

Non-isometric 3D Shape Registration

TAO JIANG

Transfer of Doctor of Philosophy



November 7, 2017

Copyright statement

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognize that its copyright rests with its author and due acknowledgment must always be made of any material contained in, or derived from, this thesis.

Publications

Jiang, T., Qian, K., Liu, S., Wang, J., Yang, X. and Zhang, J., 2017. Consistent as-similar-as-possible non-isometric surface registration. *The Visual Computer*, pp.1-11.

Qian, K., **Jiang, T.**, Wang, M., Yang, X. and Zhang, J., 2016. Energized soft tissue dissection in surgery simulation. *Computer Animation and Virtual Worlds*, 27(3-4), pp.280-289.

Wang, Z., Liu, S., Qian, R., **Jiang, T.**, Yang, X. and Zhang, J.J., 2016, November. Human motion data refinement unitizing structural sparsity and spatial-temporal information. In *Signal Processing (ICSP), 2016 IEEE 13th International Conference on* (pp. 975-982). IEEE.

Contents

Table of contents	iii
1 Introduction	1
1.1 Background	1
1.2 Main challenges	4
1.3 Research aims	5
1.4 Research Objectives	5
1.5 Contribution	6
1.6 Structure of the following chapters	7
2 Literature Review	8
2.1 3D geometric deformation	8
2.1.1 Basic concept	9
2.1.2 Linear deformation	10
2.1.3 Nonlinear deformation	13
2.1.4 3D geometric deformation summary	16
2.2 3D shape correspondence	17
2.2.1 Similarity-based correspondence	17
2.2.2 Rigid alignment	18
2.2.3 Non-rigid alignment	19
2.2.4 3D shape correspondence summary	20
2.3 3D shape registration	21
2.3.1 Rigid registration	21
2.3.2 Isometric registration	21
2.3.3 Non-isometric registration	22
2.3.4 3D shape registration Summary	24

3	3D Geometric Deformation	25
3.1	Notation and related methods	26
3.2	Consistent ASAP energy	28
3.3	Optimization	29
3.4	Experiments	32
3.5	Conclusion	33
4	3D shape registration	35
4.1	CASAP surface registration	36
4.1.1	Correspondence constraints	37
4.1.2	Feature point constraints	39
4.1.3	Optimization	40
4.2	Coarse-to-fine fitting strategy	41
4.2.1	Fitting Steps	41
4.2.2	Weights and parameters	43
4.3	Experiments and Results	43
4.4	Conclusion	47
5	Future Work	48
5.1	Element inversion remove	49
5.1.1	Locally injective mapping	49
5.1.2	Distortion measurement	51
5.1.3	Summary	51
5.2	Automatic feature selection for correspondence	52
5.3	Future Research Plan	52
	References	53

Chapter 1

Introduction

1.1 Background

3D shape registration has been a fundamental topic in computer graphics, computer vision, medical image processing and many other fields for many years. It has various applications such as geometric modeling, 3D shape acquisition, animation reconstruction and template tracking. With the development of 3D geometry acquisition technology, high-resolution and highly detailed 3D scans of objects in real-world can be obtained via current 3D scanning systems or depth cameras. However, these scanned data usually cannot be utilized directly without any further manual refinement in many real-world industries such as computer animation, films, computer games, manufacturing and medicine. Therefore, it is essential to create clean, complete and high-fidelity 3D directly usable shapes from scanning data. 3D shape registration as an essential technique to do so has been arousing intensive attentions. There are plenty of works dedicating to this purpose [Amberg et al. 2007; Papazov & Burschka 2011; Li et al. 2008; Yang et al. 2015; Yamazaki et al. 2013; Li et al. 2009; Huang et al. 2008].

Surface registration transforms multiple three-dimensional data sets into the same coordinate system so as to align overlapping components of these sets [Tam et al. 2013]. Given a template shape and a target

shape, the aim of 3D shape registration is to find a mapping between them that optimally transforms the template onto the target. According to the type of mapping (1.1), 3D shape registration is roughly divided into two categories: rigid and non-rigid. Rigid registration is to find a global rigid-body transformation that aligns two shapes. It is useful in nondeformable shape registration, but it cannot handle local transformation as it assumes that two shapes are only related by a rigid transformation. For deformable shape registration, non-rigid registration is required, which can be categorized into isometric and non-isometric. Isometric registration aims at finding a set of local rigid transformations. It is able to handle shapes undergoing isometric or quasi-isometric transformation like face expression or body movement, but it does not allow local scalability due to its length preserving property. Non-isometric registration introduce more degree of freedom in each local transformation, which can address wider range of local deformation between shapes. It can be classified into: equiareal, smooth and similar. Equiareal registration aims to preserve the area of each local cell of shapes. It has scale-preserving property but is unable to address size difference between the template and the target. In contrast, smooth registration minimizing smooth regularization energy or transformation between neighboring elements is suitable for size difference. However, it allows piecewise stretching transformation, which can result in shear distortion and losing template details. Similar registration fits the deformation gradient into a similarity matrix, which is an isotropic scale factor times a rotation matrix. The scale factor is able to handle size difference, while the rotation matrix part prevents local stretch and distortion. Thus, it has been widely used in works [Yamazaki et al. 2013; Yoshiyasu et al. 2014; Papazov & Burschka 2011] to align surfaces with different size and detail. However, the energies they adopt to constrain the local deformation similarity are not consistent. This may lead to fold over and self-intersection during transformation.

The process of shape registration can be approximately summarized in three steps: first, generally aligning the global positions, scales, orientations between shapes; second, searching the correspondences between

Mapping		df	Property
Rigid		\mathbf{I}	shape-preserving
Non-rigid	Isometric	\mathbf{R}	length-preserving
	Equiareal	$\det(df)=1$	scale-preserving
	Non-isometric	$\min \ df\ ^2$	smooth deformation
Similar		$s\mathbf{R}$	angle-preserving

Table 1.1: *Classes of mapping.* df is the deformation gradient, s is a scalar, \mathbf{I} is an identity matrix, \mathbf{R} is a rotation matrix.

shapes; finally, attract the template onto the target by deformation methods according to the correspondences. The initial positions, orientations, sizes and details, as well as their resolutions of data, can be largely different between shapes. Therefore, 3D shape registration methods which are capable of dealing with the difference mentioned before and robust to noise and outliers with lower chance of fold over occurrence are highly desirable. To attain these purposes, choosing appropriate constraints in each step is required. According to the survey [Tam et al. 2013], constrains for non-rigid registration can be categorized into markers, templates, deformation-induced constraints, features, saliency, regularization, envelopes of motion and search constraints. Since rigid and isometric registration are unable to deal with shapes with large size difference, which indeed exists in real-world circumstance, we mainly focus our research on non-isometric case. In terms of non-isometric registration, however, not all of the constrains listed above can be applied. Some constrains may be suitable for isometric case but not for non-isometric one. For example, features and signature constrains can be well defined under isometric circumstance. There are some isometry invariant features, such as heat kernel signature (HKS)[Sun et al. 2009] and wave kernel signature (WKS)[Aubry et al. 2011], which can be employed to seek for the correspondence as they are consistent under isometric deformation. However, due to the large variations in the shapes' pose, size, local scale and geometric detail in the case of non-isometric, it is difficult to define such a signature that are invariant to non-isometric transformation, which makes the correspondence searching become a more challenging problem for non-isometric registration. Consequently, it is necessary to design appropriate constraints in each concern of non-isometric shape

registration.

This research aims at tackling the challenges of non-isometric 3D shape registration including devising the underlying transformation regularization and looking for the correspondent pairs between the template and the target shapes. The potential application of the developed technology could have huge impact, for example, it will allow us to easily build virtual personalized 3D character for games and films and transfer necessary features of avatar such as face performance, motion data, textures and even internal structure to virtual characters.

1.2 Main challenges

The main challenges for non-isometric 3D shape registration includes: deformation regularization and correspondence matching.

Deformation regularization

- **Geometric consistency** Because there are often large variations in the orientation, size, shape and pose of two shapes, aligning the models globally while also capturing surface details at fine scale is difficult. Difference in initial positions, orientations and resolutions can also affect algorithm performance and convergence rate.
- **Mesh-connectivity preservation** During the registration of two shapes with dramatic difference in size and details, the template undergoes large deformation, which makes it susceptible to shear distortion, self-intersection and fold over. Obtaining a high-quality, clean and usable shape for application is very challenging.

Correspondence matching

- **Semantic consistency** Semantic consistency is crucial for 3D shape registration. For example, during face registration, features around eyes, mouths and noses should correspond to each other.

It is very challenge to achieve this goal in the case of non-isometric shape registration as these features are prone to variant under non-isometric deformation.

- **Less user efforts** In order to capture the details on the target, handle large deformation as well as maintain the semantic consistency, previous works require specifying dozens of landmarks from user input, which is not very efficient and prone to errors. The registration technique should not ask for a large amount of user efforts to specify many landmarks manually.

1.3 Research aims

The aim of this research is to solve these key technical challenges in non-isometric 3D shape registration with large variation in size, pose, detail between the template and the target. The major tasks range from deformation regularization, correspondence matching, 3D surface registration. This research will propose a semi-automated method for registering a template shape onto a largely deformed target with little user efforts, which could be applied in 3D character creation, 3D facial registration and dynamic fussion.

1.4 Research Objectives

In order to achieve the above mentioned aim, following objectives need to be accomplished:

- **Literature Review:** review and investigate current researches on 3D geometric deformation, correspondence matching and 3D surface registration. Identifying the limitation of current approaches in non-isometric situations.
- **Non-isometric 3D Geometric Deformation :** design a novel non-isometric 3D shape deformation technique that is able to handle large deformation with mesh-connectivity well preserved.

- **Non-isometric 3D Shape Registration:** equipped the above deformation technique as regularization, find appropriate correspondences between the template and the target, design a novel non-isometric registration method with little user efforts for shapes have large variation in size and detail.

1.5 Contribution

The main contributions of this research are summarized as follows:

- We propose a novel shape deformation method, called consistent as-similar-as-possible (CASAP) deformation. It fits local transformation into scale and rotation, which is not only able to handle large deformation, but also reduces the shear distortion. Moreover, contrast to as-similar-as-possible (ASAP) deformation method, the energy in our deformation technique is consistent, leading to a parametrization invariance behavior and converge to the continuous case as the mesh is refined.
- With CASAP energy as deformation regularization, we further propose a non-isometric surface registration approach. It not only produces more accurate fitting results with little user input, but also preserves angles of triangle meshes and allows local scales to change. Furthermore, a coarse-to-fine strategy is proposed to further improve the robustness and efficiency of our approach.
- Taking local geometrical feature descriptors into account, we propose a new matching energy to choose more reasonable correspondent pairs between template and target models.

1.6 Structure of the following chapters

- **Chapter 2** Literature review on the related research topics, including 3D geometric deformation, 3D shape correspondence and 3D shape registration.
- **Chapter 3** Introduce a new algorithm which is able to deal with dramatic local deformation as well as preserve mesh-connectivity in order to reduce the occurrence of fold over and shear distortion during transformation.
- **Chapter 4** Design and develop a novel correspondence matching method between non-isometric shapes, present a semi-automatic registration method with little user input based on coarse-to-fine strategy for shapes undergoing large deformation.
- **Chapter 5** Future plan.

Chapter 2

Literature Review

The registration process usually consists of three steps: aligning the template globally; finding correspondence between shapes; attract the template onto the target via deformation technique. This chapter will review the key technologies for shape registration including 3D geometric deformation, 3D shape correspondence and 3D shape registration.

2.1 3D geometric deformation

Geometric deformation is a fundamental and deeply researched topic in the field of computation geometry. It has important application in a lot of areas, for example computer animation, film industry, computer game and manufacturing. A multitude of works have been published aiming at this issue. According to the survey Botsch & Sorkine [2008], the deformation methods can be generally divided into two categories:

- Linear deformation;
- Nonlinear deformation.

2.1.1 Basic concept

Before diving into the review, we first introduce some basic concepts and notations of deformation which will be referred in this research. As we all know, the objects existing in nature is in the form of continuity. However, in computer they have to be represented discretely as computers can only address discrete signals. In computer, the shapes (both surface and volume) are usually discretized into discrete units. We name these discrete units as discrete cells. In general, the surface can be discretized into triangles or quadrilaterals, while the basic 3D discrete cells for volume are the tetrahedron, quadrilateral pyramid, triangular prism and hexahedron. In this research, we mainly focus on triangulation surface. Although shapes represented in computer are in discrete forms, the deformation energy should converge to the continuous case as the discretization refined. We call this feature as consist property as it should exhibit reasonable consistency of shape deformation results for different discretizations, which is particulary useful in the case of a poor triangulation that has large variation in the size of the triangles.

For triangulation mesh, the global deformation can be decomposed into each triangle or each vertex. Moreover, these triangles or these vertices are not independent from each other, they are connected by neighboring edge sets. There are three kinds of neighboring edge sets for triangulation mesh: triangle, spokes, spokes and rims. It is quite important to choose appropriate edge sets as it will affect whether the energy is consistent as shown in the next subsection.

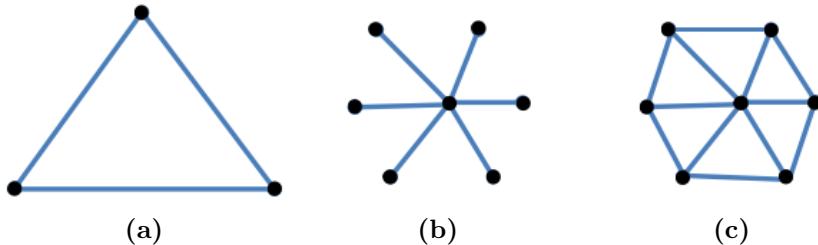


Figure 2.1: Different edge sets for triangulation mesh: (a) triangle; (b) spokes; (c) spokes-and-rims;

2.1.2 Linear deformation

Linear means we use a linear system of equations to formulate a global quadratic variational minimization problem, which is the main ingredient of the deformation algorithm. The advantage of linear deformation methods are three-folds: efficient, solving a linear system of equation is quite efficient, especially when the associated linear system is sparse; robust, the solved quadratic energy has a unique global minimum; smooth, the global energy minimization guarantees smooth and \mathcal{C}^1 continuous surface deformations. According to the target the deformation applied onto, the linear deformation methods can be categorized into linear surface based deformation and linear space deformation.

Linear surface-based deformation

The linear surface-based deformation roughly falls into three categories: shell-based deformation, multi-scale deformation and differential coordinates based deformation.

