



Max Planck Institut Informatik



Universität des Saarlandes

Naturwissenschaftlich-Technische Fakultät I
Fachrichtung Informatik

Master Thesis

Analysis of Energy Regularization for Harmonic Surface Deformation

Yeara Kozlov

Submitted: April 29, 2014

Reviewers: Dr.-Ing. Janick Martinez Esturo
Dr.-Ing. Tino Weinkauf

Statement in Lieu of an Oath

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis

Declaration of Consent

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Saarbrücken, April 29, 2014

Place and Date

Signature

Abstract

Recently it has been shown that regularization can be beneficial for a variety of geometry processing methods on discretized domains.

Linear energy regularization, proposed by Martinez Esturo et al. [MRT14], creates a global, linear regularization term which is strongly coupled with the deformation energy. It can be computed interactively, with little impact on runtime.

This work analyzes the effects of linear energy regularization on harmonic surface deformation, proposed by Zayer et al. [ZRKS05]. Harmonic surface deformation is a variational technique for gradient domain surface manipulation.

This work demonstrate that linear energy regularization can overcome some of the inherent limitations associated with this technique, can effectively reduce common artifacts associated with this method, eliminating the need for costly non-linear regularization, and expanding the modeling capabilities for harmonic surface deformation.

Acknowledgments

I would like to thank my advisor Janick Martinez Esturo for his patient guidance throughout this work, and for reviewing my draft. I would like to thank my second advisor, Tino Weinkauf for his technical support and providing the resources for this work.

Yeara Kozlov

April 29, 2014

“There is an art to the business of making sandwiches which it is given to few ever to find the time to explore in depth. It is a simple task, but the opportunities for satisfaction are many and profound.”

DOUGLAS ADAMS, THE ULTIMATE HITCHHIKER’S GUIDE

Contents

Abstract	v
Notation	xiii
1 Introduction	1
1.1 Physically Accurate Deformations	1
1.2 Plausible Deformations	2
1.3 Deformation Artifacts	2
1.4 Enhancing Harmonic Surface Deformations	2
1.5 Contributions	3
1.6 Structure	3
2 Related Work	5
2.1 Interactive Shape Manipulation	5
2.2 Shape Representation	5
2.3 Poisson Methods for Surface Editing	6
2.3.1 ARAP	8
2.4 Regularization Methods for Geometry Processing	9
2.5 Rotation Interpolation	9
3 Definitions and Mathematical Theory	11
3.1 Definitions	11
3.1.1 Triangle Mesh	11
3.1.2 Deformations and Constraints	12
3.2 Harmonic Functions	12
3.2.1 Laplace Operator	12
3.2.2 Poisson's Equation	13
3.2.3 The Laplace Equation	13
3.2.4 Harmonic Functions	13
3.3 Discrete Operators for 2-Manifold Meshes	14
3.3.1 Discrete Surface Formulation	14
3.3.2 Discrete Gradient Operator	14
3.3.3 Discrete Laplace-Beltrami Operator	15
3.4 Transformations	16
3.4.1 Transformation Matrices	16
3.4.2 Transformation Decomposition	17
3.4.3 Rotation Representation Using Quaternions	17
3.4.4 SLERP	18
3.5 Evaluating Surface Deformations	18
4 Harmonic Surface Deformation	23
4.1 Differential Surface Representation	23
4.2 Linear Methods	23
4.3 Gradient Domain Mesh Manipulation	24

4.4	Transformation Propagation using Harmonic Fields	24
4.5	Solving for Vertex Positions	25
4.6	Limitations	25
5	Energy Regularization	27
5.1	Tikhonov Regularization	27
5.2	Energy Smoothness Regularization	28
5.3	Implementation for Triangle Constant Energies on Surface Meshes	29
6	Enhancing Harmonic Surface Deformation using Energy Regularization	31
6.1	Discussion and Drawbacks	32
6.2	Approach Details	32
6.3	User Interaction	33
6.4	Discrete Mesh Representation and Operators	33
6.5	Constructing the Harmonic Field	33
6.6	Constructing the Guidance Field	34
6.7	Solving for Vertex Positions	35
6.8	Changing Regularization Weight	35
7	Results	37
7.1	Analysis Criteria	37
7.1.1	Quantitative Analysis	37
7.1.2	Qualitative Analysis	38
7.2	Visualization	38
7.2.1	Energy Visualization	38
7.2.2	Visualizing Deformation Constraints	39
7.2.3	Sampling the regularization weight	39
7.3	Comparison to Benchmark Deformations	40
7.4	Point-wise Constraints vs. Constrained Regions	45
7.5	Translation Only Deformations	45
7.6	Triangles Flips	46
7.7	Qualitative Analysis of Freestyle Deformations	50
7.8	Failure Cases	50
7.9	Performance	51
7.10	Implementation Details	51
7.11	Mesh Sources	51
8	Discussion	63
8.1	Effects of Regularization on Error Measures	63
8.2	Effects on Common Deformation Artifacts	64
8.3	High Regularization Weight	65
8.4	Handle Region Size	65
8.5	Comparison to Other Regularization Methods	66
8.6	Summary	66
9	Conclusions	69
Bibliography		71

