A Variational Framework of Multivariate Splines and Its Applications

A Dissertation Presented

by

Kexiang Wang

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

August 2010

Stony Brook University

The Graduate School

Kexiang Wang

We, the dissertation committee for the above candidate for the degree of Doctor of Philosophy, hereby recommend acceptance of this dissertation.

Hong Qin, Dissertation Advisor
Professor, Department of Computer Science

Joe Mitchell, Chairperson of Defense
Professor, Department of Computer Science

Xiangmin Jiao
Professor, Department of Computer Science

Jeff Ge
Professor, Department of Mechanical Engineering

This dissertation is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

A Variational Framework of Multivariate Splines and Its Applications

by

Kexiang Wang

Doctor of Philosophy

in

Computer Science

Stony Brook University

2010

Multivariate spline technique has proved to be a powerful mathematical tool for solving variational problems in a great number of research and engineering tasks, such as computer vision, scientific computing, engineering design, etc. As present, tensor-product B-splines and NURBS are the prevailing industrial standards and have been widely used in different disciplines. There have been a few new multivariate spline techniques developed recently bestowed with unique and favorable features, e.g., triangular B-splines and manifold splines. However, their potentials in facilitating practical scientific and industrial applications have not yet been fully explored.

In this dissertation, we presented a variational framework built upon a range of newly proposed multivariate splines, and then applied it to solve a few research problems in medical imaging, scientific computing and geometric design. More specifically, we introduced a novel image registration method empowered by triangular B-splines, which is capable of modeling local rigidities inside a global non-rigid transformation. We also developed triangular B-spline finite element method (TBFEM) and solved an elastic problem on a pseudo breast model for temporal mammogram registration. Combining B-spline with feature detection and matching techniques, we proposed a registration algorithm that specifically registers mammogram images with little human interventions. In addition, we simulated elastic deformations on thin-shell objects with complicated geometries and arbitrary topologies, which are rigorously represented by manifold splines. Moreover, we proposed the new RTP-spline, a trivariate spline with restricted boundaries and defined over polycubic parametric domain. It is virtually a sub-class of trivariate T-splines, but constructed in a different top-down fashion such that semi-standardness can be preserved via knot insertion and blending function refinement. RTP-splines are featured with the ability of local refinement, restricted boundaries, domain flexibility and efficient evaluation of basis functions, all of which would greatly benefit a variety of applications working on solid objects and/or volumetric data.

Through extensive experiments, we demonstrated that while the unique and advantageous properties of those new multivariate splines are exploited and applied to appropriate applications, our proposed framework would turn into an effective and powerful tool for solving variational problems in many science and engineering areas.

This dissertation is dedicated to my wife, Sicong, and my children, Eric and Luna. I give my deepest expression of love and appreciation for the encouragement that you gave and the sacrifices you made during this graduate program.

Contents

List of Tables

List of Figures

Acknowledgements

I owe my deepest gratitude to my thesis advisor, Professor Hong Qin, whose guidance, encouragement, patience and support from the initial to the final level enabled me to fulfill this dissertation.

I would also like to thank my committee members Professor Joseph S.B. Mitchell, Professor Jeffrey Ge, and Professor Xiangmin Jiao for their very helpful insights, comments and suggestions as well as Professor Klaus Mueller, and Professor Xianfeng Gu for serving on my thesis proposal committee.

Additionally, I would like to acknowledge all of those people who provided support and assistance with my dissertation work: Jing Hua, Xiaohu Guo, Ying He, Bo Li, Xin Li, Hongyu Wang. Many thanks also go to Professor Wei Zhao, and MD. Paul R. Fisher for their enlightenment education in medical imaging.

Chapter 1

Introduction

With the advent of advancing data acquisition technologies and increasing computing power, it's more and more demanded by scientific and industrial applications for computers of nowadays to handle massive numerical data. Raw data obtained from the real-world are unlikely to be directly converted into analytical representations, therefore must be stored as sets of discrete data points. For example, image data acquired through CT-scan are stored on a regular grid, and scanned range data are converted to digital models in forms of point samples and/or triangular meshes, approximating real world surfaces. It's well known that discrete presentation method suffers from two major disadvantages: (1) taking too much memory/disk spaces for data handling, (2) resorting to numerical approximation via bilinear/trilinear interpolation for computing differential quantities. By converting large discrete dataset into splines, we will achieve a compact representation with much smaller data size as well as a convenient way to precisely compute differential quantities such as surface geodesics, curvatures and elastic-strain tensors. Rapid and accurate calculation of such differential properties is the key to facilitate the applications which requires to solve variational problems.

Tensor-product B-spline and NURBS are the prevailing mathematical tools used in a great number of research areas, such as geometry design, physics-based modeling, and computer vision. On the contrary, a few recently developed multivariate splines such as triangular B-splines and manifold splines have not been recognized by the researchers outside shape modeling communities. These new splines schemes are equipped with many unique and favorable features, which could potentially benefit solving of certain research and engineering problems. For instance, triangular B-splines are defined over triangular domain and allows locally modeling of sharp features in arbitrary orientation, so that it may serve as a fundamental tools for applications that have irregular problem domain and the necessity of handling regional discontinuities within globally continuous data. Manifold splines marry traditional planar spline schemes and recent surface parametrization technique, able to naturally model surfaces of complicated geometries and high-genus topologies as a single-piece spline function, which is impossible for conventional ones without performing cumbersome domain manipulation like patching and trimming. Such advantage gives users the freedom in calculation differential properties of manifold surfaces and make solving of variational problems over non-trivial manifold domain possible. Therefore, it's urged for us to carry out a research work on how to apply these new spline techniques effectively into research problems other than shape modeling. And we hope the results of this work would interest more people and extend the applications of these newly developed multivariate splines into a broader research and engineering area.

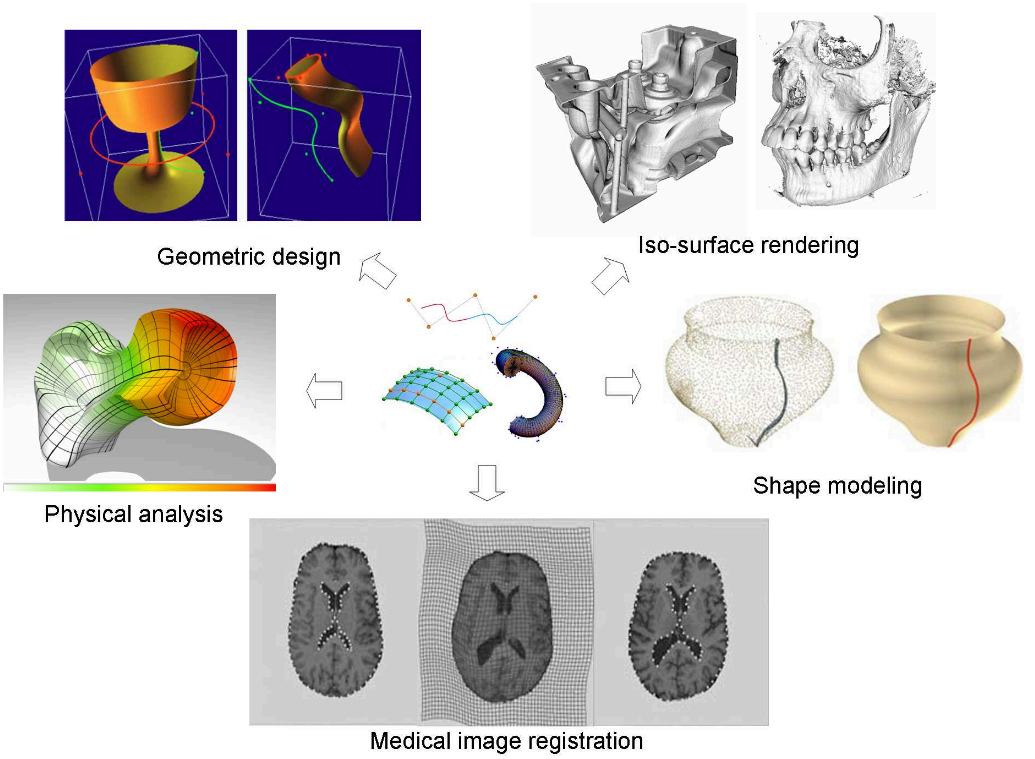


Figure 1.1: As popular mathematical tools, spline functions have been widely used in many research areas, for example, geometric design(courtesy of [97]), shape modeling(courtesy of [142]), physical analysis(courtesy of [74]), image registration(courtesy of [150]), and visualization(courtesy of [107])

1.1 Problem Statements

As mentioned above, unlike B-spline and NURBS being widely used by researchers and engineers, newly developed spline techniques such as triangular B-spline and manifold spline have not received enough attentions from people outside shape modeling community. However, these new splines are featured with unique and novel properties which would potentially facilitate many well-known applications. In this dissertation, we review a couple of multivariate spline techniques proposed recently, build a general variational framework upon it, and apply it to solve a series of practical problems. The applications attempted in this dissertation fall into the following categories.

1.1.1 Spline-based Feature-sensitive Image Registration

Image registration is the process of transforming different sets of data into one coordinate system. In medical image registration, the transformation is usually nonrigid to cope with the deformations of the subjects due to breathing, anatomical changes and so forth. To represent unknown transformation with tensor-product B-spline is a widely accepted practice in computer vision. Particularly, pre-identifying significant anatomical features, for instance nipples in mammograms, and matching them between source and reference images would substantially enhance the registration results.

Correct mammogram registration is critical for physicians to detect breast cancers at their early stages. Mammogram images are frequently containing excessive disordered texture features, therefore the optimization for intensity-based registration tends to get trapped in local minima, unlikely to yield satisfactory results. To achieve better registration results, features must be pre-segmented from mammograms and paired between source and target images. However, to pick up and match features from mammogram images manually is subjective and an extremely labor-intensive task. Therefore, it would be ideal to have an mammogram registration algorithm that can identify and align features automatically.

In medical images, features lines are usually found between rigid and soft anatomical structures, *e.g.*the interfaces between muscles and bones. The significance of these lines is that C^1 continuities could possibly vanish along them in the recovered transformation. As the transformation is oftentimes modeled by B-splines as a single nonrigid function, it's necessary to enforce C^1 discontinuities along features lines for accurate registration. Such discontinuities can be approximated by either incorporating a penalty term into variational registration framework as soft constraints, or applying boundary conditions along features lines as hard ones. Unless the transformation has such a spline representation that models C^1 continuities and discontinuities within the same framework, it's not possible to precisely recover C^1 discontinuities between rigid and soft anatomical structures.

1.1.2 Spline Thin-Shell Deformation

It's well-known that the deformations of thin-shell objects are governed by a fourth-order differential equilibrium equations developed from Kirchhoff-Love theory. According to approximate theory, the convergence of finite-element solutions to these equations requires so-called C^1 interpolation. However, in conventional finite element method, the shape functions are defined as purely local polynomial functions and the nodal degrees of freedom only consist of displacements and slopes so that it's not possible to ensure C^1 continuity anywhere especially across the element boundaries. Inclusion of high-order derivatives among element nodal variables would introduce spurious oscillations in the solutions. Alternative approaches of degenerated solid elements, reduced-integration penalty method and many others [11, 70] have been proposed to compromise C^1 interpolation requirement. Nevertheless, they still rely on C^0 elements which exhibit poor performance in the presence of sever element distortion and it's difficult to integration them with established finite-element system. A recent paradigm proposed in [16] induce C^1 continuous shape functions from Loop's scheme and solve thin-shell problem directly on subdivision surfaces. However the shape functions used have no closed-form, thus unable be evaluated easily. Moreover, extraordinary points need special treatment and the number of them depends on the mesh connectivities rather than the intrinsic topologies of the underlying surfaces. It's noteworthy that Element-Free Galerkin(EFG) has also been attempted in thin-shell simulation, which unfortunately requires extra efforts in combining CAD models with simulation process via data conversion.

In CAD systems of nowadays, shell surfaces are oftentimes represented by spline functions. Also it's preferred to compute elastic deformations over spline-based thin-shell surfaces directly without converting them into an intermediate representation. However, it's not possible for conventional planar splines to complete this task by modeling displacement field on complicated shell surfaces as a single-piece function without domain patching/trimming. So it's necessary to adopt manifold spline technique [39] proposed recently, and develop a novel simulation system, where both geometries and deformations of shell objects are modeled by manifold splines

1.1.3 Trivariate Polycube Splines

Trivariate splines are smooth functions piecewise defined on three dimensional domain. They have received increasing attention as more and more 3D data are available today for processing in research and engineering areas. For instance, super splines [107] are quadratic trivariate functions proposed for precise iso-surface visualization. The are defined over uniform tetrahedral partition of gridded volume. Trivariate simplex splines are constructed on unstructured tetrahedral grids in order to improve the modeling and rendering of volumetric objects as well as their associated physical attributes. The major drawbacks of trivariate simplex splines are the inefficiency of computing basis functions and their derivatives, and the difficulty in configuring sub-knots to avoid undesired degeneracies. Similar to their counterparts in lower dimensions, trivariate tensor-product B-splines and NURBS are prevailing solid modeling tools and have been used in many research works due to their regular domain structure and computational efficiency. However, to model complicated geometries with trivariate tensor-product B-splines and NURBS requires gluing different spline patches and imposing continuity constraints between them, which is extremely cumbersome and unfriendly to users.

T-spline is an extension of NURBS and its partition-of-unity(POU) property is guaranteed by dividing the sum of all basis functions everywhere. Several NURBS patches may patch together into a polycube, forming a trivariate T-spline as a single-piece continuous representation. The flexibility of polycubic domains enables ones to model unstructured volume data, or solid objects with bifurcations and arbitrary topologies, which would potentially facilitate many applications in solid modeling, scientific computing and so on.

In this dissertation, we pursue to improve trivariate T-splines, particularly those defined over polycubes, by solving two problems. First, adding up all basis functions and conducting a division operation everywhere is computationally expensive. Can we construct a semi-standard [116] trivariate polycube spline so that POU requirement is automatically satisfied? Second, is it possible to restrict blending function influential regions inside the polycubic domain such that control points are prohibit from contributing to other regions across concaved corners? If yes, this kind of polycubic spline would serve as a promising tool for isogeometric analysis.

1.2 Contributions

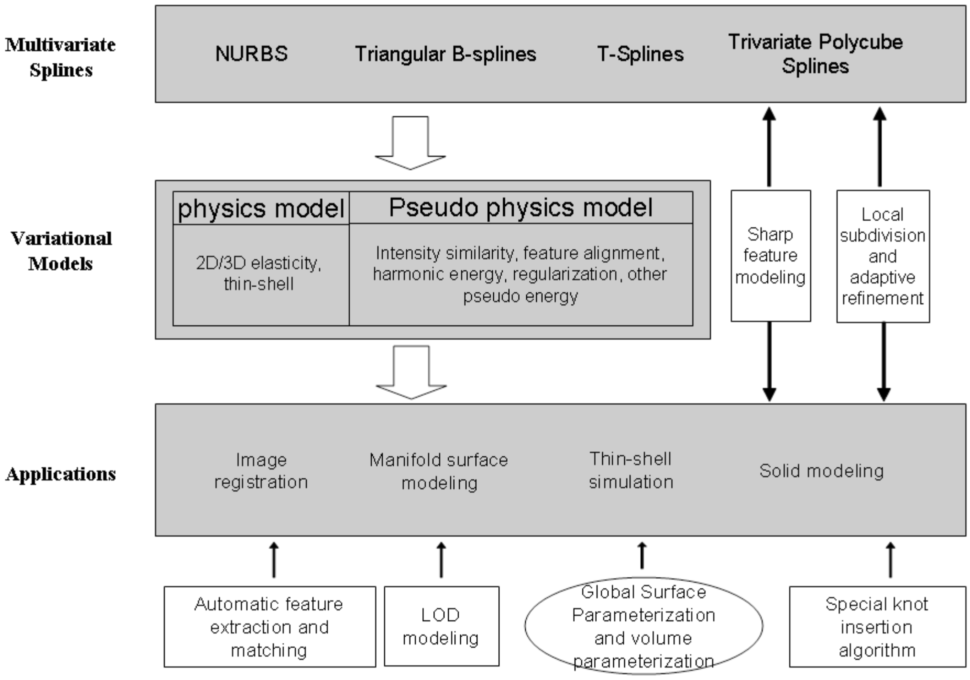


Figure 1.2: The overview of the variational framework of multivariate splines and the its applications

In this dissertation, we systematically study the multivariate spline theory, then present an computational framework for solving a range of variational problems. Incorporated with additional proposed techniques, for example automatic feature extraction and matching, our framework exhibits its success and effectiveness in solving many variational problems in the areas of shape modeling, image registration and physics-based simulation.

In summary, our specific contributions include:

1. We introduce a registration algorithm which successfully recovers non-rigid

deformation between two images while retaining sharp features on pre-identified local rigid objects. Both global non-rigidity and local rigidities are naturally modeled by an unified transformation representation of triangular B-spline. In contrast to conventional methods, our algorithm can precisely model sharp features with fewer degrees of freedom. (Chapter 3)

2. We develop triangular B-spline finite element method (TBFEM), and apply it to solve elastic PDE on a pseudo physical model, simulating the deformation between temporal mammogram pairs for image alignment. The experimental results show that the registration accuracy is improved when features boundaries are properly modeled by triangular B-spline. (Chapter 4)
3. We propose an novel image registration method designed specifically for mammograms. images. It automatically identifies significant texture features and breast contours, matches them between source and target images, and then recovers the transformation by solving a variational problem, which is guided by soft feature constrains and maximization of intensity similarities. Through extensive experiments, the proposed algorithm proves a effective one requiring minimal human interventions. (Chapter 5)
4. We develop a new paradigm that integrates thin-shell simulation and geometric design on arbitrary manifold spline surfaces. Particularly, we extend triangular B-spline from planar domain to manifold domain, and apply it to simulate elastic deformation of thin-shell objects with complicated geometries and non-trivial topologies. (Chapter 6)
5. We propose the technique of restricted trivariate polycube splines (RTP-splines), inspired by volumetric T-splines, but significantly differing in its top-down construction way. Our top-down construction approach includes

four major steps: extending given polycubes to bounding-box domain, building B-spline volumes, inserting duplicate knots to isolate polycubic domain as well as retaining restricted boundaries, and finally removing exterior cells of original polycube. The proposed RTP-splines is semi-standard, that is, POU is guaranteed without any enormous efforts in summing and dividing basis functions everywhere. Aside from local refinement ability inherit from T-spline, RTP-splines are also featured with restricted boundary, efficient evaluation of blending functions and their derivatives, and flexible polycubic domain, all of which would potentially benefit many applications of nowadays, where sold objects and volumetric data are involved. (Chapter 7)

1.3 Dissertation Organization

This proposal is organized in the following fashion. In Chapter 2, we begin with detailed review of a few dominate spline techniques, which is followed by a brief review on their applications in different disciplines. In Chapter 3, we present a novel image registration algorithm based on triangular B-spline. It is capable of recovering a global non-rigid transformation as well as retaining local rigidities, under a unified registration framework. In Chapter 4, we develop triangular B-spline finite element method, or TBFEM and apply it to solve an elastic problem on a pseudo physical model for temporal mammogram registration. Chapter 5 is devoted to an automatic registration method particularly designed for aligning mammogram images. It combines B-spline with proposed texture-based feature extraction and matching technique, and uses local intensity similarities as the guide to the solution. In Chapter 6, we simulate elastic deformations of complicated thin-shell objects by solving governing PDE with manifold splines. In Chapter 7, we present restricted trivariate polycube splines (RTP-splines) and conduct experiments on a few solid models. Finally, we conclude the dissertation and propose a few future works in Chapter 8.

Chapter 2

Background Review

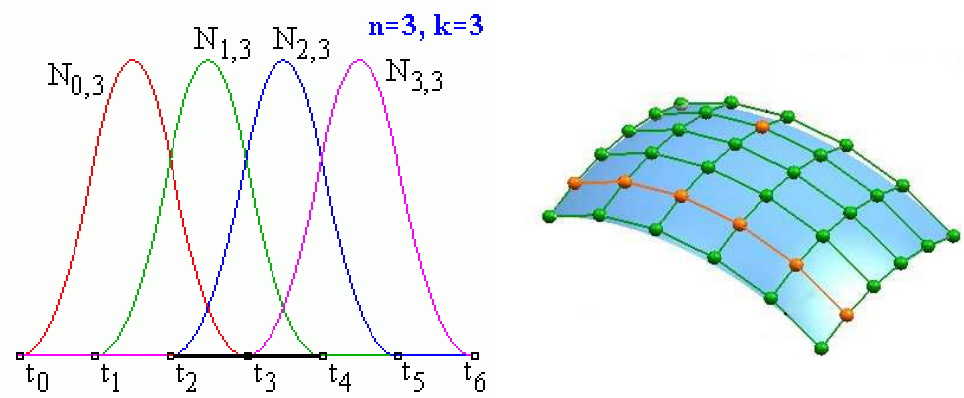
This section first survey the theory of multivariate splines. Particularly, we will cover tensor-product B-spline and NURBS, triangular B-splines, Powell-Sabin spline and T-splines. After that, we will review the their contribution in a variety of practical applications.

2.1 Multivariate Splines

It is well known that spline functions play very important roles in both theories and practices in many sciences and engineering areas. Due to their variety and complexity, it's important to study the theory of multivariate splines. Between the 1960's and early the 1970's, Birkhoff, Garabedian and deBoor [24] studied and established a series of theories on Cartesian tensor product multivariate splines. Although the Cartesian tensor product multivariate spline has its own application value, they are a simple extension of univariate spline functions, so they have many limitations. To combat such shortcoming, new spline schemes have been invented for the last few decades. They include triangular B-spline, T-spline and etc.

2.1.1 Tensor-product B-spline and NURBS

B-splines (short for Basis Splines) go back to Schoenberg who introduced them in 1946 [113, 114] for the case of uniform knots. B-splines over nonuniform knots go back to a review article by Curry in 1947 [18]. de Boor derived the recursive evaluation of B-spline curves [9]. It was this recursion that made B-splines a truly viable tool in CAGD. Before its discovery, B-splines were defined using a tedious divided difference approach which was numerically unstable. Later on, Gordon and Riesenfeld realized that de Boor's recursive B-spline evaluation is the natural generalization of the de Casteljau algorithm and Bézier curves are just subset of B-spline curves. Versprille [133] generalization of B-spline curves to NURBS (non-uniform rational B-spline) which has become the standard curve and surface form in the CAD/CAM industry [93].



(a) (b)

Figure 2.1: (a) shows the non-rational basis functions of quadratic B-spline. (b) shows a patch of surface represented by NURBS

A NURBS curve generalizes the B-spline. It is the combination of a set of piecewise rational functions with $n+1$ control points $\mathbf{p}_i$ and associated weights w_i:

$$\mathbf{c}(u) = \frac{\sum_{i=0}^{n} \mathbf{p}_i w_i B_{i,k}(u)}{\sum_{j=0}^{n} w_j B_{j,k}(u)}$$

where u is the parametric variable and $B_{i,k}(u)$ are B-spline basis functions. Assuming basis functions of degree $k-1$, a NURBS curve has $n+k+1$ knots t_i in nondecreasing sequence: $t_0 \leq t_1 \leq \ldots \leq t_{n+k}$. The basis functions are defined recursively as:

$$B_{i,1}(u) = \begin{cases} 1 & \text{for } t_i \leq u < t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

with

$$B_{i,k}(u) = \frac{u - t_i}{t_{i+k-1} - t_i} B_{i,k-1}(u) + \frac{t_{i+k} - u}{t_{i+k} - t_{i+1}} B_{i+1,k-1}(u)$$

The parametric domain is $t_{k-1} \leq u \leq t_{n+1}$. In many applications, the end knots are repeated with multiplicity k in order to interpolate the initial and final control points $\mathbf{p}_0$ and $\mathbf{p}_n$. Figure 2.1(a) shows the basis functions of uniform quadratic B-spline.

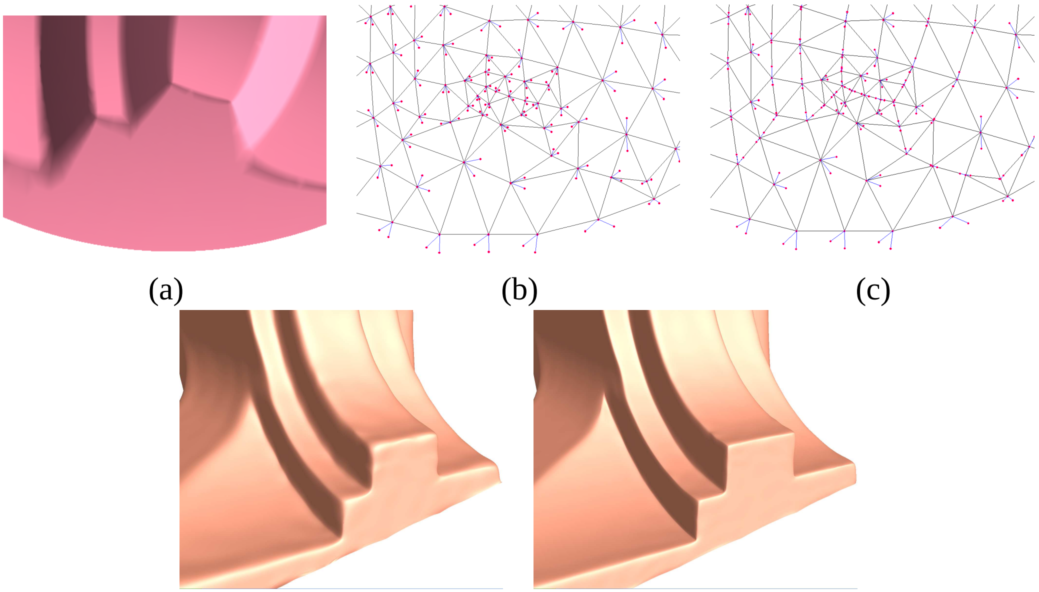
A NURBS surface is the generalization of the tensor-product B-spline surface. It is defined over the parametric variables u and v as

$$\mathbf{s}(u,v) = \frac{\sum_{i=0}^{m} \sum_{j=0}^{n} \mathbf{p}_{i,j} w_{i,j} B_{i,k}(u) B_{j,l}(v)}{\sum_{i=0}^{m} \sum_{j=0}^{n} w_{i,j} B_{i,k}(u) B_{j,l}(v)}$$

A NURBS surfaces has $(m+1)(n+1)$ control points $\mathbf{p}_{i,j}$. Assuming basis functions along the two parametric axes of degree $k-1$ and $l-1$, respectively, the number of knots is $(m+k+1)(n+l+1)$. The nondecreasing knot sequence is $t_0 \leq t_1 \leq \ldots \leq t_{m+k}$ along the u-axis and $s_0 \leq s_1 \leq \ldots \leq s_(n+l)$ along the v-axis. The parametric domain is $t_{k-1} \leq t_{m+1}$ and $s_{l-1} \leq s_{n+1}$. If the end knots have multiplicity k and l in the u and v axis respectively, the surface patch will interpolate the four corners of the boundary control points. Figure 2.1(b) shows a surface represented by NURBS.

NURBS generalize the non-rational parametric form. Like non-rational B-splines, the rational basis functions of NURBS sum to unity, they are infinitely smooth in the interior of a knot span provided the denominator is not zero, and at a knot they are at least C^{k-1-r} continuous with knot multiplicity r, which enables them to satisfy different smoothness requirements. They inherit many of the properties of uniform B-splines, such as the strong convex hull property, variation diminishing property, local support, and invariance under standard geometric transformations. More material of NURBS and further detailed discussion of its properties can be found in [7, 31, 88–90, 92, 129]

2.1.2 Triangular B-spline


(a) (b) (c)

(d) (e)

Figure 2.2: Modeling features using triangular B-spline with degenerate knots. (a) the parametrization of the fandisk model. (b) the domain triangulation and regular knot configurations (no three knots in a domain triangle are collinear). (c) Place the knots along the user-specified edges of domain triangulation. (d) A cubic spline surface reconstructed using the knot configurations in (b). The spline is C^2 continuous everywhere. (e) A cubic spline surface reconstructed using the knot configurations in (c). The spline is C^2 continuous on smooth regions and C^0 on sharp features.

Based on the blossom or polar form [99] and *B*-patch [118], Dahmen, Micchelli and Seidel [20] propose a general spline scheme in *s*-dimensional space, which constructs a collection of multivariate B-splines whose linear span comprises all polynomials of degree at most n. The bivariate case is called triangular B-spline or DMS spline. Due to its elegant construction and many attractive properties for geometric modeling, triangular B-spline has received much attention since its inception. Fong and Seidel [33] present the first prototype implementation of triangular B-splines and show several useful properties, such as affine invariance, convex hull, locality, and smoothness. Greiner and Seidel [38] show the practical feasibility of multivariate B-spline algorithms in graphics and shape design. Pfeifle and Seidel [87] demonstrate the fitting of a triangular B-spline surface to scattered functional data through the use of least squares and optimization techniques. Gormaz and Laurent study the piecewise polynomial reproduction of triangular B-spline and give a direct and intuitive proof [37]. Franssen et al. [35] propose an efficient evaluation algorithm, which works for triangular B-spline surfaces of arbitrary degree.

The construction of triangular B-spline in [35] is as follows: let points $\mathbf{t}_i \in \mathbb{R}^2$, $i \in \mathbb{N}$, be given and define a triangulation

$$T = \{\Delta(I) = [\mathbf{t}_{i_0}, \mathbf{t}_{i_1}, \mathbf{t}_{i_2}] : I = (i_0, i_1, i_2) \in I \subset \mathbb{N}^2\}$$

of a bounded region $D \subseteq \mathbb{R}^2$. Next, with every vertex $\mathbf{t}_i$ of T we associate a cloud of knots $\mathbf{t}_{i,0}, \ldots, \mathbf{t}_{i,n}$ such that $\mathbf{t}_{i,0} = \mathbf{t}_i$ and for every triangle $I = [\mathbf{t}_{i_0}, \mathbf{t}_{i_1}, \mathbf{t}_{i_2}] \in T$,

1. all the triangles $[\mathbf{t}_{i_0,\beta_0}, \mathbf{t}_{i_1,\beta_1}, \mathbf{t}_{i_2,\beta_2}]$ with $\beta = (\beta_0, \beta_1, \beta_2)$ and $|\beta| = \sum_{i=0}^{2} \beta_i \leq n$ are non-degenerate.
2. the set

$$interior(\cap_{|\beta| \leq n} X_\beta^I) \neq \emptyset, X_\beta^I = [\mathbf{t}_{i_0,\beta_0}, \mathbf{t}_{i_1,\beta_1}, \mathbf{t}_{i_2,\beta_2}] \qquad (2.1)$$

3. If I has a boundary edge, say, $(\mathbf{t}_{i_0}, \mathbf{t}_{i_1})$, then the entire area $[\mathbf{t}_{i_0,0}, \ldots, \mathbf{t}_{i_0,n}, \mathbf{t}_{i_1,0}, \ldots, \mathbf{t}_{i_1,n})$ must lie outside of the domain.

Then the triangular B-spline basis function N^I_β, $|\beta| = n$, is defined by means of simplex splines $M(\mathbf{u}|V^I_\beta)$ (for details about simplex splines, please refer to [79].) as

$$N(\mathbf{u}|V^I_\beta) = |d^I_\beta| M(\mathbf{u}|V^I_\beta)$$

where $V^I_\beta = \{\mathbf{t}_{i_0,0}, \ldots, \mathbf{t}_{i_0,\beta_0}, \ldots, \mathbf{t}_{i_2,0}, \ldots, \mathbf{t}_{i_2,\beta_2}\}$ and

$$d^I_\beta = d(X^I_\beta) = det \begin{pmatrix} 1 & 1 & 1 \\ \mathbf{t}_{i_0,\beta_0} & \mathbf{t}_{i_1,\beta_1} & \mathbf{t}_{i_2,\beta_2} \end{pmatrix}$$

Assuming (2.1), these B-spline basis functions can be shown to be all non-negative and to form a partition of unity. Then, the triangular B-spline is defined as

$$\mathbf{F}(\mathbf{u}) = \sum_{I \in I} \sum_{|\beta|=n} \mathbf{c}_{I,\beta} N(\mathbf{u}|V^I_\beta) \tag{2.2}$$

where $\mathbf{c}_{I,\beta}$ is the control point. This spline globally C^{n-1} if all the sets X^I_β, $|\beta| \leq n$ are affinely independent.

One favorable characteristic of triangular B-splines is that by adjusting knots along feature lines explicitly, we can model local sharp features (*i.e.*C^0 continuity) in the approximated space, while keeping the C^{n-1} smoothness over the other regions. This advantage is demonstrated in a surface reconstruction example shown in Figure 2.2.

2.1.3 Powell-Sabin Spline

Powell-Sabin splines are functions in the space $S_2^1(\Delta_{ps})$ of C^1 continuous piecewise quadratic functions on a Powell-Sabin refinement [95]. Such a refinement Δ_{ps} can be obtain from an arbitrary triangulation Δ by splitting each triangle into six sub-triangles with a common interior point. In contrast to triangular Bézier splines, where imposing smoothness conditions between patches requires a great number of nontrivial relations among the control points to be satisfied. the C^1 continuity of a Powell-Sabin spline is guaranteed for any choice of the control points.