Shell-based deformation method minimizes the elastic energy subject to user-defined boundary constraints. The elastic energy measures how much the object has been deformed from its initial configuration, which is the main requirement for physically based surface deformations. For two-manifold surfaces, the elastic energy considers local stretching and bending within the object. Terzopoulos et al. [1987]; Celniker & Gossard [1991] propose physically-based deformation methods minimizing stretching and bending under deformation constrains, which corresponds to thin-shell models of non-planar rest states. Deformation based on a discretization of variational bending energy minimization is mathematically understood and yields smooth and tangent-continuous deformations [Guskov et al. 1999; Kobbelt et al. 1998; Botsch & Kobbelt 2004; Bickel et al. 2008]. However, for these approaches all computations and linearizations are performed with respect to a fixed reference mesh, large deformations might lead to shape distortions and detail loss.

To preserve surface details, the above methods require a multi-scale

decomposition, which splits a surface into a smooth base surface (low frequencies) and displacement vectors (high frequencies). Multiresolution deformation changes the smooth base surface and adding the details back onto it then yields the desired multi-scale deformation [Kobbelt et al. 1999]. In particular, Kobbelt et al. [1998] introduces a mesh deformation technique by solving a constrained minimization of the thin-plate energy at a desirable coarse resolution. The user specifies deformation constraints through a handle polygon. Original mesh details are added back to the resulting smooth mesh to produce a final solution. This technique only gives the user limited control over the mesh shape through sparse constraints on the handle polygon. The rest of the mesh geometry is uniquely determined by the minimization. Displacement volumes [Botsch & Kobbelt 2003] encode the high frequencies by prism elements enclosed between the original and the base surface, which avoids detail distortion, but comes at the considerably higher cost of a non-linear detail reconstruction. Although both representation (displacement vectors/volumes) can be combined with any underlying deformation technique, the required multiscale decomposition can become quite difficult for geometrically or topologically complex models.

To avoid the multi-scale decomposition, other methods modify differential surface properties instead of its spatial coordinates, and then solve a linear Poisson system for a deformed surface with the desired differential coordinates [Lipman et al. 2004; Sorkine et al. 2004; Yu et al. 2004; Zayer et al. 2005; Lipman et al. 2005]. The methods of Yu et al. [2004]; Zayer et al. [2005] use gradients of affine deformations, i.e., their rotation and scale/shear components, for transforming surface gradients. As a consequence, these methods work well for rotations, but are insensitive to translations: Adding a translation to a given deformation does not change its gradient, and thus has no influence on the resulting surface gradients. But since even pure translations induce local rotations of tangent planes, these methods are counter-intuitive for modifications containing large translations. In contrast, the shape editing approach of Sorkine et al. [2004] aims to preserve the differential coordinates or Laplacian coordinates. It implicitly solves for local rotations of ver-

tex neighborhoods, but due to linearizations their method has problems with large rotations, as was shown in their follow-up paper [Lipman et al. 2005]. In that paper, Lipman et al. minimize bending by preserving relative per-vertex orientations. They solve a linear system for per-vertex orientations, and reconstruct vertex positions. Since the first system does not consider position constraints, their technique also neglects the connection between translations and rotations, it exhibits the same translation-insensitivity as gradient-based methods.

Linear space deformation

The need for low-dimensional control of deformation fields was identified early in computer graphics. Among the first approaches was Free-Form Deformation [Sederberg & Parry 1986], which relied on regular lattices to specify spatial deformations. It parameterizes a space deformation with a 3D lattice and provides an efficient way to apply coarse deformations to complex shapes. However, achieving a fine-scale deformation may require a detailed, hand-designed control lattice [Coquillart 1990; MacCracken & Joy 1996] and an large amount of user manipulation. Although more intuitive control can be provided through direct manipulation [Hsu et al. 1992], the user is still restricted by the expressibility of the FFD algorithm. With their Wires concept, Singh & Fiume [1998] present a flexible and effective space deformation algorithm motivated by armatures used in traditional sculpting. A collection of space curves tracks deformable features of an object, providing a coarse approximation to the shape and a means to deform it. Singh & Kokkevis [2000] generalize this concept to a polygon-based deformer. Botsch & Kobbelt [2005] use triharmonic radial basis functions for real-time freeform shape editing. An incremental least-squares method is introduced to approximately solve the involved linear systems in a robust and efficient manner.

Cage-based deformation methods were an important step forward, because control polytopes offer much better adaptability to the input shapes. The underlying theme of many cage-based methods is to generalize barycentric coordinates from simplices to general polytopes. Mean

value coordinates (MVC) for closed polyhedrons [Ju et al. 2005] offer many desirable properties and can be calculated using closed-form expressions, but are not fully shape-aware. This shortcoming has been addressed by harmonic coordinates [Joshi et al. 2007]. Lipman et al. [2007] introduce positive mean value coordinates (PMVC). Unlike the MVC, the modified coordinates are unconditionally positive, and require only a local computation. The methods mentioned above are affine-invariant and not shape-preserving. Lipman et al. [2008] introduce green coordinates for closed polyhedral cages. It does not only depend on vertex-based basis, but also on the cage faces, which leads to space deformations with a shape-preserving property. While many new intriguing coordinates and their underlying mathematical properties have been studied in recent years [Hormann & Sukumar 2008; Weber et al. 2011; Li & Hu 2013], the problem common to all cage-based method remains: the design of control cages requires experience with polygonal modeling: the cage should be close to the shape and have enough density to represent the shape.

2.1.3 Nonlinear deformation

The surface deformation problem is inherently non-linear, it requires deducing local rotations of the surface based on position displacements. Therefore, a linear method can only provide an approximate result, or a compromise must be made in terms of the problem setup, e.g., requiring more complex interactive input from the user, for example adding more user handle constraints or increasing the cage density. The inherent limitations of linear methods motivated us to investigate non-linear deformation techniques. Nonlinear deformation can also be classified into nonlinear surface-based deformation and nonlinear space deformation.

Nonlinear surface-based deformation

Pyramid coordinates [Kraevoy & Sheffer 2004, 2005] can be considered as the nonlinear versions of Laplacian coordinates, leading to differential

coordinates invariant under rigid motions, which can be used for deformation as well as for morphing. Huang et al. [2006] employ a nonlinear version of the volumetric graph Laplacian, which also features nonlinear volume preservation constraints. In order to increase the performance and efficiency of the optimization, they use a subspace approach: the original mesh is embedded in a coarse control cage, and the optimization is performed on the cage vertices while considering the constraints from the original mesh vertices in a least-squares manner. An alternative approach to subspace methods is the handle-aware isoline technique of [Au et al. 2007]. In a preprocessing step one constructs a set of iso-lines of the geodesic distance from either the fixed regions or the handle regions. For each of these iso-lines, a local transformation for a Laplacian-based deformation is found by a nonlinear optimization. The number of required iso-lines is relatively small, which guarantees an efficient numerical optimization and thereby allows for interactive editing. Shi et al. [2007] combine Laplacian-based deformation with skeleton-based inverse kinematics, which allows for easy and intuitive character posing, featuring control of lengths, rigidity and joint limits, but it in turn requires a complex cascading optimization for the involved nonlinear energy minimization. Botsch et al. [2006] propose a nonlinear version of the shell-based minimization of bending and stretching energies. The surface is modeled as a thin layer of triangular prisms, which are coupled by a nonlinear elastic energy. During deformation the prisms are kept rigid, which allows for a robust geometric optimization. A hierarchical optimization is used to increases the computational efficiency.

Sorkine & Alexa [2007] propose a surface deformation based on the as-rigid-as-possible (ARAP) energy. The ARAP energy measures the local deviation of the differential of a mapping between two shapes from rigidity. The neighboring edge sets adopted in this work are spokes, which requires a positive weighting scheme to guarantee the correct minimization of the energy. Chao et al. [2010] take into account all the opposite edges in the triangles incident to a vertex, the neighboring edge sets in their work are spokes and rims, which guarantee correct minimization of the energy even if the weights are negative. However, the discretization

of [Chao et al. 2010] is only consistent for volumetric case with tetrahedron cells in 3D or parameterization with triangle edge sets in 2D, it is not consistent for the surface case using spokes and rims edge sets in 3D. In order to come up with a consistent discretization for surface in 3D, Levi & Gotsman [2015] introduce a new ARAP-type energy, named SR-ARAP (ARAP with smooth rotations), they add a bending term in the ARAP energy to enable the discretization consistent, which achieves results with quality that competes with the volume deformation.

Nonlinear space deformation

Sumner et al. [2007] compute detail-preserving space deformations by formulating an energy functional that explicitly penalizes deviation from local rigidity by optimizing the local deformation gradients to be rotations. In addition to static geometries, their method can also be applied to hand-crafted animations and precomputed simulations. Botsch et al. [2007] extend the PriMo framework [Botsch et al. 2006] to deformations of solid objects. The input model is voxelized in an adaptive manner, and the resulting hexahedral cells are kept rigid under deformations to ensure numerical robustness. The deformation is governed by a nonlinear elastic energy coupling neighboring rigid cells. Another class of approaches uses divergence-free vector fields (the divergence of the vector fields is zero) to deform shapes [Angelidis et al. 2006; von Funck et al. 2006]. The advantage of those techniques is that by construction they yield volume-preserving and intersection-free deformations. As a drawback, it is harder to construct vector fields that exactly satisfy user-defined deformation constraints.

Müller et al. [2005] present meshless deformations based on shape matching. Instead of using the vertex connection information, they divide the set of vertices into clusters. Afterwards, affine transformations can be applied on each clusters. Based on this work, Rivers & James [2007] introduce fast lattice shape matching for real-time deformation. Since the original method can be very slow for stiff models - per-vertex costs scale cubically with region width - they exploit the inherent sum-

mation redundancy of shape matching and provide large-region matching at constant per-vertex cost. With this approach, large lattices can be simulated in linear time complexity.

Jacobson [2013] introduces the harmonic, biharmonic, triharmonic equations w.r.t. surface displacement fields correspond to minimizers of the Dirichlet, Laplacian, Laplacian gradient energies. They offer a detail derivation to obtain a linear system to solve this second-order elliptic partial differential equation. Weber et al. [2007] uses harmonic functions that are provably monotonic and bounded but have only C^0 smoothness near constrained boundary. Bounded biharmonic weights are proposed in Jacobson et al. [2011] to minimize the Laplacian energy subject to bound constraints. They spreads the influences of the controls in a shape-aware and localized manner, even for objects with complex and concave boundaries. Jacobson et al. [2010] use mixed finite elements to provide a discretization for biharmonic and triharmonic equations on meshes. They factorize the original equations into low-order equations by introducing additional variables, and introduce lumping mass matrices to eliminate unneeded variables. Their minimizers exhibit increasing orders of continuity but lose the maximum principle and show oscillations. Jacobson et al. [2012] propose smooth shape-aware functions with controlled extrema. They provide a framework for minimizing quadratic energies on manifolds while constraining the solution to obey the maximum principle in the solved region.

2.1.4 3D geometric deformation summary

3D geometric deformation is a crucial part of shape registration as it serves as a regularization to attract the template onto the target. This section investigates the state of the art linear and nonlinear deformation techniques, and compares them in terms of handling large deformation and consistency. The linear methods can only approximate the local rotations, which are commonly seen in surface deformation. Nonlinear approaches are able to handle local rotations. However, some of them do not allow local scales, which are required in addressing large local

deformation. Although some methods allow local scale but they are not consistent, which prone to fold-over and self-intersection, especially in the case of poor triangulation. In this research, a novel deformation algorithm will be proposed to address large difference in size, as well as to guarantee the energy consistent to minimize the occurrence of fold-over and self-intersect.

2.2 3D shape correspondence

Establishing a meaningful shape correspondence is a fundamental task in applications such as object recognition, statistical shape modeling, shape morphing and deformation transfer. In this section, we will give an overview of the correspondence research from three point of views: similarity-based correspondence, rigid alignment and non-rigid alignment.

2.2.1 Similarity-based correspondence

One of the most fundamental ways of computing correspondence is to estimate the similarity between pairs of shape elements or feature points collected from the shapes and derive a correspondence from those estimates, which is sometimes called the feature matching approach. The elements are commonly characterized by shape descriptors. There are plenty of shape descriptors can be computed for each shape primitive and used in conjunction with correspondence algorithms for 2D surfaces or 3D point sets (Table 2.1).

A correspondence is obtained by selecting assignments between pairs of elements while optimizing an objective function composed of two terms. The first term seeks to maximize the similarity between the descriptors of corresponding elements, while the second term seeks to minimize the distortion that would be introduced in the shapes if they were deformed to align their corresponding elements. However, the second term can be estimated without explicitly aligning the shapes. Ideally,

Descriptor	Type of dataset
Shape context	Point sets
Spin images	Oriented points
Multi-scale features	Oriented points
Curvature maps	Surfaces
Integral invariants	Surfaces
Spherical harmonics and wavelets	Surfaces
Salient geometric features	Surfaces
Part-aware metric	Surfaces
Heat Kernel Signature	Surfaces

Table 2.1: *Shape descriptors can be used for shape correspondence.*

satisfying these objectives should translate into a solution that is geometrically or semantically meaningful. Such a solution is typically obtained with a standard optimization method (e.g., quadratic programming).

Feature matching can be applied in any context where it is possible to compute a set of descriptors for the elements. Example applications include registration of 3D scans [Castellani et al. 2008] and deforming surfaces [Anguelov et al. 2005], or skeleton matching [Biasotti et al. 2006]. Moreover, this approach is not restricted to its own domain and can be combined with alignment-based approaches to provide a proper initialization to these methods [Rusinkiewicz & Levoy 2001], or to restrict the size of the solution space [Gelfand et al. 2005; Kin-Chung Au et al. 2010; Chang & Zwicker 2008; Aiger et al. 2008].

2.2.2 Rigid alignment

Rigid alignment is to find a global geometric transformation that aligns the shapes. One example application is the rigid alignment of geometry scans used for shape acquisition. The goal here is to capture a real-world static 3D shape and obtain its digital representation. However, it may not be possible to capture the entire object in a single scanning pass due to self-occlusions and physical constraints of the scanner, so it might become necessary to acquire multiple scans and optimally align them to reconstruct the full object [Turk & Levoy 1994; Rusinkiewicz & Levoy 2001; Gelfand et al. 2005; Aiger et al. 2008]. The key characteristic

of the rigid alignment problem is that the objects do not change from one scanning pass to another. Thus, it is assumed that each scan can be transformed with a single rigid transformation in order to align it perfectly with the other scans. Rigid transformations comprise mainly translations and rotations, and one of their important characteristics is that they reside in a low-dimensional space.

