words, for every 3D point in the given cloud of points, a parameter in 2D must be assigned, then the best parameter will be found in order to have the minimum distance.

Implicit representations, unlike the parametric ones, avoid the parametrization problem, and this property makes them useful for computer vision applications. Its capability to describe complicated boundaries/manifolds through its coefficient vector, and the nonexistence of parametrization, has been exploited in fields such as: range image reconstruction [146], pose estimation [87], point sets registration [97, 54, 141, 106], shape description [90], position invariant object recognition [94], 3D image segmentation [157] and registration [155], to mention a few.

Through this thesis implicit functions have been used for representation, since they do not require any parametrization. Hence, this review of state-of-the-art is mainly focused on this kind of representations. It should be noticed that implicit representations provide more information compared to parametric ones, since they describe other part of region/space where the object lies in. Its zero level set, specifically, describes the object; its other level sets can describe the inner/outer part of the object. So it can be claimed that implicit functions provide more general representation. In [152], for instance, implicitization of parametric curves has been studied.

Implicit function describes the object through the *zero set* that is defined as the set of points where the function reaches zero: $Z(f_c) = \{\mathbf{x} : f_c(\mathbf{x}) = 0\}$. The fitting procedure seeks for the best parameter c defining the zero set that obtains the minimum distance between the given points and its zero set. In this subsection we review some of the solution spaces where the parameters are chosen from.

It should be mentioned that implicit representations and triangular meshes are somehow connected. For instance, in order to visualize an implicit function, generally, a triangular mesh through the marching cube algorithm is extracted in a grid of points. This grid can be computed from the implicit function in any resolution. The marching cube algorithm detects the patches of the zero set surfaces by considering eight neighbors in each cell; then, it determines whether and how any polygon needs to be reconstructed.

An implicit polynomial (IP) is one of the simplest example to define the solution space.

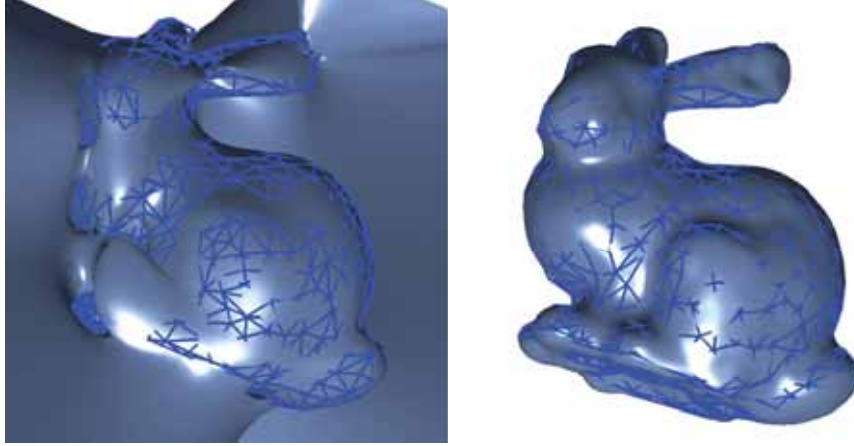


Figure 2.1: Fitting the *Bunny* data set containing 817 points; (*left*) result by using an implicit polynomial [2]; (*right*) result by using IBS (in both cases the object triangular mesh is also depicted to highlight the accuracy of the results).

IPs are generated through the linear combination of the monomials in the format of $x^i y^j z^k$. An implicit polynomial of degree n is represented as:

$$f_{\mathbf{c}}(\mathbf{x}) = \sum_{\substack{(i+j+k) \leq n \\ \{i,j,k\} \geq 0}} c_{i,j,k} \cdot x^i \cdot y^j \cdot z^k \quad (2.1)$$

where $\{c_{i,j,k}\}$ defines the IP coefficients. Following a common order for both coefficient set $\{c_{i,j,k}\}$ and monomial set $\{x^i y^j z^k\}$ let us define each set as a vector. We can consider both sets in the vector form, such that both are ordered in an increasing way of the degree $(i + j + k)$. This common order in both vectors helps to define the implicit function as an inner product of two column vectors:

$$f_{\mathbf{c}}(\mathbf{x}) = \mathbf{c}^T \mathbf{m}(\mathbf{x}) = \mathbf{m}(\mathbf{x})^T \mathbf{c} \quad (2.2)$$

where $\mathbf{m}(\mathbf{x})$ is the vector of monomials and \mathbf{c} is the polynomial coefficient vector. The monomial \mathbf{m} is a vector function depending on the point coordinates while the coefficient vector \mathbf{c} is a constant vector defining the shape through the zero set. Hence, the objective of fitting is to find the best coefficient vector as a representative of the whole object.

Implicit polynomials offer a smooth and compact representation of object just through a simple compact parameter vector [9]. Implicit polynomial compactness has been also an attractive point to be exploited when a high level reasoning is needed (e.g., object recognition [9], object modeling [158], [151], etc). They are defined linearly with respect to the parameter vector, and this property makes them popular for linear fitting. On the other hand, each term



Figure 2.2: Basis functions construct the solution space: (*left*) RBF ; (*right*) IBF.

of the monomial vector doesn't have a compact support¹. Consequently, any change in any parameter can lead to a global change in the whole shape of the zero set. Figure 2.1(*left*) shows how fitting with IP results in outliers around the zero set. This problem is reduced in [110] for instance, through regularizing the IP coefficients. In order to address this problem completely, the solution space should be defined through another representation different than IP (e.g., [91], [68]).

A Radial Basis Function (RBF) is a class of functions defined through some special basis functions called radial. The values of these radial functions only depend on the distance of the given point to a fixed point \mathbf{x}_i referred to as control point [91]. Hence the values do not change in a fixed *radius* of the control point:

$$f(\mathbf{x}) = P(\mathbf{x}) + \sum_{i=1}^n c_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) \quad (2.3)$$

where P is a low degree IP, ϕ is the radial function, $\{\mathbf{x}_i\}$ is the set of control points, and \mathbf{c} is the coefficient vector. A special radial basis function called Thin Plate Spline (TPS) is defined as $\phi(r) = r^2 \log r$. A linear combination of this function and its translations explore the whole solution space. Figure 2.2(*left*) shows a basis function used to cast the whole space. The authors in [28] use RBF to approximate the surface fitting the 3D range data. They present a regularization method for RBF that enable it to describe the object in different resolution/order of smoothness. Despite being elegant and flexible, RBFs do not have the basis functions with a compact support. Consequently, they suffer the global control problem as IPs, although they are more flexible.

The authors in [17] use implicit RBFs to reconstruct a surface from the given cloud of points. They manage to reduce the number of radial basis functions by using a greedy criterion. The optimal RBF coefficients are found through solving a linear system of equations. In this system the regularization parameter is also contributed to control the rigidity of the final implicit surface. Applying RBFs for representation has been extended in [142] by using a global regularization and in [114] by using centers among the vertices of the Voroni diagram.

Fourier descriptor is another useful representation model that has been used for surface

¹support of an implicit function is where it doesn't vanish.

reconstruction. In [149] the authors use a Fourier representation to formulate the object surface through two-variables Fourier descriptor. The optimal surface is reconstructed directly through the projection of a striped lighting system. Super-quadratic is used in [135] as a lower degree surface to fit a surface to the 3D range data.

Implicit B-Splines (IBSs) offer a smooth and flexible representation, which can be locally controlled. Its basis functions have compact supports (see Fig. 2.2 (*right*)); hence, each parameter has a local contribution to the whole shape; in other words changing one parameter only changes a local part of the zero set. Figure 2.1(*right*) depicts how an IBS presents a more accurate and flexible representation than an IP. Indeed, IBS avoids outliers around the object thanks to the local control property. Moreover, it is linearly defined with respect to its parameter vector. This property makes it appropriate for linear fitting [68] as well as the geometric distance based fitting [3], [153]. In addition, some regularization constraints can be easily imposed in order to control the global shape of the object.

An implicit B-spline in 2D is defined as a combination of the tensor product basis functions:

$$f(\mathbf{x}) = f(x, y) = \sum_{i=1}^M \sum_{j=1}^N c_{i,j} B_i(x) B_j(y) \quad (2.4)$$

where $B_i(x)$ and $B_j(y)$ are the spline basis functions, and the matrix $[c_{i,j}]_{M \times N}$ is the control lattice that controls the shape of the IBS. For the sake of simplicity we consider $M = N$, then the basis functions $B_i(x)$ and $B_j(y)$ have the same behavior but defined in different domains. Similarly, the basis function in 3D is a tensor product of three spline basis functions. Since each B_i has a compact support in 1D, the final tensor product will have a compact support in a higher dimension.

In summary, IBSs are quite flexible representations to describe 2D boundaries and 3D objects. They provide smooth implicit functions that can be locally controlled. IBSs have been already used in computer graphics and CAD communities. For instance, [68] uses IBSs for fitting a given set of points associated with normal vectors. The fitting process tries to find an appropriate IBS whose gradient vectors have similar orientations to the given normal vectors [133]. Moreover, it is shown in [31] how the implicit fitting result can be modified by a parametric fitting in a dual evolution framework. A conversion of parametric curves to implicit form is studied in [152] as well.

2.1.2 Fitting Methodology

Having confirmed the solution space, the fitting procedure seeks for the best solution based on a fitting criterion. This criterion is called a distance function and it measures the closeness of the given point set to the obtained zero set. In general the fitting methodology can be divided into two main categories referred to as: *geometric* and *algebraic* approaches. Algebraic approaches are among the simple methods for approximating the distance between a point and curve/surface, while in geometric ones a reliable approximation for the real distance (orthogonal distance) between the point and curve/surface is required. In this case the distance between a point and the surface is usually defined as the shortest distance between this point and its correspondence on the surface.

Geometric Approaches

In general, geometric fitting methods can be formulated as an optimization problem where the best curve/surface parameter must be found:

$$Dist(\Gamma, f_c) = \sum_{i=1}^N \min_{\hat{\mathbf{p}}_i} d^2(\mathbf{p}_i, \hat{\mathbf{p}}_i) \quad (2.5)$$

where $\Gamma_0 = \{\mathbf{p}_i\}_1^N$ is the given set of points and each $\hat{\mathbf{p}}_i$ is the corresponding foot-point to \mathbf{p}_i on the zero set defined as $Z(f_c)$. Here we consider the l_2 norm to calculate the distance d , and consequently a non linear least squares optimization must be solved.

Theoretically, both unknown surface parameters and the correspondences must be found simultaneously, but practically this problem is tackled by first assuming an initial surface, and then refining it till convergence is reached. So, the fitting problem is split up into two stages: *i*) point correspondence search; and *ii*) surface parameter refinement. The first stage deals with the minimization of the individual summands in (2.5) with respect to the foot-point, while the second one concerns the minimization of the whole summation with respect to the surface parameters.

Regarding the first stage two different strategies have been proposed in the literature: (a) finding the shortest distance by solving a non-linear system (e.g., [2], [3]); and (b) computing an estimation of the shortest distance (e.g., [134], [19], [42]). Once the foot-points are approximately located, the second stage tries to update the parameter vector. This minimization is usually an iterative process that can be solved by gradient-based algorithm like gradient descent or Levenberg-Marquadt algorithm. In Chapter 3 more detail on this stage will be provided, but now we only consider different techniques for estimating the foot-points.

In [2] Ahn et al. propose a method to find the correspondence on the surface, which is based on its geometric properties. This foot-point, $\hat{\mathbf{p}}$, is somewhere on the surface satisfying $f_c(\hat{\mathbf{p}}) = 0$. Furthermore, the line connecting the data point with the foot-point must be parallel to the ∇f_c at the foot-point, where ∇ is the gradient operator. In other words, we must have $\nabla f_c \times (\hat{\mathbf{p}} - \mathbf{p}) = 0$. Merging these two conditions, the following system of equations must be solved:

$$\begin{pmatrix} f_c \\ \nabla f_c \times (\hat{\mathbf{p}} - \mathbf{p}) \end{pmatrix} = \mathbf{0}. \quad (2.6)$$

This equation could be solved by the Newton–Raphson algorithm for non-linear system of equations. Figure 2.3 illustrates how the foot-point is iteratively approximated by this geometric fitting method. Although this method is precise enough, and even covers some well-known method in the literature like [134] and [115], it is quite time-consuming due to the iterations.

In [3] the orthogonal fitting is extended for general error functions, such as l_1 and l_∞ norm of the residual error instead of the common l_2 norm. This highlights the importance of the error function selection for the fitting process. The authors present the fitting algorithm as an evolutionary process of a surface along its normal direction. They discuss and compare their approach with other common error functions including the algebraic types.

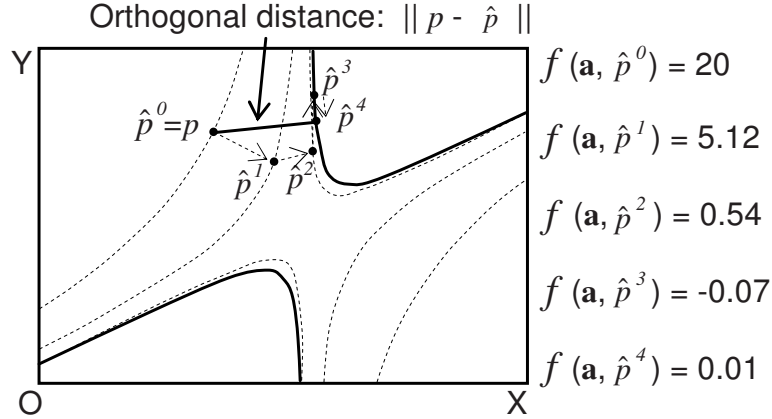


Figure 2.3: Orthogonal distance from a given point p computed by means of the iterative approach [2].

Instead of computing the shortest distance through (2.6), [42] proposes to approximate it, avoiding iterative approaches as a result. In that work, which is an extension of [19] for more general surfaces, first a normal vector \vec{n}_p for each point p is computed by using principal components analysis (PCA) in a small $M \times M$ neighborhood centered at each point [49]. In other words, $\vec{n}_p = (n_1, n_2, n_3)$ is defined as the eigenvector of the local covariance matrix Σ associated with the smallest eigenvalue:

$$\Sigma = \frac{1}{s} \sum_{i=1}^s (\mathbf{p}_i - \bar{\mathbf{p}})(\mathbf{p}_i - \bar{\mathbf{p}})^T \quad (2.7)$$

where $\bar{\mathbf{p}} = \frac{1}{s} \sum_{i=1}^s \mathbf{p}_i$ is the vector showing the mean position of the neighboring points in the $M \times M$ region. Finally, \hat{p} is computed as the intersection of the surface $f_c(\mathbf{x}) = 0$ with a line passing through p and parallel to \vec{n}_p :

$$\frac{x - x_p}{n_1} = \frac{y - y_p}{n_2} = \frac{z - z_p}{n_3}. \quad (2.8)$$

The intersection is used as an approximation for the foot-point \hat{p} in (2.5).

In [134], Taubin proposes an approximation for (2.5), which is based on the first order Taylor expansion of the distance function. The distance could be computed through normalizing the algebraic distance by the gradient norm:

$$Dist(\Gamma, f_c) = \sum_{i=1}^N \left(\frac{|f_c(p_i)|}{\|\nabla f_c(p_i)\|} \right)^2. \quad (2.9)$$

This approximated distance is used in an iterative weighted least squares method as well as in a nonlinear optimization framework. In addition, a new constraint is imposed on the coefficient vector, which is based on the data points as well as on the coefficients. The

approximated distance proposed by Taubin [134] may not reach the correspondence point lying on the zero set, which could affect the final fitting result. In fact, instead of considering the zero set, the level set where the point is lying on is affected by this optimization process. Finally, every point forces its level set to move in order to reach a lower accumulated distance, and the best set of parameters is found.

Algebraic Approaches

Unlike the geometric fitting methods that seek for a reliable approximation to the orthogonal distance, algebraic approaches are based on some simple and computationally efficient criteria to measure the closeness of the given point set to the zero set [100], [30]. As a simple criterion, the deviation of $f(\mathbf{p}_i)$ from zero can measure the distance between the point \mathbf{p}_i and the zero set $Z(f)$. This criterion can be formulated in the least squares form:

$$Dist(\Gamma, f_{\mathbf{c}}) = \sum_{i=1}^N f^2(\mathbf{p}_i) \quad (2.10)$$

this accumulated residual error is a function of the parameter vector \mathbf{c} in a linear least squares form. This fact will be more clear when this error function is represented in the vector form. Remember that each value $f(\mathbf{p}_i)$ can be represented as the inner product of the parameter vector and the monomial calculated in the given point: $f(\mathbf{p}_i) = \mathbf{m}(\mathbf{p}_i)^T \mathbf{c}$. Putting each $\mathbf{m}(\mathbf{p}_i)^T$ as a row in the *monomial matrix* \mathbf{M}_{Γ} leads to another representation:

$$Dist(\Gamma, f_{\mathbf{c}}) = \|\mathbf{M}_{\Gamma} \mathbf{c}\|^2 = \mathbf{c}^T \mathbf{M}_{\Gamma}^T \mathbf{M}_{\Gamma} \mathbf{c} \quad (2.11)$$

which is a quadratic function of the parameter vector.

The least squares solution can be computed quite fast, but it is usually unstable. This problem will disturb the shape of zero set as a consequence. There has been many works to tackle this problem through proposing a better fitting criterion. For instance, as mentioned in the previous subsection, [134] uses a first order distance approximation to formulate a least square fitting function. Indeed, it considers the normalized form $f/\|\nabla f\|$ instead of f in (2.10). This method is formulation in (2.9) as the first iteration of the geometric method to find the foot-point. Figure 2.4 depicts the approximation accuracy of this metric in a simple 2D example.

The 3L algorithm is another fitting techniques that uses geometric clues in the fitting function in order to obtain a more stable result [11]. In this method the original data is supported by two additional offset, the inner and the outer ones. Then, the fitting procedure considers these additional sets to find the optimal parameter. In more detail, it constructs two additional offsets $\Gamma_{+\delta} = \{\mathbf{p}_i^+\}_1^N$ inside and $\Gamma_{-\delta} = \{\mathbf{p}_i^-\}_1^N$ outside of the given boundary Γ_0 . Then it can be formulated as a least squares form:

$$Dist(\Gamma, f_{\mathbf{c}}) = \sum_{i=1}^N f_{\mathbf{c}}(\mathbf{p}_i)^2 + \sum_{i=1}^N (f_{\mathbf{c}}(\mathbf{p}_i^+) - \epsilon)^2 + \sum_{i=1}^N (f_{\mathbf{c}}(\mathbf{p}_i^-) + \epsilon)^2 \quad (2.12)$$

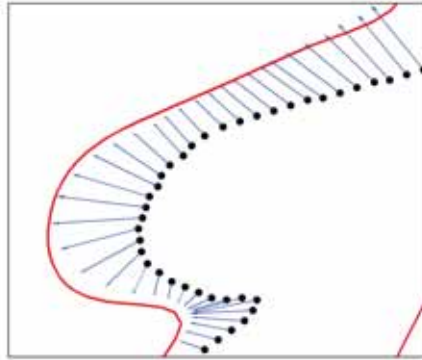


Figure 2.4: An algebraic criterion [134] is used to approximate the distance between a set of points and the zero set.

where $\pm\epsilon$ are the expected values of f_c in the additional offsets. The 3L algorithm has been originally proposed for IP fitting, and it will be detailed in Chapter 4.

In similar work, [132] and [133] consider the associated normal vectors in the least squares formulations. The optimal function is found such that its gradient in the data set is parallel to the normal vectors. Then, the zero set is hopefully more similar to the data set. This method is referred to the Gradient-one in the literature, since it considers the normals with the unit length. The 3L algorithm and the Gradient-one have been extended in [110] and [45], both of them tackling the stability problem. In [110] a ridge regression method is proposed to handle the instability problem related to the monomial matrix. In [45] the sensitivity function of the zero set with respect to the parameter vector is analyzed, and two techniques are proposed: Min-Max minimizes an upper bound of this function, while Min-Var tries to minimize the variance of it.

The idea of using associated normal vectors in fitting algorithms is also used for implicit B-spline fitting. Juttler in [67] provides a least squares framework to find the optimal algebraic spline curves with the help of normal vectors. The information of the associated normal vectors helps the positional information of the given data set. Considering these constraints as well as a global tension term results in a quadratic objective function, which is easily minimized by setting its derivative to zero. This idea is later generalized in [68] to reconstruct algebraic surfaces as well.

Moving Least Squares (MLS) offers another class of techniques for shape representation [5], [53]. MLS is based on a Weighted Least Squares (WLS) defined in a local neighborhood of a fixed point. Solving this WLS, this point moves and another WLS will be considered. The authors in [5] use a MLS formulation for finding a local map describing the object in a neighborhood. In fact, it is used to project the points on the considered surface. The authors in [43] extend this technique to increase the stability and the flexibility to describe corners and sharp boundaries.

Most of the abovementioned fitting techniques use least squares, or Levenberg-Marquadt Algorithm (LMA) to find the optimal parameters. Trust region algorithm is another optimization technique that can be used for surface reconstruction. Yang et al. [153] formulate a geometric distance approximation to find the best implicit B-spline. They use trust region to find the global optimal coefficient vector. In a different work Aigner et al. in [3] use a geometric distance optimized by Gauss-Newton optimization techniques.

Distance field is another method to represent the given cloud of points in a different level [21], [92]. This technique is used both in fitting and registration problems. In a couple of continuous works, the authors in [97], [145] present a second order approximation of the distance fields. This approximation is based on the curvature information of the data set. It is used for parametric B-spline curves and surfaces fitting in the squared distance minimization (SDM) scheme. Pottmann et al. [98] use SDM for parametric active B-spline curves and surfaces. Moreover, SDM is used in [21] to fit subdivision surfaces. The control points of a coarse mesh are adjusted through SDM in a fast converging method.

Multiple partition of unity (MPU) is a popular technique in computer graphics for rendering a cloud of points [95], [88]. Generally speaking, MPU describes the object based on a set of small patches attached together in a smooth way. In more detail, the whole set of points is recursively subdivided to construct an octree structure. Each cell of the octree is described by a low degree IP (e.g., quadratic) to capture the local structure of the point set in the cell. Finally all these local approximations are integrated into a global approximation through a partition of unity approach, which guarantees the continuity and smoothness. In [88] the authors use the gradient one algorithm and ridge regression techniques to obtain a better result for a local description in MPU.

Labatut et al. in [74] formulate the reconstruction problem in an energy minimization framework. The optimal surface is explicitly described by a triangular mesh. Their proposed energy function is based on the inside/outside labeling of Delaunay tetrahedra. In addition to the surface quality, the visibility constraints are also considered in the formulation. In addition to all the aforementioned methods, there have been other reconstruction techniques including variational methods [77], finite-element method (FEM) [124], Bayesian surface reconstruction [54], and Poisson formulations [69] that are not presented here.

2.2 Registration

Point set registration is an important problem in computer vision being tackled during last three decades [86], [80]. Two sets of point referred to as *data set* (source) noted by $\mathcal{P} = \{\mathbf{p}_i\}_1^{N_d}$ and *model set* (target) noted by $\mathcal{Q} = \{\mathbf{q}_j\}_1^{N_m}$ with N_d and N_m points, respectively, are given. Then, the best transformation must be found to move the first set close to the second one. The transformation can vary from rigid to non-rigid deformations to tackle 2D image or 3D cloud of point registration.

Among the many applications of this topic we can mention: 3D scanning using multi-view laser scans [27] or time-of-flight camera [121]; in-hand object modelling [147]; considering non-rigid deformation in 3D scanning [16]; 3D urban scene reconstruction [159];

2D-3D fusion in 3D urban scene reconstruction [80]; large scale 3D modeling [46]; motion capture using depth camera [82]; pose estimation from 2D images [48]; motion capture from a set of 2D videos [40]; reassembling fractured objects from 3D scanned pieces [56].

In this section we review the most related work in image/cloud of points registration. The state of art is classified in three main categories: *i*) the deformation model that defines the solution space (e.g., Thin Plate Splines (TPS) [24]; Free Form Deformations (FFD) [58]; Laplacian deformation [128]); *ii*) the distance used to measure the registration error (e.g., point-to-point [10], point-to-quadratics [97], distance field [58]); *iii*) the optimization techniques used to minimize the registration error (e.g., branch and bound [96], dynamic programming [122] and LMA [34]).

2.2.1 Deformation Model

Affine Group

One of the simple transformation is rigid -transformation or Euclidian transformation, which is functioning from Euclidian space to itself. It includes rotation and translation parameters, and comprises only three parameters (one for rotation and two for translation) in 2D space and six parameters (three for rotation and three for translation) in 3D case. This transformation is formulated as follows:

$$\mathbf{T}(\mathbf{p}) = \mathbf{R}\mathbf{p} + \mathbf{t} \quad (2.13)$$

where \mathbf{R} is the rotation matrix, and \mathbf{t} is the translation vector. The rotation matrix must be an orthonormal matrix (i.e., it is orthogonal: $\mathbf{R}^T\mathbf{R} = \mathbf{I}$, and its columns have unit length). Otherwise the transformation is referred to as affine transformation, which is not distance-preserving anymore.

The rotation matrix can be parametrized based on the amount of rotations. In 2D case the rotation can be represented based on a single parameter θ referred to as rotation angle:

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}. \quad (2.14)$$

The rotation matrix in 3D can be decomposed into three rotation matrices around x , y and z axes (also referred to as yaw, pitch and roll). Each of these matrices are the extended form of (2.14) in 3D. The rotation around the y axis, for instance, is formulated as follows:

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}. \quad (2.15)$$

In addition to rotation matrices, kinematics and quaternion provide another ways of representing the rigid transformation [10].

In general, rigid transformations are Euclidean isometric. In other words, every rigid transformation T preserves the distance between any pair of points:

$$d(\mathbf{T}(\mathbf{p}), \mathbf{T}(\mathbf{q})) = d(\mathbf{p}, \mathbf{q}) \quad (2.16)$$

where d is the Euclidean distance (defined by L^2 norm). It must be noticed that if the metric d is defined on a manifold (not in \mathbb{R}^2 space) then the transformation is called a geodesic isometry.

Extrinsic Deformation

If the transformation is not a rigid body motion (i.e., Euclidean isometry) it is called a non-rigid deformation. These deformation models can be divided into two main categories: the *intrinsic* and the *extrinsic* models. The intrinsic deformations refer to those only applied on the curve or surface manifolds (e.g., Laplacian deformation), while the extrinsic ones consider the whole domain/volume the object lies in (e.g., Thin Plate Splines). The first one is appropriate for 2D boundary or 3D manifold matching, while the latter one is mainly suggested for image/volume matching. In this subsection we review and classify the most representative deformation models, the intrinsic ones are presented next.

In a simple case, a non-rigid deformation can be modeled as a set of locally rigid transformations. In [6] the authors consider a 3×4 affine matrix per data point. Therefore, they end up with a $4N_d \times 3$ matrix containing the deformation parameters, where N_d is the number of data points. The consistency of these local affine transformations is controlled by a stiffness term.

Thin Plate Spline (TPS) is a well-known model for non-rigid deformations [12]. It provides a continuous mapping from \mathbb{R}^k to \mathbb{R}^k (where $k = 2$ for images and $k = 3$ for volume deformations). This class of mappings is defined through radial basis functions introduced in section 2.1.1:

$$\mathbf{T}(\mathbf{x}) = \mathbf{d} \cdot \mathbf{x} + \sum_{i=1}^n \mathbf{c}_i \phi(\|\mathbf{x} - \mathbf{w}_i\|) \quad (2.17)$$

where \mathbf{d} represents the affine parameters, $\{\mathbf{w}_i\}$ is the set of control points, and $\{\mathbf{c}_i\}$ defines the corresponding displacement vectors. The smoothness of \mathbf{T} is controlled by the second order tension term (known as thin plate spline):

$$\mathcal{L}(\mathbf{T}) = \iint \left\| \frac{\partial^2 \mathbf{T}}{\partial x^2} \right\|^2 + 2 \left\| \frac{\partial^2 \mathbf{T}}{\partial x \partial y} \right\|^2 + \left\| \frac{\partial^2 \mathbf{T}}{\partial y^2} \right\|^2 dx dy. \quad (2.18)$$

This term can be considered in the final energy function, and it can measure the stiffness of TPS mapping.

Free Form Deformation (FFD) provides another region/volume based deformation [123]. Similar to TPS it is applied on a region in \mathbb{R}^k and map it into \mathbb{R}^k ($k = 2$ or 3 depending on the case). FFD differs from TPS due to the basis functions it uses. In TPS radial basis functions defined by $\phi(r) = r^2 \log r$ are applied. These functions do not have a compact support. It means that any change in one coefficient parameter \mathbf{c}_i may leads to overall changes on the whole mapping. This problem is sorted out in FFD since it uses B-spline basis functions. FFD in 2D is defined as follows:

$$\mathbf{T}(\mathbf{x}) = \mathbf{T}(x, y) = \sum_{i=1}^M \sum_{j=1}^N \mathbf{c}_{i,j} B_i(x) B_j(y) \quad (2.19)$$

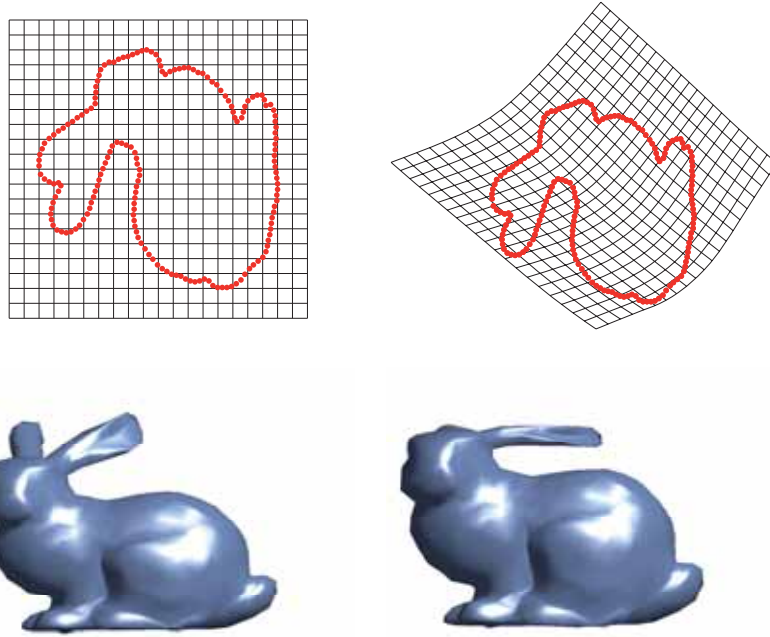


Figure 2.5: FFD can deform the whole region/space through moving its control lattice.

where $B_i(x)$ and $B_j(y)$ are the B-spline basis functions as defined in section 2.1.1, and $\mathbf{c}_{i,j}$ is a 2D grid of control vector that control the amount of deformation in each local neighborhood. Without loss of generality the FFD domain can be restricted to the unit square $[0, 1]^2$ in 2D case (see Fig. 2.5(top)) or unit cube $[0, 1]^3$ in 3D case (see Fig. 2.5(bottom)).

FFD benefits from the B-spline properties; since each B-spline basis has a compact support, the FFD control vectors only affects a local neighborhood of the domain. Moreover, FFD will be an identical mapping when the control grid is regularly distributed on a rectangle covering the domain (refer to Chapter 6). So the deviation from the initial regular grid \mathbf{c}^0 , defined as $\delta\mathbf{c} = \mathbf{c} - \mathbf{c}^0$ can be used to describe the deviation from the original shape. The resulting FFD defined by the incremental values $\delta\mathbf{c}$ is called Incremental FFD (IFFD) [58]. The smoothness term to measure the rigidity of IFFD, denoted by $\delta\mathbf{F}$, is defined as:

$$\mathcal{L}(\mathbf{T}) = \iint \left\| \frac{\partial\delta\mathbf{T}}{\partial x} \right\|^2 + \left\| \frac{\partial\delta\mathbf{T}}{\partial y} \right\|^2 dx dy. \quad (2.20)$$

FFD and IFFD have been successfully applied for non-rigid image/volume registration. Moreover there are many extensions of FFD like Extended FFD that consider different structure for control lattice [26] as well as hierarchical B-spline [150] and multi-level FFD [120] that propose a coarse-to-fine grid refinement.

Igarashi et al. [60] propose another alternative to avoid setting skeleton or FFD domain

before deformation. Their method is still an extrinsic model applied on the object triangular mesh. The criterion used in their method to be preserved during the deformation is the proportion between the triangles' edges in the original mesh. They provide a least squares form to find the optimal vertex position gaining a similar proportion while some vertex are imposed by the user. Moving least squares is another framework used in [118] for image deformation. Their technique finds a local affine transformation for every point that devotes more weights to closer points in the least squares problem. Each optimal affine transformation corresponding to the given point can be found in a closed form.

Triangular mesh representation, though simple, provides a good framework for modelling both non-rigid deformation and registration. In [32] a coarse triangular polygon is used to represent 2D objects. For 2D polygons without holes, these polygons are connected in a tree structure. This special structure facilitate the object detection through using dynamic programming. Statistical shape prior provides another useful tool for registration and object recovery [4]. Salzmann et al. use triangular meshes to extract the statistical information of the 3D objects [113], [112]. This information is related to the angles between neighboring faces in the given meshes while the edge length are preserved. This information is dimensionally reduced by PCA. Afterwards the mean shape and the major variations obtained by PCA are used to recover the 3D shape from 2D images. This method can be counted as extrinsic one since the deformation is applied on the whole 2D domain. Among other extrinsic deformation models we can refer to applying of mean value coordinate [66], harmonic coordinate [64], and green coordinate [84].

Intrinsic Deformation

The *intrinsic* deformation models, on the other hand, consider only the curve/surface to be deformed, not the whole domain it lies on. Laplacian mesh deformation, for instance, provides a deformation technique to deform the object boundary/manifold. It process the δ coordinate or differential coordinates of every point, defined as difference between the point and the center of mass of its neighbors (see Fig. 2.6). It is proved that the whole object can be retrieved through its δ coordinate if only one point is given. Sorkine [128] uses these information in a least squares form to deform the original shape by moving some of its boundary points. An over-determined system of equations is proposed containing the constrains defined by δ coordinates and the constraints imposed by the user.

Variational implicit functions provide another useful tool for shape transformation. Turk and O'Berin [139] consider the shape (either curve or surface) evolution as an interpolation problem in a higher dimension. More specifically they consider two given k -dimensional shapes in two parallel $(k + 1)$ -dimensional planes $t = 0$ for the first object and $t = 1$ for the second one (see Fig. 2.7). Fitting the final $N + 1$ dimensional given points together with the smoothness term, an implicit function will be found that hands over all the in between shapes (for different value of parameter $t \in [0, 1]$).

Deformation as curve/surface evolution has been well-studied in both computer graphics and computer vision communities [144]. Differential geometry provides a theoretical framework to consider the shape evolution as moving from one point to another in the shape space [71]. Kilian et al. [70] use Riemannian metric in the shape space to solve the shape evolution

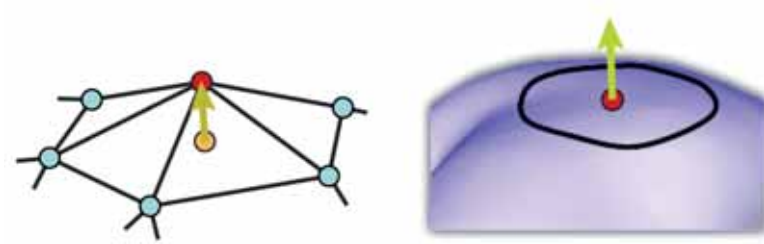


Figure 2.6: δ -coordinates are used to encode local geometric information (illustration from [128]).

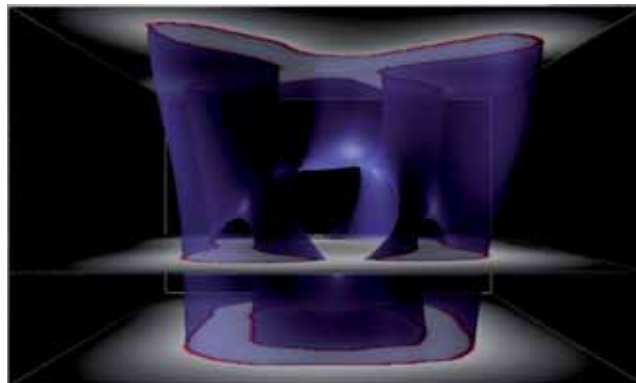


Figure 2.7: Shape transformation using variational implicit functions (illustration from [139]).

as a point set interpolation in this space. They propose two different metrics to provide the geodesic path (locally shortest path in the given metric sense) between two surfaces in the shape space.

2.2.2 Registration Distance

Registration techniques can be divided into two main categories based on the initial configuration of two given sets. The coarse registration, on the one hand, refers to those cases where the data set and model set are away, and no initial configuration is provided. The fine registration, on the other hand, refers to the finer step to minimize the registration distance.

Coarse Registration

Principle Component Analysis (PCA) is one of the simplest coarse registration method. It is a statistical method suggested when no correspondence is provided. It transform the data set to align its origin and major axes with those of model set. Having constructed the covariance matrices of data set and model set, their corresponding major axes can be found by eigenvectors of this matrix. PCA aligning methods find the best rotation based on the obtained set of orthonormal axes [111].

PCA aligning technique, though quite simple, is unreliable especially for circular objects and in presence of noise or outliers. Moreover, for partial overlap or occluded objects the main axes is a wrong clue to align the data with model. In order to improve PCA many methods have been proposed in the literature. Singular Value Decomposition (SVD) is another statistical method for coarse registration. In this method every data point is paired with its correspondence in the model set, which could be its closest point in the simplest case. The best parameters can be found by Singular Value Decomposition (SVD). This method constructs the *cross-covariance* matrix between the data and model sets, and then finds the best rotation matrix through the decomposition outputs [7].

Finding the corresponding point for the given data point has been a challenging problem in computer vision and pattern recognition [111]. In order to find a correspondence some feature descriptor should be used meanwhile. Belongie et al. [8], for instance, propose a robust descriptor referred to as *shape context*. It describes the local distribution of the given point by considering log-polar histogram bins around it. Then, this information is later used for bipartite graph matching that ends up in the best correspondence matching. The matching result is later used to find the optimal rigid and non-rigid parameters (using TPS).

Coarse registration methods in 3D exploit different point descriptors to solve the matching problem. Spin image is one the well known 3D descriptor used both for object recognition and surface registration [63]. Spin image provides a 2D histogram in the given point on a 3D surface (see Fig. 2.8). It considers the tangent plane passing through this point, and then projects the neighboring point on this plane to compute their distance to the tangent plane β as well as their distance to the normal vector α . Then a table is generated to accumulate the number of points in the neighborhood having different values of α and β as the table coordinates.

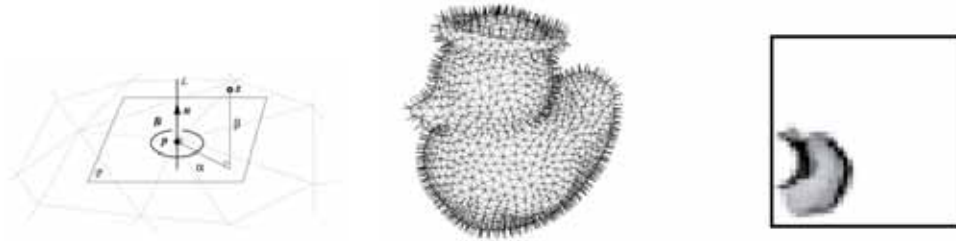


Figure 2.8: Spin images projects local neighboring points on the tangent plane (illustration from [63]).