The first B-spline representation of Powell-Sabin spline was derived by Shi *et al.* [121]. However, their construction approach had serious drawbacks from the numerical point of view. Dierckx [26] resolved the numerical problem by constructing a normalized B-spline basis for Powell-Sabin splines. This representation has a very nice geometric interpretation involving the tangent control triangles for manipulating the Powell-Sabin surfaces. Since then, the normalized Powell-Sabin spline has been receiving much attention in the computer aided geometric design community. Surface approximation and interpolation using Powell-Sabin spline have been reported in [27, 73, 145]. Windmolders and Dierckx solved the subdivision problem for uniform Powell-Sabin splines, that is on triangulations with all equilateral triangles [146]. Recently, Vanraes *et al.*present the subdivision rule for general Powell-Sabin spline [131].

Let Ω be a polygonal domain in $\mathbb{R}^2$ and let Δ be a conforming triangulation of Ω, constituted of triangles ρ_j, $j = 1,\ldots,N_t$, having vertices $V_i := (x_i,y_i), i = 1,\ldots,N_v$. A Powell-Sabin refinement, Δ_{ps} of Δ is the refined triangulation, obtained by subdividing each triangle of Δ into six sub-triangles as follows. Select an interior point Z_j in each triangle ρ_j and connect it with the three vertices of ρ_j and with the points $Z_{j_1}, Z_{j_2}, Z_{j_3}$ where $\rho_{j_1}, \rho_{j_2}, \rho_{j_3}$ are the triangles adjacent to ρ_j (See Figure 2.3).

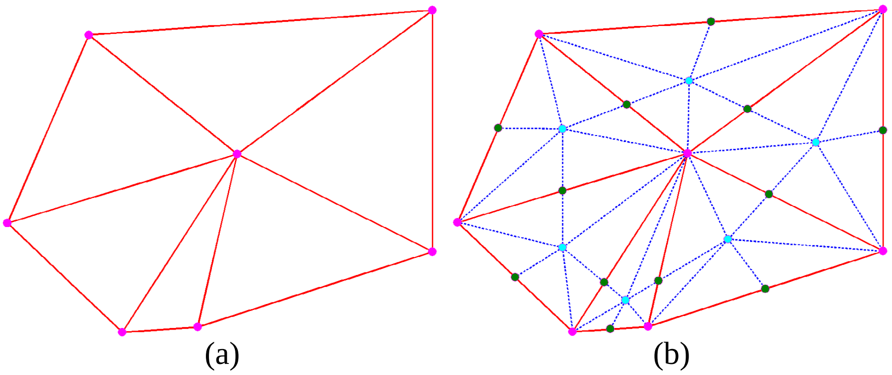


Figure 2.3: The Powell-Sabin refinement Δ^* (b) of a triangulation Δ (a).

We denote by $S_2^1(\Delta_{ps})$ the space of piecewise C^1 continuous quadratic polynomials on Δ_{ps}. Powell-Sabin [95] proved that the dimension of the space $S_2^1(\Delta_{ps})$ equals $3N_v$ and any element of $S_2^1(\Delta_{ps})$ is uniquely determined by its value and its gradient at the vertices of Δ, i.e., there exists a unique solution $s(x,y) \in S_2^1(\Delta_{ps})$ for the interpolation problem

$$s(V_i) = f_i \quad \frac{\partial_s}{\partial_x}(V_i) = f_{x,i} \quad \frac{\partial_s}{\partial_y}(V_i) = f_{y,i} \quad i = 1, \dots, N_v. \tag{2.3}$$

So given the function and its derivative values at each vertex V_i, the Bézier ordinates on the domain sub-triangles are uniquely defined and the continuity conditions between sub-triangles are automatically fulfilled.

Dierckx [26] showed that each piecewise polynomial $s(x,y) \in S_2^1(\Delta_{ps})$ has a unique representation

$$s(x,y) = \sum_{i=1}^{N_v} \sum_{j=1}^{3} c_{ij} B_i^j(x,y), (x,y) \in \Omega \tag{2.4}$$

where the basis functions form a partition of unity, i.e.,

$$B_i^j(x,y) \geq 0 \tag{2.5}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{3} B_i^j(x,y) = 1 \textbf{ forall } x,y \in \Omega \tag{2.6}$$

Furthermore, these basis functions have local support: $B_i^j(x,y)$ vanishes outside the so-called molecule M_i of vertex V_i, which is the union of all triangles $\mathcal{T}_k$ containing V_i.

The basis functions $B_i^j(x,y)$ can be obtained by finding three linearly independent triplets $(\alpha_{ij},\beta_{ij},\gamma_{ij})$, $j=1,2,3$ for each vertex V_i. $B_i^j(x,y)$ is the unique solution of the interpolation problem with $(f_k, f_{xk}, f_{yk}) = (\delta_{ki}\alpha_{ij}, \delta_{ki}\beta_{ij}, \delta_{ki}\gamma_{ij})$, where δ_{ki} is the Kronecker delta. The triplets $(\alpha_{ij},\beta_{ij},\gamma_{ij})$, $j=1,2,3$ are determined by the following Dierckx's algorithm [26, 130]:

1. For each vertex v_i, find its Powell-Sabin triangle points, which are the immediately surrounding Bézier domain points of the vertex v_i and vertex v_i itself.
2. For each vertex v_i, find a triangle $t_i(Q_{i1},Q_{i2},Q_{i3})$ which contains all the Powell-Sabin triangle points of v_i from all the triangles in the molecule M_i. Denote $Q_{ij} = (X_{ij},Y_{ij})$ the position of vertices Q_{ij}.
3. Three linearly independent triplets of real numbers $\alpha_{ij},\beta_{ij},\gamma_{ij}$, $j=1,2,3$ can be derived from the Powell-Sabin triangle t_i of a vertex v_i as follows:
    - $(\alpha_{i1},\alpha_{i2},\alpha_{i3})$ = barycentric coordinate of v_i with respect to t_i
    - $(\beta_{i1},\beta_{i2},\beta_{i3}) = ((Y_{i2}-Y_{i3})/h, (Y_{i3}-Y_{i1})/h, (Y_{i1}-Y_{i2})/h)$
    - $(\gamma_{i1},\gamma_{i2},\gamma_{i3}) = ((X_{i3}-X_{i2})/h, (X_{i1}-X_{i3})/h, (X_{i2}-X_{i1})/h)$,

$$\text{where } h = \det \begin{pmatrix} 111 \\ X_{i1}X_{i2}X_{i3} \\ Y_{i1}Y_{i2}Y_{i3} \end{pmatrix}.$$

We then define the control triangles as $T_i(C_{i1}, C_{i2}, C_{i3})$. Dierckx proved that the normalized Powell-Sabin spline has a very nice geometric interpretation that the control triangle is tangent to the spline surface [26]. Figure 2.4 illustrates an example of Powell-Sabin spline surface over a planar triangulated domain.

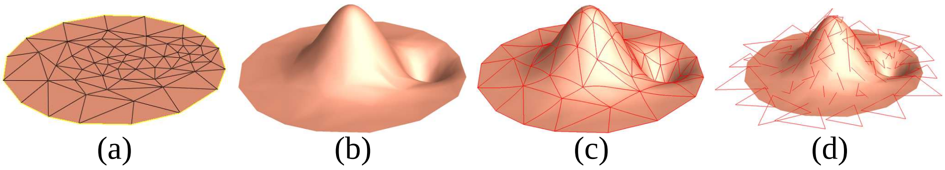
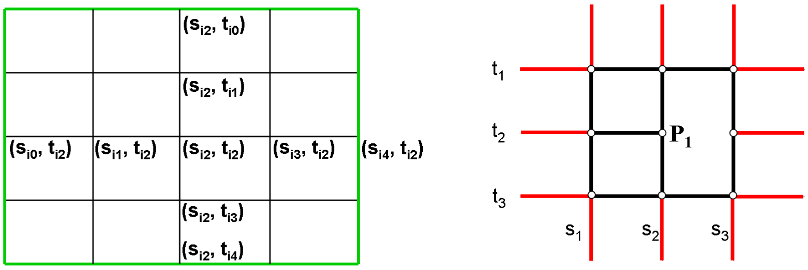


Figure 2.4: Powell-Sabin spline over a planar domain: (Courtesy of [44]) (a) Domain triangulation; (b) Spline surface; (c) Spline surface, the red curves correspond to the edge in the domain triangulation; (d) Spline surface overlaid by the control triangles (shown in red) which are tangent to the surface;

2.1.4 T-spline

T-splines, developed by Sederberg, Zheng, Bakenov, and Nasri [117], are a generalization of NURBS surfaces that are capable of significantly reducing the number of superfluous control points by using the T-junction mechanism. The main difference between a T-spline control mesh and a NURBS control mesh is that T-splines allow a row or column of control points to terminate at anywhere without strictly enforcing the rectangular grid structure throughout the parametric domain. Consequently, T-splines enable much better local refinement capabilities than NURBS. Furthermore, using the techniques presented in [117], it is possible to merge adjoining T-spline surfaces into a single T-spline without adding new control points. However, this patching process requires that the knot intervals of the to-be-merged edges must establish an one-to-one correspondence between the two surfaces. Sederberg et al. also develop an algorithm to convert NURBS surfaces into T-spline surfaces, in which a large percentage of superfluous control points are eliminated [116]. Most recently Song and Yang [123] generalized T-spline scheme to weighted T-spline and demonstrated its applicability in 3D free-form deformation.



(a) (b)

Figure 2.5: (a) Local knot lines for basis function $B_i(s,t)$; (b) $\mathbf{P}_i$ is a T-junction.

T-spline is a *PB*-spline for which some order has been imposed on the control points by means of a control grid called a *T-mesh*. A T-mesh is basically a rectangular grid that allows T-junctions. Each edge in T-mesh is a line segment of constant s (which is called s-edge) or constant t (which is called t-edge) [117]. A T-junction is a vertex shared by one s-edge and two t-edges, or by one t-edge and two s-edges. For example, $\mathbf{P}_1$ (see Figure 2.5(b)) is a T-junction. Each edge in a T-mesh is labeled with a knot interval, constrained by the following rules:

1. The sum of knot intervals on opposing edges of any face must be equal.
2. If a T-junction on one edge of a face can be connected to a T-junction on an opposing edge of the face (thereby splitting the face into two faces) without violating Rule 1, the edge must be included in the T-mesh.

In contrast to tensor-product B-spline that uses a rectangular grid of control points, *PB*-spline is point-based and requires no topological relationship among control points. The equation for a *PB*-spline is given by:

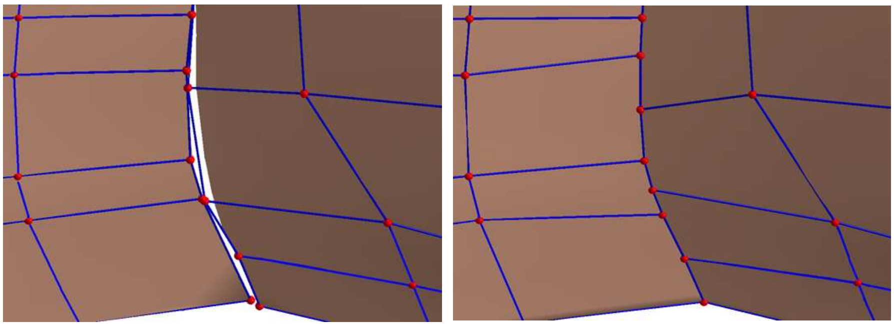
$$\mathbf{P}(s,t) = \frac{\sum_{i=1}^{n} \mathbf{P}_i B_i(s,t)}{\sum_{i=1}^{n} B_i(s,t)} \quad (s,t) \in D$$

where the $\mathbf{P}_i$ are control points. The $B_i(s,t)$ are basis functions written as

$$B_i(s,t) = N_{i0}^3(s) N_{i0}^3(t)$$

where $N_{i0}^3(s)$ is the cubic B-spline basis function associated with the knot vector $\mathbf{s_i} = [s_{i0}, s_{i1}, s_{i2}, s_{i3}, s_{i4}]$ and N_{i0}^3 is associated with the knot vector $\mathbf{t_i} = [t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}]$ as illustrated in Figure 2.5(a). Every control point has its influence domain $D_i = (s_{i0}, s_{i4}) \times (t_{i0}, t_{i4})$. The T-spline equation is very similar to the equation for a tensor-product rational B-spline surface, except that knot vectors $\mathbf{s}_i$ and $\mathbf{t}_i$ are deduced from the T-mesh neighborhood of $\mathbf{P}_i$.

Knot vector $\mathbf{s}_i$ and $\mathbf{t}_i$ for the basis function $B_i(s,t)$ are determined as follows. Let (s_{i2}, t_{i2}) are the knot coordinate of $\mathbf{P}_i$. Consider a ray in parameter space $\mathbf{R}(\alpha) = (s_{i2} + \alpha, t_{i2})$. Then s_{i3} and s_{i4} are the s coordinates of the first two s-edges intersected by the ray. The other knots can be found in like manner.



(a) B-spline surfaces (b) T-splines

Figure 2.6: A gap between two B-spline surfaces, fixed with a T-spline (Courtesy of [117])

.

T-junctions allow T-splines to be locally refinable, that is control points can be inserted into the control grid without propagating an entire row or column of control points. Figure 2.6(a) shows a small rectangular area blown up to magnify a hold where neighboring B-spline surfaces do not match exactly. Figure 2.6(b) shows the model after being converted into gap-free T-spline, thereby eliminating the need for repair.

2.1.5 Other Multivariate Splines

Besides the aforementioned splines, there are many other multivariate splines developed in the last few decades. They were usually originated for specific tasks, gradually received more attention from other communities later. Thin plate splines(TPS) were introduced by Duchon to model spatial transformation. Both global affine and local non-affine components are successfully represented by a group of radial basis functions(RBF). TPS has become an effective and popular tool in image alignment, shape matching and many other computer vision applications. [8]. Rössl *et al's* [107] introduce quadratic trivariate super spline on uniform tetrahedral partition to reconstruct continuous models from gridded volume. Bernstein-Bézier technique is used for evaluation of spline value and its gradient. Box splines are extension of B-splines with equidistant knots. Namely they can be viewed as the projection of higher dimensional boxes. Box splines consists of triangular polynomial pieces, and can simply construct arbitrary two-dimensional surfaces, *i.e.*, manifolds. A very comprehensive discussion of box splines is given in the book [25]. Cauchy-Navier spline(CNS) was first proposed by Davis *et al.* [22] for matching 3D Magnetic Resonance images of the breast. The development and application of the CNS is similar to that of Thin-Plate splines. It is time-dependent and based on a physical model of a homogeneous isotropic elastic body. Such a physical model is especially appropriate for matching medical and biological images. Due to the limit of pages, we are unable to review these splines in details. But their applicational potential will be explored in our future research.

2.2 Applications of Multivariate Splines

In the last few decades, many research literatures have been written on bivariate, respectively multivariate splines. This work has been motivated in many cases by the aim to develop powerful tools for fields of applications, such as scattered data fitting, the construction and reconstruction of surfaces and the numerical solution of boundary-value problems.

2.2.1 Shape Representation

In computer graphics, polygonal models are still the most common form of surface representation. However, polygonal models of the integument attempt to describe a basically continuous surface with linear approximations requiring many polygons to obtain a visually smooth surfaces. Mathematically, splines are powerful and effective tools to represent continuous and smooth surfaces by parameterizing them with a number of control points. NURBS, generalizing B-splines [113, 114] and Bézier splines, have become industry standard tools for representation of geometry and thus been incorporated into many commercial modeling systems [91]. In spite of the popularity, tensor-product splines have limitations arising from their rectilinear nature. Inserting a single control point on B-spline surface requires propagating an entire row or column of control points. To combat this, Hierarchical B-splines [34] [61] were introduced for local refinement using an efficient representation and constructing multilevel spline spaces. T-splines [116, 117] are a generalization of NURBS surface that are capable of significantly reducing the number of superfluous control points when doing local refinement. One the other hand, triangular splines, such as DMS-splines [20], Powell-Sabin splines [95] and box-splines [25], are introduced to handle representation of continuous surfaces over irregular domains. Nevertheless, all these splines are defined on planar domain. It is hard for them to model complicated surfaces with non-trival topologies without patching and stitching, which is cumbersome. Manifold spline [39] is a systematical extension of traditional planar domain from $\mathcal{R}^2$ to manifold domain, which can elegantly model open/close surfaces with arbitrary topologies. Manifold spline is theoretically founded upon the technique recently proposed by Gu *et al.* [40, 41]. Implicit functions are another widely used method for shape representation. Gradients and high-order derivatives determined from volumetric implicit functions are essential to volume rendering, iso-surface extraction and other applications. Hua *et al.* [52] represent volumetric implicit function by using hierarchical trivariate B-splines, and then use such framework for direct physics-based modeling of the underlying shapes [51]. Rössl *et al.*construct a super-spline representation of complicated surfaces from their discrete implicit function values, and efficiently visualize them with ray casting.

2.2.2 Shape Modeling

Free-form deformation (FFD), as introduced by Sederberg *et al.* [115], is known to be a shape modification method that has been extensively applied to computer animation and geometric modeling. This technique deforms an object

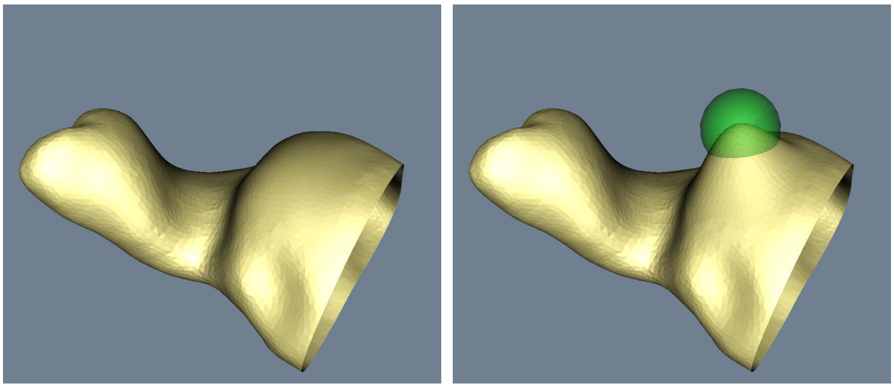

(a) (b)

Figure 2.7: The surface of a hearing-aid model is treated as free-form deformable model and interactively modified by users. The green sphere is a handle to guide the desired deformation.

by embedding it within a solid, which is usually parameterized by a group of control points. Davis *et al.* [23] used rational Bézier spline bases for the parametric volume, and later Griessmair *et al.* ffd-bspline used a trivariate B-spline. Lamousin *et al.* [63] used NURBS volume for embedding volume. Hsu *et al's* [49] proposed an FFD with trivariate B-spline that allows users to manipulate points on the surface of the embedded object directly. Most recently, Song *et al.* [123] developed w-TFFD based on weighted T-spline, which allows multilevel construction and deformation. Instead of deforming the embedding volume, Hua *et al.* [51, 52] directly manipulate underlying shapes by modifying their implicit volumes, which are parameterized with trivariate B-splines.

Free-form deformable models were first introduced to computer graphics by Terzopoulos *et al.* [128]. They employed elasticity theory to construct differential equations that model the behavior of non-rigid curves, surfaces, and solids as functions of time. This technique provides users with a intuitive way for interactive shape modeling and sculpture. It is ideal for free-form deformable models to be represented by splines, which will unify the stages of shape construction and designing. Qin *et al.* [97] introduced D-NURBS surfaces, an extension to traditional NURBS that permits more natural control of the surface geometry. Figure 2.7 shows that the surface of a hearing-aid model is converted to free-form deformable model, then is adjusted interactively by professionists in a computer-aided design system.

2.2.3 Image Processing

Digital images are captured with designated resolution. They usually need resampling to achieve desired resolution in computer vision applications. B-spline interpolation, or cubic convolution [71], is one of the most popular mathematical tools for this task. In nonrigid image registration, multivariate spline also plays an important role in representing continuous spatial transformation between source and target images. FFD with spline basis functions has been shown to be a valuable tool in medical image registration [10, 108]. In [124], Bézier spline are used to express deformation. However, Bernstein basis functions have no strict local support, which leads to large computational complexity and insufficient deformation. In [101, 108], B-spline is utilized for MRI and Mammogram registration. B-spline basis functions have strict local support and better flexibility. Since the quality of the quality of the registration is directly related to the transformation's degree of freedom [47], Wang *et al.* [137] propose to replace uniform B-spline basis functions by NURBS, which allows nonuniform distribution of control points and knot vectors. Recently, Xie *et al.* [150] propose to use hierarchical B-spline to registration brain MRI images, which offers local refinement for preserving fine details. Thin-spline splines (TPS) as a mathematical interpolator introduced by Duchon have become an alternative choice for general image warping. Kim *et al.* [57, 58] present an automated thin-plate spline, where an arbitrary set of landmarks are supplied initially and then are iteratively repositioned until certain criteria is met. Rohr *et al.* [106] propose an approximating thin-plate spline with landmarks weighted by the uncertainty in their localization, which relax the interpolation constrain and allows for semi-auto landmark extraction. In addition to non-rigid image registration, multivariate splines, especially cubic B-spline curves and surfaces, are widely used to represent evolvable surfaces or volumes in deformable models system, which offers a unique and powerful tools for image segmenting, matching and tracking [77].

2.2.4 Scientific Computation

Due to their intrinsic properties of non-negativity and "partition-of-unity", multivariate spline functions are ideal to serve as basis functions in finite element method to solve scientific and engineering problems. In the field of computer graphics, Qin *et al.* [97, 98] proposed D-NURBS, a physics-based framework for intuitive geometric shape design. In this framework, pseudo physical energies are defined over NURBS surfaces to enforce certain criteria, then solved by finite element method with NURBS basis functions to achieve optimal surface design. In the broad areas of physics and engineering industry, many spline functions have been widely incorporated in finite element method to solve partial differential equations in mechanics, thermodynamics and etc. Dag *et al.* [19] apply splitting method to Burger's equation and solve the equations by using both quadratic and cubic B-spline Gakerkin finite element techniques. A finite element solution of Kdv equation is presented in [2]. The numerical solutions obtained by B-spline finite element is found to in better agreement with the exact solutions than other numerical solutions in the literature. Yang [154] present graded B-spline finite element solution, which allows for direct interaction between the design and analysis model of heterogeneous objects without laborious meshing operation. Khalifa *et al.* [56] numerically solved modified regularized long wave equation by collocation method using cubic B-spline finite element. In additional to popular B-spline scheme, other elements such as Hermite triangular, Powell-Sabin etc have also been attempted for solving specific problems. For example, in [125], Powell-Sabin splines are attempted for numerical solution of partial differential equations defined on polygonal domains. Special discussion also goes to the treatment of Dirichlet and Neumann boundary conditions for Powell-Sabin spline finite element in [125].

Chapter 3

Triangular B-spline based Image Registration

3.1 Motivation

For the last decade, image registration has become an important technique for various computer vision and medical applications, fusing the information from images acquired either at different times or on multiple modalities. A number of reviews have been documented in [10,72,158]. The earliest attempts made by [85,94] typically restrict the deformation between the corresponding images to be rigid and consider global geometric differences only. Later, non-rigid registration was introduced in [12, 108] to additionally cope with local differences, resulting from different anatomy, intraoperative deformation, or distortion induced during imaging process. It is often assumed by the non-rigid registration that the objects in the matching images behave as if they were a single elastic body, *i.e.*, the stiffness is constant everywhere. However, this is rarely the case when the imaged anatomy contains both rigid and soft structures. A practical clinical example was described in [30] where the shape of the brain changed after subdural electrodes were implanted in a surgical procedure. Neither a single rigid body motion nor a nonlinear model with invariant smoothness can accurately represent the transformation between preoperative and postoperative scans since the electrodes translate and rotate only, while the others deform nonlinearly. Therefore, more appropriate methods are required to combine the modeling of both rigidity and non-rigidity in the recovered transformation. Especially, the C^0 continuity on the borders of rigid structures needs to be simulated correctly for precise registration.

In principle, we could build patient-specific physical model to predict the interaction between rigid structures and soft tissues. However, it is impractical to achieve solution with desired accuracy due to high computational cost and insufficient details on mass, elasticity, and other mechanical properties. The efforts made so far were either based on interpolatory spline scheme or through a variational framework. Little *et al.* [68] incorporated independent rigid objects in a modified thin-plate spline (TPS) based nonrigid registration. Anisotropic landmarks were introduced by Rohr *et al.* [104] to TPS to enforce local rigidity constraints. Duay *et al.* [29] simulated the rigid motions by adaptively adjusting TPS radial basis functions according to local stiffness. Tanner *et al.* [126] represented the deformation using B-splines and locally couple control points in order to model local rigidities. Most recently, Loeckx *et al.* [69] introduce a penalty term to keep voxel-based rigidities in their variational framework by enforcing the orthogonality of Jacobian matrix. Nevertheless, none of the above approaches, except that in [68], can precisely describe C^0 continuity in the displacement field. In spite of the attempt made in [78], it's not straightforward for thin-plate splines to be incorporated with variational framework, which is quite a powerful tool for intensity-based image registration. On the other hand, tensor-product B-splines has been widely used for optimization-based registration approaches [108] [103] [132]. Although it is possible for tensor-product B-splines to describe sharp features when the corresponding knots collapse, such features can not lie in arbitrary direction due to the regular domain of B-splines.

In this chapter, we develop a novel non-rigid registration algorithm in which the recovered deformation field is represented by triangular B-splines. We first build the domain triangulation and adjust corresponding knots to the boundaries of pre-segmented rigid structures. As a result, the C^0 continuity is guaranteed at the desired places in the displacement field. The landmarks, selected at the vicinities of rigid objects, are brought into correspondence between source and target images as point-based constraints. The optimal transformation is then estimated by minimizing a composite energy function, which measures image discrepancy, deformation distortion, and desired local rigidities. Empowered by the numerous advantages of triangular B-splines, such as flexible domain, local control, space-varying smoothness modeling, etc., our registration approach makes the following contributions: The local linear motion in the global non-rigid transformation, caused by rigid structures, can be accurately recovered using relatively fewer degrees of freedom (DOFs), as long as the feature lines are properly aligned in the domain triangulation. With C^0 continuity modeled at the interface between rigid and non-rigid objects, the deformable region nearby can move more freely and tend to improve the registration quality considerably.

3.2 Method

Given source image I_s, and target image, I_t, defined on the domain $\Omega \subset \mathbb{R}^2$, the problem of registration is to find an optimal geometrical transformation $\mathbf{T} : \Omega \to \mathbb{R}^2$

such that the pixels in both images are matched properly.

3.2.1 Transformation Model

To reduce the global geometric differences between I_s and I_t, an initial alignment is achieved using conventional rigid registration algorithm. This obtained transformation will be used as the initial estimation for the following registration.

The concept of free-form deformation (FFD) is to deform an object by manipulation underlying control points. In our work, the FFD is decomposed as an identity transformation plus a displacement field, which is represented by triangular B-splines as:

$$\mathbf{T}(\mathbf{x}) = \mathbf{x} + \mathbf{u} = \mathbf{x} + \sum_{i=1..m} \phi_i B_i(\mathbf{x}), \tag{3.1}$$

where ϕ_i is the control point and B_i is the associated basis function.

Unlike tensor-product B-spline based FFD [103], whose domain is a rectangular lattice, our triangular B-spline based FFD has its domain built upon a tessellation of either triangles for 2D or tetrahedra for 3D. It is not difficult to triangulate the reference image domain Ω using established techniques. In order to model the sharp features (see Fig.3.2(c)) at the boundaries of pre-identified rigid bodies, we have to keep them in the triangulated tessellation. Such constraints can be satisfied using the triangulation algorithm proposed by Shewchuk [120]. According to the definition of triangular B-splines, the free-form deformation field has C^{n-1} continuity everywhere if there is no degeneracy for any triple of knots in the same triangle. However, we purposely collapse adjacent sub-knots to pre-identified feature lines in order to model desired C^0 continuity.

Due to the flexibility of the domain triangulation, it is also possible for users to overlay the registration domain exactly upon the region of interest (ROI), rather than covering the entire reference image. Thus, the computational effort will be saved considerably, especially when the ROI can be successfully extracted.

3.2.2 Point-based Constraints

Point-based constraints are incorporated in our framework for better registration. The points on the boundary contours of rigid structure with high curvature are good candidates for landmarks (see Fig.3.2(d)). Assuming strict rigidity of bony structures, only two pairs of landmarks are required to fully recover local linear transformation (*i.e.*,translation and rotation), if there is no rotoinversion. In practice, we often introduce more constraints to ensure the stability of the registration.

Let $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ be the set of landmarks chosen on the reference domain (I_t in our implementation). Their correspondences in I_s are $\mathcal{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$ such that:

$$\mathbf{T}(\mathbf{p}_j; \Phi) = \mathbf{q}_j \quad \text{for} \quad j = 1 \ldots n, \tag{3.2}$$

where Φ denotes the set of the control points of triangular B-splines. The above equations are treated as hard constraints and have to be strictly satisfied in the following optimization process. In most cases, the linear system of (3.2) is underdetermined. But it is possible to become over-constrained when excessive landmarks are selected on a single spline patch. Two approaches can be used to solve such problem. One is to subdivide the triangular mesh, where there are overly-condensed landmarks, and re-initialize the domain triangulation. The drawback of it is that the problem dimension is increased accordingly. The other approach aims to find a compromised solution for (3.2), which will be discussed later.

3.2.3 Cost Function

In this chapter, we register I_s to I_t using a variational approach, in which a metric measuring image similarity and constraints of global smoothness and local rigidity are combined into an overall cost function E_{total} that is defined as:

$$E_{total} = \alpha E_I + \beta E_R + \gamma E_S, \tag{3.3}$$

where α, β, and γ control the relative influence among three energy terms. In (3.3), E_I is the driving force behind the registration process and aims to maximize the image similarity, whereas E_R is a constraint term to ensure local rigidity and E_S tries to regularize the transformation as smooth as possible.

A number of approaches have been proposed in literature to calculate either similarity or dissimilarity between images. Mutual information [17, 134] and correlation ratio [102] are the methods to measure image similarities, while the sum-of-squared-difference (SSD) measures the dissimilarities. In our current work, we simply use SSD metric to test the feasibility of our registration algorithm. The differences between I_s and I_t, represented by E_I, is evaluated by:

$$E_I = \frac{1}{2}\int_{\Omega} \|I_s(\mathbf{T}(\mathbf{x};\Phi)) - I_t(\mathbf{x})\|^2\, d\mathbf{x}. \tag{3.4}$$

In the theory of continuum physics, the non-rigid transformation is often measured by *Green-St. Venant* strain tensor $\mathbf{E}$ defined as

$$\mathbf{E} = \nabla\mathbf{u} + \nabla\mathbf{u}^T + \nabla\mathbf{u}^T\nabla\mathbf{u} \tag{3.5}$$

Then a necessary and sufficient condition to obtain a local rigid transformation is

$\mathbf{E} = \mathbf{0}$ for rigid structures regions. This is identical to that proposed in [69], where the Jacobian matrices are considered instead. We enforce the rigidity constraint by defining a penalty term as the integral of the *Frobenius* norm of $\mathbf{E}$. Since different structures in the image exhibit different deformation properties, and do not need to deform similarly, we introduce a characteristic function $w(\mathbf{x})$ to separate the rigid objects from deformable regions. The value of $w(\mathbf{x})$ is 1 on rigid structures and 0 elsewhere. The penalty term for local rigidity is given by:

$$E_R = \frac{1}{2}\int_{\Omega} w(\mathbf{x})\,\|\mathbf{E}\|_F^2\,d\mathbf{x}, \tag{3.6}$$

where $\|\cdot\|_F$ denotes the *Frobenius* norm.

A regularization term E_S, measuring the bending energy of a thin plate metal subject to external forces [103, 108], is also incorporated to discourage improbable or impossible transformations. It depends on the 2nd derivatives of the deformation and is written as:

$$E_S = \frac{1}{2}\int_{\Omega} (1 - w(\mathbf{x}))\left(\left\|\frac{\partial^2 \mathbf{u}}{\partial x^2}\right\|^2 + \left\|\frac{\partial^2 \mathbf{u}}{\partial y^2}\right\|^2 + 2\left\|\frac{\partial^2 \mathbf{u}}{\partial x^2}\right\|^2\right) d\mathbf{x}, \tag{3.7}$$

where the function $w(\mathbf{x})$ makes the regularization term valid only over non-rigid regions.

3.2.4 Optimization

The optimization problem is stated to find an ideal $\mathbf{\Phi}$ such that the overall energy (3.3) is minimized with the constraints in (3.2) satisfied. There are various algorithms available to accomplish such constrained nonlinear programming task. In particular, we convert the constrained optimization problem to a unconstrained one, rather than applying established methods directly. Then a simplex line search approach described in [96] is performed to update the parameters iteratively along the steepest descent of gradient until the cost function can not be decreased any further.

Putting (3.1) and (3.2) together, we discretize the point-based constraints and write them in a matrix format:

$$\mathbf{P}+\mathbf{C}\boldsymbol{\Phi}=\mathbf{Q}, \tag{3.8}$$

where $\mathbf{P}$ and $\mathbf{Q}$ are the vectors collecting the landmark positions in I_t and I_s respectively, the vector $\boldsymbol{\Phi}$ consists of the control points of triangular B-splines, and the triangular B-spline basis functions constitute the matrix $\mathbf{C}$ which is extremely sparse and rank-deficient.