Scan alignment is just one example of many applications that rely on the assumption of rigidity in the datasets. If the input shapes are given as sets of 3D points, the problem of rigid alignment can be posed as: for each point set, find the rigid transformation that maximizes the number of points in the set that align to points in the other sets. This goal is usually dependent on a threshold that indicates when two points are close enough and can be considered as matching to each other [Irani & Raghavan 1999]. Since finding the best aligning transformation might be a complex task, the feature matching approach can be brought in to aid in the search for the optimal alignment.

2.2.3 Non-rigid alignment

Sometimes it might be necessary to lift the assumption that each scan can be perfectly aligned with a rigid transformation, e.g., when large amounts of noise are present in the scans. More significant examples of datasets that cannot be perfectly aligned with a rigid transformation include the correspondence of articulated shapes [Elad & Kimmel 2003; Anguelov et al. 2005; Jain et al. 2007; Chang & Zwicker 2008; Huang et al. 2008], where certain parts of the shapes can be bended independently, the correspondence of anatomical shapes (e.g, organs) [Audette et al. 2000], which can deform in an elastic manner and introduce stretching to localized portions of the shape, and finally the correspondence between shapes with different geometries but that represent a same class of objects or which have parts that are semantically related [Allen et al. 2003; Zhang et al. 2008]. In the latter case, the alignment problem can be concluded as establishing a correspondence between shapes that can differ in both local stretching and bending.

To this end, it becomes necessary to add more degrees of freedom (DOF) to how the shapes can be brought into correspondence. This can be achieved by generalizing two aspects of the problem. First, non-rigid (possibly non-linear) transformations can be taken into consideration, e.g., thin-plate splines [Chui & Rangarajan 2003]. Secondly, these transformations can be applied separately to local portions of the shape. For example, the transformation applied to a shape can be represented as a set of displacement vectors (one per shape vertex) [Pauly et al. 2005]. Then, finding the best transformation amounts to computing the displacements that bring each vertex in correspondence with the target shape. The distinction to the rigid case is that the DOF of geometric transformations being considered is now inherently high-dimensional.

2.2.4 3D shape correspondence summary

In the process of shape registration, finding correspondence is the prior step before deforming the template, it determines the goal positions of the deformed template. This section reviews the methods solving rigid and non-rigid correspondence problems. Rigid alignment is only effective with static shapes. For deformable shapes, similarity-based correspondence methods work well between isometric shapes, as the descriptors it adopted can be isometrically invariant such as Heat Kernel Signature. However, there is no such shape descriptor which is invariant to large non-isometric deformation. Non-rigid alignment assumes the template can be transformed onto the target in a specified type of deformation, however, the space of solution to large deformation itself is quite high dimensional and hard to find. In this research, we design a novel matching energy to choose correspondence between non-isometric shapes, the correspondences are first chosen based on local geometric feature descriptors and then extended to local optimal candidates.

2.3 3D shape registration

3D registration is an active research topic in computer graphics and computer vision. Given a template and a target, the goal of 3D registration is to find a mapping between them in order to optimally transforms the template onto the target. Shape registration methods can fall roughly into 3 classes: rigid, isometric, non-isometric.

2.3.1 Rigid registration

The purpose of rigid registration is to find a global rigid transformation that aligns two shapes. It is quite similar to rigid alignment in last subsection, just has a different final goal. The most classic algorithms for rigid registration are Iterative Close Point (ICP) Besl & McKay [1992] and its variants Rusinkiewicz & Levoy [2001]. ICP alternates between searching correspondences by choosing closest points on the target w.r.t. the points on the template and computing a rigid transformation to response the correspondences. The results closely dependent on the initial position and orientation of the shapes, large distance between them tend to produce unstable correspondence and ICP is susceptible to converge to local minimum. Aiming to solve this problem, Gelfand et al. [2005] initially approximate the optimal transformation via feature matches, this approximation can be fed into ICP to improve the converge accurate and rate. Another shortage of ICP is that it is sensitive to noise and outliers which often appear in 3D scans. To tackle this issue, Bouaziz et al. [2013] put forward a new variant of ICP algorithm with sparsity-inducing norms, successfully eliminating the affect of the noise and outlier and obtaining superior results.

2.3.2 Isometric registration

Isometric registration is to solve a local rigid transformation for each vertex on the template in order to align the template onto the target. Li et al. [2008] propose a isometric registration algorithm for partial scans

of deforming shapes within a single non-linear optimization. The rigid energy term they adopted penalizes the deviation of each local transformation from a pure rigid motion, which is only composed of rigid and orientation transformation. Qixing et al. [2008] regard isometric registration as an optimization problem. To solve this problem, they alternate between correspondence searching and deformation calculation. Under the assumption of local rigid deformation, robust correspondence are obtained via a pruning mechanism based upon consistency in geodesic distance. Chang & Zwicker [2008] convert the isometric registration problem into a discrete labeling problem. They first sample the motion explicitly, which provides a priori set of possible rigid transformation between articulate parts of shapes, then finding an optimal assignment of transformation to each part of shapes. Tevs et al. [2009] propose a global isometric matching approach based on a novel RANSAC-like randomized sampling algorithm. This method is robust to topological noise and is able to output matching alternatives by sampling the space of plausible solutions. Maron et al. [2016] approximate the global minimum of the procrustes matching problem with convex relaxation for non-rigid isometric and near-isometric shape matching problem. The advantage of isometric registration is that they can achieve automatically, but they are incapable of handling models with different sizes.

2.3.3 Non-isometric registration

Non-isometric registration is a wide range of topics including all kinds of registration are not isometric. Here we only focus on techniques related to our work: similar registration, affine registration and smooth registration.

Similar registration

Similar registration is to preserve the local similarity of shapes, it remains the angle of intersection of every pair of the intersecting arcs unchanged during deformation process. Sorkine et al. [2004] offer a linear approx-

imation of similarity matrix to make deformed Laplacian coordinates consistent. However, this method only works well under small rotations as the approximation removes the quadratic term. Thus it cannot handle large rotation. Yamazaki et al. [2013] extend ARAP energy to *as-similar-as-possible* (ASAP) energy with spokes edge sets. The work in [Papazov & Burschka 2011] is a variation of shape matching [Müller et al. 2005] called similarity shape matching. Although these techniques utilize similar mapping to enable them to address size difference and shear distortion, they do not consider the smoothness regularization, which results in they are incapable of handling large changes in pose or shape. Yoshiyasu et al. [2014] incorporate smooth regularization into the total energy, however, it is a unweighted energy, which does not take into account the impact of the length of edges. Moreover, the edge sets they adopt is spokes, which leads to an inconsistent energy.

Affine registration

Affine registration allows a affine transformation for each template vertex, which will have a larger solution space due to more freedom the affine matrix has. Amberg et al. [2007] employ a locally affine regularisation to each vertex and smooth the difference in the transformation of neighbouring vertices. The method is robust to incomplete data and is able to handle a wide range of initial conditions. Yang et al. [2015] propose a sparse non-rigid registration method with an l_1 -norm regularized model for transformation estimation, they allow an affine transformation for each point in the template to cover a wide range of non-rigid deformations. Affine registration is able to handle shapes in different size, but the affine matrix with more DOF may also introduce shear and distortion, which easily lead to fold-over and intersection.

Smooth registration

Smooth registration is based on harmonic mapping or its varieties or other smoothness regularization. Weise et al. [2009] provide a complete

integrated system for live facial puppetry which not only is able to track high-resolution facial expression in real time but also can transfer the expression to another face. They minimize a membrane energy on the displacement vectors to smooth the deformation on the template. Yeh et al. [2011] present a novel approach for template-based 3D model fitting , they approximate the input geometry with a linearized biharmonic surface, and then use iterative Laplacian editing and a local surface flattening mechanism to avoid foldovers. There are some works [Allen et al. 2003; Amberg et al. 2007] based on other smoothness regularization whose purpose are to make the deformation between neighbors as smooth as possible, this idea is similar to the bending term added in Levi & Gotsman [2015]. Similar to affine registration, smooth registration still has the same advantage and problem. In addition, it requires many landmarks from user input to obtain a good initial shape.

2.3.4 3D shape registration Summary

In summary, 3D shape registration with large size and detail difference is challenging problem mainly because of the requirement of large deformation which is susceptive to fold-over and intersection. This section investigate all kinds of registration methods and their capabilities. Rigid registration is only suitable for static models. Isometric registration can work with deformable objects but only limit to isometric shapes or near isometric shapes. Affine and smooth registration are able to handle models with different sizes, but they are too weak to against shear distortion. Although similar registration can overcome these disadvantages, the energy itself is not consistent, which easily leading to fold-over and self-intersection when the triangulation mesh is poor. In this research, a novel registration technique will be present to solve large shape variation as well as minimize the change of fold-over occurrence.

Chapter 3

3D Geometric Deformation

3D geometric deformation plays a key role in computer graphics and computer vision, and is the cornerstone of computer animation used in computer games and films. Large deformation is a challenging problem as it tends to produce stretch and shear on the shape, meanwhile, appropriate adjustment on local scale is required to adapt to the size difference. The mesh-connectivity on the deformable shape is supposed to be maintained as the users interactively editing the shape, but in the context of large deformation, they are susceptible to change as the distortion and scale may be easily brought into.

In the last two decades, there have been plenty of methods proposed on surface editing research aiming to solve these problems. However, these approaches either are unable to handle large deformation or suffer from fold-over due to their inconsistent energy. In this chapter, we will propose a novel deformation algorithm which not only is capable of addressing shapes with large difference in size but also ascertain a consistent energy to reduce the chance of fold-over and self-intersection. In this chapter we will introduce our consistent as-similar-as-possible (CASAP) deformation technique that not only is able to handle large deformation but also well preserve the mesh-connectivity.

3.1 Notation and related methods

Before stepping into the problem, we first introduce some basic concepts and notations of deformation which will be used in this research. We denote by \mathcal{S} a triangle mesh. The piecewise linear geometric embedding of \mathcal{S} is defined by the vertex positions $\mathbf{p} \in \mathbf{R}^3$. Assume \mathcal{S} is being deformed into \mathcal{S}' that has the same connectivity and a different geometric embedding \mathbf{p}' . The discrete cell corresponding to vertex i is denoted by \mathcal{C}_i and its deformed version \mathcal{C}'_i . \mathcal{E}_i is the corresponding edge sets in the cell \mathcal{C}_i .

Sorkine & Alexa [2007] propose an as-rigid-as-possible energy to measure the local rigidity variation, in which the neighboring edge sets adopted are spokes. Given two meshes \mathcal{S} and \mathcal{S}' consisting of vertices \mathbf{p} and \mathbf{p}' respectively, the discrete ARAP energy between these two meshes is defined as:

$$E_{ARAP}(\mathcal{S}, \mathcal{S}') = \sum_i \sum_{(j,k) \in \mathcal{E}_i} w_{jk} \|(\mathbf{p}'_j - \mathbf{p}'_k) - \mathbf{R}_i(\mathbf{p}_j - \mathbf{p}_k)\|^2, \quad (3.1)$$

where w_{jk} are weighting coefficients, $\mathbf{R}_i \in RO(3)$ are optimal local rotation matrices. In the shape deformation setup, deforming a mesh \mathcal{S} involves fixing handle points and solving for the rest of the \mathbf{p}' by minimizing (3.1). The intuition for minimizing the ARAP energy is to find a mesh \mathcal{S}' that is locally a rigid transformation of the source mesh \mathcal{S} . More specifically, the differential of a mapping from an edge set in \mathcal{S} to a corresponding edge set in \mathcal{S}' should be optimally a rotation, thus synchronizing corresponding edge vectors of the two edge sets.

Although ARAP shape deformation gained popularity, the energy is not consistent as it lacks a term measuring bending energy. Levi & Gotsman [2015] introduce a new ARAP-type energy, named SR-ARAP (ARAP with smooth rotations). The new energy used for surface deformation is consistent and can produce results with quality that competes with the volume deformation. They add a smoothing term to the ARAP surface energy to penalize the difference between the map differential of an edge

set and the map differential of its neighboring edge sets. The SR-ARAP energy is defined as:

$$E_{SR-ARAP}(\mathcal{S}, \mathcal{S}') = \sum_i \sum_{(j,k) \in \mathcal{E}_i} w_{jk} \|(\mathbf{p}'_j - \mathbf{p}'_k) - \mathbf{R}_i(\mathbf{p}_j - \mathbf{p}_k)\|^2 + \alpha A \sum_{\mathcal{E}_l \in \mathcal{N}(\mathcal{E}_i)} w_{il} \|\mathbf{R}_i - \mathbf{R}_l\|_F^2, \quad (3.2)$$

where $\mathcal{N}(\mathcal{E}_i)$ are the neighboring edge sets of \mathcal{E}_i ; α is a weighting coefficient; A is the area of the whole mesh surface, which is used to make the energy scale invariant (scaling the edges by s , would scale the first term by s^2 , which is the scale of A in the second term); w_{il} is scalar weight; $\|\cdot\|_F$ denotes the Frobenius norm. The second term is the bending energy, which penalizes the rigidity difference between a cell and its neighboring cells. In this way, they have made up the missing bending energy in ARAP to form a *consistent* one.

Although SR-ARAP energy overcomes weaknesses in the ARAP surface deformation, which achieves the consistent discretization for the surfaces, it is still not capable of handling surfaces with different sizes since it aims at preserving the local rigidity of each cell. Yamazaki et al. [2013] propose an as-similar-as-possible energy to measure the local similarity variation, they add a freedom to the differential of a mapping with an isotropic scale s_i , which enables the energy to allow local scale to each discrete cell leading to capable to handling large size difference. The ASAP energy is defined as:

$$E_{ARAP}(\mathcal{S}, \mathcal{S}') = \sum_i \sum_{(j,k) \in \mathcal{E}_i} w_{jk} \|(\mathbf{p}'_j - \mathbf{p}'_k) - s_i \mathbf{R}_i(\mathbf{p}_j - \mathbf{p}_k)\|^2, \quad (3.3)$$

However, similar to ARAP energy, ASAP energy is also not consistent, which results in susceptible to fold-over and self-intersection.

In order to tackle the problems mentioned above, we propose a novel consistent energy called consistent as-similar-as-possible (CASAP) energy by introducing local scaling to the rotation in the prior SR-ARAP energy. It deforms surface meshes in an as-similar-as-possible manner, which is not only able to handle shapes with different size, but also reduce the occurrence of fold-over and self-intersection.