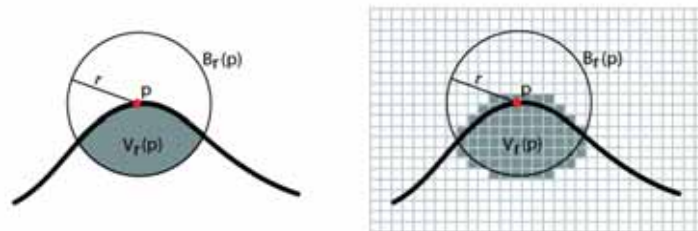


Figure 2.9: Volume integral descriptor can be used to find the feature points (illustration from [41]).

Volume integral descriptor proposed by Gelfand et al. [41] is another 3D point descriptor based on the local geometry on the surface. It considers different balls with different radius centered at the given point, and then compute the integral volume of the inside surface by discretizing the domain (see Fig. 2.9). The obtained value is normalized by dividing it by the whole volume. Considering this simple descriptor at different scales, the authors in [41] propose a method to find the feature points with unique properties. Among other 3D descriptors we can refer to the tensor representation [89], isometry invariant descriptor [148] and surface moment invariant [137].

In addition to point descriptor, which is more appropriate to find the feature points, there must be some technique to measure the similarity between the data set and its matching in the model set. Some local techniques like comparing the descriptor histograms are widely used for matching. But for a global similarity measurement, we must consider the consistency of matching in a geometric sense. *Coordinate root mean squared error* (cRMS) is a proper way to measure the similarity [41]. In this metric the Euclidean distances between the transformed data points and their correspondence is calculated as a measure; but it requires the best transformation parameter to transform the data set. A better alternative could be the *distance root mean squared error* (dRMS) as introduced in [41]. This metric is equivalent to cRMS but it does not require any optimal transformation parameters. This metric compares all the internal distances between any pairs in the data set with the distances of their correspondences in the model set. Huber et al. [59] propose a geometric approach to rigid registration by checking

the surface consistency. Further coarse registration techniques can be found in [109], [55], [83] and [136].

Fine Registration

For the fine registration a coarse initialization is provided that put the data set close to the model set. Hence, fine registration techniques seek for the best deformation parameters that minimize the *distance* between the given sets. In order to define or estimate the registration distance different tools can be exploited, e.g., point-to-point distance, distance fields, implicit functions and probabilistic models. These classes of techniques are briefly described as follows.

Point-to-point distance: the Iterative Closest Point (ICP) algorithm is one of the classical registration approaches that iterates a two-step scheme till convergence [10], [20]. In the first step it considers every data point and finds its correspondence as the closest point in the model set. Then, in the second step, ICP tries to find the best rotation and translation parameters (deformation parameter in non-rigid case [6]) to move the data points close to their correspondences. Given the data set $\mathcal{P} = \{\mathbf{p}_i\}_1^{N_d}$ and the model set $\mathcal{Q} = \{\mathbf{q}_j\}_1^{N_m}$, the registration distance in ICP can be formulated as follows:

$$\varphi(\Theta) = \sum_{i=1}^{N_d} \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_{j(i)}\|^2 \quad (2.21)$$

where Θ is the vector of rigid parameters (including rotation and translation), and $j(i)$ is the model point index corresponding to data point i . The best parameters can be found by either SVD or a quaternion-base algorithm as proposed by the authors in [10]. This two-step process (finding the correspondences first and the optimal deformation parameters after) is *iterated* till convergence is reached.

ICP algorithm is based on a point-to-point distance, which makes it inaccurate especially when the data and model sets have different densities, and the data points and their correspondence do not coincide. Chen and Medioni [20] propose a point-to-plane distance instead. This registration distance measures the distance from a point to the tangent line/plane in the correspondence:

$$\varphi(\Theta) = \sum_{i=1}^{N_d} [(\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_{j(i)}) \cdot \mathbf{n}_{j(i)}]^2 \quad (2.22)$$

where \mathbf{n}_k is the normal vector at the model point \mathbf{q}_k . This distance gives a better estimation of the real distance between the given sets, and using that in ICP algorithm speeds up the optimization convergence. In addition to distance criteria in ICP algorithm, different variants of this algorithm consider the way to assign the correspondence to every data point [108]. Some extensions of ICP like Trimmed ICP [23] either improves the optimization step or like non-rigid ICP [6] extends the deformation space.

Robust Point Matching (RPM) proposed by [24] considers the ICP corresponding assignment as a soft-assignment problem. In other words, the authors consider all the model points



Figure 2.10: Distance field of a 2D bunny point set and its derivatives along two coordinate axes.

as potential correspondences instead of devoting only one:

$$\varphi(\Theta) = \sum_{i=1}^{N_d} \sum_{j=1}^{N_m} m_{ij} \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_j\|^2 \quad (2.23)$$

where m_{ij} is the fuzzy coefficient showing the importance of model point j as a correspondence for data point i such that $m_{ij} \in [0, 1]$ and $\sum_{j=1}^{N_m} m_{ij} = 1$ for all $i = 1 \dots N_d$. Note that in a normal ICP algorithm m_{ij} is assigned either 0 or 1. Using this formulation the authors in [24] propose an EM-like algorithm comprising two steps: first, updating the correspondences m_{ij} , and then optimizing the transformation parameters. For the second step they simplify the formula by considering a new estimation as the correspondence, which is a linear combination of other model points. Then, their formula is similar to (2.21) with $\mathbf{q}_i = \sum_{j=1}^{N_m} m_{ij} \mathbf{q}_j$ as a virtual correspondence for \mathbf{p}_i .

Distance Field: this representation has been widely used to approximate a robust registration error in both rigid and non-rigid domains [85]. The approach as presented in [34] does not require any point-to-point correspondence, and overcomes the non-differentiable nature of ICP by using a derivable distance transform—Chamfer distance. The error function derived from that distance field is a smooth function, and its derivatives can be analytically computed; hence, it can be minimized through LMA to find the optimal registration parameters. Using the distance transform speeds up the ICP like algorithms:

$$\varphi(\Theta) = \sum_{i=1}^{N_d} w_i D(\mathbf{R}\mathbf{p}_i + \mathbf{t})^2 \quad (2.24)$$

where w_i is the weight of data point i and $D(\mathbf{x}) = \min_j \epsilon(\|\mathbf{x} - \mathbf{q}_j\|)$ is the distance transform that can be computed before running the algorithm. Once the distance field and its gradients are computed (see Fig. 2.10), the registration algorithm does not require any correspondence computation. The main disadvantage of [34] is the precision dependency on the grid resolution, where the Chamfer distance transform and discrete derivatives are evaluated. Hence, this technique cannot be directly applied when the point set is sparse or unorganized.

Huang et al. [57], and [58] measure the registration error by comparing the distance fields

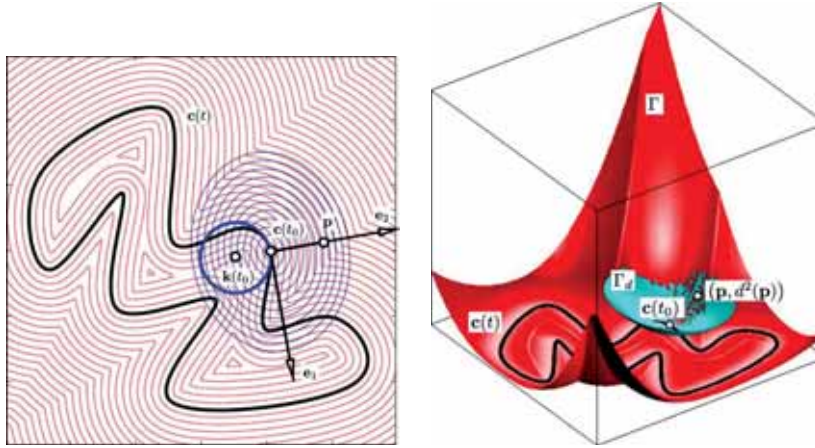


Figure 2.11: (*left*) A local quadratic approximation of the distance field using the curvature information. (*right*) A 3D presepective of this local approximation (illustration from [99]).

induced by data set and model set. They measure the similarity between two distance fields by accumulating the difference between the distance values over the whole region. This accumulation is encoded in the form of double integral that makes the method computationally expensive. They used FFD to capture the non-rigid deformation and the gradient descent algorithm to find its best parameters. This method is later extended by Taron et al. [131] to consider the uncertainty of the point sets encoded in the covariance matrix. Vector Distance Field (VDF) has been also used as an extension of distance field to solve the registration problem [29], [22].

The distance field used in [34] is a discrete field and its derivatives used in LMA are not precise enough. Holzer et al. [48] use distance transform for 3D pose estimation. In [99] the authors present a local quadratic approximation of the distance function based on the curvature information (see Fig. 2.11). These local approximations define the distance field of the model points, and reformulate the registration problem as an optimization problem that can be solved by Newton's method. This distance approximation has been also used in L^1 norm and minimized through linear programming [37]. Since the approximated distance needs curvature information of the point set, it is computationally expensive and sensitive to noise.

Implicit function: it provides another framework to represent a set of points in 2D/3D. As mentioned in section 2.1.2 there are different fitting methods to find an optimal implicit function to describe a set of points. The outcome of fitting procedure is a smooth interpolating function that is robust to noise. This representation can be exploited to deal with the registration problem beyond the point level. Claes et al. [25] tackled the 3D-3D registration problem as a 3D-model registration. Their method is based on a Variational Implicit Surface (VIS) to describe the model set. After finding the optimal RBF interpolating the model point, they develop an algebraic distance to measure the registration error. This distance is a non-linear function of six rigid parameters, which is minimized through a conjugate gradient method.

This idea is later on extended to tackle the multi-view fine registration. RBF has been also used in [146] to tackle both the registration problem and the surface reconstruction.

Following a similar approach, [156] proposes a fast registration method based on solving an energy minimization problem derived from an implicit polynomial fitted to the given model set [158]. This IP is used to define a gradient flow that drives the data set to the model set without using point-wise correspondences. The energy functional is minimized by means of a heuristic two step process. Firstly, every point in the given data set moves freely along the gradient vectors derived from the IP. Secondly, the outcome of the first step is used to define a single transformation that represents this movement in a rigid way. These two steps are alternately repeated until convergence is reached. The weak point of this approach is the first step that lets the points move independently in the proposed gradient flow. Furthermore, the proposed gradient flow is not precise, specially close to the boundaries. Implicit polynomials have been also used in [130] to represent both the data set and model set, and then register two IPs based on their moment information. Similarly, Lee and Lai [75] use MPUs to describe both data and model sets in registration problems, and then reformulate the problem as a registration of two MPUs.

Probabilistic Models: Kernel Correlation (KC) proposed by Tsin and Kanade [138] formulates the registration problem in a different way. The authors define the registration error as a kernel correlation between two sets:

$$\varphi(\Theta) = - \sum_{i=1}^{N_d} \sum_{j=1}^{N_m} KC(\mathbf{R}\mathbf{p}_i + \mathbf{t}, \mathbf{q}_j) \quad (2.25)$$

where KC is the kernel correlation between two points. When KC is chosen as a Gaussian kernel correlation it can be easily computed as:

$$KC(\mathbf{p}_i, \mathbf{q}_j) = (2\pi\sigma^2)^{-D/2} \exp(-\|\mathbf{p}_i - \mathbf{q}_j\|^2/2\sigma^2) \quad (2.26)$$

which is a function of point entropy as well. Then, the author of [138] show how maximizing the kernel correlation results in an optimal registration parameters.

Jian and Vemuri [61] present a probabilistic model for point set registration. They describe both data set and model set by *Gaussian Mixture Models* (GMMs), and the problem is treated as an alignment problem between two density functions. This modeling provides a more robust method for non-rigid registration through TPS [62]. This method has been pursued in [14] to tackle the 3D rigid registration. Although the formulation they provide avoids the correspondence search like the ICP kind algorithm ([10], [20]), and the gradient information can be explicitly derived and exploited in an optimization stage, it considers all the combinations between the data points and the model points, which is quite expensive.

Probabilistic approaches based on mixture models are highly dependent on the number of mixtures used for modelling the sets. This problem is generally solved by assuming a user defined number of mixtures or as many as the number of points. The former scheme needs the points to be clustered, while the latter one results in a very expensive optimization problem that cannot handle large data sets or could get trapped in local minimum when complex sets are considered. Generally speaking, although these methods do not require any correspondence search, all points in the model set are implicitly considered as a potential correspondence for each single point in the given data set.

Instead of formulating the problem as aligning two mixture models, the authors in [13] presents a continuously differentiable energy function based on Gaussian Fields. They propose an intuitive physical interpretation of the registration problem in terms of Gaussian force fields that are made by the model set to attract the given data set. The proposed differentiable cost function directly relates the point coordinates with the registration parameters. Thus, the registration parameters are obtained by means of gradient based optimization techniques. Similarly to the probabilistic approaches presented above, the main problem of [13] lies on the optimization step, which can result in a non-linear complex function. Even though no correspondence search is required for these methods, all the model points are implicitly considered as a potential correspondence for each single point in the given data set.

Ho et al. in a recent work [47] present an algebraic method to find the optimal affine registration parameters. First they consider two point set distributions, and then they find the best affine matching of two moments up to the third degree. Other probabilistic techniques like maximum likelihood [14], [126], Expectation Maximization (EM) algorithm [50], Expectation Conditional Maximization [50], Coherent Point Drift (CPD) [93] and other statistical modelling of shapes [131] have been explored in the literature.

Distance in Shape Space: Differential geometry provides a theoretical framework for shape analysis and shape similarity measurement [65]. The authors in [73] propose a distance function between two surfaces that is invariant to rigid motions and re-parametrization. They use a special representation for the surface referred to as q-map. The distance invariance to re-parametrization is obtained by considering different basis elements for the tangent space to generate the most common surface parametrization. Then the distance is defined as the minimum value over the whole set of possible parametrization. Curve/surface distance definition in the shape space is further studied in [38] and [15].

2.2.3 Optimization Method

In this section the optimization frameworks used to solve the aforementioned registration techniques are briefly described. Moreover, these techniques are classified based on coarse or fine registrations. The first category usually ends up with a combinatorial technique (e.g. branch and bound), while the latter uses a continuous optimization framework (e.g., gradient based).

Coarse Registration

Branch and bound is a well known algorithm in discrete optimization; it has been widely used for registration problems, specially for coarse rigid registrations. Gelfand et al. [41] after proposing a descriptor to find the feature points apply a branch and bound algorithm to find the correspondences in two sets of 3D points. Their method starts with a greedy algorithm to initialize the lower bound to restrict the search. Then, the branch and bound algorithm is applied to considered those combinations with a matching error less than the bounds. Those branches (combinations) with matching error exceeding the bounds will be pruned, otherwise they will be recursively branched to consider more possible combinations

for matching. Finding a global optima in coarse registration problems has been studied in [96] and [78].

Dynamic Programming is another discrete optimization technique that breaks the problem to smaller subproblem, and then finds the main solution by solving these subproblems. This technique has been used in registration problems [122]. Felzenszwalb [32] successfully uses this technique thanks to a tree structure used for the representation. Sebastian et al. propose this optimization framework to align two given 2D curves. Among other optimization techniques we may refer to graph matching techniques [154], Hungarian method for bipartite graph matching [8], shortest path algorithms [119], particle filtering [116], particle swarm optimization [79], least trimmed squares [23], and genetic algorithm.

Fine Registration

Most of the objective functions defined for fine registration have nonlinear form with respect to the registration parameters. The gradient descent method is one of the simplest techniques proposed to optimize these function forms. The authors in [58] propose an energy function to find the optimal FFD parameters. This function includes a data fitting part for checking the compatibility of two implicit distance fields and an external energy term for controlling the rigidity of FFD deformation. Their optimization technique is the gradient descent, which simply use the gradient information of the energy term with respect to FFD control parameters. An extended version of gradient descent algorithm, referred to as Adaptive Stochastic Gradient Descent (ASGD), has been recently used in [72] for image registration. Quadratic programming is another useful method to optimize the registration error while some constraints must be met [81]. Linear programming and second order cone programming are among the suggested method in [37] to minimize the registration error defined in L^1 norm.

Non-linear least squares form is one of the typical error terms in registration problems. Gauss Newton Algorithm (GNA) and Levenberg-Marquadt Algorithm (LMA) are among the techniques especially designed for these types of forms. Fitzgibbon in [34] proposes a non-linear least squares function based on distance field formulation, and LMA is consequently suggested to solve this objective function. Gauss-Newton type has been also proposed in [97] for minimizing the registration error in rigid case.

Chapter 3

Geometric Fitting

This chapter presents a geometric fitting method for implicit polynomial fitting. We propose a simple distance estimation to approximate the orthogonal distance of a point from an implicit curve/surface. It is computed as the height of a simplex built between the point and surface (i.e., a triangle in 2D or a tetrahedron in 3D), which is used as a coarse but reliable estimation of the orthogonal distance. This distance can be described as a function of the coefficients of the implicit polynomial. Moreover, it is differentiable and has a smooth behavior. Hence, it can be used in any gradient based optimization. Furthermore, the use of this distance in a Levenberg-Marquardt framework is shown, which is specially devoted for nonlinear least squares problems. The proposed estimation is a generalization of the gradient based distance estimation, which is widely used in the literature. Experimental results, both in 2D and 3D data sets, are provided. Comparisons with the state of the art are presented, showing the advantages of the proposed approach.

3.1 Problem Formulation

The two major approaches in implicit polynomial fitting—*algebraic* and *geometric*—are briefly presented here to show the motivations of the proposed approach. Implicit polynomial fitting aims at finding the best polynomial that describes a given set of points by means of its *zero set*. In other words, the value of the polynomial should reach zero at the location of the given data points. Let $f_c(\mathbf{x})$ be an implicit polynomial of degree d represented as:

$$f_c(\mathbf{x}) = \sum_{\substack{(i+j+k) \leq d \\ \{i,j,k\} \geq 0}} c_{i,j,k} \cdot x^i \cdot y^j \cdot z^k \quad (3.1)$$

or, in a vector form:

$$f_{\mathbf{c}}(\mathbf{x}) = \mathbf{m}^T \mathbf{c} \quad (3.2)$$

where $\mathbf{c} = [c_{0,0,0}, c_{1,0,0}, \dots, c_{0,0,d}]^T$ is the column vector of polynomial coefficients having as many components as the combination of $(d + 3)$ taken 3 at a time without repetitions: $C_3^{d+3} = \frac{(d+3)!}{d!3!}$; and \mathbf{m} is the column vector of monomials: $\mathbf{m} = \mathbf{m}(\mathbf{x}) = [x^0 y^0 z^0, x^1 y^0 z^0, \dots, x^0 y^0 z^d]^T$; the fitting problem consists of first defining a criterion—or residual error—for measuring the closeness of the zero set, $Z_f = \{\mathbf{x} : f_{\mathbf{c}}(\mathbf{x}) = 0\}$, to the given data set, and then minimizing this criterion to find the best coefficient vector \mathbf{c} .

Let $P = \Gamma_0 = \{p_i\}_1^N$ be the set of given data points with coordinates \mathbf{x} (picked up from object boundaries in 2D or surfaces in 3D); then the fitting problem is defined as:

$$\hat{\mathbf{c}} = \underset{\mathbf{c}}{\operatorname{argmin}} \operatorname{Dist}(P, f_{\mathbf{c}}) \quad (3.3)$$

where $\underset{\mathbf{c}}{\operatorname{argmin}}$ stands for the polynomial coefficient vector \mathbf{c} where the *Dist* expression attains its minimum value; there are two different approaches to find that best coefficient vector $\hat{\mathbf{c}}$ as detailed next.

In the geometric fitting approaches the distance between a point and the surface is usually defined as the shortest distance between this point and its correspondence on the surface (i.e., orthogonal distance). Thus, in the general case of geometric methods, we have the following optimization problem:

$$\operatorname{Dist}(P, f_{\mathbf{c}}) = \sum_{i=1}^N \min_{\hat{p}_i} d^2(p_i, \hat{p}_i) \quad (3.4)$$

where each \hat{p}_i is the correspondence of p_i on the surface. Here we consider the l_2 norm to calculate the distance d , and consequently a non linear least squares optimization must be solved.

Theoretically, both unknown surface parameters and the correspondences must be found simultaneously, but practically this problem is tackled by first assuming an initial surface, and then refining it till convergence is reached. So, the fitting problem is split up into two stages: *i*) point correspondence search; and *ii*) surface parameter refinement. The first stage deals with the summands in (3.4), while the second one concerns about (3.3).

Point correspondence search: Regarding the first stage two different strategies have been proposed in the literature: *(a)* finding the shortest distance by solving a non-linear system (e.g., [2, 3]); and *(b)* computing an estimation of the shortest distance (e.g., [134, 19, 42]).

Surface parameter refinement: As a result from the previous stage the set of points $\{\hat{p}_i\}_{i=1}^n$, corresponding to every p_i in the given data set has been found. Afterward, it must be followed by an optimization framework to refine the surface parameter. Although different optimization algorithms could be used (e.g., Genetic Algorithm (GA) [42], Trust region [44], Quasi-Newton method [145], Particle Swarm [1]) in the current work the Levenberg-Marquardt algorithm (LMA) [36] has been chosen since it exploits gradient information provided by the proposed distance estimation. LMA, in some sense, interpolates between the Gauss-Newton algorithm and the gradient descent (more details about LMA are given in Section 3.3).

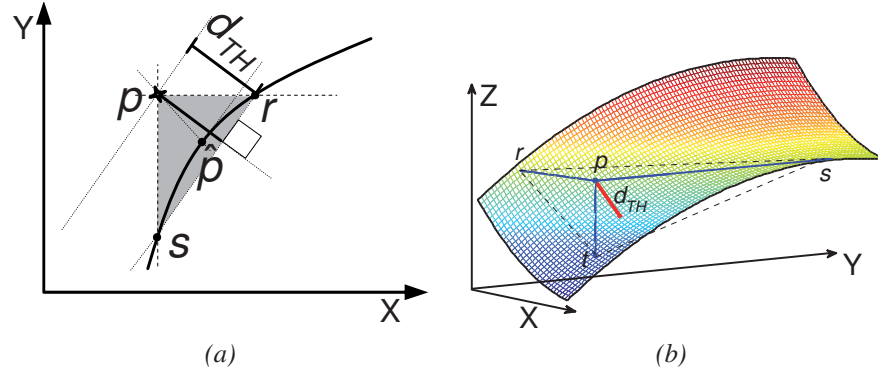


Figure 3.1: Simplex used for estimating the geometric distance: (a) 2D case; (b) 3D case.

In this chapter a novel geometric approach is presented to tackle IPs fitting through an estimation of the orthogonal distance. In spite of being focussed on the geometric framework, the polynomial coefficients can be firstly initialized by using some algebraic fitting algorithms (e.g., the 3L algorithm [11]). This initialization process is intended for speeding up the convergence of the algorithm; other strategies, for instance starting with the smallest bounding circle/sphere can be used. The proposed geometric approach consists of two stages. First, the residual error from the given set of points to the initial IP coefficients is estimated by means of the proposed approach. Then, the IP coefficients are accordingly updated through LMA. The two stages are repeated till convergence is reached; they are detailed next.

3.2 Approximated Residual Error

The first contribution of the current work lies in a direct approach to estimate the orthogonal distance. It works as follows. First a *simplex* is constructed through each point and its intersections along the coordinate axis. A simplex is a triangle in 2D and a tetrahedron in 3D, as depicted in Fig. 3.1(a) and Fig. 3.1(b) respectively. Without loss of generality, the 3D case is considered here. In this case, having constructed the tetrahedron, its height segment is considered as an approximation of the geometric distance. This tetrahedron is defined by the given point and three intersections satisfying $f_c(x, y_i, z_i) = 0$, $f_c(x_i, y, z_i) = 0$ and $f_c(x_i, y_i, z) = 0$, where $p = p_i(x_i, y_i, z_i)$ is the given point. In the particular case tackled in this work, since the fitted curve/surface is defined by the implicit polynomial (3.1), the intersecting points are found by computing the closest root of a one dimensional function to the data point.

Once the intersecting points have been obtained, a direct formula is used to estimate the geometric distance. Let r , s and t be the three intersections with the current surface, which create a triangular planar patch (see Fig. 3.1(b)). Since the volume of the tetrahedron is defined as the product of the area of each base by its corresponding height, three sets of expressions lead us to the same value. Hence, the height of the tetrahedron, d_{TH} , could

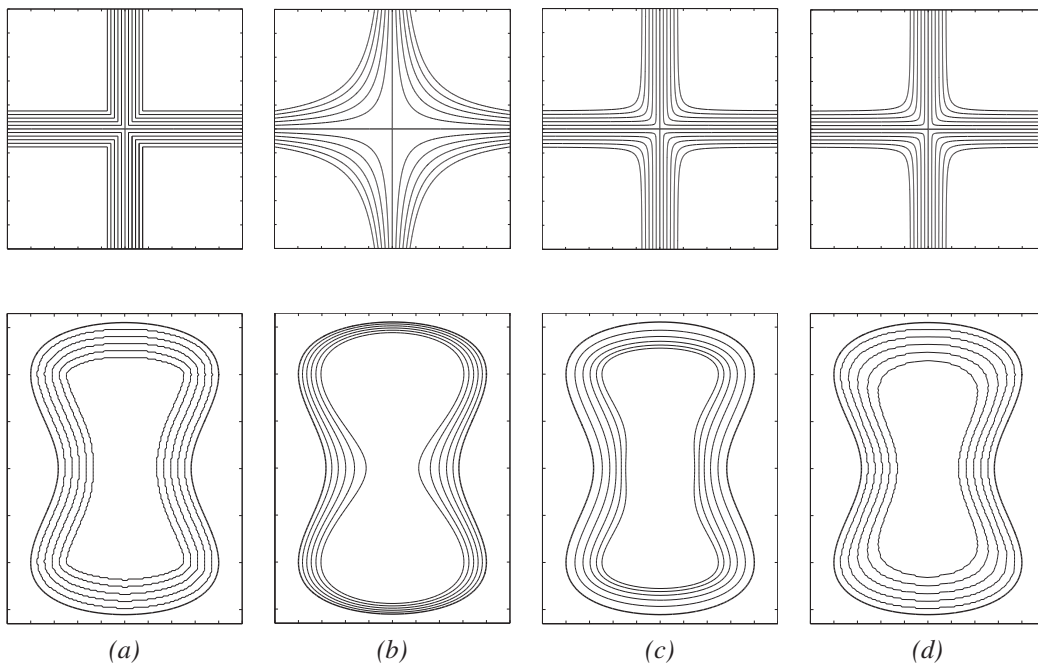


Figure 3.2: Contour of constant distance for: (a) Orthogonal distance; (b) algebraic distance; (c) [134]; (d) proposed distance estimation.

easily be computed from the following relationship:

$$\begin{aligned} d_{TH} &= (|pr| \cdot |ps| \cdot |pt|) / |\vec{r}\vec{s} \times \vec{r}\vec{t}| \\ &= \frac{|pr| \cdot |ps| \cdot |pt|}{\sqrt{(|pr| \cdot |ps|)^2 + (|pr| \cdot |pt|)^2 + (|ps| \cdot |pt|)^2}} \end{aligned} \quad (3.5)$$

where \times refers to the cross product operator between two vectors. Similar relationship can be obtained in the 2D case by using the triangle area instead of the tetrahedron volume. More details can be found in [117].

As presented above, in order to estimate the distance, the intersections of the curve/surface along the coordinate axis must be found at first. In the quadric case these intersections can be directly found ([117]). However, for higher degree cases, an iterative method should be used to find the roots. In the current implementation Newton's method has been used [129]. In case the first iteration is considered, an approximation of the root can be obtained through the first order Taylor expansion. For instance, the expansion along the x axis can be expressed as follows:

$$f(x, y_i, z_i) \simeq f(x_i, y_i, z_i) + f_x(x_i, y_i, z_i) \cdot (x - x_i) \quad (3.6)$$

where f_x corresponds to the partial derivative in the x direction and $x = r$ is the intersection of the surface with the line passing through p in the x direction. Hence, the segment $|pr|$ can be easily estimated as:

$$|pr| \simeq -f(p_i) / f_x(p_i) \quad (3.7)$$

considering similar approximations for the other two intersections, the proposed distance for the point p_i could be approximated as follows:

$$\begin{aligned} d_{TH} &\simeq \frac{|f/f_x| \cdot |f/f_y| \cdot |f/f_z|}{f^2 \sqrt{(1/f_x \cdot f_y)^2 + (1/f_x \cdot f_z)^2 + (1/f_y \cdot f_z)^2}} \\ &= \frac{|f|}{\|\nabla f\|} \end{aligned} \quad (3.8)$$

thus, the proposed distance is a generalization of the Taubin's method when the intersections are approximated.

The preciseness of the proposed distance is presented for two examples in Fig. 3.2 and compared with other approximated distances as well as the orthogonal one. The first row of the figure shows the iso-contours¹ of the set $\{(x, y) : xy = 0\}$, which consists of two intersecting lines, and the second row shows iso-contours of a regular curve $\{(x, y) : 8x^2 + (y^2 - 4)^2 - 32 = 0\}$. As illustrated in last two columns, our method and Taubin's behave similarly in the linear case (when the Jacobian matrix is linear with respect to the point coordinates). In the second example our method outperforms compared with other approximations and have a quite similar result to the iso-contours obtained by the orthogonal method.

¹Contours with the same distance from the zero set.

3.3 Implicit Polynomial Fitting

As a result from the previous stage the distance from each single data point to the current curve/surface has been found. The accumulation of all these distances provides a good criteria for curve surface fitting:

$$Dist(P, f_{\mathbf{c}}) = \sum_{i=1}^N d_{TH}^2(p_i). \quad (3.9)$$

This distance is in the least squares form where each term is non-linear with respect to the coefficient vector \mathbf{c} . It provides a straightforward method to approximate the orthogonal distance. Hence it can be used in an appropriate optimization algorithm to find the best parameters describing the given set of points. We already used this distance in a RANSAC based framework to find the quadratic surface parameters [117]. Other optimization techniques, like Genetic Algorithm (GA) [42] or Quasi-Newton method [145], have been already used in surface fitting.

The current work not only propose a simple and fast distance estimation approach but also, as a second contribution, it shows how this estimation can be used in a non-linear framework. In the current work, the Levenberg-Marquardt algorithm (LMA) has been used [36] to optimize the distance (3.9) with respect to the curve/surface parameters. LMA is specifically designed for non-linear functions in the least squares form, which is the case in (3.9). It starts from an initial coefficient vector $\mathbf{c}^0 = \mathbf{c}$, obtained by some algebraic fitting technique (as mentioned above the result from the 3L algorithm has been used as initialization). LMA updates these parameters iteratively as follows:

$$\begin{aligned} \mathbf{c}^{t+1} &= \mathbf{c}^t + \beta \Delta \mathbf{c}, \\ (J^T J + \lambda \text{diag}(J^T J)) \Delta \mathbf{c} &= J^T \mathbf{D} \end{aligned} \quad (3.10)$$

where β is the refinement step; $\Delta \mathbf{c}$ represents the refinement vector for the surface parameters; λ is the damping parameter in LMA; J is the Jacobian matrix of \mathbf{D} ; and the vector $\mathbf{D} = (d_1(\mathbf{c}), \dots, d_n(\mathbf{c}))^T$ corresponds to the distances ($d_i(\mathbf{c}) = d_{TH}(p_i)$), whose l_2 norm must be minimized. Parameter refinement (3.10) must be repeated till convergence happens.

Each iteration of LMA contains two stages: 1) distance estimation; and 2) Jacobian matrix computation. In the first stage all the intersections along the coordinate axis must be found. For this purpose Newton's method is applied to find the root of the parametric function $f(\mathbf{p}_i + t\mathbf{d})$, which is a one dimensional function with respect to t . The direction vector \mathbf{d} is set to $e_1 = (1, 0, 0)^T$, $e_2 = (0, 1, 0)^T$ and $e_3 = (0, 0, 1)^T$ for each axis. Having computed all the intersections along the coordinate axis the terms: $|pr|$, $|ps|$ and $|pt|$, and consequently the distance (3.5), can be estimated. As mentioned above, it should be noticed here that if we stop the Newton's method after one iteration, the proposed distance will be computed easily through (3.8) which is the same as [134].

In order to handle LMA, the value of the functional (3.9) and its partial derivatives, which are used to build the Jacobian matrix, should be provided. These values show the sensitivity of each d_i in (3.5) with respect to the parameter vector \mathbf{c} . The Jacobian matrix could be

directly derived through the differentiation rules as follow:

$$J_{ij} = \partial_j(d_i) = [|\vec{r}\vec{s} \times \vec{r}\vec{t}| \cdot \partial_j(|pr| \cdot |ps| \cdot |pt|) - (|pr| \cdot |ps| \cdot |pt|) \cdot \partial_j|\vec{r}\vec{s} \times \vec{r}\vec{t}| / |\vec{r}\vec{s} \times \vec{r}\vec{t}|^2] \quad (3.11)$$

where $\partial_j = \partial/\partial c_j$ is the differentiation operator with respect to parameters. Since the intersection r , s , and t lies on the surface, $|\vec{r}\vec{s} \times \vec{r}\vec{t}|$, $|pr|$, $|ps|$ and $|pt|$ can be implicitly expressed as a function of the surface parameters. In order to calculate each term of (3.11), the implicit differentiation rule must be used for each intersection. For instance, for a given point p_i , the term $|pr|$ is computed just by considering its x -component: $|pr| = (r_i^x - p_i^x)$ and its partial derivatives as follows:

$$\partial_j|pr| = \frac{dr^x}{dc_j} = -\frac{\partial f/\partial c_j}{\partial f/\partial r^x} = -\frac{m_j(r)}{f_x(r)} \quad (3.12)$$

where $m_j(r)$ is the j -th monomial component calculated in the intersection. The term $|\vec{r}\vec{s} \times \vec{r}\vec{t}|$ can be expressed based on the intersections as mentioned in (3.5). Then its partial derivatives can be computed based on the other single terms.

Having estimated the geometric distance (3.5) and its Jacobian matrix through (3.11) the LMA algorithm iterates equations (3.10) till convergence is reached. In the current work convergence criteria has been defined using the deviation between the IP normal and the local normal at each point (see illustration in Fig. 3.3). This criteria, on the one hand is easy to be computed; and on the other hand, it is robust enough to be used with different geometries. Note that the local normal at each data is already computed during the initialization stage (the 3L algorithm). So the only required computation is regarding the angle estimation:

$$\theta_i = \cos^{-1}\left(\frac{n_i \cdot \nabla f(p_i)}{\|\nabla f(p_i)\|}\right) \quad (3.13)$$

additionally, since $\cos^{-1}|_{[0,1] \rightarrow [0,\pi/2]}$ is monotonic, just the absolute value of the inner expression, without calculating the cosine inverse, is considered. Therefore the criterion used for measuring the goodness of the current fitting result is:

$$\xi(\mathbf{c}) = \frac{1}{N} \sum_{i=1}^N \left| 1 - \frac{n_i \cdot \nabla f(p_i)}{\|\nabla f(p_i)\|} \right| \quad (3.14)$$

where N is the number of points in the original data set. LMA iterates while (3.14) decreases more than a user defined threshold $\Delta\xi = \xi_t - \xi_{t-1}$ or a maximum number of iterations is reached.

3.4 Experimental Results

Our experiments with the proposed distance estimation comprises two part. In the first part we study the accuracy of this distance for error approximation. This is shown to be useful for RANSAC based approaches to estimate the optimal curve/surface parameters in a random scheme. In the second part we apply the proposed distance estimation in a non-linear optimization framework (LMA) to find the parameters in an iterative scheme.

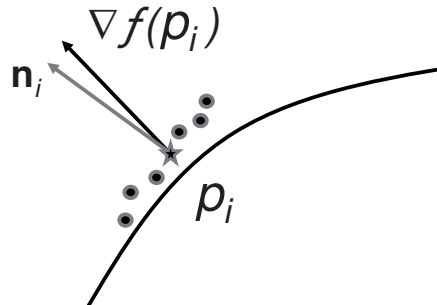


Figure 3.3: Convergence criteria defined as the deviation between the IP normal and the local normal is each point.

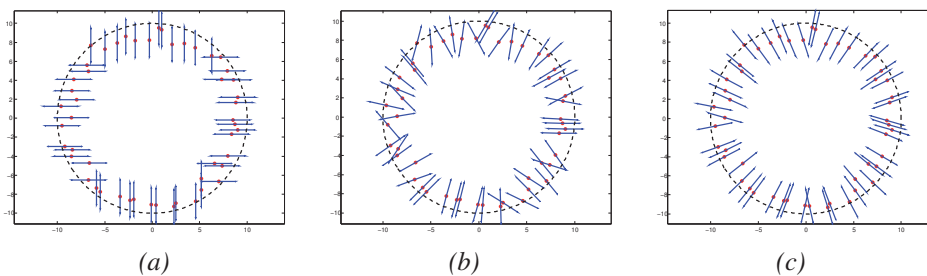


Figure 3.4: Estimated normal vectors for raw data points by using different approaches. (a) PGD approach. (b) Based on PCA in a local neighborhood. (c) Proposed approach (THI direction).

3.4.1 Residual Error Approximation

This section provides results obtained with synthetic and real data points. Synthetic data points are used as ground truth for evaluating results of the proposed approach when noisy data are considered. Real data points have been obtained with the K2T structured light camera system at the University of South Florida and are used for empirically validating the proposed approach.

Synthetic Data: In this section results from 2D and 3D synthetic data points are presented. The first illustration correspond to a set of 50 noisy 2D data points, randomly distributed over a circle. Fig. 3.4 presents normal vectors \vec{n}_0 estimated by using different approaches. An outer, and concentric, circle is just depicted for highlighting these orientations. Fig. 3.4 (a) shows the orientations obtained by using the PGD criterion [19]; since the shortest distance along $\{(1, 0), (0, 1)\}$ is selected, the estimated \vec{n}_0 is parallel to one of the axes. Fig. 3.4 (b) presents normal vectors estimated by using the approach presented in [42]. Since this approach is based on a PCA analysis in a small neighborhood centered at each point, it is easily affected by noisy data. It can be appreciated that some estimations are completely wrong. On the contrary, results obtained with the proposed approach are robust to noisy data (see Fig. 3.4 (c)). In most of the cases estimated normal vectors are correctly oriented towards the center of the circle.

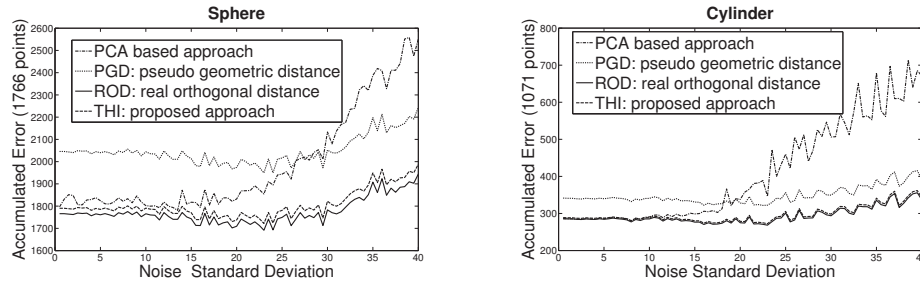


Figure 3.5: Accumulated error by using different approaches as a function of noise standard deviation.