Notation

$\mathcal{S}, \mathcal{S}'$	Source, target surfaces
$\mathcal{V}, \mathcal{V} $	Set of mesh vertices, its cardinality
$\mathcal{E}, \mathcal{E} $	Set of mesh edges, its cardinality
$\mathcal{T}, \mathcal{T} $	Set of mesh triangles, its cardinality
\mathbf{x}_i	Vertex position vector with coordinates $(x, y, z)^\top$
\mathbf{x}	Stacked mesh vertices position vector, $ \mathcal{V} \times 3$
\mathcal{N}_i	Neighborhood of vertex v_i
\mathbf{L}	Laplace-Beltrami operator
\mathbf{G}	Discrete gradient operator
\mathbf{h}	Harmonic guidance scalar field
\mathbf{y}	Stacked triangle gradients vector, $3 \mathcal{T} \times 3$
\mathbf{g}	Deformation guidance field, $3 \mathcal{T} \times 3$
\mathbf{T}	Transformation matrix
\mathbf{A}	Triangle area weighted norm, $3 \mathcal{T} \times 3 \mathcal{T} $
\mathbf{B}_d	$d \mathcal{E} \times d \mathcal{E} $ edge-length diagonal matrix. For manifold mesh in \mathbb{R}^d
\mathbf{D}	Discrete energy differential operator
β	Energy regularization weight parameter
\mathbf{W}_β	Beta weighted, energy regularization norm
e_f^i	Triangle-constant deformation energy on a single triangle
E	Deformation energy
$E_{\mathcal{S}}$	Smoothness term
$E_{\mathcal{D}}$	Data term
$E_{\mathcal{R}}$	Regularized deformation energy
E_{area}^N	Total normalized area change
$E_{\text{iso}}^N, E_{\text{conf}}^N$	Total normalized isometric, conformal errors
$e_{\text{iso}}^i, e_{\text{conf}}^i$	Local isometric, conformal errors

Chapter 1

Introduction

Geometry processing is a discipline in computer graphics that is concerned with efficient manipulation and analysis of geometric data. Common tasks include surface reconstruction, noise reduction in geometric data, surface simplification and interactive design.

A lot of research in geometry processing has been devoted to various deformation energies. Many methods in geometry processing were inspired by analogous method in image processing. Regularization, a common technique in image processing, did not get as much attention in geometry processing, as designing effective regularization methods for geometry processing is more complex than in the image processing domain.

Recently, regularization for geometry processing got more attention. Three recent papers [Lip12, AL13, SKPSH13] proposed non-linear regularization methods for a variety of geometry processing tasks, as artifact reduction techniques. In contrast, Martinez Esturo et al. [MRT14] demonstrated that *linear energy regularization* is beneficial for a variety of variational geometry processing methods.

Surface deformation is a common task in geometry processing. Deforming models involves interactive modeling sessions driven by a user, who deforms an object by manipulating a subset of the surface vertices. As the task is performed done in real time, it is important for deformation methods to run interactively.

Harmonic Guidance for Surface Deformation by Zayer et al. [ZRKS05], is a variational approach to surface deformation technique that can produce high quality, smooth, realistic looking deformations interactively. However, it is susceptible to artifacts near the modeling regions. In this work, I will examine a global, linear energy regularization method for harmonic surface deformations as an artifact reduction technique, eliminating the need for costly non-linear regularization.

1.1 Physically Accurate Deformations

In the physical world, objects deform due to external forces acting on them. Material properties such as elasticity, and object properties such internal structure, affect how objects deform when forces act on the object.

Objects undergo two types of deformations: elastic and plastic deformation. Elastic deformation is reversible - the objects returns to its original shape once external forces vanish. Plastic deformation is a non reversible change to the shape of an object due to the application of force (stress) on the object. In mechanics, a deformable body is an object which changes its shape while being acted upon by any external force. Clothing and skin are deformable objects which are commonly modeled in computer graphics.

Creating realistic deformations by simulating physical forces is hard. It requires modeling and simulating the internal structure of the object and solving non-linear systems of the physical forces involved. Solving non-linear systems interactively requires either a complex optimization scheme or strictly limiting the size

of the system being solved. The former is hard to implement and the latter requires using a simplified representation of the surface, which reduces the quality of the deformation.

Harmonic surface deformation is a linear method, which is fast enough to deform relatively large meshes interactively. Being a linear method, it is limited in the types of deformations that can be performed realistically, and it is prone to artifacts in the resulting surfaces. See Figure 1.1 for examples of common artifacts.

1.2 Plausible Deformations

Many deformations in computer graphics are not physically accurate. Many times, the manipulated model is a simplistic representation of a real world object. For example, many volumetric objects with complex internal structure are represented by a surface mesh. Instead of simulating physically accurate deformations, many shape manipulation methods concentrate on creating realistic-looking deformations. To a human observer, they look like a realistic deformation that a real world object could have undergone. These deformations are called *plausible* deformations. The results are not physically accurate, and might be impossible in reality, but human observers judge them as realistic.

1.3 Deformation Artifacts

Many interactive methods for surface modeling use linear energy terms to approximate the bending and stretching energies involved in deformation. Linear energies are only a rough approximation of the energies involved, and typically, these deformations suffer from artifacts when deformation are not carefully designed. Artifacts reduce the quality of the deformation, by breaking its topology and geometry in unexpected, unrealistic ways. Common artifacts include:

- **Distortion of local geometry.** Local geometric details should not be distorted by a global deformation. This is also known as preservation of shape.
- **Preservation of topology.** Unless specified otherwise, topology should stay constant.
- **Degenerate Triangles.** In some cases, the area of the triangle collapses to zero.
- **Protruding Elements.** Physical deformations minimize stretching energies. Protruding elements are caused when energies are not modeled accurately.
- **Local Self-Intersections.** Flipped Triangles - these cases are 2D specific.
- **Global Self-Intersections.** Global intersections happen easily for strong deformations, as many deformation methods contain no collision detection mechanism.
- **Volume Loss.** These artifacts commonly occur in skeleton-based deformation with skinning weights [KS12]. The volume occurs close to the joints of the skeleton when bones are rotated. Although harmonic surface deformation is not a skeleton-based deformation technique, I observed similar artifacts when deforming long limbs.

See Figure 1.1 for examples.