By solving the original optimization problem in the Null-Space of $\mathbf{C}$, we can successfully remove the point-based constraints. Then, the new parameter vector $\boldsymbol{\Psi}$ in Null-Space is related to the old one $\boldsymbol{\Phi}$ by the equation:

$$\boldsymbol{\Phi}=\mathbf{N}\boldsymbol{\Psi}+\boldsymbol{\Phi}_0, \tag{3.9}$$

in which $\mathbf{CN}=0$ and $\mathbf{C}\boldsymbol{\Phi}_{\mathbf{0}}=\mathbf{Q}-\mathbf{P}$. We use Gaussian-Jordan-Elimination-like approach proposed in [36] to construct $\mathbf{N}$, and solve for $\boldsymbol{\Phi}_{\mathbf{0}}$ by either singular value decomposition (SVD) or QR decomposition, both of which are computationally viable here, since most columns in $\mathbf{C}$ are zero.

Instead of estimating the gradient of E_{total} using finite-difference approximation, we analytically calculate the derivative with respect to $\mathbf{\Psi}$ and obtain:

$$\frac{\partial E_{total}}{\partial \mathbf{\Psi}} = \mathbf{N}^{\mathbf{T}} \left(\alpha \frac{\partial E_I}{\partial \mathbf{\Phi}} + \beta \frac{\partial E_R}{\partial \mathbf{\Phi}} + \gamma \frac{\partial E_S}{\partial \mathbf{\Phi}} \right), \tag{3.10}$$

where

$$\frac{E_I}{\partial \phi_i} = \int_{\Omega} \left(I_s(\mathbf{T}(\mathbf{x})) - I_t(\mathbf{x}) \right) \nabla I_s(\mathbf{T}(\mathbf{x})) B_i(\mathbf{x}) d\mathbf{x}.$$

Let ϕ_{ij} denotes the $(3i + j)$-th component of $\mathbf{\Phi}$ and $B_{i,j}$ be the derivative of the basis function in j direction, where $j = 1, 2, 3$ for x, y and z coordinates, respectively. The derivative of the local rigidity penalty term is:

$$\begin{aligned}
\frac{E_R}{\partial \phi_{ij}} &= \int_{\Omega} w(\mathbf{x}) \sum_{s,t=1,2,3} \mathbf{M}_{st} \frac{\partial \mathbf{M}_{st}}{\partial \phi_{ij}} d\mathbf{x} \\
\mathbf{M}_{st} &= \sum_{\substack{k=1\ldots m \\ r=1,2,3}} \phi_{kt} B_{k,s} + \phi_{ks} B_{k,t} + \phi_{ks} \phi_{kt} B^2_{k,r} \\
\frac{\partial \mathbf{M}_{st}}{\partial \phi_{ij}} &= \sum_{r=1,2,3} \delta_{jt} \left(B_{i,s} + \phi_{is} B^2_{i,r} \right) + \delta_{js} \left(B_{i,t} + \phi_{it} B^2_{i,r} \right)
\end{aligned}$$

in which δ_{ij} is *Dirac* function which equals to 1 if and only if $i = j$. Likewise, the derivative of the regularization term is given by:

$$\frac{E_S}{\partial \phi_{ij}} = \int_{\Omega} (1 - w(\mathbf{x})) \left(\sum_{s,k=1,2,3} \phi_{kj} B_{k,s} B_{i,s} + \sum_{s,t,k=1,2,3} \phi_{k,j} B_{k,st} B_{i,st} \right) d\mathbf{x},$$

where $B_{i,st}$ stands for the second derivative of basis functions. For details on efficient evaluation of triangular B-spline basis functions and their derivatives, please refer to [86] [35] [45].

Note that the integration operation in (3.3) is performed only on the pixels of ROI. Therefore, we could significantly speedup the registration procedure if all the basis functions and their derivatives over the interested region are pre-computed.

3.3 Experimental Results

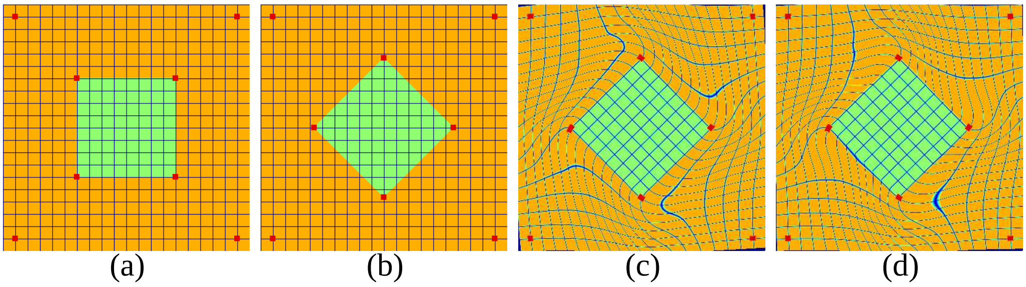

(a) (b) (c) (d)

Figure 3.1: The first experiment: (a) Source image. (b) Target image. (c) Registration result when sharp features are NOT modeled. The minimized energy terms are $E_S = 97.6$, $E_R = 39.9$. (d) Registration result when sharp features are modeled, the minimized energy terms are $E_S = 71.8$, $E_R = 30.8$.

In order to evaluate the feasibility and applicability of the proposed algorithm, we test it on both synthetic and real data. Cubic triangular B-splines are chosen in the experiments to compare with the frequently used cubic tensor-product B-splines.

The first example demonstrated in Fig.3.1 doesn't consider matching image intensities(*i.e.*, $\alpha = 0$ in (3.3)), but tries to align corresponding points instead. A green square is included in the source image to represent a rigid object, and its counterpart is included in the target image with a rotation of 45°. 8 pairs of landmarks are selected at the corners of both the image and the rigid square, and applied as the point-based constraints in the registration. The target image is chosen as the reference domain, which is triangulated into 32 patches, and the cubic triangular B-splines built on it have 361 control points. After applying our algorithm without and with sharp feature modeling respectively, the achieved registration results are plotted in Fig.3.1(c) and Fig.3.1(d). It is noticeable that the background and the square are more smoothly connected in Fig.3.1(c) than in Fig.3.1(d), because they are treated as a single elastic object in the former one, but considered as separate parts in the latter one. It is more physically appropriate to model C^0 continuity between the background and the square, when we simulate the interaction between them. Therefore, the method with sharp feature modeling can achieve better registration result (the minimized energy terms are $E_S = 71.8$, $E_R = 30.8$) than the other one ($E_S = 97.6$, $E_R = 39.9$), when the same parameter setting ($\beta = \gamma = 1$) is used.

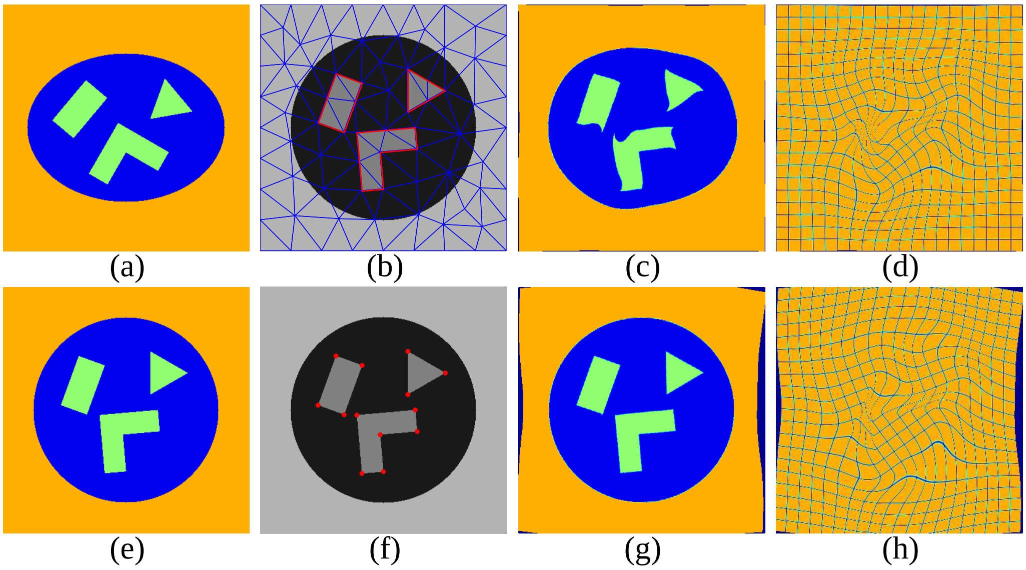


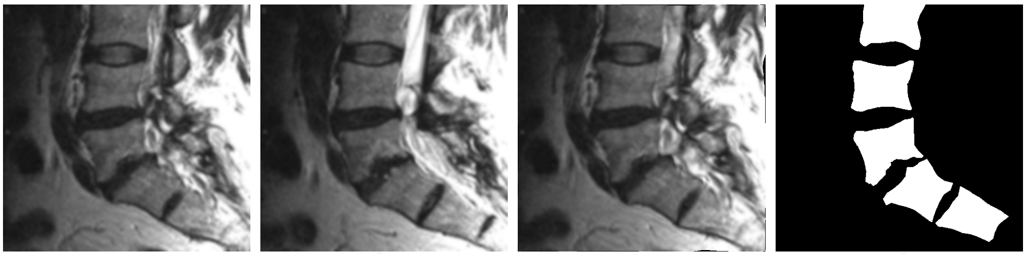
Figure 3.2: The second experiment: (a) Source image. (e) Target image. (b) The domain triangulation with feature lines highlighted in red. (f) 13 landmarks are highlighted in red. (c) Registration result obtained from tensor-product B-spline based method. (g) Registration result obtained from triangular B-spline based method. (d) Deformation recovered using tensor-product B-spline based method. (h) Deformation recovered using triangular B-spline based method.

For the second example, both images (see Fig.3.2(a)(e)) include three geometric objects to represent rigid structures, whose positions are quite different in the source and the target images. The reference domain (shown in Fig.3.2(b)) has

130 triangles and the triangular B-splines thus have 631 control points. 13 pairs of landmarks are picked up to ensure correct alignment between rigid structures (see Fig.3.2(f)). The registration result and the recovered deformation field are shown in Fig.3.2(g) and Fig.3.2(h). An alternative approach using tensor-product B-splines is also applied for the comparison purpose. Its domain is defined on a 25×25 to match the number of triangular B-spline control points. The comparison between the results from both approaches (shown in Fig.3.2(c) and Fig.3.2(g)) indicates that the tensor-product based method fails to align the images at a desired resolution, when there exist large deformations near rigid structures. In sharp contrast, triangular B-spline is built on a flexible domain, so that its modeling power can be ideally concentrated on the interested region for better registration. Furthermore, its power of modeling sharp features helps to improve the registration quality far more better.

Two MRI images of human spines (see Fig.3.3(a)(b)) are used in the third experiment. The spinal bones are first segmented from the target image, then the characteristic function w is set accordingly to decide where the rigidity constraints should be applied. The source image is registered to the target image as shown in Fig.3.3, in which all of the rigid structures are successfully matched.

Our algorithm is implemented using MS VC++, and all experiments are conducted on a platform with 2.8GHz Pentium IV CPU and 1G RAM. Both synthetic images have the size of 400×400, and the size of the MRI images used for the third experiment is 512×512. The running time for the three experiments are about 1 minute, 6 minutes, and 12 minutes respectively.



(a) (b) (c) (d)

Figure 3.3: The third experiment: (a) Source image. (b) Target image. (c) Registration result. (d) Rigid structures segmented from the target image.

3.4 Summary and Discussion

This chapter presents a nonrigid registration technique in which the transformation between corresponding images are represented by triangular B-splines. By preserving feature lines in the domain triangulation and adjusting knots accordingly, the proposed method successfully recovers local rigid motions and accurately simulates C^0 continuities at desired regions, using relatively fewer degrees of freedom and lower degree polynomials. The actual registration is done through the use of a variational framework, in which a constrained optimization problem is solved to reduce the differences between images and enforce both local rigidity and global smoothness at the same time. The method has been tested on both synthetic examples and real data for its efficacy.

Although tensor-product B-spline based approaches are still dominating in the field of non-rigid registration, their applicability is somehow limited due to the structure of their regular domain. On the contrary, our registration method can correctly delineate the boundaries of rigid bodies in its domain triangulation at a much coarser level, and thus model the local rigid motions more accurately. Furthermore, with the degenerate knots on the boundaries of rigid structures, C^0 continuity is automatically guaranteed in the described displacement field, and can be naturally coupled with the optimization process. This advantage enables us to precisely simulate the behavior of rigid objects inside elastic tissues. From the registration point of view, the deformable regions around the rigid structures may become less constrained by the regularization term and contribute extra flexibility to the minimization of the cost function (3.3). As a result, the registration quality can be considerably improved. An alternative way to model C^0 continuities could be to separate rigid and non-rigid regions into different domain pieces. However, extra efforts must be spent to keep the overall transformation consistent across different pieces in a different hierarchy, and in general, the variational approaches over irregular domains in a hierarchical fashion have not been fully explored. In this chapter, only rigid structures with simple geometric shapes are considered in our experiments for the feasibility test. To accommodate more complicated structures, we can subdivide the domain mesh adaptively along their boundaries until the desired accuracy is achieved. The landmarks applied in our registration are interactively selected by users based on their knowledge and subjectivity. Naturally, the registration result is affected by the quality of landmark selection.

Chapter 4

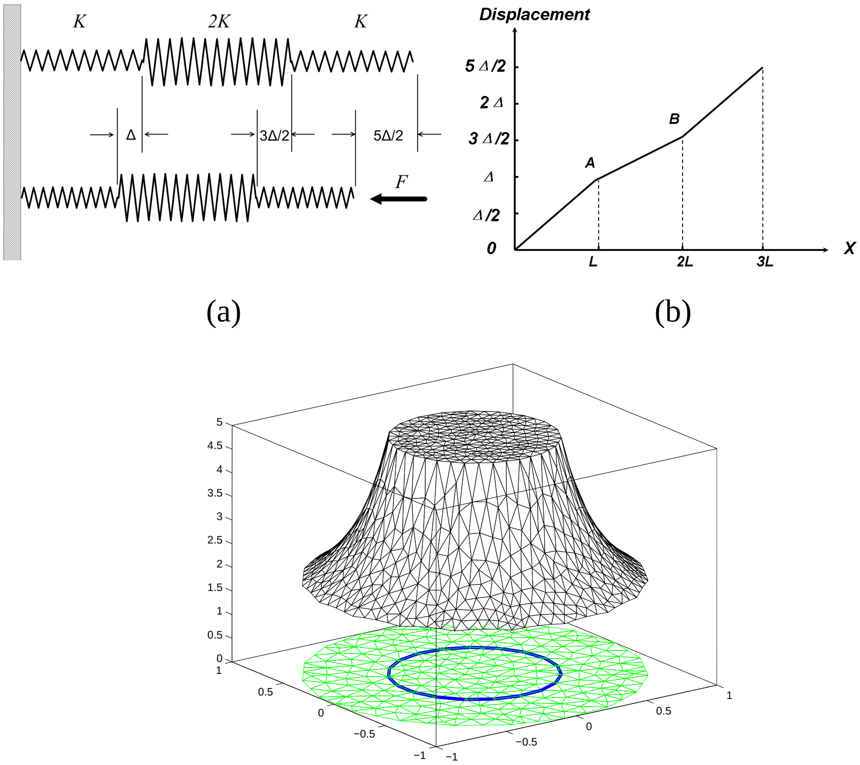
Registration of Temporal Mammogram Using Triangular B-splines Finite Element Method (TBFEM)

4.1 Motivation

Breast cancer is one of the most common causes for cancer-related death, with annual mortality of over $400,000$ women worldwide. Taking regular mammographic screening and comparing corresponding mammogram are necessary for early detection of breast cancer, which is also the key to successful treatment. To seek abnormality through comparison, the clinical diagnosis involves either pairs of mammogram from the bilateral breasts of the same patient or a series of mammogram acquired from the same breast at different time. However, the first method tends to be unreliable when the left and right breasts contain significantly different structures. But the latter one, which aims at detection of temporal changes in the same breast, produces more robust results. Unfortunately, a temporal pair of mammogram may vary quite significantly due to the spatial disparities caused by the variety in acquisition environments, including 3D position of the breast, the amount of the pressure applied, etc. Such disparities can be corrected through the process of *Temporal Registration*. This chapter contributes to the existing state of the art in temporal registration of digital mammography.

Earliest attempts for mammogram registration typically assumed rigidity and affinity of breast deformation. Yin *et al.* [156] align mammograms using an optimal rigid transformation which minimizes the least square error between two group of control points. However, due to the elastic nature of the breast, it's far from correct to match mammograms using solely rigid models. Later, radial basis functions (RBF) based on Thin-plate Spline (TPS) [110] and Cauchy-Navier Spline (CNS) [148] are incorporated to build a global smooth nonrigid transformation from a local displacement vector field representing spatial differences between corresponding control points. However, those registration method depending on control points are prone to failure when the pre-segmentation is accurate enough. To this end, recent techniques tend to incorporate the metrics measuring intensity similarities between corresponding images. Writh *et al.* [149] align subregions according to local mutual information, then combine them into a global transformation using TPS. Hadjiiski *et al.* [42] propose an automatic regional registration method, which bases on the identification of corresponding lesions in temporal mammogram pairs. A pyramid-based multiresolution technique given by Kostelec *et al.* [60] integrates a least square measurement with TPS transformation to match bilateral mammograms. Likewise, Rueckert *et al.* [109] hierarchically match corresponding breast images, but using a B-spline based free-form deformation(FFD) technique instead. Due to the large variety of breast tissues and their mechanical behaviors, however, it's more appropriate to register temporal mammograms by utilizing model-driven simulation. Pathmanathan *et al.* [83] build a patient-specific nonlinear 3D model to predict the tumor location. Kita *et al.* [59] simulate the deformation of breasts, and accordingly establish the correspondences between their CC and MLO mammographic views. Richard *et al.* [101] [100] build a 2D FEM model from X-ray mammograms, then conduct the registration by deforming it subject to both feature and intensity-driven constraints.

Our approach is inspired by Richard *et al.*'s work [101]. But we employ a novel triangular B-spline finite element method (TBFEM) instead and recover large deformation between temporal mammograms following nonlinear elasticity theory. Triangular B-splines, introduced by Dahmen *et al.* [20], has many favorable features, such as flexible simplex-based domain, space-varying continuities, local control, etc. The most unique one of them is the ability to model local sharp features along in the approximated smooth solution. A example of this is illustrated in Figure.4.1(c). Therefore, the incorporation of TBFEM within our registration framework gives the following advantages over conventional FEMs: i) The region of interest (ROI) can be accurately described and selected as the registration domain, while the tensor-product B-splines methods [109] necessitate extra efforts to refine the problem domain along the irregular boundaries of ROI. ii) In contrast to other simplex-based elements, such as *Lagrange Polynomials*, our TBFEM offers a global smooth solution (Note that the continuities of the solutions given by traditional Lagrange FEM are not ensured across the element boundaries). iii) Users are allowed to model the spatially varying continuities in the approximated deformation field by manipulating the knot configuration according to pre-identified features. Sharp features may appear in the displacement field when the elastic object contains different materials (see Figure.4.1(a)(b) for this concept). iv) The accurate simulation of the elastic deformation incorporating material heterogeneity can be achieved by using relatively fewer finite-elements.



(a) (b)

(c)

Figure 4.1: (a) A one dimensional cascaded spring system system consisting of three springs with stiffness of *k*, 2*k* and *k* respectively. When the system is compressed by an external force **F**, it is deformed and the displacement caused along *x*-axis is plotted in (b). The sharp features *A* and *B* in the displacement profile are built at the joint points between different springs. (c) A functional approximated by a single triangular B-spline, where both C^1 and C^0 continuities co-exist.

4.2 Method

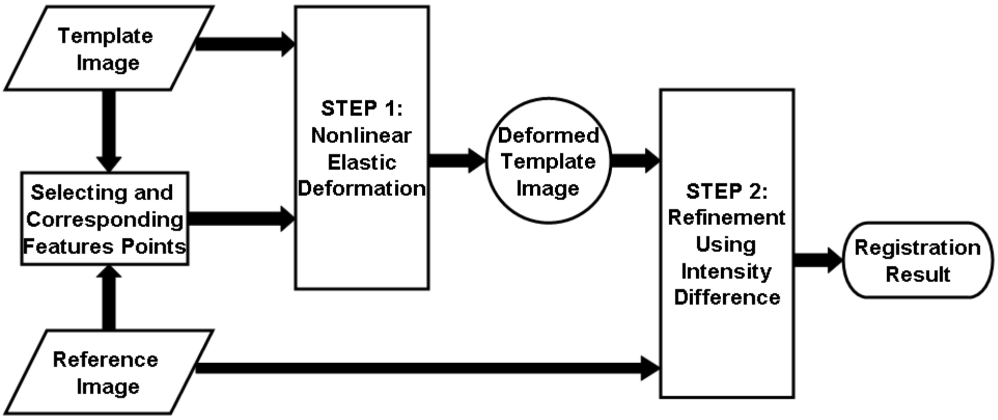


Figure 4.2: Overview of the registration process.

The temporal registration problem can be stated as: given both previous mammogram $\mathcal{T}$ (*template*) and current one $\mathcal{R}$ (*reference*), we are asked to find an optimal transformation ϕ such that the disparities between them are reduced maximally. Our registration process(see Figure.4.2) consists of two consecutive steps, whose details will be developed in the following.

4.2.1 Nonlinear Elastic Deformation

First, the registration domain Ω is defined over the breast region, which is previously segmented from $\mathcal{R}$ and usually circumvented by the breast skin contour and partial image boundaries. The domain Ω is then triangulated with the user-specified feature lines(see Figure.4.3(a)). Then we adjust the knots configuration such that the knots collapse to their adjacent sharp features. Thus, C^0 continuity will be successfully modeled in the solution of the recovered displacement field.

The objective of this registration step is to estimate the displacement field $\mathbf{u}^* : \bar{\Omega} \to \mathbb{R}^2$ between $\mathcal{T}$ and $\mathcal{R}$ such that they are aligned as much as possible. Since the breasts compressed during mammography usually undergo large deformation ($> 5\%$), our framework follows nonlinear elastic theory and governs $\mathbf{u}$ by:

$$\begin{cases} \mathbf{A}(\mathbf{u}) = -\nabla \cdot (\mathbf{I} + \nabla \mathbf{u})(\lambda tr(\mathbf{E}(\mathbf{u}))\mathbf{I} + 2\mu \mathbf{E}(\mathbf{u})) = \mathbf{f} \\ \mathbf{u} = \mathbf{u_0} \quad \text{on} \quad \Gamma^d \\ (\mathbf{I} + \nabla \mathbf{u})(\lambda \mathbf{tr}(\mathbf{E}(\mathbf{u}))\mathbf{I} + \mathbf{2}\mu \mathbf{E}(\mathbf{u})) = \mathbf{g_0} \quad \text{on} \quad \Gamma^n \end{cases} \tag{4.1}$$

in which *St. Venant-Kirchhoff* elastic material is assumed and *Green-St. Venant strain tensor* $\mathbf{E}(\mathbf{u})$ is written in its second order:

$$\mathbf{E}(\mathbf{u}) = \frac{\mathbf{1}}{\mathbf{2}}(\nabla \phi^{\mathbf{T}} \nabla \phi - \mathbf{I}) = \frac{\mathbf{1}}{\mathbf{2}}(\nabla \mathbf{u}^{\mathbf{T}} + \nabla \mathbf{u} + \nabla \mathbf{u}^{\mathbf{T}} \nabla \mathbf{u}) \tag{4.2}$$

The body force $\mathbf{f}$ does not exist in this step. And λ and μ are *lame* coefficient related to elastic properties.

Due to the nature of registration problems, both *Dirichlet* and *Neumann* conditions in Equation.(4.1) are dropped, and replaced by a bunches of discretized geometric constraints [101]. Such constraints in our framework consist of two set of control points $\mathcal{P}$ and $\mathcal{Q}$, selected from $\mathcal{R}$ and $\mathcal{T}$ respectively. The majority of the control points come from the breast skin contours and their correspondences are established following the approach proposed by Wirth [147]. Furthermore, salient anatomical structures(vessels branches, nipple, etc.) and pathological points(microcalcification, etc.) are ideal to serve as auxiliary control points. Different from the control points automatically matched on the breast contour using arc-length parametrization, the interior points need to be manually selected and corresponded to each other in our current implementation. Note that such process can be automated by incorporating the technique proposed by Paquerault *et al.* [82]. An example of the control points selected from both the template and reference images is shown in Figure.4.3(b)(c).

The geometric constraints given by $\mathcal{P}$ and $\mathcal{Q}$ are formulated as follows:

$$\phi(\mathbf{p_i}) = \mathbf{q_i} \quad \mathbf{p_i} \in \mathcal{P}, \mathbf{q_i} \in \mathcal{Q}, \mathbf{i} = 1, \ldots, |\mathcal{P}| \tag{4.3}$$

Where the $\mathbf{p_i}$ and $\mathbf{q_i}$ are corresponding points. These constraints can be viewed as a bunch of displacement vectors from $\mathcal{R}$ to $\mathcal{T}$.

To solve Equation.(4.1), we linearize it with *Newton's Method* [55], thus the solution can be approximated incrementally by:

$$\begin{aligned} \mathbf{A}(\mathbf{u^{n+1}}) &\approx \mathbf{A}(\mathbf{u^n}) + \mathbf{A}'(\mathbf{u^n})\delta\mathbf{u^n} = \mathbf{f^n} + \delta\mathbf{f^n} \\ \delta\mathbf{f^n} &= \mathbf{f^{n+1}} - \mathbf{f^n} = \mathbf{A}(\mathbf{u^{n+1}}) - \mathbf{A}(\mathbf{u^n}) \\ \delta\mathbf{u^n} &= \mathbf{u^{n+1}} - \mathbf{u^n} \end{aligned} \tag{4.4}$$

in which the total displacement field $\mathbf{u}$ is iteratively updated with the increment of $\delta\mathbf{u^n}$, which in turn is the solution of:

$$\mathbf{A}'(\mathbf{u^n})\delta\mathbf{u^n} = \mathbf{f^n} \tag{4.5}$$

Finally we discretize Equation.(4.5) in our TBFEM model and solve it in the approximate space $\hat{V} := \text{span}\{B_1, B_2, \ldots, B_N\}$, where B_i denotes the triangular B-spline shape function. After applying *Galerkin Method* [15], we achieve its discretization form written by:

$$\sum_{k=1}^{N} \delta\hat{u}_k^n \int_\Omega \sum_{i,j,p,q=1}^{3} \hat{a}_{ijpq}(\mathbf{u^n})\partial B_k \partial B_l dx = 0 \quad l = 1, \ldots, N \tag{4.6}$$

in which $\delta\hat{\mathbf{u}}^{\mathbf{n}} = \sum_{k=1}^{N} \delta\hat{u}_i^n B_i$ and $\hat{a}_{ijpq}$ denotes a FEM discretization operator, the detail of which is available in [15]. The equation above is ill-conditioned unless combined with constraints given in Equation.(4.3). Suppose that there are total M steps in the elastic deformation, the constraints contributed to the deformation at the n^{th} step are:

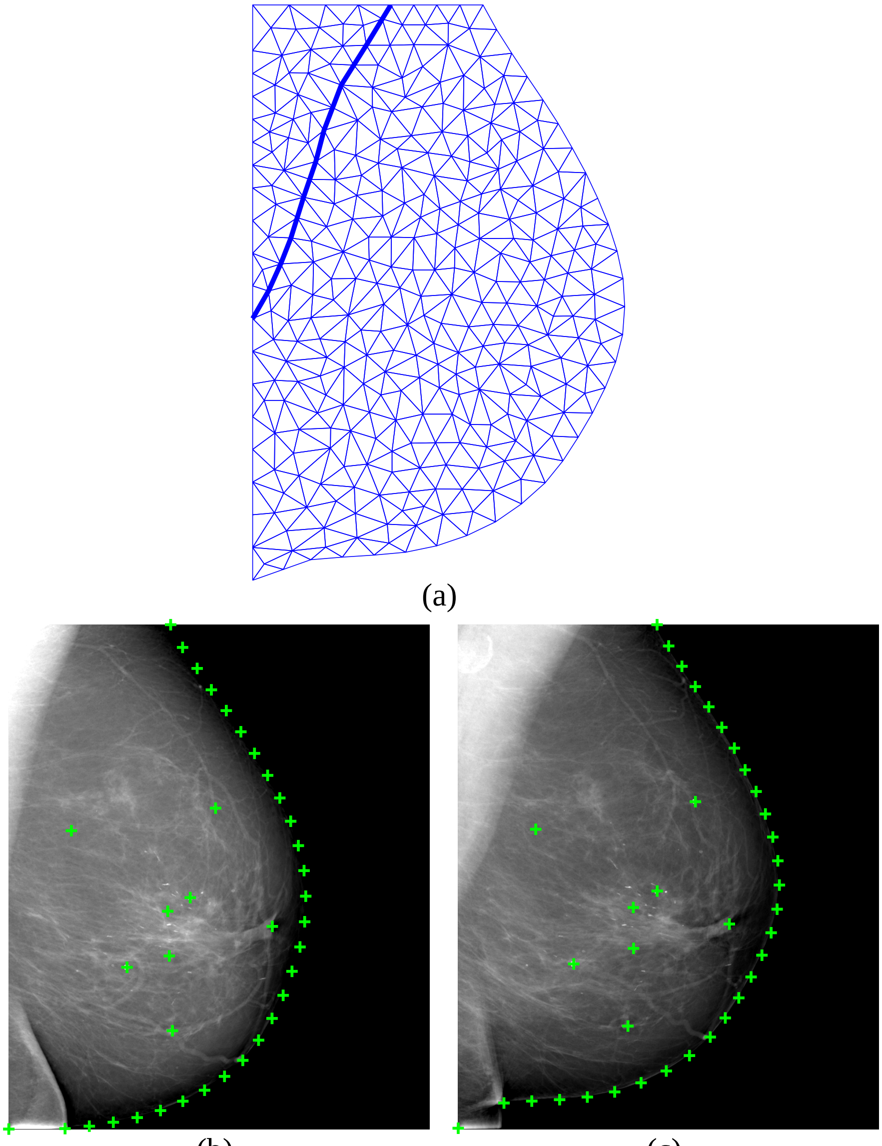
$$\sum_{j=1}^{N} B_j(\mathbf{p_i})\delta\hat{u}_j^n = \frac{1}{M-n+1}(\mathbf{q_i} - \mathbf{u^n}(\mathbf{p_i})) \quad i = 1, \ldots, |\mathcal{P}| \tag{4.7}$$

In essence, the constraints above progressively drag the control points in $\mathcal{R}$ to their corresponding location in $\mathcal{T}$. A linear-interpolation scheme is employed here to predict the position of control points of the next time step. Note that alternative schemes can be incorporated as well.

Substituting Equation.(4.7) into Equation.(4.5), we can get a constrained linear problem:

$$\begin{aligned} \mathbf{M^n \Delta U^n} &= 0 \\ \text{s.t.} \quad \mathbf{C \Delta U^n} &= \mathbf{D}(\mathbf{U^n}) \\ \mathbf{U^{n+1}} = \mathbf{U^n} + \mathbf{\Delta U^n} \quad \text{and} & \quad \mathbf{U^0} = 0 \end{aligned} \tag{4.8}$$

In current implementation, we first convert the problem above to an unconstrained system by using *Null-space project* technique [36], then solve it by Conjugate Gradient method(CG). Note that if there is only one time-step assumed for the simulation, the deformation will degenerate to a linear elastic one.



(a)

(b)

(c)

Figure 4.3: (a) The triangulated registration domain Ω, where the feature line is highlighted between the pectoral muscles and breast tissue. (b)(c) 8 interior and 29 contour control points in template and reference images respectively.

4.2.2 Refinement with Intensity Difference Minimization

Let the displacement field obtained in the previous step be $\mathbf{u}^*$, the deformed $\mathcal{T}$ denoted by $\mathcal{T}'(\mathbf{x}) = \mathcal{T}(\mathbf{x}+\mathbf{u}^*)$ which is in rough alignment with $\mathcal{R}$. To further improve the registration results, we perturb $\mathcal{T}'$ with an additional displacement field $\mathbf{v}$ by minimizing a intensity-based metric, which measures the sum of squared difference (SSD) between $\mathcal{T}'$ and $\mathcal{R}$:

$$\mathcal{E}(\mathcal{T}', \mathcal{R}; \mathbf{v}) = \alpha \frac{1}{2} \int_{\Omega} \|\mathcal{T}'(\mathbf{x}+\mathbf{v}) - \mathcal{R}(\mathbf{x})\|^2 d\mathbf{x} \tag{4.9}$$

where α weighs the contribution of $\mathcal{E}$.