3.2 Consistent ASAP energy

Assuming we are deforming a mesh \mathcal{S} into \mathcal{S}' in an as similar as possible way, unlike [Sorkine & Alexa 2007; Yamazaki et al. 2013; Yoshiyasu et al. 2014] regarding spokes as the edge sets, the edge sets chosen in this research is spokes and rims (denoted as \mathcal{E}_i) in order to arrive at an analyzable energy Chao et al. [2010]. If the deformation $\mathcal{C}_i \rightarrow \mathcal{C}'_i$ is similar, then there exists a scale factor $s_i > 0$ and a rotation matrix \mathbf{R}_i such that

$$\mathbf{p}'_j - \mathbf{p}'_k = s_i \mathbf{R}_i (\mathbf{p}_j - \mathbf{p}_k), \forall (j, k) \in \mathcal{E}_i, \quad (3.4)$$

where \mathcal{E}_i consists of the set of edges incident to vertex i (the spokes) and the set of edges in the loop (the rims) of vertex i in the surface mesh \mathcal{S} . When the deformation is not similar, we can still find the best approximating scale factor s_i and rotation \mathbf{R}_i by minimizing a weighted cost function

$$E(C_i, C'_i) = \sum_{(j,k) \in \mathcal{E}_i} w_{jk} \|(\mathbf{p}'_j - \mathbf{p}'_k) - s_i \mathbf{R}_i (\mathbf{p}_j - \mathbf{p}_k)\|^2, \quad (3.5)$$

where w_{jk} is the edge weighting coefficient.

In order to measure the similarity of a deformation of the whole mesh, we sum up over the deviations from similarity per cell which yields us following ASAP energy functional:

$$E_{ASAP}(\mathcal{S}, \mathcal{S}') = \sum_i E(C_i, C'_i) = \sum_i \sum_{(j,k) \in \mathcal{E}_i} w_{jk} \|(\mathbf{p}'_j - \mathbf{p}'_k) - s_i \mathbf{R}_i (\mathbf{p}_j - \mathbf{p}_k)\|^2. \quad (3.6)$$

However, according to Chao et al. [2010], the ASAP energy we obtained so far is not *consistent* yet. In fact, the resulting ASAP energy differs from the continuous one by a bending term. Inspired by Levi & Gotsman [2015] we incorporate the smooth regularization into (3.6)

leading us to a *consistent* ASAP energy:

$$E_{\text{CASAP}}(\mathcal{S}, \mathcal{S}') = \sum_i \left(\sum_{(j,k) \in \mathcal{E}_i} w_{jk} \|(\mathbf{p}'_j - \mathbf{p}'_k) - s_i \mathbf{R}_i (\mathbf{p}_j - \mathbf{p}_k)\|^2 + \alpha A \sum_{\mathcal{E}_l \in \mathcal{N}(\mathcal{E}_i)} w_{il} \|\mathbf{R}_i - \mathbf{R}_l\|_F^2 \right), \quad (3.7)$$

where $\mathcal{N}(\mathcal{E}_i)$ are the neighboring edge sets of \mathcal{E}_i . The second term we add is the bending energy Levi & Gotsman [2015], which penalizes the similarity difference between a vertex and its neighboring vertex. In this way, we have made up the missing bending energy in ASAP energy to form a *consistent* one (Figure 3.1).

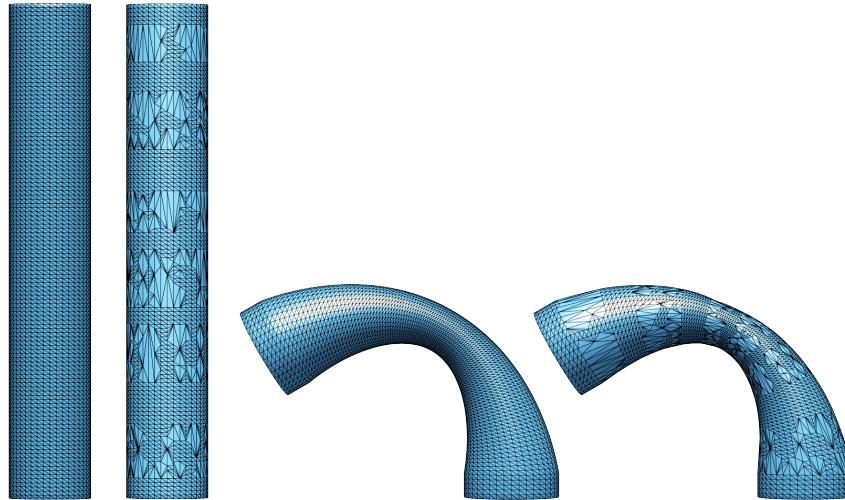


Figure 3.1: CASAP deformation on the same object with different resolutions results in very similar qualitative behaviors.

3.3 Optimization

In this subsection, we introduce the optimization algorithm to minimize the CASAP energy in (3.7). Note that except the vertex positions \mathbf{p}_i are unknown, s_i and \mathbf{R}_i in (3.7) are also unknown for each vertex. We employ the alternating optimization scheme following [Sorkine & Alexa 2007; Yamazaki et al. 2013; Levi & Gotsman 2015] to solve them respectively. Each iteration consists of a local step followed by a global step. In local step, we optimize s_i and \mathbf{R}_i with \mathbf{p}'_i fixed. By contrast, \mathbf{p}'_i are

optimized with s_i and \mathbf{R}_i fixed in global step.

Local step In this step, \mathbf{p}'_i are fixed, and then we solve \mathbf{R}_i , s_i in sequence to construct *consistent* ASAP energy (3.7). For convenience, let us denote the edge $\mathbf{e}_{jk} := \mathbf{p}_j - \mathbf{p}_k$ and $\mathbf{e}'_{jk} := \mathbf{p}'_j - \mathbf{p}'_k$. Then we can change the formula (3.7) for cell i as

$$\sum_{(j,k) \in \mathcal{E}_i} w_{jk} \|\mathbf{e}'_{jk} - s_i \mathbf{R}_i \mathbf{e}_{jk}\|^2 + \alpha A \sum_{\mathcal{E}_l \in \mathcal{N}(\mathcal{E}_i)} w_{il} \|\mathbf{R}_i - \mathbf{R}_l\|_F^2. \quad (3.8)$$

First we solve for optimal rotation \mathbf{R}_i . Extending the equation (3.8) and dropping the terms that do not contain \mathbf{R}_i , we are remained with

$$\begin{aligned} & \underset{\mathbf{R}_i}{\operatorname{argmin}} \ Tr(-\mathbf{R}_i(2 \sum_{(j,k) \in \mathcal{E}_i} s_i \mathbf{e}_{jk} \mathbf{e}'_{jk}^T + 2\alpha A \sum_{\mathcal{E}_l \in \mathcal{N}(\mathcal{E}_i)} w_{il} \mathbf{R}_l^T)) \\ &= \underset{\mathbf{R}_i}{\operatorname{argmax}} \ Tr(\mathbf{R}_i \mathbf{S}_i), \end{aligned} \quad (3.9)$$

where Tr is the trace of a matrix, \mathbf{S}_i is defined as

$$\mathbf{S}_i = 2 \sum_{(j,k) \in \mathcal{E}_i} s_i \mathbf{e}_{jk} \mathbf{e}'_{jk}^T + 2\alpha A \sum_{\mathcal{E}_l \in \mathcal{N}(\mathcal{E}_i)} w_{il} \mathbf{R}_l^T.$$

Following Sorkine & Alexa [2007] we derive the optimal rotation \mathbf{R}_i from the singular value decomposition of $\mathbf{S}_i = \mathbf{U}_i \boldsymbol{\Sigma}_i \mathbf{V}_i^T$:

$$\mathbf{R}_i = \mathbf{V}_i \mathbf{U}_i^T. \quad (3.10)$$

The determinate of a rotation matrix should be one, if $\det(\mathbf{R}_i) < 0$ then we change the sign of the column of \mathbf{U}_i corresponding to the smallest singular.

Then we compute scale factor s_i . Since the second term in (3.8) is independent with s_i , we only extend the first term and divide extended

terms by s_i

$$\operatorname{argmin}_{s_i, \mathbf{R}_i} \operatorname{Tr} \left(\sum_{(j,k) \in \mathcal{E}_i} w_{jk} \left(\frac{1}{s_i} \|\mathbf{e}'_{jk}\|^2 - 2\mathbf{R}_i \mathbf{e}_{jk} \mathbf{e}'_{jk}^T + s_i \|\mathbf{e}_{jk}\|^2 \right) \right). \quad (3.11)$$

Taking derivative (3.11) w.r.t. s_i and letting the derivative to be zero yields

$$s_i = \left(\frac{\sum_{(j,k) \in \mathcal{E}_i} w_{jk} \|\mathbf{e}'_{jk}\|^2}{\sum_{(j,k) \in \mathcal{E}_i} w_{jk} \|\mathbf{e}_{jk}\|^2} \right)^{\frac{1}{2}} \quad (3.12)$$

Global step In this step, vertex positions \mathbf{p}'_i are optimized from s_i, \mathbf{R}_i obtained by the local step.

Taking partial derivative of (3.7) w.r.t. the position \mathbf{p}'_i (note that the second term has nothing to do with \mathbf{p}'_i), we arrive at

$$\begin{aligned} \frac{\partial E(\mathbf{p}')}{\partial \mathbf{p}'_i} = & 2 \sum_{j \in \mathcal{N}(i)} (w_{ij} (3(\mathbf{p}'_i - \mathbf{p}'_j) - (s_i \mathbf{R}_i + s_j \mathbf{R}_j + s_m \mathbf{R}_m)(\mathbf{p}_i - \mathbf{p}_j)) \\ & + w_{ji} (3(\mathbf{p}'_i - \mathbf{p}'_j) - (s_i \mathbf{R}_i + s_j \mathbf{R}_j + s_n \mathbf{R}_n)(\mathbf{p}_i - \mathbf{p}_j))), \end{aligned} \quad (3.13)$$

where $\mathcal{N}(i)$ is one-ring neighbors of vertex \mathbf{p}'_i ; s_m, s_n and $\mathbf{R}_m, \mathbf{R}_n$ are the scalar factors and rotation matrices of the vertices $\mathbf{p}_m, \mathbf{p}_n$ which are the opposite vertices of the edge \mathbf{e}_{ij} . Setting partial derivative (3.13) to zero gives the following sparse linear system of equations:

$$\begin{aligned} \sum_{j \in \mathcal{N}(i)} (w_{ij} + w_{ji}) (\mathbf{p}'_i - \mathbf{p}'_j) = & \frac{1}{3} \sum_{j \in \mathcal{N}(i)} (w_{ij} (s_i \mathbf{R}_i + s_j \mathbf{R}_j + s_m \mathbf{R}_m) \\ & + w_{ji} (s_i \mathbf{R}_i + s_j \mathbf{R}_j + s_n \mathbf{R}_n)) (\mathbf{p}_i - \mathbf{p}_j). \end{aligned} \quad (3.14)$$

Notice that the linear combination on the left-hand side is the discrete Laplace-Beltrami operator applied to \mathbf{p}' . Now the system of equations

can be reduced as

$$\mathbf{L}\mathbf{p}' = \mathbf{d}, \quad (3.15)$$

where \mathbf{L} represents the discrete Laplace-Beltrami operator, which only depends on the initial mesh, thus it can be pre-factored for efficiency; \mathbf{d} is given by the right-hand side of (3.14).

Up to now, the optimization of *consistent* ASAP energy can be summarised as Algorithm (1).

Algorithm 1 *Consistent* ASAP Energy Optimization

- 1: **while** not converged **do**
 - 2: Compute \mathbf{R}_i by solving equations (3.10).
 - 3: Compute s_i by solving equations (3.12).
 - 4: Compute \mathbf{p}' and update surface \mathcal{S}' by solving equation (3.15).
 - 5: **end while**
-

3.4 Experiments

We compare our *consistent as-similar-as-possible* (CASAP) deformation approach with other three deformation methods: ARAP [Sorkine & Alexa 2007], SR-ARAP [Levi & Gotsman 2015] and ASAP [Yamazaki et al. 2013]) in Figure 3.2. The result of ARAP is not satisfying because its energy is not *consistent*. SR-ARAP overcomes the weakness of ARAP, offering a *consistent* energy. However, it cannot handle local scalability. ASAP allows piecewise scale but its energy is not *consistent*, which may lead to undesirable results such as fold-over and self-intersection. CASAP combines the benefits of ASAP with the advantages of the SR-ARAP approach such that it can not only handle local scale but also guarantee the deformation smoothness. Moreover, in terms of isometric deformation, it produce competitive results as good as SR-ARAP. Figure 3.1 shows the advantages of our *consistent* energy.

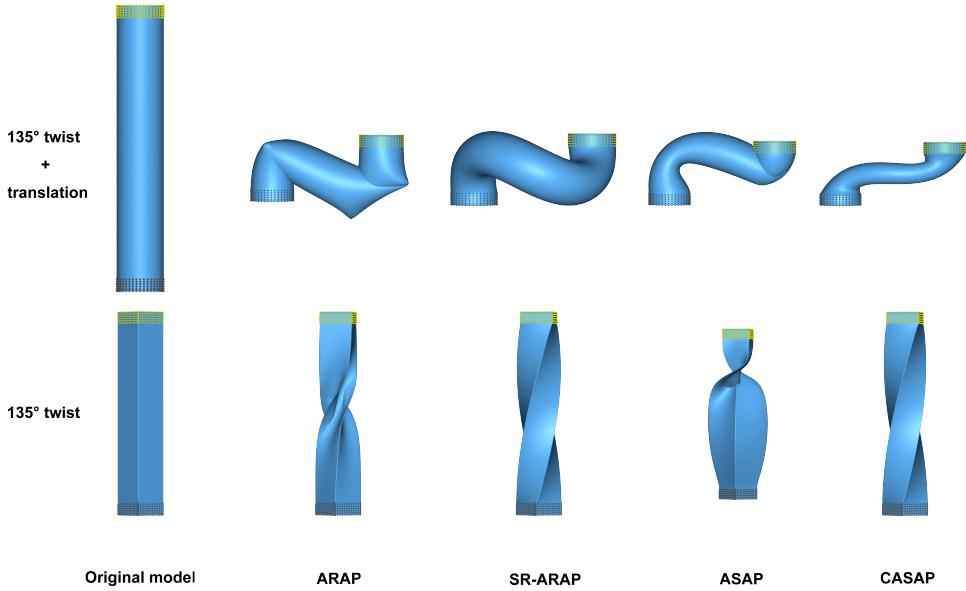


Figure 3.2: Different deformation approaches comparison. Rows show different transformations, while columns represent different deformation methods. The grey points are fixed and the yellow ones indicate control points.