1.4 Enhancing Harmonic Surface Deformations

In this work, I will examine a linear approach for energy regularization for harmonic surface deformation proposed by Martinez Esturo et al. It has some novel properties that can be used to enhance the results of harmonic surface deformation with little effect on its performance:

- Linear - the energy regularization term is linear and non-iterative. The regularized deformation is solved for in a single step. This is a unique property, most regularization methods involve non-linear energies and iterative algorithms.
- Simple - the method is simple to implement, requires very little change to existing systems, and can be implemented using off-the-self optimized solvers.
- Specific - the regularization term is strongly coupled with the energy being optimized.
- Generic - can be used with different energy terms.
- Single parameter - it only introduces one new parameter. Other regularization approaches introduce several parameters.

1.5 Contributions

- I demonstrate the effectiveness of a global, linear, single parameter approach to energy regularization for harmonic surface deformation.
- I demonstrate that moderate use of energy regularization improves deformation results without introducing new artifacts.
- I demonstrate that this method is effective for shape deformations using small handle regions. Naive harmonic surface deformation, using small modeling regions, results in strong artifacts and low quality deformations.
- I demonstrate that even well designed deformations with large constrained regions benefit from some regularization.

See Figure 1.1 for example results.

1.6 Structure

The thesis is structured as follows: first, I discuss related work (Chapter 2) and introduce the definition and mathematical foundation for the rest of this work (Chapter 3). I examine harmonic surface deformation (Chapter 4) and linear energy regularization (Chapter 5) in detail. I apply linear energy regularization to harmonic surface deformations (Chapter 6). I present the results (Chapter 7), followed by discussion (Chapter 8) and conclusions and future outlook (Chapter 9).

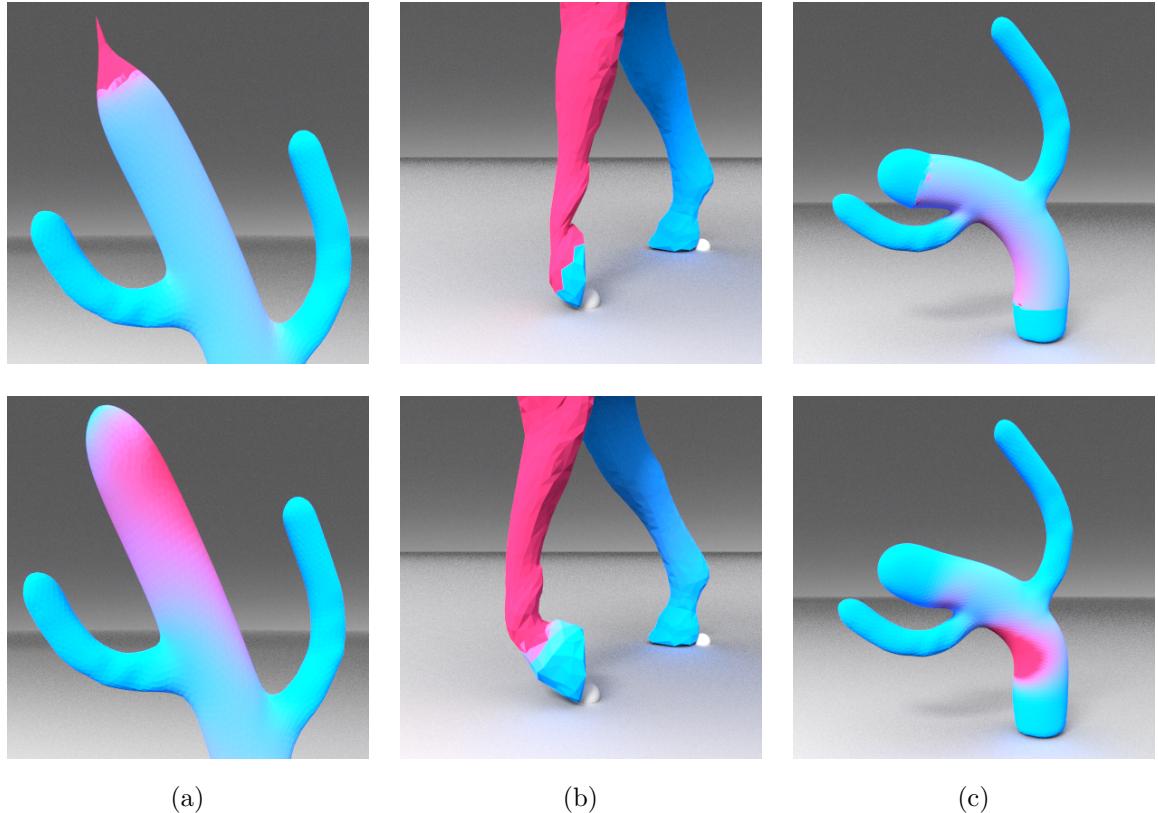


Figure 1.1: *Top row:* A selection of deformation artifacts introduced by harmonic surface deformations. (a) Protruding elements, (b) volume loss and (c) local shape distortion around constrained regions are suppressed using energy regularization. Blue colors correspond to low energies and red colors correspond to high energies. *Bottom row:* energy regularization represses these artifacts.

Chapter 2

Related Work

In this chapter I will discuss other research related to the domain of this work. The chapter is divided into five sections. Section 2.1 contains an overview of interactive shape editing. Section 2.2 discusses shape representations for interactive shape manipulation. Section 2.3 discusses Poisson methods for geometry processing. Section 2.4 discusses some of the recent work on energy regularization for geometry processing. Finally, Section 2.5 presents several approaches to rotation interpolation.

2.1 Interactive Shape Manipulation

The goal of many shape editing approaches is to interactively create meaningful deformations, while preserving surface properties such as local details and curvature. Geometric models carry shape details at many scales. Different shape manipulation methods are suitable for shape manipulation on different scales.

Many deformation methods can be divided into two categories - methods which minimize physically based energies and methods which modify differential coordinates.

Physically based deformation methods minimize bending and stretching energies. These physical energies are non-linear. Solving for accurate physical energies interactively limits the size of the system which can be solved, due to the systems being non-linear. Surface deformation usually involves calculating the thin-shell formulation of these energies [TPBF87][BPGK06].