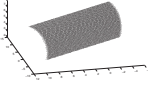
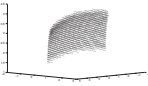
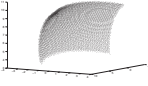
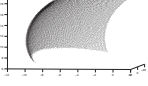
Fig. 3.5(*top*) presents results obtained with a set of 3D data points uniformly distributed over the surface of a sphere. An accumulated real orthogonal distance (RGD), from the whole set of 3D data points to an outer and concentric sphere, is used as ground truth. Estimated distances, also accumulated over the whole set of points, are computed by using the three different approaches (i.e., PGD [19], PCA [42] and THI). The experiment has been repeated by increasing the standard deviation of the noise added to the set of 3D points. It can be seen that the proposed approach behaves quite similar to the real orthogonal distance independently of the added noise. On the contrary, as it was expected, results obtained with the PCA based approach are easily biased when the noise standard deviation grows. Similar results have been obtained by using 3D data points uniformly distributed over the surface of a cylinder (Fig. 3.5(*bottom*)). In this case, distances to an outer and concentric cylinder are considered. The real orthogonal distance, accumulated through the whole set of points, is used as ground truth. It can be appreciated that the distances estimated with the proposed approach (THI) are very similar to the real values; on the other hand values computed with the proposed approach are not affected by noisy data.

Real Data: The objective of this section is to show the viability of the proposed approach by analyzing surfaces fitted using different metrics. A simple RANSAC-based [33] fitting approach has been used for finding the best set of quadric surface parameters in each case; developing a novel fitting technique is out of the scope of current work. The prior knowledge about the geometry of the surface to be fitted and a 3D data preprocessing step are used to reduce the size of active parameters in c . The preprocessing step consists in rotating and translating the set of 3D data points in order to place the world coordinate system at the centroid of the set of points and to orient it according to the main axis of the 3D data. Once data have been transformed to the new coordinate system the RANSAC-based fitting approach is used for computing the surface parameters (parameter vector² \mathbf{b}).

Table 3.1 shows the original images (left column) as well as the patches extracted for fitting (second column) that only contains 3D data points from a quadric surface. Surface parameters have been obtained through the RANSAC-based fitting by using three different estimations of the orthogonal distance. Results are presented on columns 3 to 5 of Table 3.1. The fastest results were obtained by using the PGD estimation; however, as presented in

²It is defined according to the kind of surface to be fitted—e.g., cylinder: $(\mathbf{b} = \{b_2 = b_3, b_8, b_9\})$.

Table 3.1: Surfaces fitted by using different metrics; Rad: radius; C: center; AEr: Accumulated Error according to the used metric; CPU time in sec.

3D Data to be Fitted	PGD [19]	PCA [42]	THI: Prop. Ap.
 5100 points	Rad = 4.286 C = (0.08, 3.92) AEr = 805.27 CPU = 25.378	Rad = 4.063 C = (-0.07, 3.64) AEr = 765.45 CPU = 310.391	Rad = 4.091 C = (-0.06, 3.68) AEr = 685.47 CPU = 29.14
 7875 points	Rad = 3.269 C = (2.4, 2.1, 1.4) AEr = 1.40e+3 CPU = 19.596	Rad = 3.184 C = (2.4, 1.9, 1.4) AEr = 1.65e+3 CPU = 1013.415	Rad = 3.359 C = (2.4, 1.9, 1.2) AEr = 1.05e+3 CPU = 87.784
 4945 points	Rad = 7.168 C = (3.7, 3.1, 4.7) AEr = 757.35 CPU = 18.203	Rad = 6.457 C = (3.0, 2.9, 4.9) AEr = 653.31 CPU = 429.6	Rad = 6.751 C = (3.4, 3.0, 4.9) AEr = 573.98 CPU = 62.537
 7700 points	Rad = 10.132 C = (0.6, 2.0, 0.2) AEr = 1.12e+3 CPU = 28.184	Rad = 10.805 C = (0.02, 1.5, 0.1) AEr = 963.364 CPU = 1142.977	Rad = 10.289 C = (0.4, 1.7, 0.4) AEr = 862.88 CPU = 103.046

the synthetic case, this distance estimation is not accurate and could leads to a biased result. Note that in the real data case it is not possible to judge the goodness of obtained results since there are not ground truth values for the surface parameters. In spite of that, it can be appreciated that in most of the cases the surface parameters obtained with PCA based approach are very similar to the ones computed using the proposed approach. In other words, it could be assumed that PGD based approach results in a more inaccurate set of surface parameters. Finally, although the proposed approach gives similar results to the PCA based approach, it should be highlighted that it runs up to eleven times faster. Furthermore, recall that PCA based approach needs the right tuning for the neighbor point selection.

3.4.2 Implicit Polynomial Fitting

The proposed method, which belongs to the geometric fitting category, is implemented and compared with the most important methods in the literature, both algebraic and geometric. The results presented in this section are evaluated using the *fitting error* (FE) computed for every single points with [2]. It is used to obtain a quantitative criterion for comparison, which is referred to as *Accumulated Fitting Error* (AFE): $AFE = \sum_{i=1}^N FE_i$. In all the cases the given data points are centralized and scaled between $[-1,1]$. The parameters of initialization (3L algorithm), optimization (LMA) and stopping criterion are empirically set up as presented in Table 3.2. The same initialization and stopping criterion have been used once the proposed

approach is compared with other approaches.

Table 3.2: Parameters set up.

Initialization (the 3L alg.)		Optimization (LMA)		Stopping criterion	
δ	ϵ	λ	β	$\Delta\xi$	# iteration
0.1	1	0.01	0.5	<0.1	<25

In the two dimensional case, different sets of points picked from quadric contours sampled with non-uniform distributions have been fitted with the proposed approach and compared with other approaches. Fig. 3.6(a) depicts the result of the proposed method when a non-uniformly distributed 2D data set is fitted. Both algebraic and proposed method converge to a similar result, but problems arise when noise is added to the points. Fig. 3.6(b) highlights the robustness of the proposed method to noise; whereas the algebraic one misses the elliptic structure of the data, and fits the patch as a split hyperbola. Fitzgibbon *et al.* [35] propose a fitting method just for 2D elliptic cases based on algebraic approaches. From this simple example, one can understand the hardship for algebraic methods when the function space is bigger than the quadratic one.

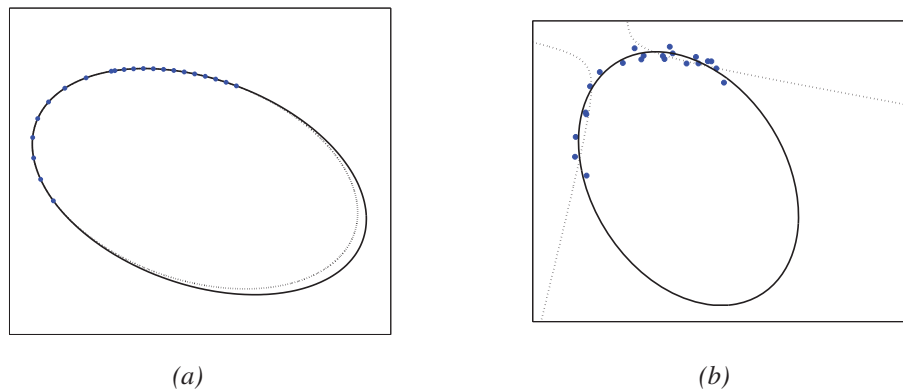


Figure 3.6: Fitting a set of points from an ellipse. (a) Without noise: Algebraic (dotted line) and proposed method (solid line). (b) Noisy data case: Algebraic (dotted line) misses the elliptic structure, while the proposed approach (solid line) reaches a good result.

The proposed approach is also implemented for fitting higher degrees IPs. Figure 3.7 shows 2D contours fitted by fifth and sixth degree IPs (depending on the shape complexity) using: the 3L algorithm (Fig. 3.7(a)); the approach proposed in [134] (Fig. 3.7(b)); the proposed approach (Fig. 3.7(c)); and a non-linear orthogonal distance based approach [2] used as a ground truth (Fig. 3.7(d)). The fitting error, computed over the whole set of points with [2], is used as a quantitative criterion for comparison. In all the cases the accuracy obtained with the proposed approach considerably improves the one obtained with the 3L algorithm, and in most of the cases gives better results than [134]; actually, it is comparable (in one case better since the stopping criteria has been reached, see fourth row) to the results

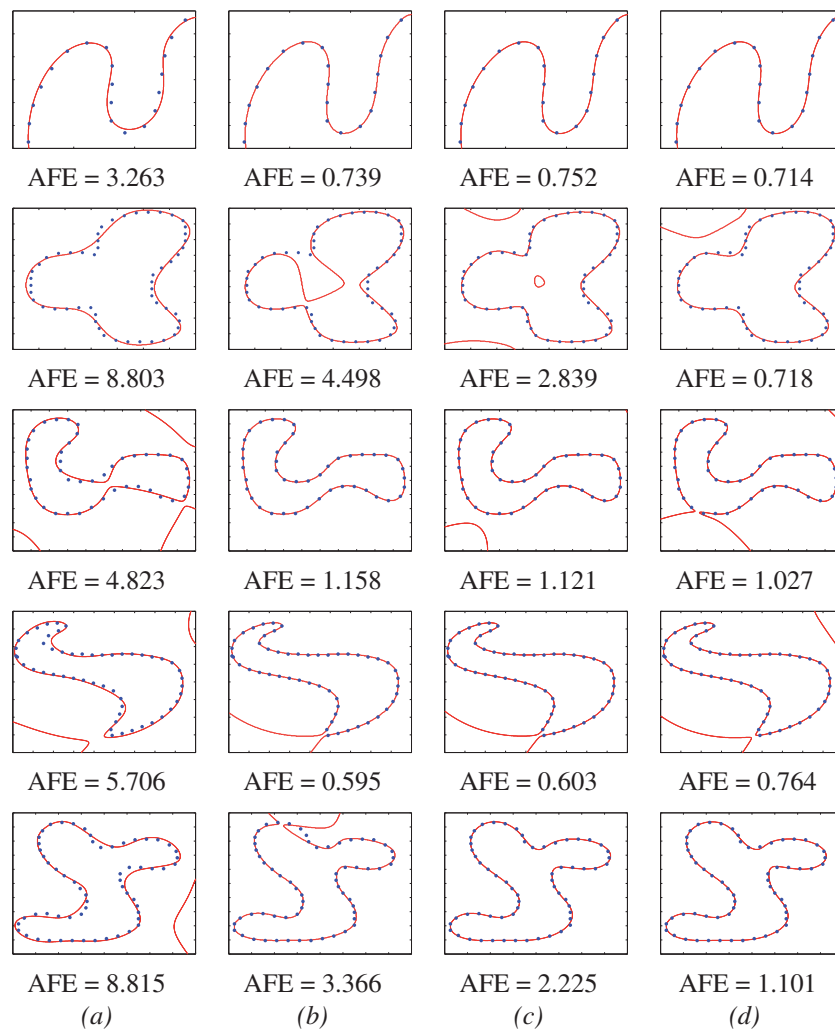


Figure 3.7: 2D contours fitted by fifth (1st and 2nd rows) and sixth (3rd, 4th and 5th rows) degree IPs, results from: (a) the 3L algorithm; (b) [134]; (c) proposed approach; (d) [2], which is used as ground truths. AFE shows the accumulated fitting error respectively. The fourth row shows a case where [2] stops due to the maximum iteration criterion.

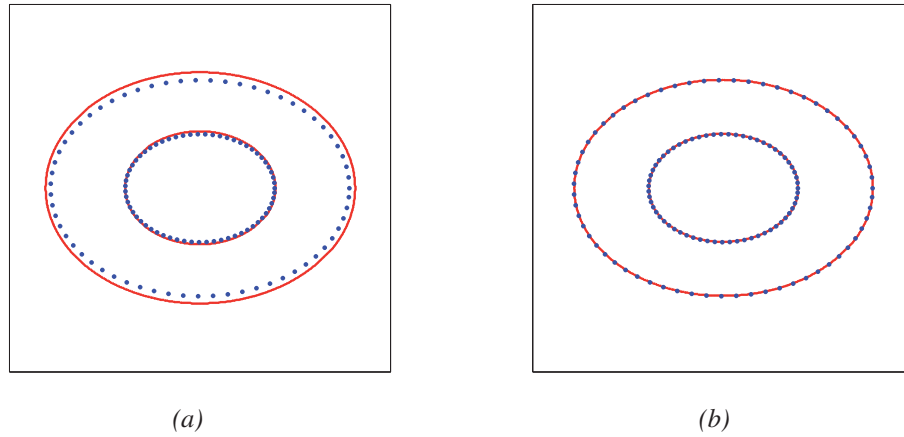


Figure 3.8: Fitting two concentric ellipses. (a) Result from the 3L algorithm. (b) Result from the proposed approach.

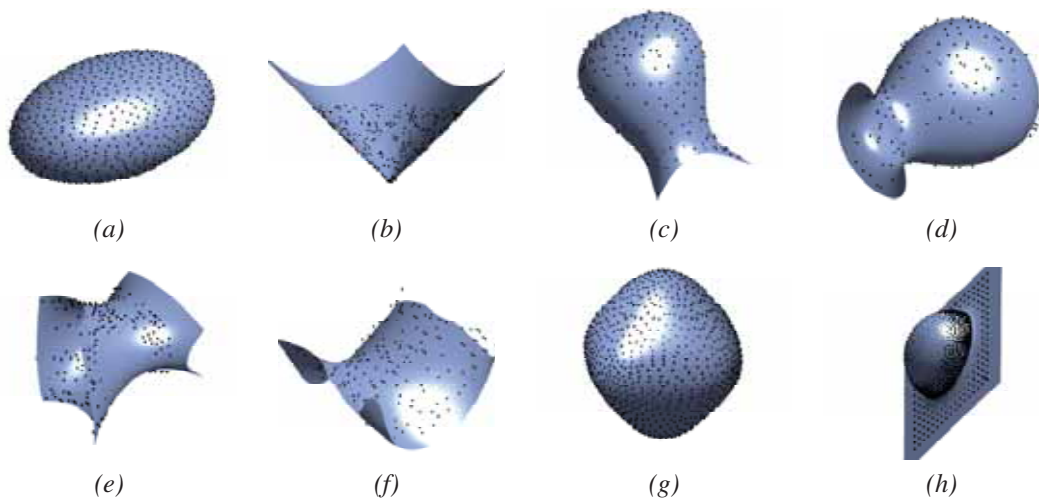


Figure 3.9: Synthetic data sets fitted with the proposed approach.

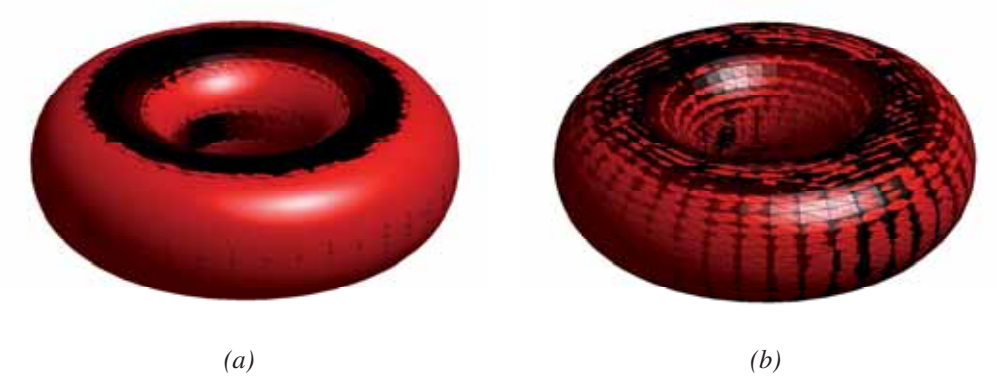


Figure 3.10: Solid surface representing a fourth degree IP; wire frame is used to visualize given data points. (a) IP obtained from the 3L algorithm. (b) Result from the proposed approach (note the similarity between wire frame and the surface from the computed IP).

Table 3.3: Synthetic data set: AFE corresponding to the illustrations presented in Fig. 3.9.

	IP degree	3L alg. [11]	Orthogonal fitting [2]	Prop. approach
Fig. 3.9(a)	second	9.58	5.56	5.39
Fig. 3.9(b)	second	2.42	1.32	1.20
Fig. 3.9(c)	fifth	1.89	0.69	0.68
Fig. 3.9(d)	third	1.93	1.73	1.69
Fig. 3.9(e)	fifth	2.67	1.28	1.03
Fig. 3.9(f)	third	3.80	1.29	1.31
Fig. 3.9(g)	fourth	2.20	0.50	0.51
Fig. 3.9(h)	third	1.17	0.42	0.40

obtained when the non-linear approach is used. Although out of the scope of the current work, it should be mentioned that in the 2D case the proposed approach is about ten times faster than [2]. Finally, another challenging 2D shape defined by two concentric ellipses has been fitted by a fifth degree IP using the proposed approach; Figure 3.8(a) shows the result from the 3L algorithm used as initialization of the proposed approach. The final result is depicted in Fig. 3.8(b).

The proposed approach has been also evaluated with 3D data sets, both synthetic and real data sets were fitted with low and high degree IPs. On average, in the 3D case, the proposed approach is not as good as in the 2D case, but it is about twice faster than [2]. Figure 3.9 shows eight different results obtained with the proposed approach; in all the cases the results are quite similar to the ones obtained with [2], and considerably better than those obtained with [11]. Table 3.3 presents the Accumulated Fitting Error obtained with the different approaches for a quantitative comparison. Note that these results were obtained once the stopping criteria

Table 3.4: Data set from AIM@SHAPE: AFE corresponding to the illustrations presented in Fig. 3.11.

	IP degree	3L alg. [11]	Orthogonal fitting [2]	Prop. approach
Fig. 3.11(a)	fourth	8.17	5.31	5.37
Fig. 3.11(b)	seventh	6.17	5.76	5.85
Fig. 3.11(c)	seventh	1.07	0.56	0.63
Fig. 3.11(d)	seventh	3.28	1.51	1.68
Fig. 3.11(e)	sixth	3.30	2.32	2.27
Fig. 3.11(f)	fourth	4.57	1.90	1.88
Fig. 3.11(g)	sixth	3.41	2.92	2.94
Fig. 3.11(h)	sixth	7.60	6.61	6.62

has been reached; if larger number of iterations are allowed, [2] achieves better results. The proposed algorithm has been tested with a more challenging 3D data set with a different topology; Figure 3.10 presents results from both the 3L algorithm (AFE=0.06), which is used as an initialization of proposed approach, and the final result obtained after 10 iterations (AFE= 1.00×10^{-4}). In this case a fourth degree IP has been used (solid surface); given data points are represented by means of a wire frame just for a visual comparison.

In addition to the synthetic objects, a data set from AIM@SHAPE³ has been used for evaluating the proposed approach. Figure 3.11 shows eight illustrations of fourth, sixth and seventh degree IPs obtained with the proposed approach. Table 3.4 presents the Accumulated Fitting Error obtained with the different approaches for a quantitative comparison. Figure 3.12 illustrates the independence to initial guess by using a sphere covering the given data set as an initialization (see Fig. 3.12(a)). First, second and third iterations of the proposed approach are shown in Fig. 3.12(b), (c) and (d) respectively; result obtained after 25 iterations is already depicted in Fig. 3.11(a). Surface parameter refinements through these iterations are depicted in Fig. 3.13. Figure 3.13(a) corresponds to the evolution of the 35 IP coefficients, while Fig. 3.13(b) shows how the AFE decreases with the iterations. Finally, Fig. 3.13(c) depicts the accumulated angle (3.13) used as a convergence criteria. It should be mentioned that this criteria has a similar behavior than Fig. 3.13(b) but its complexity is considerably lower.

3.5 Conclusions

This chapter presents a novel geometric approach for 2D/3D implicit polynomial fitting, which is based on a fast geometric distance estimation. Despite other geometric estimations, which are based on a single direction to find the foot-point associated to each data point, the proposed one is based on two or three directions (depending on the data dimension). The

³<http://shapes.aimatshape.net/>

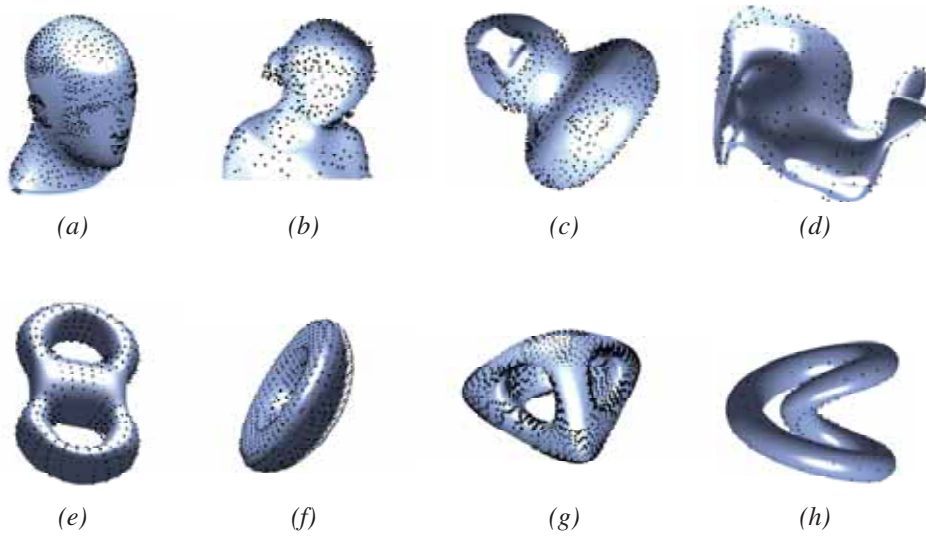


Figure 3.11: Data set from AIM@SHAPE fitted with the proposed approach.

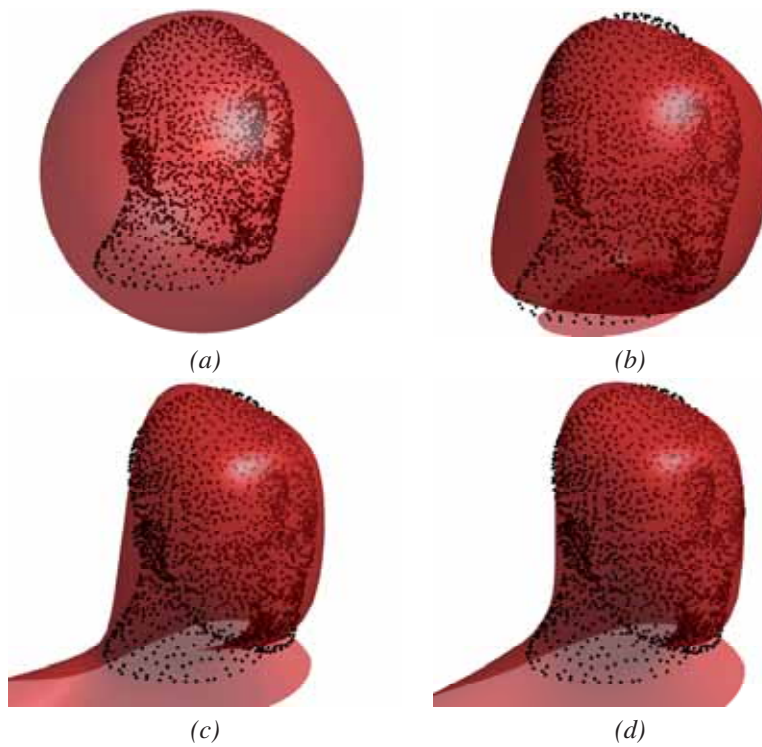


Figure 3.12: (a) Fitting with a rough initialization. (b), (c) and (d) First, second and third iterations respectively.

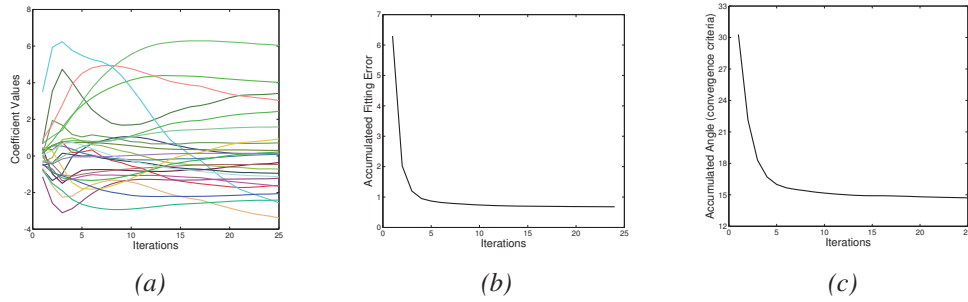


Figure 3.13: Parameter evolution of Fig. 3.12 along 25 iterations: (a) IP coefficient values; (b) AFE; (c) accumulated angle used as convergence criteria.

smoothness and accuracy of the proposed distance have been shown. Additionally, the implicit connection between this distance and the IP coefficients has been presented and shown to be differentiable. This property allows the use of any gradient based optimization techniques. In the current work the Levenberg-Marquardt algorithm is applied to find the best set of surface parameters in an iterative way. Comparisons with state of the art techniques are presented. Moreover, the proposed distance is proved to be a generalization of the distance presented in [134].

Chapter 4

Algebraic Fitting

This chapter presents a linear fitting method based on the implicit B-spline representation. IBS is a flexible representation capable to describe complex objects both in 2D and 3D. In this paper a novel IBS fitting is presented in the 3L framework that is linearly solved based on least squares formulation. All the computations are based on the blending functions, which are couples of simple quadratic or cubic patches. We show how the regularization term and additional constraints can be considered in the proposed framework. The proposed method is efficiently implemented acquiring a low complexity. Experimental results show the flexibility and accuracy of the proposed algorithm to describe different boundaries and objects. Comparisons with other fitting methods both in algebraic and geometric frameworks highlight the advantage of the proposed approach.

This chapter focuses on the algebraic distance based approaches. These methods are computationally fast, but less accurate compared to the geometric ones. Instead of considering the orthogonal distance, they try to find the optimal parameters through imposing some algebraic constraints on the zero set defined by the parameters. The minimization formulations of these methods are in the linear least squares form that can be efficiently solved in a single step without iteration.

The fitting algorithm used in this chapter is based on the 3L algorithm [11] originally proposed for IP fitting. This algorithm belongs to algebraic fitting methods that use some geometric clues. It uses two additional offsets, inside and outside of the original data Fig. 4.1(*left*). Then, it tries to trap the zero set between these additional offsets, and somewhere close to the original data set. In the first part of this chapter a method for relaxing the original 3L algorithm is proposed to make it more flexible. It adapts the expected values to be obtained in the inner and outer offsets using some geometric information. This process is continued while some consistency criteria is met.

In the second part we extend the 3L algorithm for Implicit B-Spline (IBS) fitting. This method results in a flexible and accurate enough representation. In addition the proposed

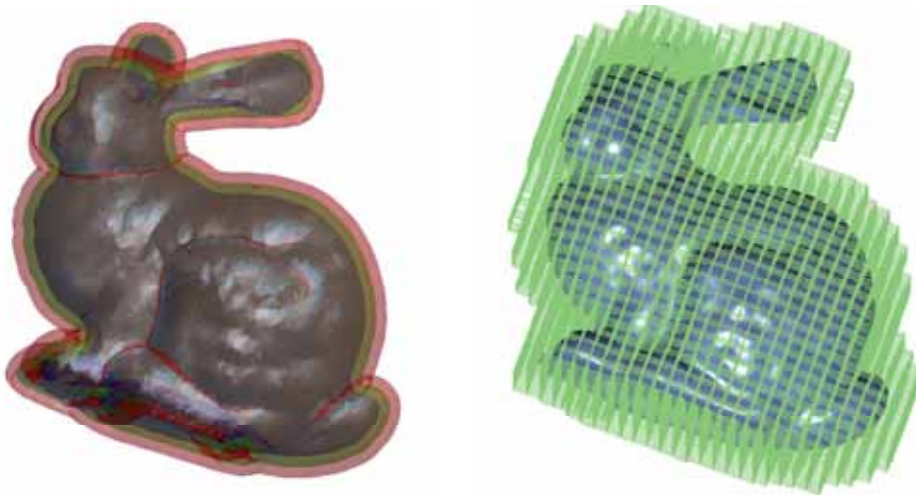


Figure 4.1: The proposed efficient 3L-IBS method for fitting; (*left*) three level sets; (*right*) the active control points.

framework allows to consider additional constraints imposed by user. These constraints can control either the position or the orientation of the zero set in some specific points defined by the user. All these formulations are presented in a unified framework that will be referred to as 3L-IBS. Moreover, for the sake of simplicity, all the formulations are presented based on the blending functions, which are couple of quadratic or cubic patches that assemble together. This methodology helps the reader to implement all the presented idea. As the final contribution, a modification of the proposed 3L-IBS algorithm is provided, which is computationally efficient. This technique defines the optimization on the active parts of the control lattice, hence a lower solution space will be considered instead. For instance Fig. 4.1(*right*) illustrates the active control lattice that contains around $11k$ parameters instead of $125k$.

4.1 Problem Formulation

In the algebraic fitting domain, least squares formulation is a simple and common framework for curve/surface fitting, as previously mentioned. Since the objective function $E = \mathbf{c}^T \mathbf{M}^T \mathbf{M} \mathbf{c}$ in (2.11) is quadratic with respect to the parameter vector, it can be linearly minimized. But minimizing this term might lead to the trivial answer $\mathbf{c} = \mathbf{0}$, thus there must be an additional constraint to avoid this solution. One of the techniques, to regularize the solution space, is to force the coefficient vector to satisfy $\sum_i c_i = 1$.

Although simple and computationally efficient, this least squares solution might lead to unstable results, since the presented algebraic metric does not use geometric clues of the data set. Hence the metric should be strengthened in order to approximate the orthogonal distance between the zero set and the given point set. Moreover, some additional constraints with

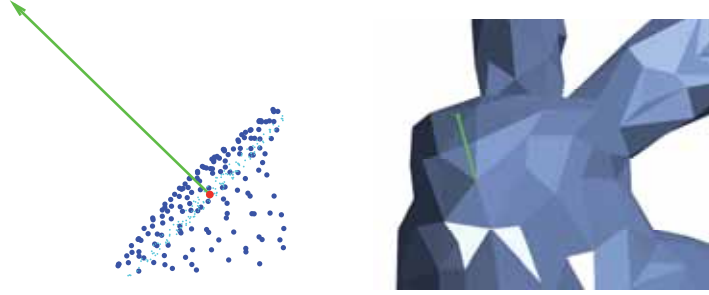


Figure 4.2: Computing the normal vector: (*left*) using the principal component analysis; (*right*) using the neighboring triangle orientations in the triangular mesh.

geometric meaning could be imposed to have a better description.

The 3L algorithm is an algebraic method addressing this problem [11]. This method is originally proposed for fitting IPs. It generates an inner offset $\Gamma_{-\delta}$ and an outer one $\Gamma_{+\delta}$ at the distance $\pm\delta$ from the original data set Γ_0 . These sets are generated along the normal vectors that can be obtained either through the triangular mesh [110] or through the PCA applied in each local neighborhood. Figure 4.2 illustrates these two techniques to compute the local normal vector. Moreover, Fig. 4.3(b) depicts how the additional offsets are supporting a data set in 2D.

Having provided all the three level sets: $\{\Gamma_{-\delta}, \Gamma_0, \Gamma_{+\delta}\}$, the 3L algorithm¹ tries to find the optimal function reaching $+\epsilon$ inside, $-\epsilon$ outside, as well as zero for the original data set. This set of constraints can be formulated as an over-determined system of equations $\mathbf{M}_{3L}\mathbf{c} = \mathbf{b}$ where

$$\mathbf{M}_{3L} = \begin{bmatrix} \mathbf{M}_{\Gamma_{-\delta}} \\ \mathbf{M}_{\Gamma_0} \\ \mathbf{M}_{\Gamma_{+\delta}} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -\epsilon \\ \mathbf{0} \\ +\epsilon \end{bmatrix} \quad (4.1)$$

\mathbf{M}_{3L} is a block matrix of \mathbf{M}_{Γ_0} , $\mathbf{M}_{\Gamma_{+\delta}}$, and $\mathbf{M}_{\Gamma_{-\delta}}$ containing the monomials calculated in the original data set, the inner and the outer offsets. This over-determined system of equation can be seen as a summation of squared form (see (2.12)).

The column vector $\pm\epsilon$ in (4.1) contains the expected value of f in the additional offsets. Since the shape of the zero set is the same for \mathbf{c} and $\kappa\mathbf{c}$ for any constant κ , we can fix the value $\epsilon = 1$ without disturbing the final zero set.

The overdetermined system proposed by the 3L algorithm can be considered as a minimization of an energy function:

$$\begin{aligned} E(\mathbf{c}) &= \|\mathbf{M}_{3L}\mathbf{c} - \mathbf{b}\|^2 = (\mathbf{M}_{3L}\mathbf{c} - \mathbf{b})^T (\mathbf{M}_{3L}\mathbf{c} - \mathbf{b}) \\ &= \mathbf{c}^T \mathbf{M}_{3L}^T \mathbf{M}_{3L} \mathbf{c} - 2\mathbf{c}^T \mathbf{M}_{3L}^T \mathbf{b} + \mathbf{b}^T \mathbf{b} \end{aligned} \quad (4.2)$$

¹3L stands for 3 Levels considered in the algorithm.

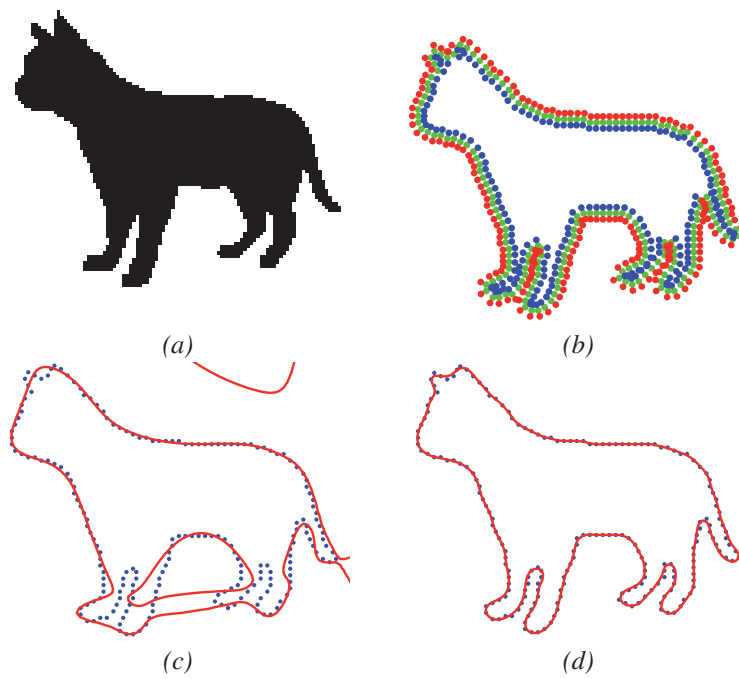


Figure 4.3: Fitting a set of 2D points: (a) cat silhouette [127] (b) data set and additional offsets; (c) the fitting result by using a degree 30th implicit polynomial (496 coefficients) [2]; (d) the fitting result from the proposed algorithm using a 30×30 IBS (391 active coefficients).

It must be mentioned that this least squares problem does not have any trivial solution, hence there is no need for any extra constraint. This function is quadratic with respect to the coefficient vector. Using matrix calculus the derivative of this function with respect to vector \mathbf{c} can be easily derived:

$$\partial F / \partial \mathbf{c} = (\mathbf{M}_{3L}^T \mathbf{M}_{3L}) \mathbf{c} + (\mathbf{M}_{3L}^T \mathbf{M}_{3L})^T \mathbf{c} - 2\mathbf{M}_{3L}^T \mathbf{b} \quad (4.3)$$

Then, the least squares solution for \mathbf{c} is obtained by setting the derivative equal to zero:

$$\mathbf{c} = \mathbf{M}_{3L}^\dagger \mathbf{b} = (\mathbf{M}_{3L}^T \mathbf{M}_{3L})^{-1} \mathbf{M}_{3L}^T \mathbf{b} \quad (4.4)$$

where \mathbf{M}_{3L}^\dagger denotes the pseudoinverse of \mathbf{M}_{3L} .

The 3L algorithm is fast and more stable compared to other algebraic fitting techniques. It finds the solution directly from the monomial matrix without any iterative computation. The monomial matrix is easily computed in the given set of data point. In spite of being computationally efficient it suffers from some problems as detailed below.

The first problem is regarding the numerical instability problem which is common in linear least squares method. In [110] a ridge regression (RR) method is presented to increase the stability by adding a regularization term to (4.2). This regularization term is in the form of $\sum \mathbf{c}_i^2 = \mathbf{c}^T \mathbf{c}$, and its effect is controlled by the regularization parameter μ . This term is quadratic, and it can be generalized to $\mathbf{c}^T \mathbf{D} \mathbf{c}$ where \mathbf{D} is a diagonal matrix containing different weights for each coefficient. The optimal coefficient can be still found as a least squares solution similarly to (4.4):

$$\mathbf{c} = (\mathbf{M}_{3L}^T \mathbf{M}_{3L} + \mu \mathbf{D})^{-1} \mathbf{M}_{3L}^T \mathbf{b}. \quad (4.5)$$

The diagonal matrix \mathbf{D} can be easily chosen as the identity matrix, or more elegantly chosen in proportion to the diagonal of $\mathbf{M}_{3L}^T \mathbf{M}_{3L}$. The authors in [110] impose some conditions on the diagonal matrix in order to obtain an algebraic invariant result. Although more stable, the RR method leads to a coarser result.

The second problem is due to the generating additional offsets. For this purpose, a normal direction should be approximated first through a triangular mesh or PCA. Then the problem is how to determine the value δ which shows the distance of the offsets to the original set. The authors in [11] choose δ as a percentage of the object size. Moreover, it depends on the point density, the complexity of the object and the degree of IP. The 3L algorithm might be affected by noise when a small value for δ is chosen, and might be inaccurate for larger values.

The third problem is how to choose the value ϵ that shows how the implicit function behaves in the additional offset. It should be noticed that the additional offsets are used in order to guarantee the stability of the final result. But insisting on these additional constraints affects the accuracy of the final result. Our experiments show the values obtained by the optimal IP in the additional offset are not equal everywhere, and might change from point to point. Hence a varying ϵ should be considered in different points. In [105] a technique to adjust these values proportionally is presented.

The final problem considered in the current work is the limitation of implicit polynomials to define the solution space of the 3L algorithm. Increasing the IP degree helps us to describe

more complex objects, but increases the instability as well. In the current work implicit B-splines are used as a flexible representation. In this article it is shown how to produce a more flexible implicit representation, which is stable at the same time.

4.2 Relaxed 3L-IP Fitting

As mentioned in the previous section the 3L algorithm is based on the construction of two additional data sets (level sets) that are determined from the original data set. Although the algorithm produces a result within one pass and no iterative computations are required, it has three major problems as detailed below.

First of all, like other algebraic methods, the 3L algorithm has also numerical instability problems. In a recent work, [110] proposes a statistical approach to increase the global stability of the 3L algorithm. This method is based on the ridge regression (RR), which is a way to regularize the block matrix M_{3L} . RR improves the condition number of the block matrix by adding a diagonal matrix as mentioned in (4.5).

Moreover, in [110] a diagonal matrix is introduced such that the Euclidean invariant property is maintained. Unfortunately, this method leads us to a coarser fitting result than the original 3L algorithm, although more stable.

Second, the accuracy of the fitting result depends on the δ value used for computing the two additional level sets. In the original paper [11] the authors propose to define these offsets as a percentage of the object size; however there is no rule for setting the right value. Actually, in [11] the authors present some experiments showing the variability of the results depending on that value (varying these offsets from 0.5 percent till 20 percent of the object size).