3.5 Conclusion

We proposed a novel consistent as-similar-as-possible surface deformation method, which not only allows local scale to each discrete cell but also achieves the consistent discretization for surfaces. The important features of our approach are (1) robustness, resulting from the minimization procedure that is guaranteed to not increase energy in each step; (2) simplicity, as each iteration of the minimization solves a linear system; (3) efficiency, because the laplace system matrix is constant throughout the iterations and can be pre-factored just for once. We fitted the mapping differential into a similarity matrix, which is an isotropic scale factor times a rotation matrix. The scale factor is able to handle size difference while the rotation matrix part prevents local stretch and distortion. Meanwhile, the added rotation smoothing term compensates the bending energy which makes CASAP energy consistent, reducing the risk of fold-over and self-intersect occurrence. Our technique fills the missing gaps between SR-ARAP and ASAP. It combines the benefits of SR-ARAP and the advantage of ASAP, producing a consistent

discretization and allowing local scalability to handle large deformation consistently without compromising the efficiency.

Chapter 4

3D shape registration

This chapter presents a novel consistent non-isometric registration approach to handle large deformation and reduce the chance of occurrence of fold-over and intersection. The proposed *consistent* energy only requires a small number of landmarks with little user manual input. It constrains the template in an *as-similar-as-possible* way so that local scales are allowed to change and angles of triangle meshes are as much as preserved, which enables it to address shapes in large size variation with less shear distortion. Extensive experimental results have demonstrated the effectiveness of CASAP in comparison to the state-of-the-art approaches.

Over the last two decades, *non-rigid registration* has been an active research topic [Van Kaick et al. 2011]. *Non-rigid registration* can be generally divided into *isometric* and *non-isometric*. *Isometric registration* aims at finding a set of local rigid transformations but lacks local scalability due to its length-preserving property. *Non-isometric registration* can be further classified into: *equiareal*, *smooth* and *similar*. Specifically, *equiareal registration* has scale-preserving property but is unable to address size difference between the template and the target. In contrast, *smooth registration* based on smoothness regularization afford to size difference. However, it allows piecewise stretching transformation, which can result in shear distortion. *Similar registration* fits the deformation gradient into a similarity matrix, which is an isotropic scale factor times

a rotation matrix. The scale factor is able to handle size difference while the rotation matrix part prevents local stretch and distortion. Thus it has been widely used in many applications [Yamazaki et al. 2013; Yoshiyasu et al. 2014; Papazov & Burschka 2011] to align surfaces with different sizes and details. However, the energies they adopt to constrain the local deformation similarity are not *consistent*. This may tend to produce fold-over and self-intersection during transformation.

To our best knowledge, there is no such a surface registration method which takes the local scalability and consistency into consideration at the same time. In this report, we propose a *consistent as-similar-as-possible* (CASAP) surface registration approach. Given a small number of feature points, CASAP is not only able to fit the template to the target with different size and poses, but also preserves the structure of the template well, which is a quite important property in surface cross-parameterization.

4.1 CASAP surface registration

Given a template surface \mathcal{S} and a target surface \mathcal{T} , the goal of surface registration is to deform the surface \mathcal{S} into \mathcal{S}' so that \mathcal{S}' can be sufficiently close to surface \mathcal{T} with structure preserved and less distortion. With that purpose in mind, we enforce *consistent as-similar-as-possible* (ASAP) regularization on the template surface when we are attracting it to the target. Let $\mathbf{p}, \mathbf{p}', \mathbf{q}$ denote the vertex positions on surface $\mathcal{S}, \mathcal{S}', \mathcal{T}$ respectively, we define the cost function as

$$E(\mathbf{p}') = w_d E_d(\mathbf{p}') + w_c E_c(\mathbf{p}') + w_f E_f(\mathbf{p}'), \quad (4.1)$$

where E_d measures ASAP deformation consistency, E_c penalizes distances between the points of template and their correspondences on the target, and E_f penalizes distances between the feature points of template and target surface. The weights before these energy terms adjust the influence they account for in total energy. As E_d has been introduced in last chapter, in the following subsections, we will introduce the last

two energy terms.

4.1.1 Correspondence constraints

In order to attract points on the template towards the target, we need to find their correspondent vertices on the target surface. Many works [Yamazaki et al. 2013; Yoshiyasu et al. 2014; Gilles et al. 2010] regard the closest points as goal positions, however, correspondences chosen by these approaches are not quite appropriate as they only consider distances between the closest points of template and target surface. Inspired by [Qixing et al. 2008; Papazov & Burschka 2011] we will also consider feature descriptors and smooth factor. Starting from the closest points on the target, we then flood over their neighbors to find out the smallest matching energy points until converge. We define matching energy E_m between points of the template and the target as

$$E_m(\mathbf{p}_i, \mathbf{q}_j) = \|d_f(\mathbf{p}_i, \mathbf{q}_j) - \overline{d_f(\mathbf{p}_i, \mathbf{q}_j)}\|^2, \quad (4.2)$$

where \mathbf{p}_i is vertex i on template surface and \mathbf{q}_j is vertex j on target surface; the feature descriptors distance is defined as $d_f(\mathbf{p}_i, \mathbf{q}_j) = f(\mathbf{p}_i) - f(\mathbf{q}_j)$, where $f(v)$ is the feature vector for vertex v , we concatenate all feature descriptors into a single feature vector; the mean value distance $\overline{d_f(\mathbf{p}_i, \mathbf{q}_j)} = \frac{1}{|\mathcal{N}(j)|+1} \sum_{k \in \mathcal{N}(j) \cup j} d_f(\mathbf{p}_i, \mathbf{q}_k)$, where $\mathcal{N}(j)$ is the 1-ring neighbors of vertex j on the target surface. The minimization of E_m will result in a smoother correspondence field than the methods that depend on the descriptor similarity only.

There is a great number of feature descriptors that characterize the geometric properties of the point or of its neighborhood, often in a multi-scale way, for example, various notions of curvature (Gaussian, mean) Meyer et al. [2003], diffusion-based descriptors, such as the Heat or Wave Kernel Signatures Sun et al. [2009]; Aubry et al. [2011], or more classical descriptors such as spin images or shape contexts Johnson & Hebert [1999]; Belongie et al. [2002]. In our experiment we concatenate vertex position, vertex normal, multi-scale mean curvatures Panizzo

et al. [2010], Wave Kernel Signatures Aubry et al. [2011] and Scale-invariant Heat Kernel Signatures Bronstein & Kokkinos [2010] to form a feature vector.

In order to prevent unnecessary matchings, we filter out the pairs if the distance between them exceeds D or if the angle between their normals exceeds a threshold Θ . Thus the algorithm of finding correspondence $\mathbf{q}_{\text{idx}(i)}$ on the target surface for each point on the template can be summarized as Algorithm (2), where $\text{idx}(i)$ is the index of the target point that is matched with template vertex i . After given the correspondence

Algorithm 2 Find correspondence for template vertex p_i

```

1: Find the closest point  $\mathbf{q}_j$  on the target
2: if the distance between  $\mathbf{p}_i$  and  $\mathbf{q}_j$  exceeds  $D$  or the angle between
   their normals exceeds  $\Theta$  then
3:   return NULL
4: end if
5:  $k = j$ 
6: do
7:    $j = k$ 
8:   Find  $k \in \mathcal{N}(j) \cup j$  which minimizes  $E_m$ 
9: while  $k \neq j$ 
10:  $\text{idx}(i) = k$ 
11: return  $\mathbf{q}_{\text{idx}(i)}$ 

```

of template vertices, the template surface can be attracted towards the target according to the matching pairs. However, in order to avoid extreme distortion in tangential space, rather than attracting the template points to their correspondences directly, we attract them to the projections of their correspondences on their normals denoted by $\text{Proj}(\mathbf{q}_{\text{idx}(i)})$ (Figure 4.1). Now the correspondent constraint energy in (4.1) can be expressed as

$$E_c(\mathbf{p}') = \|\mathbf{C}_c \mathbf{p}' - \text{Proj}(\mathbf{D}_c \mathbf{q})\|_F^2, \quad (4.3)$$

where \mathbf{p}', \mathbf{q} are the vertex positions on surface $\mathcal{S}', \mathcal{T}$ respectively, and $\mathbf{C}_c, \mathbf{D}_c$ are the sparse matrices that define the filtered matching correspondences between \mathcal{S}' and \mathcal{T} . Assuming the m -th correspondence is \mathbf{p}_i

on \mathcal{S}' and $\mathbf{q}_{\text{idx}(i)}$ on \mathcal{T} , then

$$\mathbf{C}_c(m, n) = \begin{cases} 1, & \text{if } n=i \\ 0, & \text{if } n \neq i \end{cases}, \quad \mathbf{D}_c(k, n) = \begin{cases} 1, & \text{if } n=\text{idx}(i) \\ 0, & \text{if } n \neq \text{idx}(i) \end{cases},$$

note that m and k may have different dimensions as the number of vertices between the template and the target are not necessary to be the same.

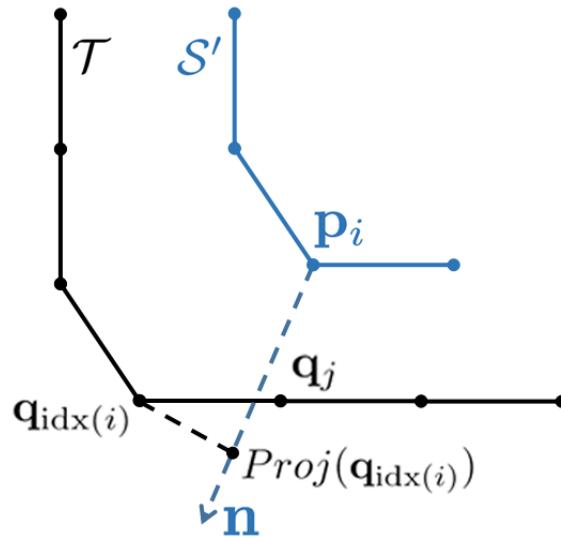


Figure 4.1: \mathbf{q}_j is the closest vertex on the target to \mathbf{p}_i ; $\mathbf{q}_{\text{idx}(i)}$ is the correspondent vertex found by minimizing the matching energy; \mathbf{n} is the normal vector of \mathbf{p}_i ; $\text{Proj}(\mathbf{q}_{\text{idx}(i)})$ is the projection of $\mathbf{q}_{\text{idx}(i)}$ onto normal vector \mathbf{n} .

4.1.2 Feature point constraints

For the fitting of the template's pose and size to the target, several feature correspondences are required to be established. Feature point constraints are designed to drag feature points on the template towards corresponding target ones. This constraint energy can be represented as

$$E_f(\mathbf{p}') = \|\mathbf{C}_f \mathbf{p}' - \mathbf{D}_f \mathbf{q}\|_F^2, \quad (4.4)$$

where $\mathbf{C}_f, \mathbf{D}_f$ are the sparse matrices that define the feature point pairs between \mathcal{S}' and \mathcal{T} .

4.1.3 Optimization

In this subsection, we introduce the optimization algorithm to minimize the total energy in (4.1). There are two loops in the optimization: the outer loop searches for the correspondent vertices to construct E_c , the inner loop optimizes the deformed vertex positions by minimizing $E(\mathbf{p}')$. Once the inner loop is converged, weights are adjusted and a new outer iteration starts again.

Taking derivative of the total energy (4.1) w.r.t. \mathbf{p}' gives us a linear system:

$$\mathbf{A}^T \mathbf{A} \mathbf{p}' = \mathbf{A}^T \mathbf{b}, \quad (4.5)$$

where

$$\mathbf{A} = \begin{pmatrix} w_d \mathbf{L} \\ w_c \mathbf{C}_c \\ w_f \mathbf{C}_f \end{pmatrix}, \mathbf{b} = \begin{pmatrix} w_d \mathbf{d} \\ w_c \text{Proj}(\mathbf{D}_c \mathbf{q}) \\ w_f \mathbf{D}_f \vec{q} \end{pmatrix},$$

where \mathbf{L} and \mathbf{d} from equation (3.15).

Up to now, the routine of *consistent* ASAP surface registration can be summarised as Algorithm (3).

Algorithm 3 *Consistent* ASAP Surface registration

- 1: Specify the feature points.
 - 2: **while** not converged **do**
 - 3: **while** not converged **do**
 - 4: Compute \mathbf{R}_i by solving equations (3.10).
 - 5: Compute s_i by solving equations (3.12).
 - 6: Compute \mathbf{p}' and update surface \mathcal{S}' by solving equation (4.5).
 - 7: **end while**
 - 8: Adjust weights in (4.5) and construct E_c
 - 9: **end while**
-

4.2 Coarse-to-fine fitting strategy

To improve the efficiency and robustness of registration, we take a coarse-to-fine fitting strategy. Instead of fitting overall template surface from the beginning, a coarse mesh is extracted from the original template mesh and then fitted to several feature points from the target model to roughly adjust the overall size of the template. In this way, approximated goal positions are obtained which is a better initial guess of fine fitting leading to fast converge and it also reduces the fold-over occurrence. Afterwards, a dense mesh is rebuilt from the deformed coarse mesh and fine fitting step is performed to produce the final result.

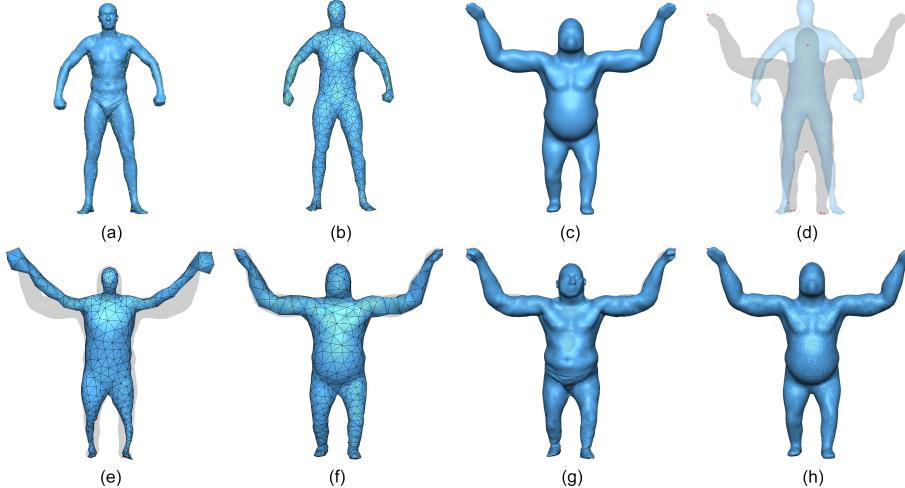


Figure 4.2: *Surface registration algorithm overview: (a) 500 sampled points (marked as yellow dots) from 12500 vertices of the original template model via farthest point sampling technique; (b) Remeshing from the sampled points as embedded coarse mesh; (c) Input of target surface; (d) The feature points specified by users (red dots for target and cyan dots for template); (e) Coarse fitting; (f) Mid-scale fitting; (g) Reconstructed through embedded deformation; (h) Fine fitting.*

4.2.1 Fitting Steps

There are four fitting steps: initialization, coarse fitting, mid-scale fitting and fine fitting:

Initialization In this step, a coarse mesh is extracted from the template first. We employ the farthest point sampling approach [Moenning & Dodgson 2003] to sample certain number of vertices to represent the shape of objects approximately (Figure 4.2a). Note that all of the sampled vertices are the subset of the original vertex set. Then the geodesic remeshing technique [Surazhsky et al. 2005] is used to generate the coarse mesh from the sampled points (Figure 4.2b).