Similar to Equation.(4.1), the governing PDE in this step can be written as:

$$\mathbf{A}(\mathbf{v}) = -\nabla \cdot (\lambda tr(\mathbf{E}(\mathbf{v}))\mathbf{I} + 2\mu \mathbf{E}(\mathbf{v})) = \mathbf{f} \tag{4.10}$$

Note that we assume linear elasticity here because only small deformations are allowed in the second step. Thus, the second order strain tensor is dropped and $\mathbf{A}$ becomes a linear differential operator. Instead of using geometric constraints, we incorporate artificial body forces derived from the minimization of $\mathcal{E}$. In addition, we pin the images at both the upper and lower left corners to avoid unnecessary floating. Consequently, Equation.(4.10) is discretized to:

$$\mathbf{M}^{\mathbf{0}}\mathbf{V} = \mathbf{F}(\mathbf{V}) \tag{4.11}$$

where $\mathbf{F}(\mathbf{V})$ denotes the virtual body force, whose elements are:

$$(\mathbf{F}(\mathbf{V}))_i = \alpha \int_{\Omega} (\mathcal{T}(\mathbf{x}+\mathbf{v}) - \mathcal{R}(\mathbf{x})) \nabla \mathcal{T}(\mathbf{x}+\mathbf{v}) B_i d\mathbf{x} \tag{4.12}$$

Note that the stiffness matrix $\mathbf{M^0}$ is identical to that in Equation.(4.8) at the previous registration step. A gradient descent like algorithm is employed to solve it:

$$\begin{aligned} \mathbf{M^0}\boldsymbol{\Delta}^{\mathbf{n+1}} &= \mathbf{F}(\mathbf{V^n}) && (4.13) \\ \mathbf{V^{n+1}} &= h\boldsymbol{\Delta} + (1-h)\mathbf{V} && (4.14) \end{aligned}$$

associated with a small positive value h. The iteration stops when a predefined threshold ε is met:

$$\frac{\mathcal{E}(\mathcal{T},\mathcal{R};\mathbf{V^{n+1}})}{\mathcal{E}(\mathcal{T},\mathcal{R};\mathbf{V^n})} < \varepsilon$$

4.3 Experiment and Results

Two temporal pairs of 2D X-ray mammograms in MLO views obtained with one year interval (see Figure.4.4 and Figure.4.5) are used to test the registration framework proposed in this chapter. All of the mammograms have size of 2294×1914, resolution of 94.1μm and 12-bit intensity depth. To suppress the noise as well as speedup the registration process, a Gaussian filter with a kernel of 200 pixels is applied to the mammograms before the registration.

The breast region is automatically segmented from each image based on a threshold which is the value of the gray-level corresponding to the first peak in the smoothed histogram of the image. Our registration domain is then defined over the breast region, which is further delaunay-triangulated with pre-identified sharp features as geometric constraints [119].

For both pair of temporal mammograms, we approximate the underlying displacement field using second order triangular B-splines, in which sharp features are naturally accommodated. A simple heterogeneous model is incorporated with

TBFEM to model different elastic material in mammograms. The Young's Modules of $E = 10^4$ and $E = 10^2$ are assigned to pectoral muscles and breast region respectively to model stiff and soft tissue. These values are chosen empirically and only their ratio matters in the first registration step, in which there are no Neumann conditions involved [28]. We choose poisson ratio ν as 0.495 to simulate incompressible breast tissues.

During first registration step, control points are selected automatically on the parameterized breast skin contours and manually from the breast interior region. 29 pairs of contour points and 8 pairs of interior points are picked for the first experimental case, while 33 and 6 pairs respectively for the second. The total nonlinear deformation is divided into 20 time steps; In the second step, the regularization term, weighted by the Young's Module E, counteracts the artificial image forces, whose magnitude is controlled by the coefficient α. Choosing correct values for these registration parameters is essential to the success of our algorithm. In our case studies, we empirically set α to 10^{-3} and h to 10^{-2}, and receive satisfactory results as well as adequate numerical stabilities. The registration result of both experiments are illustrated in Figure.4.4. and Figure.4.5.

To evaluate our approach, we quantitatively compare it with two other similar registration methods. One of them is also based on TBFEM but without sharp features modeling (*i.e.*knots are all fixed), the other uses second order *Lagrange* triangular elements instead. The experimental results are documented in Table.4.2 where the registration qualities are measured by the post-registration improvement between template and reference images. It is noticeable that the pre-registration disparity is mainly reduced in the deformation step while the second registration step makes only small contribution to the final result. The reason is that the salient information provided by X-ray mammograms are apt to trap the intensity-based optimization within local minima. We also find that the registration quality in our framework can be improved by incorporation of prior knowledge of feature lines. However, these improvements seem small in Table.4.2 because of the massive pre-registration error between template and reference images. Compared with conventional triangular FEM (Lagrange triangular FEM) with the same degree of freedom (DOF) and degree of order, TBFEM can delineate the recovered deformation field more accurately, and thus is superior in our simulation-based registration framework.

Our algorithm was implemented with MSVC++. The experiment was performed on a platform with 3GHz CPU and 4G RAM. Each step of nonlinear elastic deformation takes up to 10 seconds. At refinement stage, we used a set of images re-sampled from the original mammograms with different resolutions. The performances of refinement step working on multi-resolution images are documented in Table 4.1.

Image size	# of Iterations steps	Timing
2294×1914	28	157m
1147×957	21	22m
573×478	16	7m19s
286×239	12	92s

Table 4.1: Statistics of refinement step applied on multi-resolution mammogram images (case 1)

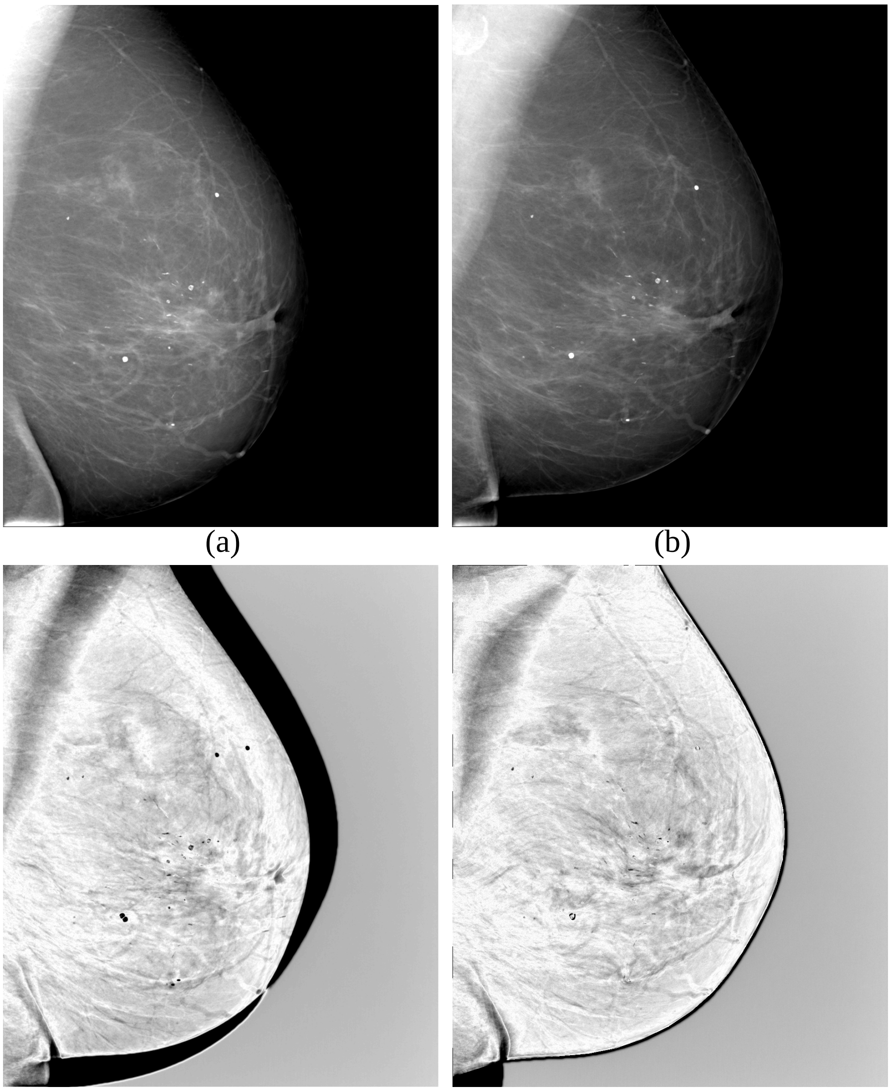
4.4 Summary

In this chapter, we presented a simulation-based registration framework for temporal pair of 2D x-ray mammograms. A novel triangular B-spline finite element method(TBFEM) is incorporated to accurately model the recovered deformation,

		TBFEM with feature modeling	TBFEM without feature modeling	2nd-order Lagrange triangular FEM
Case 1	step 1	86.41%	86.29%	85.93%
	step 2	86.45%	86.32%	85.97%
Case 2	step 1	94.69%	94.66%	94.56%
	step 2	94.72%	94.69%	94.58%

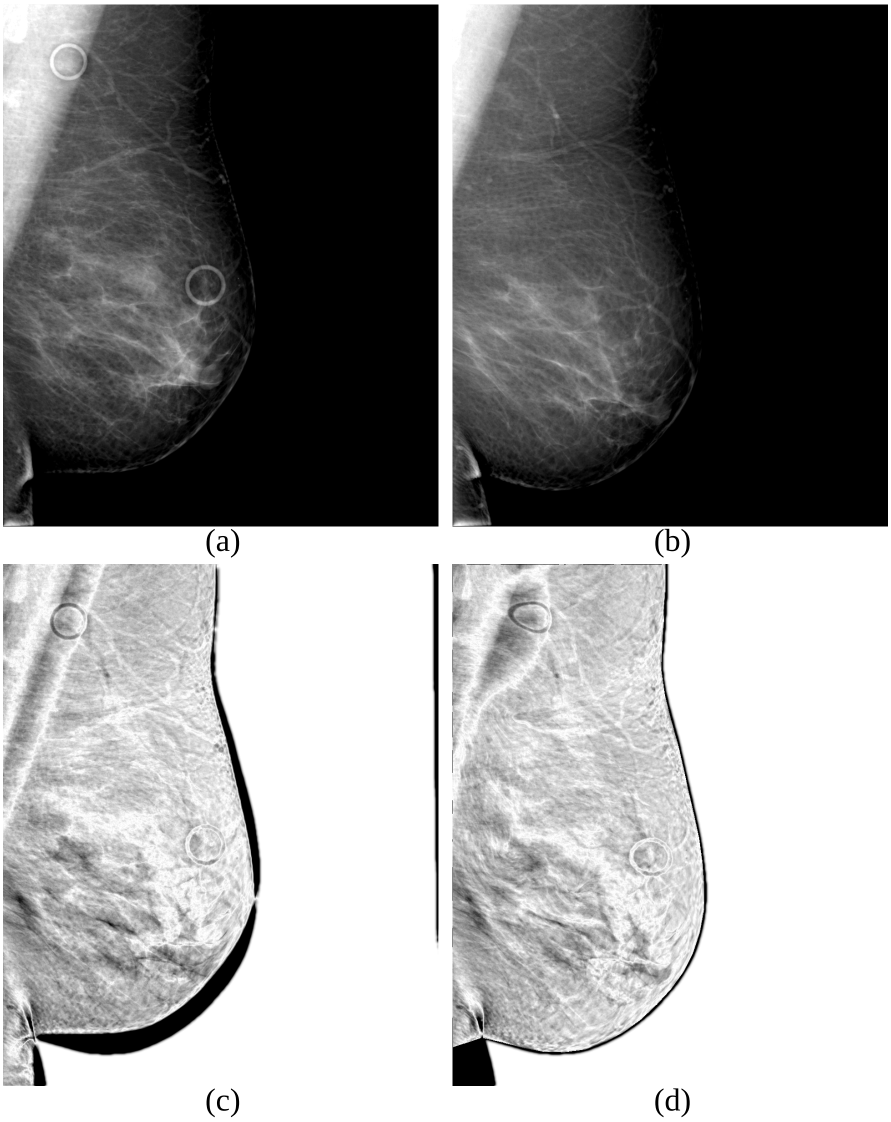
Table 4.2: The registration quality is measured by the post-registration improvement between template image $\mathcal{T}$ and reference image $\mathcal{R}$, which is formulated as $(|\mathcal{T}-\mathcal{R}|^2-|\mathcal{T}_*-\mathcal{R}|^2)/|\mathcal{T}-\mathcal{R}|^2$ where $\mathcal{T}_*$ denotes the new $\mathcal{T}$ after the registration is conducted. In Case-1, there are 2514 DOFs with 600 triangular elements, and for Case-2, there are 2600 DOFs with 619 triangles.

as well as the sharp features between different tissue properties using the technique of knots collapsing. Our registration algorithm employs a two-stepped scheme: the massive disparities between temporal mammograms are first reduced through a nonlinear elastic simulation; then the mapping between template and reference images is further refined according to intensity-based information. The results of our experiment have also shown that the TBFEM incorporated with our framework is superior to traditional FEM method by improving registration quality considerably.



(a) (b) (c) (d)

Figure 4.4: Image registration for case one (a) Template Image $\mathcal{T}$ (b) Reference Image $\mathcal{R}$ (c) Pre-registered error (d) Post-registered error after two-step registration technique is applied



(a) (b)

(c) (d)

Figure 4.5: Image registration for case two (a) Template Image $\mathcal{T}$ (b) Reference Image $\mathcal{R}$ (c) Pre-registered error (d) Post-registered error after two-step registration technique is applied

Chapter 5

Automatic Registration of Mammograms using Anisotropic Features

5.1 Motivation

Breast cancer is one of the most common causes for cancer-related death, with annual mortality of over $400,000$ women worldwide. Taking regular mammographic screening and comparing corresponding mammogram are necessary for early detection of breast cancer, which is also the key for successful follow-up treatment. However, the comparative analysis can be difficult because of the great variability in the appearance of mammograms. Therefore, the technique of *registration* is often applied to reduce the spatial disparity between mammogram pairs during Computer Aided Diagnosis (CAD).

The earliest attempt [156] for mammogram registration typically assumed rigidity and affinity of breast deformation. Nevertheless, due to the elastic nature of the breast, it is much more appropriate to match mammograms using nonrigid models. The warping techniques based on Thin-plate Spline (TPS) [110] and Cauchy-Navier Spline (CNS) [148] is widely used to estimate a global nonrigid transformation from local spatial differences between corresponding control points. Since the accuracy of control points extraction is usually not trustworthy, the differences between image intensities are often considered as the matching criterion for better registration [149]. However, mammogram images are frequently containing excessive disordered texture features, therefore the optimization for intensity-based registration tends to get trapped in local minima, unlikely to yield satisfactory results.

In this chapter, we present an automated framework for mammographic registration, which is inspired by the work of [111]. Instead, we invent a novel method to match breast skin boundaries, and apply an accurate unwarping technique which matches both the positions and anisotropic attributes of selected landmarks simultaneously. The breast region is first segmented with the technique proposed in [81], then the skin contours are smoothed by using cubic B-splines approximation. To robustly match the corresponding breast boundary points, we seek an optimal transformation such that the mutual information given by the curvature functionals of both skin contours is maximized. Then, the texture-based feature points associated with orientation attributes are selected from the interior region of breast images using Gabor filters, and then matched appropriately across two images. The extracted feature points can be naturally characterized by the way that they are more distinguishable from surrounding pixels than the others. Finally, we extend the warping technique initially proposed in [105], in order to integrate both the orientation and intensity information in our imaging framework for better local registration.

5.2 Extraction of Breast Boundary

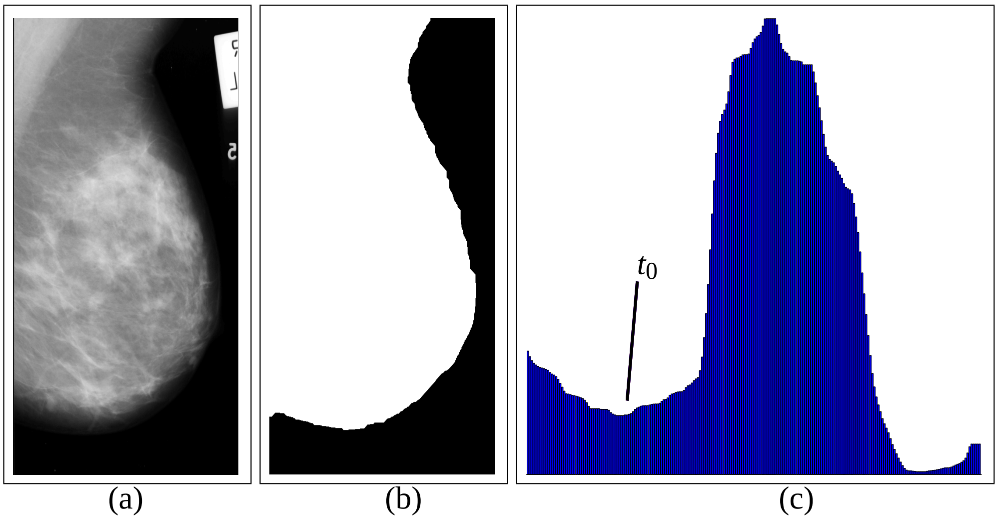


Figure 5.1: (b) shows the segmented breast region (white) from the mammogram image (a). (c) displays the histogram H of the mammogram, where the threshold value denoted by t_0 is selected at the first valley left to the maximum histogram value.

Our goal is to obtain the breast boundary by segmenting the breast region from the mammogram. We use histogram thresholding technique to distinguish the bright breast region from the dark background. An example is given in Fig.5.1, where the indicated threshold t_0 identifies the pixels left to it as the background, while the others as the breast region. It's obvious that the success of the segmentation largely depends on how the threshold value t_0 is chosen. A number of strategies to decide the histogram threshold can be found in the literature [21, 64]. In our implementation, we select the threshold t_0 as follows: First, the lowest and highest bins of the histogram H of the mammogram are purposely discarded since they account for the background noise. Then, H is further smoothed by applying a low-pass filter, e.g. median filter or mean filter. After identifying the maximum histogram value p_b according to the approach proposed in [81], we choose the threshold t_0 as the first valley value left to p_b. A example of the segmented breast region using the threshold t_0 is shown in Fig.5.1(b).

Since the border of the breast region segmented as above is rather noisy, we smooth it by sequentially applying a pair of morphological operations, *closing* and *opening*. An octagonal-shaped operator with radius of 10 is used to preserve the shape of breast region and gives the best smoothing result. Then we can extract the breast skin contour by traveling along the border pixels of the segmented breast region. If the contour is not smooth enough, we will filter it by using cubic B-spline approximation method.

5.3 Matching Boundary Points

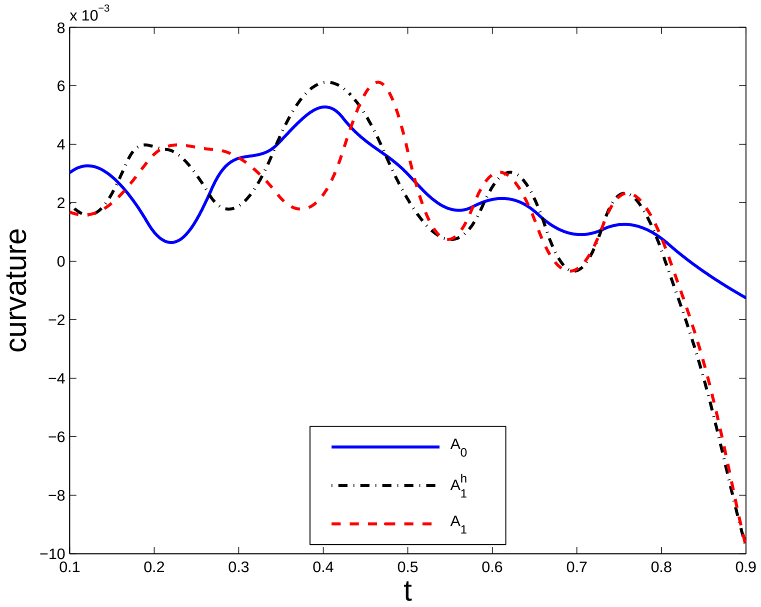


Figure 5.2: A_0, A_1 represent the discretized curvature functions of C_0 and C_1. A_1^h is the curvatures of C_1 after being stretched.

The deformation of breast skin contours determine how the interior deformation take place to a large degree. Rather than match the boundary points with the assumption of local linear stretching [111], we treat the skin contours as nonlinearly stretchable and align the points on them by maximizing the mutual information between the corresponding curvature functionals.

Let C_0, C_1 represent the corresponding boundary contours, and c_0, c_1 denote their curvature functions, respectively. To facilitate the registration process, C_0, C_1, c_0 and c_1 are uniformly parameterized to the domain $[0,1]$, and treated as functionals. We further convert the value of c_0,c_1 into finite bins $[1\ldots M]$, thus obtain their discretized functional representation as A_0 and A_1 (see Fig.5.2). It is obvious that if both A_0 and A_1 are considered as one dimensional images, the problem of matching boundary points can be converted to one dimensional registration problem. That is, given the discretized curvature functions A_0 and A_1, we are asked to find an optimal transformation (or stretching) $h : [0,1] \to [0,1]$ such that

$$h = \arg\min \left(-MI(A_0, A_1^h) + \lambda \int_0^1 \left| \frac{\partial}{\partial t}(h-t) \right|^2 dt \right) \tag{5.1}$$

where the stretched curvature function, denoted by A_1^h, is equal to $A_1(h(t))$, and the mutual information MI is:

$$MI(A_0, A_1^h) = -\sum_{i=1}^{M} \sum_{j=1}^{M} p_h^{01}(i,j) \log \frac{p_h^{01}(i,j)}{p^0(i) p_h^1(j)}$$

in which p^0, p_h^1 represent the corresponding marginal probabilities in A_0 and A_1^h, and p_h^{01} denotes the joint probability of (i,j) between A_0 and A_1^h. In equation (5.1), the minimization of the first term MI aims to matching the boundary contours according to the likelihood between their curvatures, while the second term discourages undesirable transformations, which helps to improve the numerical stability.

To decrease the dimension of the optimization problem stated in equation (5.1), we represent h by using cubic B-splines approximation. And the error function is minimized through the gradient descent method. Note that it's non-trivial to calculate the gradient of the mutual information *MI* because it depends on the discretized bin values, thus is discontinuous. However, this obstacle can be tackled by estimating *MI* with *Parzen window* technique, and an efficient method to evaluate the gradient can be found in work documented in [48].

5.4 Extraction and Matching of Texture Features

In order to recover the local deformation in the interior of the breast region, it's desirable to extract texture-based features and match them between corresponding mammograms. Similar to the selector introduced in [111], where the features with rotation and invariant properties are extracted and steerable filters are used, we propose to employ Gabor filters to detect those features, because they have been reported more robust and more responsive to oriented features [3] than steerable filters.

The real Gabor filter kernel oriented at angle $\theta = -\pi/2$ is defined as:

$$g(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)\right] \cos(2\pi f x) \tag{5.2}$$

where the parameters σ_x, σ_y and f are decided from the following rules: Let τ be the full-width at half-maximum of the Gaussian term along x axis. Then, $\sigma_x = \tau/2.3$, and $f = 1/\tau$. The value of σ_y is defined as $\sigma_y = l\sigma_x$, where l denotes the elongation of the filter along y axis. In current implementation, we empirically set $\tau = 5$ and $l = 5$.

The kernels at other angles can be obtained by rotating (5.2) over the range $[-\pi/2, \pi/2]$. In our experiment, we used a filter bank of Gabor filters, $g_k(x,y), k =$

$0,1,\cdots,15$, oriented at the angles of $\alpha_k = -\pi/2 + \pi k/16$. If the a image $I(x,y)$ is processed, the filtered images become $W_k(x,y) = (I * g_k)(x,y)$, where the asterisk operator denotes linear convolution.

Due to the huge dimension of mammograms, it's not necessary to detect feature points over the entire breast region. Instead, we constrain the selection done only on certain points, which are decided by *Canny Edge Detection* technique. Since many overlapping structures exist in mammogram images, the local estimation of feature orientations is not reliable. To this end, we propose that the response $S_k(x,y)$ at pixel (x,y) to the kth Gabor filter is measured as the average of neighboring responses, and defined by:

$$S_k(x,y) = \frac{1}{|N|} \sum_{i \in N} |W_k(x_i, y_i)|^2 \tag{5.3}$$

where N represents a 5×5 neighborhood. Let $S_{k_1} > S_{k_2} > S_{k_3}$ be the first three largest magnitude of responses in descending order at position (x,y), and the corresponding angles are $\gamma_1(x,y) = \alpha_{k_1}$, $\gamma_2(x,y) = \alpha_{k_2}$ and $\gamma_3(x,y) = \alpha_{k_3}$. To find the bifurcate structures among all candidate points, we check for the following condition:

$$\frac{S_{k_1} - S_{k_2}}{S_{k_2} - S_{k_3}} < 0.1$$

If it's satisfied at a certain point (x,y), we consider there exists a bifurcate structure.

Let P and Q be the set of the junction points detected on both mammogram images. The correspondence between between P and Q can be established as follows:

1. Let both mammogram images denoted by I_0 and I_1. We estimate two approximate transformations T and T^{-1} between them, where T mappers from

I_0 to I_1, while T^{-1} is its inverse. Note that these mappings can be easily constructed using thin-plate spline approximation, using the boundary correspondences that have been established in section (5.3).

2. If either P or Q is empty, stop.

3. Pick $\mathbf{p}$ as the point with the maximum value of S_{k_1} from P. Denote its principal and secondary orientation by $\gamma_1(\mathbf{p})$ and $\gamma_2(\mathbf{p})$, respectively. Thus the actual angle between them is calculated as:

$$\theta(\mathbf{p}) = \begin{cases} \gamma_2 - \gamma_1 & : \quad \gamma_2 - \gamma_1 > 0 \\ \gamma_2 - \gamma_1 + \pi & : \quad otherwise \end{cases}$$

   To improve the robustness of our algorithm, we also compare the nearby intensities of the feature points for best matching. To this end, an additional image region $L(\mathbf{p})$ centered at $\mathbf{p}$ with size of 30×30 is selected, then after cancellation of shearing and rotation effects, its normalization form $R(\mathbf{p})$ is written by $I_0(\mathbf{A}(L-\mathbf{p})+\mathbf{p})$, where

$$\mathbf{A} = \begin{bmatrix} \cos\gamma_1 & -\cos\gamma_1/\tan\theta - \sin\gamma_1 \\ \sin\gamma_1 & -\sin\gamma_1/\tan\theta + \cos\gamma_1 \end{bmatrix}$$

4. Let $\mathbf{p}' = T(\mathbf{p})$ be the estimated transformed point of $\mathbf{p}$ in I_1. We search in Q for candidate feature point $\mathbf{q}$, such that: 1) $|\mathbf{p}' - \mathbf{q}| < r_1$; 2) the smallest angle between $\gamma_1(\mathbf{p})$ and $\gamma_1(\mathbf{q})$ is less than $\pi/4$; 3) $|\theta(\mathbf{p}) - \theta(\mathbf{q})| < \pi/4$. r_1 is the maximum distance between each pair of feature point, which is set to 20 in our experiment. If we can't find any candidate in Q, then remove $\mathbf{p}$ from P

and jump to 2; otherwise, we select the best one, $\bar{\mathbf{q}}$, whose normalized local image $R(\bar{\mathbf{q}})$ shares the maximum mutual information with $R(\mathbf{p})$.

5. To avoid condensed feature points, which may introduce large distortion in the recovered transformation, those points in P with distances to $\mathbf{p}$ less than r_2 are removed; Likewise, the neighboring points to $\mathbf{q}$ are also dropped in Q. r_2 is the threshold that decides the minimum distance among features points.
6. Swap I_0 and I_1, P and Q, T and T^{-1}, then go to 2.

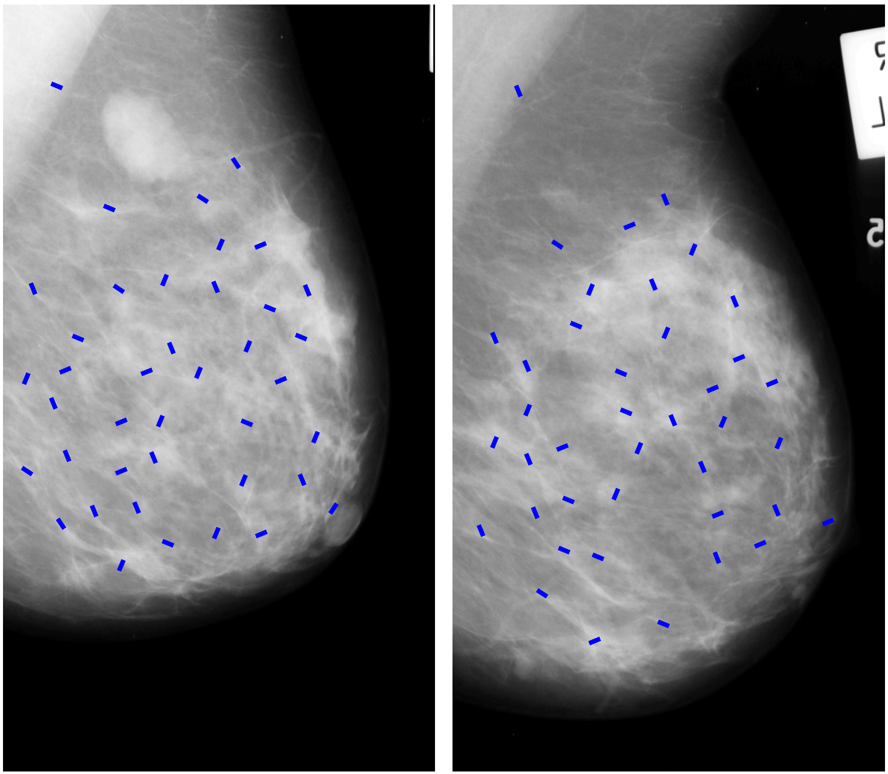

(a) (b)

Figure 5.3: (a)(b) show 40 pairs of anisotropic features extracted in the left and right mammograms, respectively.

Fig.5.3 shows a pair of bilateral mammograms, in which the anisotropic features are extracted using our approach.

5.5 Recovering Transformation with Anisotropic Features

Each of the feature point extracted above is associated with an orientation that is the principal direction of γ_1. Therefore, for better registration result, it's more appropriate to align the orientations of the landmarks in addition to the matching of positions [105].

We denote $\mathbf{p}_i$ and $\mathbf{q}_i$ the corresponding landmarks in I_0 and I_1. Their orientations are represented by two unit vector $\mathbf{d}_i$ and $\mathbf{e}_i$, which points to either γ_1 or $\gamma_1 \pm \pi$. Then the transformation $\mathbf{u}$ between I_0 and I_1 can be recovered by solving the following constrained optimization problem:

$$E(\mathbf{u}) = M(I_0, I_1, \mathbf{u}) + \lambda_1 \sum_{i=1}^{n} \det\left(\mathbf{e}_i, (\mathbf{u}\nabla - (\mathbf{u}\nabla)^T)|_{\mathbf{p}_i}\mathbf{d}_i\right) \tag{5.4}$$

Where $\mathbf{u}$ denotes the transformation to be recovered. The first term M is designed to match the intensity information between two images as much as possible. The criteria, *summed squared differences*(SSD), is currently incorporated in our current implementation. It is obvious that other metrics, for example, mutual information and correlation, are also possible here. The second term is the penalty for the misalignment between the orientations of corresponding landmarks. Note that $(\mathbf{u}\nabla - (\mathbf{u}\nabla)^T)|_{\mathbf{p}_i}\mathbf{d}_i$ is the rotated vector of $\mathbf{d_i}$ after the transformation, which is required to be collinear with $\mathbf{e}_i$ to register anisotropic information.

Note that there is no regularization term included in equation (5.4). This is because we discretize the transformation field by using cubic B-spline representation, which already has an inherent nature for regularization. To achieve better registration result, we introduce several pseudo landmarks at the image corners and boundaries to avoid unnecessary image floating. In the optimization of equation (5.4), all of the positional correspondences are treated as hard constraints, which further ensure the correct matching between the orientations associated with landmarks.

5.6 Results

Our registration framework is demonstrated by matching mammograms from the MIAS digital mammogram database. Three bilateral pairs of left and right images are selected and demonstrated in Fig.5.4, representing fatty-glandular tissues (MIAS 015/016), dense-glandular tissues (MIAS 35/36) and fatty tissues (MIAS 75/76) respectively. The right mammogram is registered to the left one in each case. The effectiveness of the registration process can be evaluated using comparative measures such as image-subtraction. By comparing pre- and post-registration errors, we found that much of the misregistration in the pre-registration difference image occurs along the periphery of the breast. After the breast boundary points aligned using technique proposed in this chapter, most of the peripheral differences can be removed from the subtraction image. In addition, the matching of texture-based anisotropic features selected from the interior of the breast region also helps to further improve the registration result.

5.7 Conclusion

In this chapter, we presented an automatic imaging framework to register corresponding mammograms with little human intervention. It combines a robust contour-matching algorithm for the matching of breast boundaries, and a novel feature-matching technique, which unwarps corresponding mammograms according to the texture-based anisotropic features automatically selected from the breast region. The experimental results also indicate that the proposed approach can provide useful information for better detection of breast abnormalities. In future, we will test our registration method on real clinical data for further evaluation of its robustness and efficacy.