Third, it should be noticed that the whole set of points contained in the three level sets $\{\Gamma_{-\delta}, \Gamma_0, \Gamma_{+\delta}\}$ defining the block matrix M_{3L} are equally considered when the least squares solution is obtained (4.5). On the other hand, the initially given problem only contains the level set Γ_0 . Hence, it is easy to conclude that the constraints added for stabilizing the fitting solution could affect the accuracy of the result.

Regardless to the pros and cons of these improvements, none of them pay attention to the expected values of the additional data in (4.1). Indeed, forcing the IP reaching $+c$ inside and $-c$ outside could lead us to a biased result. However, changing the value of c will give the same zero set. What makes the fitting result biased is the constant proportion considered for all the inner and outer additional data points. Figure 4.4(c) depicts an illustration where the perfect result² would only be obtained after relaxing the expected values for the inner and outer offsets, while keeping them to zero for original data. This figure shows the original data and the expected height for the supplementary offsets; and it also illustrates how these values change from point to point. The main contribution of the current work is to estimate these expected values in order to reach a more accurate fitting.

In the next section the last two problems mentioned above are tackled looking for a more

²This result corresponds to the fitting of the given data points computed with [2] (see Section 2.1.2), inner and outer data sets are not considered.

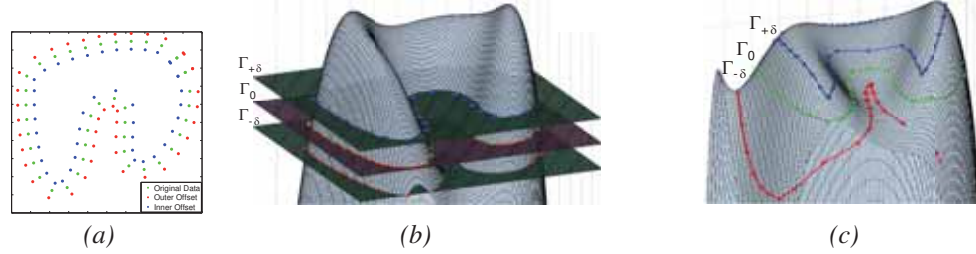


Figure 4.4: (a) Level sets: original data (Γ_0), outer offset ($\Gamma_{-\delta}$) and inner offset ($\Gamma_{+\delta}$). (b) A 3D illustration of the original 3L algorithm. (c) The best fitting polynomial obtained by [2], showing that the values in the inner and outer sets should be relaxed.

accurate result. A strategy for relaxing the additional constraints is proposed to decrease the fitting error while maintaining the structural shape of the object. The proposed approach consists of two stages as detailed below.

4.2.1 Relaxing additional constraints

As mentioned earlier, the 3L algorithm tries to encode the geometric information of the data by adding two supplementary sets supporting the original one. In the original work [11], it is suggested to have an equal value for the whole inner and outer offsets. However, as shown in Fig. 4.4(c), the perfect fitting result could be only reached by means of different values for each point in the inner and outer data sets. In other words, the right hand values in (5) (constant vectors $\pm c$) should be relaxed in order to obtain the most accurate fitting result. In the current work a novel idea to adjust these values for offsets, based on the position of the point and the approximated IP at the current iteration, is proposed.

Lets $f(\mathbf{x})$ be the IP at the current iteration; $p_i = (x_i, y_i)$ a given data point³; s_i and t_i its inner and outer offset respectively. As mentioned above these two points are obtained along the unit normal $\mathbf{n}_i = (n_i^x, n_i^y)$ from the local PCA based approximation. Moving on the surface from s_i to t_i can be parameterized as follow:

$$g(t) = f(x_i + n_i^x t, y_i + n_i^y t) \quad (4.6)$$

where f shows the value of the fitted IP (see (2.2)). The expected value for $g(0) = f(x_i, y_i)$ is zero, but its value in s_i and t_i must be estimated.

Based on the above definition, $g(\delta)$ and $g(-\delta)$ show the value of the IP achieved in s_i and t_i respectively. Considering the function g at these two points a fair proportion for next iteration could be obtained, instead of using the fixed values $\pm c$ in (4.1). For this purpose a first order Taylor approximation, around $t = 0$, of (4.6) could be computed as:

$$g(\pm\delta) \approx [n_i^x f_x(x_i, y_i) + n_i^y f_y(x_i, y_i)]\delta. \quad (4.7)$$

³Without loss of generality, and only to make clear its understanding, this discussion is presented for the 2D case, but it could be extended to the 3D case.

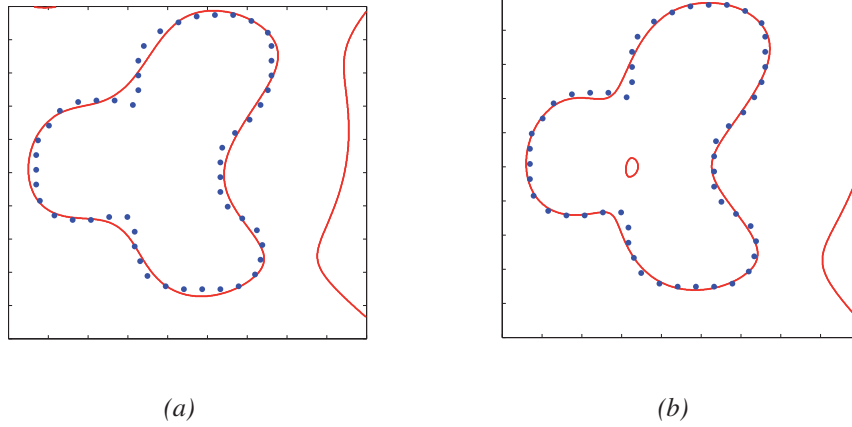


Figure 4.5: (a) Coarse approximation used as an initialization of the proposed approach ($\delta = 10$ percent of the object size, AFE= 0.1989). (b) Fitting result obtained with the proposed approach (four iterations, AFE=0.1288).

However, this value could be approximated again by considering an approximation of the normal vectors by means of the implicit function (i.e., $n_i \approx \nabla f(p_i) = (f_x(x_i, y_i), f_y(x_i, y_i))^T$):

$$g(\pm\delta) \approx \pm \|\nabla f(p_i)\|^2 \delta \quad (4.8)$$

so the next expected value for the given point should be $[0, g(\delta), g(-\delta)]$ for the original set and the inner and outer one respectively. This process is applied for every given point and then vector \mathbf{b} in (4.1) updated. Then, the least squares method (4.5) is used for computing the new IP coefficients. Note that this least squares solution is obtained just after a matrix multiplication, since the pseudoinverse matrix in (4.5) is computed only once, at the beginning. The whole procedure is iterated till convergence is reached as explained in the next section.

4.2.2 Convergence criteria

Stopping the above iteration represents a key point. On the one hand it should be something easier to compute; on the other hand it should be robust enough to be used with different geometries.

In the proposed method, a coarse fitting from the 3L algorithm is chosen as an initialization, and in each iteration the total angle between the gradient vector at each data and its approximated unit normal n_i , from local PCA (see Section 2.1.2), is measured. It should be mentioned that the approximated normal is already calculated when computing the two additional level sets. So the only required computation is regarding to the angle estimation:

$$\theta_i = \cos^{-1} \left(\frac{n_i \cdot \nabla f(p_i)}{\|\nabla f(p_i)\|} \right) \quad (4.9)$$

additionally, since $\cos^{-1}|_{[0,1] \rightarrow [0,\pi/2]}$ is monotonic, just the absolute value of the inner expression, without calculating the cosine inverse, is considered. Therefore the criterion used for measuring the goodness of the current fitting result is:

$$\xi(\mathbf{a}) = \sum_{i=1}^N 1 - \left| \frac{n_i \cdot \nabla f(p_i)}{\|\nabla f(p_i)\|} \right| \quad (4.10)$$

where N is the number of points in the original data set. The process iterates while (4.10) decreases.

Finally, it should be mentioned that in spite of the iterative nature of the proposed approach, it is more related to algebraic approaches than to the Euclidean based ones, where at every iteration the shortest distance between every single point and the current fitted IP should be computed [30]. Figure 4.5(a) shows the coarse fifth degree fitting from the 3L algorithm used as an initialization of the proposed approach; the final fitting result is obtained after four iterations (Fig. 4.5(b)).

4.3 3L-IBS Fitting Method

In this section the main contributions of this article are presented. First of all, the 3L algorithm is adopted for IBS fitting, which is more flexible for shape description. We use a simple definition of IBS based on the blending functions. All the computations in this article are based on this definition, resulting in a method easy to implement. In addition to the 3L-IBS algorithm, different extensions are also presented including global shape regularization and constraint imposing. Moreover, the proposed technique is modified to be computationally efficient. Without loss of generality, and only to make clear its understanding, the whole discussion in this section is presented for the 2D case, but it could be easily extended to the 3D case.

4.3.1 3L-IBS

This subsection we explains how the 3L algorithm can be extended for IBSs. For the sake of simplicity in implementation, the IBS definition is based on a couple of *blending functions*, which are small assembling patches. These functions blend together in different scales and transformations to make the B-spline basis functions. This section shows how to construct the monomial matrix for the 3L algorithm.

Since a square control lattice is considered, both sets of basis functions behave similarly. Having considered a row-by-row order, we can represent the control lattice and the basis functions in a vector form:

$$f(\mathbf{x}) = \mathbf{c}^T \mathbf{m}(\mathbf{x}) = \mathbf{m}(\mathbf{x})^T \mathbf{c} \quad (4.11)$$

where \mathbf{c} and $\mathbf{m}(\mathbf{x})$ are the vector form of the control values $[c_{i,j}]$ and the monomials $[B_i(x)B_j(y)]$; as in (2.4) $B_i(x)$ and $B_j(y)$ are the spline basis functions:

$$c_k = c_{i,j}, \quad m_k(x, y) = B_i(x)B_j(y). \quad (4.12)$$

Note that the indices of both vectors (i.e., \mathbf{c} and \mathbf{m}) must refer to the same element. For instance, in our case the k th element of these vectors should preserve the following relation with the indices:

$$i = \lfloor k/N \rfloor + 1, \quad j = k - \lfloor k/N \rfloor N. \quad (4.13)$$

The way in which the value of basis functions is computed is based on the blending functions that are defined on $[0, 1]$. Without loss of generality, through this paper a cubic B-spline formulation is used made out of the following patches:

$$\begin{aligned} b_0(u) &= (1-u)^3/6, & b_1(u) &= (3u^3 - 6u^2 + 4)/6, \\ b_2(u) &= (-3u^3 + 3u^2 + 3u + 1)/6, & b_3(u) &= u^3/6. \end{aligned} \quad (4.14)$$

Using these cubic patches leads to a C^2 continuity. Figure 4.6 shows how a B-spline is defined out of these blending functions. In order to have N basis functions in the interval $[0, 1]$ the step of the *knot*⁴ sequence is chosen as $\Delta = 1/(N - 3)$ (see Fig. 4.6(top)).

The definition of B-spline presented in (2.4) can be computed directly based on the blending functions:

$$f(\mathbf{x}) = f(x, y) = \sum_{r=0}^3 \sum_{s=0}^3 c_{i+r, j+s} b_r(u) \cdot b_s(v) \quad (4.15)$$

where the indices start from:

$$i = \lfloor x/\Delta \rfloor + 1, \quad j = \lfloor y/\Delta \rfloor + 1 \quad (4.16)$$

and the given coordinates in XY will be mapped in UV as:

$$\begin{aligned} u &= x/\Delta - \lfloor x/\Delta \rfloor, \\ v &= y/\Delta - \lfloor y/\Delta \rfloor. \end{aligned} \quad (4.17)$$

This definition provides us with the computational efficiency useful for calculating the monomial matrix and the regularization presented in next section. In order to adapt the IBS solution space in the 3L formulation, the monomial matrix M_{3L} must be constructed for the IBS case. Actually, as in (4.1), it contains the monomial vectors computed in the original data set, the inner and outer offsets. The monomial vector for a given point in M_{3L} is computed through the definition in (4.15). Note that the value $b_r(u)b_s(v)$ corresponds to the monomial of $B_{i+r}(x)B_{j+s}(y)$ and will be used to fill in entry k according to (4.13). Once the M_{3L} is constructed, the optimal control lattice can be directly computed in the least squares form:

$$\mathbf{c} = \mathbf{M}_{3L}^\dagger \mathbf{b} = (\mathbf{M}_{3L}^T \mathbf{M}_{3L})^{-1} \mathbf{M}_{3L}^T \mathbf{b}. \quad (4.18)$$

4.3.2 Regularization

The matrix $(\mathbf{M}_{3L}^T \mathbf{M}_{3L})^{-1}$ in (4.18) is a highly sparse matrix, which could be singular leading to more than one optimal solution. Actually, some of the control parameters do not have any

⁴knot: position where two blending functions join.

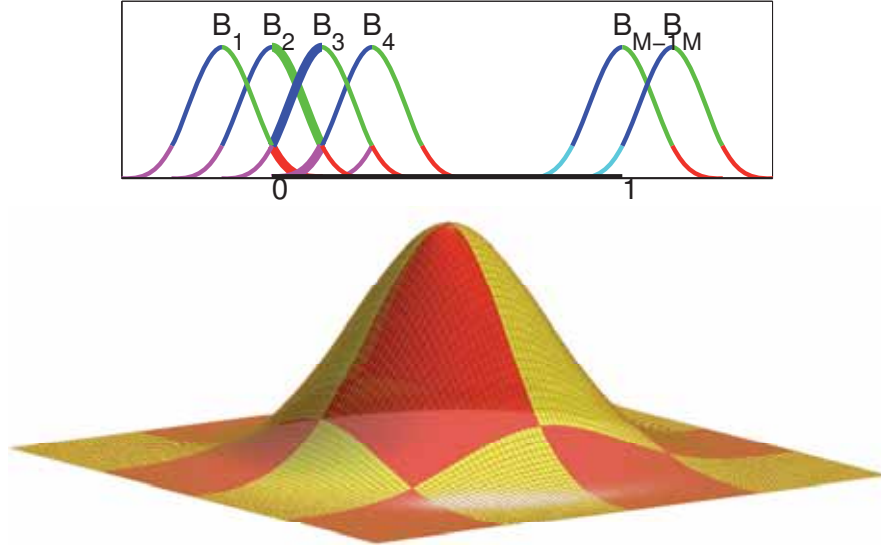


Figure 4.6: Cubic B-spline basis functions: (top) 2D basis functions made out of blending functions; (bottom) a tensor product 3D basis functions defined on the unit square.

contribution during the least squares optimization. Hence, they are not considered during the minimization, which leads to a subspace of solutions.

In order to avoid such a kind of singularity problems, Ridge Regression (RR) could be used. RR is a widely used method for regularizing ill-posed problems [110]. In this technique the monomial matrix is added by a diagonal matrix in order to obtain a nonsingular one:

$$\mathbf{c} = (\mathbf{M}_{3L}^T \mathbf{M}_{3L} + \lambda \cdot \text{diag})^{-1} \mathbf{M}_{3L}^T \mathbf{b}. \quad (4.19)$$

The diagonal matrix, diag , could be either an identity matrix or the diagonal matrix of $\mathbf{M}_{3L}^T \mathbf{M}_{3L}$ or a combination of both of them.

In the current work a global tension term is considered to regularize the control parameters [68]. This term is computed by measuring the curvature of f over the whole domain:

$$T(\mathbf{c}) = \iint_{XY} f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2 dx dy \quad (4.20)$$

which will be added to the objective function:

$$E(\mathbf{c}) = \|\mathbf{M}_{3L} \mathbf{c} - \mathbf{b}\|^2 + \mu T(\mathbf{c}) \quad (4.21)$$

where μ controls the regularization smoothness. This term is the same one used as a bending energy in Thin Plate Spline (TPS) for deformations. Figure 4.7 illustrates the effect of regularization term on the rigidity of the implicit function. In other words, a high regularization

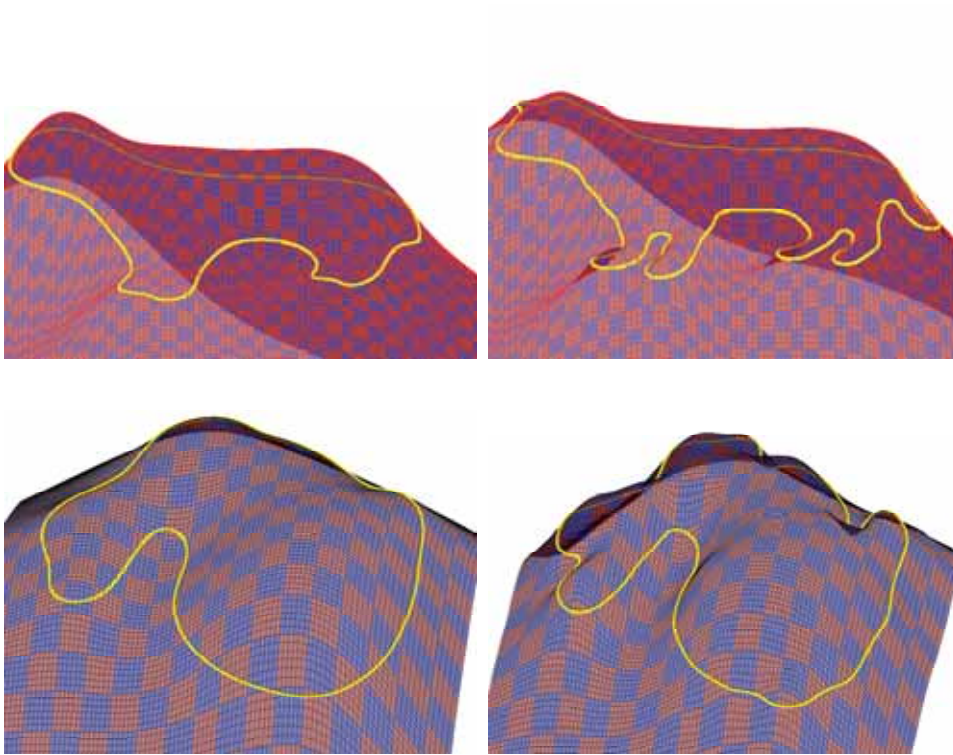


Figure 4.7: Illustrations of the fitted IBS with different regularization parameters (μ in (4.22)); resulting zero sets are highlighted.

parameter results in a more rigid function (Fig. 4.7(*left*)), while a low one leads to a more relaxed implicit function (Fig. 4.7(*right*)).

It must be emphasized that this term can be analytically computed for the given control parameters. This is due to the linear definition of IBS, and the linearity of the integral operator. Furthermore, since it is a quadratic term, its derivative will be still linear, and the final solution could be solved through a linear system of equations [68]. In Section 4.3.4 more details are presented to simplify this term in the form of $T(\mathbf{c}) = \mathbf{c}^T \mathbf{H} \mathbf{c}$, where matrix \mathbf{H} is a $N^2 \times N^2$ symmetric matrix including the integral of basis functions' derivatives. The regularization matrix \mathbf{H} is a sparse matrix as illustrated in Fig. 4.10. In section 4.3.4 more detail about the computation of this matrix is presented.

The regularization matrix \mathbf{H} is a sparse matrix, and its entries are repetitive since a regular knot is used for IBS. Therefore it can be calculated very fast before running the algorithm. Once this matrix is computed, the optimal IBS coefficient vector can be found linearly through the least squares solution:

$$\mathbf{c} = (\mathbf{M}_{3L}^T \mathbf{M}_{3L} + \mu \mathbf{H})^{-1} \mathbf{M}_{3L}^T \mathbf{b}. \quad (4.22)$$

This formula is easily derived by considering the first derivative of the energy term defined in (4.21):

$$\partial F / \partial \mathbf{c} = 2(\mathbf{M}_{3L}^T \mathbf{M}_{3L}) \mathbf{c} - 2\mathbf{M}_{3L}^T \mathbf{b} + 2\mu \mathbf{H} \mathbf{c}. \quad (4.23)$$

4.3.3 Additional constraints

As already mentioned, in algebraic fitting approaches the fitting function is sought in order to satisfy a set of constraints. The more geometric meaningful constraints are the more accurate result will be obtained, and this is the motivation of the 3L algorithm and the Gradient-one method. In addition to the data sets used by these methods, the user may like to force the shape to pass through some additional specific points (referred to as *positional constraint*) or to obtain a determined orientation in some specific points (referred to as *orientational constraint*).

Lets us to consider the problem as minimizing $E(\mathbf{c})$ in (4.21), such that the zero set of the optimal f passes through some specific points: $\{\mathbf{q}_1, \dots, \mathbf{q}_k\}$, defining the positional constraints. In other words $g_i = f(\mathbf{q}_i) = \mathbf{w}_i^T \mathbf{c} = 0$, where $\mathbf{w}_i = \mathbf{m}(\mathbf{q}_i)$ is the monomial vector calculated in the point \mathbf{q}_i . These monomial vectors $\mathbf{w}_1^T, \dots, \mathbf{w}_k^T$ construct the monomial matrix \mathbf{W} defined for the positional constraint. Hence these constraints can be reformulated as $\mathbf{W} \mathbf{c} = \mathbf{0}$. So the constrained problem can be formulated as:

$$\begin{aligned} \min_{\mathbf{c}} E(\mathbf{c}) &= \|\mathbf{M}_{3L} \mathbf{c} - \mathbf{b}\|^2 + \mu \mathbf{c}^T \mathbf{H} \mathbf{c} \\ \text{s.t.} & \\ \mathbf{W} \mathbf{c} &= \mathbf{0}. \end{aligned} \quad (4.24)$$

In order to solve this constrained minimization problem, the Lagrange multiplier method can be used [11]. The Lagrange function defined for this problem can be easily shown in the

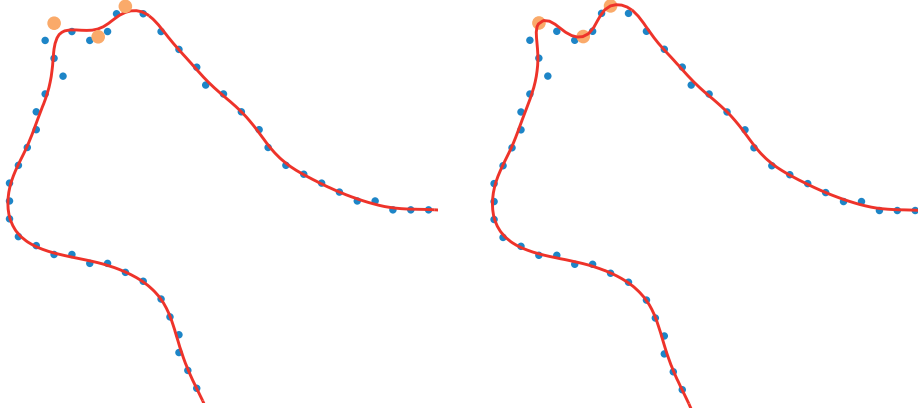


Figure 4.8: Positional constraints can be applied for the proposed 3L-IBS method. (*left*) the original point and fitted IBS; (*right*) the constrained IBS passing through the three user-defined points.

matrix form as follows:

$$\Lambda(\mathbf{c}, \lambda) = E(\mathbf{c}) - \sum_{j=1}^k \lambda_j \mathbf{w}_j^T \mathbf{c} \quad (4.25)$$

$$= E(\mathbf{c}) - \boldsymbol{\lambda}^T \mathbf{W} \mathbf{c}. \quad (4.26)$$

Since all the terms are either linear or quadratic, it is easy to compute the derivatives. Based on the Lagrange method, derivative of the above mentioned function, must be set to zero. Derivative with respect to λ leads to the positional constraints $\mathbf{W} \mathbf{c} = \mathbf{0}$, while with respect to \mathbf{c} will result in:

$$2(\mathbf{M}_{3L}^T \mathbf{M}_{3L}) \mathbf{c} - 2\mathbf{M}_{3L}^T \mathbf{b} + 2\mu \mathbf{H} \mathbf{c} = \mathbf{W}^T \boldsymbol{\lambda}. \quad (4.27)$$

As a result the optimal control lattice \mathbf{c} for the constrained 3L-IBS method can be easily computed as:

$$\mathbf{c} = \mathbf{A}[\mathbf{M}_{3L}^T \mathbf{b} + 1/2 \mathbf{W}^T \boldsymbol{\lambda}] \quad (4.28)$$

where $\mathbf{A} = (\mathbf{M}_{3L}^T \mathbf{M}_{3L} + \mu \mathbf{H})^{-1}$ and the Lagrange multiplier $\boldsymbol{\lambda}$ is still unknown. It can be easily found by multiplying (4.28) by \mathbf{W} from left; then, since $\mathbf{W} \mathbf{c} = \mathbf{0}$, due to the positional constraints we have:

$$\boldsymbol{\lambda} = -2(\mathbf{W} \mathbf{A} \mathbf{W}^T)^{-1} \mathbf{W} \mathbf{A} \mathbf{M}_{3L}^T \mathbf{b}. \quad (4.29)$$

Figure 4.8(*left*) shows an illustration where the fitting curve does not interpolate some data points. This is due to the low accuracy of the least squares defined in the 3L-IBS formulation in (4.22). This problem can be easily solved thanks to the constrained 3L-IBS technique proposed in (4.28). Figure 4.8(*right*) shows three positional constrained defined

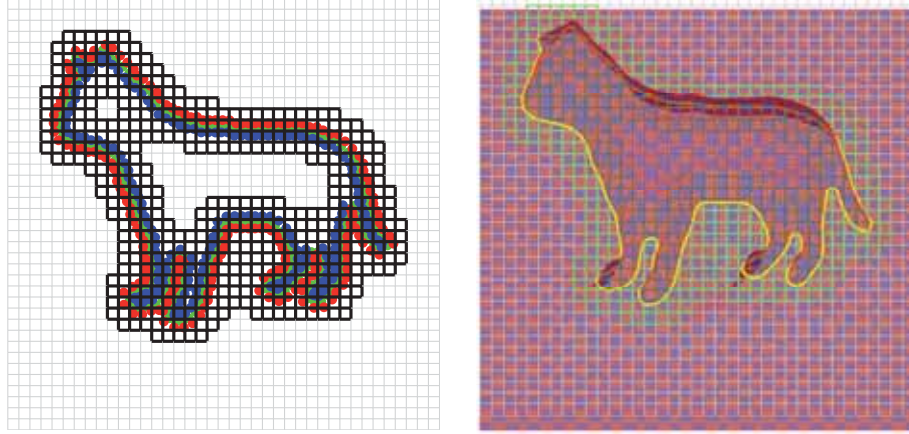


Figure 4.9: Computationally efficient: (*left*) the active control points are highlighted; (*right*) the efficient algorithm considers only a narrow strips around the active control points.

by the user. As illustrated the fitted IBS perfectly interpolates these three user-defined points. It must be mentioned that the constrained 3L-IBS formulation can be easily extended for the orientational constraints as well.

4.3.4 Efficient 3L-IBS Algorithm

The 3L-IBS algorithm presented above is a precise fitting method that can describe an object through a set of control values. The 3L-IBS algorithm is fast since it is an algebraic formulation, and it is flexible since it uses IBS to represent 2D/3D objects. For instance the 3D object in Fig. 2.1(*right*) uses an IBS with a control lattice of $20 \times 20 \times 20$. Hence, to show such an object in a moderate resolution, 8000 control values must be optimized. Even though the proposed 3L-IBS algorithm uses a fast algebraic framework, solving a system of equations with 8000 unknowns will be really expensive. Moreover, if the point cloud contains only $1k$ points in the 3D space, the block matrix \mathbf{M}_{3L} in (4.1) of size 3000 by 8000 and the regularization matrix \mathbf{H} of size 8000 by 8000 must be constructed. Hence a memory of 8.8×10^7 cells must be allocated to keep the values of floating point format.

The problem regarding the memory usage can be handled, since the matrices \mathbf{M}_{3L} and \mathbf{H} are highly sparse. However the problem of the high dimension unknown space remains unsolved. In this work we consider a lower dimensional space to find the optimal IBS. We exploit the fact that only a part of the control lattice is engaged with the fitting algorithm, and the rest of them follows the engaged ones through the regularization term. Figure 4.9 illustrates how the proposed 3L-IBS is efficient. In Fig. 4.9(*left*) a 40×40 IBS is used to fit a 2D silhouette of cat [127]. In our efficient case around 680 control points are considered as active ones, hence the optimization part will be much faster (around 8 times faster). The efficiency of the proposed algorithm will be highlighted more for the 3D case, where out



Figure 4.10: The images $\log(H)$ of the regularization matrix H for a cubic control lattice of size $N = 15$; (*left*) when all the control points are considered (3375×3375); (*right*) when the proposed efficient algorithm is used (889×889)

method is much faster and more practical than normal 3L-IBS. In this section we explain the detail of our efficient fitting method.

Figure 4.10 illustrates the regularization matrix \mathbf{H} of the cubic control lattice of size $N = 15$ for the *Bunny* data set. Indeed, the logarithm of entries is illustrated to depict the non-zero values. Figure 4.10(*left*) shows the $\log(\mathbf{H})$ when the whole control lattice is considered in the 3L-IBS algorithm. Figure 4.10(*right*) instead shows a smaller regularization matrix, which is only calculated for the active control points. The first regularization matrix is more than 3 times bigger than the second matrix, though both are sparse.

The regularization term in (4.21) is an important part of the IBS fitting, which controls the solution rigidity in a high dimensional solution space. This term is quadratic as mentioned in section 4.3.2. In this section we show how the regularization matrix \mathbf{H} can be computed for the efficient 3L-IBS. First of all assume that $\hat{\mathbf{c}}$ is the active control vector and $\hat{\mathbf{m}}$ is the corresponding monomial vector, which contains the tensor products of B-spline functions. It must be indicated that the order of data in both vectors should be the same. Then, for the points in active region: $\mathbf{c}^T \mathbf{m}(\mathbf{x}) = \hat{\mathbf{c}}^T \hat{\mathbf{m}}(\mathbf{x})$.

In the efficient 3L-IBS method we propose to consider the regularization term on the active region. Then, using the vector form defined in (4.11) we have $f_{xx}(\mathbf{x}) = \mathbf{c}^T \hat{\mathbf{m}}_{xx}(\mathbf{x})$ due to the linear definition of B-spline. Therefore:

$$\begin{aligned} T(\hat{\mathbf{c}}) &= \hat{\mathbf{c}}^T \mathbf{H} \hat{\mathbf{c}} \\ &= \hat{\mathbf{c}}^T \left[\iint_{XY} \hat{\mathbf{m}}_{xx} \hat{\mathbf{m}}_{xx}^T + 2\hat{\mathbf{m}}_{xy} \hat{\mathbf{m}}_{xy}^T + \hat{\mathbf{m}}_{yy} \hat{\mathbf{m}}_{yy}^T dx \cdot dy \right] \hat{\mathbf{c}} \end{aligned} \quad (4.30)$$

where \mathbf{H} is a $N^2 \times N^2$ symmetric matrix including the integral of basis functions' derivatives.

In this section we show how the regularization matrix \mathbf{H} can be efficiently constructed based on the monomial vectors. For any $k, l \in \{1, 2, \dots, N^2\}$, the couples (i_k, j_k) and (i_l, j_l) can be found from the table that is constructed when the active control points are detected and sorted (when the whole control lattice is considered these indices can be set based on (4.13)). Hence, based on the definition, the k th entry of \mathbf{m}_{xx} is equal to $(B''_{i_k}(x).B_{j_k}(y))$ (similarly for the l th entry), thus we have:

$$\begin{aligned} & [\iint_{XY} \mathbf{m}_{xx} \mathbf{m}_{xx}^T dx dy]_{(k,l)} \\ &= \iint_{XY} (\mathbf{m}_{xx})_k \cdot (\mathbf{m}_{xx})_l dx dy \\ &= \iint_{XY} (B''_{i_k}(x)B_{j_k}(y)) \cdot (B''_{i_l}(x)B_{j_l}(y)) dx dy \\ &= \int_X B''_{i_k}(x)B''_{i_l}(x) dx \cdot \int_Y B_{j_k}(y)B_{j_l}(y) dy \end{aligned}$$

therefore, the double integral on the whole XY domain is converted to two single integrals on X and Y domains; by applying a similar analysis on the other terms in (4.30) the following compact terms are obtained:

$$\Pi_{m,n}^{(k)} = \int_X B_m^{(k)}(x) \cdot B_n^{(k)}(x) dx \quad (4.31)$$

where $B^{(k)}$ refers to k -th derivative of B and $m, n \in \{0, 1, \dots, N\}$. These matrices are $N \times N$, and we can compute the entries of \mathbf{H} through them:

$$H_{k,l} = \Pi_{i_k, i_l}^{(2)} \Pi_{j_k, j_l}^{(0)} + 2\Pi_{i_k, i_l}^{(1)} \Pi_{j_k, j_l}^{(1)} + \Pi_{i_k, i_l}^{(0)} \Pi_{j_k, j_l}^{(2)}. \quad (4.32)$$

Thus, in order to compute the $N^2 \times N^2$ matrix \mathbf{H} , we must compute three $N \times N$ matrices $\mathbf{\Pi}^{(0)}$, $\mathbf{\Pi}^{(1)}$ and $\mathbf{\Pi}^{(2)}$. Now the objective is to compute the entries of these matrices based on the blending function. For this purpose we consider three 4×4 matrices considering different combinations of the blending functions:

$$\begin{aligned} \pi_{r,s}^{(k)} &= \int_0^1 b_r^{(k)}(u) \cdot b_s^{(k)}(u) du \\ \pi_{r,s}^{(1)} &= \int_0^1 b_r'(u) \cdot b_s'(u) du \\ \pi_{r,s}^{(2)} &= \int_0^1 b_r''(u) \cdot b_s''(u) du \end{aligned} \quad (4.33)$$

where $b^{(k)}$ refers to k -th derivative of b and $r, s \in \{0, 1, 2, 3\}$. Then, we start moving on the region, and accumulating those values of $\pi^{(0)}$, $\pi^{(1)}$ or $\pi^{(2)}$ that contribute in that subregion. Note that we use a special property of the B-spline basis functions: when they are considered in a subregion (with the length of Δ) they are shrunk versions of one of the cubic patches.

Once the regularization matrix \mathbf{H} is computed, the optimal IBS coefficient vector can be found linearly through the least squares solution in (4.22). Figure 4.11 depicts how the regularization term benefits the smoothness of the IBS zero set. In all these cases the efficient version of the regularization matrix is used, which is smaller than the normal cases.

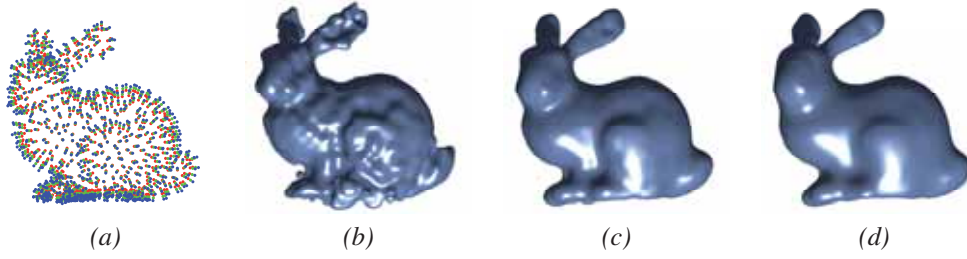


Figure 4.11: The efficient 3L-IBS method to reconstruct the *Bunny* data set containing only 817 points; (a) the offset of *Bunny* ($\delta = 1\%$); (b)-(d) different settings for the regularization parameter $\lambda = 1, 10^2, 10^3$ respectively.

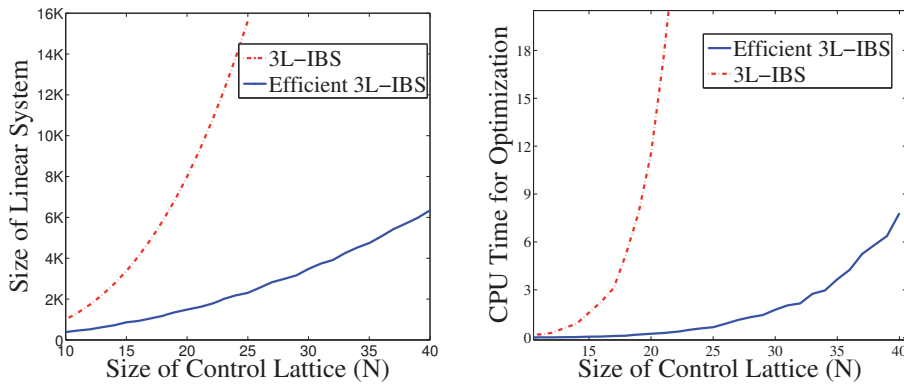


Figure 4.12: Complexities of the 3L-IBS and its efficient version for *Bunny* data set containing 10k points; (left) the size of the optimization problems; (right) the CPU-time spent for finding the optimal control lattice.

Automatic setting: As mentioned in the previous section the regularization term controls the rigidity of the implicit function. A high regularization term results in a more stable and rigid result but less accurate. The amount of rigidity can be automatically adjusted. In our implementation we start from high regularization term (e.g., $\lambda = 10^4$ for the 3D case) and then we decrease it slowly while some condition is still valid.

In this paper we check out the compatibility of the gradient vectors of the implicit function with the normal vector of point set. While the gradient vectors are close to normals we continue, but once it has started disturbing we stop. The approximated unit normal n_i , is already calculated through local PCA or the triangular mesh. It should be mentioned that the approximated normal is already calculated when computing the two additional level sets. Hence the only required computation is regarding to the angle estimation similar to (3.13).

4.4 Experimental Results

Experimental results are presented to compare the for relaxing 3L-IP and 3L-IBS fitting methods with the state of art. The comparisons are applied for both 2D and 3D data sets, which are either public ([52], [127]) or obtained through a 3D scanner (i.e. Konica Minolta Vivid 9i). The comparisons are based on the CPU time the geometric accuracy. The latter is computed through some approximation of the geometric distance from the point set to their corresponding foot-points on the curve/surface. This distance is denoted by AFE (accumulated fitting error) through the whole section. More detail about the foot-point computation can be found in Section 2.1.2.

4.4.1 Relaxed 3L-IP Fitting Results

2D and 3D data sets. Several data sets have been fitted with the proposed approach and compared with the best results obtained with the 3L algorithm. Furthermore, the results obtained with [2] are provided. Figure 4.13 shows 2D contours fitted by sixth degree IPs using the 3L algorithm (Fig. 4.13(a)), the proposed approach (Fig. 4.13(b)) and a non-linear orthogonal distance based approach [2] (Fig. 4.13(c)). In all the cases the accuracy obtained with the proposed approach improves considerable the one obtained with the 3L algorithm (see Table 3.2); moreover, it is comparable to the results obtained when the non-linear approach is used. Although out of the scope of the current work, it should be mentioned that the proposed approach is about ten times faster than [2].

Figure 4.14(*left*) depicts the fitting results obtained with the three approaches when 3D data points are considered; note that although similar AFEs are obtained the geometry computed with proposed approach is more similar to the ground truth than the one obtained with the 3L. Finally, Fig. 4.14(*right*) presents results obtained after fitting a set of real data points corresponding to a partial view of a sphere. In this case not only qualitative better results are obtained with the proposed approach but also quantitatively.