Coarse Fitting We utilize the similarity constraints and feature point constraints to fit the coarse mesh to the feature points on the target so that the size and post of the template are roughly adjusted to the target (Figure 4.2e).

Mid-scale Fitting After fitting the template roughly to the target using feature points, the coarse mesh is deformed gradually toward the target. Apart from the two kinds of constraints adopted in the first step, correspondence constraints are also applied to achieve template attraction (Figure 4.2f).

Fine Fitting In this stage, a dense mesh is reconstructed from the deformed coarse mesh by embedded deformation [Sumner et al. 2007] (Figure 4.2g). The extracted coarse mesh is considered as deformation graph laid under the the dense mesh. From formulas (3.12) and (3.10), we associate an affine transformation with each vertex in the coarse graph. The deformed positions of vertices in the dense mesh can be calculated from the transformations of the deformation graph. We use the same approach as Yoshiyasu et al. [2014] to rebuild the dense mesh. Again, all the constraints are performed to fit the dense mesh to the target (Figure 4.2h).

4.2.2 Weights and parameters

In the initialization step, we regard the feature point constraints as boundary condition to induce deformation. In next two steps, we set $D = 0.02r_{box}$ and $\Theta = 90^\circ$ in our examples, where r_{box} is the length of the bounding box diagonal. As for the weights in the linear system (4.5), we use $w_d = 1000$, $w_c = 5$, $w_f = 10^5$ in the coarse fitting stage and divide w_d by 1.1 after every iteration until it less than 1. In the fine fitting, we take the same procedure with $w_f = 1$.

4.3 Experiments and Results

We tested our algorithm on various surfaces. For surface deformation, the data (cylinder and bar) in [Botsch & Sorkine 2008] are adopted. We show twist and translation deformation on these meshes in Figure 3.2. For surface registration, we use the human head mesh, 3D face scanning, the human body and animal models which are from SCAPE, TOSCA data sets. All the algorithms are implemented in MATLAB and all the statistics are measured on an Intel Xeon E5 3.4 GHz 64-bit workstation with 16GB of RAM.

Generic models We apply CASAP registration technique to register from one human head with holes to a face scanning from another human (Figure 4.3); from a human body to a gorilla (Figure 4.3, 4.2); from a pig to a horse (Figure 4.4). Each pair has large difference on size or details. CASAP not only is able to handle size difference as shown in whole-body registration example in Figure 4.2, but also can capture geometrical details such as the human expression (Figure 4.3) and preserve the structure of the template well, thus reducing the risk of producing fold-over (Figure 4.3, 4.4).

Comparisons We compare our registration technique to other state-of-the-art algorithms: *as-conformal-as-possible* surface registration (ACAP) [Yoshiyasu

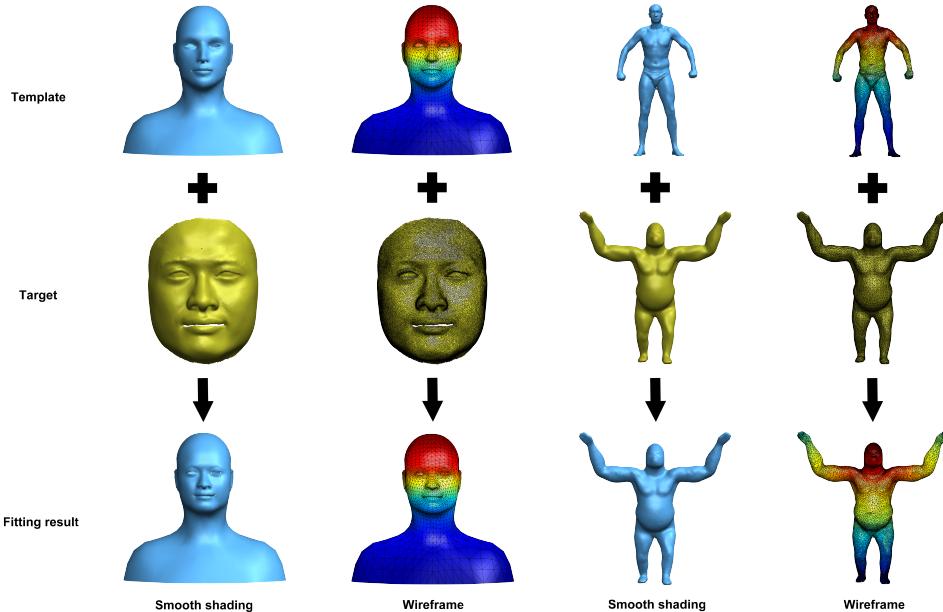


Figure 4.3: Consistent as-similar-as-possible (*CASAP*) non-isometric registration. Given a small number of feature correspondences (seven for the head registration and nine for the whole-body registration) only, *CASAP* not only is capable of fitting the template towards the target with different size (revealed in the whole-body registration example), but also captures the details well (shown in the face registration example) and preserves the structure of the template (seen from the colored wireframe shading mode).

et al. 2014], similarity-invariant shape registration (ASAP) [Yamazaki et al. 2013], the embedded deformation technique (ED) [Sumner et al. 2007], the shape matching based registration technique that minimizes the *as-similar-as-possible* energy (SM-ASAP) [Papazov & Burschka 2011], the Laplacian surface editing technique (LSE) [Sorkine et al. 2004] and the registration technique that utilizes the point-based deformation smoothness regularization (PDS) [Amberg et al. 2007] in Figure 4.4. ACAP employs nonlinear conformal stiffness and regularization terms in registration process, which produces the closest results to CASAP. However, since the regularization energy it adopts is not *consistent*, fold-overs still occur around the left wrist of gorilla and the neck of horse. ASAP and SM-ASAP do not require specifying feature points, but they are only able to handle surfaces with close initial alignment and similar poses. ED is an isometric counterpart of ACAP. As it cannot adjust local scale,

ED may produce poor initial shape estimation, which makes parts of surface converge to inaccurate places as shown at the right leg of gorilla. LSE cannot handle large deformation as it use a linear approximation of similar transformation. PDS is based on smoothness regularization, but it is too weak to handle shear distortions. Only CASAP exhibits no fold-over and almost no distortion in the examples, which produce quite pleasant visual results.

Table 4.1: Iteration steps and timings (in seconds). $\#O$, $\#I$ indicate the number of outer iteration steps and total inner iteration steps respectively. “Inner” indicate the average time required for each inner iteration step. “Total” indicate the total registration time.

	CASAP			ACAP			ASAP			ED			SM-ASAP			LSE			PDS									
	#O	#I	Inner	Total	#O	#I	Inner	Total	#O	#I	Inner	Total	#O	#I	Inner	Total	#O	#I	Inner	Total								
Gorilla	54	2882	0.035	100.87	73	313	0.338	105.94	20	683	0.038	25.95	10	1547	0.379	586.91	9	50	0.376	18.783	1	1	0.357	0.357	500	2166	0.036	77.976
Horse	54	2915	0.032	93.28	73	276	0.423	116.62	20	1501	0.033	49.53	10	1597	0.508	811.95	9	48	0.355	17.025	1	1	0.371	0.371	500	1870	0.035	65.45

From the perspective of quantitative evaluation, following the same criterion as in [Yoshiyasu et al. 2014], we measure 1) distance error, which is the average distance from the vertices of the deformed template to the corresponding points of the target relative to the bounding box diagonal, 2) angle error, which is the average angle deviation from the template, 3) bending error, which is the average deviation in dihedral angles from the template, 4) intersection error, which is the number of self-intersecting faces. These statistics can be found in Figure 4.5. All the errors of CASAP are the smallest among all the techniques except the bending error in horse example, in which the error of LSE is least. However, the registration result of LSE is not desirable due to its disability of handling large rotations. The number of self-intersecting faces in CASAP is zero, which reveals the ability of CASAP to reduce the change of fold-over and shear distortion appearance.

The number of iteration steps and timings are shown in Table 4.1. The time required for a single inner iteration of CASAP is minimum. Although it requires more iteration steps than ACAP to converge, the total registration time it spends is less than ACAP.

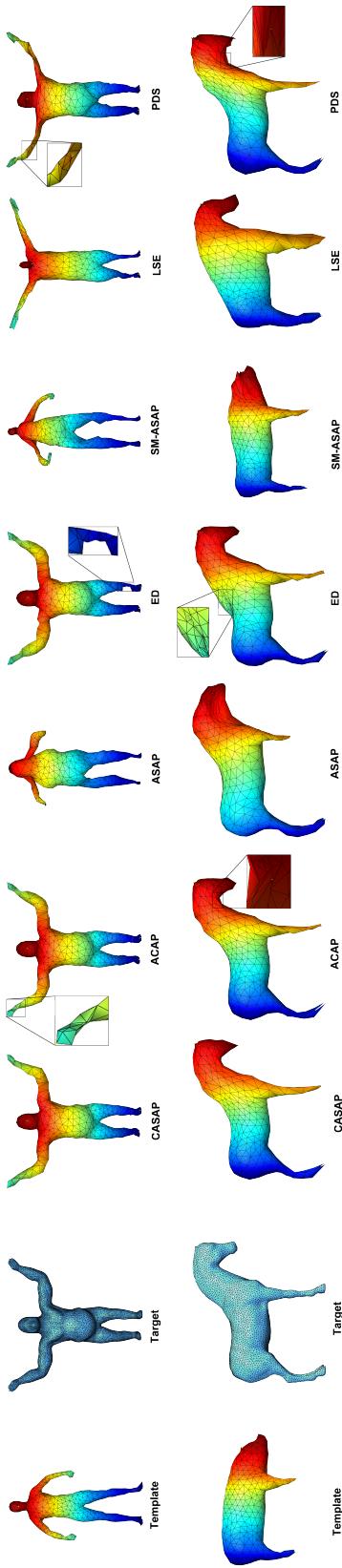


Figure 4.4: Different surface registration methods comparison. The left two columns are inputs while the rest are outputs by different surface registration methods. The yellow and red dots indicate the feature points on the template and the target respectively. The corresponding points are with same colors.

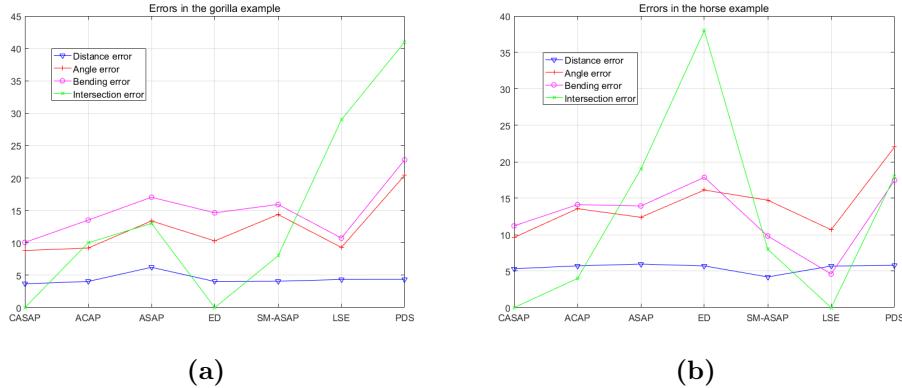


Figure 4.5: Quantitative comparison for the gorilla and horse registration results respectively.

4.4 Conclusion

We have presented a novel surface registration approach (CASAP) that constrains deformations locally as similar as possible. With the proposed *consistent* regularization energy, CASAP not only results in *consistent* discretization for surface, but also requires a small number of landmarks with only little user effort. Besides, CASAP is constrained in an as-similar-as-possible way so that angles of triangle meshes are well preserved and local scales are allowed to change. Experiments have shown that CASAP produced more accurate fitting results and preserved angles better than previous methods.

Chapter 5

Future Work

Although our *consistent* ASAP deformation technique and the coarse-to-fine strategy can efficiently reduce the chance of fold-over, it cannot avoid this problem completely, especially for parts with large curvature. Another limitation is that although our method only requires litter user efforts to specify several feature points on the shapes, it cannot achieve automatic goal without any user intervention. The following challenges need to be solved:

- A constrain penalizing element inversion need to be designed, or solutions must be provided when fold-overs or flips happen.
- An automatic feature correspondence algorithm capable of handling large, non-isometric shape variations are required.

To tackle these challenges, my future work will continue on tackling these challenges to provide more robust, accurate and automatic non-isometric shape registration techniques, which is not only able to find the correspondence between non-isometric shapes automatically, but also avoid inverse element to achieve robust to fold-over and self-intersection. In the following sections I will discuss existing techniques that are used to deal with these problems and their limitations, and outline my future plan to address these limitations.

5.1 Element inversion remove

In surface modeling and physics-based animation, a common way to avoid inversion elements is to design penalizing inversion constraints [Irving et al. 2004; Chao et al. 2010; Stomakhin et al. 2012; Setaluri et al. 2014; Civit-Flores & Susín 2014]. There are some specialized material models such as Neo-hookean elasticity [Bonet & Wood 1997] contain energies that increase to infinity as the area/volume of a deformed element degenerates to zero. Unfortunately, the numeric solutions to these energies are extremely complicate and may bring the classical Newton's method to a halt [Schüller et al. 2013]. In the context of inverse elastic shape design [Chen et al. 2014] discusses how to deal with these numerical complexities.

Schüller et al. [2013] propose Locally Injective Mapping (LIM), they apply a custom barrier function which results in real-time feedback to the user. Jin et al. [2014] improved LIM subsequently by online remeshing. LIM guarantees that no inverse elements will produce as long as the initial input configuration is inversion-free. Poranne & Lipman [2014] present an interactive inversion-free deformation approach with provable guarantees, however, their method is only limited to 2D deformations. Based on projections onto approximate tangent planes, Kovalevsky et al. [2015] provide an efficient algorithm to calculate bounded distortion mappings.