For highly detailed surface representation, physically accurate energies are too computationally expensive for interactive shape manipulation. To overcome this, it is common to use differential surface representation and approximate physical energies by linear terms. Linear methods allow to economically perform physically plausible deformations. The surfaces created are not physically accurate, but “look” like a realistic shape deformation. However, these methods are limited in the type of operations that create realistic looking deformations. Additionally, all linear methods are susceptible to self-intersections, because surface details are not coupled. For an in-depth comparison of linear methods refer to Botsch and Sorkine [BS08].

2.2 Shape Representation

In computer graphics, surfaces are usually represented by meshes. An obvious choice for representing surface geometry is by the positions of the mesh vertices. However, this representation does not capture shape properties such as curvature. In free-form shape modification the vertex positions are manipulated directly. It is known as *shape design*, and requires more user guidance to create plausible results [SP86].

A second approach to shape representation is multi-resolution representation. The surface is decomposed into several levels of details, which can be thought of as frequency bands. The deformation is computed for

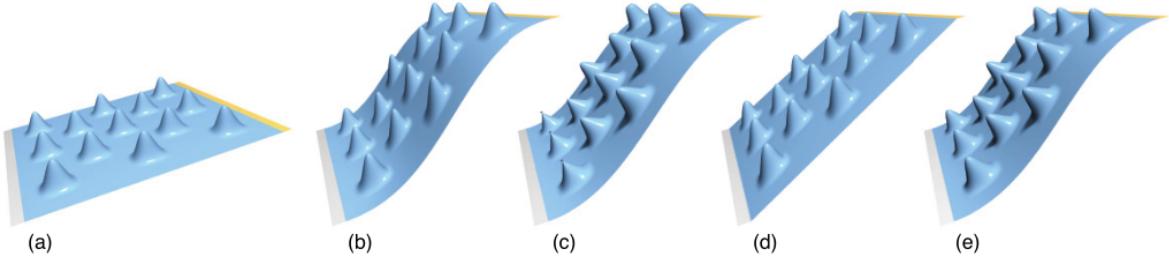


Figure 2.1: Translation only deformations: (a) the original surface is deformed using a translation only transformation. Results for various methods: (b) coarse representation with physical energies. Small geometric details are not rotated. (c) Multi-resolution techniques which represent details as surface normals creates distortion in the geometric details. (d) Gradient domain methods are translation-insensitive, fail to create plausible results. (e) Deformation transfer, a combination of multi-resolution and differential representation, creates better results. Source: Botsch et al. [BSPG06]

the coarse, or low frequency, representation of the surface, which is then propagated to finer resolutions. First uses of this approach for interactive deformations were by Zorin et al. [ZSS97] and Kobbelt et al. [KCVS98]. More recently, Botsch et al. [BSPG06] combined deformation transfer and multi-resolution representation to create detail-preserving deformations. However, many of these methods require a non-linear reconstruction step which is too slow for interactive editing.

Differential coordinates are a third approach to shape representation. This approach focuses on preserving the surface details, and is commonly used with linear energy terms. Rather than representing the surface directly by the spatial coordinates of the mesh vertices, it is represented by the differential properties of these coordinates. To deform a surface, the differential properties are modified. The new surface is solved for based on the modified representation. This is the surface representation used in this work.

Differential methods can be grouped into two classes: rotation sensitive methods and translation sensitive methods. Rotation sensitive methods use the gradients of affine transformations to construct a deformation guidance field [YZX⁺04, ZRKS05]. These methods are sensitive to rotations, but not to translations. See Figure 2.1 examples. Since translations do introduce local changes to the tangent plane of the surface, these methods are not suitable for shape deformations that involve large translations. A partial solution to this problem is to specify both a translation and a rotation. Translation sensitive methods, such as [SCOL⁺04] can handle large translations, but do not handle large rotation well, as demonstrated in [LSLCO05].

Due to using a linear model, all of these methods are suboptimal for surface deformations that involve large rotations. However, they have several desirable qualities. Many of these methods are single step, non-iterative methods, meaning all mesh coordinates are computed at once, by solving one linear system. Those linear system can be usually solved with available off-the-shelf sparse system solvers. They are suitable for deforming relatively large meshes interactively. Additionally, these methods are relatively simple to implement and preserve local surface details well.

2.3 Poisson Methods for Surface Editing

Poisson image processing is a common technique for gradient domain image manipulation. Typically, a Poisson problem is solved for a bounded domain within an image, using the surrounding pixel intensities as boundary constraints, using a user defined guidance field. Humans vision is sensitive to second order variations in intensities [PGB03] which are suppressed by Poisson methods. Poisson methods are therefore attractive for image manipulation purposes. Perez et al. [PGB03] provided a very successful framework for Poisson image editing. Common applications include patch filling and seamless cloning.

The success of Poisson techniques for image processing inspired a similar approach for geometry processing [YZX⁺04]. Poisson methods allow the user to act on a small subset of the surface vertices to create

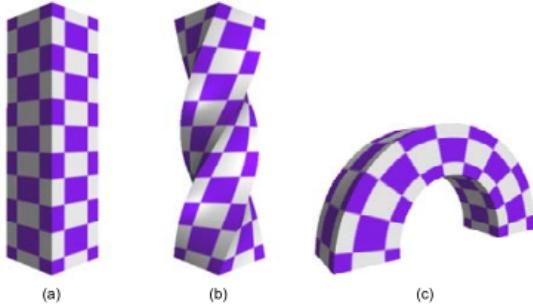


Figure 2.2: Laplacian mesh edition. (a) Original mesh. (b) Deformed mesh. (c) A second deformation. This method creates plausible and smooth deformations for well-designed constraints. Source: Yu et al. [YZX⁺04]

meaningful deformations.