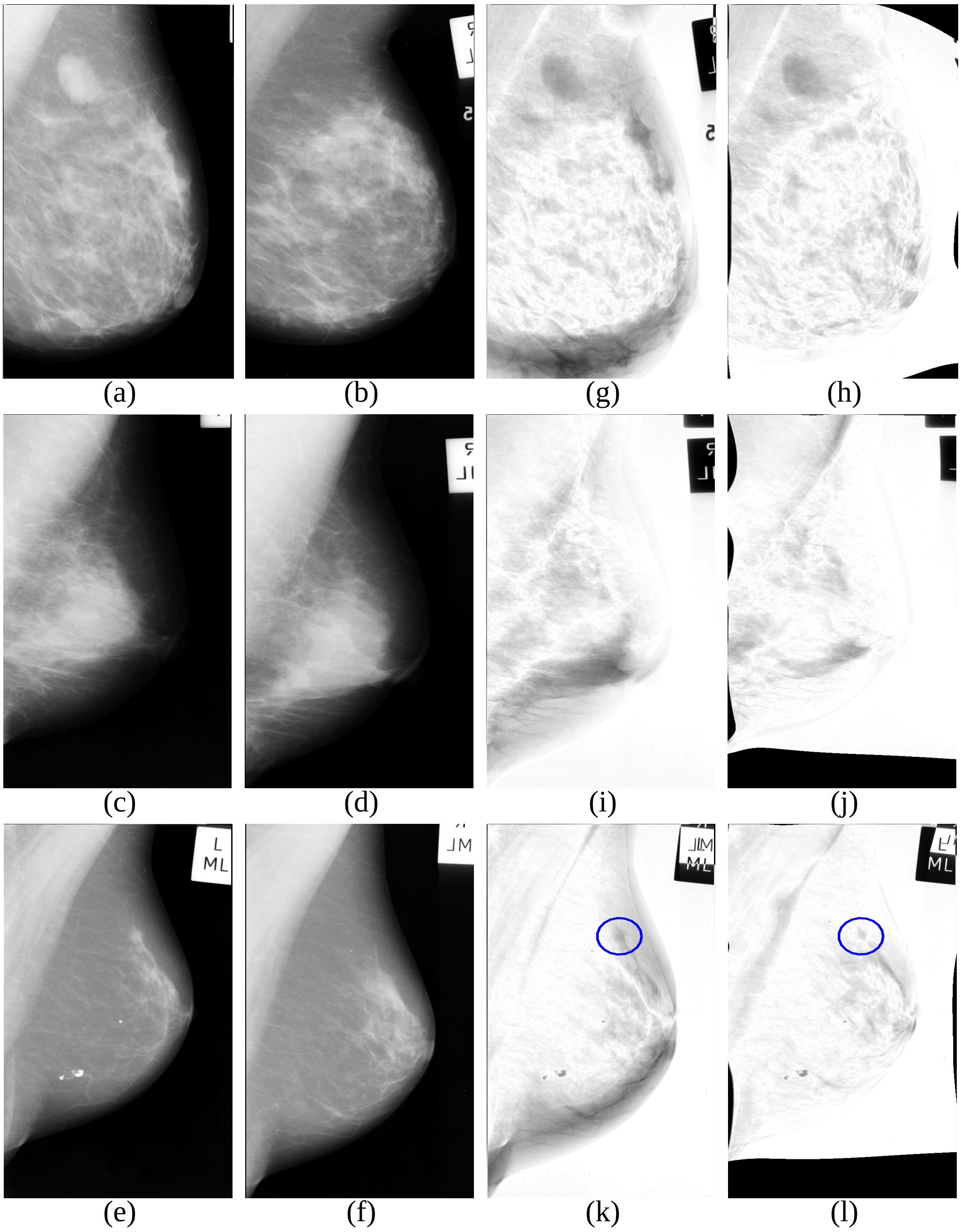


Figure 5.4: (a)-(f) represent MIAS15/16(fatty-glandular), MIAS35/36(dense-glandular) and MIAS75/76(fatty) pairs, respectively. (g)(i)(k) show the pre-registration error. (h)(j)(l) demonstrate the post-registration error. The asymmetry structure (highlighted with the blue circle) in (l) is more distinguishable from the surrounding pixels than that in pre-registration error map of (k).

Chapter 6

Spline Thin-Shell Simulation of Manifold Surfaces

6.1 Motivation

Flexible plates and shells are the fundamental geometric structures found in many fields of applied engineering nowadays. Since physics-based method is of great popularity for geometric modeling and simulation in CAD/CAM, the simulation of thin-shell objects is frequently required in modern engineering design practice. However, the modeling and simulation of thin-shells have traditionally been treated as two different stages due to the lack of a common representation scheme. An intermediate data conversion process is often employed to couple the modeling and simulation, but it may deteriorate both accuracy and robustness of the whole system. Therefore, an unified representation would be ideal to overcome such difficulties.

In theory, FEM can provide an approximate solution to the problem of thin-shell deformation, but it still remains as a challenging problem due to two obstacles:

Traditional finite-element is exclusively defined on planar domain, thus incapable of describe smooth surfaces and accompanying vector fields of complex manifolds and topologies without patching/trimming; Thin-shell finite-element must be at least C^1 continuous to ensure the convergence of the solution according to Kirchhoff-Love theory. However, traditional finite-elements, endowed with purely local polynomial shape functions, usually suffer from the difficulties in enforcing the desired C^1 continuity across the element boundaries.

A number of different approaches have been attempted to combat the aforementioned obstacles in thin-shell simulation. Due to the inherent difficulties in C^1 interpolation, alternative methods have been proposed to compromise the C^1 continuity requirement, such as degenerated solid elements, reduced-integration penalty methods, and many others [11,70]. Most recently, Cirak *et al.* [16] used the shape functions induced by subdivision rules for thin-shell finite-element simulation. Despite their modeling advantages, the subdivision surfaces do not allow close-form analytic for their basis functions, and have more unnecessary extraordinary points depending on the connectivity of the control mesh (instead of the intrinsic topology of the manifold). Another noteworthy FEM presented in [62] uses Element-Free Galerkin (EFG) method to simulate and analyze Kirchhoff shells and plates. However, it requires extra efforts to combine the model geometry with the simulation process via data conversion. In general, all these approaches fail to provide an effective way to handle thin-shell surfaces with sophisticated topology.

In this chapter we articulate a novel framework that naturally couples the modeling and simulation processes for arbitrary thin-shell surfaces. Spline surfaces are prevalent in commercial modeling systems because of their unique advantages in shape modeling, manufacturing and visualization. With the recent development of manifold spline theory [39], which enables the flexible construction of splines over any manifold of arbitrary topologies, we particularly introduce a novel thin-shell finite-element based on triangular B-spline [20] defined over manifold domain. The advantages of our method over the previous state-of-the-art thin-shell simulation include: First, the shell objects of arbitrary topology can be easily modeled by manifold triangular B-splines, with a minimum number of singular points intrinsic to the topological structures of the manifolds; Second, the C^1 continuity requirement can be easily achieved for triangular B-splines; Finally, our spline-based primitive naturally integrates geometric modeling with physical simulation by avoiding unnecessary data conversion and meshing procedure, which can lead to faster product design and development cycle.

6.2 Spline Representation of Manifold Surfaces

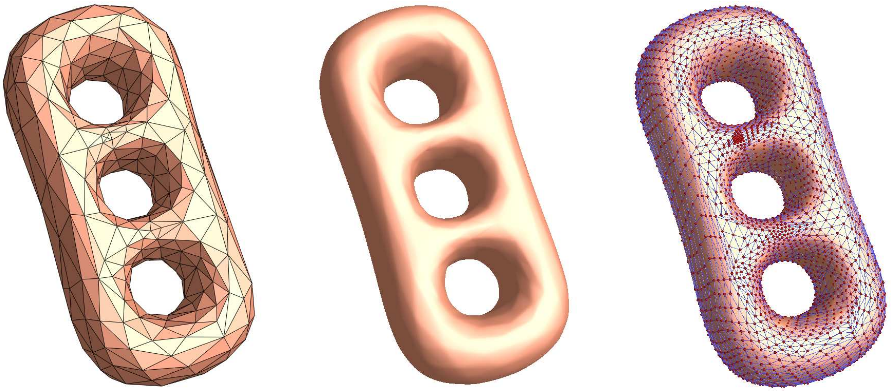

(a) domain (b) spline (c) control points

Figure 6.1: A genus-3 manifold triangular B-spline. (a) domain manifold with 742 triangles. (b) cubic manifold triangular B-spline surface. (c) spline overlaid with control points

In [39], Gu, He and Qin systematically build the theoretic framework of manifold spline, which locally is a traditional spline, but globally defined on the manifold. First, the manifold is covered by a special atlas, such that the transition functions are affine. Then, the knots are defined on the manifold and the evaluation of polar form is carried out on the charts. Although on different charts, the knots are different, the evaluation value is consistent and independent of the choice of charts. Furthermore, the existence of such atlas depends on the domain topology. This new paradigm unifies traditional subdivision surfaces and splines.

The geometric intuition of the definition of manifold spline is that first we replace a planar domain by the atlas of the domain manifold, and then all the constituent spline patches naturally span across each other without any gap. The central issue of constructing manifold splines is that the atlas must satisfy some special properties in order to meet all the requirements for the evaluation independence of chart selection.

In [39], Gu et al. show that for a local spline patch, the only admissible parameterizations differ by an affine transformation. This requires that all the chart transition functions are affine. Furthermore, they show that given a domain manifold M of genus g, a manifold triangular B-spline can be constructed with no more than $|2g-2|$ extraordinary points.

The manifold triangular B-spline can be written as follows:

$$\mathbf{F}(\mathbf{u}) = \sum_{I} \sum_{|\beta|=n} \mathbf{c}_{I,\beta} N(\phi(\mathbf{u})|V^{I}_{\beta}), \quad \mathbf{u} \in M \tag{6.1}$$

where $\mathbf{c}_{I,\beta} \in \mathbb{R}^3$ are the control points. Given a parameter $\mathbf{u} \in M$, the evaluation can be carried out on arbitrary charts covering $\mathbf{u}$.

Manifold triangular B-splines have many valuable properties which are critical for geometric and solid modeling. For examples, manifold triangular B-splines are piecewise polynomial defined on the manifold domain of arbitrary triangulation. Therefore, the computation of various differential properties, such as normals, curvatures, principal directions, are robust and efficient. The splines have local support, i.e., the movement of a single control point $\mathbf{c}_{I,\beta}$ only influences the surface on the triangle I and on the triangles directly surrounding I. The manifold triangular B-splines are completely inside the convex hull of the control points. The degree n manifold triangular B-splines are of C^{n-1}-continuous if there are no degenerate knots. Furthermore, by intentionally placing knots along the edges of the domain triangulation, we can model sharp features easily. The manifold spline of genus $g(\geq 1)$ has $2g-2$ singular points. See Figure.6.1 for an example of genus-3 manifold triangular B-spline.

6.3 Spline Thin-shell Simulation

6.3.1 Elastic Thin-shell Mechanics

The mechanical response of a spline surface with an attached thickness property can be computed with the classical Kirchhoff-Love shell theory. In the interest of smooth technical flow, let us briefly review the derivation of thin-shell equations. Detailed presentation of classical shell theories can be found elsewhere in mechanical engineering literatures.

Thin-shell is a particular form of three-dimensional solid whose thickness is significantly small as compared with the other two dimensions. Let $\mathbf{X}(\theta_1,\theta_2)$ denote the middle surface of the thin shell, where θ_1 and θ_2 are the parametric coordinates of the surface. The generic configuration of the shell can be described as

$$S=\{\mathbf{x}\in R^3|\mathbf{x}=\mathbf{X}(\theta_1,\theta_2)+\theta_3\mathbf{X}_{,3}(\theta_1,\theta_2),\quad -\frac{h}{2}\leq\theta_3\leq\frac{h}{2}\},$$

where $\mathbf{X}_{,3}$ is a unit director vector normal to the middle surface of the shell both in the reference and deformed configuration under the Kirchhoff-Love hypothesis. The internal energy of the shell depends on the differential quantities of the middle surface, such as the metric and curvature tensor. Assuming linearized kinematics, the displacement field of the middle surface is introduced as $\mathbf{u}(\theta_1,\theta_2)=\mathbf{X}(\theta_1,\theta_2)-\mathbf{X}^0(\theta_1,\theta_2)$, where the superscript "0" is used to denote the measurement in the reference configuration. Thus, the linearized membrane and bending strain tensor can be expressed as:

$$\varepsilon_{ij}=\frac{1}{2}(\mathbf{X}^0_{,i}\cdot\mathbf{u}_{,j}+\mathbf{X}^0_{,j}\cdot\mathbf{u}_{,i}), \tag{6.2}$$

$$\rho_{ij}=-\mathbf{u}_{,ij}\cdot\mathbf{X}^0_{,3}+(\mathbf{J}^0)^{-1}[\mathbf{u}_{,1}\cdot(\mathbf{X}^0_{,ij}\times\mathbf{X}^0_{,2})+\mathbf{u}_{,2}\cdot(\mathbf{X}^0_{,1}\times\mathbf{X}^0_{,ij})]. \tag{6.3}$$

where $\mathbf{J}=|\mathbf{X}_{,1}\times\mathbf{X}_{,2}|$, $\mathbf{X}_{,3}=\mathbf{J}^{-1}(\mathbf{X}_{,1}\times\mathbf{X}_{,2})$, and $|\mathbf{X}_{,3}|=1$. Here, the subscripts take the values of 1 and 2, and a comma denotes partial differentiation. Note that, the derivation of the membrane and strain is independent of the introduction of the Kirchhoff-Love hypothesis.

Under the assumption of linearity of elasticity, the strain energy density is defined as follows:

$$W(\mathbf{u})=\frac{1}{2}\frac{Eh}{1-\nu^2}H^{\alpha\beta\gamma\delta}\varepsilon_{\alpha\beta}\varepsilon_{\gamma\delta}+\frac{1}{2}\frac{Eh^3}{12(1-\nu^2)}H^{\alpha\beta\gamma\delta}\rho_{\alpha\beta}\rho_{\gamma\delta}, \tag{6.4}$$

in which, the first term is the membrane strain energy density and the second one is the bending strain energy density. Thus, the overall potential energy is as follow:

$$E(\mathbf{u})=\int_\Omega W(\mathbf{u})d\Omega+E_{ext}=E_{int}+E_{ext},$$

where E_{int} is the internal elastic energy and E_{ext} is the potential of the applied forces.

Following the principle of minimum potential energy, we can get the stable equilibrium configurations of the thin-shell. The Euler-Lagrange equations corresponding to the minimum principle may be expressed in the weak form as:

$$\langle DE_{int}(\mathbf{u}), \mathbf{v}\rangle + \langle DE_{ext}(\mathbf{u}), \mathbf{v}\rangle = 0 \tag{6.5}$$

where $\mathbf{v}$ is the trial displacement field.

6.3.2 Spline Element Discretization

Following the construction of manifold triangular B-splines given in (6.1), we can extract the basis functions and write them by:

$$\varphi^l(\phi(\mathbf{v})) = \sum_{\xi(I,\beta)=l} N(\phi(\mathbf{v})|V^I_\beta) \quad \mathbf{v} \in M \tag{6.6}$$

in which $\xi : \mathbb{N} \times \mathbb{N}^3 \to \mathbb{N}$ associates each local simplex-spline with an unique global shape functions it contributes to, ϕ is the conformal mapping, and $\phi(\mathbf{v})$ denotes the point in the planar domain, mapped from a manifold point $\mathbf{v}$. We will use these expression in the following discussion, and represent $\phi(\mathbf{v})$ by $\mathbf{x}$ if necessary.

Thus, we can easily extend the membrane and bending strain tensors from planar parametric domain to manifold domain and write them in the form:

$$\varepsilon(\phi(\mathbf{v})) = \sum_{l=1}^{L} \mathbf{M}^l(\phi(\mathbf{v}))\mathbf{u}_l, \tag{6.7}$$

$$\rho(\phi(\mathbf{v})) = \sum_{l=1}^{L} \mathbf{B}^l(\phi(\mathbf{v}))\mathbf{u}_l \tag{6.8}$$

where $\mathbf{B}^l$ are the membrane and bending strain matrices, and $\{\mathbf{u}_l, l = 1, \ldots, L\}$ are the nodal displacement vectors.

Substituting equations (6.7) and (6.8) into (6.5) yields the linear equations developed from the manifold domain as:

$$\mathbf{KU} = \mathbf{F} \tag{6.9}$$

where $\mathbf{K}$ is the stiffness matrix, $\mathbf{U}$ is the collection of nodal displacement $[\mathbf{u}_1^T \cdots \mathbf{u}_L^T]^T$, and $\mathbf{F}$ is the nodal force vector. $\mathbf{K}$ is a block matrix which can be conveniently assembled by filling in the following 3×3 matrices:

$$\mathbf{K}^{IJ} = \int_M \left[\frac{Eh}{1-\nu^2} (\mathbf{M}^I)^T \mathbf{H} \mathbf{M}^J + \frac{Eh^3}{12(1-\nu^2)} (\mathbf{M}^I)^T \mathbf{H} \mathbf{M}^J \right] dM$$

with the constitutive matrix $\mathbf{H}$ made of contravariant metric tensors, the definition of which is available in [16]. The construction of $\mathbf{F}$ will be discussed later.

6.3.3 Implementation Details

Numerical Integration: The thin-shell FEM simulation needs to compute the Kirchhoff energy of the deformed shell surfaces. However, the evaluation of the integrations over arbitrary manifold surfaces has been a challenging problem, which is usually awkwardly handled by piecewise parameterizations. With the global conformal mapping coupled with triangular B-splines theory, we can conduct the integration on an equivalent planar domain instead, and use any established numerical integration techniques. In our system, the shell elements are selected as the triangles of the tessellation, from which the triangular spline is constructed. Then we regularly subdivide each element into small congruent triangles, and compute the integration using triangle Gaussian quadratures.

Boundary Condition Handling: To facilitate the process of intuitive geometric design, we include point-based constraints as the input for our thin-shell simulation system. The users are allowed to pick up a group of points on the spline surfaces, i.e. $\mathbf{P}^0 = \{p_1^0, p_2^0, \ldots, p_n^0\}$, and assign them with desired positions after the deformation, i.e. $\mathbf{P} = \{p_1, p_2, \ldots, p_n\}$, where n denotes the total number of the point constraints. This linear constraints thus defined can be grouped in a matrix format as:

$$\mathbf{P}^0 + \mathbf{C}\mathbf{u} = \mathbf{P}$$

where $\mathbf{C}$ is an extremely sparse matrix that stores the basis function values at corresponding constraint points $\mathbf{P}^0$. To combine the constraints with the Equation (6.9), we solve for $\mathbf{u}$ in the Null-space of $\mathbf{C}$, such that:

$$\mathbf{u} = \mathbf{N}\mathbf{u}' + \mathbf{u}^0$$

where $\mathbf{C}\mathbf{N} = 0$ and $\mathbf{C}\mathbf{u}^0 = \mathbf{P} - \mathbf{P}^0$. We use Gaussian-Jordan-elimination-like approach [14] to construct $\mathbf{N}$, and solve for $\mathbf{u}^0$ by either singular value decomposition (SVD) or QR decomposition method. Due to the extreme sparsity and rank-deficiency of $\mathbf{C}$, such method is computationally viable to handle point-based geometric constraints.

Level-of-Detail (LOD) Simulation: The shell objects with affluent surface details requires massive number of degrees of freedom (DOF) for accurate geometric modeling. However, the triangular B-splines models having large number of control points are not suitable for interactive geometric design. Thus, we incorporate a level-of-detail (LOD) strategy to accommodate thin-shell deformation of sophisticated models. Any thin-shell surfaces $\mathcal{S}$ can be decomposed to a smooth spline-based surface $\mathbf{S}_0$ and a scalar function d describing the additional displacements, i.e.:

$$\mathbf{S}(\mathbf{x}) = \mathbf{S}_0(\mathbf{x}) + d(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})$$

where $\mathbf{n}$ is the normal vector of $\mathbf{S}_0$. Practically, $\mathbf{S}_0$ can be estimated by fitting the original surface using manifold triangular B-spline with relatively small number of control points [45]. Then the magnitudes of the fitting errors along the normal directions will be further modeled as a spline-based function d with more degree of freedoms. For the LOD simulation of a complicated thin-shell model, our system allows users to sculpt on the base surfaces $\mathbf{S}_0$, then the previously recorded details will be automatically applied to give the final design results. Figure.6.2 gives two examples of geometric design with LOD thin-shell simulation.

6.4 Results

Our system is implemented on a Microsoft Windows XP PC with Intel Pentium IV 3.0GHz CPU, 1.0GB RAM, and an nVidia GeForce Fx 5600 Ultra GPU. We have run a variety of examples to verify and test the efficacy and performance of our method. These examples includes a female face, the stanford bunny, a torus and a kitty. Both the face and the bunny are LOD-modeled. And both the torus and the kitty models have non-trivial genus. The statistics and performances of thin-shell simulation on these examples are documented in Table.6.1.

		Kitty	Torus	Bunny(LOD)	Face(LOD)
$\mathbf{S_0}(\mathbf{x})$	#DOF	990	324	662	256
	Order	3	3	3	5
$d(\mathbf{x})$	#DOF	N/A	N/A	3106	3181
	Order	N/A	N/A	4	5
#Constraints		8	6	30	9
Timing	Stage1	25.06s	3.25s	17.81s	7.34s
	Stage2	0.56s	0.15s	8.38s	0.53s
	Stage3	2.34s	1.19s	2.09s	1.53s

Table 6.1: Model statistics and performance data. Stage one is assembling **K**, stage two is to handle boundary constraints, and stage three is deformation.

6.5 Summary

In this chapter, we propose a novel paradigm that successfully simulates the elastic deformation of thin-shell objects. We also provide users with a LOD sculpting tool for esthetical geometric design. The experiment results show demonstrate that the proposed thin-shell FEM method has the following advantages over the traditional ones. It can easily achieve the C^1 continuity requirement, and represent arbitrary thin-shell surfaces using splines with minimum number of singular points. Our spline-based primitive naturally integrates geometric modeling with physical simulation in the entire CAD/CAM process, thus unnecessary data conversion and meshing procedure is total avoided. For future work, we will extend current framework to handling large thin-shell deformation by considering nonlinear elastic energy, and solve the simulation problem in temporal dimension for animation applications.

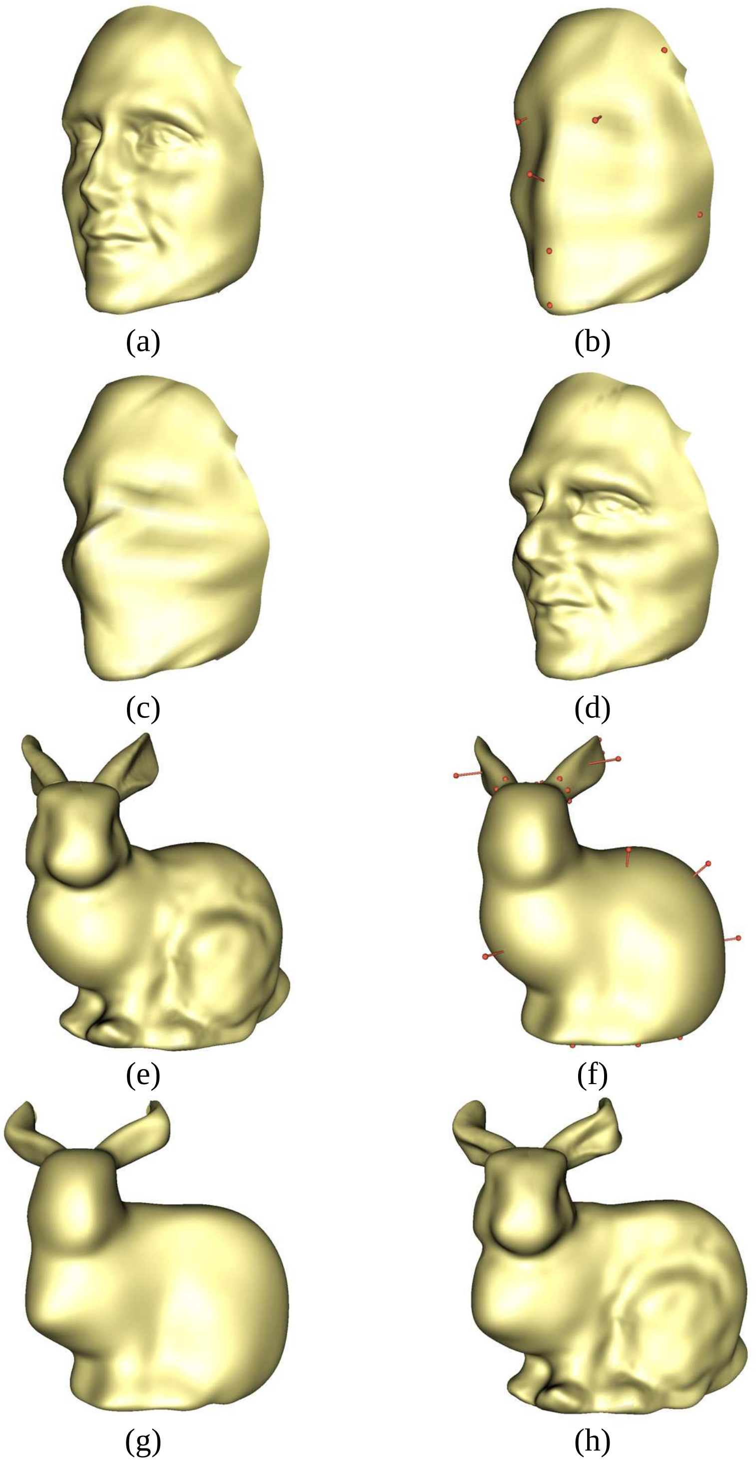
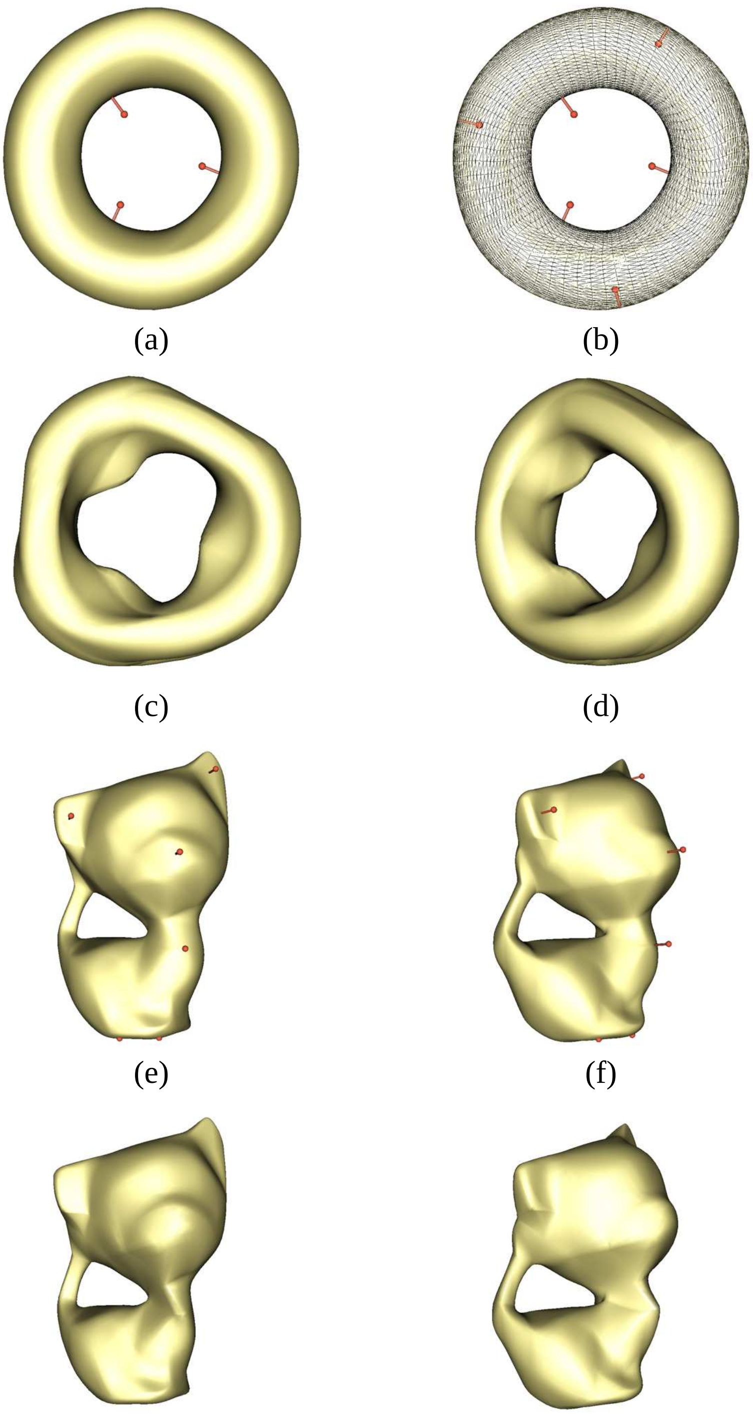


Figure 6.2: LOD thin-shell simulation (a)(e) the original surfaces with feature details. (b)(f) the base surfaces with geometric constraints. (c)(g) the base surfaces after thin-shell deformation. (d)(h) the original surface after LOD thin-shell deformation.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 6.3: (a)(b) 6 points constraints applied on the torus surface. (c)(d) torus shell after deformation. (e)(f) the front and side view of the kitty with points constraints. (g)(h) the front and side view of the deformed kitty shell.

Chapter 7

Restricted Trivariate Polycube Splines (RTP-splines)

7.1 Motivation

Volumetric data of massive size are now available in a wide variety of scientific and research fields, because of the rapid advancement of modern data acquisition technologies. A frequently occurring problem is how to convert acquired 3D raw data of discrete samples into a continuous representation upon which simulation and analysis processes can be efficiently developed and accurately computed. The majority of traditional solid modeling techniques during the past four decades have been established upon the following theoretic foundations: constructive solid geometry (CSG), boundary representation (B-reps), and cell/space decomposition. Most of these representations lack the ability of smoothly modeling solid geometry, which is required by modern engineering design in order to directly apply physical simulations on modeled solids, without the necessity of expensive remeshing of finite-element structure and shape data conversion between discrete and continuous representations and between linear finite elements and higher piecewise splines in 3D. In practice, real-world objects (directly acquired via the scanning process) have complex geometry and non-trivial topologies. Therefore, constructing efficient representations for general solid objects in favor of physical simulation and engineering design remains to be a very challenging task.

Trivariate simplex splines [50] have been developed to model multi-dimensional, material attributes of volumetric objects. However, computing blending functions and their derivatives on simplex splines is not straightforward and inefficient, compared with NURBS and tensor-product B-splines. Also, how to automatically place boundary sub-knots to avoid numerical degeneracies remains to be an open problem. Trivariate simplex splines are defined over an unstructured tetrahedral grid, which can be easily obtained from triangular meshes by certain mesh generation softwares such as *Tetgen* [122]. Although solid object of complex topologies and geometries can be modeled by trivariate simplex splines upon such unstructured grids, the majorities of simulation solvers have preferences on structured grid. This is because, low-quality tetrahedral meshes usually cause large simulation errors or numerical instability. Motivated by current industrial practice in various engineering design and analysis systems, we focus on designing a volumetric spline modeling framework based on structured grid domains.

In the framework of *isogeometric analysis* proposed by [53, 157], trivariate tensor-product B-splines/NURBS are directly used for modeling smooth geometry, material attributes, and physical simulation of solid objects simultaneously. Martin et al. [75] convert a solid femur mesh to a cylindrical trivariate B-spline by parameterizing the model into a solid cylinder. Due to the topological limitation of the cylinder domain, the constructed trivariate tensor-product splines can not model solid objects with bifurcations and arbitrary topologies, without enormous efforts in patch gluing/trimming, and imposing smoothness constraints along patch boundaries. Furthermore, local refinement required in level-of-detail modeling is not supported by tensor-product splines because basis function refinement will introduce many superfluous control points across the entire domain. As an extension to NURBS, T-splines [116, 117] solve this problem on semi-regular grid domains. To the best of our knowledge, no work has generalized T-splines for three dimensional, multi-attribute data and directly applied them to volumetric geometry and data modeling.

Directly generalizing T-spline surface to volumetric data is not straightforward and far from trivial. A general T-spline function defined over a bivariate domain can be formulated as

$$\mathbf{F}(u,v) = \frac{\sum_{i=1}^{n} w_i \mathbf{p}_i B_i(u,v)}{\sum_{i=1}^{n} w_i B_i(u,v)} \quad (u,v) \in \mathbb{R}^2, \tag{7.1}$$

where $\mathbf{p}_i$ are control points associated with weight w_i, and $B_i(u,v)$ denote basis functions. With this definition, two pieces of T-spline patches can be stitched together by blending boundary basis functions, and we form a new T-spline that can preserve smoothness across the boundary. Trivariate T-splines inherit such nice features, and T-splines defined on polycube volumetric domains can be similarly constructed by gluing a group of T-spline cubes. However, the calculation of this T-spline function and its derivatives requires to divide blending functions by the sum of all the contributed ones. This will make the evaluation computationally inefficient. Recently, *Semi-standard* T-splines introduced in [116] guarantee $\sum_{i=1}^{n} w_i B_i(u,v) \equiv 1$ in Equation (7.1) across the entire domain. In this setting, computing $\mathbf{F}(u,v)$ and its derivatives can be much more efficient.

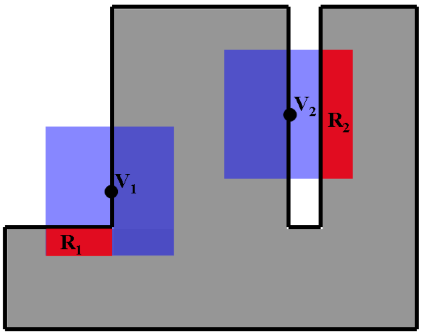


Figure 7.1: Extra support regions. On a concave domain, if the supporting box region of a blending function intersects with the domain boundary (e.g., boxes of v_1 and v_2), extra control points (e.g., in red regions) could contribute to the function blending unnecessarily.