5.1.1 Locally injective mapping

Locally injective mapping is critical and desirable in shape deformation and mesh parameterization.

According to Tutte's theorem, a convex combination mapping in 2D from a disk-topology mesh is bijective if the target domain is convex [Tutte 1963; Floater 2003]. However, the bijectivity will lose if the domains are concave or self-overlapping, and even cannot be guaranteed on 3D convex domains [Floater & Pham-Trong 2006]. Boundary-free meth-

ods aiming at isometric/conformal parameterization are shown in the well-known MIPS method [Hormann & Greiner 2000] and its variations [Sander et al. 2001; Degener et al. 2003]. They also have their applicatons in mesh deformation [Eigensatz & Pauly 2009] and mesh improvement [Freitag & Knupp 2002; Jiao et al. 2011]. However, the calculation speed of MIPS is slow and it tends to converge to local minimum, as shown in [Sheffer et al. 2007].

Local injectivity can be achieved by parameterization methods like ABF++ [Sheffer et al. 2005] and circle pattern [Kharevych et al. 2006], but it is difficult to extend them to 3D and are not applicable to mesh deformation. [Schüller et al. 2013] provide a barrier term in their nonlinear optimization to avoid inverted elements so that the mapping always keeps locally injective. Although their method is efficient enough to offer interactive feedback to users for moderate-size meshes, it is out of control under extreme distortion. [Lipman 2012] build a maximal convex subspace to bound the maximal distortion and prevent inverted mesh elements. While their algorithm converge, locally injective mapping with bounded distortion can be achieved but with high computational cost, since a quadratic programming or semidefinite programming problem needs to optimize at each iteration. Moreover, a feasible solution space may not exist with empty convexified subspace or small upper bound of distortion. Poranne & Lipman [2014] also achieve smooth and low-distortion mappings by introducing the bounded distortion mapping technique to 2D meshless deformation. But it is hard to extend this method to 3D meshless deformation since the bound constraint is complicate. Weber & Zorin [2014] triangulate both the source and target domain to find a fixed boundary 2D mapping. [Jin et al. 2014] change mesh connectivity in 2D mesh deformation to ensure a valid map. These two methods do not take control on maximal distortion and are only limited in 2D mesh deformation.

5.1.2 Distortion measurement

The standard 2D MIPS energy measures the distortion of conformality of the mapping: $\sigma_1\sigma_2^{-1} + \sigma_2\sigma_1^{-1}$ where σ_1, σ_2 are the singular values of the Jacobian of the mapping associated with a triangle. Degener et al. [2003] propose to minimize $(\sigma_1\sigma_2^{-1})(\sigma_2\sigma_1^{-1})(\sigma_1\sigma_2 + \sigma_1^{-1}\sigma_2^{-1})^\theta$ to measure isometric distortion, which penalizes both conformal distortion and area distortion.

Other type energies like Dirichlet energy $\sigma_1^2 + \sigma_2^2$, stretch energy $\max \sigma_1, \sigma_2$, and Green-Lagrange energy $(\sigma_1^2 - 1)^2 + (\sigma_2^2 - 1)^2$ are also based on the measurement of singular values. As rigid as possible (ARAP), as similar as possible (ASAP) and as killing as possible (AKAP) are other three ways aiming at minimizing isometric/conformal distortions [Alexa et al. 2000; Igarashi et al. 2005; Sorkine & Alexa 2007; Solomon et al. 2011], but the mapping is not guaranteed to be local injective due to the solvers [Liu et al. 2008; Solomon et al. 2011] they used. All of the above methods sum up the distortions in a least squares way and do not consider the maximum distortion or the distortion distribution. To address these issues, Levi & Zorin [2014] provides a strict L_∞ -norm minimizer to take control of distortion control. To minimize the maximal conformal distortion, Weber et al. [2012] compute extremal quasiconformal maps while satisfying boundary constraints.

5.1.3 Summary

In this section, we reviewed the state-of-the-art methods of removing element inversion and distortion measurement. Since locally injective mapping guarantees that there is no inversion element as long as the initial input configuration is inversion-free. Following the work of [Schüller et al. 2013; Liu et al. 2016], we will add a barrier function into the total energy to make our mapping locally injective. Meanwhile, distortion will be measured to insure an inversion-free initial configuration and to evaluate the results of registration.

5.2 Automatic feature selection for correspondence

Geometric features on shapes can be presented in different ways, e.g., ridge and valley lines [Ohtake et al. 2004], prominent tip points [Zhang et al. 2005], or points with most unusual signatures [Gelfand et al. 2005]. Line-type features are usually unstable under shape articulation. The most prominent features of a model part are its extremity. They are suitable to be chosen to find feature correspondence as they are stable under bending and stretching. Regarding shape extremities as features enhances the correlation between correspondence analysis and object recognition. They have also been utilized in applications such as mesh parameterization [Zhang et al. 2005] and segmentation [Katz et al. 2005].

In this section, we reviewed the methods of automatic feature selection for correspondence between large non-rigid shapes. Following [Zhang et al. 2008], we will extract extremities for the template and the target respectively, then use a search tree to find out the best correspondence between these extremities via a best-first strategy.

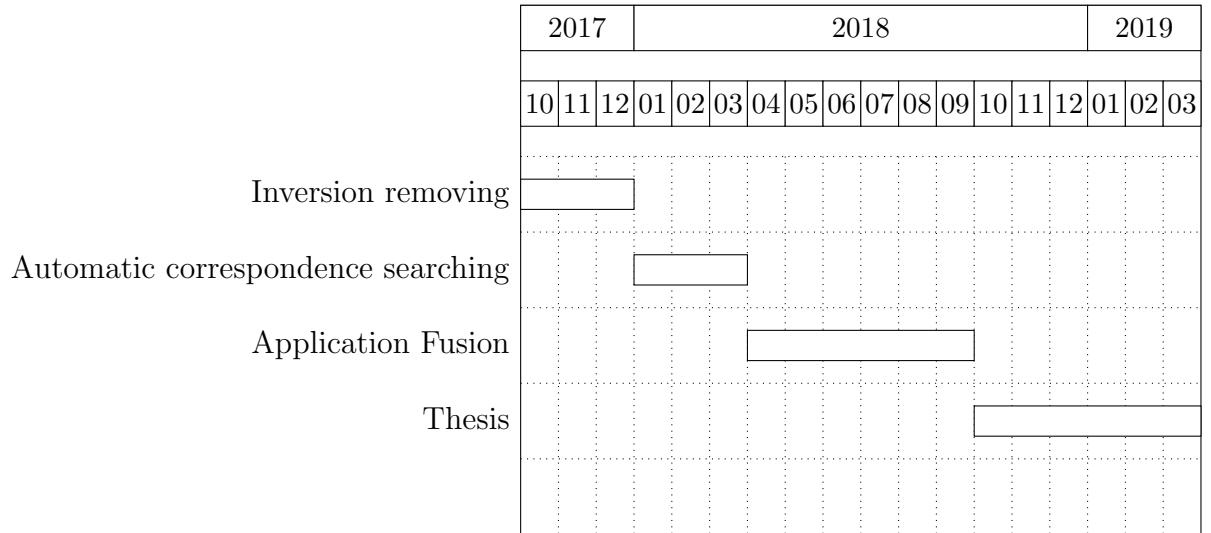
5.3 Future Research Plan

Considering the research progress so far and the unsolved challenges, I will continue my research on 3D shape deformation and 3D shape registration. First I will work on removing the fold-over completely and investigating automatically searching the correspondence between non-isometric shapes. Then, I will work on the applications of the developed technique like patient specific modeling and dynamic fusion. The research plan is detailed as follow:

- Applying inversion removing technique to avoid fold-overs and self-overlaps (3 Months);
- Using shape extremities to find correspondences between non-isometric

shapes automatically and fitting it into our registration energy (3 Months);

- Applying and extending our novel registration method to dynamic fusion allowing large deformation during fusion (6 Months);
- Writing PhD Thesis (6 Months).



Bibliography

- D. Aiger, et al. (2008). ‘4-points congruent sets for robust pairwise surface registration’. *ACM Transactions on Graphics (TOG)* **27**(3):85.
- M. Alexa, et al. (2000). ‘As-rigid-as-possible shape interpolation’. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 157–164. ACM Press/Addison-Wesley Publishing Co.
- B. Allen, et al. (2003). ‘The space of human body shapes: reconstruction and parameterization from range scans’. In *ACM Transactions on Graphics (TOG)*, vol. 22, pp. 587–594. ACM.
- B. Amberg, et al. (2007). ‘Optimal step nonrigid icp algorithms for surface registration’. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. IEEE.
- A. Angelidis, et al. (2006). ‘Swirling-sweepers: Constant-volume modeling’. *Graphical Models* **68**(4):324–332.
- D. Anguelov, et al. (2005). ‘The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces’. In *Advances in neural information processing systems*, pp. 33–40.
- O. K.-C. Au, et al. (2007). ‘Handle-aware isolines for scalable shape editing’. In *ACM Transactions on Graphics (TOG)*, vol. 26, p. 83. ACM.
- M. Aubry, et al. (2011). ‘The wave kernel signature: A quantum mechanical approach to shape analysis’. In *Computer Vision Work-*

- shops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 1626–1633. IEEE.
- M. A. Audette, et al. (2000). ‘An algorithmic overview of surface registration techniques for medical imaging’. *Medical image analysis* **4**(3):201–217.
- S. Belongie, et al. (2002). ‘Shape matching and object recognition using shape contexts’. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(4):509–522.
- P. J. Besl & N. D. McKay (1992). ‘Method for registration of 3-D shapes’. In *Robotics-DL tentative*, pp. 586–606. International Society for Optics and Photonics.
- S. Biasotti, et al. (2006). ‘Sub-part correspondence by structural descriptors of 3D shapes’. *Computer-Aided Design* **38**(9):1002–1019.
- B. Bickel, et al. (2008). ‘Pose-space animation and transfer of facial details’. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 57–66. Eurographics Association.
- J. Bonet & R. D. Wood (1997). *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press.
- M. Botsch & L. Kobbelt (2003). ‘Multiresolution surface representation based on displacement volumes’. In *Computer Graphics Forum*, vol. 22, pp. 483–491. Wiley Online Library.
- M. Botsch & L. Kobbelt (2004). ‘An intuitive framework for real-time freeform modeling’. In *ACM Transactions on Graphics (TOG)*, vol. 23, pp. 630–634. ACM.
- M. Botsch & L. Kobbelt (2005). ‘Real-Time Shape Editing using Radial Basis Functions’. In *Computer graphics forum*, vol. 24, pp. 611–621. Wiley Online Library.
- M. Botsch, et al. (2006). ‘PriMo: coupled prisms for intuitive surface modeling’. In *Symposium on Geometry Processing*, no. EPFL-CONF-149310, pp. 11–20.

- M. Botsch, et al. (2007). ‘Adaptive space deformations based on rigid cells’. In *Computer Graphics Forum*, vol. 26, pp. 339–347. Wiley Online Library.
- M. Botsch & O. Sorkine (2008). ‘On linear variational surface deformation methods’. *IEEE Transactions on Visualization and Computer Graphics* **14**(1):213–230.
- S. Bouaziz, et al. (2013). ‘Sparse iterative closest point’. In *Computer graphics forum*, vol. 32, pp. 113–123. Wiley Online Library.
- M. M. Bronstein & I. Kokkinos (2010). ‘Scale-invariant heat kernel signatures for non-rigid shape recognition’. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 1704–1711. IEEE.
- U. Castellani, et al. (2008). ‘Sparse points matching by combining 3D mesh saliency with statistical descriptors’. In *Computer Graphics Forum*, vol. 27, pp. 643–652. Wiley Online Library.
- G. Celniker & D. Gossard (1991). ‘Deformable curve and surface finite-elements for free-form shape design’. *ACM SIGGRAPH computer graphics* **25**(4):257–266.
- W. Chang & M. Zwicker (2008). ‘Automatic registration for articulated shapes’. In *Computer Graphics Forum*, vol. 27, pp. 1459–1468. Wiley Online Library.
- I. Chao, et al. (2010). ‘A simple geometric model for elastic deformations’. *ACM Transactions on Graphics (TOG)* **29**(4):38.
- X. Chen, et al. (2014). ‘An asymptotic numerical method for inverse elastic shape design’. *ACM Transactions on Graphics (TOG)* **33**(4):95.
- H. Chui & A. Rangarajan (2003). ‘A new point matching algorithm for non-rigid registration’. *Computer Vision and Image Understanding* **89**(2):114–141.
- O. Civit-Flores & A. Susín (2014). ‘Robust treatment of degenerate elements in interactive corotational fem simulations’. In *Computer Graphics Forum*, vol. 33, pp. 298–309. Wiley Online Library.

- S. Coquillart (1990). *Extended free-form deformation: a sculpturing tool for 3D geometric modeling*, vol. 24. ACM.
- P. Degener, et al. (2003). ‘An Adaptable Surface Parameterization Method.’. *IMR* **3**:201–213.
- M. Eigensatz & M. Pauly (2009). ‘Positional, Metric, and Curvature Control for Constraint-Based Surface Deformation’. In *Computer Graphics Forum*, vol. 28, pp. 551–558. Wiley Online Library.
- A. Elad & R. Kimmel (2003). ‘On bending invariant signatures for surfaces’. *IEEE Transactions on pattern analysis and machine intelligence* **25**(10):1285–1295.
- M. Floater (2003). ‘One-to-one piecewise linear mappings over triangulations’. *Mathematics of Computation* **72**(242):685–696.
- M. S. Floater & V. Pham-Trong (2006). ‘Convex combination maps over triangulations, tilings, and tetrahedral meshes’. *Advances in Computational Mathematics* **25**(4):347–356.
- L. A. Freitag & P. M. Knupp (2002). ‘Tetrahedral mesh improvement via optimization of the element condition number’. *International Journal for Numerical Methods in Engineering* **53**(6):1377–1391.
- N. Gelfand, et al. (2005). ‘Robust global registration’. In *Symposium on geometry processing*, vol. 2, p. 5.
- B. Gilles, et al. (2010). ‘Creating and Animating Subject-Specific Anatomical Models’. In *Computer Graphics Forum*, vol. 29, pp. 2340–2351. Wiley Online Library.
- I. Guskov, et al. (1999). ‘Multiresolution signal processing for meshes’. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 325–334. ACM Press/Addison-Wesley Publishing Co.
- K. Hormann & G. Greiner (2000). ‘MIPS: An efficient global parametrization method’. Tech. rep., ERLANGEN-NUERNBERG UNIV (GERMANY) COMPUTER GRAPHICS GROUP.