Generally, Poisson methods for surface deformations map from a source to a (possibly unknown) target shape, where both shapes have the same topology. The deformation problem is reduced to finding new vertex positions (geometry) on constant domain connectivity. Vertex positions can be expressed as scalar functions over the surface domain. The problem is formulated as a linear optimization problem with a guidance field. Fixed vertices act as positional constraints that reduce the solution space to a single solution. A lot of research is focused on constructing these guidance fields to create desired deformation properties.

Yu et al. [YZX⁺04] first introduced a general framework for Poisson mesh manipulation. Their formulation allows for mesh deformations, smoothing and merging between multiple meshes, homologous to the same operations in image editing. User deformation constraints are propagated over a restricted influence region to create a guidance field. Figure 2.2 shows some of their results. In this work, deformation constraints are propagated over the entire mesh.

Zayer et al. [ZRKS05] use harmonic fields to propagate deformation constraints over the entire mesh and create a deformation guidance field. Finding new vertex position is formulated as a Poisson problem. This work introduces energy regularization into their deformation method. Their work is discussed in detail in Chapter 5.

Sumner et al. [SZGP05] learn a gradient-domain feature space of meaningful deformations from user provided examples. New deformations are created by constraining fixed regions to features from the feature space and solving a Poisson problem for new vertex positions. They achieve interactive run times for meshes of up to 10k vertices. Their approach has two drawbacks: they require the user to create up to ten example deformations, and some of their results suffer from obvious artifacts. See Figure 2.3 for an example.

Sumner and Popović [SP04] use gradient field manipulation to transfer deformations from one mesh to



Figure 2.3: Artifact examples from MeshIK [SZGP05]. *Left to right:* (1) source deformation. (2) The front leg was constrained and translated forward. (3) The red region is fixed, and the leg is translated backwards. Note the sharp change in mesh curvature around the front shoulder. (4) The tail is deformed while the rest of the mesh is fixed. The tail curvature exhibits sharp changes. Source: Sumner et al. [SZGP05]

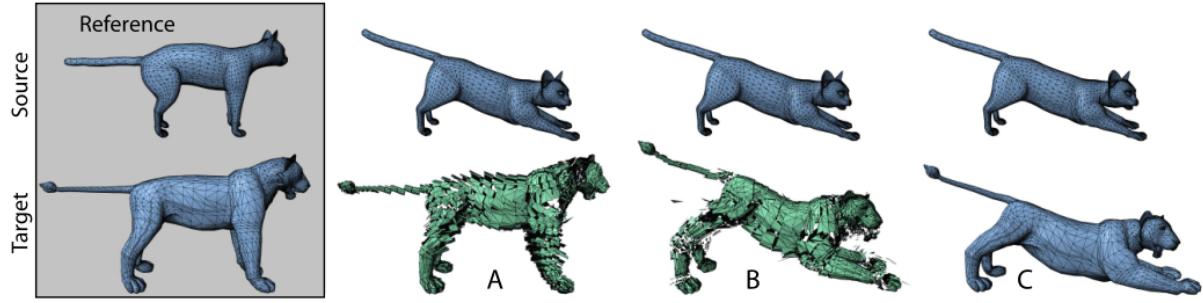


Figure 2.4: Deformation transfer between a source and a target mesh. (A) Using non translation transformation parts alone does not ensure mesh connectivity. (B) Using translational components of the deformation creates disconnected shape. (C) Using non-translational deformation components while adding vertex position singularity constraints results in a connected, deformed mesh. Source: and Popović [SP04]

another. After finding correspondences between the source and the target mesh, the target mesh gradients are transformed according to the deformation field of the source mesh. The new surface is reconstructed from the deformation guidance field. See Figure 2.4 for an overview of their method.

Poisson methods can be used for other geometry processing tasks. Kazhdan et al. [KBH06] use Poisson formulation for surface reconstruction., improving the recovery of surface details. Jones et al. [JDD03] use Poisson formulation to do one-step, feature-preserving mesh smoothing. Xu et al. [XZWB05] use Poisson formulation for gradient domain morphing between two meshes.

All Poisson methods have several properties in common. Naive Poisson methods have no mechanisms to detect surface self-intersections. The resulting mesh is usually only piecewise-linear. If the object is represented by a surface representation, there is no mechanism to preserve its volume. Solving the Poisson systems is done using a sparse linear system solver. For many methods, one of the most computationally expensive step is the construction of a guidance field.

2.3.1 ARAP

A popular approach to constructing deformation fields for Poisson problems is to use As-Rigid-As-Possible energies [ACOL00, SA07]. These energies minimize the non-rigid component of the transformation at the triangle or tetrahedral level. Rigid transformations are important for preserving surface details, which is an important property of deformation schemes that are applied to realistic representations of 3D objects.

Alexa [Ale03] first proposed this approach. Given two 2D shapes with boundaries correspondences and compatible triangulations, he creates a least-distorting morph between the shapes by locally minimize the non-rigid component of the deformation field. Since a single global transformation is too simplistic for representing the complex transformations that the body undergoes, he creates a least-distorting transformation for each triangle in the mesh independently. From these optimal per-triangle transformations, he construct a single, conforming transformation for all triangles by solving a sparse linear system.

This approach is extended to surface meshes embedded in 3D space by Sorkine and Alexa [SA07]. They suggest an non-linear, iterative scheme that optimizes a globally conforming, locally rigid transformations. Their approach creates is effective, but has two main drawbacks: although it guarantees to converge to a conforming deformation field, it might reach a sub-optimal local energy minimum, and its robustness decreases when the mesh is refined, i.e. artifacts cannot be avoided. Additionally, their optimization scheme is not efficient for large systems, where they suggest a multi-resolution approach.