However, how to construct a semi-standard T-spline, especially over non-trivial parametric domains, is a challenging problem. Another issue is that, conventional T-splines are defined with open boundaries, i.e., the support regions of blending functions may go across the domain boundaries. Such an open-boundary scheme upon polycube domain will cause control points to unnecessarily contribute to extra domain regions. Two examples are shown as red-regions in Figure 7.1. This might cause geometric inconsistency in modeling underlying solid objects, and in physical simulations. Therefore, it is ideal to have a trivariate spline inherit from T-splines, that (1) is defined within the largest visible region inside the domain, and (2) has the property of semi-standardness. Such novel splines will greatly facilitate direct modeling and physical simulations of arbitrary solid objects with complex geometries and sophisticated topologies. The spline constructed in this chapter has these properties, and we call it the Restricted Trivariate Polycube Spline (RTP-spline). We present a framework of RTP-splines construction and the data conversion of volumetric models to this spline representation.

The main contributions of this chapter include:

1. A new spline (RTP-spline) scheme is uniquely formulated over polycube domain, with blending functions restricted inside domain boundaries. The RTP-splines also have the following advantages:
    - It is capable of local refinement;
    - Computing RTP-spline functions and their derivatives is much more efficient than that on traditional T-spline surfaces;
    - The polycube domain enables natural modeling of arbitrary solid objects, since low distortions and few singularity points are introduced in volumetric parametrization when the domain mimics the geometries and topologies properly;
    - The restricted boundaries of RTP-spline effectively ensure the physical modeling and simulations adhere to the geometries of underlying objects.
2. We develop a novel framework to construct RTP-splines in an effective top-down fashion.
3. We construct RTP-splines on several volumetric models with both geometry and synthesized texture information (to mimic material properties), which demonstrates that our RTP-splines can model not only geometry but also multi-attribute fields within an unified paradigm.

7.2 Related Works

Research on spline-based volumetric modeling has gained much attention recently. 4D uniform rational cubic B-spline volume is used to constructively model FRep solids defined by real-valued functions [112]. The method presented in [76] represents and specifies physical attributes across a trivariate NURBS volume.

However, it is more desirable in engineering design to have an integrated modeling framework that represents geometry, material attributes, and conducts simulations simultaneously. Trivariate NURBS are used to model skeletal muscle with anisotropic attributes [157], on which NURBS-FEM analysis is directly conducted. Martin et al. [75] present a method based on volumetric harmonic functions to parameterize a volumetric solid to a solid cylinder in order to fit a single trivariate B-spline to geometric data and model simulation attributes. A modeling technique based on triangular simplex spline [50] is developed to model and render multi-dimensional, material attributes for solid objects with complicated geometries and topologies.

The splines proposed in this chapter are founded upon the T-spline technique, which is invented in [117]. T-spline is a generalization of NURBS, but permits T-junctions on its control mesh and enables local insertion of additional knots without introducing superfluous control points. A local refinement method is proposed in [13, 116] to simplify NURBS surfaces to T-spline representations by removing superfluous control points. The merge of B-spline patches defined over different local domains for getting a single T-spline representation on the manifold domain is thoroughly discussed in [54].

Bazilevs et al. [4] propose an isogeometric analysis framework based on T-splines. Its main focus is on planar T-splines for surfaces, and volumetric T-splines is only briefly mentioned without offering any technical details. Generalized trivariate T-splines (whose control points are associated with weights) are employed by [123] to model free-form deformation fields. For the purpose of shape metamorphosis, 3D level sets represented by T-splines are adopted in [32, 151–153] for its efficiency. This is because, the distribution of T-spline control points can be made adaptive to the geometry of the morphing objects.

Our work relies on the generation of a polycube domain and parametrization upon it. The concept of polycube domain is originally proposed for seamless texture mapping with low distortion [127]. Polycube volume and its surface parameterization can be constructed either manually [127, 135, 136] or automatically [46, 67]. Based upon specially-designed surface parametrization, [135] builds manifold bivariate T-spline over a polycube that can handle models with arbitrary topology. A few recent work [65,74,143] studies the parameterization of a solid object to canonical domains such as spheres, polycubes, etc. Volumetric parameterization typically starts from any given surface mapping, and parameterizing volumetric data onto a solid polycube domain serves as an important pre-processing step for the conversion of any solid model to RTP-splines.

7.3 Preliminaries and Notations

In this section, we introduce the general algorithm to construct trivariate T-spline with duplicate knots on regular box domain, review the theory of basis function refinement, and define necessary notations for the rest of this chapter.

7.3.1 Trivariate T-spline with Duplicate Knots

Defined on a grid structure that allows T-junctions (or T-mesh), the T-spline proposed in [117] is a generalization of non-uniform B-splines (or NURBS). When considering a simple cube domain, the definition of T-spline surfaces can be straightforwardly extended to three dimensions and generate trivariate T-splines on T-lattice grids. We can use “T-junctions” to refer to the intersections between faces and/or lines. Let $T(\mathcal{V},\mathcal{C},\mathcal{F})$ denote a rectilinear grid structure that permits T-junctions, where $\mathcal{V}$, $\mathcal{C}$, and $\mathcal{F}$ are sets of vertices, cells, and faces, respectively.

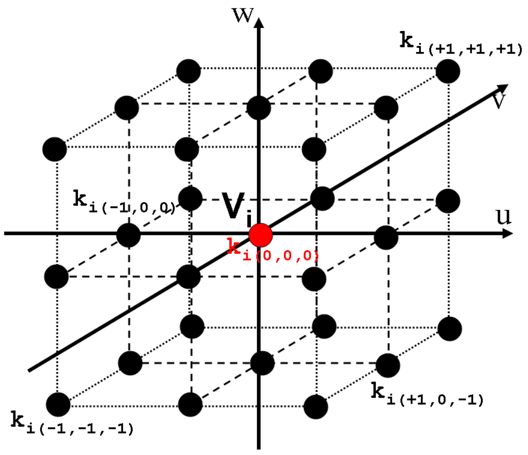


Figure 7.2: At vertex $\mathbf{v}_i$, there are at most 27 duplicate knots that can be placed on an imaginary $3 \times 3 \times 3$ grid. The center one is master knot (red), while the rest are sub-knots (black). Examples of indexing duplicate knots are also shown.

In order to define a T-spline upon T, we specify a set of knots on T, which are denoted by $\mathcal{K}$. In this chapter, each knot must reside at a certain vertex. At most 27 duplicate knots are allowed at every vertex, and they are organized on a $3 \times 3 \times 3$ grid of infinitesimal size, as shown in Figure 7.2. Every vertex $\mathbf{v}$ has a *master knot*, and some other optional duplicate knots, called *sub knots*. We refer a knot at $\mathbf{v}_i$ by $\alpha, \beta, \gamma \in \{-1, 0, +1\}$, in which the triplet (α, β, γ) indicates an unique nodal position on the local grid. The coordinate of $\alpha, \beta, \gamma \in \{-1, 0, +1\}$ can be written in the form of

$$\mathbf{k}_{i(\alpha,\beta,\gamma)} = \mathbf{v}_i + \frac{\varepsilon}{2}[\alpha, \beta, \gamma]^T \tag{7.2}$$

where ε denotes an infinitesimal size. If we assume $\varepsilon \neq 0$, every knot is considered differently, and its coordinate is called the *topological coordinates*. Two knots are *topologically equivalent* if their topological coordinates are the same. If $\varepsilon = 0$, then the coordinates of all duplicate knots at $\mathbf{v}_i$ degenerate into an identical *numerical* *coordinate*, and these knots are *numerically equivalent*. In the rest of this chapter, when there is no ambiguity, we represent a knot by a simpler notation $\mathbf{k}_j$, where j indicates the index of $\mathbf{k}$ in $\mathcal{K}$.

Given T and $\mathcal{K}$, a trivariate T-spline can be defined as

$$\mathbf{F}(u,v,w) = \frac{\sum_{i=1}^{|\mathcal{B}|} \mathbf{p}_i B_i(u,v,w)}{\sum_{i=1}^{|\mathcal{B}|} B_i(u,v,w)} \quad (u,v,w) \in \mathbb{R}^3 \tag{7.3}$$

where (u,v,w) denotes 3D parametric coordinates, $\mathbf{p}_i$ are control points, and $\mathcal{B} = \{B_i(u,v,w)\}$ is the collection of blending functions. Each $B_i(u,v,w)$ is a tensor-product of three B-spline basis functions, written as

$$B_i(u,v,w) = N_{i0}^3(u) N_{i1}^3(v) N_{i2}^3(w) \tag{7.4}$$

where $N_{i0}^3(u)$, $N_{i1}^3(v)$ and $N_{i2}^3(w)$ are defined along u, v, and w directions, respectively. In the case of cubic T-spline, the univariate function N_{ij}^3 is constructed upon knot vector $\Xi_i^j = [\xi_{i0}^j, \xi_{i1}^j, \xi_{i2}^j, \xi_{i3}^j, \xi_{i4}^j]$, which is deduced from T and a collection of knots $\mathcal{K}$.

The way to infer Ξ_i^j is parallel to that for T-mesh. Starting from a knot $\mathbf{k} = (\xi_{i2}^0, \xi_{i2}^1, \xi_{i2}^2)$, ξ_{i3}^0 and ξ_{i4}^0 are found by shooting a ray $L(t) = (\xi_{i2}^0 + t, \xi_{i2}^1, \xi_{i2}^2)$ into parametric domain. ξ_{i3}^0 and ξ_{i4}^0 are the coordinate values at the first two intersections where $L(t)$ comes across either a knot or a face. If $L(t)$ goes out of the domain without two intersections, the last one is replicated such that $\xi_{i3}^0 = \xi_{i4}^0$ (see Figure 7.3), or $\xi_{i2}^0 = \xi_{i3}^0 = \xi_{i4}^0$. The other knots are determined in a similar fashion.

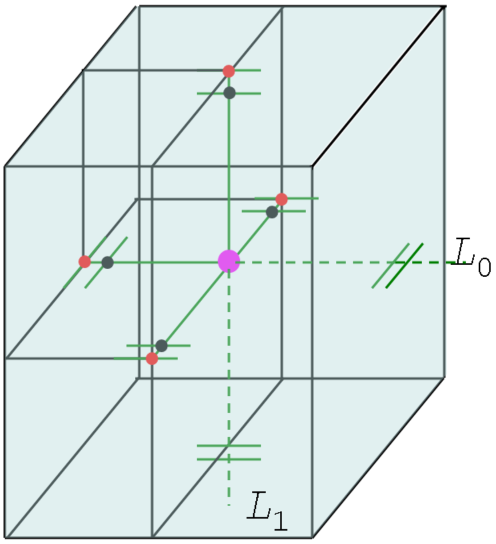


Figure 7.3: Knot vector extraction on T-lattice. The intersection between L_0 and the boundary appears twice in the resulting knot vector, and the same is true for L_1.

k	c_1	c_2	knot vector of $N_1(\xi)$	knot vector of $N_2(\xi)$
$\xi_0 \leq k < \xi_1$	$\frac{k-\xi_0}{\xi_3-\xi_0}$	1	$[\xi_0, k, \xi_1, \xi_2, \xi_3]$	$[k, \xi_1, \xi_2, \xi_3, \xi_4]$
$\xi_1 \leq k < \xi_2$	$\frac{k-\xi_0}{\xi_3-\xi_0}$	$\frac{\xi_4-k}{\xi_4-\xi_0}$	$[\xi_0, \xi_1, k, \xi_2, \xi_3]$	$[\xi_1, k, \xi_2, \xi_3, \xi_4]$
$\xi_2 \leq k < \xi_3$	$\frac{\xi_4-k}{\xi_4-\xi_0}$	$\frac{k-\xi_0}{\xi_3-\xi_0}$	$[\xi_1, \xi_2, k, \xi_3, \xi_4]$	$[\xi_0, \xi_1, \xi_2, k, \xi_3]$
$\xi_3 \leq k \leq \xi_4$	$\frac{\xi_4-k}{\xi_4-\xi_0}$	1	$[\xi_1, \xi_2, \xi_3, k, \xi_4]$	$[\xi_0, \xi_1, \xi_2, \xi_3, k]$

Table 7.1: Refining $N(\xi)$ by inserting k into knot vector $[\xi_0, \xi_1, \xi_2, \xi_3, \xi_4]$ generates two basis functions $N_1(\xi)$ and $N_2(\xi)$, which are scaled by c_1 and c_2, respectively.

7.3.2 Refinement of B-spline functions

To refine blending functions on trivariate T-splines, we need to review knot insertion algorithm for univariate B-spline functions. Let $\Xi = [\xi_0, \xi_1, \xi_2, \xi_3, \xi_4]$ be a knot vector and $N(\xi)$ denote the cubic B-spline basis function defined on it. If there is an additional knot $k \in [\xi_0, \xi_4]$ inserted into Ξ, $N(\xi)$ can be written as a linear combination of two scaled B-spline functions as

$$N(\xi) = c_1 N_1(\xi) + c_2 N_2(\xi) \tag{7.5}$$

where c_1, c_2 and knot vectors for $N_1(\xi)$ and $N_2(\xi)$ are determined by Table 7.1. Note that $N_2(\xi)$ and $N(\xi)$ always share the center knot ξ_2.

7.4 Constructing RTP-splines

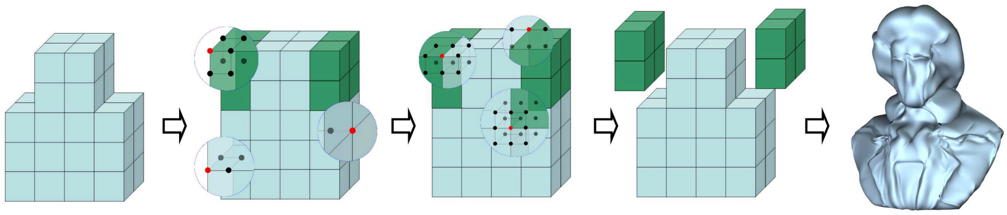


Figure 7.4: Overview of RTP-spline construction. The construction consists of four major steps, extending polycube domain to its bounding-box, building B-spline volume with bounded boundaries, inserting knots and conducting local refinement, and then removing exterior regions.

The construction of RTP-splines includes four major steps(see Figure 7.4): (1) extending given polycube P domain to a box domain, (2) building trivariate B-splines with restricted boundaries, (3) inserting duplicate knots and performing local refinement to separate interior and exterior blending functions, and (4) producing RTP-spline by removing structures/knots outside P. These steps are discussed in the following four subsections respectively.

7.4.1 Extension of Polycubes to Bounding-Boxes

Following notations introduced in Section 7.3.1, on the trivariate T-spline domain, let $P = (\mathcal{V}^P, \mathcal{C}^P, \mathcal{F}^P)$ be a given polycube structure, where $\mathcal{V}^P$, $\mathcal{C}^P$ and $\mathcal{F}^P$ denote vertices, cuboids and cell faces respectively. In order to extend P to a box volume with rectilinear grid, there must be no T-junctions nor intersections between cell faces on P. Our parameterization polycube domains (see Section 7.5.1) do not contain T-junctions. If other polycube mapping methods are used to construct the parametric domain and the generated domain has T-junctions, then we can always eliminate them simply by splitting the cells across the domain, through the extended planes of these intersecting cell faces. Now P can be extended to its bounding-box domain $T(\mathcal{V}, \mathcal{C}, \mathcal{F})$ by filling in some solid cuboids $G = (\mathcal{V}^G, \mathcal{C}^G, \mathcal{F}^G)$, where $\mathcal{V}^G = \mathcal{V} - \mathcal{V}^P$, $\mathcal{C}^G = \mathcal{C} - \mathcal{C}^P$, $\mathcal{F}^G = \mathcal{F} - \mathcal{F}^P$. G represents the exterior of P and we call it the *ghost region*.

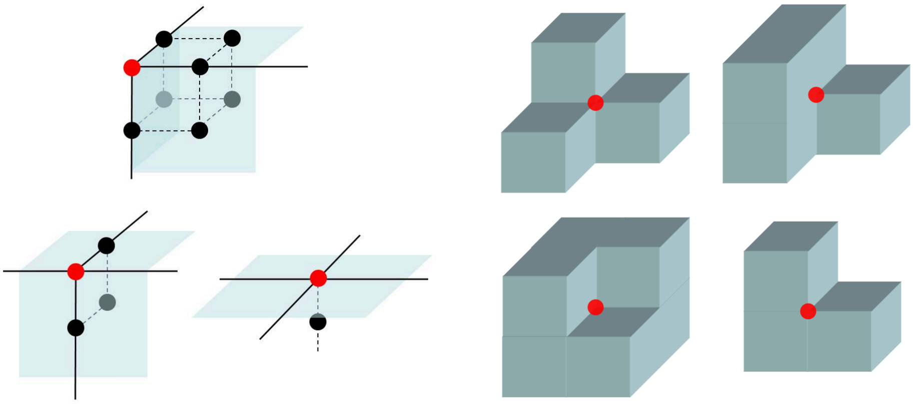
Note that there is a rectilinear grid embedded in the space of T, and the grids coordinates in k-axis direction are represented by

$$\mathbf{S}_k = [s^k_1, s^k_2, \ldots, s^k_{n_k}] \quad k = 1, 2, 3$$

where n_k is the resolution of rectilinear grid along k-axis.

7.4.2 Building B-spline Volume with Restricted Boundary

With the bounding box domain T constructed, it is not difficult to construct a trivariate tensor-product B-spline from the rectilinear grid structure on T by using $\mathbf{S}_1$, $\mathbf{S}_2$ and $\mathbf{S}_2$ as knot vectors. We must augment $\mathbf{S}_k$ to have a valid B-spline definition on the $n_1 \times n_2 \times n_3$ control grid. One approach is to add extra knots with



(a) (b)

Figure 7.5: (a) Knot configuration at corner, edge and face vertices for restricted boundaries. (b) Examples of extraordinary corners on a polycube.

coordinates outside domain region, generating an open-boundary scheme. In this chapter, we replicate the knots at both ends of $\mathbf{S}_k$ in order to restrict the B-spline blending function within domain T, i.e., $\mathbf{S}_k$ turns into

$$\mathbf{S}_k = [s_1^k, s_1^k, s_1^k, s_1^k + \varepsilon, s_2^k, \ldots, s_{n_k}^k - \varepsilon, s_{n_k}^k, s_{n_k}^k, s_{n_k}^k]$$

in which 6 extra knots are added to both ends. Note that the use of $s_1^k + \varepsilon$ here is to emphasize its topological difference from s_1^k, while ε is actually treated as 0 for B-spline evaluation. This also applies to $s_{n_k}^k - \varepsilon$ here.

Therefore, the trivariate tensor-product B-spline defined on T is formulated as

$$\mathbf{F}(u,v,w) = \sum_{i=1}^{n} \mathbf{p}_i B_i(u,v,w) \quad (u,v,w) \in \mathbb{R}^3 \tag{7.6}$$

where $n = n_1 \times n_2 \times n_3$ is the number of control points, and $B_i(u,v,w)$ are blending functions defined in Equation (7.4).

As an extension of tensor-product B-spline, we can locally construct blending functions using the method for T-spline (Section 7.3.1), instead of computing them directly from 3 global knot vectors. The knot set $\mathcal{K}$ to be used for local knot extraction is a tensor-product $[s_1^0, s_1^0+\varepsilon, s_2^0, \ldots, s_{n_0}^0-\varepsilon, s_{n_0}^0] \times [s_1^1, s_1^1+\varepsilon, s_2^1, \ldots, s_{n_1}^1-\varepsilon, s_{n_1}^1] \times [s_1^2, s_1^2+\varepsilon, s_2^2, \ldots, s_{n_2}^2-\varepsilon, s_{n_2}^2]$. Superimposing $\mathcal{K}$ with T reveals that duplicated knots only exist at corner, edge and face vertices, and their configurations are depicted in Fig 7.5(a). These sub-knots serve for a common purpose: to guarantee the partition-of-unity of blending functions inside boundary cells.

7.4.3 Local Refinement and Knot Insertion

We need to disjoin the blending functions defined on P from those defined on G, i.e. the support regions of these two group of blending functions will not overlap at all. This task is fulfilled via a two-step knot insertions.

- First, we insert duplicated knots on the boundary of P, to separate interior and exterior blending functions centered at master knots. (Section 7.4.3.2)
- Second, we iteratively eliminate violation cases in $\mathcal{B}$ by knot insertion. (Section 7.4.3.3)

Introducing new knots may change the underlying local knot vectors of existing blending functions. Therefore, we need to design an algorithm (Section 7.4.3.1) to resolve disagreements of blending functions $\mathcal{B}$ with new knot configuration, in order to preserve the partition-of-unity during the knot insertion.

7.4.3.1 Local Refinement of Blending Functions

We need to introduce an algorithm to refine blending functions $\mathcal{B}$ accordingly, whenever there are new knots added or any changes made to domain structure T.

The refinement algorithm proposed in [13, 116] works on a T-mesh, but its primary goal is to keep the shape of T-spline surfaces when new control points are inserted. In this chapter, we enhance this algorithm by extending it to rectilinear grid and supporting duplicated knots explicitly. We convert the previously obtained trivariate B-spline to a generalized T-spline volume and rewrite Equation 7.6 as

$$\mathbf{F}(u,v,w) = \frac{\sum_{i=1}^{|\mathcal{B}|} w_i \mathbf{p}_i B_i(u,v,w)}{\sum_{i=1}^{|\mathcal{B}|} w_i B_i(u,v,w)} \quad (u,v,w) \in \mathbb{R}^3 \tag{7.7}$$

where each control point $\mathbf{p}_i$ is associated with a weight w_i, which is collected in $\mathcal{W}$. At the beginning, all w_i are equal to 1, and $\sum_{i=1}^{|\mathcal{B}|} w_i B_i(u,v,w) \equiv 1$ holds for every (u,v,w).

If there are new knots inserted, or T undergoes any changes such as vertex insertion or cell splitting, we denote the new knot set by $\mathcal{K}^*$ and the new domain structure by $T^*(\mathcal{V}^*, \mathcal{C}^*, \mathcal{F}^*)$. And the new blending function set $\mathcal{B}^*$ and weights $\mathcal{W}^*$ can be generated by Algorithm 26 from inputs $\mathcal{K}^*$, T^*, $\mathcal{W}$ and $\mathcal{B}$.

In Algorithm 26, the superscript indicates the index of the blending function with which a variable is associated and subscript references the central knots of a blending function. For example, B_i^t is a blending function centered at knot $\mathbf{k}_i$ that originates from the t-th blending function in $\mathcal{B}$. The star superscript indicates that the variables are obtained from updated domain T^*, *e.g.*, Ξ_i^* denotes a knot vector implied by current T^* and centered at $\mathbf{k}_i^*$. The basic idea of Algorithm 26 is as follows. First, we decouple blending functions from their knots. Then, by either inserting new knots or refining basis functions (Section 7.3.2), we keep resolving the discrepancies between $\mathcal{B}$ and the local knot vectors implied by $\mathcal{K}^*$ and $\mathcal{T}^*$. A cell splits into two halves if the vertices on its edges can form an axis-aligned cutting plane. Finally, we merge same blending functions, with their weights being

Algorithm 1: Blending functions refinement for trivariate T-spline with duplicate knots

```
Input: $T^*(\mathcal{V}^*, \mathcal{C}^*, \mathcal{F}^*)$, $\mathcal{K}^*$, $\mathcal{B}$ and $\mathcal{W}$.
Output: updated $T^*$, updated $\mathcal{K}^*$, $\mathcal{B}^*$ and $\mathcal{W}^*$ while
        $\sum_{i=1}^{|\mathcal{B}^*|} w_i^* B_i^* \equiv \sum_{i=1}^{|\mathcal{B}|} w_i B_i$
$Q \Leftarrow \{(w_i^i, B_i^i) : w_i \in \mathcal{W}, B_i \in \mathcal{B}\}$
while $\exists (w_i^t, B_i^t) \in Q : \Xi_i^t \neq \Xi_i^*$ do
    forall the $(w_i^t, B_i^t) \in Q$ do
        obtain knot vectors $\Xi_i^*$ from $T^*$
        if $\Xi_i^t$ is numerically equivalent to $\Xi_i^*$ then
            $\Xi_i^t \Leftarrow \Xi_i^*$
        else if $\Xi_i^*$ is more refined than $\Xi_i^t$ then
            add a knot from $\Xi_i^*$ to $\Xi_i^t$ and do the refinement: $B_i^t = c_1 \tilde{B}_j^t + c_2 \tilde{B}_i^t$
            (Section 7.3.2)
            $\tilde{w}_j^t \Leftarrow w_i^t \cdot c_1$;    $\tilde{w}_i^t \Leftarrow w_i^t \cdot c_2$
            $Q \Leftarrow Q - \{(w_i^t, B_i^t)\} \cup \{(\tilde{w}_j^t, \tilde{B}_j^t), (\tilde{w}_i^t, \tilde{B}_i^t)\}$
        else if $\Xi_i^t$ has a knot $\mathbf{k}_{j(\alpha,\beta,\gamma)} \notin \mathcal{K}^*$ then
            $\mathcal{K}^* \Leftarrow \mathcal{K}^* \cup \mathbf{k}_{j(\alpha,\beta,\gamma)}$
            if $\mathbf{k}_{j(0,0,0)} \notin \mathcal{K}^*$ then
                $\mathcal{K}^* \Leftarrow \mathcal{K}^* \cup \{\mathbf{k}_{j(0,0,0)}\}$
                $\mathcal{V}^* \Leftarrow \mathcal{V}^* \cup \{\mathbf{v}_j\}$ ; // Insert a new vertex
            endif
        endif
    endfall
    forall the $c \in \mathcal{C}^*$ do
        if any new vertices on $c$ forming an axis-aligned plane cuts $c$ into $c_1$
        and $c_2$ then
            $\mathcal{C}^* \Leftarrow \mathcal{C}^* - \{c\} \cup \{c_1, c_2\}$ ; // divide $c$ into $c_1$ and $c_2$
        endif
    endfall
endw
$\mathcal{B}^* \Leftarrow \{B_i : (w_i^t, B_i^t) \in Q\}$
$\mathcal{W}^* \Leftarrow \{w_j = \sum_{\forall (w_j^t, B_j^t) \in Q} w_j^t\}$
```

summed up.

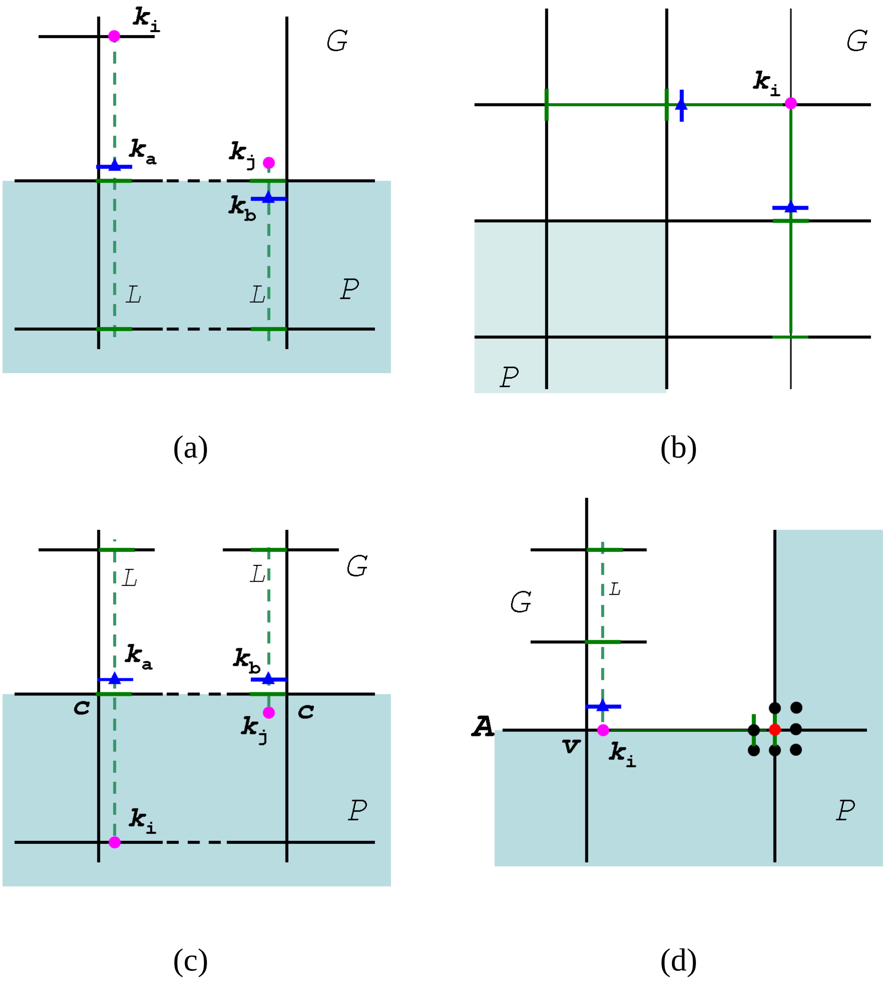
In comparison with the refinement algorithm proposed in [13, 116], Algorithm 26 disregards control points, conducts refinement and knot extraction on a much more complicated 3D grid, and explicitly support duplicated knots (i.e. combing numerically equivalent but topologically inequivalent knot vectors). Note that in the refinement result, the center of each blending function must lie at a knot, but it's not necessary for every knot to have an associated blending function. Moreover, the algorithm does not change the value of denominator in Equation 7.7 for every parametric point after the refinement. This keeps the **semi-standardness** of the original splines, and can be explained by

$$w_i^t B_i^t \equiv w_i^t \cdot c_1 B_j^t + w_i^t \cdot c_2 B_i^t \equiv \tilde{w}_j^t \tilde{B}_j^t + \tilde{w}_i^t \tilde{B}_i^t$$

7.4.3.2 Knot Insertion on Polycube Boundary

The purpose of inserting duplicated knots to the boundary of P is to cut off the connection between the polycube domain and the rest. In addition to the three kinds of boundary vertices shown in Figure 7.5(a), polycube structures have many other types of boundary vertices, especially at corners. Some examples of corner vertices on polycube are given in Figure 7.5(b). To create restricted boundary for polycube domain, we develop a general approach to configure duplicate knots at arbitrary boundary vertices, the description of which is as the follows. For a boundary vertex $\mathbf{v}_i$, we insert the master knot and all the sub-knots that topologically lies inside T (due to the definition of T, the boundary is considered as the interior of T, and the same is for P). We color the sub-knots topologically inside P by red, and those inside G by blue. We compare the knot configuration and colors on opposite faces of local $3\times3\times3$ grid for each axis direction. If they are identical, all the knots on both faces are removed. For instance, if there are $\mathbf{k}_{i(\mp1,\beta,\gamma)}$ for every existing $\mathbf{k}_{i(\pm1,\beta,\gamma)}$, and they share the same color, then all the knots $\{\mathbf{k}_{i(\pm1,\beta,\gamma)} \quad \beta,\gamma = -1,0,+1\}$ are removed. Then, Algorithm 26 is applied to generate a new $\mathbf{F}(u,v,w)$ to accommodate newly inserted knots. The motivation of this method is to minimize the sub-knots inserted so that the spline function is as smooth as possible.

7.4.3.3 Disjointing Blending Functions



(a) (b)

(c) (d)

Figure 7.6: Violation cases to be eliminated in RTP-spline construction. The pink dot denotes the center knot of a violating blending function. The blue triangles are the duplicate knots inserted in order to resolve the violations. The shaded region stand for the interior polycube domain.

In order to disconnect P from G by disjointing their blending functions, there are two conditions must be satisfied:

(a). there are no exterior blending functions influencing domain P;

(b). removing G does not affect the shape of blending functions inside P.

The knot insertion introduced in Section 7.4.3.2 guarantees that the blending functions lie at the interior and exterior master knots are separated by the polycube boundary, except for the following four violation cases, which need to be resolved for conditions (a) and (b).

1. A ray L from ghost knot $\mathbf{k}_i$ ($\mathbf{k}_j$) crosses the boundary and gets into P. Knot $\mathbf{k}_a$ ($\mathbf{k}_b$) is added to resolve the violation. (Figure 7.6(a))
2. In the vicinity of a convex corner, although L does not go inside P, the support region of blending function at $\mathbf{k}_i$ still overlaps P. For this case, extra knots are inserted at where L intersects the extended faces from the corner cell. (Figure 7.6(b))
3. From a sub knot $\mathbf{k}_i$ ($\mathbf{k}_j$) that lies topologically inside P, L intersects boundary at $\mathbf{c}$ and ends at either a ghost knot or a ghost face. Sub knot $\mathbf{k}_a$ ($\mathbf{k}_b$) is added to $\mathbf{c}$ to make the blending function centered at $\mathbf{k}$ vanish on the boundary. (Figure 7.6(c))
4. From a knot $\mathbf{k}_i$ that lies exactly on the boundary, L goes directly into G without meeting any other knots at vertex $\mathbf{v}_i$. In this cases, an exterior sub knot is added at $\mathbf{v}_i$ so that the continuity of blending function on the boundary is reduced to C^0. (Figure 7.6(d))

Case 1 and 2 deal with the violations for condition (a), where newly inserted knots break down the blending functions centered at ghost sub-knots, and restrict them inside G. Violations of condition (b) are fixed in case 3 and 4. During removal of ghost regions, if whose knots vectors include interior ghost knots of G, these blending function will be affected. To resolve such violations, we can either eliminate interior ghost knots from the knot vectors (case 3), or introduce C^0 continuity to the blending function on polycube boundary (case 4). To understand the later one, let's examine a cubic B-spline function $N(u)$ defined on knot vector $[k_0, k_1, k_1, k_1, k_2]$ where $k_0 < k_1 < k_2$. $N(u)$ has C^0 continuity at k_1, so that its shape between k_0 and k_1 is independent of k_2 and that between k_1 and k_2 is independent of k_0. It's worth to mention that in case 4, it's impossible for the blending function at $\mathbf{k}_i$ to influence the domain region beyond the plane A, due to the boundary knot insertion in Section 7.4.3.2.