Zero set. In order to force that the fitted IP pass through particular data points, known as positional control points, [11] proposes to incorporate additional linear constraints. Figure

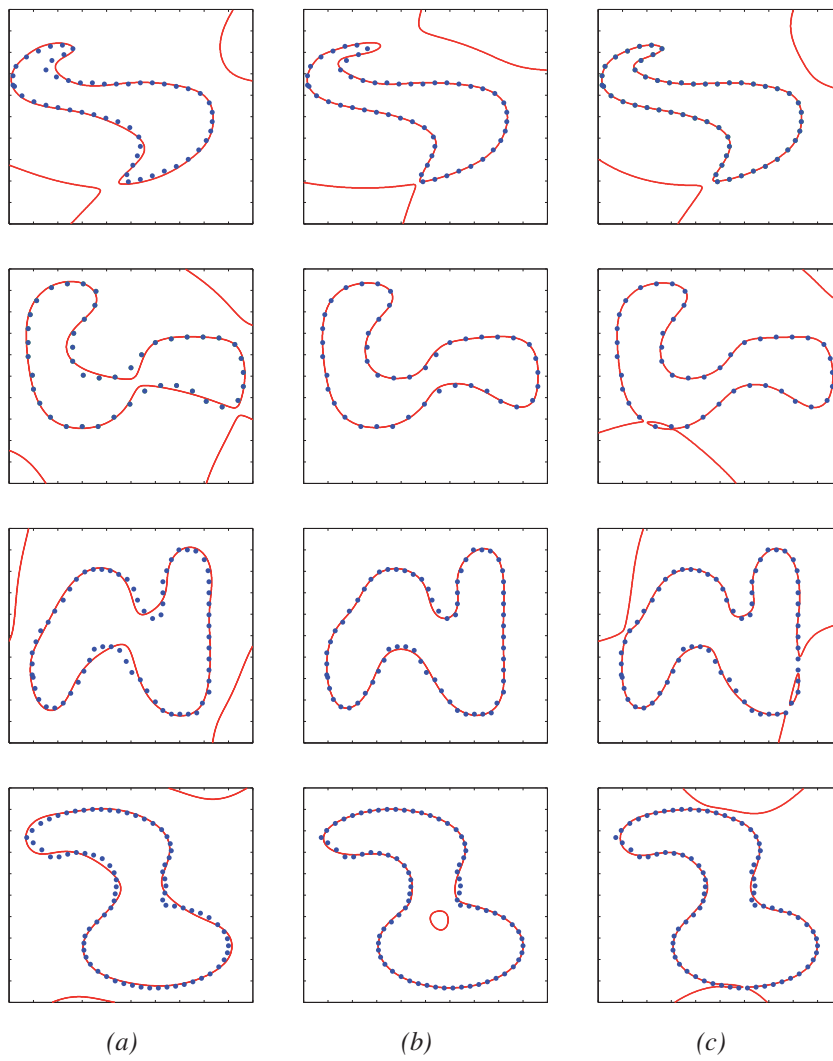


Figure 4.13: (a) Results from the 3L algorithm. (b) Results from the proposed approach. (c) Results from [2] used as ground truths.

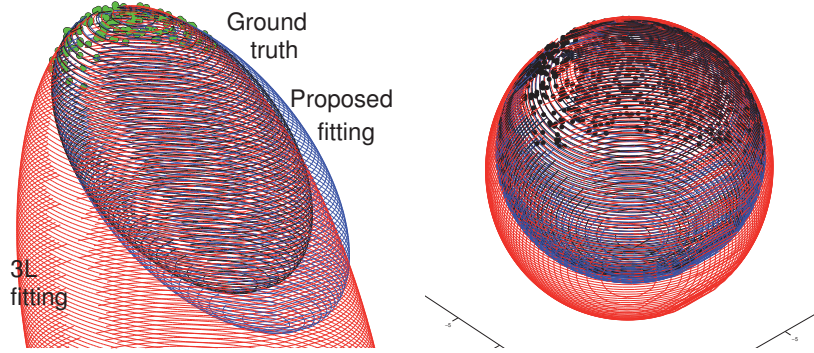


Figure 4.14: (*left*) Non-uniform synthetic noisy data from an ellipsoid (127 points), fitted with the 3L algorithm ($\delta = 10\%$, AFE=4.3376), the proposed approach (AFE=3.1234) and [2] (AFE=3.2441). (*right*) 3D data points obtained with a structured light camera from a sphere; outer mesh correspond to the result obtained with the 3L algorithm (AFE=242.6154) while inner spheres show the results from the proposed approach (AFE=85.1591) and the ground truth (AFE=85.4815).

Table 4.1: Accumulated fitting errors to compare the results obtained by different approaches (3L: the 3L algorithm [11]; GO: the Gradient One [133]; MM: Min-Max [45]; MV: Min-Var [45]; PA: Proposed Approach; GA: Geometric Approach [2]).

	3L	GO	MM	MV	PA	GA
Fig. 4.13(1)	5.70	13.24	13.76	13.75	1.59	0.84
Fig. 4.13(2)	4.82	13.96	3.91	10.48	1.65	0.96
Fig. 4.13(3)	8.36	16.50	10.96	15.82	4.11	3.75
Fig. 4.13(4)	7.98	10.04	5.91	8.04	2.74	3.95
Fig. 4.16(1)	13.70	30.13	8.76	22.71	1.22	0.55
Fig. 4.16(2)	3.40	10.33	5.47	8.80	0.73	0.84

4.15(a) shows the best fitting result obtained with the 3L algorithm. It has been achieved after trying different δ values and it corresponds to a $\delta = 4$ percent of the object size. Figure 4.15(b), (c) and (d) depict results from the 3L algorithm after incorporating additional constraints, as proposed in [11], to force that two, three and four points respectively belong to the zero set. It can be concluded that increasing the number of positional constraint does not result in a reduction of the AFE. The result of the proposed approach (see Fig. 4.5(b))(AFE=0.1288) is quite similar to the values obtained after manually tuning δ in the 3L algorithm or after adding positional control points.

Non-uniform sampling and open boundaries. Two particular challenges for fitting algorithms arise when data points are non-uniformly distributed or when they correspond to an open contour/surface. Figure 4.16 presents two illustrations obtained when the 3L algorithm, the proposed approach and [2] were used. Note that the results obtained with the proposed approach (AFE) are quite similar to the ones obtained by using [2] (see Table 4.1), but they

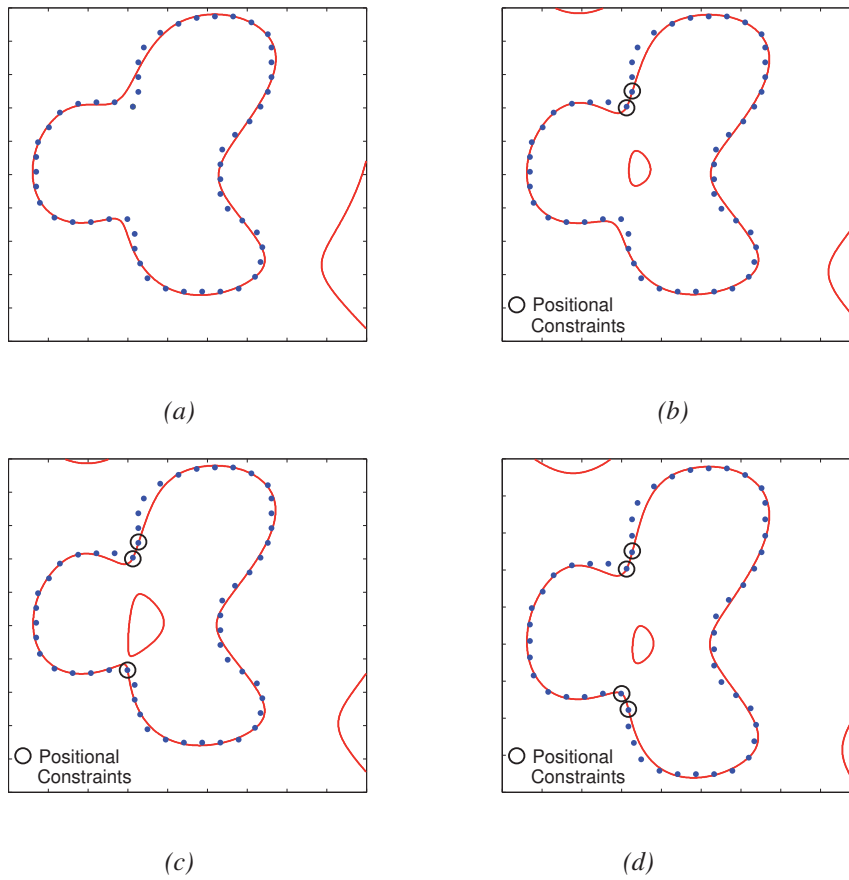


Figure 4.15: (a) Best result from the 3L algorithm obtained by setting $\delta = 4$ percent of the object size (AFE=0.1285). Results after incorporating: (b) two positional constraints (AFE=0.1179); (c) three positional constraints (AFE=0.1068); and (d) four positional constraints (AFE=0.1105).

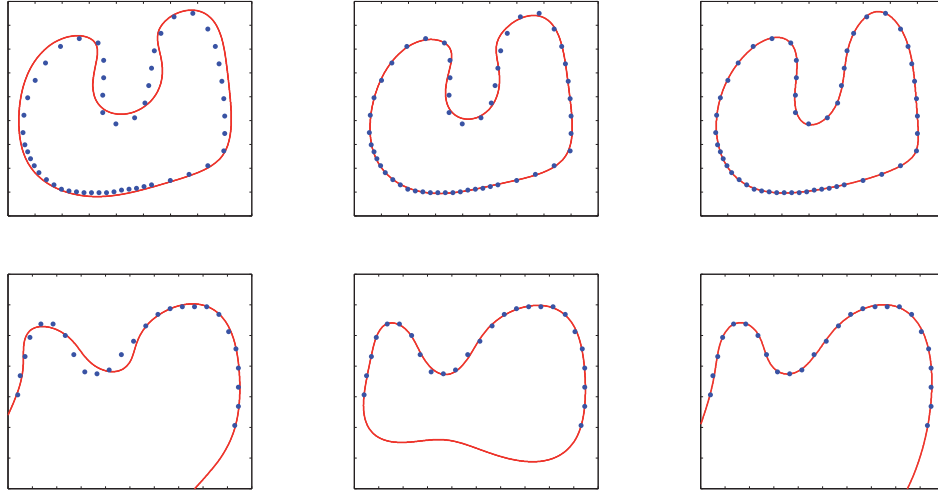


Figure 4.16: Non-uniform sampling and open boundary cases: (*left*) Results from the 3L algorithm; (*middle*) Results from the proposed approach; (*right*) Results from an orthogonal distance based approach [2], non-linear fitting.

were generated almost ten times faster. A fourth degree IP was fitted in the open contour case (Fig. 4.16(*right*)) while a fifth degree IP was considered for the non-uniform point distribution case (Fig. 4.16(*left*)).

Quantitative Comparisons. Table 4.1 shows the AFE for five different methods as well as the proposed one. It should be noticed that the last column corresponds to [2] that is obtained by solving a nonlinear optimization, and needs more computation. As mentioned above it is used as a ground truth. All other methods belong to the algebraic category, which are solved by a simple least squares method. The proposed algorithm has obtained similar results to the geometric one, while is much faster. The Min-Max and Min-Var methods [45] are similar to the Gradient-One algorithm [133], which incorporates orientation in the optimization framework. All these methods try to obtain more *stable* fitting results while neglecting the accuracy. Fig. 4.17 shows how the Gradient-One algorithm fails to describe the corners. Even though the normal directions are preserved, but the zero set is away from the original data.

4.4.2 3L-IBS Fitting Results

2D data sets. Several data sets have been fitted with the proposed approach and compared with the best results obtained with the original 3L algorithm [11], and the Gradient-one [133, 68]. Figure 4.18(1st column) illustrates six data sets with different complexities. In the second column the zero sets of the optimal IPs obtained by the original 3L algorithm are

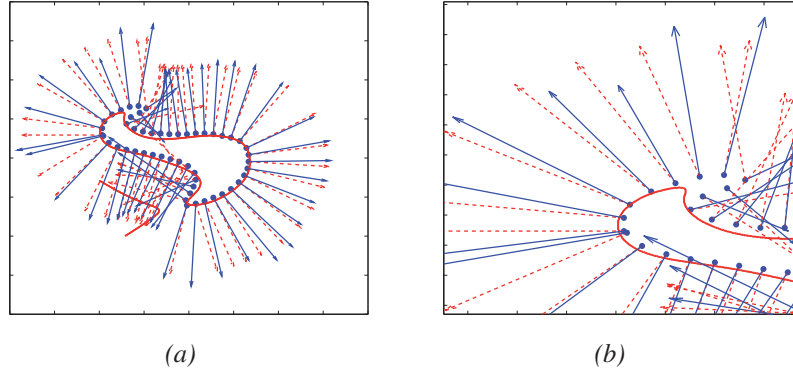


Figure 4.17: (a) The result obtained by the Gradient-One algorithm [133]. (b) An enlargement showing how the fitting algorithm ignores the positional constraint, the gradient vectors have a similar length and orientation though.

presented (the IP degrees can be understood from Table 4.2). The third and fourth columns correspond to the optimal IBS obtained by the Gradient-one and the proposed algorithm. The results obtained through IPs (2nd row) suffers from outliers and inaccuracies. This problem gets worse when the object gets more complex.

The problem of IP solution space is the global support of IP monomials. In other words, none of the IP monomials vanish in infinity. Hence all the IP coefficients will be active for every point in the space. This problem is solved when IBS is used instead to describe the solution space. The cubic IBSs meet C^2 continuity and have compact supports. As illustrated in the 3rd and 4th rows of Fig. 4.18 the optimal IBSs have smooth zero set without any outlier. Moreover, as stated in Table 4.2, the optimal IBSs achieve lower error than IPs.

Table 4.2: Size of solution spaces and accumulated fitting errors (AFEs) to compare the results obtained by different approaches (3L: 3L algorithm [11]; GO: Gradient One [133]; PA: Proposed Approach).

Figure	3L		GO		PA	
	Size	AFE	Size	AFE	Size	AFE
Fig. 4.18-(oni)	231	1.0148	400	0.9957	190	0.2788
Fig. 4.18-(dude)	351	1.0612	625	1.3362	305	0.6183
Fig. 4.18-(fish)	231	2.1222	400	0.8979	178	0.7475
Fig. 4.18-(homer)	496	1.2625	900	0.5797	303	0.2546
Fig. 4.18-(camel)	861	2.6782	1600	1.1398	603	0.7668
Fig. 4.18-(hand)	1081	6.2209	2500	0.8444	865	0.4000

The Gradient-one algorithm, like the 3L algorithm, uses the normal information that is locally estimated in each data point. In our implementation we used a Gradient-one tech-

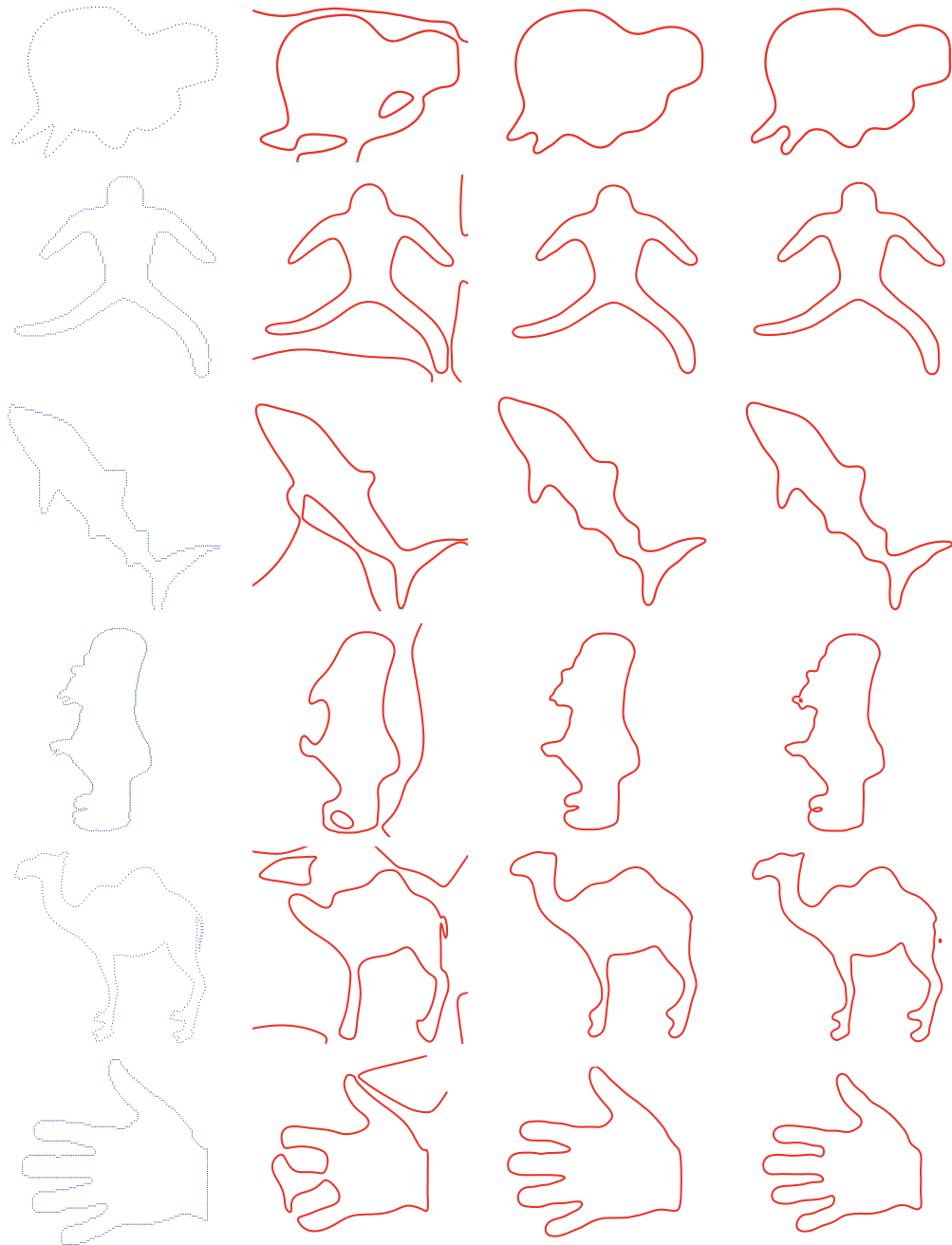


Figure 4.18: (1st col) original 2D data sets. Results obtained by: (2nd col) 3L-IP [2]; (3rd col) Gradient-one algorithm; (4th col) the proposed 3L-IBS.

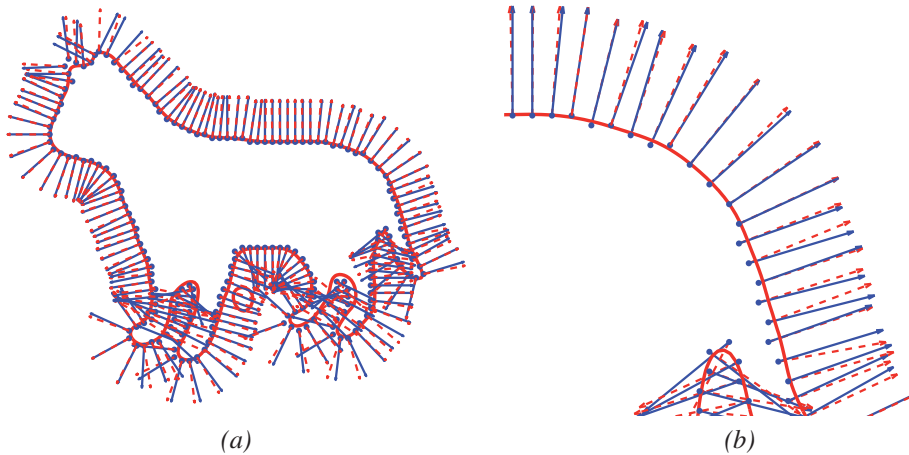


Figure 4.19: (*left*) The result obtained by the Gradient-one algorithm [133]. (*right*) An enlargement showing how the fitting algorithm ignores the positional constraint, the gradient vectors have a similar length and orientation though.

nique applied in IBS solution space. As illustrated in Fig. 4.18 the result from Gradient-one algorithm has a similar accuracy than the proposed 3L-IBS algorithm. But since our method is more based on the point set positions than their orientation, the final zero set is closer to the data set. Figure 4.19, for instance, illustrates that the final zero set does not interpolate some data points though the orientations at these points are compatible with the local normals.

In addition to high accuracy, the proposed method is much faster in comparison. As stated in Table 4.2 the size of the parameter space in the proposed algorithm is smaller than the Gradient-one. Actually, as depicted in Fig. 4.12 the parameter size increases quadratically with respect to the control lattice resolution N , while in the Gradient-one it increases in a cubic growth. Hence in our case the least squares is optimized in a smaller parameter space, and this is why our algorithm is much faster. This point is even more highlighted when it comes to the 3D case (see CPU time in Table 4.2).

The results in 3D case are presented in two different classes. First, as depicted in Fig. 4.20 different techniques are applied to reconstruct the surface from a low density point set. The data size and other information is listed in Table 4.3. For all of these examples low degree IPs (e.g, degree 10 \sim 20) and low resolution IBS ($N \approx 20$) are used. As illustrated in the second column, using IPs usually leads to lots of outlier around the data set, but when IBSs are used the results are more stable. In fact the results by IPs are less stable than the ones by IBS, and the IP coefficients corresponding to higher monomials can disturb the final result. In the IBS cases, the degree of monomials do not exceed the cubic degree. In addition to the visual superiority of IBS results, Table 4.4 shows that the accuracy in the proposed techniques is higher than others.

In the second class of experimental results in 3D, high density cloud of points are considered. Figure 4.21 shows five different objects from the public data set in [52]. The number

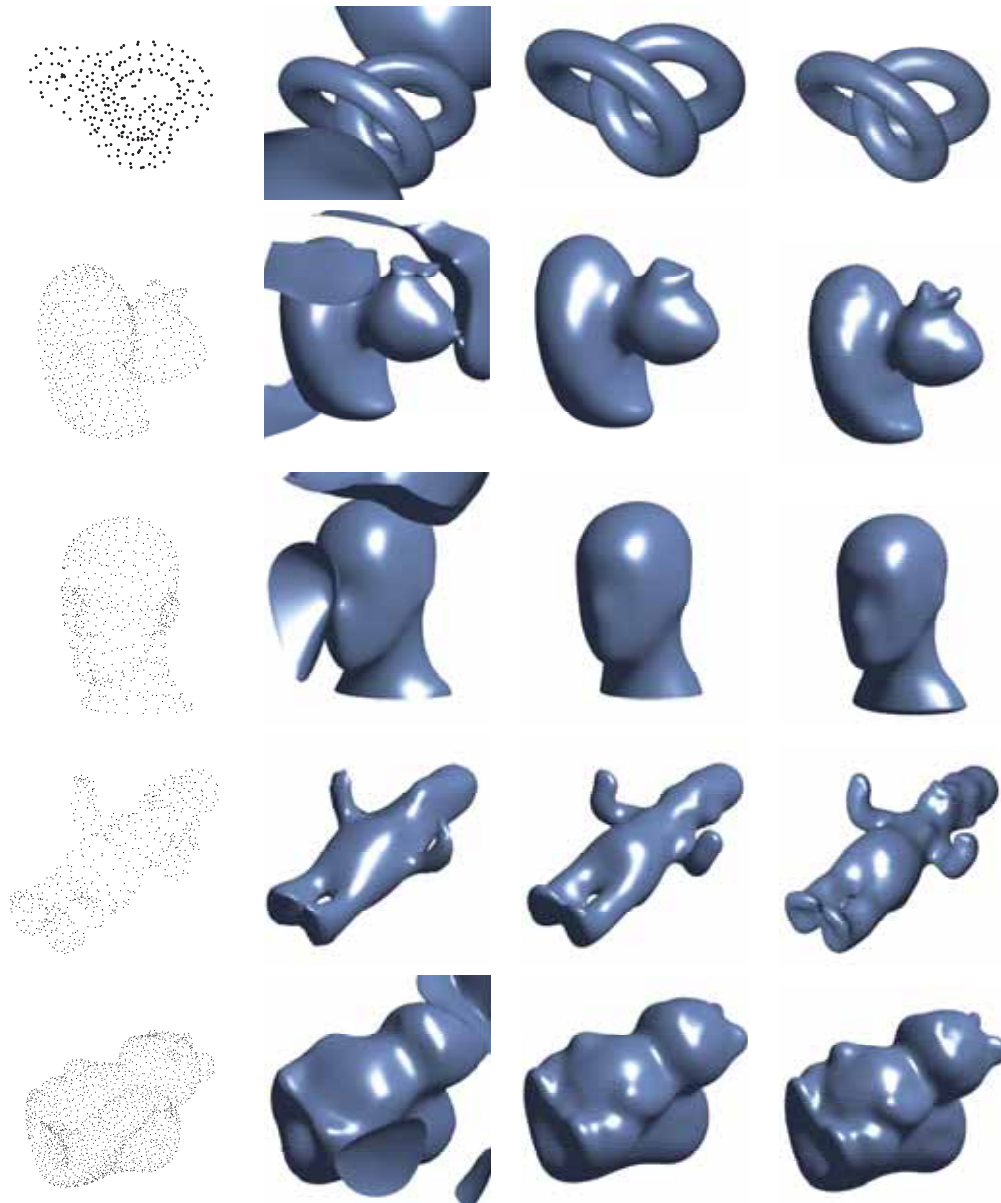


Figure 4.20: (1st col) original 3D data sets. Results obtained by: (2nd col) 3L-IP [2]; (3rd col) Gradient-one algorithm; (4th col) the proposed 3L-IBS.

Table 4.3: Low resolution 3D data sets; (POL: the size of polynomial basis; DAT: number of data points; LAT: size of control lattice; ACT: size of active control lattice).

	Name	DAT	POL	LAT	ACT
Fig. 2.1	Bunny	817	1771	8000	1461
Fig. 4.20(1)	Knot	240	816	8000	1484
Fig. 4.20(2)	Duck	745	1771	8000	151
Fig. 4.20(3)	Mannequin	639	816	3375	749
Fig. 4.20(4)	Homer	585	3654	27000	1453
Fig. 4.20(5)	Squirrel	1717	2925	27000	3253

Table 4.4: CPU times (in sec.) spent for the optimization and accumulated fitting errors (AFEs) to compare the results obtained by different approaches (3L: the 3L algorithm [11]; GO: the Gradient One [133]; PA: Proposed Approach).

Figure	3L		GO		PA	
	CPU	AFE	CPU	AFE	CPU	AFE
Fig. 2.1	1.181	5.185	3.137	3.894	0.219	3.230
Fig. 4.20(1)	0.078	0.334	2.643	0.130	0.183	0.050
Fig. 4.20(2)	0.554	2.032	2.898	1.698	0.257	1.314
Fig. 4.20(3)	0.143	3.369	0.697	3.916	0.102	2.718
Fig. 4.20(4)	2.369	3.158	34.829	2.040	0.246	1.692
Fig. 4.20(5)	2.532	7.778	29.437	3.470	0.784	2.856

Table 4.5: High resolution 3D data sets; (DAT: number of data points; LAT: size of control lattice; ACT: size of active control lattice; CPU: the CPU time (in sec.) spent for optimization).

	Name	DAT	LAT	ACT	CPU
Fig. 4.21-(1st)	<i>Kitten</i>	11039	30	2554	1.46
Fig. 4.21-(2nd)	<i>Hand</i>	16519	40	3649	2.37
Fig. 4.21-(3rd)	<i>Bimba</i>	8857	40	6013	6.64
Fig. 4.21-(4th)	<i>Armadillo</i>	8229	50	8316	16.13
Fig. 4.21-(5th)	<i>Cow</i>	10850	50	5003	4.62
Fig. 4.22	<i>Bunny</i>	10680	50	10479	13.89
Fig. 4.23(<i>top</i>)	<i>Face1</i>	12000	60	9383	3.29
Fig. 4.23(<i>bot.</i>)	<i>Face2</i>	12000	60	15391	15.14

of data points to be fitted is between $8k$ to $12k$, as presented in Table 4.5. The IBS control lattice resolution varies between $N = 30$ to $N = 50$ based on the object complexity. The second column of Fig. 4.21 illustrates how the flexibility of the proposed 3L-IBS technique allows to describe the details in the 3D objects. Moreover, it must be emphasized that the CPU-time spent for the optimization part is only few seconds (see Table 4.5). Figure 4.22 illustrates a high resolution IBS for a high density cloud of points of *Bunny*. The fitted surface is also magnified to give a better contrast between the object mesh and the fitted IBS, which has been obtained through the proposed 3L-IBS reconstruction technique.

Finally, the proposed surface reconstruction technique is applied on a cloud of 3D points of a face obtained by a 3D scanner (Konica Minolta Vivid 9i). The obtained point set is noisy, and it misses some points that are invisible to the 3D scanner. The original data set contains $168k$ points; we only consider $12k$ points randomly selected to save memory. Figure 4.23 depicts how a moderate resolution IBS ($N = 60$ with around $15k$ active control points) results in a smooth surface (with at least C^2 continuity), which compensates the missing parts.

4.5 Conclusion

This chapter tackled the algebraic fitting methods to reconstruct 2D curves/3D surfaces from the given cloud of points. In the first part an algebraic method based on the adaptive 3L algorithm is presented that makes the original one more flexible. In the second part a fast and flexible fitting technique based on IBS is presented. The flexibility of the method is due to the use of implicit B-splines to define the solution space, and the use of a fast algebraic fitting technique to define the objective function to be minimized. The proposed technique is referred to as the 3L-IBS algorithm since it uses 3L level sets to find the best IBS parameters. Thanks to the local control property of IBS, the optimal control lattice obtained by the 3L-IBS can be edited after running algorithm. Furthermore, the proposed technique is modified in order to only consider the active part of control lattice. It makes the proposed method much faster, while the flexibility is kept. Experimental results are provided for different 2D and 3D public data sets as well as for real 3D clouds of points obtained by a 3D scanner.

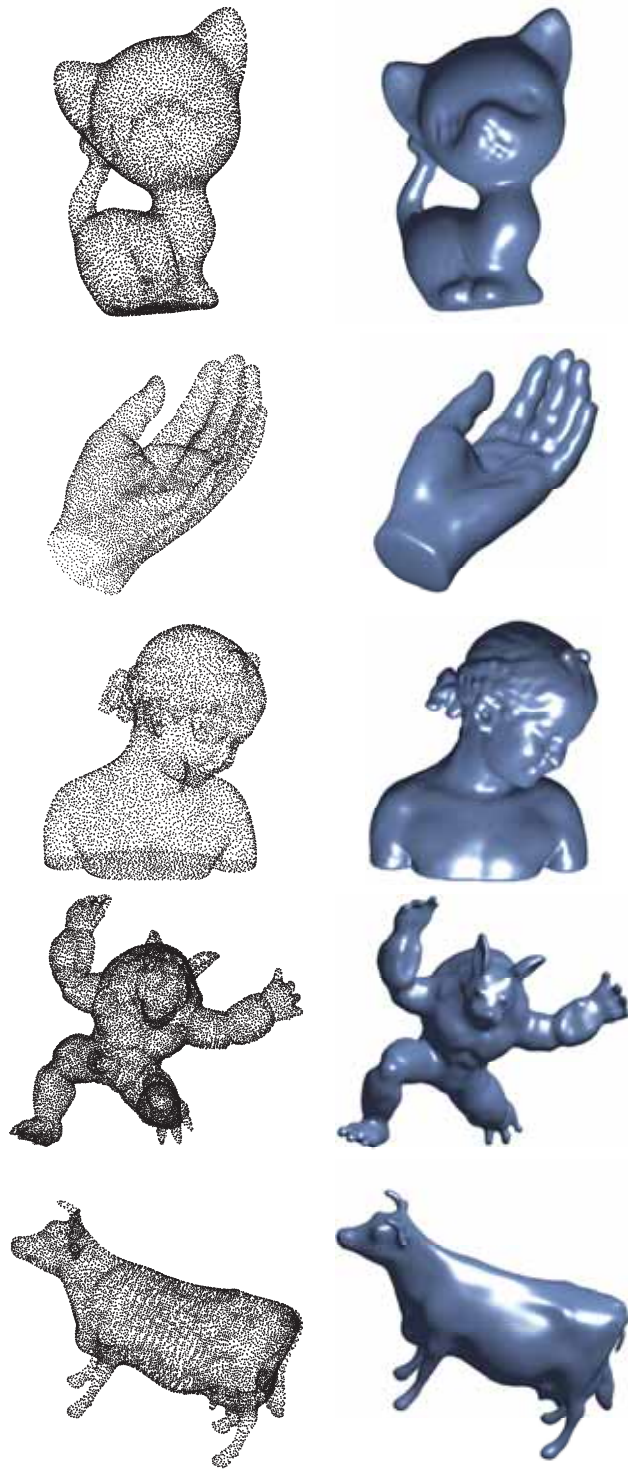


Figure 4.21: (*left*) Original high resolution 3D data sets. (*right*) Results obtained by the proposed 3L-IBS.



Figure 4.22: (*left*) The efficient 3L-IBS method for fitting a dense set of points. (*right*) Enlargement of a region of the surface.

The comparisons with different algebraic techniques with different solution space show the superiority of the proposed algorithm in both CPU time and fitting error.

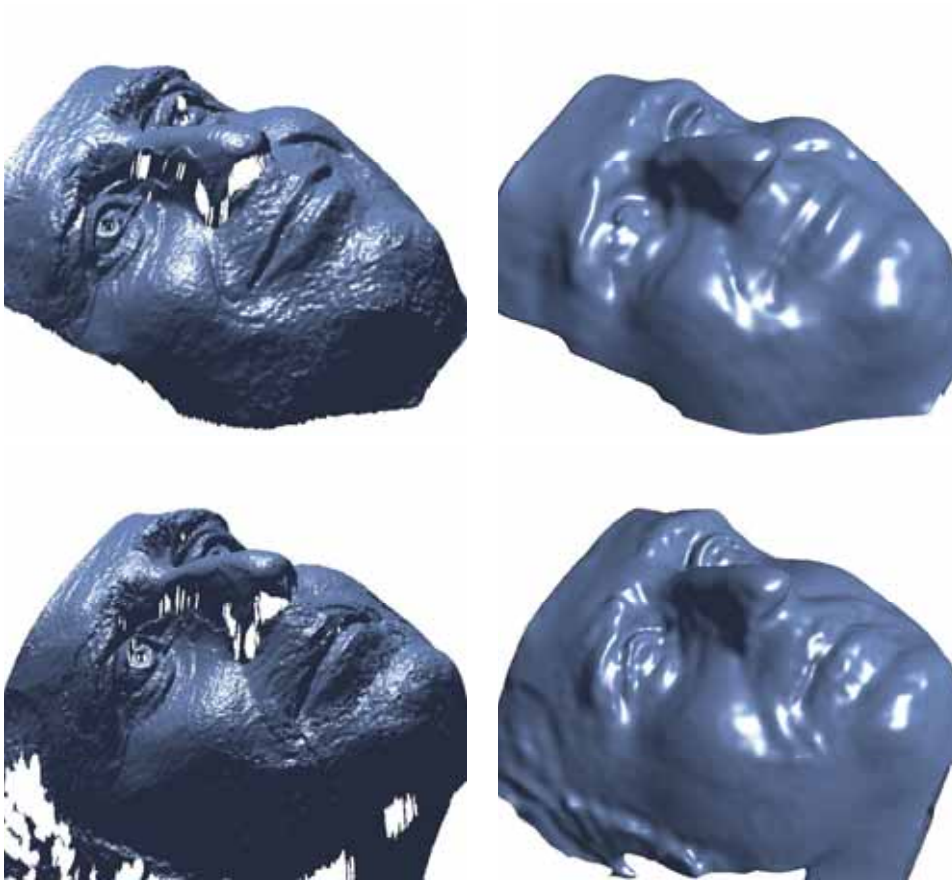


Figure 4.23: (*left*) Triangular meshes of a face. (*right*) Optimal implicit surface obtained through our efficient 3L-IBS fitting method.

Chapter 5

Rigid Registration

This chapter presents a novel formulation, which derives in a smooth minimization problem, to tackle the rigid registration between a given point set and a model set. Unlike most of the existing works, which are based on minimizing a point-wise correspondence term, we propose to describe the model set by means of an implicit representation. It allows a new definition of the registration error, which works beyond the point level representation. Moreover, it could be used in a gradient-based optimization framework. The proposed approach consists of two stages. Firstly, a novel formulation is proposed that relates the registration parameters with the distance between the model and data set. Secondly, the registration parameters are obtained by means of the Levenberg-Marquardt algorithm. Experimental results and comparisons with state of the art show the validity of the proposed framework.

5.1 Introduction

Registration problem has been largely studied in the computer vision community since the last two decades (e.g., [10], [20], [101], [138], [13]). It aims at finding the best *transformation* that places both the given data set and its corresponding model set into the same reference system as *close* as possible. Hence there are two main concepts in this definition to be clarified. The first concept is transformation model, which could varies from rigid to non-rigid deformation model. The second one is regarding the definition of closeness, and how a distance between the model set and data set can be defined. In this chapter we focus on rigid registration methods using different distance definition.

In rigid registration the transformation space is restricted to rigid body motions. In other words the only parameters to be found are related to rotation and translation. Therefore, the optimization problem that finds the best parameter is defined in a low dimensional space (3 parameters in 2D and 6 parameters in 3D). The different rigid registration approaches proposed in the literature can be broadly classified into two categories, depending on whether

an initial information is required (*fine registration*) or not (*coarse registration*); see [111] for a detailed survey.

Coarse rigid registration refers to methods that transform the data set based on a rough criterion. These methods are usually non-iterative, and can be used as an initialization for fine registration. Principal Component Analysis (PCA), for instance, is a coarse registration technique that aligns principal directions. It constructs the covariance matrix between the data and model sets, and then finds the best rotation based on the eigenvectors of this matrix. The best translation moves the data centroid to the model centroid. This method is based on the first order statistics and does not measure the distance between two sets. Hence it may fail to register finely specially when the point density is non-uniform or two sets have a partial overlap.

Fine registration approaches, in general, find the best rigid transformation by iterating two steps. The first step defines the correspondences between the points of the given data set and the points in the model set. They are used to compute the registration residual error. Then, the second step computes the best set of parameters by minimizing this residual error. These steps are repeated until some convergence criteria is reached. The Iterative Closest Point (ICP) algorithm is one of the classical registration approaches following this two-step scheme. It has been originally presented in ([10], [20]); different improvements have been proposed in the literature looking for more efficient and robust solutions (e.g., [125], [160], [47]). Note that in all these approaches the correspondence search step affects the whole registration, which is tackled as a discrete evolution problem.

Some effort to link the registration with the representation problem has been made by using high level representations in order to avoid the correspondence search problem. Implicit polynomials have been used in [130] to represent both the data set and model set. Probabilistic representations have been also used to describe both data set and model set (e.g., [143], [61], [13]). In [34], [99] and [156] the point-wise problem is avoided by using a distance field of the model set. More details about all these approaches are given in next section.

In order to avoid the correspondence search in the first stage, different techniques have been proposed in the literature: *i*) Implicit polynomials have been used in [130] to represent both the data set and model set. Then an accurate pose estimation is computed through constructing two covariants. *ii*) Probabilistic representations have been used to describe both data set and model set (e.g., [143], [61], [13]). *iii*) In [34] the point-wise problem is avoided by using a distance field of the model set; the value and behavior of this distance field is computed in a discrete domain. *iv*) In [99] the behavior of the distance field is approximated analytically based on the curvature information. *v*) An implicit polynomial is used in [156] to fit the distance field, which later defines a gradient field leading the data set towards that model set.