- K. Hormann & N. Sukumar (2008). ‘Maximum entropy coordinates for arbitrary polytopes’. In *Computer Graphics Forum*, vol. 27, pp. 1513–1520. Wiley Online Library.
- W. M. Hsu, et al. (1992). ‘Direct manipulation of free-form deformations’. In *ACM Siggraph Computer Graphics*, vol. 26, pp. 177–184. ACM.
- J. Huang, et al. (2006). ‘Subspace gradient domain mesh deformation’. In *ACM Transactions on Graphics (TOG)*, vol. 25, pp. 1126–1134. ACM.
- Q.-X. Huang, et al. (2008). ‘Non-rigid registration under isometric deformations’. In *Computer Graphics Forum*, vol. 27, pp. 1449–1457. Wiley Online Library.
- T. Igarashi, et al. (2005). ‘As-rigid-as-possible shape manipulation’. In *ACM transactions on Graphics (TOG)*, vol. 24, pp. 1134–1141. ACM.
- S. Irani & P. Raghavan (1999). ‘Combinatorial and experimental results for randomized point matching algorithms’. *Computational Geometry* **12**(1-2):17–31.
- G. Irving, et al. (2004). ‘Invertible finite elements for robust simulation of large deformation’. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 131–140. Eurographics Association.
- A. Jacobson (2013). *Algorithms and interfaces for real-time deformation of 2d and 3d shapes*. Ph.D. thesis.
- A. Jacobson, et al. (2011). ‘Bounded biharmonic weights for real-time deformation.’. *ACM Trans. Graph.* **30**(4):78.
- A. Jacobson, et al. (2010). ‘Mixed finite elements for variational surface modeling’. In *Computer Graphics Forum*, vol. 29, pp. 1565–1574. Wiley Online Library.
- A. Jacobson, et al. (2012). ‘Smooth Shape-Aware Functions with Controlled Extrema’. In *Computer Graphics Forum*, vol. 31, pp. 1577–1586. Wiley Online Library.

- V. Jain, et al. (2007). ‘Non-rigid spectral correspondence of triangle meshes’. *International Journal of Shape Modeling* **13**(01):101–124.
- X. Jiao, et al. (2011). ‘Simple and effective variational optimization of surface and volume triangulations’. *Engineering with Computers* **27**(1):81–94.
- Y. Jin, et al. (2014). ‘Remeshing-assisted Optimization for Locally Injective Mappings’. In *Computer Graphics Forum*, vol. 33, pp. 269–279. Wiley Online Library.
- A. E. Johnson & M. Hebert (1999). ‘Using spin images for efficient object recognition in cluttered 3D scenes’. *IEEE Transactions on pattern analysis and machine intelligence* **21**(5):433–449.
- P. Joshi, et al. (2007). ‘Harmonic coordinates for character articulation’. In *ACM Transactions on Graphics (TOG)*, vol. 26, p. 71. ACM.
- T. Ju, et al. (2005). ‘Mean value coordinates for closed triangular meshes’. In *ACM Transactions on Graphics (TOG)*, vol. 24, pp. 561–566. ACM.
- S. Katz, et al. (2005). ‘Mesh segmentation using feature point and core extraction’. *The Visual Computer* **21**(8-10):649–658.
- L. Kharevych, et al. (2006). ‘Discrete conformal mappings via circle patterns’. *ACM Transactions on Graphics (TOG)* **25**(2):412–438.
- O. Kin-Chung Au, et al. (2010). ‘Electors voting for fast automatic shape correspondence’. In *Computer Graphics Forum*, vol. 29, pp. 645–654. Wiley Online Library.
- L. Kobbelt, et al. (1998). ‘Interactive multi-resolution modeling on arbitrary meshes’. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pp. 105–114. ACM.
- L. Kobbelt, et al. (1999). ‘Multiresolution hierarchies on unstructured triangle meshes’. *Computational Geometry* **14**(1-3):5–24.
- S. Z. Kovalevsky, et al. (2015). ‘Large-scale bounded distortion mappings.’. *ACM Trans. Graph.* **34**(6):191–1.

- V. Kraevoy & A. Sheffer (2004). ‘Cross-parameterization and compatible remeshing of 3D models’. In *ACM Transactions on Graphics (TOG)*, vol. 23, pp. 861–869. ACM.
- V. Kraevoy & A. Sheffer (2005). ‘Template-Based Mesh Completion.’. In *Symposium on Geometry Processing*, vol. 385, pp. 13–22.
- Z. Levi & C. Gotsman (2015). ‘Smooth rotation enhanced as-rigid-as-possible mesh animation’. *IEEE Transactions on Visualization and Computer Graphics* **21**(2):264–277.
- Z. Levi & D. Zorin (2014). ‘Strict minimizers for geometric optimization’. *ACM Transactions on Graphics (TOG)* **33**(6):185.
- H. Li, et al. (2009). ‘Robust single-view geometry and motion reconstruction’. In *ACM Transactions on Graphics (TOG)*, vol. 28, p. 175. ACM.
- H. Li, et al. (2008). ‘Global Correspondence Optimization for Non-Rigid Registration of Depth Scans’. In *Computer Graphics Forum*, vol. 27, pp. 1421–1430. Wiley Online Library.
- X.-Y. Li & S.-M. Hu (2013). ‘Poisson coordinates’. *IEEE Transactions on visualization and computer graphics* **19**(2):344–352.
- Y. Lipman (2012). ‘Bounded distortion mapping spaces for triangular meshes’. *ACM Transactions on Graphics (TOG)* **31**(4):108.
- Y. Lipman, et al. (2007). ‘GPU-assisted positive mean value coordinates for mesh deformations’. In *Symposium on geometry processing*.
- Y. Lipman, et al. (2008). ‘Green coordinates’. In *ACM Transactions on Graphics (TOG)*, vol. 27, p. 78. ACM.
- Y. Lipman, et al. (2004). ‘Differential coordinates for interactive mesh editing’. In *Shape Modeling Applications, 2004. Proceedings*, pp. 181–190. IEEE.
- Y. Lipman, et al. (2005). ‘Linear rotation-invariant coordinates for meshes’. *ACM Transactions on Graphics (TOG)* **24**(3):479–487.
- L. Liu, et al. (2008). ‘A local/global approach to mesh parameterization’.

- In *Computer Graphics Forum*, vol. 27, pp. 1495–1504. Wiley Online Library.
- T. Liu, et al. (2016). ‘Fast and Robust Inversion-Free Shape Manipulation’. In *Computer Graphics Forum*, vol. 35, pp. 1–11. Wiley Online Library.
- R. MacCracken & K. I. Joy (1996). ‘Free-form deformations with lattices of arbitrary topology’. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 181–188. ACM.
- H. Maron, et al. (2016). ‘Point registration via efficient convex relaxation’. *ACM Transactions on Graphics (TOG)* **35**(4):73.
- M. Meyer, et al. (2003). ‘Discrete differential-geometry operators for triangulated 2-manifolds’. In *Visualization and Mathematics III*, pp. 35–57. Springer.
- C. Moenning & N. A. Dodgson (2003). ‘Fast marching farthest point sampling’.
- M. Müller, et al. (2005). ‘Meshless deformations based on shape matching’. *ACM Transactions on Graphics (TOG)* **24**(3):471–478.
- Y. Ohtake, et al. (2004). ‘Ridge-valley lines on meshes via implicit surface fitting’. *ACM transactions on graphics (TOG)* **23**(3):609–612.
- D. Panozzo, et al. (2010). ‘Efficient multiscale curvature and crease estimation’. *Proceedings of Computer Graphics, Computer Vision and Mathematics (Brno, Czech Republic)* **1**(6).
- C. Papazov & D. Burschka (2011). ‘Deformable 3D shape registration based on local similarity transforms’. In *Computer Graphics Forum*, vol. 30, pp. 1493–1502. Wiley Online Library.
- M. Pauly, et al. (2005). ‘Example-based 3D scan completion’. In *Symposium on Geometry Processing*, no. EPFL-CONF-149337, pp. 23–32.
- R. Poranne & Y. Lipman (2014). ‘Provably good planar mappings’. *ACM Transactions on Graphics (TOG)* **33**(4):76.

- H. Qixing, et al. (2008). ‘Non-Rigid Registration Under Isometric Deformations’. In *Computer Graphics Forum*, vol. 5. North Holland.
- A. R. Rivers & D. L. James (2007). ‘FastLSM: fast lattice shape matching for robust real-time deformation’. *ACM Transactions on Graphics (TOG)* **26**(3):82.
- S. Rusinkiewicz & M. Levoy (2001). ‘Efficient variants of the ICP algorithm’. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pp. 145–152. IEEE.
- P. V. Sander, et al. (2001). ‘Texture mapping progressive meshes’. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 409–416. ACM.
- C. Schüller, et al. (2013). ‘Locally injective mappings’. In *Computer Graphics Forum*, vol. 32, pp. 125–135. Wiley Online Library.
- T. W. Sederberg & S. R. Parry (1986). ‘Free-form deformation of solid geometric models’. *ACM SIGGRAPH computer graphics* **20**(4):151–160.
- R. Setaluri, et al. (2014). ‘Fast grid-based nonlinear elasticity for 2D deformations’. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 67–76. Eurographics Association.
- A. Sheffer, et al. (2005). ‘ABF++: fast and robust angle based flattening’. *ACM Transactions on Graphics (TOG)* **24**(2):311–330.
- A. Sheffer, et al. (2007). ‘Mesh parameterization methods and their applications’. *Foundations and Trends® in Computer Graphics and Vision* **2**(2):105–171.
- X. Shi, et al. (2007). ‘Mesh puppetry: cascading optimization of mesh deformation with inverse kinematics’. In *ACM Transactions on Graphics (TOG)*, vol. 26, p. 81. ACM.
- K. Singh & E. Fiume (1998). ‘Wires: a geometric deformation technique’. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pp. 405–414. ACM.

- K. Singh & E. Kokkevis (2000). ‘Skinning Characters using Surface Oriented Free-Form Deformations.’. In *Graphics interface*, vol. 2000, pp. 35–42.
- J. Solomon, et al. (2011). ‘As-Killing-As-Possible Vector Fields for Planar Deformation’. In *Computer Graphics Forum*, vol. 30, pp. 1543–1552. Wiley Online Library.
- O. Sorkine & M. Alexa (2007). ‘As-rigid-as-possible surface modeling’. In *Symposium on Geometry processing*, vol. 4.
- O. Sorkine, et al. (2004). ‘Laplacian surface editing’. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pp. 175–184. ACM.
- A. Stomakhin, et al. (2012). ‘Energetically consistent invertible elasticity’. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 25–32. Eurographics Association.
- R. W. Sumner, et al. (2007). ‘Embedded deformation for shape manipulation’. In *ACM Transactions on Graphics (TOG)*, vol. 26, p. 80. ACM.
- J. Sun, et al. (2009). ‘A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion’. In *Computer graphics forum*, vol. 28, pp. 1383–1392. Wiley Online Library.
- V. Surazhsky, et al. (2005). ‘Fast exact and approximate geodesics on meshes’. In *ACM Transactions on Graphics (TOG)*, vol. 24, pp. 553–560. ACM.
- G. K. Tam, et al. (2013). ‘Registration of 3D point clouds and meshes: a survey from rigid to nonrigid’. *IEEE Transactions on Visualization and Computer Graphics* **19**(7):1199–1217.
- D. Terzopoulos, et al. (1987). ‘Elastically deformable models’. In *ACM Siggraph Computer Graphics*, vol. 21, pp. 205–214. ACM.
- A. Tevs, et al. (2009). ‘Isometric registration of ambiguous and partial data’. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1185–1192. IEEE.

- G. Turk & M. Levoy (1994). ‘Zippered polygon meshes from range images’. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pp. 311–318. ACM.
- W. T. Tutte (1963). ‘How to draw a graph’. *Proceedings of the London Mathematical Society* **3**(1):743–767.
- O. Van Kaick, et al. (2011). ‘A survey on shape correspondence’. In *Computer Graphics Forum*, vol. 30, pp. 1681–1707. Wiley Online Library.
- W. von Funck, et al. (2006). ‘Vector field based shape deformations’. *ACM Transactions on Graphics (TOG)* **25**(3):1118–1125.
- O. Weber, et al. (2011). ‘A complex view of barycentric mappings’. In *Computer Graphics Forum*, vol. 30, pp. 1533–1542. Wiley Online Library.
- O. Weber, et al. (2012). ‘Computing extremal quasiconformal maps’. In *Computer Graphics Forum*, vol. 31, pp. 1679–1689. Wiley Online Library.
- O. Weber, et al. (2007). ‘Context-Aware Skeletal Shape Deformation’. In *Computer Graphics Forum*, vol. 26, pp. 265–274. Wiley Online Library.
- O. Weber & D. Zorin (2014). ‘Locally injective parametrization with arbitrary fixed boundaries’. *ACM Transactions on Graphics (TOG)* **33**(4):75.
- T. Weise, et al. (2009). ‘Face/off: Live facial puppetry’. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer animation*, pp. 7–16. ACM.
- S. Yamazaki, et al. (2013). ‘Non-rigid shape registration using similarity-invariant differential coordinates’. In *2013 International Conference on 3D Vision-3DV 2013*, pp. 191–198. IEEE.
- J. Yang, et al. (2015). ‘Sparse Non-rigid Registration of 3D Shapes’. In *Computer Graphics Forum*, vol. 34, pp. 89–99. Wiley Online Library.
- I.-C. Yeh, et al. (2011). ‘Template-based 3d model fitting using dual-

- domain relaxation'. *IEEE Transactions on Visualization and Computer Graphics* **17**(8):1178–1190.
- Y. Yoshiyasu, et al. (2014). ‘As-Conformal-As-Possible Surface Registration’. In *Computer Graphics Forum*, vol. 33, pp. 257–267. Wiley Online Library.
- Y. Yu, et al. (2004). ‘Mesh editing with poisson-based gradient field manipulation’. In *ACM Transactions on Graphics (TOG)*, vol. 23, pp. 644–651. ACM.
- R. Zayer, et al. (2005). ‘Harmonic guidance for surface deformation’. In *Computer Graphics Forum*, vol. 24, pp. 601–609. Wiley Online Library.
- E. Zhang, et al. (2005). ‘Feature-based surface parameterization and texture mapping’. *ACM Transactions on Graphics (TOG)* **24**(1):1–27.
- H. Zhang, et al. (2008). ‘Deformation-Driven Shape Correspondence’. In *Computer Graphics Forum*, vol. 27, pp. 1431–1439. Wiley Online Library.