Once all the violations are resolved, the refinement algorithm given in Section 7.4.3.1 is applied again to update $\mathcal{B}$ and $\mathcal{W}$. Since new knots may be introduced, the above steps are repeated until no violations is found and no further refinement is necessary on T. In our experiment, constructing RTP-splines for most models requires only one or two iterations. And our proposed algorithm is guaranteed to terminate because no vertices is added, and there are limited number of knots can be inserted, which will make T end up with each cell being a Bézier solid.

7.4.4 Generating RTP-splines

By removing G and the ghost knots that topologically lie inside G, we obtain a RTP-spline, defined over polycube domain P as a single-piece smooth function. The restricted boundary of RTP-spline is guaranteed by the construction steps in Section 7.4.2, Section 7.4.3.2 and Section 7.4.3.3, while the algorithm introduced in Section 7.4.3.1 ensures the partition-of-unity and semi-standardness of RTP-spline. As the denominator is 1 everywhere over the parametric domain, we can simplify

Equation 7.7 to obtain the formulation of RTP-spline:

$$\mathbf{F}(u,v,w) = \sum_{i=1}^{|\mathcal{B}|} w_i \mathbf{p}_i B_i(u,v,w) \quad (u,v,w) \in \mathbb{R}^3 \tag{7.8}$$

7.5 Modeling Solid Objects

It is a challenging task to build single-piece and smooth spline representations for arbitrary solid objects, especially those with bifurcations and high genus. In this section, we explain how to convert a volumetric model M discretized as a tetrahedra mesh M^T, into a RTP-spline representation. Given the triangular boundary of M, we define a polycube domain P adaptive to its geometry and topology, then compute a volumetric mapping $\mathbf{f} : P \to M$. A RTP-spline is constructed on polycubes P by using the method given in Section 7.4, and then we fit the spline to M^T.

7.5.1 Volumetric Parametrization

Computing lowly distorted volumetric parameterization is an important issue for the RTP-spline construction. Tensor-product trivariate splines usually need to be defined over a parametric (box) domain, and the quality of the parameterization can affect the fitting efficacy of splines. In this chapter, we first compute the volumetric harmonic mapping between the given solid object and a predefined polycube, then build our splines upon this polycube domain.

A volumetric parameterization of a solid model M embedded in $\mathbb{R}^3$ on a polycube P is a bijective mapping $\mathbf{f} : P \to M, P, M \subset \mathbb{R}^3$. The polycube P can be computed either manually [135, 136, 155] or automatically [46, 67]. These techniques also provide the boundary mapping $\mathbf{f}'$ from the polycube boundary surface (denoted as ∂P) to the boundary of M (∂M). We use such a surface mapping $\mathbf{f}' : \partial P \to \partial M$ as the boundary condition of f. The volumetric parameterization is then defined as the seeking of a harmonic energy minimizer:

$$\begin{cases} \Delta \mathbf{f}(\mathbf{x}) = 0 & \mathbf{x} \in P, \\ \mathbf{f}(\mathbf{x}) = \mathbf{f}'(\mathbf{x}) & \mathbf{x} \in \partial P. \end{cases}$$

where Δ is the 3-dimensional Laplace operator, defined for each real function f in $\mathbb{R}^3$ as

$$\Delta f = \nabla \cdot \nabla f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}.$$

$\Delta \mathbf{f} = 0$ for $\mathbf{f} = (f^1, f^2, f^3)$ is equivalent to $\Delta f^i = 0$ in all the $i = 1, 2, 3$ coordinate directions.

We compute the volumetric polycube mapping using the method of fundamental solutions (MFS) [65, 66]. The idea and algorithm are recapped as follows and we refer more details to [65, 66].

Based on the maximum principal of harmonic functions, critical points of harmonic functions exist only on the boundary. Furthermore, function values in the interior region of P are fully determined by the boundary values $f(\mathbf{x}), \mathbf{x} \in \partial P$ and can be computed by Green's functions. Specifically, the real harmonic function value $f(\mathbf{x})$ can be computed as the integration of its boundary values and the kernel function (i.e. fundamental solutions associated with the 3D Laplacian operator Δ). The kernel function of Δ has the following formula:

$$K(\mathbf{x}, \mathbf{x}') = \frac{1}{4\pi} \frac{1}{|\mathbf{x} - \mathbf{x}'|},$$

which matches the electrostatistics. In other words, solving a harmonic function can be converted to designing a specific electric field determined by a electronic particle system, whose electric potential mimics f and shall satisfy the boundary condition f' on ∂P.

The computation pipeline is to first place a set of charge points $\{q_s\}$ outside the domain $q_s \in \partial \widetilde{P}, P \subset \widetilde{P} \subset \mathbb{R}^3$. Then we conduct a boundary fitting which solves the charge distribution $\{w_s\}$ on these points $\{q_s\}$. The harmonic function $f(x)$ is represented using the MFS equation:

$$f(\mathbf{x}, W, Q) = \sum_{s=1}^{N_s} w_s \cdot K(\mathbf{x}, q_s), \mathbf{x} \in P, q_s \in \partial \widetilde{P}.$$

The generated function f is guaranteed to be harmonic, and we only need to enforce the boundary condition on ∂P. For the boundary fitting, we sample N_c collocation points on the domain boundary ∂P to set up the constraint equations. If we have N_s charge points and N_c collocation points, for a real harmonic function f (e.g. on an individual axis direction) we only need to solve an $Ax = b$ linear system where A is an $N_c * N_s$ matrix.

Following the algorithm of [66], instead of solving one such big linear system, we compute a set of smaller harmonic functions by solving smaller linear systems and linearly combine them together to get a more accurate boundary fitting. The computation is also more efficient than solving a single big matrix. The mapping $\mathbf{f} : P \to M$ computed here provides us a lowly distorted parameterization of M on the polycube P.

7.5.2 RTP-spline volume fitting

Let $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$ be the vertices of tetrahedra mesh M^t. The problem of fitting RTP-spline $\mathbf{F}(u,v,w)$ to volumetric object M resorts to minimizing the following equation, with respect to control points $\mathbf{p}_i$:

$$\sum_{i=1}^{m}(\mathbf{F}(\mathbf{f}^{-1}(\mathbf{v}_i)) - \mathbf{v}_i)^2 \tag{7.9}$$

which can be rewritten in matrix format of

$$\frac{1}{2}\mathbf{P}^T\mathbf{B}^T\mathbf{B}\mathbf{P} - \mathbf{V}^T\mathbf{B}\mathbf{P} \tag{7.10}$$

where $\mathbf{P}_j = \mathbf{p}_j^T$, $\mathbf{V}_i = \mathbf{v}_i^T$, and $\mathbf{B}_{ij} = \mathbf{I}_{3\times3}B_i(\mathbf{f}^{-1}(\mathbf{v}_j))$. This is a typical least square problem, and we solve it using the optimization package *MOSEK*([1]).

If the fitting results do not meet certain pre-defined criteria, they can always be improved by adaptively performing subdivisions over the regions with large fitting errors and then refitting the spline function $\mathbf{F}$. Each cell to be divided is broken down into two, four or eight small sub-cells, depending on its shape. Our goal is to keep the aspect ratio of sub-cells as low as possible. Then Algorithm 26 is employed to refine RTP-spline blending functions and introduce new degree-of-freedoms for better fitting. Note that such refinement algorithm is originally proposed for trivariate T-splines that defined over box domains, but it can be applied to our RTP-splines after a minor revision: whenever a master-knot is added to a new boundary vertex, additional sub-knots must be inserted as well in order to keep the boundary restriction. What sub-knots are required in this scenario has been explained in Section 7.4.3.2.

7.6 Experimental Results

Model	#Data Points	Control Points	RMS Error	Timing
Bimba	35511	4543	1.20×10^{-3}	31.21s
Kitten	60144	3820	1.27×10^{-3}	44.53s
Eight	26384	2888	3.69×10^{-3}	20.65s
hand	1502700	9035	5.54×10^{-4}	1150s
head	472122	12880	2.91×10^{-4}	422.4s
Beethoven	103361	1001	1.80×10^{-3}	67.79s
Beethoven (2nd level)	103361	3283	1.34×10^{-3}	80.78s
Beethoven (3rd level)	103361	14699	7.18×10^{-4}	123.28s

Table 7.2: Statistics of solid reconstruction with RTP-splines

Model	Sample Points	Polycube Spline			General T-spline		
		$\hat{B}$	$\hat{B}'$	$\hat{B}''$	$\hat{B}$	$\hat{B}'$	$\hat{B}''$
bimba	2512	0.18s	0.6s	1.12s	0.35s	1.14s	2.62s
kitten	23076	1.61s	5.21s	9.59s	2.95s	9.75s	23.1s
eight	9768	0.71s	2.42s	4.36s	1.37s	4.43s	10.2s

Table 7.3: Comparison between the computational cost for calculating spline basis functions and their derivatives on RTP-splines and those on general T-splines. The total cost is to compute basis functions and their directives at all sample points. The basis functions of RTP-splines are $\hat{B}_i(u,v,w) = w_i B_i(u,v,w)$, while those of T-splines are $\hat{B}_i(u,v,w) = w_i B_i(u,v,w) / \sum_{j=1}^{|\mathcal{B}|} w_j B_j(u,v,w)$.

A system consisting of RTP-spline construction, volumetric parametrization and data fitting is implemented in VC++ and the program is running on a 3GHz Pentium-IV PC with 4GRAM. Our experimental data include the volumetric models of bimba, beethoven, the Eight(genus 2), kitten(genus 1), hand(5 bifurcations), and head(with brain excavated).

In our experiments, it takes only seconds to construct RTP-splines from experimental solid models, in which the Beethoven model of level 3 takes the longest time: 6 seconds. In comparison, fitting RTP-splines to data is much more computationally expensive, and their statistics are documented in Table.7.2, in which the data points are the vertices of original tetrahedral meshes, and the fitting results are measured by RMS errors that are normalized to the dimension of corresponding solid models. The table shows that the volumetric fitting on proposed RTP-splines is computationally efficient and can yield reasonable results. Also, the capability of RTP-splines in supporting local refinements allows users to improve fitting results by adaptively introducing new degree-of-freedoms to the regions with large errors. For example, the RMS fitting errors on beethoven models is reduced from 1.80×10^{-3} to 7.18×10^{-4} after two rounds of adaptive local subdivision, and the geometric details on model surfaces are also gradually revealed (Figure 7.11).

Due to the uniqueness of RTP-splines construction, computing blending functions and their derivatives on RTP-splines is much faster than that on traditional T-splines. To demonstrate this, we conducted a few experiments and compare the evaluation times for spline functions and their derivatives on RTP-splines and those on traditional T-splines. The comparisons are conducted on bimba, kitten, and the Eight models. For the purpose of fairness, we slightly modify RTP-spline source code to emulate T-spline, by adding denominator calculation for every basis function. The comparison results are documented in Table 7.3. They show that the computational costs for calculating $\hat{B}$, $\hat{B}'$ and $\hat{B}''$ on RTP-splines are respectively reduced to 53%, 54% and 42% of those on general T-splines.

By increasing the dimension of control point $\mathbf{p}$ in Equation 7.8, RTP-splines can model not only geometry but also other attributes simultaneously. To demonstrate this, we synthesized texture information for the head model, and then fit a RTP-spline to both its geometry and associated attributes. Two types of textures are synthesized. One is the distance field to the boundary (head surface and the insider brain surface). The other is a procedure 3D texture, generated by the fractal sum of Perlin noise [84] ($\sum_{i=1}^{4} noise(i\mathbf{p}), \mathbf{p} \in \mathbb{R}^3$). The fitting result is shown in Figure 7.12. As the distance values are proportional to the dimension of head model, we still use normalized RMS error to measure the fitting error for distance field, which is 6.0×10^{-4} in our experiment. For the noise texture, the maximum scale value of original noise function is 0.81 and the minimum is -1.33. The absolute RMS fitting error is 7.3×10^{-4}.

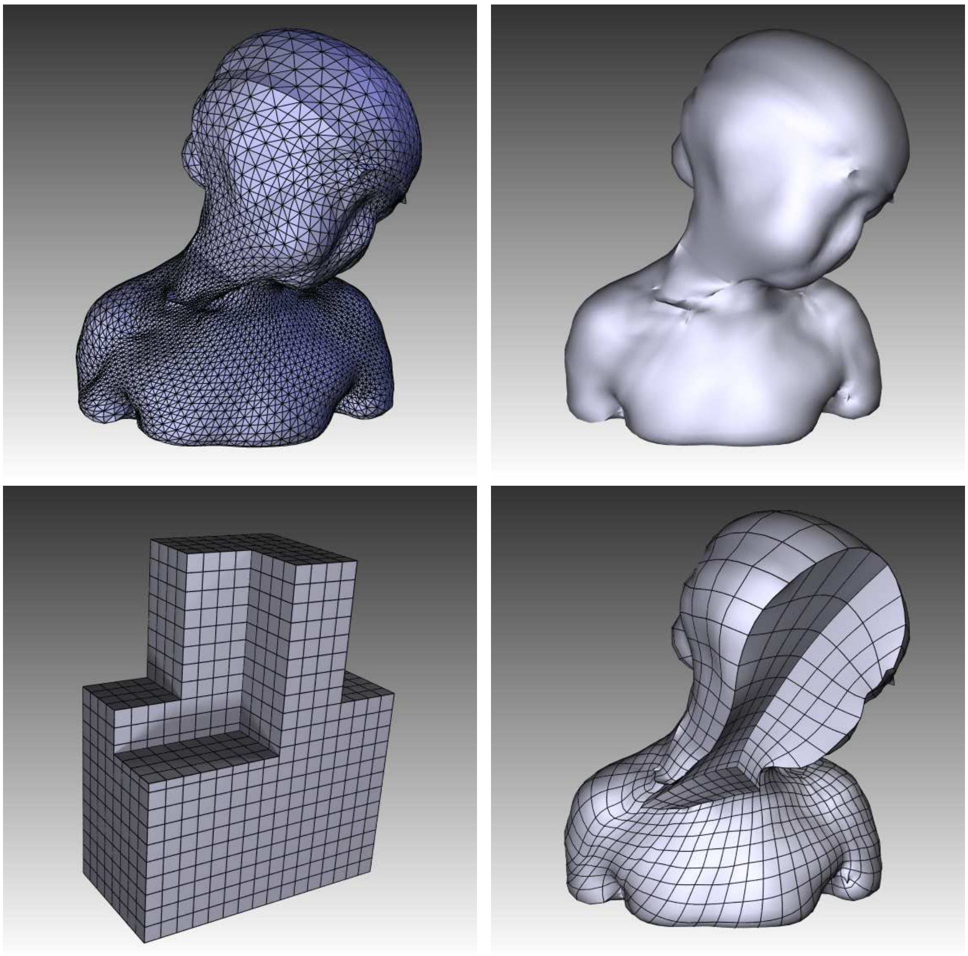


Figure 7.7: RTP-spline volume generated from the bimba tetrahedron meshes.

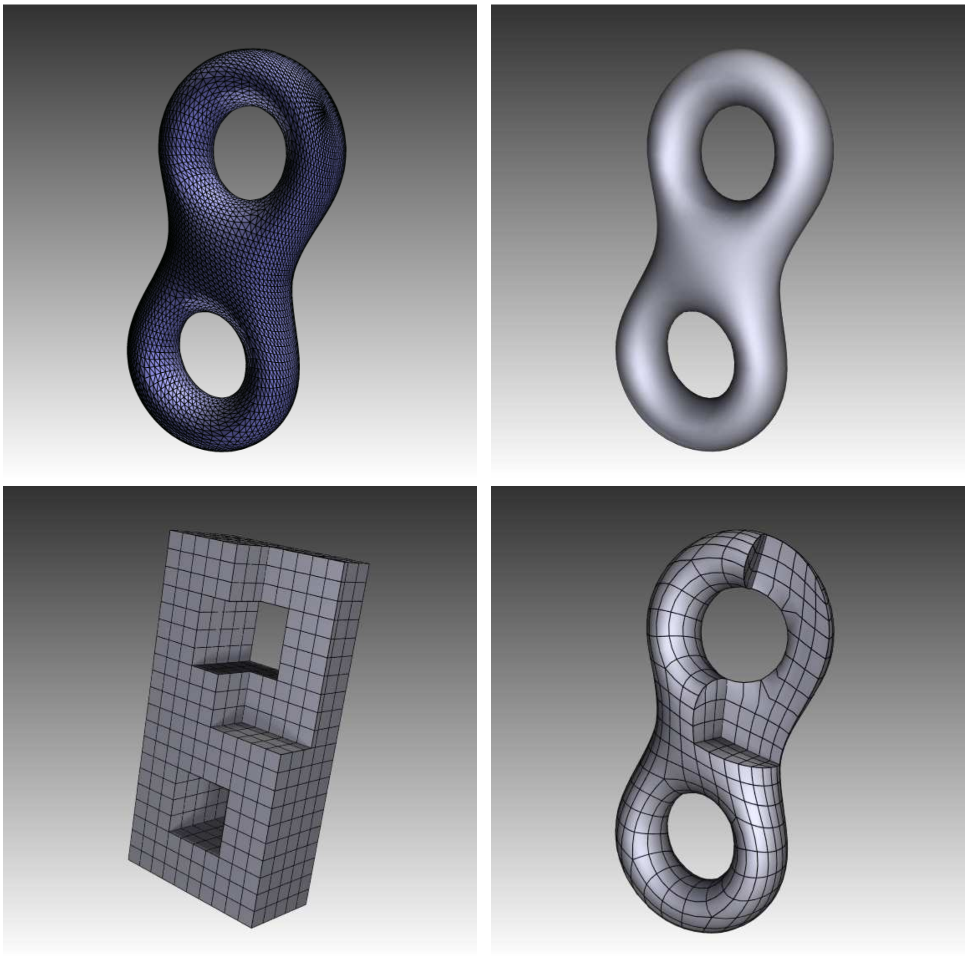


Figure 7.8: RTP-spline volume generated from the “eight” tetrahedron meshes.

7.7 Summary

In this chapter we propose the concept and construction algorithm of RTP-splines and present an effective framework to convert volumetric data into representations of RTP-splines. Due to the topological flexibility of the polycube domain, RTP-spline can naturally model solid objects with bifurcations and high genus, while ensuring lower parametrization distortions in comparison to traditional splines defined over box domains. Our novel algorithm guarantees that the initially

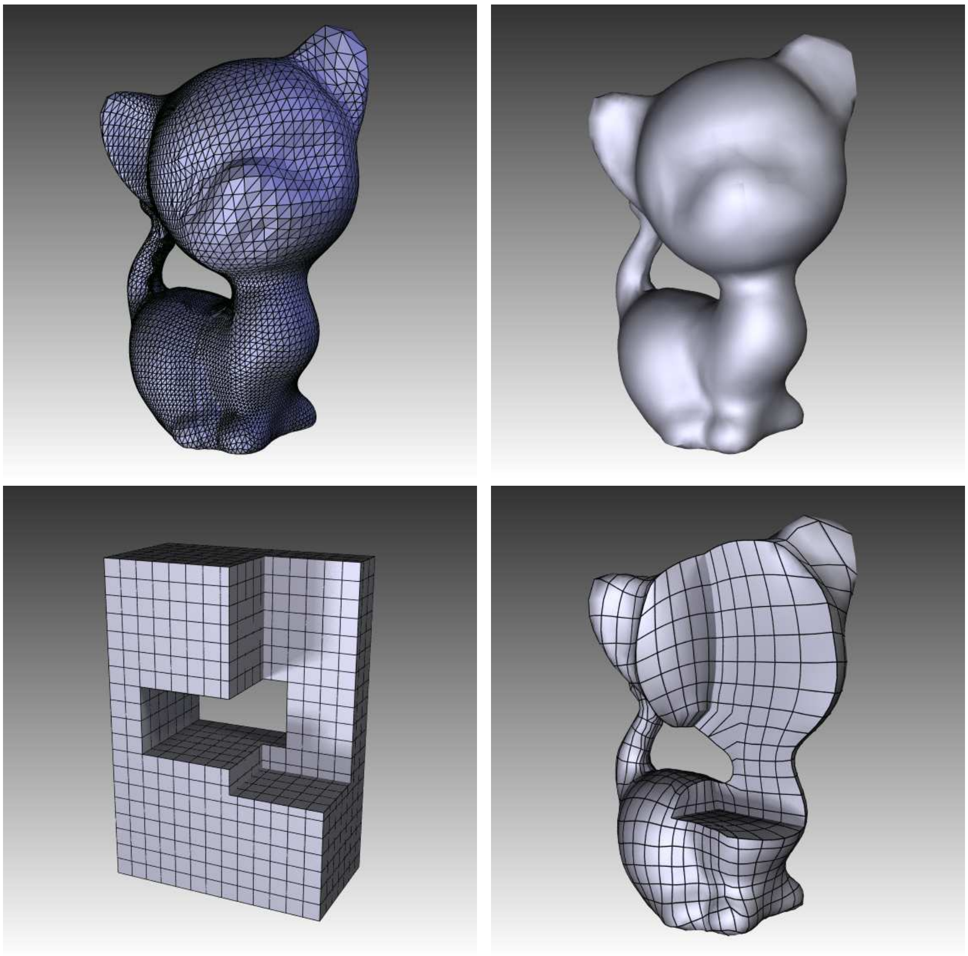


Figure 7.9: RTP-spline volume generated from the kitten tetrahedron meshes.

constructed RTP-splines is semi-standard, so that it enables the efficient computation of spline functions and their derivatives, without the overhead of dividing the sum of all basis functions. The proposed RTP-spline supports local refinement, and a refinement algorithm is developed to preserve the semi-standardness on the RTP-splines undergoing knot insertion and local subdivision. The particular restricted boundary requirement of RTP-spline presents control points from affecting domain regions by crossing boundaries. We demonstrate the efficacy of our RTP-splines as a powerful solid modeling tool in various experiments, which convert tetrahedral models into RTP-splines representations, modeling both their geometries and

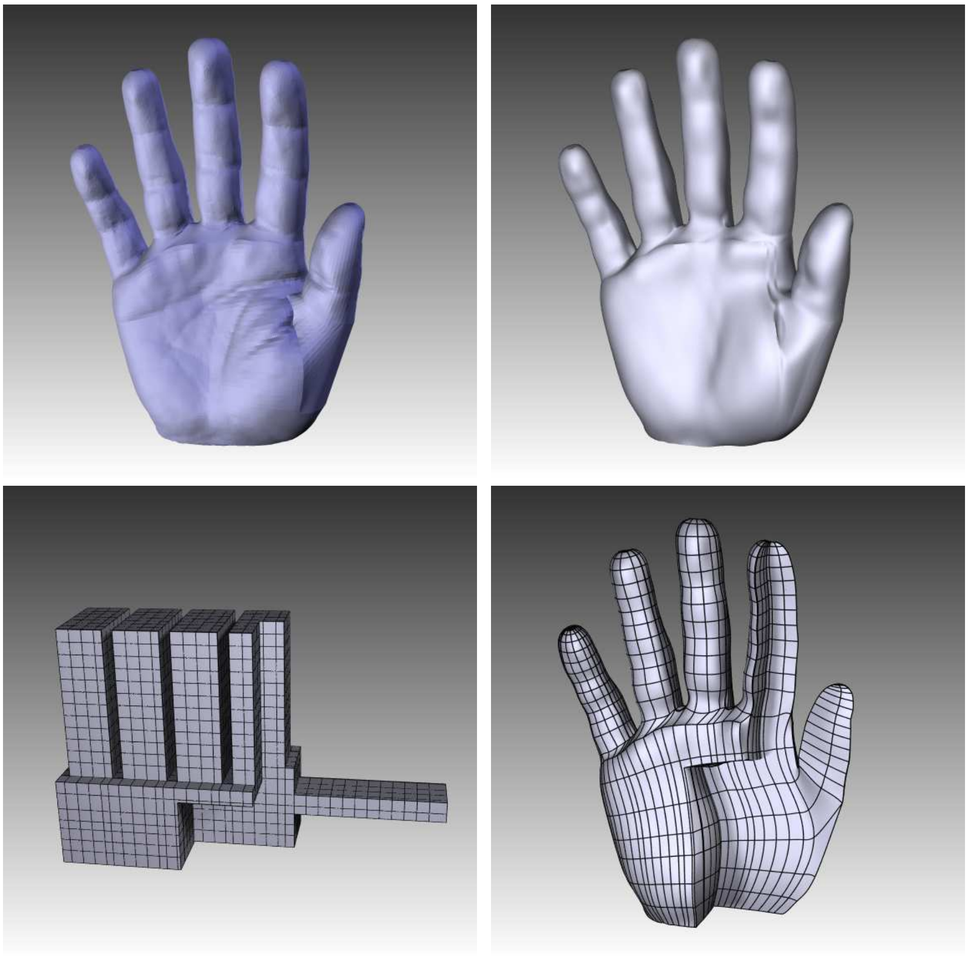


Figure 7.10: RTP-spline volume modeled from the hand tetrahedron meshes.

other attributes within an unified paradigm. Due to those favorable features of RTP-splines, we will explore the isogeometric analysis founded upon RTP-splines in the near future. Also, the particular polycube domains of RTP-splines can be naturally decomposed into a set of regular structures, which may enable GPU-friendly computing and image-based geometric shape processing.

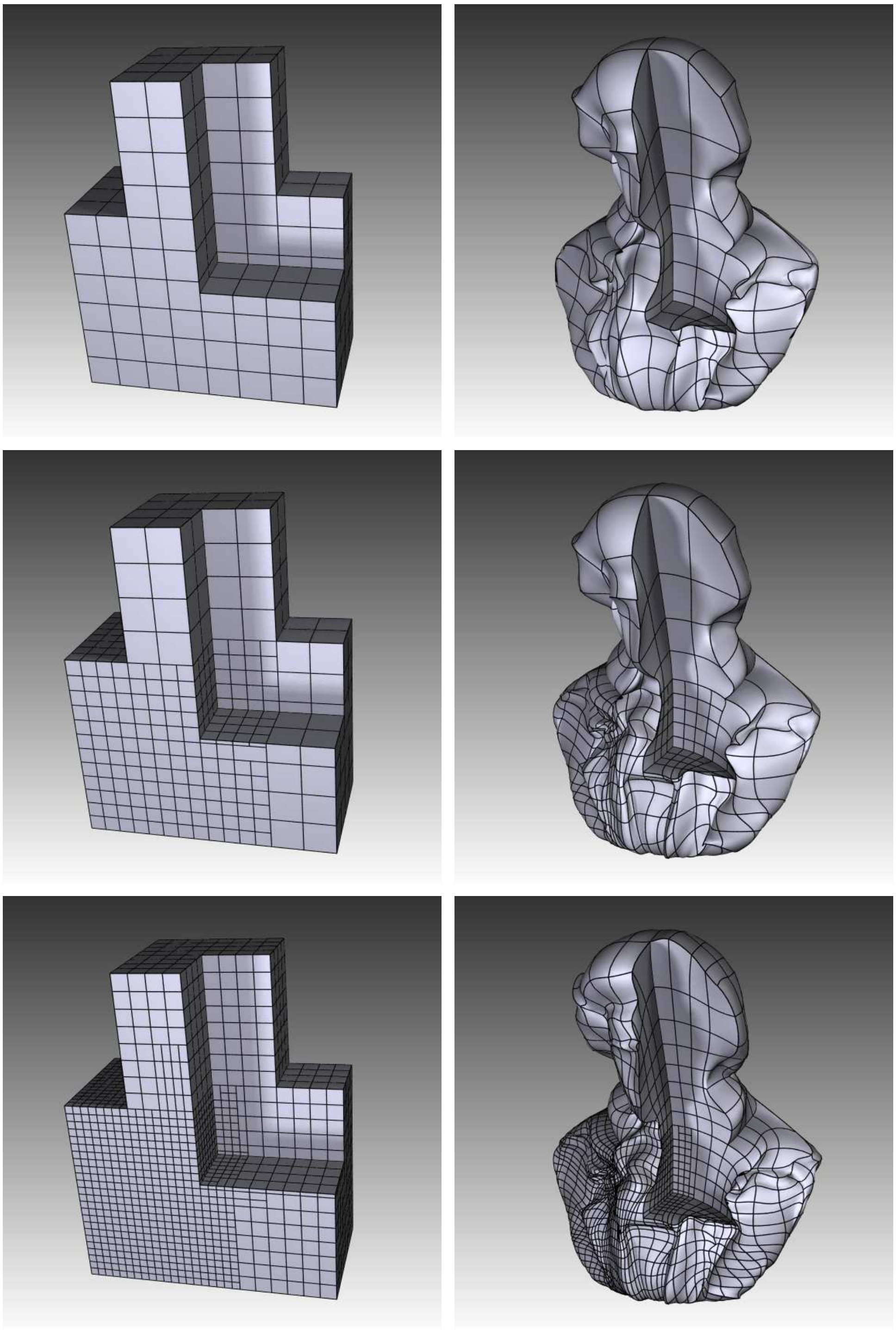


Figure 7.11: From top to bottom, the pictures show the fitting results for Beethoven model, after 0, 1 and 2 levels of local subdivision respectively.

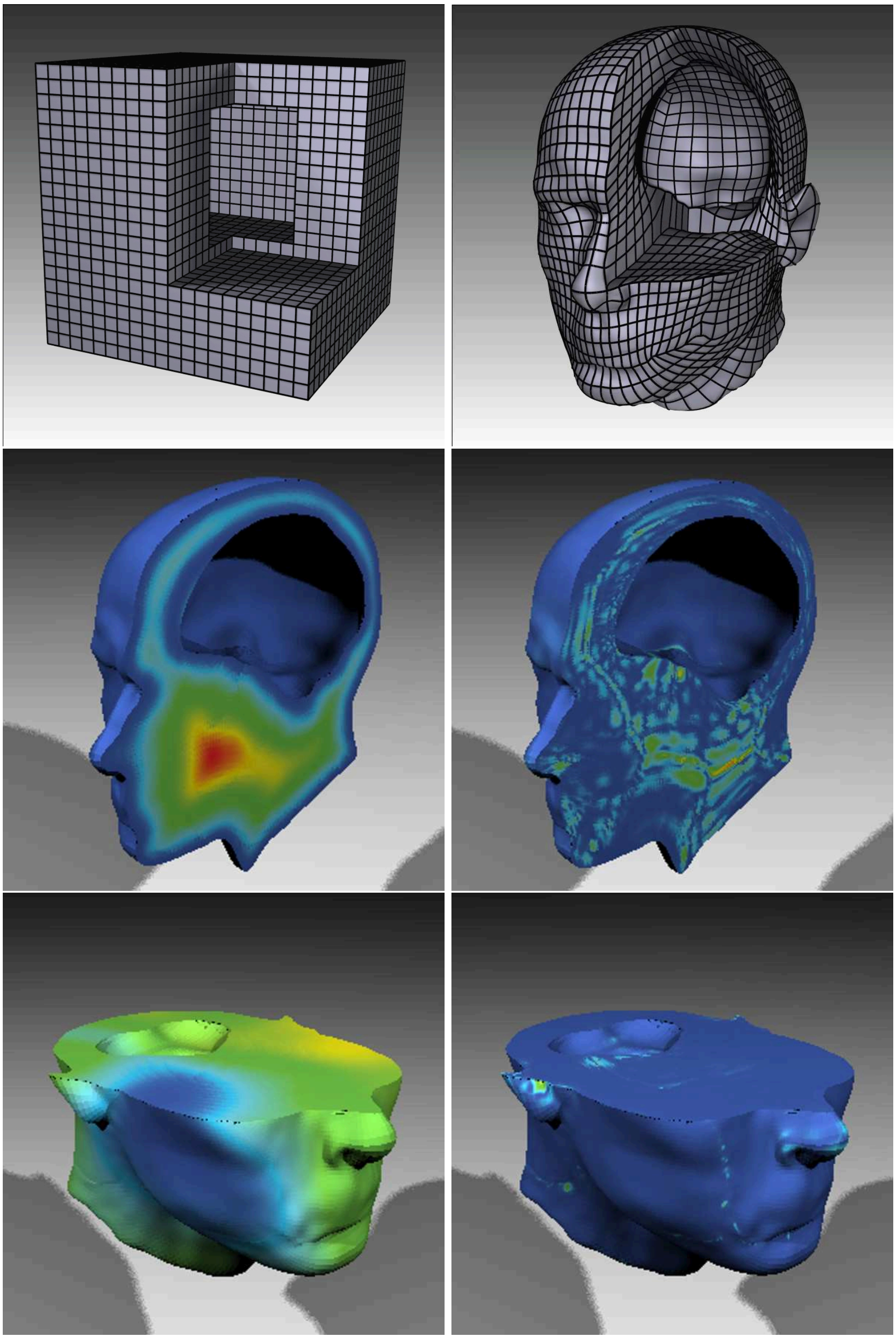


Figure 7.12: Fitting both geometry and attributes of the head model. The first row show the polycube domain(left) with one corner removed to reveal internal structures, and the volumetric meshes(right) generated from fitted RTP-splines. The second row show the fitted result(left) for the distance-field texture and its error map(right). The third row show the fitted result(left) for synthesized noise texture and its error map(right). In these pictures, red color denotes the largest scalar value while blue color denotes the smallest one.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

In this dissertation, we presented an computational framework based on the theory of multivariate splines and used it to solve a series of practical problems successfully. Triangular B-splines and T-splines are mainly employed in the framework due to the fact that they have not received enough attentions outside computer graphics communities while their unique advantageous features could potentially benefit many applications in research and engineering areas. In addition, other techniques, such as feature extraction and curve matching, are also developed and incorporated into our framework for tackling specific problems. Through our extensive experiments, the proposed framework proves as a successful and effective tool for solving practical variational problems found in geometric shape modeling, surface editing, imaging processing and scientific computation. Our specific contributions include:

1. We introduced a registration algorithm that recovers global non-rigid transformation between two images while keeping the sharp features near pre-identified local rigid structures. With proper knots alignment along feature lines, we effectively model both global non-rigidity and local rigidity in an unified displacement field represented by triangular B-spline. In contrast to conventional methods where additional mathematical terms are in need for feature constraints, our method recovers local rigidity with fewer degrees of freedom and accurately models sharp features in a more nature way [139].
2. We developed triangular B-spline finite element method (TBFEM), and use it to solve elastic PDE on a pseudo physical model, in order to simulate the deformation between a pair of temporal mammograms. The discontinuities of the deformation due to the different elasticities of pectoral muscles and breast tissues are naturally modeled by triangular B-splines. The experimental results show that the proposed registration method is effective and accurate [140].
3. We proposed an automatic mammogram registration algorithm. The breast contours are first aligned according to the curvature map. The significant anisotropic texture features are extracted and paired. The alignment between two mammograms is finally inferred from the deformation on a simple spline-based elastic model, which is stimulated by intensity similarity forces and constrained by paired features. The method was tested on a series of selected bilateral mammogram images and proves a effective registration approach requiring little human intervention [141].
4. We developed a new paradigm integrates thin-shell simulation with manifold spline surfaces for geometric design. We particularly employed triangular B-spline to solve Kirchhoff-Love equations on manifold domain which is

globally parameterized. The proposed framework can successfully simulate thin-shell deformation of arbitrary manifold shell surfaces with non-trivial topologies [138].