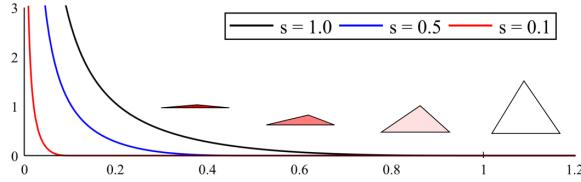


Figure 2.5: Barrier function for triangle size. The value goes to ∞ when the size of the triangle approaches zero. The barrier function affects the energy only when elements are close to being degenerate. Source: Schüller et al. [SKPSH13]

2.4 Regularization Methods for Geometry Processing

Recently, energy regularization has been explored as a method to reduce artifacts in various geometry processing tasks, such as mesh deformations and mappings. Four recent publications explored methods for distortion and artifact reductions in geometric processing, three of which propose non-linear approach to regularization, and a fourth approach, based on linear energy regularization.

Lipman [Lip12] presents a generic tool for constructing planar, orientation preserving (i.e., no triangle flips allowed) triangular mesh deformations, while limiting worst-case distortion. His method, however, is computationally complex which results in non interactive run times, and hard to implement.

Schüller et al. [SKPSH13] expand on the approach presented by Lipman [Lip12] by designing a barrier energy function (see Figure 2.5) to repress zero-area (degenerate) elements, protruding elements and flipped triangles in interactive run times. They use an iterative scheme to solve for an injective mapping to a new surface. They guarantees inversion-free deformation or mapping, . However, for larger meshes (10k vertices and above) or very strong deformation, the time until convergence significantly rises, and interactivity is lost.

Aigerman and Lipman [AL13] extend Lipman's approach [Lip12] to volumetric meshes. Their algorithm takes a deformation created by common deformation techniques and returns a similar deformation which is injective and minimizes the distortion of the mesh volumetric elements. Even for 2D meshes, their run time is not interactive.

All three methods explored the use of local energy regularization as a tool to reduce element inversion, protruding elements, and degenerate elements that might occur in a deformation. The methods presented guarantee inversion free, and in some cases, protrusion free, deformations. However, they all require solving use non-linear system which result in loss of interactivity for moderate to large meshes. Additionally, these methods are specifically designed to suppress very specific artifacts.

In this work I will use a fourth approach to regularization by Martinez Esturo et al. [MRT14]. They propose a linear, energy regularization method which is generic, and suitable for many variational geometry processing tasks.

It is the only approach that is suitable for regularization of surface meshes embedded in \mathbb{R}^3 . The methods mentioned above are only suitable for planar meshes in \mathbb{R}^2 or volumetric meshes in \mathbb{R}^3 . However, linear energy regularization cannot guarantee artifact free deformations, in constant to non-linear methods. These non-linear methods can guarantee results which are free from specific artifacts. Linear energy regularization is presented in further detail in Chapter 5.

2.5 Rotation Interpolation

To create a deformation guidance field, I will need to interpolate between rotations from multiple deformation handles. Interpolating between multiple rotations is not a trivial task, due to the non-linear nature of rotations. Several attempts have been made to streamline this task.

In computer graphics, spherical linear interpolation (SLERP) [Sho85] (see Section 3.4.4) is the standard solution for interpolating between two rotations. But there is no standard methods for interpolating between more than two rotations. Alexa [Ale02] proposes a method for constructing commuting linear combinations of rotation matrices using new operator definitions that rely on matrix roots. His work is dissected and concluded to be incorrect by Bloom et al. [BBM04].

In this work, I follow the approach used by Zayer et al. in *Harmonic guidance for surface deformation* [ZRKS05], who use two different strategies for rotation interpolation. For interpolating between two rotations only, SLERP is used. When more than two rotations are interpolated, rotations are represented as quaternions and each component is interpolated independently by solving for a harmonic scalar field. This interpolates between the quaternion smoothly.

Chapter 3

Definitions and Mathematical Theory

This chapter is discusses the mathematical foundation for this work. Section 3.1 introduces the terms used in the rest of this work. Section 3.2 contains a short overview of the differential equations and their properties. In Section 3.3 contains the derivation of the discrete differential operators used. Section 3.4 discusses various properties of transformations and rotations representation. Finally, Section 3.5 introduces the error measures which will be used to analyze the results.

3.1 Definitions

3.1.1 Triangle Mesh

Definition. *Vertex*

A vertex is defined as a 3-dimensional position vector with coordinates $v_i = (x, y, z)^\top \in \mathbb{R}^3$. \mathcal{V} is the set of mesh vertices.

Definition. *Edge*

An edge connects two vertices $e_i = (v_j, v_k) \in \mathcal{V} \times \mathcal{V}$. The set of mesh edges is $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$.

Definition. *Vertex neighborhood*

A vertex neighborhood \mathcal{N}_i is a one-ring of neighbor vertices around the vertex v_i . See Figure 3.1.

Definition. *Face*

A face t_i is a closed set of edges, which does not contain any internal vertices. The set of mesh faces is \mathcal{T} .

Definition. *Mesh*

A mesh is a collection of vertices and faces $\mathcal{S} = (\mathcal{V}, \mathcal{T})$. Sometimes it is useful to represent a mesh by a collection of edges as well, in that case $\mathcal{S} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$.

Definition. *Triangle mesh*

A collection of triangles.

Definition. *2-manifold mesh (with boundary)*

A 2-manifold surface is locally homeomorphic to disc at any point on the surface, or to half disc at boundaries.

A triangle mesh is a 2-manifold if:

- The surface does not self-intersect
- The surface contains no non-manifold edges.
- The surface contains no non-manifold vertices.
- The surface is connected.

Definition. *Manifold edge*

A manifold edge is incident to exactly two faces (one or two faces in the case of meshes with boundary). See Figure 3.2.

Definition. *Non-manifold vertex*

A non-manifold vertex is created when two or more surface sheet meet in a single vertex. See Figure 3.2.

A triangular manifold mesh is defined on *discrete* vertices, but the resulting surface is continuous, consisting of piecewise linear parametrization functions. See [BKP⁺10]. The rest of this work assumes connected, 2-manifold triangle meshes embedded in \mathbb{R}^3 .