Object representation and point set registration are common problems in the computer vision community. In general, they are tackled as standalone problems and studied separately. The current work places a bridge that connects both problems looking for an efficient solution. Being inspired by the Computer Graphics (CG) and Computer Aided Design (CAD) communities a compact object representation is adopted to reformulate the registration problem in a unified *representation-registration* framework.

The object representation field focuses on developing compact models that allow to deal with large amount of data. Nowadays, due to the improvement in 3D scanners, we are surrounded by a high amount of raw data as a 2D or 3D cloud of points, and having a smooth and compact representation is one of the important objectives that benefit computer vision applications. In the current work we exploit one of the most flexible representations to tackle the registration of two clouds of points.

The current work proposes a novel and fast formulation that exploits a compact and smooth representation as an interface for the registration. It consists of two main stages: in the first stage, an implicit representation is provided for the model set, which is based on an Implicit B-Spline (IBS). The proposed fitting method is in the least squares form, hence it is quite fast. In the second stage, we use an approximated distance to define the residual error in registration. This distance is induced by the IBS fitted in the first stage. The final registration distance is differentiable with respect to the registration parameters and allows solving the registration problem through a gradient based optimization algorithm. Due to the compactness of the proposed representation the whole scheme can be used in a coarse-to-fine framework. The rest of this paper is organized as follows. Section 5.2 presents both the proposed representation and registration approaches. Experimental results and comparisons with state of the art are presented in Section 5.3. Finally, conclusions are given in Section 5.4.

Let us consider two sets of points, referred to as data set $\mathbf{P} = \{p_i\}_1^{N_d}$, and model set $\mathbf{Q} = \{q_i\}_1^{N_m}$ (Fig. 5.1(a)). In the rigid case, the registration problem aims at finding the best rotation and translation in order to take the data set as *close* as possible to the model set. For this purpose many point-to-point comparisons must be done to measure the closeness. For the non-structured case it will take $O(N_d N_m)$ just for obtaining the distance measurement. Although more elaborated solutions using data structure have been introduced, our proposal is to replace the model set with a proper *interface* that facilitates the distance measurement (Fig. 5.1). Then the optimal configuration can be found through measuring the distance to this interface.

In order to work with an interface instead of point set a proper geometric model should be used. Triangle meshes and parametric NURBS are among the common tools in these domains, but they suffer either the geometry limitation or the parametrization problem. Implicit functions, on the contrary, provide a flexible representation without any parametrization problem. They describe objects in 2D/3D through the zero set $\{\mathbf{x} : f(\mathbf{x}) = 0\}$. Implicit Polynomial (IP) [11] is one of the simple choices for f . IPs can describe a given object through a set of coefficients, but they are not flexible due to outliers. Radial Basis Function (RBF) [28] provides another solution space for implicit representations. They are smooth and flexible, but small changes in the coefficient vector can lead to a global change in the whole object.

In this work Implicit functions (IPs or IBSs) are used to represent the model set. These functions propose a smooth and flexible representation without any need of parametrization [3]. Moreover IBS is constructed out of B-spline basis functions, which have compact supports. Hence they have local control (i.e, changes in one coefficient will change a part of the object). Figure 5.1(b) illustrates the flexibility of IBSs to describe a complex 3D shape. In the current work the optimal IBS is easily obtained by means of the 3L algorithm [11], which is a fast algebraic fitting method in the linear least squares form.

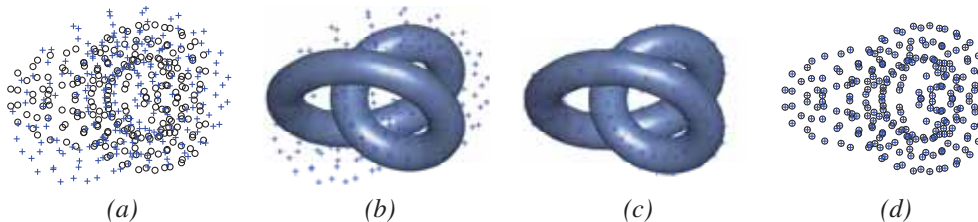


Figure 5.1: Using an interface for point sets registration: (a) Initial position of data (+) and model (o) sets; (b) Data set (+) and model set represented by an IBS; (c) Registration result of data set (+) and the IBS; (d) The same result but represented by using the model set (o) and transformed data set (+).

Once the model set has been described by an IBS or IP the registration problem can be tackled in a point-to-model scheme, which leads to a correspondence free registration method. Firstly, the model set is described with an implicit function, and then the approximated distance between the data set and the fitted polynomial is minimized to find the best rigid transformation. Figure 5.1(b) shows an implicit B-spline that is considered instead of the model set. It should be mentioned that this implicit function is just used as an interface to tackle the registration problem. For instance, in the extreme case, when the data set \mathbf{P} is a rigid transformation of the model set \mathbf{Q} and f_c is the best polynomial fitting the model set, then it could be proved that:

$$\min_{\mathbf{R}, \mathbf{t}} \text{Dist}(\mathbf{R}\mathbf{P} + \mathbf{t}, f_c) = \text{Dist}(\mathbf{Q}, f_c)$$

where Dist refers to the orthogonal distance of a set of points to the implicit polynomial; and $[\mathbf{R}, \mathbf{t}]$ refers to the rotation and translation of the rigid transform.

The remainder of this chapter is organized as follows. Section 5.2 presents the proposed registration approach based on a non-linear minimization of the distance between the given data set and an implicit representation of the model. Experimental results and comparisons with state of the art are presented in Section 5.3. Finally, conclusions and future work are given in Section 5.4.

5.2 Proposed Approach

The proposed approach consists of two main steps. The first step formulates the registration error based on the approximated distance between the current data set and the implicit function used for representing the model set. This formulation relates the error function with the registration parameters. The second step finds the optimal rigid parameters that minimize the proposed registration error through its gradient information. Before running the aforementioned two steps an implicit fitting techniques is applied to describe the model set. It should be noticed that the proposed formulation is valid for any implicit representation (e.g., IPs, implicit RBFs and B-splines).

5.2.1 Distance Formulation

The registration process seeks for the best transformation parameter Θ which contains rotation $\mathbf{R} = \mathbf{R}_\theta$ and translation $\mathbf{t} = [t_x, t_y]^T$ in rigid case. The optimal parameter moves the data set $\mathbf{P} = \{p_i\}_1^N$, in a rigid way, as close as possible to the model $f_c(\mathbf{x})$:

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}} \left(\sum_{i=1}^N \operatorname{Dist}^2(\mathbf{R}p_i + \mathbf{t}, f_c) \right) \quad (5.1)$$

for this purpose, the distance function, Dist , between the data set and the model should be approximated. In the current work, the estimation of the orthogonal distance proposed in [134] is used. This approximation is based on the first order Taylor expansion of the distance function. It has some interesting properties including: *i*) independence of the zero set representation; and *ii*) invariance to rigid body transformation. It is computed through normalizing the algebraic distance by the gradient norm:

$$\operatorname{Dist}(p, f_c) \approx \frac{|f_c(p)|}{\|\nabla f_c(p)\|} \quad (5.2)$$

using this approximation in (5.1) the registration parameters can be found by minimizing the following function:

$$\begin{aligned} \operatorname{Dist}_\Theta &= \sum_{i=1}^N \left(\frac{f_c(\mathbf{R}p_i + \mathbf{t})}{\|\nabla f_c(\mathbf{R}p_i + \mathbf{t})\|} \right)^2 \\ &= \sum_{i=1}^N (w_i f_c(\mathbf{R}p_i + \mathbf{t}))^2 = \sum_{i=1}^N d_i^2 \end{aligned} \quad (5.3)$$

where:

$$\begin{aligned} d_i &= w_i f_c(\mathbf{R}p_i + \mathbf{t}), \\ w_i &= 1/\|\nabla f_c(\mathbf{R}p_i + \mathbf{t})\| \end{aligned} \quad (5.4)$$

show the distance of each item and the weight to approximate this distance. Thus the point-to-point registration will be done in a higher level using a curve or surface as an interface. It will provide a rich structure as well as many advantages like robustness to noise and missing data. Figure 5.2 illustrates how the implicit functions provides a good approximation of the registration distance through its gradient vectors.

5.2.2 Distance Optimization

The distance presented above provides a correspondence free formulation for the registration problem, which is directly related to rigid parameters. This relation could be exploited in many optimization algorithms. Here we use gradient based algorithms like gradient descent and Levenberg–Marquardt algorithm (LMA). The gradient information shows the sensitivity of distance with respect to rigid parameters as illustrated in Fig. 5.3.

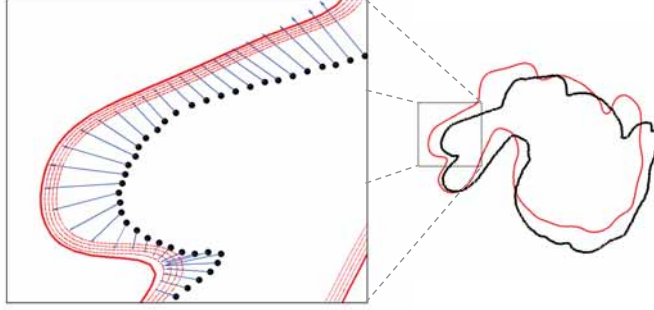


Figure 5.2: The level curves of the IBS is used to approximate the distance. In addition, the distance sensitivity with respect to small changes in rotation and translation can be approximated as well.

LMA is a well-known technique in non-linear optimization [36], which is particularly proposed for functions in the form of sum of squared residuals as the case in (5.3). This method proposes a tradeoff between two well known methods in nonlinear optimization: the Gauss–Newton algorithm and the gradient descent algorithm. In order to handle LMA, the value of the function (5.3) and its partial derivatives, which are expressed in a *Jacobian* matrix J , should be provided. Since LMA uses the gradient information of the objective function, the first order distance approximation in (5.3) captures this information; hence, better approximations would not benefit the result of LMA. Figure 5.3 illustrates how the proposed registration distance provides the distance sensitivity with respect to registration parameters. It should be mentioned that the derivatives of (5.3) must be calculated with respect to the parameters $\Theta = [\theta \quad \mathbf{t}_x \quad \mathbf{t}_y]^T$, where θ , \mathbf{t}_x and \mathbf{t}_y capture the three degrees of freedom of the rigid registration. Hence, the first column of the *Jacobian* matrix can be computed as follow:

$$J(i, 1) = \frac{\partial d_i}{\partial \theta} = (\partial w_i / \partial \theta) f_c(\mathbf{R}p_i + \mathbf{t}) + w_i \frac{\partial f_c(\mathbf{R}p_i + \mathbf{t})}{\partial \theta} \quad (5.5)$$

since the implicit function f_c is a smooth function, w_i could be considered as a constant weight, then the first term could be ignored:

$$J(i, 1) = w_i (\mathbf{R}'_{\theta} p_i \cdot \nabla f_c(\mathbf{R}p_i + \mathbf{t})) \quad (5.6)$$

where \mathbf{R}'_{θ} is the derivative of the rotation matrix w.r.t. the rotation angle, and ∇f_c is the gradient with respect to (x, y) components. Similarly, other columns of the *Jacobian* matrix can be calculated as:

$$J(i, 2) = \frac{\partial d_i}{\partial \mathbf{t}_x} = w_i \frac{\partial}{\partial x} f_c(\mathbf{R}p_i + \mathbf{t}), \quad (5.7)$$

$$J(i, 3) = \frac{\partial d_i}{\partial \mathbf{t}_y} = w_i \frac{\partial}{\partial y} f_c(\mathbf{R}p_i + \mathbf{t}).$$

For the 3D case the *Jacobian* matrix includes six columns corresponding to three rotation and three translation parameters. As a general formula each entry of this matrix could be computed as:

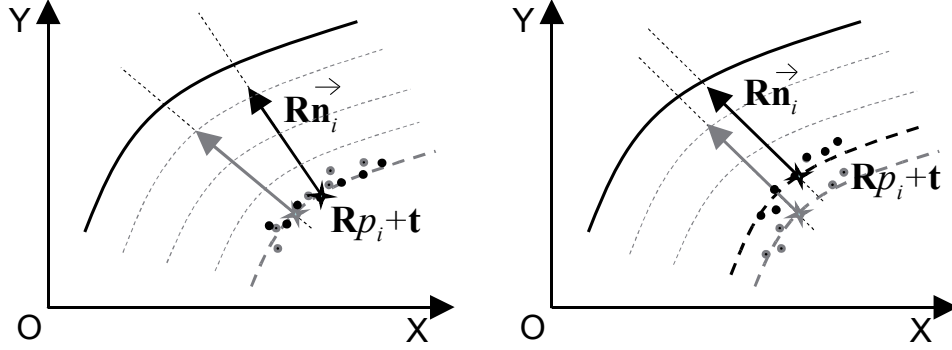


Figure 5.3: (*left*) Sensitivity of the distance with respect to small changes in rotation. (*right*) Sensitivity of the distance with respect to the translation along y axis.

$$J(i, j) = w_i \left(\frac{\partial}{\partial \Theta_j} (\mathbf{R}p_i + \mathbf{t}) \right) \cdot \nabla f(\mathbf{R}p_i + \mathbf{t}) \quad (5.8)$$

where Θ_j is the j th parameter of $\Theta = [\theta, \phi, \psi, \mathbf{t}_x, \mathbf{t}_y, \mathbf{t}_z]$.

Having estimated the proposed distance (5.3) and its Jacobian matrix through (5.5), (5.6) and (5.7) it is easy to perform LMA in order to refine the rigid parameters Θ :

$$\begin{aligned} \Theta^{k+1} &= \Theta^k + \beta \Delta \Theta, \\ (J^T J + \lambda \text{diag}(J^T J)) \Delta \Theta &= J^T D \end{aligned} \quad (5.9)$$

where β is the refinement step; $\text{diag}(J^T J)$ is the diagonal matrix containing the elements of $(J^T J)$; $\Delta \Theta$ represents the refinement vector for the rigid parameters; λ is the damping parameter in LMA; and the vector D is a column vector containing $\text{Dist}(\mathbf{R}p_i + \mathbf{t}, f_c)$, \mathbf{R} and \mathbf{t} are the current rotation and translation respectively. In the current implementation they are initialized as $\theta = 0$, $\mathbf{t}_x = 0$ and $\mathbf{t}_y = 0$; more evolved initializations, such as using simple SVD based techniques, could be used since we are tackling the rigid registration case. Parameter refinement (5.9) must be repeated till convergence is reached.

Figure 5.4 shows an illustration of the convergence region of the proposed registration framework. This region corresponds to the 2D bunny shape case study and its fitting IBS shown in Fig. 5.10(*top*). The axes on this plot correspond to $[\theta, \mathbf{t}_x, \mathbf{t}_y]$ parameters. This figure depicts that the proposed registration method converges to the optimal parameter (the point shown in the center) independent of the initialization in the region (relative position of data and model sets). Different layers in this 3D plot correspond to level surfaces with the same distance (5.3). Our experimental results show that the regularization parameter used in IBS fitting has impact on the convergence region. Indeed a larger regularization parameter leads to a low resolution IBS, and a wider convergence region as a consequence.

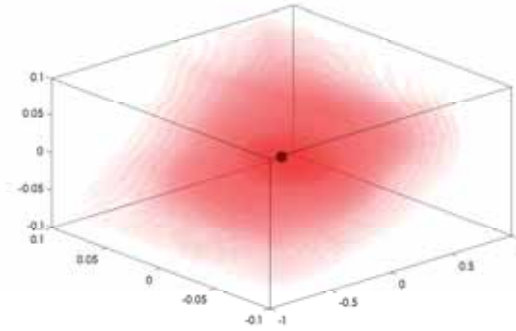


Figure 5.4: Convergence region of the proposed approach with respect to $[\theta, t_x, t_y]$ for the 2D object and IBS in Fig. 5.10(top).

5.3 Experimental Results and Comparisons

The experimental results are presented for the proposed registration distance using different implicit functions. The first part considers simple IPs to represent the model sets. Despite the simplicity of the representation the obtained registration results are quite promising. In the following section this representation is promoted to IBSs that provide more accurate and richer representations.

5.3.1 Registration using IPs

The proposed approach has been evaluated using different data sets and model sets. Additionally four techniques (i.e., [61], [34], [156] and [99]) from the state of the art, together with the classical ICP [10], have been implemented for a comparative study in the 2D and 3D cases. Each techniques iterates till one of the stopping criteria is reached: maximum number of iterations ($\#Iter=30$) or relative registration error smaller than a given threshold. The relative registration error is defined as: $\epsilon = |E_t - E_{t-1}|/E_t$, where E_t refers to the error between the model and data set at iteration t . In our implementation ($\epsilon < 0.001$) has been used.

On the contrary to the relative registration error, which is an internal measure, an *Accumulated Residual Error* (ARE) is used during the comparisons. It is computed by measuring the accumulated error, in a point-wise manner, from the data set to a *reference set*. This reference set corresponds to a highly detailed description of the model set. It contains the model set and on average is defined by a set of points ten times larger than the model set. Each residual error is computed by finding the nearest point in between the registered data set and the reference set.

Figure 5.5 shows initial configurations for four different data and model sets. The first row corresponds to closed contours with a full overlap. Data sets have been obtained by rotating and translating the corresponding model set, and by adding Gaussian noise to study the robustness of all the techniques. Accuracy and number of iterations are provided in Table

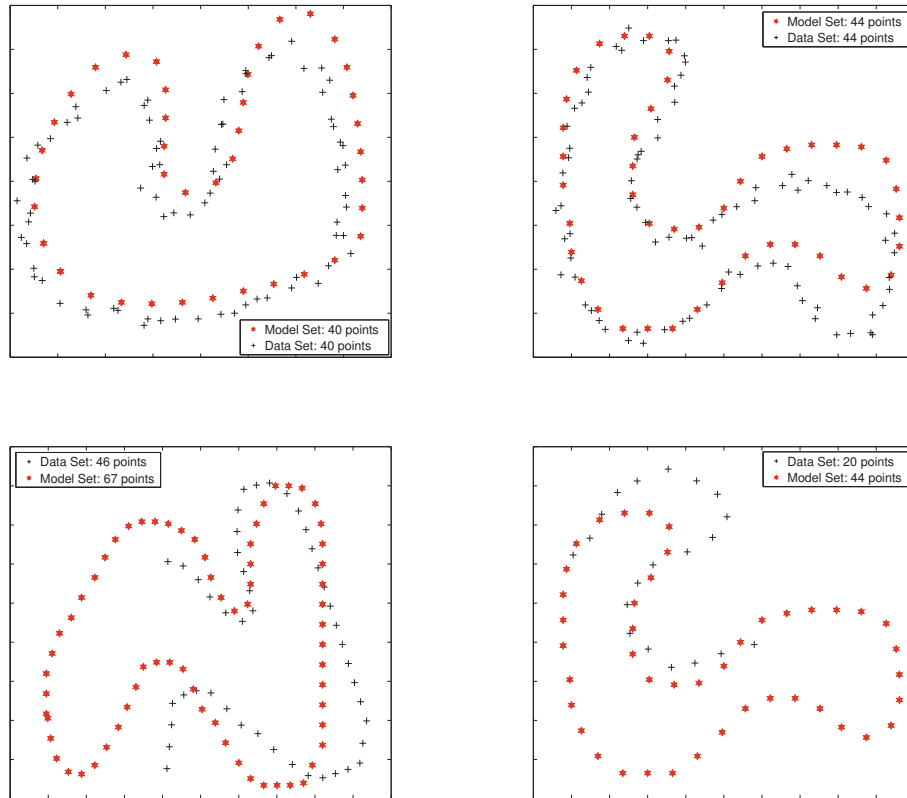


Figure 5.5: Initial positions of data sets and model sets for noisy (*top*) and partial overlap (*bottom*) examples registered with the different approaches.

5.1 and Table 5.2 respectively. It should be highlighted that the proposed approach converges in all the cases and most of the time with the smallest error and lowest number of iterations, in spite of the noise in the data set. In these examples IPs of degree six have been used for fitting the model sets. The IP degree could be automatically determined through the algorithm in [158], which is based on the QR decomposition of the monomial matrix. Figure 5.5 (*bottom*) presents two examples where data set partially overlaps the corresponding model set; data and model sets correspond to uniform sampling of different boundaries. Model sets have been fitted by sixth degree IPs in both cases. Both of them have been registered using the proposed technique and the five aforementioned ones; the obtained registration accuracy is given in the third and fourth rows of Table 5.1, as well as the number of iterations when one of the stopping criteria is reached.

Figure 5.6 presents challenging situations where model sets and data sets contain different densities of points. Fig. 5.6(*left*) shows the initial configurations while Fig. 5.6(*right*) depicts the results obtained by using the proposed approach. Quantitative results from these

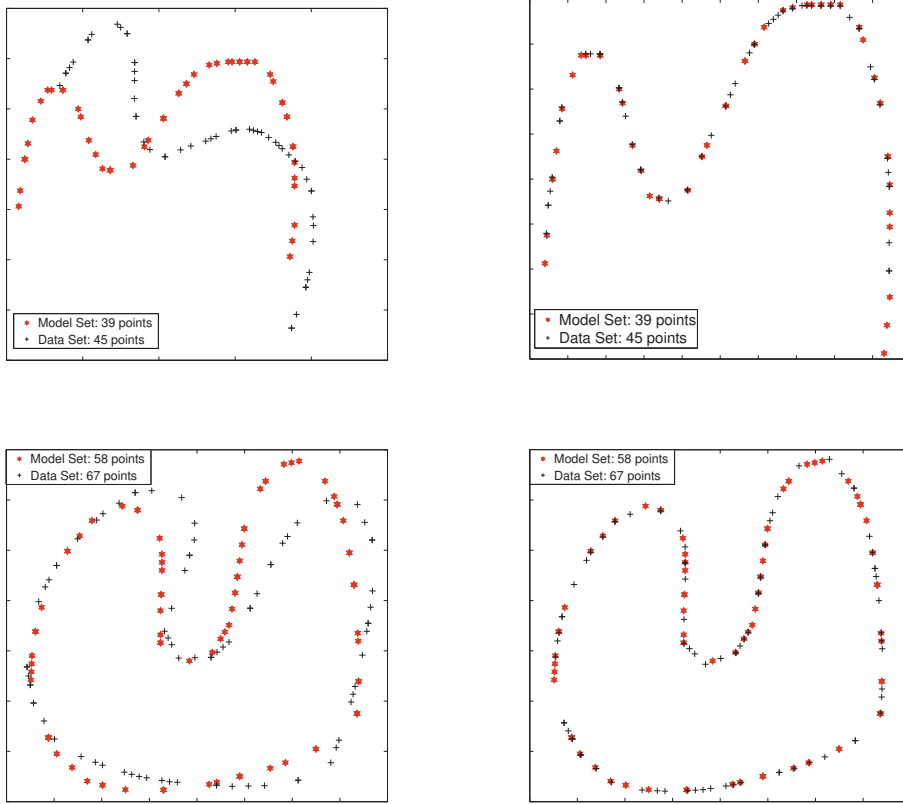


Figure 5.6: Model sets and data sets containing different density of points. (*left*) Initial configurations. (*right*) Final results from the proposed approach.

two examples are presented in Table 6.2. The challenge in these examples lie on the non-existence of any point to point correspondence, although both clouds of points correspond to the same contour. The proposed approach, since the model set is represented by a unified IP, is robust in this kind of situations.

In addition to 2D cases presented above, 3D real objects from public data sets ([52] and [51]) have been registered with the proposed approach and compared with state of the art techniques. The illustration presented in Fig. 5.7(*left* – 1st) corresponds to a data set defined by 811 points. The model set contains 926 points and is represented by means of a seventh degree IP. The result obtained with the proposed approach is shown in Fig. 5.7(*right*). Quantitative information about the registration process, as well as comparisons with other approaches are provided in Table 5.3; the stopping criteria considered in Table 5.1 is also used here.

Figure 5.7 presents three additional experimental results using 3D real data sets; Figure

Table 5.1: Comparisons of registration results for 2D cases (ICP: Iterative Closest Point [10]; GMM: Gaussian Mixture Models [61] ; DT: Distance Transform [34]; GF: Gradient Flow [156]; DA: Distance Approximation [99]; PA: Proposed Approach).

	ICP	GMM	DT	GF	DA	PA
Fig. 5.5(<i>top-left</i>)	1.62	2.32	1.65	1.64	1.65	1.59
Fig. 5.5(<i>top-right</i>)	1.41	1.34	1.42	1.42	1.40	1.32
Fig. 5.5(<i>bot.-left</i>)	0.91	5.11	4.00	0.98	0.92	0.53
Fig. 5.5(<i>bot.-right</i>)	0.52	1.75	0.20	0.29	0.35	0.18
Fig. 5.6(<i>top</i>)	0.26	0.89	0.39	0.48	0.42	0.19
Fig. 5.6(<i>bottom</i>)	1.54	2.48	0.57	1.92	1.22	0.34

Table 5.2: Number of iterations of different registration methods for 2D cases.

	ICP	GMM	DT	GF	DA	PA
Fig. 5.5(<i>top-left</i>)	13	13	25	27	11	4
Fig. 5.5(<i>top-right</i>)	10	7	28	15	9	6
Fig. 5.5(<i>bot.-left</i>)	7	10	30	16	10	9
Fig. 5.5(<i>bot.-right</i>)	13	18	27	20	17	15
Fig. 5.6(<i>top</i>)	14	10	29	12	13	11
Fig. 5.6(<i>bottom</i>)	16	13	30	28	12	13

Table 5.3: Comparisons of registration results for 3D cases (ICP: Iterative Closest Point [10]; GMM: Gaussian Mixture Models [61] ; DT: Distance Transform [34]; GF: Gradient Flow [156]; DA: Distance Approximation [99]; PA: Proposed Approach).

	ICP	GMM	DT	GF	DA	PA
Fig. 5.7(1)	77.19	147.19	85.05	75.71	77.32	75.72
Fig. 5.7(2)	63.69	112.55	63.71	61.61	64.14	61.52
Fig. 5.7(3)	53.45	108.21	52.70	42.39	53.90	42.75
Fig. 5.7(4)	49.25	124.58	53.15	46.49	48.14	46.09
Fig. 5.8(1)	456.64	139.77	109.29	1364.6	146.54	7.40
Fig. 5.8(2)	42.20	185.28	36.09	32.01	47.80	29.30

5.7(*left*) shows initial position of data and model sets both represented by means of triangular meshes to highlight the details. Figure 5.7(*middle*) depicts IPs describing model sets together with the points of their corresponding data sets. A seventh degree IP is used in the (*top*) row to represent the 745 points of the model set, while the data set contains 609 points. A fifth degree IP is used in the (*middle*) row, in this case the data set contains 625 points while the model set is defined by 639 points. Finally, a sixth degree IP is used to describe the 817 points of the model of the example presented in the (*bottom*) row; in this case the data set contains 724 points. Figure 5.7(*right*) presents the registration obtained with the proposed approach. Statistics about their registration process and comparisons with state of the art techniques are presented in Table 5.3.

Table 5.4: Number of iterations of different registration methods for 2D cases.

	ICP	GMM	DT	GF	DA	PA
Fig. 5.7(1)	20	30	14	26	13	10
Fig. 5.7(2)	16	30	6	11	14	8
Fig. 5.7(3)	26	30	9	29	15	10
Fig. 5.7(4)	24	30	15	28	18	13
Fig. 5.8(1)	30	30	30	30	30	27
Fig. 5.8(2)	30	30	30	30	30	30

Finally, two cases where model sets and data sets are partially overlapped are presented in Fig. 5.8. The (*top*) row shows a simple example where the data set (860 points) and model set (835 points) are picked from the same ellipsoid, which is described by a second degree IP in the presented approach. These two sets are partially overlapped (about 40%) as shown in the last column. Despite the simplicity of the problem, none of the techniques presented in Table 5.3, except our approach, converge to the right configuration. All these registration techniques are trapped in a local minimum, while our approach exploits the extrapolation provided by the fitted surface. The (*bottom*) row presents another illustration of partial overlap. In this case, although all the techniques have similar behavior, the proposed approach has the smallest ARE.

The evolution of ARE for registering Fig. 5.7(*bottom*) is illustrated in Fig. 5.9. It can be appreciated that the proposed approach has the smallest ARE and the fastest convergence. Although GF [156] reaches the same optimal ARE its convergence is slower; the oscillation in DT [34] is due to the discrete approximation of the distance field, which is not the case of the proposed approach that has a smooth behavior.

5.3.2 Registration using IBSs

The proposed approach has been evaluated using different 2D and 3D data sets and model sets from public repositories ([52] and [51]). In all the 2D examples, just to visually appreciate the result, the same set of points is used as data and model sets. Notice that the proposed approach does not consider the points in the model set during the registration, despite that after the registration the data points appear on the model points. Figure 5.10 shows illustrations of the proposed approach using these 2D cases. In the example presented in the first row each set contains 115 points. The contours in the other four rows are defined by 167, 174, 148 and 164 points respectively. It should be highlighted that the obtained result is independent of the accuracy of the IBS used to represent the model set. This can be appreciated in the examples presented in the first and third rows. In these cases, even though the IBSs do not fit the model sets accurately, the optimal registration parameters are obtained.

In addition to the qualitative evaluation presented with 2D contours 3D real objects have been registered with the proposed approach and compared with four techniques (i.e., [61], [34], [156] and [99]) from the state of the art, together with the classical ICP [10]. Each techniques iterates till the maximum number of iteration ($\#Iter=40$) is reached or the relative

Table 5.5: Points in data and model sets of the 3D examples [52] presented in Fig. 5.11.

Figure	Name	Data set	Model set
Fig. 5.1	Knot	240 pts.	240 pts.
Fig. 5.11(1 st row)	Kitten	1286 pts.	1557 pts.
Fig. 5.11(2 nd row)	Armadillo	1007 pts.	1151 pts.
Fig. 5.11(3 rd row)	Bimba	811 pts.	926 pts.
Fig. 5.11(4 th row)	Bunny	724 pts.	817 pts.
Fig. 5.11(5 st row)	Hand	1256 pts.	1412 pts.

registration error is smaller than the given threshold ($\epsilon < 0.005$); relative registration error is defined as: $\epsilon = |E_t - E_{t-1}|/E_t$, where E_t refers to the registration error between the model and data set at iteration t . The registration error is used as a quantitative value for the comparisons and it is computed by accumulating the residual error, in a point-wise manner, from data set to a *reference model set*. The reference model set corresponds to a highly detailed description of the model set (it contains on average ten times the number of points in the model set). Residual errors are computed by finding the nearest point in between the registered data set and the reference model set.

Figure 5.11 presents five experimental results obtained with the proposed approach when 3D real data sets are considered. All the information regarding the number of points in data and model sets is provided in Table 5.5. These sets are obtained from the same surface but not necessarily containing the same points. Like in the 2D cases, some of the points in data and model sets are the same, which allow us to visually evaluate the accuracy of proposed approach. Figure 5.11(*left*) presents the initial position of data and model sets. Figure 5.11(*middle*) shows the IBSs representing the different model sets together with the data sets. Finally, Fig. 5.11(*right*) presents the registration results obtained with the proposed approach; IBS surfaces used to represent model sets are kept just to facilitate the visualization. Note how in the cases of the "Hand" even though the IBS used for representing the model set is not accurate (Fig. 5.11(*bottom – middle*)) the proposed approach get the best result (see Table 5.6); in Fig. 5.11(*bottom – right*) a more accurate IBS is depicted just to facilitate the visualization. Table 5.6 and Table 5.7 presents the registration error and the number of iterations, respectively, for all the algorithms tested during the comparisons. In all the cases the proposed approach finds the optimal parameter quite precisely, and in less number of iterations. The novel approach presented in this paper tackles the noise and outliers, specially those affecting the model set, thanks to the fitting algorithm used in the first stage. It could also handle situations where data and model sets contain different densities.

5.4 Conclusions and Future Work

In this chapter a flexible implicit representation is exploited to tackle the registration problem. As the first contribution IBSs are used to describe cloud of points. The optimal IBS is obtained through a linear least squares formulation; furthermore its smoothness can be easily

Table 5.6: Comparisons of IBS registration results for 3D cases (ICP: Iterative Closest Point [10]; GMM: Gaussian Mixture Models [61]; DT: Distance Transform [34]; GF: Gradient Flow [156]; DA: Distance Approximation [99]; PA: Proposed Approach).

	ICP	GMM	DT	GF	DA	PA
Fig. 5.1	8.51	13.77	5.54	1.52	5.63	0.06
Fig. 5.11(1 st row)	3.67	36.87	4.10	3.67	3.68	3.67
Fig. 5.11(2 nd row)	2.66	49.83	2.90	5.50	2.73	2.75
Fig. 5.11(3 rd row)	18.60	33.89	3.23	3.36	3.04	3.06
Fig. 5.11(4 th row)	2.49	22.95	2.82	3.72	2.59	2.48
Fig. 5.11(5 th row)	4.86	87.09	5.46	9.24	4.88	4.85

Table 5.7: Number of iterations of different registration methods for 2D cases.

	ICP	GMM	DT	GF	DA	PA
Fig. 5.1	14	30	7	30	15	11
Fig. 5.11(1 st row)	22	30	3.67	15	22	8
Fig. 5.11(2 nd row)	12	30	8	28	12	9
Fig. 5.11(3 rd row)	30	30	11	30	19	8
Fig. 5.11(4 th row)	15	30	8	19	13	8
Fig. 5.11(5 th row)	29	30	12	30	30	12

controlled by the regularization parameter. As the second contribution we use the flexibility of IBS to propose a registration distance. The resulting distance and its gradient information can be easily computed and it fits the requirement of any gradient based optimization algorithm. Experimental results and comparisons are provided showing both fast convergence and robustness in challenging situations. It should be mentioned that this work is mainly focuses on the registration distance approximation. However the transformation domain is only restricted to the rigid class. As a future work the proposed registration distance will be considered in a non-rigid deformation space.

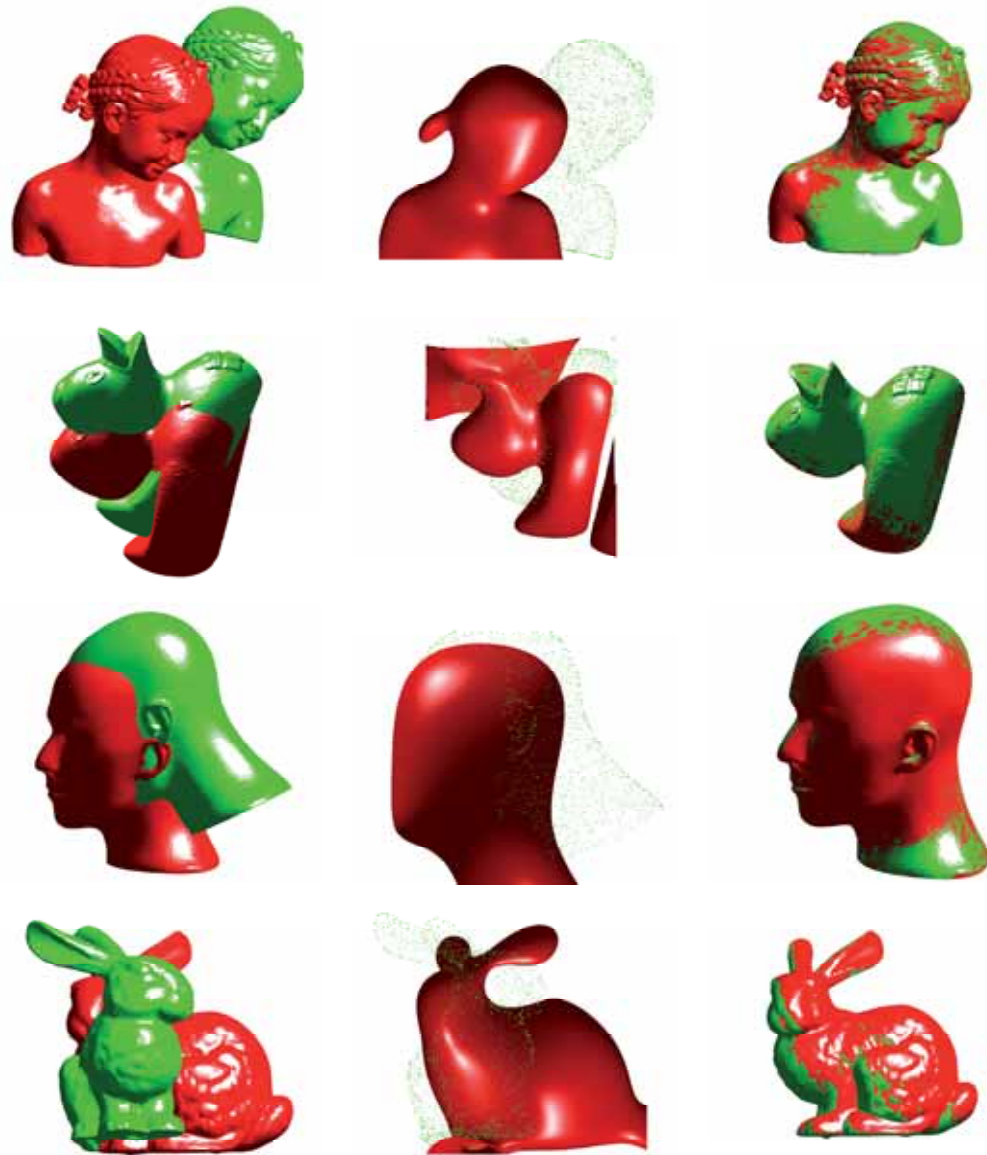


Figure 5.7: Real data sets (from [52] and [51]) registered with the proposed approach and state of the art techniques. (*left*) Initial set up of the given data and model sets represented by means of triangular meshes to highlight details. (*middle*) IPs representing model sets and data points. (*right*) Results of the proposed registration approach represented through triangular meshes to make easier a visual evaluation.

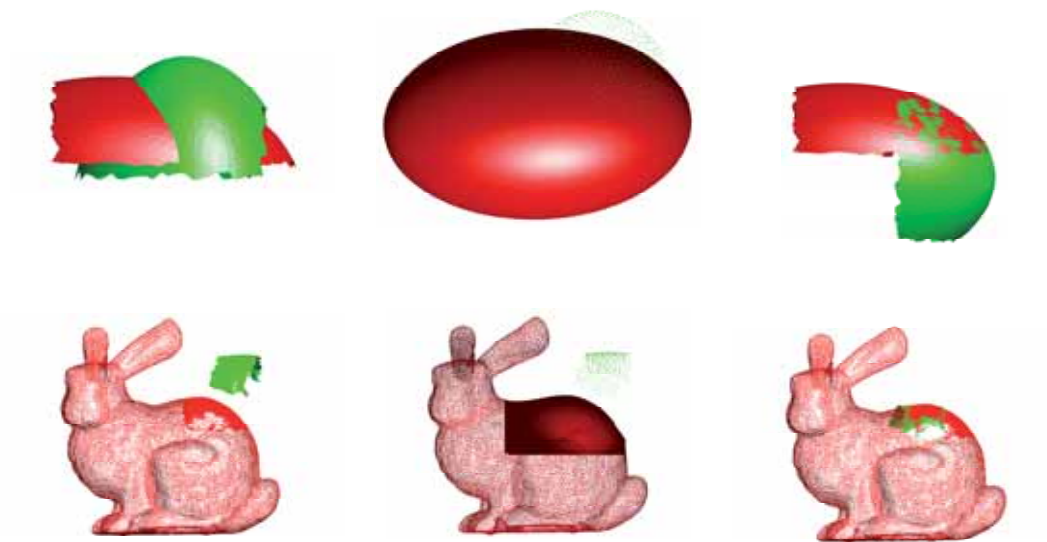


Figure 5.8: Partial overlap cases. (*left*) Initial set up of data sets and model sets to be registered. (*middle*) IPs representing model sets and data points from the data sets. (*right*) Results from the proposed approach.