5. We proposed a method to construct restricted trivariate polycube splines, where are inspired by general volumetric T-splines, but differs significantly in its construction way. Our top-down approach starts from the bounding box of a given polycubic domain, converts it into B-spline volume, performs special knot insertion and blending function refinements, and then remove exterior cells of the polycubic structure. Aside from the ability of local subdivision inherit from T-splines, the proposed RTP-splines are bestowed with three unique features, domain flexibility, restricted boundary and fast basis function evaluation, all of which are desirable for engineering analysis and other applications on solid objects and volumetric data.

8.2 Future Work

Although B-splines nowadays play an important role in many research and engineering areas, other multivariate spline schemes recently developed having not been widely recognized outside computer graphics communities. Moreover, with the advent of computing power and data acquisition technique, more and more applications are required to handle volume/solid data. Therefore, we aim the future work in the following directions:

1. **Image registration with novel multivariate splines** The goal of image registration is to find an optimal free-form transformation between two images. Currently B-spline is prevailingly used in computer vision to model deformation field. However, there are many novel new spline schemes can benefit

image registration because of their special advantages, such as local refinement and feature modeling. One example would be image registration with T-splines. Xie and Farin [150] preserve the image details in their registration method by using hierarchical B-splines. However, multiple layers of grids have to be applied from coarse to dense, which is computation intensive. On the contrary, T-splines have more compact and flexible control region, thus is ideal to model different level-of-detail on a single layer. Another example is to register user-specified non-rectangular regions of a pair of volumetric images, which could be modeled by the trivariate splines proposed in Chapter 7.

2. **Precise iso-surface extraction and visualization** One approach to extract and display ios-surfaces from an unorganized point cloud is to construct implicit representation of underlying surfaces, then extract the geometric surfaces with marching cube algorithm and render the meshes. Rather than render an approximate mesh, it's however more desirable to visualize the implicit surfaces directly. This requires calculation of intersection points between casting rays and the implicit surfaces, and differential properties, *i.e.*normals and tangential planes, at these locations. The implicit representation generated from the approach [80] is not an unified analytical formula, therefore unable to be used for precise intersection point calculation. Volumetric T-spline is a good candidate for implicit function representation due to the following facts. First, T-splines are defined on regular domains so that the intersection points can be analytically extracted by solving a cubic equation (in the case of cubic T-splines). Second, T-splines can model fine details in implicit functions via local refinement.

3. **Volumetric data denoise and fairing** The data obtained from real world are oftentimes contaminated by equipment defect or unpredicted incidents during

data acquisition. Many techniques have been invented for data cleaning. For example, Gaussian filter is used to get rid of high frequency noise. In other works, even missing part in scanned range data can reconstructed through radial basis function interpolation along with the help of certain heuristic information. More recently, variational models are proposed for implicit surface fairing [5, 6, 43, 144]. However, in this work all the data points are organized on a regular grid and differential properties are estimated from nearby grid points, both of which hinder calculation in variational model from achieving accurate results. Therefore it might be ideal for multivariate splines to take the role in modeling implicit functions. Implicit function thus represented by splines is a single piece analytical formula, on which accurate evaluation of fairing equations can be performed anywhere. Moreover, some novel splines, such as T-splines, can be locally refined, allowing users to keep fairing errors below desired thresholds without global refinement.

4. **Isogeometric analysis based on RTP-splines** Isogeometric analysis [53] is a recently developed approach that integrates finite element analysis directly on geometric objects modeled by NURBS in CAD system, without changing the geometries or their parametrization. The advantage of this method is obvious, that is, the basis functions are shared between shape design and physical analysis in CAD system, thus no data conversion is necessary and the exact geometry is maintained at different stages. However, NURBS is tensor-product spline that can not naturally model solid objects having bifurcations and arbitrary topologies without enormous efforts in patching and trimming. We have demonstrated in Chapter 7 that our proposed RTP-splines can successfully model complicated solid objects with bifurcations and nontrivial topologies. Moreover, computing basis functions and their derivatives

over RTP-splines is quite efficient due to the unique construction way, where POU is guaranteed without the necessity of summing and dividing all basis functions everywhere. All of these advantageous features make RTP-spline a promising tool in the field of isogeometric analysis for solid objects.

Bibliography

[1] Mosek, http://www.mosek.com/.

[2] E. N. Aksan and A. Özdes. Numerical solution of kortewegcde vries equation by galerkin b-spline finite elementnext term method. In *Applied Mathematics and Computation*, volume 175, pages 1256–1265, 2006.

[3] F. J. Ayres and R. M. Rangayyan. Performance analysis of oriented feature detectors. In *Proc. of SIBGRAPI*, volume XVIII, October 2005.

[4] Y. Bazilevs, V. Calo, J. Cottrell, J. Evans, S. Lipton, M. Scott, and T. Sederberg. Isogeometric analysis using t-splines. *Computer Methods in Applied Mechanics and Engineering*, 2009.

[5] M. Bertram. Fairing scalar fields by variational modeling of contours. In *IEEE Visualization*, pages 387–392, 2003.

[6] M. Bertram. Volume refinement fairing isosurfaces. In *IEEE Visualization*, pages 449–456, 2004.

[7] W. Bohm, G. Farin, and J. Kahmann. A survey of curve and surface methods in cagd. In *Computer Aided Gemetric Design*, volume 1, pages 1–60, 1984.

[8] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(6):567–585, 1989.

[9] C. d. Boor. On calculating with *b*-splines. *J. Approx. Theory*, 6(1):50–62, 1972.

[10] L. G. Brown. A survey of image registration techniques. *ACM Comput. Surv.*, 24(4):325–376, 1992.

[11] M. Bucalem and K. Bathe. Finite element analysis of shell structures. *Arch. Comput. Method Eng.*, 4:3–61, 1997.

[12] P. Cachier, J.-F. Mangin, X. Pennec, D. Rivière, D. Papadopoulos-Orfanos, J. Régis, and N. Ayache. Multisubject non-rigid registration of brain mri using intensity and geometric features. In *MICCAI*, pages 734–742, 2001.

[13] D. L. Cardon. T-spline simplification. Master's thesis, Brigham Young University, Provo, UT 84602, USA, 2007.

[14] G. Celniker and W. Welch. Linear constraints for deformable non-uniform b-spline surfaces. In *Proc. ACM Symp. Interactive 3D Graphics*, pages 165–170, 1992.

[15] P. G. Ciarlet. *Mathematical elasticity, Volume 1: Three-dimensional elasticity, Volume 20 of studies in mathematics and its applications*. North-Holland, Amsterdam, 1988.

[16] F. Cirak, M. Ortiz, and P. Schröder. Subdivision surfaces: a new paradigm for thin-shell finite element analysis. *Int. J. Numer. Methods Eng.*, 47:2039–2072, 2000.

[17] A. Collignon. *Multi-modality medical image registration by maximization of mutual information*. PhD thesis, Catholic University of Leuven, Leuven, Belgium, 1998.

[18] H. B. Curry. Review. *Math. Tables Aids Comput.*, 2:167–169, 211–213, 1947.

[19] I. Dag, S. Bulent, and A. Boz. B-splinenext term galerkin methods for numerical solutions of the burgers equation. In *Applied Mathematics and Computation*, volume 116, pages 506–522, 2005.

[20] W. Dahmen, C. A. Micchelli, and H.-P. Seidel. Blossoming begets *B*-spline bases built better by *B*-patches. *Mathematics of Computation*, 59(199):97–115, 1992.

[21] D. H. Davies and D. R. Dance. Automatic computer detection of clustered calcifications in digital mammograms. *Phys. Med. Biol.*, 35:1111–1118, 1990.

[22] M. Davis, A. Khotanzad, D. Flamig, and S. Harms. A physics-based coordinate transformation for 3-d image matching. 16(3):317–328, June 1997.

[23] O. R. Davis and R. P. Burton. Free-form deformation as an interactive modeling tool. In *Imaging Technology*, volume 17, pages 181–187, 1991.

[24] C. de Boor. Splines as linear combinations of *B*-splines. A survey. In *Approximation theory, II (Proc. Internat. Sympos., Univ. Texas, Austin, Tex., 1976)*, pages 1–47. Academic Press, New York, 1976.

[25] C. deBoor, K. Höllig, and S. Riemenschneider. *Box splines*. Springer Verlag, 1993.

[26] P. Dierckx. On calculating normalized powell-sabin b-splines. *Computer Aided Geometric Design*, 15(1):61–78, 1997.

[27] P. Dierckx, S. Van Leemput, and T. Vermeire. Algorithms for surface fitting using Powell-Sabin splines. *IMA Journal of Numerical Analysis*, 12:271–299, 1992.

[28] M. M. Doyley, P. M. Meaney, and J. C. Bamber. Evaluation of an iterative reconstrution method for quantitative elastography. *Phys. Med. Biol.*, 45:1521–1540, 2000.

[29] V. Duay, P.-F. D'Haese, R. Li, and B. M. Dawant. Non-rigid registration algorithm with spatially varying stiffness properties. In *ISBI*, pages 408–411, 2004.

[30] P. J. Edwards, D. L. G. Hill, J. A. Little, and V. A. Sahni. Medical image registration incorporating deformations. In *BMVC*, pages 691–699, 1995.

[31] G. Farin. From conics to nurbs: A tutorial and survey. In *IEEE Computer Graphics and Applications*, volume 12, pages 78–86, 1992.

[32] R. Feichtinger, M. Fuchs, B. Jüttler, O. Scherzer, and H. Yang. Dual evolution of planar parametric spline curves and t-spline level sets. *Computer-Aided Design*, 40(1):13–24, 2008.

[33] P. Fong and H.-P. Seidel. Control points for multivariate B-spline surfaces over arbitrary triangulations. *Computer Graphics Forum*, 10(4):309–317, 1991.

[34] D. Forsey and R. H. Bartells. Hierarchical b-spline refinement. In *Computer Graphics*, volume 22, pages 205–212, 1988.

[35] M. Franssen, R. C. Veltkamp, and W. Wesselink. Efficient evaluation of triangular b-spline surfaces. *Computer Aided Geometric Design*, 17(9):863–877, 2000.

[36] P. Gill, W. Murray, and M. Wright. *Practical optimization*. Academic Press, 1981.

[37] R. Gormaz and P.-J. Laurent. Some results on blossoming and multivariate *B*-splines. In *Multivariate approximation: from CAGD to wavelets (Santiago, 1992)*, volume 3 of *Ser. Approx. Decompos.*, pages 147–165. World Sci. Publishing, River Edge, NJ, 1993.

[38] G. Greiner and H.-P. Seidel. Modeling with triangular *B*-splines. *IEEE Computer Graphics and Applications*, 14(2):56–60, Mar. 1994.

[39] X. Gu, Y. He, and H. Qin. Manifold splines. In *Proc. ACM Symp. Solid and Physical Modeling*, pages 27–38, 2005.

[40] X. Gu and S.-T. Yau. Computing conformal structures of surfaces. *Communications in Information and Systems*, 2(2):121–146, 2002.

[41] X. Gu and S.-T. Yau. Global conformal surface parameterization. In *Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 127–137, 2003.

[42] L. Hadjiiski, H. Chan, B. Sahiner, N. Petrick, and M. A. Helvie. Automated registration of breast lesions in temporal pairs of mammograms for interval changes analysis-local affine transformation for improved localization. *Med. Phys.*, 28(6):1070–1079, 2001.

[43] H. Hagen, G. Brunett, and P. Santarelli. Variational principles in curve and surface design. In *Surveys on Mathematics for Industry*, volume 3, pages 1–27, 1993.

[44] Y. He, M. Jin, X. Gu, and H. Qin. A c^1 globally interpolatory spline of arbitrary topology. In *VLSM*, pages 295–306, 2005.

[45] Y. He and H. Qin. Surface reconstruction with triangular b-splines. In *GMP*, pages 279–290, 2004.

[46] Y. He, H. Wang, C.-W. Fu, and H. Qin. Technical section: A divide-and-conquer approach for automatic polycube map construction. *Comput. Graph.*, 33(3):369–380, 2009.

[47] P. Hellier, C. Barillot, I. Corouge, B. Gibaud, G. L. Goualher, D. L. Collins, A. Evans, G. Malandain, N. Ayache, G. E. Christensen, and H. J. Johnson. Retrospective evaluation of intersubject brain registration. In *IEEE Trans. Med. Imag.*, volume 22, pages 1120–1130, 2003.

[48] G. Hermosillo, C. Chefd'Hotel, and O. D. Faugeras. Variational methods for multimodal image matching. *International Journal of Computer Vision*, 50(3):329–343, 2002.

[49] W. Hsu, J. Hughes, and H. Kaufman. Direct manipulation on free-form deformation. In *ACM SIGGRAPH '92*, volume 26, pages 177–184, 1992.

[50] J. Hua, Y. He, and H. Qin. Trivariate simplex splines for inhomogeneous solid modeling in engineering design. *ASME Transactions: Journal of Computing and Information Science in Engineering*, June 2005. in press.

[51] J. Hua and H. Qin. Dynamic implicit solids with constraints for haptic sculpting. *smi*, 00:119, 2002.

[52] J. Hua and H. Qin. Haptics-based volumetric modeling using dynamic spline-based implicit functions. In *VVS '02: Proceedings of the 2002 IEEE symposium on Volume visualization and graphics*, pages 55–64, Piscataway, NJ, USA, 2002. IEEE Press.

[53] T. J. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2004.

[54] H. Ipson. T-spline merging. Master's thesis, Brigham Young University, Provo, UT 84602, USA, 2005.

[55] C. T. Kelley. *Iterative methods for linear and nonlinear equations*. SIAM, Philadelphia, 1995.

[56] A. K. Khalifa, K. R. Raslan, and H. M. Alzubaidi. A collocation method with cubic b-splinesnext term for solving the mrlw equation. In *Journal of Computational and Applied Mathematics*, volume 212, pages 406–418, 2008.

[57] B. Kim. Affine versus warping in coregistration of mr imaging and autoradiography. In *Radiology*, volume 197, pages 161–172, 1994.

[58] B. Kim. Mutual information for automated multimodal image warping. In *SPIE Med. Imaging*, pages 349–354, 1996.

[59] Y. Kita, R. Highnam, and M. Brady. Correspondence between different view breast x-rays using a simulation of breast deformation. In *CVPR*, pages 700–707, June 1998.

[60] P. J. Kostelec, J. B. Weaver, and D. M. Healy. Multiresolution elastic image registration. *Med. Phys.*, 25(9):1593–1604, 1998.

[61] R. Kraft. Adaptive and linearly independent multilevel *b*-splines. In *Surface Fitting and Multiresolution Methods*, pages 209–218. Vanderbilt University Press.

[62] P. Krysl and T. Belytschko. Analysis of thin plates by the element-free galerkin method. *Comput. Mech.*, 17:26–35, 1996.

[63] H. Lamousin and W. Waggenspack. Nurbs-based free-form deformation. In *IEEE Computer Graph Appl.*, volume 14, pages 59–65, 1994.

[64] T. K. Lau and W. F. Bischof. Automated detection of breast tumors using the asymmetry approach. *Comp. Biomed. Res.*, 24:273–295, 1991.

[65] X. Li, X. Guo, H. Wang, Y. He, X. Gu, and H. Qin. Meshless harmonic volumetric mapping using fundamental solution methods. *IEEE Trans. on Automation Science and Engineering*, 6, 2009.

[66] X. Li, H. Xu, S. Wan, Z. Yin, and W. Yu. Feature-aligned harmonic volumetric mapping using mfs. *Comput. Graph., to appear*, 2010.

[67] J. Lin, X. Jin, Z. Fan, and C. Wang. Automatic polycube-maps. In *proc. Gometric Modeling and Processing*, pages 3–16, 2008.

[68] J. A. Little, D. L. G. Hill, and D. J. Hawkes. Deformations incorporating rigid structures. *Computer Vision And Image Understanding*, 66(2):223–232, May 1997.

[69] D. Loeckx, F. Maes, D. Vandermeulen, and P. Suetens. Nonrigid image registration using free-form deformations with a local rigidity constraint. In *MICCAI (1)*, pages 639–646, 2004.

[70] R. MacNeal. Perspective on finite elements for shell analysis. *Finite Elem. Anal. Des.*, 30:175–186, 1998.

[71] E. Maeland. On the comparison of interpolation methods. *IEEE Trans Med. Imag.*, MI-7:213–217, 1988.

[72] J. Maintz and M. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.

[73] C. Manni and P. Sablonniere. Quadratic spline quasi-interpolants on powell-sabin partitions. 2004. submitted.

[74] T. Martin, E. Cohen, and M. Kirby. Volumetric parameterization and trivariate b-spline fitting using harmonic functions. In *SPM '08: Proceedings of the 2008 ACM symposium on Solid and physical modeling*, pages 269–280, 2008.

[75] T. Martin, E. Cohen, and R. M. Kirby. Volumetric parameterization and trivariate b-spline fitting using harmonic functions. *Comput. Aided Geom. Des.*, 26(6):648–664, 2009.

[76] W. Martin and E. Cohen. Representation and extraction of volumetric attributes using trivariate splines. In *Proceedings of the 7th ACM symposium on solid modeling and applications*, pages 234–240, 2001.

[77] T. Mclnerney and D. Terzopoulos. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1:91–108, 1996.

[78] C. R. Meyer, J. L. Boes, B. Kim, P. H. Bland, K. R. Zasadny, P. V. Kison, K. Koral, K. A. Frey, and R. L. Wahl. Demonstration of accuracy and clinical versatility of mutual information for automatic multimodality image fusion using affine and thin-plate spline warped geometric deformations. *Med. Image Anal.*, 1(3):195–206, 1996/7.

[79] C. A. Micchelli. On a numerically efficient method for computing multivariate b-splines. *Multivariate approximation theory*, 51:211–248.

[80] Y. Ohtake, A. Belyaev, and H.-P. Seidel. Multi-level partition of unity implicits. *ACM Trans. Graph.*, pages 463–470, 2003.

[81] T. Ojala, J. Näppi, and O. Nevalainen. Accurate segmentation of the breast region from digitized mammograms. *Comp. Med. Imaging & Graphics*, 25:57–59, 2001.

[82] S. Paquerault, L. M. Yarusso, J. Papaioannou, Y. Jiang, and R. M. Nishikawa. Radial gradient-based segmentation of mammographic microcalcifications. *Med. Phys.*, 31(9):2648–2657, 2004.

[83] P. Pathmanathan, D. Gavaghan, J. Whiteley, S. M. Brady, M. Nash, P. Nielsen, and V. Rajagopal. Predicting tumore location by simulating large

deformation of the breast using a 3d finite element model and nonlinear elasticity. pages 217–224, 2004.

[84] K. Perlin. An image synthesizer. *SIGGRAPH Comput. Graph.*, 19(3):287–296, 1985.

[85] T. Peters, B. Davey, P. Munger, R. Comeau, A. Evans, and A. Olivier. Three-dimensional multimodal image-guidance for neurosurgery. *IEEE Transactions on medical imaging*, 15(2):121–128, 1996.

[86] R. Pfeifle and H.-P. Seidel. Faster evaluation of qudartic bivariate dms spline surfaces. In *Proceedings of Graphics Interface'94*, pages 182–189, 1994.

[87] R. Pfeifle and H.-P. Seidel. Fitting triangular *B*-splines to functional scattered data. In *Graphics Interface '95*, pages 26–33, 1995.

[88] L. Piegl. Modifying the shape of rational *b*-spline. part 1:curves. In *Computer-Aided Design*, volume 21, pages 509–518, 1989.

[89] L. Piegl. Modifying the shape of rational *b*-spline. part 1:surfaces. In *Computer-Aided Design*, volume 21, pages 538–546, 1989.

[90] L. Piegl. On nurbs: A survey. In *IEEE Computer Graphics and Applications*, volume 11, pages 55–71, 1991.

[91] L. Piegl. On nurbs: A survey. *IEEE Computer Graphics and Applications*, 11(1):55–71, 1991.

[92] L. Piegl and W. Tiller. Curve and surface constructions using rational *b*-splines. In *Computer-Aided Design*, volume 19, pages 485–498, 1987.

[93] L. A. Piegl and W. Tiller. *The NURBS Book*. Springer, 2 edition, 1996.

[94] P. Potamianos, B. Davies, and R. D. Hibberd. Intraoperative registration for percutaneous surgery. In *First International Symposium on Medical Robotics and Computer Asisted Surgery*, volume 1, pages 98–105, 1995.

[95] M. J. D. Powell and M. A. Sabin. Piecewise quadratic approximations on triangles. *ACM Trans. Math. Softw.*, 3(4):316–325, 1977.

[96] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 1992.

[97] H. Qin and D. Terzopoulos. D-NURBS: A physics-based framework for geometric design. *IEEE Trans. Vis. Comput. Graph.*, 2(1):85–96, 1996.

[98] H. Qin and D. Terzopoulos. Triangular NURBS and their dynamic generalizations. *Computer Aided Geometric Design*, 14(4):325–347, 1997.

[99] L. Ramshaw. Blossom are polar forms. *Computer-Aided Geom. Design*, 6(4):323–358, 1989.

[100] F. Richard and L. Cohen. *A new Image registration technique with free boundary constraints : application to mammography*, volume 89. 2003.

[101] F. Richard and C. Graffigne. An image-matching model for the registration of temporal or bilateral mammogram pairs. pages 756–762, June 2000.

[102] A. Roche, G. Malandain, X. Pennec, and N. Ayache. The correlation ratio as a new similarity measure for multimodal image registration. In *MICCAI*, pages 1115–1124, 1998.

[103] T. Rohlfing, C. R. M. Jr., D. A. Bluemke, and M. A. Jacobs. An alternating-constraints algorithm for volume-preserving non-rigid registration of contrast-enhanced mr breast images. In *WBIR*, pages 291–300, 2003.

[104] K. Rohr, M. Fornefett, and H. S. Stiehl. Spline-based elastic image registration: Integration of landmark errors and orientation attributes. *Computer Vision And Image Understanding*, 90(2):153–168, May 2003.

[105] K. Rohr, M. Fornefett, and H. S. Stiehl. Spline-based elastic image registration: integration of landmark errors and orientation attributes. *CVIU*, 90:153–168, 2003.

[106] K. Rohr, H. S. Stiehl, R. Sprengel, W. Beil, T. M. Buzug, J. Weese, and M. H. Kuhn. Point-based elastic registration of medical image data using approximating thin-plate splines. In *VBC*, pages 297–306, 1996.

[107] C. Rossl, F. Zeilfelder, G. Nurnberger, and H.-P. Seidel. Visualization of volume data with quadratic super splines. *Visualization IEEE*, 19-24:393–400, 2003.

[108] D. Rueckert, L. Sonoda, C. Hayes, D. L. Hill, M. O. Leach, and D. J. Hawkes. Non-rigid registration using free-form deformations: Application to breast mr images. *IEEE Trans. Med. Imaging*, 18(8):712–721, 1999.

[109] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes. Nonrigid registration using free-form deformation: application to breast mr images. *IEEE Trans. on Med. Imaging*, 18(8):712–721, 1999.

[110] M. Sallam. *Image unwarping and difference analysis: a technique for detecting abnormalities in mammograms*. PhD thesis, University of South Florida, USA, 1997.

[111] M. Y. Sallam and K. W. Bowyer. Registration and difference analysis of corresponding mammogram images. *Medical Image Analysis*, 3(2):103–118, 1999.

[112] B. Schmitt, A. Pasko, and C. Schlick. Constructive modeling of FRep solids using spline volumes. In *Proceedings of the 6th ACM symposium on solid modeling and applications*, pages 321–322, 2001.

[113] I. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions, part a: On the problem of smoothing or graduation, a first class of analytic approximation formulas. *Quart. Appl. Math.*, 4:45–99, 1946.

[114] I. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions, part b: On the problem of osculatory interpolation, a second class of analytic approximation formulae. *Quart. Appl. Math.*, 4:112–141, 1946.

[115] T. Sederberg and S. Parry. Free-form deformation of solid geometric models. In *ACM SIGGRAPH '86*, volume 20, pages 151–160, 1986.

[116] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche. T-spline simplification and local refinement. *ACM Trans. Graph.*, 23(3):276–283, 2004.

[117] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCs. *ACM Trans. Graph.*, 22(3):477–484, 2003.

[118] H.-P. Seidel. Symmetric recursive algorithms for surfaces: B-patches and the de Boor algorithm for polynomials over triangles. *Constr. Approx.*, 7:257–279, 1991.

[119] J. R. Shewchuk. Triangle: engineering a 2d quality mesh generator and delaunay triangulator. In *Proc. of 1st Worksh. Applied Computational Geometry*, pages 124–132, May 1996.

[120] J. R. Shewchuk. Automated three-dimensional registration of magnetic resonance and positron emission tomography brain images by multiresolution optimization of voxel similarity measures. *Med. Phys.*, 24:25–35, 1997.

[121] X. Shi, S. Wang, W. Wang, and W. R.H. The C^1 quadratic spline space on triangulations. Technical Report Report 86004, Department of Mathematics, Jilin University, 1996.

[122] H. Si. Tetgen: A quality tetrahedral mesh generator and three-dimensional delaunay triangulator, 2005.

[123] W. Song and X. Yang. Free-form deformation with weighted t-spline. *Vis. Comput.*, 21(3):139–151, 2005.

[124] G. Soza, M. Bauer, P. Hastreiter, C. Nimsky, and G. Greiner. Nonrigid registration with use of hardware-based 3d bezier functions. In *Conf. MICCAI*, pages 549–556.

[125] H. Speleers, P. Dierckx, and S. Vandewalle. Numerical solution of partial differential equations with powellcsabin splinesnext term. In *Journal of Computational and Applied Mathematics*, volume 189, pages 643–659, 2006.

[126] C. Tanner, J. A. Schnabel, D. Chung, M. J. Clarkson, D. Rueckert, D. L. G. Hill, and D. J. Hawkes. Volume and shape preservation of enhancing lesions when applying non-rigid registration to a time series of contrast enhancing mr breast images. In *MICCAI*, pages 327–337, 2000.

[127] M. Tarini, K. Hormann, P. Cignoni, and C. Montani. Polycube-maps. *ACM Trans. Graph.*, 23(3):853–860, 2004.

[128] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. In *Proc. SIGGRAPH'87*, pages 205–214, 1987.

[129] W. Tiller. Rational *b*-splines for curve and surface representation. In *IEEE Computer Graphics and Applications*, volume 3, pages 61–69, 1983.

[130] E. Vanraes, P. Dierckx, and A. Bultheel. On the choice of the PS-triangles. Report TW 353, Department of Computer Science, K.U.Leuven, 2003.

[131] E. Vanraes, J. Windmolders, A. Bultheel, and P. Dierckx. Automatic construction of control triangles for subdivided Powell-Sabin splines. *Computer Aided Geometric Design*, 21(7):671–682, 2004.

[132] B. C. Vemuri, S. Huang, S. Sahni, C. M. Leonard, C. Mohr, R. Gilmore, and J. Fitzsimmons. An efficient motion estimator with application to medical image registration. *Medical Image Analysis*, 2(1):79–98, 1998.

[133] K. J. Versprille. *Computer-Aided Design Applications of the Rational B-Spline Approximation Form*. PhD thesis, Syracuse University, 1975.

[134] P. A. Viola. *Alignment by maximization of mutual information*. PhD thesis, Massachusetts Institute of Technology, Boston, MA, USA, 1995.

[135] H. Wang, Y. He, X. Li, X. Gu, and H. Qin. Polycube splines. *Computer Aided Design*, 40(6):721–733, 2008.

[136] H. Wang, M. Jin, Y. He, X. Gu, and H. Qin. User-controllable polycube map for manifold spline construction. In *Proc. ACM Symp. on Solid and Physical Modeling*, pages 397–404, 2008.

[137] J. Wang and T. Jiang. Nonrigid registration of brain mri using nurbs. In *Pattern Recognition Letters*, volume 28, pages 214–223, 2007.

[138] K. Wang, Y. He, X. Guo, and H. Qin. Spline thin-shell simulation of manifold surfaces. In *Computer Graphics International*, pages 570–577, 2006.

[139] K. Wang, Y. He, and H. Qin. Incorporating rigid structures in non-rigid registration using triangular b-splines. In *VLSM*, pages 235–246, 2005.

[140] K. Wang, Y. He, H. Qin, P. R. Fisher, and W. Zhao. Temporal registration of 2d x-ray mammograms using triangular b-spline finite element method (tbfem). In *SPIE*, 2006.

[141] K. Wang, H. Qin, P. R. Fisher, and W. Zhao. Automatic registration of mammograms using texture-based anisotropic features. In *ISBI*, pages 864–867, 2006.

[142] W. Wang, H. Pottmann, and Y. Liu. Fitting b-spline curves to point clouds by curvature-based squared distance minimization. *ACM Trans. Graph.*, 25(2):214–238, 2006.

[143] Y. Wang, X. Gu, T. F. Chan, P. M. Thompson, and S. T. Yau. Volumetric harmonic brain mapping. In *ISBI ’04: IEEE International Symposium on Biomedical Imaging: Macro to Nano.*, pages 1275–1278, 2004.

[144] W. Welch and A. Witkin. Variational surface modeling. In *SIGGRAPH ’92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 157–166, 1992.

[145] K. Willemans and P. Dierckx. Smoothing scattered data with a monotone Powell-Sabin spline surface. *Numerical Algorithms*, 12:215–232, 1996.

[146] J. Windmolders and P. Dierckx. Subdivision of uniform Powell-Sabin splines. *Computer Aided Geometric Design*, 16:301–315, 1999.

[147] M. Wirth. *A nonrigid approach to medical image registration: matching images of the breast.* PhD thesis, RMIT University, Victora, Australia, June 1999.

[148] M. Wirth and C. Choi. Mammogram registration using the cauchy-navier spline. In *SPIE Medical Imaging*, volume 4322, pages 1654–1665, 2001.

[149] M. Wirth, J. Narhan, and D. Gray. Nonrigid mammogram registrationg using mutual information. In *SPIE Medical Imaging*, volume 4684, pages 562–573, 2002.

[150] Z. Xie and G. E. Farin. Image registration using hierarchical b-splines. *IEEE Trans. Vis. Comput. Graph.*, 10(1):85–94, 2004.

[151] H. Yang and B. Jüttler. 3d shape metamorphosis based on t-spline level sets. *The Visual Computer*, 23(12):1015–1025, 2007.

[152] H. Yang and B. Jüttler. Meshing non-uniformly sampled and incomplete data based on displaced t-spline level sets. In *Shape Modeling International*, pages 251–260, 2007.

[153] H. Yang and B. Jüttler. Evolution of t-spline level sets for meshing non-uniformly sampled and incomplete data. *The Visual Computer*, 24(6):435–448, 2008.

[154] P. Yang and X. Qian. A *b*-spline-based approach to heterogeneous objects desing and analysis. In *Computer Aided Design*, volume 39, pages 95–111, 2007.

[155] C. Y. Yao, H. K. Chu, T. Ju, and T. Y. Lee. Compatible quadrangulation by sketching. *Comput. Animat. Virtual Worlds*, 20(2-3):101–109, 2009.

[156] F.-F. Yin, M. L. Giger, K. Doi, C. J. Vyborny, and R. R. Schmidt. Computerized detection of masses in digital mammograms. *Med. Phy.*, 32(3):445–452, 1994.

[157] X. Zhou and J. Lu. Nurbs-based galerkin method and application to skeletal muscle modeling. In *SPM '05: Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pages 71–78, New York, NY, USA, 2005. ACM.

[158] B. Zitová and J. Flusser. Image registration methods: a survey. *Image Vision Comput.*, 21(11):977–1000, 2003.