3.1.2 Deformations and Constraints

Definition. *Boundary edge*

A boundary edge is incident to exactly one face.

Definition. *Interior edge*

An interior edge is incident to exactly two face.

Definition. *Handle region*

A handle region is a vertex or a set of connected vertices that is manipulated by the user to create a deformation constraint. All vertices which belong to the same handle region transform rigidly.

Definition. *Fixed vertex*

A fixed vertex is a vertex whose position acts as a hard constraint during mesh deformations. To deform a mesh, at least one handle vertex and one fixed vertex are required.

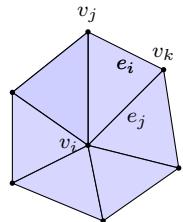


Figure 3.1: A one-ring neighborhood around the vertex v_i . e_i is a boundary edge, e_j is an interior edge.

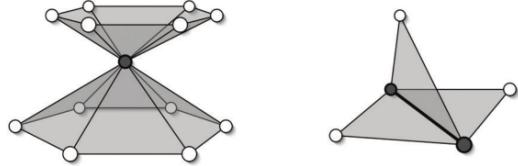


Figure 3.2: Non manifold meshes. *Left to right:* non-manifold vertex, non-manifold edge. Source: Botsch et al. [BKP⁺10]

3.2 Harmonic Functions

3.2.1 Laplace Operator

The Laplace operator is a differential operator given by the divergence of the gradient of a function [Haz02]:

$$\Delta f = \nabla^2 f = \nabla \cdot \nabla f \quad (3.1)$$

where f is a scalar function. It is a linear operator. In Euclidean space, which can be written as:

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} \quad (3.2)$$

3.2.2 Poisson's Equation

The Poisson equation is a partial elliptic differential equation of elliptic type [PR05].

$$\Delta f = \phi \quad (3.3)$$

where ϕ is a function on the domain. It is used broadly in physics and mechanics to describe a variety of phenomena such as the relationship between electrostatic field to a charge density or gravitational potential field to a mass density. It is also commonly used to model diffusion processes.

A scalar function on a bounded domain is uniquely defined by its values on the boundary of the domain and its Laplacian in the domain's interior. Given the value of the Laplace operator ϕ in the domain interior and the value of the function on the domain boundary, the Poisson equation has a unique solution [PR05].

In image processing, the Poisson equation and the related Laplace operator are used to suppress the second-order variations in the reconstructed image patch [PGB03]. The Poisson equation is used with either von-Neumann boundary conditions or with Dirichlet conditions on an arbitrary patch outline. In geometry processing, it is commonly used with Dirichlet boundary conditions.

3.2.3 The Laplace Equation

The Laplace equation is a special case of the Poisson equation :

$$\Delta f = 0 \quad (3.4)$$

It is used to describe physical phenomena such as the steady state solution to the heat equation or flow of incompressible fluids. The solutions to the Laplace equations are harmonic functions which have special properties [PR05].

3.2.4 Harmonic Functions

A real valued function f defined within a domain in Euclidean space is a harmonic function if it satisfies the Laplace equation:

$$\Delta f = 0 \quad (3.5)$$

Harmonic functions have divergence-free gradient fields [PR05]. Later in this work they are used to create a transformation propagation field on the mesh domain. Harmonics functions have desirable mathematical properties for the purpose of creating deformation guidance fields:

Regular Harmonic functions are infinitely differentiable.

The Maximum (Minimum) Property Let $f(x, y)$ be a harmonic function on a connected, planar domain D . The maximum (minimum) of $f(x, y)$ in D is achieved on the boundary ∂D .

Intuitively, in a bounded domain, a harmonic function will receive its maximum and minimum values on the boundary of that domain.

Mean Value Property Let $f(x, y)$ be a harmonic function on an open, planar set D , and let (x_0, y_0) be a point in D . For B_R , a disk of radius $r > 0$ fully contained in D , centered at (x_0, y_0) , $f(x_0, y_0)$ is the average of the value of f on ∂B_R

$$f(x_0, y_0) = \frac{1}{2\pi r} \int_{\partial B_R} f(s) ds \quad (3.6)$$

Intuitively, this means that a harmonic function does not create local minima or local maxima within a bounded domain.

The deformation guidance field is created by using the values of the harmonic field to interpolate between the deformation constraints set by the user, specifically, the transformations of the modeling handles. This means that the interpolated transformations create a smooth field, which contains no local extrema, and most importantly, restricts the maximal magnitude of the interpolated transformations to values between the user set deformation constraints.

3.3 Discrete Operators for 2-Manifold Meshes

Surface meshes can be created as synthetic models or acquired from physical objects using various techniques such as 3D scanners [CL96, TL94], shape from shading techniques [JK07] or using marching cubes on volumetric data [LC87]. The vertex connectivity describes the topology of the shape, and the vertex positions describe the geometry of the shape.

A manifold triangle mesh is a piecewise linear discretization of a continuous surface. Using differential operators on meshes, which are irregularly connected, requires using a generalization for Riemannian manifolds. The continuous gradient operator is generalized to the discrete gradient operator, and the Laplace operator is generalized to the Laplace-Beltrami operator.

To do this, the mesh surface is expressed as a linear combination of discrete vector fields, then the gradient and divergence operator are redefined for these fields.

3.3.1 Discrete Surface Formulation

Let $= (,)$ be a 2-manifold triangle mesh embedded in \mathbb{R}^3 . Let (u, v) be parameters on the mesh domain. The piecewise linear coordinate function $\mathbf{f}(u, v)$ is defined by barycentric interpolation of vertex coordinates function:

$$\mathbf{f}(u, v) = \sum_{i=1}^{|\mathcal{V}|} \mathbf{f}_i \phi_i(u, v) \quad (3.7)$$

where ϕ_i is a piecewise linear hat basis function, such that $\phi_i(v_k) = \delta_{ik}$ for the set of mesh vertices $(v_1 \dots v_{|\mathcal{V}|}) = \mathcal{V}$ and $\mathbf{f}_i = \mathbf{f}(v_i)$. Then $\mathbf{f}(u, v)$ is the linear combination of unique basis functions [BSPG06].