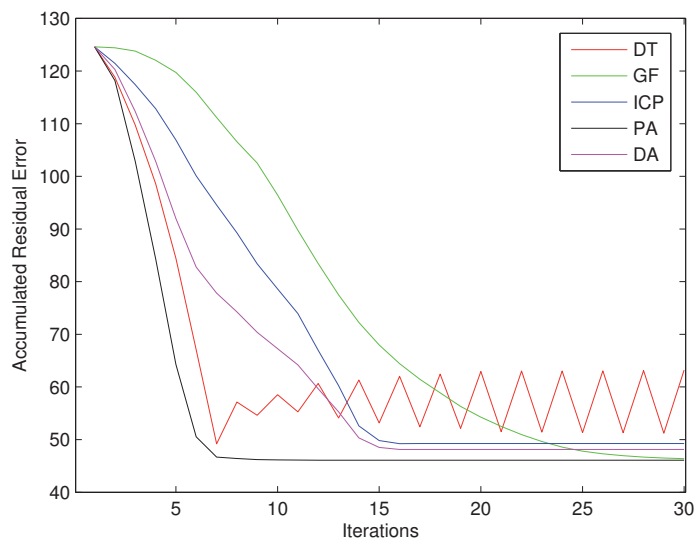


Figure 5.9: Evolution of ARE of different registration algorithms along 30 iterations.

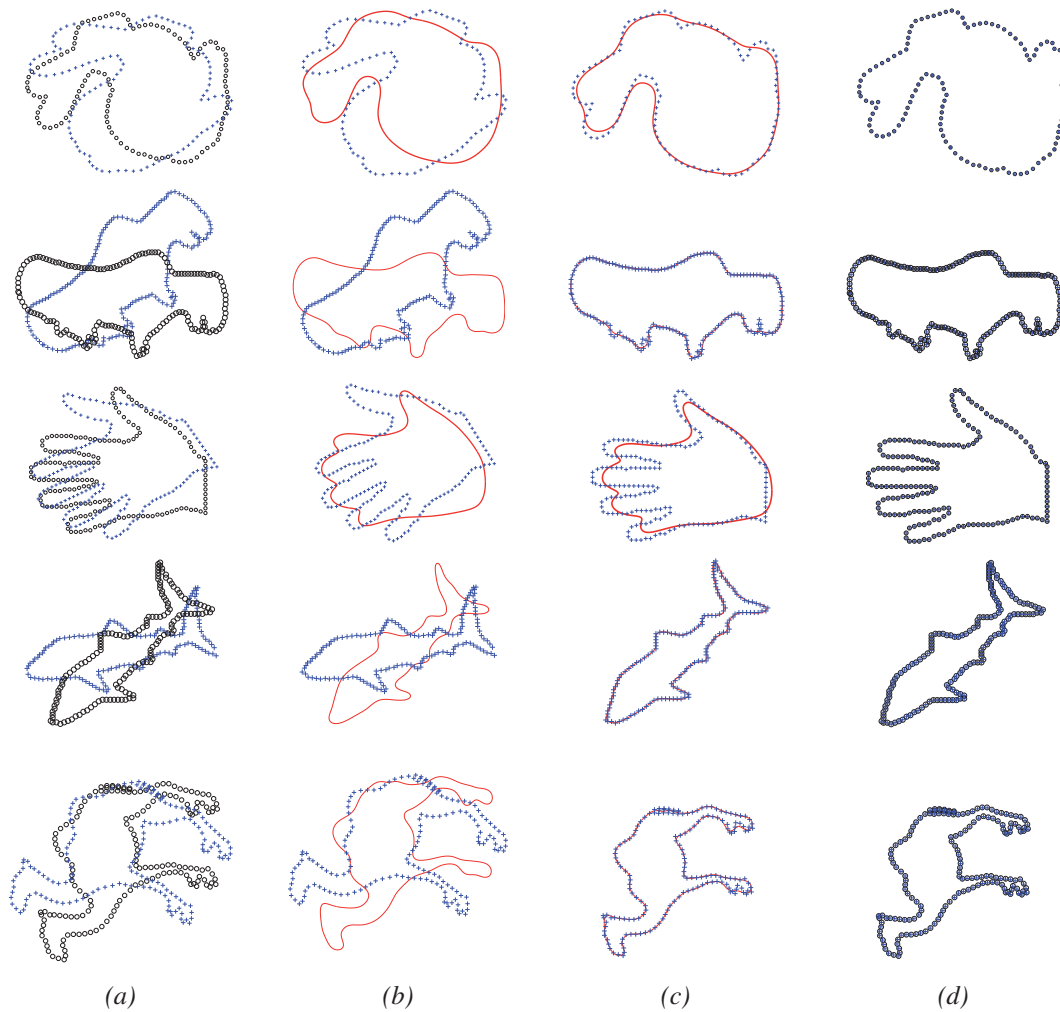


Figure 5.10: (a) Initial positions of data (+) and model (o) sets. (b) Data (+) sets and IBSs representing the models. (c) Final results of registered data (+) sets and IBSs. (d) The same result but represented by using the model (o) sets and transformed data (+) sets with the proposed approach.

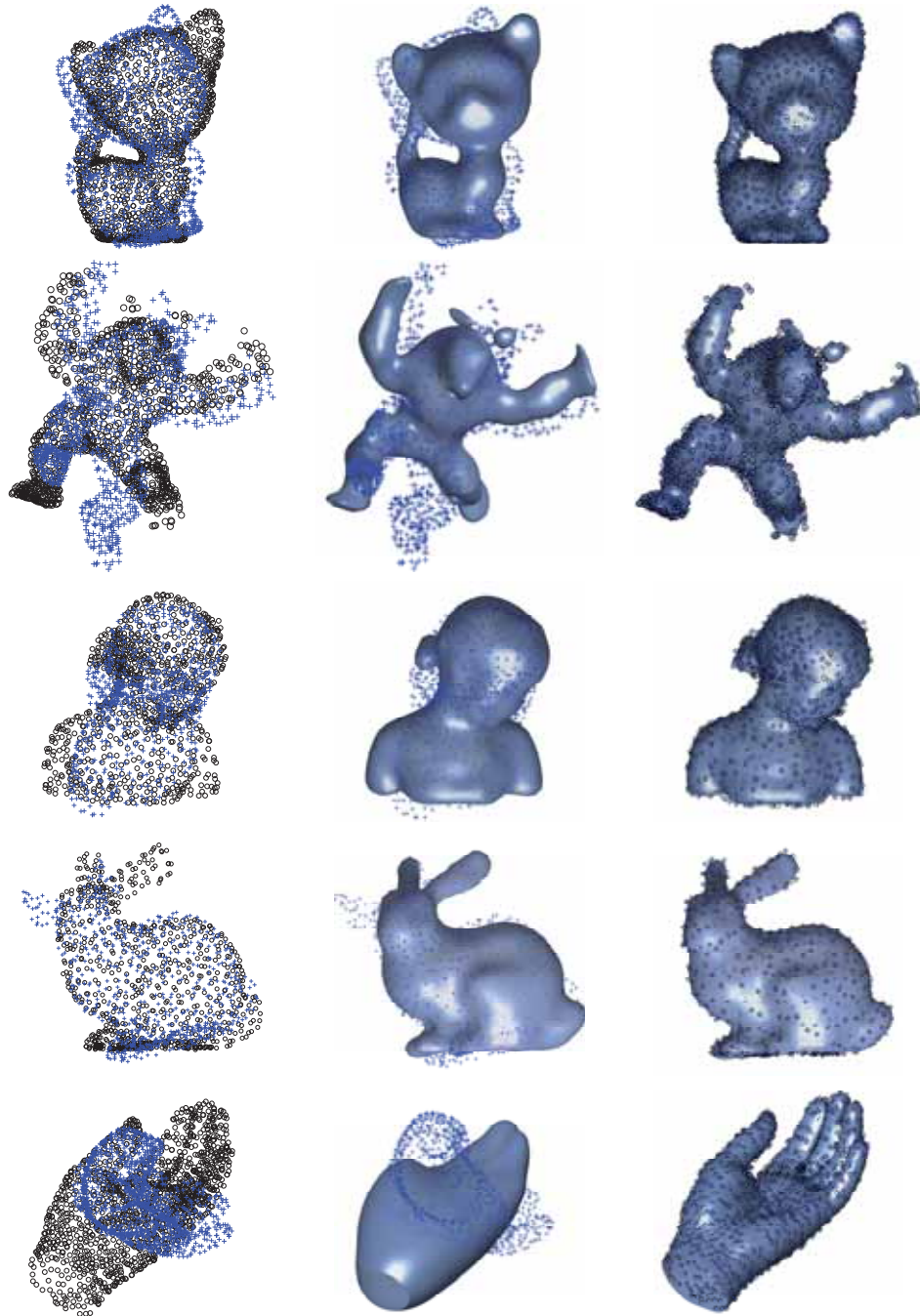


Figure 5.11: 3D cases corresponding to real data sets registered with the proposed approach and state of the art techniques: (*left*) Initial positions of data (+) sets and model (o) sets. (*middle*) Data (+) sets and model sets represented by IBSs; (*right*) Final results of registered data (+) and model (o) sets with the proposed approach.

Chapter 6

Non-Rigid Registration

This chapter proposes a non-rigid registration formulation capturing both global and local deformations in a single framework. This formulation is based on a quadratic estimation of the registration distance together with a quadratic regularization term. Hence, the optimal transformation parameters are easily obtained by solving a linear system of equations, which guarantee a fast convergence. Experimental results with challenging 2D and 3D shapes are presented to show the validity of the proposed framework. Furthermore, comparisons with the most relevant approaches are provided.

6.1 Introduction

The shape registration problem has been largely studied in the literature and represents a fundamental problem in different computer vision and image processing applications. It aims at recovering a set of transformation parameters that brings a given data set as close as possible to the corresponding model set. In the rigid case, also known as shape alignment, it involves rotations and translations. While in the non-rigid shape registration case, in addition to the rotation and translation parameters it includes a deformation stage. The development of formulations able to tackle the non-rigid registration case are attracting the interest of the research community. They are capable of handling situations with shape distortions due to deformation, noise or missing parts (e.g., [58], [16], [29]).

In general, most of the approaches proposed for non-rigid shape registration follow a two step scheme, where first a global rigid alignment is performed and then a local process deforms the shape of data set towards the given model set. The differences between the methods in the literature mainly lie in the way they formulate these two steps (most relevant approaches are summarized in the next section). Variations to this global-rigid/local-non-rigid strategy have been also proposed in the literature. For instance, an alternative has been recently introduced in [39], where shape rigidity is firstly considered locally and then a global

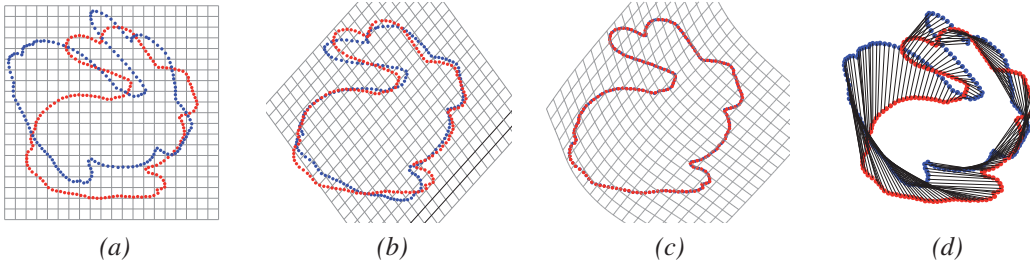


Figure 6.1: The proposed method (SD-FFD): (a) initial configuration of the data set (blue) and the model set (red); (b) result with $\lambda = 10^5$ (at iteration 15); (c) result with $\lambda = 1$ (at iteration 22); (d) the optimal FFD control lattice results in a very dense correspondence, not only on the boundary but over the whole space.

shape deformation process is performed.

On the contrary to previous approaches, the current work proposes to use a single formulation to tackle both, global and local alignment and deformation. The main contributions of current work are as follow: (i) a robust distance approximation based on local curvature information is used for non-rigid registration; (ii) the proposed objective function is in the linear least squares form, hence it can be solved by a linear system of equations; (iii) the proposed method captures all deformation from rigid to non-rigid by the same framework; there is no need to use different steps to capture global and local deformations separately; (iv) unlike the sign distance field, the proposed function is not discretely approximated.

The rest of the paper is organized as follows. The proposed technique is presented in Section 6.2. Section 6.3 gives experimental results using 2D and 3D shapes; additionally comparisons with state of the art are presented. Finally, conclusions and future work are detailed in Section 6.4.

As mentioned in Chapter 2 there have been different approximations of the distance between model and data sets. Choosing a proper error term leads to a precise and fast registration algorithm. Most of the methods previously reviewed result in a non-linear error term, which must be iteratively optimized. In the current work we exploit a quadratic distance approximation in the non-rigid registration problem. The presented objective function is in the linear least squares form and can handle both rigid and non-rigid deformations with the same framework. Figure 6.1 illustrates how the proposed method handles both the rigid alignment (Fig. 6.1(b)) and the non-rigid deformation (Fig. 6.1(c)) just by relaxing the regularization term. This term controls the rigidity of deformation during the evolution. Thanks to the proposed objective function each iteration is linearly solved. Hence, the whole framework has a fast convergence. Furthermore, the optimal deformation provides a dense correspondence between the given data and model set.

6.2 Non-rigid Registration

This section presents the main elements of the proposed framework. Firstly, in Section 6.2.1, the approximation error used to estimate the distance between the current data set and model set is presented. This distance is a quadratic term based on the curvature information of the points in the model set. Our contribution is to use this distance to capture both rigid and non-rigid deformations by means of the Free Form Deformation. In Section 6.2.2 this deformation space is defined. Finally, Section 6.2.3 details how both rigid and non-rigid registration problems can be solved in the same framework. In this section we propose a novel objective function to find the deformation parameters. This function is in the least squares form, and it can be easily optimized by solving a linear system of equations.

6.2.1 Registration Error (SD)

All the registration methods seek for the best transformation parameters to move the given *data set* (Source shape) $\mathcal{S} = \{\mathbf{s}_i\}_1^{N_d}$ close to the *model set* (Target shape) $\mathcal{T} = \{\mathbf{t}_i\}_1^{N_m}$. As mentioned in the previous Section, all registration methods can be classified based on the distance used to measure the closeness, the transformation to move the data set, and the optimization method that finds the best transformation parameters. The first and most important matter is how to choose a proper and precise distance to define the registration error term.

Approaches using the precise geometric distance between model and data sets have been proposed in the literature. For instance, a well known example is the Iterative Closest Point (ICP) algorithm. It moves the data set in each iteration based on a simple criterion: for the given data point it searches for the closest corresponding model point (or *foot-point*). Therefore, the distance used by ICP is a point-to-point distance, and ICP performs a point distance minimization (PDM) in each iteration to find the best transformation parameters. Figure 6.2(a) illustrates a simple case where ICP is stuck in a local minima. Some of the data points in the figure lie on the curve passing through the model set; hence their distance to the model set must be quite low, but ICP devotes a quite high distance to these points since the model set is quite sparse. If there could be a better approximation for the distance, the ICP would devote more weights to the data point which are still far from the model point. More elaborated approaches have been also proposed using ICP philosophy [108]; in [20] ICP is used in a tangent distance minimization (TDM) framework.

Implicit descriptions like distance fields provide another metric to measure the distance between the data and model sets. In these techniques both model set and data set [58], or only the model set [34], are described by signed distance fields at first. Then, the registration error is measured through these distance fields instead of the point sets. Unfortunately, these methods require expensive computation to build the distance fields over the whole region. In addition, since the distance fields are made discretely, the precision is up to a specific point. Finally, using distance fields for non-rigid registration results in a non-linear optimization function, which is usually solved by a gradient descent algorithm.

In this paper we use a quadratic approximation of the geometric distance in order to define

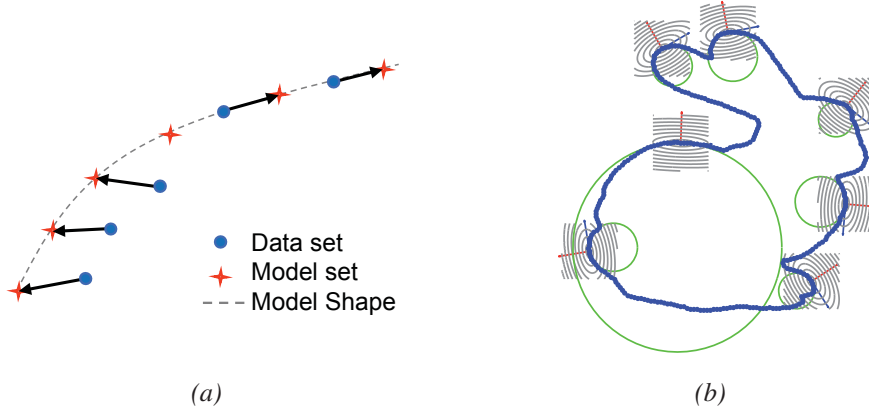


Figure 6.2: (a) Illustration of a point-to-point distance based approach (e.g., ICP [20]). (b) Local quadratic approximation [99] used in current work.

the registration error term in the least squares form. This distance is based on the curvature information in the model sets. Consider the data point \mathbf{s}_i with its closest corresponding model point \mathbf{t}_j . Then the Squared Distance (SD) of \mathbf{s}_i to the whole model set \mathcal{T} can be approximated as follows:

$$SD(\mathbf{s}_i, \mathcal{T}) = \frac{d}{d - \rho} [(\mathbf{s}_i - \mathbf{t}_j) \cdot \mathbf{T}_j]^2 + [(\mathbf{s}_i - \mathbf{t}_j) \cdot \mathbf{N}_j]^2 \quad (6.1)$$

where \mathbf{T}_j and \mathbf{N}_j are the unit tangent and unit outer normal, respectively, defined in the Frenet frame at \mathbf{t}_j . The value ρ is the curvature radius at the model point \mathbf{t}_j and d is the signed distance between the data point \mathbf{s}_i and the model point \mathbf{t}_j . The sign of d is positive if \mathbf{s}_i and \mathbf{N}_j lie on the same side, and is negative otherwise [99].

The distance approximation in (6.1), know as SD, works with the Frenet frame at the foot-point \mathbf{t}_j . It project the data point on the normal and tangent vectors firstly, and the final approximation will be quadratic with respect to these projection. This final property of SD is very important and fits our need, since it results in a least squares form. In the special case, where the data point is along the normal at the foot-point, the first quadratic term vanishes, and the distance will be equal to $|\mathbf{s}_i - \mathbf{t}_j|^2$, which is the squared point to point distance. In another special case, where the curvature of model set at \mathbf{t}_j is zero, the first quadratic term vanishes again, and the projection of the data point on the normal will be the SD approximation of quadratic distance. Figure 6.2(b) shows an illustration depicting quadratic approximations for a few points of a given 2D shape.

6.2.2 Deformation Space (FFD)

Rigid transformation is able to align the global appearance of the objects. Hence, in order to capture the local deformation we should use a more flexible family of transformation. In the current work we propose to use a Free Form Deformation (FFD) to describe any transformation from global (rigid) to local (non-rigid). FFD has been already used in the computer vision and graphics communities in the form of iFFD (incremental FFD); in other words,

they were considered only for capturing local deformation. In this work, thanks to the metric selected in Section 6.2.1, a single framework is used to apply FFD to describe the deformation space; without loss of generality let us consider the 2D case, where the FFD describes a deformation field by means of the *control lattice* $\{\mathbf{P}_{ij}\}_{M \times N}$ in 2D:

$$\mathbf{L}(x, y) = \sum_{i=1}^M \sum_{j=1}^N \mathbf{P}_{i,j} B_i(x) B_j(y) \quad (6.2)$$

where $\{B_i(x)B_j(y)\}$ are cubic spline basis functions to guarantee C^2 continuity. In our implementation we use a square control lattice ($M = N$) covering the unit square $[0, 1]^2$. The B-Spline knot sequence is uniform with a step of $\Delta = 1/(N - 3)$.

Since we consider a square control lattice, both sets of basis functions behave similarly. Having considered a row-by-row order, we can represent the control lattice and the basis functions in a vector form:

$$\mathbf{L}(x, y) = \begin{bmatrix} \mathbf{p}_x^T \mathbf{m}(x, y) \\ \mathbf{p}_y^T \mathbf{m}(x, y) \end{bmatrix} = \begin{bmatrix} \mathbf{p}_x^T \\ \mathbf{p}_y^T \end{bmatrix} \mathbf{m}(x, y) \quad (6.3)$$

where $\mathbf{m}(x)$ is the vector form of the monomials $\{B_i(x)B_j(y)\}$, and $\mathbf{p}_x, \mathbf{p}_y$ are the vector form of the x and y components of control lattice.

The FFD definition in (6.2) can be simplified through the *blending functions* which are cubic patches on $[0, 1]$, which builds up the B-Spline basis function by assembling together:

$$\begin{aligned} b_0(u) &= (1 - u)^3/6, & b_1(u) &= (3u^3 - 6u^2 + 4)/6, \\ b_2(u) &= (-3u^3 + 3u^2 + 3u + 1)/6, & b_3(u) &= u^3/6. \end{aligned} \quad (6.4)$$

Then an equivalent definition of FFD will be achieved that is computationally useful:

$$\mathbf{L}(x, y) = \sum_{r=0}^3 \sum_{s=0}^3 \mathbf{P}_{i+r, j+s} b_r(u) b_s(v) \quad (6.5)$$

where the indices start from:

$$i = \lfloor x/\Delta \rfloor + 1, \quad j = \lfloor y/\Delta \rfloor + 1 \quad (6.6)$$

and the given coordinates in XY will be mapped in UV as:

$$\begin{aligned} u &= x/\Delta - \lfloor x/\Delta \rfloor, \\ v &= y/\Delta - \lfloor y/\Delta \rfloor. \end{aligned} \quad (6.7)$$

This definition provides us with the computational efficiency useful for calculating the monomial matrix. Note that the value $b_r(u).b_s(v)$ will be accumulated in the proper cell of the monomial $\mathbf{m}(x, y)$ corresponding to $B_{i+r}(x).B_{j+s}(y)$.

The Free Form Deformation in (6.2) has $2N^2$ degrees of freedom due to the free movement of the control lattice. It should be indicated that this movement should be controlled in

order to have a meaningful deformation. Otherwise we will have only a 2D to 2D mapping. Moreover, using this FFD formulation together with the SD may leads us to a singularity problem in the least squares solution presented in the next section. In order to avoid these problems a regularization term must be considered as well.

In the current work a global tension term is considered to regularize the control lattice. On the contrary to the conventional regularization term used in iFFD, which measure the first order changes of iFFD, we use a second order term. This term, similarly to [3], is computed by measuring the curvature of \mathbf{L} over the whole domain:

$$T(\mathbf{P}) = \iint_{XY} \|\mathbf{L}_{xx}\|^2 + 2\|\mathbf{L}_{xy}\|^2 + \|\mathbf{L}_{yy}\|^2 dx dy. \quad (6.8)$$

Since the vector field $\mathbf{L}(x, y)$ is a linear function of \mathbf{P} , the whole regularization term will be a quadratic function of \mathbf{P} . Using the vector form of \mathbf{L} in (6.3) the regularization term can be simplified as follows:

$$T(\mathbf{P}) = \mathbf{p}_x^T \mathbf{H} \mathbf{p}_x + \mathbf{p}_y^T \mathbf{H} \mathbf{p}_y \quad (6.9)$$

where matrix \mathbf{H} is a $N^2 \times N^2$ symmetric matrix including the integral of basis functions' derivatives:

$$\mathbf{H} = \iint_{XY} \mathbf{m}_{xx} \mathbf{m}_{xx}^T + 2\mathbf{m}_{xy} \mathbf{m}_{xy}^T + \mathbf{m}_{yy} \mathbf{m}_{yy}^T dx dy. \quad (6.10)$$

This matrix can be analytically constructed once the size of control lattice is known. Hence it can be computed off-line, and be use during the algorithm.

6.2.3 SD-FFD: A Novel Non-Rigid Registration

So far the registration error (SD) as well as the transformation model (FFD) are defined, where the first one defines the fitting term to measure the external energy and the second one defines the solution space to describe the deformation. Assembling these two terms will result in a novel non-rigid registration method:

$$\varphi(\mathbf{P}) = \sum_{i=1}^{N_d} SD(\mathbf{L}(\mathbf{s}_i), \mathcal{T}) + \lambda T(\mathbf{P}). \quad (6.11)$$

Our proposed registration function is a function of the control lattice \mathbf{P} consisting of the data fitting term and the regularization term. As defined in the pervious Section, the regularization term $T(\mathbf{P})$ is quadratic with respect to \mathbf{P} . In addition, since SD is quadratic with respect to the given coordinates, and $\mathbf{L}(\mathbf{s}_i)$ is linear with respect to \mathbf{P} , the whole registration function in (6.11) is linear in terms of the control lattice coordinates.

Thanks to vector form representation our proposed registration distance can be reformu-

lated as follows:

$$\begin{aligned} \varphi(\mathbf{p}_x, \mathbf{p}_y) &= \sum_{i=1}^{N_d} \omega_i \left[\begin{bmatrix} \mathbf{p}_x^T \\ \mathbf{p}_y^T \end{bmatrix} \mathbf{m}(\mathbf{s}_i) - \mathbf{t}_j \cdot \mathbf{T}_j \right]^2 \\ &+ \sum_{i=1}^{N_d} \left[\begin{bmatrix} \mathbf{p}_x^T \\ \mathbf{p}_y^T \end{bmatrix} \mathbf{m}(\mathbf{s}_i) - \mathbf{t}_j \cdot \mathbf{N}_j \right]^2 + \lambda (\mathbf{p}_x^T \mathbf{H} \mathbf{p}_x + \mathbf{p}_y^T \mathbf{H} \mathbf{p}_y). \end{aligned} \quad (6.12)$$

It is now clear that the function φ is quadratic with respect to \mathbf{p}_x and \mathbf{p}_y ; hence vanishing the partial derivatives $\frac{\partial \varphi}{\partial \mathbf{p}_x}$ and $\frac{\partial \varphi}{\partial \mathbf{p}_y}$ result in two linear system of equations $\mathbf{A}_x \mathbf{p}_x = \mathbf{b}_x$ and $\mathbf{A}_y \mathbf{p}_y = \mathbf{b}_y$ where:

$$\begin{aligned} \mathbf{A}_x &= \lambda \mathbf{H} + \sum_{i=1}^{N_d} (\omega_i \mathbf{T}_j^{x2} + \mathbf{N}_j^{x2}) \mathbf{m}(\mathbf{s}_i) \mathbf{m}(\mathbf{s}_i)^T \\ \mathbf{b}_x &= \sum_{i=1}^{N_d} (\omega_i \mathbf{T}_j^{x2} + \mathbf{N}_j^{x2}) (\mathbf{t}_j^x) \mathbf{m}(\mathbf{s}_i). \end{aligned} \quad (6.13)$$

Similarly, the coefficient matrix \mathbf{A}_y and the right hand vector \mathbf{b}_y corresponding to the y coordinate of the control lattice can be obtained.

Therefore, our proposed method, SD-FFD, finds the optimal control lattice through solving two linear system of equations in each iteration. In order to converge to the global minimum, we can start with a high regularization parameter λ and decrease it gradually. It must be mentioned that SD-FFD, unlike other methods, neither uses implicit distance field, which is computationally expensive, nor relies on the single corresponding foot-point point. SD-FFD uses the local curvature information around the foot-point and use this information to build up a quadratic function.

6.3 Experimental Results

The performance of the proposed approach has been evaluated and compared with state of the art algorithms. Several 2D and 3D shapes, obtained from public databases ([52], [127] and [51]), have been registered. In all the cases the data set corresponds to a deformed shape of the model set; as an exception, Fig. 6.3(*bottom*) shows the result when shapes from different objects are registered together—data set corresponds to a *Donkey* 2D shape, while model set to a *Cat* 2D shape. Figure 6.3 shows seven illustrations of 2D shapes registered with the proposed approach. Figure 6.3(*a*) presents the initial configurations where not only deformation but also rotations and translations between model and data sets can be appreciated. In the current implementation the regularization parameter (λ), which somehow represents the registration rigidity, was automatically tuned. It starts with a high regularization value ($\lambda = 10^5$, see illustrations in Fig. 6.3(*b*)), which is mainly devoted to tackle the alignment problem. Once the ratio of registration error between consecutive iterations is below a given threshold λ is divided by 10; this relaxation is performed till $\lambda = 1$. Figure 6.3(*c*) depicts intermediate results, while Fig. 6.3(*d*) presents results after convergence is reached.

Table 6.1: Points in data and model sets of the 3D examples [52] presented in Fig. 6.1 and Fig. 6.3.

Figure	Name	Data set	Model set
Fig. 6.1	Bunny	341 pts.	341 pts.
Fig. 6.3(1 st row)	Dude	415 pts.	361 pts.
Fig. 6.3(2 nd row)	Fish	297 pts.	293 pts.
Fig. 6.3(3 rd row)	Rabbit	272 pts.	253 pts.
Fig. 6.3(4 th row)	Bird	376 pts.	299 pts.
Fig. 6.3(5 th row)	Misk	545 pts.	529 pts.
Fig. 6.3(6 th row)	Camel	598 pts.	535 pts.
Fig. 6.3(7 th row)	Cat	485 pts.	361 pts.

All the 2D examples presented above (Fig. 6.1 and Fig. 6.3) have been used to compare the results from the proposed approach with two state of the art algorithms (i.e., [58] and [39]). Additionally, the performance of the proposed framework is evaluated by using a point wise based approach. In other words, instead of using the quadratic approximation of the geometric distance (Section 6.2.1) a precise point-to-point distance is considered. This second approach is only implemented for comparisons and will be referred to as ICP-FFD. This ICP-FFD has been chosen since it is simple and can be drive from SD-FFD as a special case. During the comparisons, the techniques iterate till the maximum number of iterations (#Iter=50) is reached or the relative registration error is smaller than a given threshold (in the current implementation $\epsilon < 0.001$); relative registration error is defined as: $\epsilon = |E_t - E_{t-1}|/E_t$, where E_t refers to the registration error between the model and data set at iteration t . The registration error is used as a quantitative value for the comparisons and it is computed by accumulating the residual error, in a point wise manner, from data set to a *reference model set*. The reference model set corresponds to a highly detailed description of the model set (it contains on average ten times the number of points in the model set). Residual errors are computed by finding the nearest point in between the registered data set and the reference model set. Table 6.2 depicts the number of points in the data set (N_d), the number of points in the model set (N_m), the registration error (*Error*) and the number of iterations (#*Itr*) for all the algorithms tested during the comparisons. It should be highlighted that the proposed approach reaches the best registration in the lowest number of iterations.

Finally, the proposed approach has been evaluated using 3D shapes [52]. Figure 6.4 presents three examples of model sets, together with their corresponding deformed data sets, which were registered with the proposed approach (SD-FFD). Data sets were obtained by deforming the given model sets. In the case of the hand Fig. 6.4(*top*), the data set was obtained by opening the model set using a Laplacian deformation [128]; the data set corresponding to the eight-like shape Fig. 6.4(*middle*) has been obtained by twisting and deforming the model set's shape; finally, in the case of *Bunny* [51], the data set corresponds to a Laplacian deformation that moves down both ears and several distortions of body's parts from the model set (mainly on the back side). Figure 6.4(*a*) shows the initial configurations where data sets are rotated and translated from the model set, in addition to the deformations mentioned above. Intermediate results, obtained with $\lambda = 10^4$ and $\lambda = 10^2$ are presented in Fig. 6.4(*b*) and

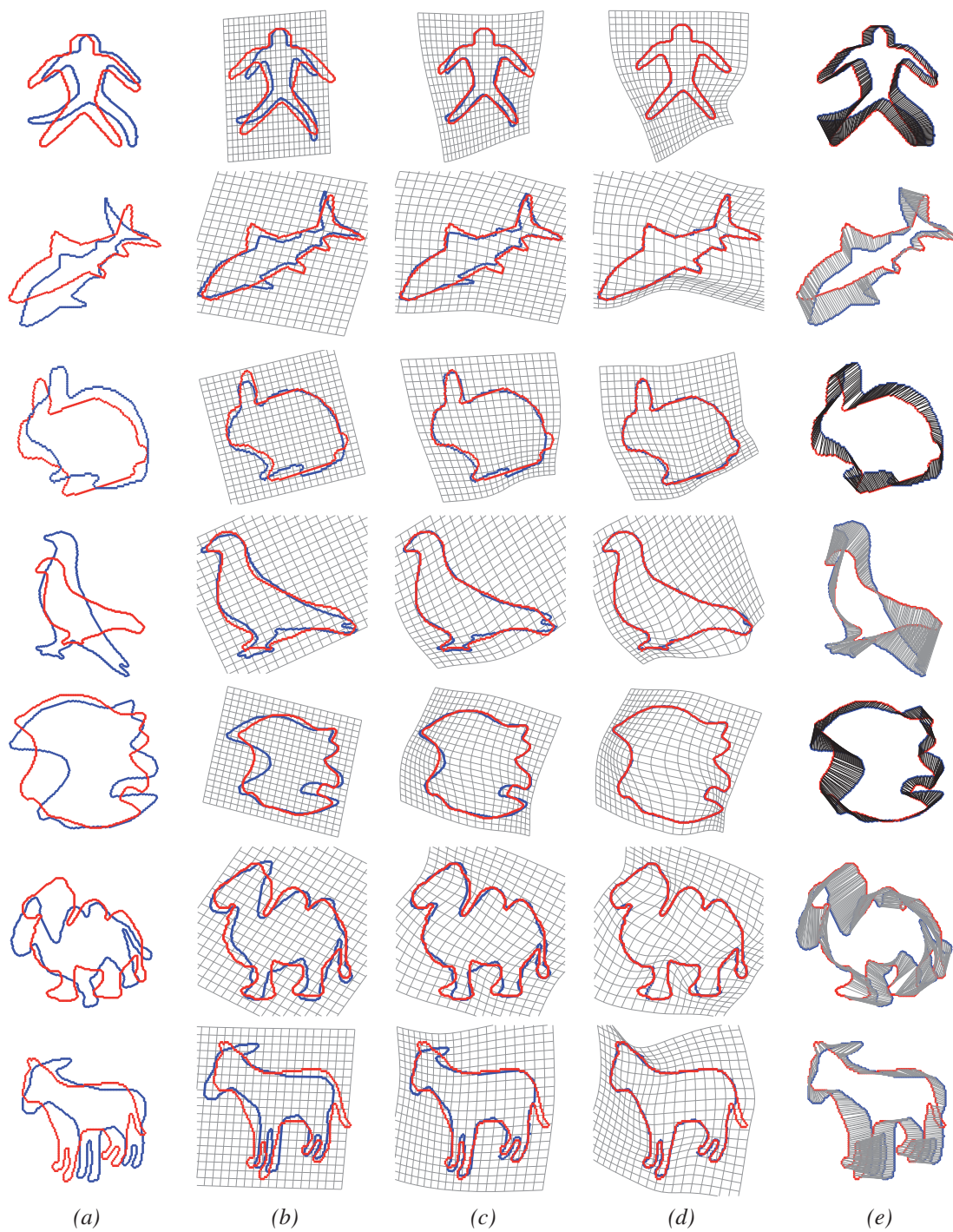


Figure 6.3: Registration results of 2D shapes ([52], [127]) using the proposed approach (SD-FFD) (a) initial configurations of data sets (blue) and model sets (red); (b) results with $\lambda = 10^5$; (c) results with $\lambda = 10^2$; (d) results with $\lambda = 1$; (e) the optimal FFD control lattice results in a very dense correspondence, not only on the boundary but over the whole space.

Table 6.2: Comparisons of non-rigid shape registration algorithms.

Figure	Huang et al. [58]		Fujiwara et al. [39]		Comp.: ICP-FFD		Prop. App.: SD-FFD	
	<i>Error</i>	<i>#Itr</i>	<i>Error</i>	<i>#Itr</i>	<i>Error</i>	<i>#Itr</i>	<i>Error</i>	<i>#Itr</i>
Fig. 6.1	2.15	47	3.77	42	1.84	28	1.67	21
Fig. 6.3(1 st row)	3.31	41	2.23	39	2.44	27	1.89	15
Fig. 6.3(2 nd row)	2.45	35	3.80	35	1.25	33	1.22	18
Fig. 6.3(3 rd row)	3.51	42	3.77	38	1.50	31	1.11	21
Fig. 6.3(4 th row)	2.49	49	2.95	43	1.82	38	1.72	28
Fig. 6.3(5 th row)	4.65	41	4.09	35	1.46	42	1.37	32
Fig. 6.3(5 th row)	3.73	50	2.98	48	1.55	29	1.26	21
Fig. 6.3(6 th row)	3.62	48	3.45	45	3.32	35	2.65	35
Fig. 6.3(7 th row)	4.16	50	3.54	41	3.34	45	2.15	39

Fig. 6.4(c) respectively. Final registration results are depicted in Fig. 6.4(d). The accuracy of the registration from the proposed approach (SD-FFD) can be appreciated from the blending of the two surfaces.

6.4 Conclusions

This chapter presents a novel formulation to tackle the non-rigid shape registration problem. It is based on both a quadratic estimation term, which measures the registration distance, and a quadratic regularization term, which controls the deformation of the data set towards the model set. The whole formulation can be solved in a single least squares framework. In summary, in this work: (i) a robust distance approximation based on local curvature information is used for non-rigid registration; (ii) the proposed objective function can be solved by a linear system of equations; (iii) all deformation from rigid to non-rigid are captured by the same framework; there is no need to use different steps to capture global and local deformations separately. Experimental results and comparisons with challenging 2D and 3D shapes are provided showing the validity of the proposed approach as well as the speed of convergence.

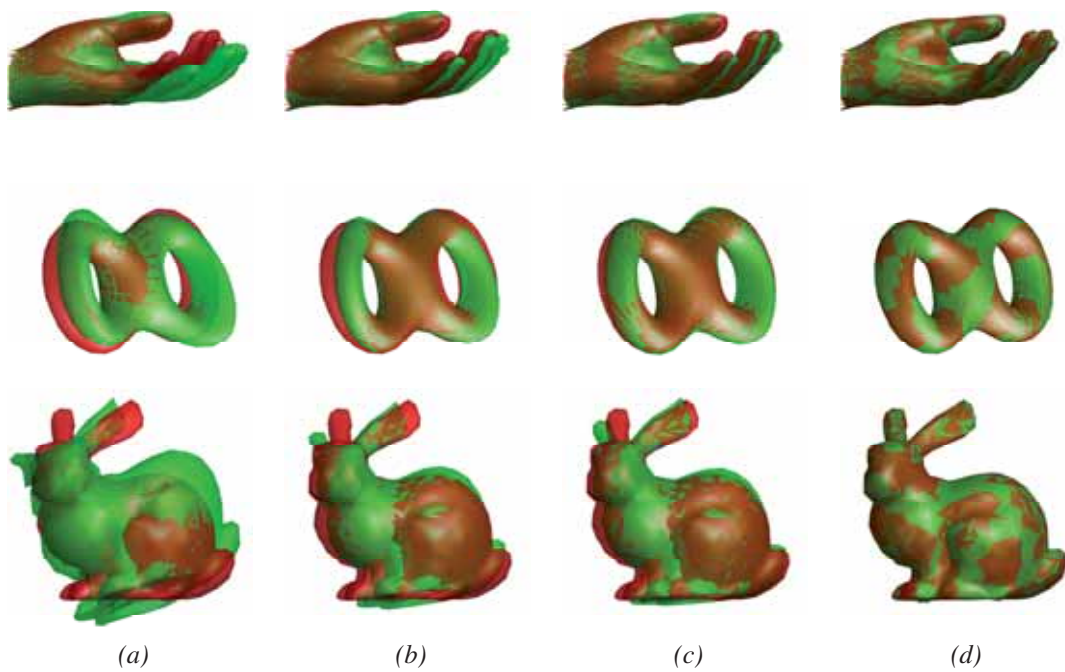


Figure 6.4: Registration results of 3D shapes using the proposed approach (SD-FFD). (a) initial configurations of data sets (green) and model sets (red); (b) results with $\lambda = 10^4$; (c) results with $\lambda = 10^2$; (d) Final registration results showing the blending of the two shapes (data & model).

Chapter 7

Conclusions

In this thesis we review and contribute in two important problems in computer vision and graphics. In the first part of the thesis, shape representation is studied from a general perspective, and state-of-the-art on this field has been reviewed and classified properly. Then, we contribute in both geometric and algebraic approaches for curve/surface fitting. In the second part, shape registration is tackled, and its state-of-the-art has been classified. Our contributions in this problem cast both rigid and non-rigid registration. In this chapter a summary of the thesis and our contributions are presented.

7.1 Summary

This thesis reviews state-of-the-art in two important problems in computer vision. In Chapter 2 the most relevant work in both shape representation and registration has been provided in two sections. In the first section shape representation techniques are summarized. We classify these methods based on the solution space they define and the fitting methodology they use. Solution space contains the set of parameters that define the curve/surface. In our review different solution space including implicit polynomials, implicit B-splines and radial basis functions have been considered. Implicit polynomials and implicit B-splines have been widely used during this thesis due to their simplicity, compactness and flexibility. Afterwards, different fitting methodologies, including both geometric and algebraic approaches are presented. Orthogonal distance and PCA based distance approximation are some examples of the first type, while the 3L and gradient-one algorithms, moving least squares and distance field based fitting are among the second type of fitting method.

In the second part of Chapter 2, state-of-the-art on shape registration is reviewed and classified based on the deformation models and the registration distances they apply. Deformation models can be simply chosen as rigid body motion or affine transformation. More elaborated models include those define non-rigid deformations. These models are classified into intrinsic and extrinsic ones. Intrinsic deformation only apply on the curve/surface being studied, while the extrinsic ones deform the whole region/space the shape is located on. We

explain different intrinsic deformation models including Laplacian deformation, shape evolution and those using variation implicit functions. The extrinsic deformation models include free form deformation, thin plate splines and those based on mean value coordinates or green coordinates. During this thesis rigid transformation and free form deformation have been used in Chapter 6 and 7 to describe the rigid and non-rigid deformations respectively.

In addition to the deformation models, different distance metrics measuring the registration error are reviewed and briefly explained. These methods are classified into coarse and fine registration techniques. PCA based, shape context, spin image and volume integral descriptors are instances of common tools in coarse registration. In fine registration, we review point-to-point registration distances including the iterative closest point, point-to-tangent, quadratic distance approximation and robust point matching. Probabilistic models are also reviewed as a widely used techniques in point set registration. This class includes the methods using gaussian mixture models, kernel correlation and coherent point drifts.

Chapter 3 of the thesis tackle the implicit function reconstruction using geometric distances. In this chapter a novel distance estimation is proposed to measure the closeness of a point to the implicit curve/surface. First a simplex (triangle/tetrahedron) is constructed between the point and implicit curve/surface. It is made by moving from the point along the coordinate axes till intersecting the zero set. Then, the height of this simplex is chosen as an estimation of the point to zero set distance. This distance estimation is quite fast and accurate, and can be in optimization algorithm to find the optimal curve/surface parameters. As a first application we exploit this distance estimation in a RANSAC based approach to find the best quadratic parameters describing the given set of points. The proposed algorithm picks enough number of points randomly to interpolate the quadratic surface, then this distance estimation is used to to measure the quality of this interpolation.

The proposed distance estimation has been also used in a more elaborated optimization algorithm to find the best surface parameters. This estimation approximates the distance between each point and the implicit surface. So, the final estimation is related to the implicit surface parameters. This relationship, as formulated in Chapter 3, provides analytical information about the sensitivity of the proposed distance with respect to the surface parameters. This sensitivity information can be provided for each point in the given cloud of points. Considering all the points and all the surface parameters, this sensitivity information construct the Jacobian matrix that is defined for non-linear leas squares form. Then, Levenberg-Marquadt algorithm, as detailed in Chapter 3, is applied to update the surface parameters to reach a better description. These two steps, distance estimation and surface parameter refinement, are iterated till convergence is reached.

Geometric distance estimation, though accurate, is still computationally expensive since it requires distance estimation for every single point at each iteration. Chapter 4 focuses on algebraic fitting approaches that are quite fast for curve/surface reconstruction. As a first contribution in this approach a relaxing technique is proposed to improve the results of the 3L algorithm. The 3L algorithm uses three levels including the data and its inside and outside offsets, where the optimal implicit function must obtain zero at the data points, $+\epsilon$ inside and $-\epsilon$ outside. The optimal function satisfying this system of equations can be found using the least squares solution. The additional levels, though help the stability of the least squares solution, are somehow restrictive. In our proposed technique we use local information in

every data points to refine the values of ϵ inside and outside of the set.

In the second part of Chapter 4 a more practical solution space is used for surface reconstruction. Implicit B-splines provide a flexible representation that benefit from some properties like smoothness and local control. We reformulate the 3L algorithm for implicit B-splines through a simple patch-based approach. The optimal surface parameter is still obtained using least squares formulation. Hence, the proposed method is quite fast and more practical since it use IBS with controlled regularity. Moreover, the proposed 3L-IBS has been efficiently implemented by considering only the active part of control lattice. The regularization term that controls the regularity of the IBS is also refined to cover the active control values.

Chapter 5 focuses on a different problem: shape registration. In this chapter a novel method for registering two clouds of points is presented. As an application of the shape registration in previous chapters, we solve the rigid registration problem using implicit functions to facilitate the distance computation during registration. In fact, the model set is replaced with an implicit function fitting this cloud of points. Then, instead of computing the point-to-point distance a point-to-model distance can be computed. As studied in the previous chapters, using an implicit interface induces many fast distance estimations. In Chapter 5 we reformulate the registration error based on one of these fast estimations. The final error is in the non-linear least squares form, and LMA can be used for optimization as a consequence. In the experimental results we used both implicit polynomials and implicit B-splines as interface, and it comes out that even using a rough representation can handle the rigid registration.

Finally in Chapter 6 non-rigid registration between two clouds of points is tackled. During the registration the data set is transformed based on a free form deformation, referred to as FFD, that belongs to extrinsic deformation models. This deformation is locally controlled by a control lattice, and its rigidity can be controlled by a tension term that is quadratic with respect to control parameters. Regarding the registration error, a quadratic distance approximation has been used. This approximation is based on the normal and curvature information estimated at each foot-point. As a novel contribution we use this distance approximation to find the optimal FFD control lattice. Since both the tension term and the registration error are in quadratic forms, our final formulation can be solved using least squares solution. Hence, we end up with a fast non-rigid registration method that is quite accurate due to the quadratic distance approximation it uses.

7.2 Contributions

Our contributions have been proposed to the community in two different problems: shape representation and shape registration. The main contribution and the outcome can be summarized as indicated bellow:

- **Geometric distance estimation:** A novel technique to estimate the distance of a point to the implicit curve/surface is proposed in Chapter 3. In order to compute this distance a simplex must be constructed first. Then, the height of this simplex is chosen as an estimation of the real distance. This distance estimation is accurate enough and can be computed quite

fast. The accuracy of this estimation as well as its relationship with the previous methods has been studied in [102]. Moreover, we exploit this estimation in a RANSAC based method to find the best implicit surface describing the given cloud of points [117].

- **Geometric surface fitting:** The proposed distance estimation provides us with more information that can be used to analytically compute the distance sensitivity with respect to the curve/surface parameters. This information, encoded in the Jacobian matrix, can feed any gradient based optimization. We apply this information in the Levenberg-Marquadt algorithm to find the best quadrics [103]. Then, we extend this method for general implicit polynomial fitting [104].

- **Relaxed 3L-IP algorithm:** Our contribution also cast the algebraic approaches for implicit fitting. The 3L algorithm has been adapted in [105] in order to relax the additional constraints imposed on the inner and outer offsets. Moreover, a criterion is presented to check out the consistency of normals and gradients during the process. The obtained method outperforms the original 3L algorithm, while it still belongs to the algebraic fitting class.

- **3L-IBS algorithm:** As a more practical representation, we use implicit B-splines for surface reconstructions. These functions inheritate the smoothness and local control properties of parametric B-splines, while do not require any parametrization. We propose a new algebraic fitting method, referred to as 3L-IBS, to find the optimal IBS parameters using the 3L algorithm [107]. Moreover, this method has been accelerated by considering only the active part of control lattice.

- **Rigid registration:** In a different work, we tackle shape registration, another main problem in computer vision. In our proposed technique, the model set of points is replaced with an implicit function that describes the model set. Hence, the point-to-point registration is solved as a point-to-model registration problem. Then, we exploit a fast distance estimation to propose a fast registration method that does not require any correspondence [106]. In our experiments we apply both implicit B-splines and polynomials to show the independence to the representation.

- **Non-rigid registration:** As our last contribution, the non-rigid registration problem is considered. Choosing FFD as a deformation model, we use a quadratic distance approximation that is based on the curvature information. Then, a non-rigid registration method is proposed to find the optimal FFD parameters. The optimization step is quite fast since the whole formulation is in the linear least squares form. The experimental results show the benefits of incorporating the curvature information during non-rigid registration.

7.3 Future work

In this thesis two different problems have been tackled and their relationship has been studied as well. Our contributions include surface reconstruction techniques (both algebraic and geometric) and shape registration (both rigid and non-rigid). Our future work comprises short-term challenges and long-term goals.

Surface reconstruction: the reconstruction techniques proposed in this thesis include both geometric and algebraic approaches. In the first category we proposed a fast distance approx-

imation that is quite accurate, and in the second category we presented an implicit B-spline fitting that is quite fast. The flexibility and power of implicit B-spline is proved in computer graphics and vision. As a short term challenge we apply the geometric distance approximation to improve the algebraic implicit B-spline fitting results.

Non-rigid registration: the proposed method in Chapter 7 can be improved to speed up the registration process. Since the distance approximation needs the curvature information in each foot-point in the model set, and this model set is not moving, the distance approximation can be computed before running the registration algorithm. As another short term task, the whole distance field can be computed in different area using a quad tree structure. Then, the distance field can be precomputed roughly for the area far from the model set, and more accurately for the area close to the model set.

Single-view reconstruction: the implicit B-spline used in this thesis provide a powerful tool to describe the real object. We managed to use this tool for reconstructing a surface from a cloud of points. Moreover, this representation is equipped with a quadratic regularization term that can be exploited to reconstruct a smooth surface when the data is sparse are partially missed. As a long term plan we study the possibility of using this representation to reconstruct the surface from a single view.

Object recognition: in the second part of the thesis an application of the shape representation has been explored in shape registration. The main idea was to use a high level representation in order not to work in the point level. Implicit functions provide compact representations that handle noise, outliers and missing data. We believe that these properties of implicit functions can be exploited in object recognition both in 2D and 3D.

Bibliography

- [1] D. Adi, S. Shamsuddin, and A. Ali. Particle swarm optimization for nurbs curve fitting. In *Proc. of the IEEE Int. Conf. on Computer Graphics, Imaging and Visualization*, pages 259–263, Tianjin, China, August 2009.
- [2] S. Ahn, W. Rauh, H. Cho, and H. Warnecke. Orthogonal distance fitting of implicit curves and surfaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):620–638, May 2002.
- [3] M. Aigner and B. Jutler. Gauss-newton-type technique for robustly fitting implicit defined curves and surfaces to unorganized data points. *IEEE International Conference on Shape Modelling and Application*, pages 121–130, 2008.
- [4] T. Albrecht, M. Lüthi, and T. Vetter. A statistical deformation prior for non-rigid image and shape registration. In *IEEE Conference on Computer Vision and Pattern Recognition*, Alaska, USA, 2008.
- [5] M. Alexa, J. Behr, D. Cohen-Or, S Fleishman, D. Levin, and T. Silva. Point set surfaces. In *IEEE Visualization*, pages 21–28, 2001.
- [6] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid icp algorithms for surface registration. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
- [7] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.
- [8] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.
- [9] H. Ben-Yaacov, D. Malah, and M. Barzohar. Recognition of 3d objects based on implicit polynomials. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(5):954–960, May 2010.
- [10] P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [11] M. Blane, Z. Lei, H. Civil, and D. Cooper. The 3L algorithm for fitting implicit polynomials curves and surface to data. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(3):298–313, March 2000.

- [12] F. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.
- [13] F. Boughorbel, M. Mercimek, A. Koschan, and M. Abidi. A new method for the registration of three-dimensional point-sets: The Gaussian fields framework. *Image and Vision Computing*, 28(1):124–137, 2010.
- [14] D. Breitenreicher and C. Schnörr. Robust 3d object registration without explicit correspondence using geometric integration. *Mach. Vis. Appl.*, 21(5):601–611, 2010.
- [15] A.M. Bronstein, M.M. Bronstein, and R. Kimmel. Topology-invariant similarity of nonrigid shapes. *International Journal on Computer Vision*, 81(3):281–301, 2009.
- [16] B. Brown and S. Rusinkiewicz. Global non-rigid alignment of 3-D scans. *ACM Trans. on Graphics*, 26(3):21, 2007.
- [17] J.C. Carr, R.K. Beatson, J.B. Cherrie, T.J. Mitchell, W.R. Fright, and B.C. McCallum. Reconstruction and representation of 3d objects with radial basis functions. *SIG-GRAPH*, pages 67–76, 2001.
- [18] A-L Chauve, P. Labatut, and J-P Pons. Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1261–1268, 2010.
- [19] Y. Chen and C. Liu. Quadric surface extraction using genetic algorithms. *Computer Aided Design*, 31(2):101–110, 1999.
- [20] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992.
- [21] K. Cheng, W. Wang, H. Qin, K. Wong, H. Yang, and Y. Liu. Fitting subdivision surfaces to unorganized point data using sdm. In *Pacific Conference on Computer Graphics and Applications*, pages 16–24, 2004.
- [22] Z-Q Cheng, W. Jiang, G. Dang, R.R. Martin, J. Li, H. Li, Y. Chen, Y. Wang, B. Li, K. Xu, and S. Jin. Non-rigid registration in 3d implicit vector space. In *IEEE International Conference on Shape Modeling and Applications*, pages 37–46, 2010.
- [23] D. Chetverikov, D. Stepanov, and P. Krsek. Robust euclidean alignment of 3d point sets: the trimmed iterative closest point algorithm. *Image and Vision Computing*, 23(3):299–309, 2005.
- [24] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2-3):114–141, 2003.
- [25] P. Claes, D. Vandermeulen, L. Van Gool, and P. Suetens. Robust and accurate partial surface registration based on variational implicit surfaces for automatic 3d model building. In *IEEE International Conference on on 3-D Digital Imaging and Modeling*, pages 385–392, 2005.

- [26] S. Coquillart. Extended free-form deformation: a sculpturing tool for 3d geometric modeling. In *SIGGRAPH*, pages 187–196, 1990.
- [27] D. Craciun, N. Paparoditis, and F. Schmitt. Multi-view scans alignment for 3d spherical mosaicing in large-scale unstructured environments. *Computer Vision and Image Understanding*, 114(11):1248–1263, 2010.
- [28] H. Dinh, G. Turk, and G. Slabaugh. Reconstructing surfaces by volumetric regularization using radial basis functions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(10):1358–1371, 2002.
- [29] H. El Munim and A. Farag. Shape representation and registration using vector distance functions. In *IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, Minnesota, USA, 2007.
- [30] P. Faber and R. Fisher. Pros and cons of euclidean fitting. In *DAGM*, pages 414–420, London, UK, 2001. Springer-Verlag.
- [31] R. Feichtinger, M. Fuchs, B. Jüttler, O. Scherzer, and Huaiping Yang. Dual evolution of planar parametric spline curves and t-spline level sets. *Computer-Aided Design*, 40(1):13–24, 2008.
- [32] P.F. Felzenszwalb. Representation and detection of deformable shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(2):208–220, 2005.
- [33] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Graphics and Image Processing*, 24(6):381–395, June 1981.
- [34] A. Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and Vision Computing*, 21(13-14):1145–1153, 2001.
- [35] A. Fitzgibbon, M. Pilu, and R. Fisher. Direct least square fitting of ellipses. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(5):476–480, May 1999.
- [36] R. Fletcher. *Practical Methods of Optimization*. New York: Wiley, 1990.
- [37] S. Flöry and M. Hofer. Surface fitting and registration of point clouds using approximations of the unsigned distance function. *Computer Aided Geometric Design*, 27(1):60–77, 2010.
- [38] M. Fuchs, B. Jüttler, O. Scherzer, and H. Yang. Shape metrics based on elastic deformations. *Journal of Mathematical Imaging and Vision*, 35(1):86–102, 2009.
- [39] K. Fujiwara, K. Nishino, J. Takamatsu, B. Zheng, and K. Ikeuchi. Locally rigid globally non-rigid surface registration. In *International Conference on Computer Vision*, pages 1527–1534, Barcelona, Spain, 2011.
- [40] J. Gall, C. Stoll, E. de Aguiar, Theobalt, B. Rosenhahn, and H-P. Seidel. Motion capture using joint skeleton tracking and surface estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1746–1753, 2009.

- [41] N. Gelfand, N.J. Mitra, L.J. Guibas, and H. Pottmann. Robust global registration. In *Symposium on Geometry Processing*, pages 197–206, 2005.
- [42] P. Gotardo, O. Bellon, K. Boyer, and L. Silva. Range image segmentation into planar and quadric surfaces using an improved robust estimator and genetic algorithm. *IEEE Trans. on Systems, Man, and Cybernetics Part B: Cybernetics*, 34(6):2303–2316, 2004.
- [43] G. Guennebaud and M. Gross. Algebraic point set surfaces. *ACM Trans. on Graphics*, 26(3):23, 2007.
- [44] H. Helfrich and D. Zwick. A trust region algorithm for parametric curve and surface fitting. *Journal of Computational and Applied Mathematics*, 73(1-2):119–134, October 1996.
- [45] A. Helzer, M. Barzohar, and D. Malah. Stable fitting of 2d curves and 3d surfaces by implicit polynomials. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(10):1283–1294, October 2004.
- [46] V. Hiep, R. Keriven, P. Labatut, and J-P. Pons. Towards high-resolution large-scale multi-view stereo. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1430–1437, 2009.
- [47] J. Ho, A. Peter, A. Rangarajan, and M. Yang. An algebraic approach to affine registration of point sets, 2009. International Conference on Computer Vision.
- [48] S. Holzer, S. Hinterstoisser, S. Ilic, and N. Navab. Distance transform templates for object detection and pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1177–1184, 2009.
- [49] Hugues Hoppe, Tony DeRose, Tom Duchamp, John Alan McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *Proc. of the 19th Annual Conf. on Computer Graphics and Interactive Techniques, SIGGRAPH*, pages 71–78, Chicago, IL, USA, July 1992.
- [50] R. Horaud, F. Forbes, M. Yguel, G. Dewaele, and J. Zhang. Rigid and articulated point registration with expectation conditional maximization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(3):587–602, 2011.
- [51] <http://graphics.stanford.edu/data/3Dscanrep/>. The Stanford 3D Scanning Repository.
- [52] <http://shapes.aimatshape.net/>. AIM@SHAPE, Digital Shape WorkBench.
- [53] H. Huang, D. Li, H. Zhang, U. Ascher, and D. Cohen-Or. Consolidation of unorganized point clouds for surface reconstruction. *ACM Trans. on Graphics*, 28(5), 2009.
- [54] Q. Huang, B. Adams, and M. Wand. Bayesian surface reconstruction via iterative scan alignment to an optimized prototype. In *Symposium on Geometry Processing*, pages 213–223, 2007.
- [55] Q-X. Huang, B. Adams, M. Wicke, and L.J. Guibas. Non-rigid registration under isometric deformations. *Computer Graphics Forum*, 27(5):1449–1457, 2008.

- [56] Q-X. Huang, S. Flöry, N. Gelfand, M. Hofer, and H. Pottmann. Reassembling fractured objects by geometric matching. *ACM Trans. on Graphics*, 25(3):569–578, 2006.
- [57] X. Huang, D. Metaxas, and T. Chen. Metamorphs: Deformable shape and texture models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 496–503, 2004.
- [58] X. Huang, N. Paragios, and D. Metaxas. Shape registration in implicit spaces using information theory and free form deformations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(8):1303–1318, August 2006.
- [59] D.F. Huber and M. Hebert. Fully automatic registration of multiple 3d data sets. *Image and Vision Computing*, 21(7):637–650, 2003.
- [60] T. Igarashi, T. Moscovich, and J.F. Hughes. As-rigid-as-possible shape manipulation. *ACM Trans. on Graphics*, 24(3):1134–1141, 2005.
- [61] B. Jian and B. Vemuri. A robust algorithm for point set registration using mixture of Gaussians. In *International Conference on Computer Vision*, pages 1246–1251, Beijing, China, 2005.
- [62] B. Jian and B. Vemuri. Robust point set registration using gaussian mixture models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(8):1633–1645, 2011.
- [63] A.E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.
- [64] P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki. Harmonic coordinates for character articulation. *ACM Trans. on Graphics*, 26(3):71, 2007.
- [65] S.H. Joshi, E. Klassen, A. Srivastava, and I. Jermyn. A novel representation for riemannian analysis of elastic curves in \mathbb{R}^n . In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
- [66] T. Ju, S. Schaefer, and J.D. Warren. Mean value coordinates for closed triangular meshes. *ACM Trans. on Graphics*, 24(3):561–566, 2005.
- [67] B. Jüttler. Least-squares fitting of algebraic spline curves via normal vector estimation. In *IMA Conference on the Mathematics of Surfaces*, pages 263–280, 2000.
- [68] B. Jüttler and A. Felis. Least-squares fitting of algebraic spline surfaces. *Adv. Comput. Math.*, 17(1-2):135–152, 2002.
- [69] Michael M. Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Symposium on Geometry Processing*, pages 61–70, 2006.
- [70] M. Kilian, N.J. Mitra, and H. Pottmann. Geometric modeling in shape space. *ACM Trans. on Graphics*, 26(3):64, 2007.

- [71] E. Klassen, A. Srivastava, W. Mio, and S.H. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(3):372–383, 2003.
- [72] S. Klein, J. Pluim, M. Staring, and M. Viergever. Adaptive stochastic gradient descent optimisation for image registration. *International Journal on Computer Vision*, 81(3):227–239, 2009.
- [73] S. Kurttek, E. Klassen, Z. Ding, and A. Srivastava. A novel riemannian framework for shape analysis of 3d objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1632, 2010.
- [74] P. Labatut, J-P. Pons, and R. Keriven. Robust and efficient surface reconstruction from range data. *Comput. Graph. Forum*, 28(8):2275–2290, 2009.
- [75] T-Y. Lee and S-H. Lai. 3d non-rigid registration for mpu implicit surfaces. In *Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2008.
- [76] V. Lempitsky. Surface extraction from binary volumes with higher-order smoothness. In *CVPR*, pages 1197–1204, 2010.
- [77] V. Lempitsky and Y. Boykov. Global optimization for shape fitting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.
- [78] H. Li and R. Hartley. The 3d-3d registration problem revisited. In *International Conference on Computer Vision*, pages 1–8, 2007.
- [79] H. Li, T. Shen, and X. Huang. Global optimization for alignment of generalized shapes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 856–863, Miami, USA, 2009.
- [80] Y. Li, Q. Zheng, A. Sharf, D. Cohen-Or, B. Chen, and N.J. Mitra. 2d-3d fusion for layer decomposition of urban facades. In *International Conference on Computer Vision*, pages 882–889, 2011.
- [81] W. Lian, L. Zhang, Y. Liang, and Q. Pan. A quadratic programming based cluster correspondence projection algorithm for fast point matching. *Computer Vision and Image Understanding*, 114(3):322–333, 2010.
- [82] M. Liao, Q. Zhang, H. Wang, R. Yang, and M. Gong. Modeling deformable objects from a single depth camera. In *International Conference on Computer Vision*, pages 167–174, 2009.
- [83] Y. Lipman and T Funkhouser. Möbius voting for surface correspondence. *ACM Trans. on Graphics*, 28(3), 2009.
- [84] Y Lipman, D. Levin, and D. Cohen-Or. Green coordinates. *ACM Trans. on Graphics*, 27(3), 2008.
- [85] M-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa. Fast directional chamfer matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1696–1703, 2010.

- [86] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, pages 674–679, 1981.
- [87] G. Marola. A technique for finding the symmetry axes of implicit polynomial curves under perspective projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(3):465–470, 2005.
- [88] B. Mederos, M. Laje, S. Arouca, F. Petronetto, L. Velho, T. Lewiner, and H. Lopes. Regularized implicit surface reconstruction from points and normals. *J. Braz. Comp. Soc.*, 13(4):7–16, 2007.
- [89] A.S. Mian, M. Bennamoun, and R.A. Owens. A novel representation and feature matching algorithm for automatic pairwise registration of range images. *International Journal on Computer Vision*, 66(1):19–40, 2006.
- [90] F. Mokhtarian and A. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(8):789–805, 1992.
- [91] B. Morse, T. Yoo, D. Chen, P. Rheingans, and K. Subramanian. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In *Shape Modeling International*, pages 89–98, 2001.
- [92] P. Mullen, F. Goes, M. Desbrun, D. Cohen-Steiner, and P. Alliez. Signing the unsigned: Robust surface reconstruction from raw pointsets. *Comput. Graph. Forum*, 29(5):1733–1741, 2010.
- [93] A. Myronenko and X.B. Song. Point set registration: Coherent point drift. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010.
- [94] C. Oden, A. Ercil, and B. Buke. Combining implicit polynomials and geometric features for hand recognition. *Pattern Recognition Letters*, 24(13):2145–2152, 2003.
- [95] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H. Seidel. Multi-level partition of unity implicits. *ACM Trans. on Graphics*, 22(3):463–470, 2003.
- [96] C. Olsson, F. Kahl, and M. Oskarsson. Branch-and-bound methods for euclidean registration problems. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(5):783–794, 2009.
- [97] H. Pottmann and S. Leopoldseder. A concept for parametric surface fitting which avoids the parametrization problem. *Computer Aided Geometric Design*, 20(6):343–362, 2003.
- [98] H. Pottmann, S. Leopoldseder, and M. Hofer. Approximation with active b-spline curves and surfaces. In *Pacific Conference on Computer Graphics and Applications*, pages 8–25, 2002.
- [99] H. Pottmann, S. Leopoldseder, and M. Hofer. Registration without ICP. *Computer Vision and Image Understanding*, 95(1):54–71, 2004.

- [100] V. Pratt. Direct least-squares fitting of algebraic surfaces. In *SIGGRAPH*, pages 145–152, 1987.
- [101] A. Restrepo-Specht, A. Sappa, and M. Devy. Edge registration versus triangular mesh registration, a comparative study. *Signal Processing: Image Communication*, 20(9–10):853–868, November 2005.
- [102] M. Rouhani and A. Sappa. Implicit polynomial representation through a fast fitting error estimation. *IEEE Trans. on Image Processing*, 21(4):2089–2098, 2012.
- [103] M. Rouhani and A.D. Sappa. A novel approach to geometric fitting of implicit quadrics. In *Advanced Concepts for Intelligent Vision Systems*, pages 121–132, Bordeaux, France, Sep 2009.
- [104] M. Rouhani and A.D. Sappa. A fast accurate implicit polynomial fitting approach. In *IEEE International Conference on Image Processing*, pages 1429–1432, 2010.
- [105] M. Rouhani and A.D. Sappa. Relaxing the 3L algorithm for an accurate implicit polynomial fitting. In *IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, USA, June 2010.
- [106] M. Rouhani and A.D. Sappa. Correspondence free registration through a point-to-model distance minimization. In *International Conference on Computer Vision*, pages 2150–2157, 2011.
- [107] M. Rouhani and A.D. Sappa. Implicit b-spline fitting using the 3l algorithm. In *IEEE International Conference on Image Processing*, pages 893–896, 2011.
- [108] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *IEEE International Conference on on 3-D Digital Imaging and Modeling*, Quebec, Canada, 2001.
- [109] Y. Sahillioglu and Y. Yemez. 3d shape correspondence by isometry-driven greedy optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 453–458, 2010.
- [110] T. Sahin and M. Unel. Fitting globally stabilized algebraic surfaces to range data. In *International Conference on Computer Vision*, pages 1083–1088, Washington, DC, USA, 2005. IEEE Computer Society.
- [111] J. Salvi, C. Matabosch, D. Fofi, and J. Forest. A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing*, 25(5):578–596, 2007.
- [112] M. Salzmann and P. Fua. Linear local models for monocular reconstruction of deformable surfaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(5):931–944, 2011.
- [113] M. Salzmann, J. Pilet, S. Ilic, and P. Fua. Surface deformation models for non-rigid 3d shape recovery. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(8):1481–1487, 2007.

- [114] M. Samozino, M. Alexa, P. Alliez, and M. Yvinec. Reconstruction with voronoi centered radial basis functions. In *Symposium on Geometry Processing*, pages 51–60, 2006.
- [115] P D Sampson. Fitting conic sections to very scattered data: An iterative refinement of the bookstein algorithm. *Computer Graphics and Image Processing*, 18:97–108, 1982.
- [116] R. Sandhu, S. Dambreville, and A. Tannenbaum. Point set registration via particle filtering and stochastic dynamics. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(8):1459–1473, 2010.
- [117] A.D. Sappa and M. Rouhani. Efficient distance estimation for fitting implicit quadric surfaces. In *IEEE International Conference on Image Processing*, pages 3521–3524, Cairo, Egypt, November 2009.
- [118] S. Schaefer, T. McPhail, and J.D. Warren. Image deformation using moving least squares. *ACM Trans. on Graphics*, 25(3):533–540, 2006.
- [119] F.R. Schmidt, D. Farin, and D. Cremers. Fast matching of planar shapes in sub-cubic runtime. In *International Conference on Computer Vision*, pages 1–6, 2007.
- [120] J. Schnabel, D. Rueckert, M. Quist, J. Blackall, A. Castellano-Smith, T. Hartkens, G. Penney, W. Hall, H. Liu, C. Truwit, F. Gerritsen, D. Hill, and D. Hawkes. A generic framework for non-rigid registration based on non-uniform multi-level free-form deformations. In *MICCAI*, pages 573–581, 2001.
- [121] S. Schuon, C. Theobalt, J. Davis, and S. Thrun. Lidarboost: Depth superresolution for tof 3d shape scanning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 343–350, 2009.
- [122] T.B. Sebastian, P.N. Klein, and B.B. Kimia. On aligning curves. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(1):116–125, 2003.
- [123] T.W. Sederberg and S.R. Parry. Free-form deformation of solid geometric models. In *SIGGRAPH*, pages 151–160, 1986.
- [124] A. Sharf, T. Lewiner, G. Shklarski, S. Toledo, and D. Cohen-Or. Interactive topology-aware surface reconstruction. *ACM Trans. on Graphics*, 26(3):43, 2007.
- [125] G. Sharp, S. Lee, and D. Wehe. ICP registration using invariant features. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(1):90–102, 2002.
- [126] G. Sharp, S.W. Lee, and D.K. Wehe. Maximum-likelihood registration of range images with missing data. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(1):120–130, 2008.
- [127] D. Sharvit, J. Chan, H. Tek, and B Kimia. Symmetry-based indexing of image databases. *Journal of Visual Communication and Image Representation*, 9(4):366–380, 1998.

- [128] O. Sorkine. Differential representations for mesh processing. *Computer Graphics Forum*, 25(4):789–807, 2006.
- [129] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer, third edition, 2002.
- [130] Jean-Philippe Tarel, Hakan Civi, and David B. Cooper. Pose estimation of free-form 3d objects without point matching using algebraic surface models. In *Proceedings of IEEE Workshop Model Based 3D Image Analysis*, pages 13–21, Mumbai, India, 1998.
- [131] M. Taron, N. Paragios, and M. Jolly. Registration with uncertainties and statistical modeling of shapes with variable metric kernels. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(1):99–113, January 2009.
- [132] T. Tasdizen, J. Tarel, and D. Cooper. Algebraic curves that work better. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2035–2041, 1999.
- [133] T. Tasdizen, J. Tarel, and D. Cooper. Improving the stability of algebraic curves for applications. *IEEE Trans. on Image Processing*, 9(3):405–416, 2000.
- [134] G. Taubin. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(11):1115–1138, November 1991.
- [135] D. Terzopoulos and D. Metaxas. Dynamic 3d models with local and global deformations: Deformable superquadrics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(7):703–714, 1991.
- [136] A. Tevs, M. Bokeloh, M. Wand, A. Schilling, and H-P. Seidel. Isometric registration of ambiguous and partial data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1185–1192, 2009.
- [137] M. Trummer, H. Suesse, and J. Denzler. Coarse registration of 3d surface triangulations based on moment invariants with applications to object alignment and identification. In *International Conference on Computer Vision*, pages 1273–1279, 2009.
- [138] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In *European Conference on Computer Vision*, pages 558–569, 2004.
- [139] G. Turk and J.F. O’Brien. Shape transformation using variational implicit functions. In *SIGGRAPH*, pages 335–342, 1999.
- [140] M. Unel and W. Wolovich. Pose estimation and object identification using complex algebraic representations. *Pattern Anal. Appl.*, 1(3):178–188, 1998.
- [141] C. Unsalan. A model based approach for pose estimation and rotation invariant object matching. *Pattern Recognition Letters*, 28(1):49–57, 2007.
- [142] C. Walder, B. Schölkopf, and O. Chapelle. Implicit surfaces with globally regularised and compactly supported basis functions. In *NIPS*, pages 273–280, 2006.

- [143] H. Wang, Q. Zhang, B. Luo, and S. Wei. Robust mixture modelling using multivariate t-distribution with missing information. *Pattern Recognition Letters*, 25(6):701–710, 2004.
- [144] J. Wang and K. Chan. Shape evolution for rigid and nonrigid shape registration and recovery. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 164–171, Miami, USA, 2009.
- [145] W. Wang, H. Pottmann, and Y. Liu. Fitting b-spline curves to point clouds by curvature-based squared distance minimization. *ACM Trans. on Graphics*, 25(2):214–238, 2006.
- [146] Y. Watanabe, T. Komuro, and M. Ishikawa. High-resolution shape reconstruction from multiple range images based on simultaneous estimation of surface and motion. In *International Conference on Computer Vision*, pages 1787–1794, 2009.
- [147] T. Weise, B. Leibe, and L. Van Gool. Accurate and robust registration for in-hand modeling. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [148] H-Y. Wu, H. Zha, T. Luo, X. Wang, and S. Ma. Global and local isometry-invariant descriptor for 3d shape comparison and partial matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 438–445, 2010.
- [149] M-F. Wu and H-T. Sheu. Representation of 3d surfaces by two-variable fourier descriptors. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(8):858–863, 1998.
- [150] Z. Xie and G.E. Farin. Image registration using hierarchical b-splines. *IEEE Trans. Vis. Comput. Graph.*, 10(1):85–94, 2004.
- [151] R. Xu and M. Kemp. Fitting multiple connected ellipses to an image silhouette hierarchically. *IEEE Trans. on Image Processing*, 19(7):1673–1682, July 2010.
- [152] H. Yalcin, M. Unel, and W. Wolovich. Implicitization of parametric curves by matrix annihilation. *International Journal on Computer Vision*, 54(1-3):105–115, 2003.
- [153] Z. Yang, J. Deng, and F. Chen. Fitting unorganized point clouds with active implicit b-spline curves. *Visual Computing*, 21(1):831–839, 2005.
- [154] Y. Zeng, C. Wang, Y. Wang, X. Gu, D. Samaras, and N. Paragios. Dense non-rigid surface registration using high-order graph matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 382–389, 2010.
- [155] B. Zheng, R. Ishikawa, T. Oishi, J. Takamatsu, and K. Ikeuchi. A fast registration method using ip and its application to ultrasound image registration. In *IPSJ Transactions on Computer Vision and Applications*, pages 209–219, September 2009.
- [156] B. Zheng, R. Ishikawa, T. Oishi, J. Takamatsu, and K. Ikeuchi. A fast registration method using IP and its application to ultrasound image registration. *IPSJ Transactions on Computer Vision and Applications*, 1:209–219, 2009.

- [157] B. Zheng, J. Takamatsu, and K. Ikeuchi. 3d model segmentation and representation with implicit polynomials. *IEICE Transactions*, 91-D(4):1149–1158, 2008.
- [158] B. Zheng, J. Takamatsu, and K. Ikeuchi. An adaptive and stable method for fitting implicit polynomial curves and surface. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(3):561–568, March 2010.
- [159] Q. Zheng, A. Sharf, G. Wan, Y. Li, N.J. Mitra, D. Cohen-Or, and B. Chen. Non-local scan consolidation for 3d urban scenes. *ACM Trans. on Graphics*, 29(4), 2010.
- [160] T. Zinßer, J. Schmidt, and H. Niemann. A refined ICP algorithm for robust 3-d correspondence estimation. In *IEEE International Conference on Image Processing*, pages 695–698, 2003.

List of Publications

This dissertation has led to the following communications:

Journal Papers

- Rouhani M. and Sappa A., Implicit Polynomial Representation through a Fast Fitting Error Estimation, IEEE Transactions on Image Processing, April 2012, volume 21(4) pp 2089-2098.

Conference Papers

- Rouhani M. and Sappa A.D., Correspondence Free Registration through a Point-to-Model Distance Minimization, IEEE Int. Conf. on Computer Vision (ICCV), Barcelona, Spain, November 2011, pp. 2150-2157.
- Sappa A.D., Gernimo D., Dornaika F., Rouhani M. and Lopez A., Moving object detection from mobile platforms using stereo data registration, chapter in Computational Intelligence paradigms in advanced pattern classification, Ed. by R. Ogiela and Lakhmi C. Jain, Springer-Verlag's Book Series, 2012, pp. 25-37.
- Rouhani M. and Sappa A.D. "Implicit B-spline fitting using the 3L algorithm, IEEE Int. Conf. on Image Processing (ICIP), Brussels, Belgium, September 2011, pp. 893-896.
- Rouhani M. and Sappa A.D., A Fast Implicit Polynomial Fitting Approach, IEEE Int. Conf. on Image Processing (ICIP), Hong Kong, September 2010, pp. 1429-1432.
- Rouhani M. and Sappa A.D., Relaxing the 3L Algorithm for an Accurate Implicit Polynomial Fitting, IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR), San Francisco, June 2010, pp. 3066-3072.
- Sappa A.D. and Rouhani M., Efficient Distance Estimation for Fitting Implicit Quadric Surfaces, IEEE Int. Conf. on Image Processing (ICIP), Cairo, Egypt, November 7-11, 2009, pp. 3521-3524.
- Rouhani M. and Sappa A.D., A Novel Approach to Geometric Fitting of Implicit Quadrics, IEEE Int. Conf. on Advanced Concepts for Intelligent Vision Systems (Acivs), Bordeaux, France, September 2009; pp. 121-132.