3.3.2 Discrete Gradient Operator

Let $\nabla \mathbf{f}$ be the continuous gradient operator of the coordinate function \mathbf{f} .

$$\nabla \mathbf{f}(u, v) = \sum_{i=1}^{|\mathcal{V}|} \mathbf{f}_i \nabla \phi_i(u, v) \quad (3.8)$$

Let (v_i, v_j, v_k) be vertices belonging to the same triangle t_l , and $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$ be their coordinates vectors. The vertices are represented by the basis functions (ϕ_i, ϕ_j, ϕ_k) respectively. The triangle gradients of

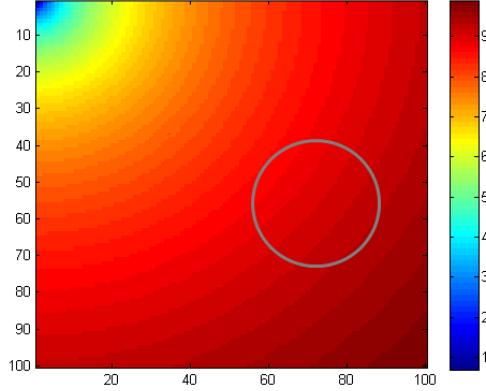


Figure 3.3: Harmonic function. For a bounded domain (gray circle) the function receives its minimum and maximum values on the domain boundary.

the basis functions $(\nabla\phi_i^l, \nabla\phi_j^l, \nabla\phi_k^l)$ are constant for each triangle t_l . To ensure that the gradients are coplanar with the triangle, they are defined as perpendicular to the face unit normal \mathbf{n} . The basis functions gradients are found by solving the linear system [BS08]:

$$(\nabla\phi_i^l, \nabla\phi_j^l, \nabla\phi_k^l) = \begin{pmatrix} (\mathbf{x}_i - \mathbf{x}_k)^T \\ (\mathbf{x}_j - \mathbf{x}_k)^T \\ \mathbf{n}^T \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{pmatrix} \quad (3.9)$$

In the case of a 2-manifold embedded in \mathbb{R}^3 , this yields a 3×3 triangle constant gradient operator $\mathbf{G}_l = (\nabla\phi_i^l, \nabla\phi_j^l, \nabla\phi_k^l)$ for each mesh triangle.

The triangle constant gradients \mathbf{G}_l can be inserted into a matrix \mathbf{G} with dimensions of $3|\mathcal{T}| \times |\mathcal{V}|$, such that the operation is formulated as a global gradient operator $\mathbf{G}\mathbf{x} = \mathbf{y}$ on a vector of the stacked vertex positions \mathbf{x} of dimensions $|\mathcal{V}| \times 3$. The resulting vector \mathbf{y} is a $3|\mathcal{T}| \times 3$ vector of all stacked triangle gradients [MSRT13].

3.3.3 Discrete Laplace-Beltrami Operator

The Laplace-Beltrami operator is a generalization of the Laplace operator to Riemannian manifolds. It is defined as the divergence of the gradient of the function, and it is linear. The continuous formulation of the Laplace operator is:

$$\Delta \mathbf{f} = \operatorname{div} \operatorname{grad} \mathbf{f} \quad (3.10)$$

The discrete divergence of the piecewise constant gradient field on the mesh yield a discrete Laplace operator:

$$\Delta \mathbf{f}(v_i) = \operatorname{div}(\nabla \mathbf{f})(v_i) \quad (3.11)$$

$$= \sum_{t_l \in \mathcal{T}_i} \operatorname{area}(t_l) (\nabla\phi_i|_{t_l})^T \mathbf{G}_l \quad (3.12)$$

where \mathcal{T}_i is the set of triangles incident to the vertex v_i [BSPG06]. This can be expressed as a global operator on the mesh:

$$\mathbf{G}^\top \mathbf{A} \mathbf{G} = \mathbf{L} \quad (3.13)$$

where \mathbf{A} is a $3|\mathcal{T}| \times 3|\mathcal{T}|$ diagonal matrix with the triangle areas. This is equivalent to the standard formulation of the cotangent discretization of the Laplace-Beltrami operator [BS08].

3.4 Transformations

In this work, I will deal with two types of transformations: translations and rotations. Both of these transformations are affine and preserve collinearity. Other affine transformations include scaling and shearing. Reflecting transformation creates artifacts and are therefore disallowed.

3.4.1 Transformation Matrices

In this work, the triangle mesh is embedded in \mathbb{R}^3 . A point in \mathbb{R}^3 is represented by coordinates $\mathbf{P} = (x, y, z)^\top$. In homogeneous coordinates, it is represented by a 4-vector $\mathbf{P} = (x, y, z, 1)^\top$. In homogeneous coordinates, transformations are represented by a 4×4 matrix. Transformation matrices are formulated as follows:

Translation by vector $\Delta\mathbf{x} = (\Delta x, \Delta y, \Delta z)^\top$ is expressed by the matrix:

$$\mathbf{T}_T = \begin{pmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Translation are non-ambiguous. Only a single translation exists that transforms $\mathbf{P} \rightarrow \mathbf{P}' = \mathbf{P} + \Delta\mathbf{x}$.

A rotation is defined by an axis and an angle. Traditionally, they are represented by rotation matrices around the x , y , and z axis. The order of applying rotational matrices is important as rotations do not commute.

$$\mathbf{T}_{R_x} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x & 0 \\ 0 & -\sin \theta_x & \cos \theta_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{T}_{R_y} = \begin{pmatrix} \cos \theta_y & 0 & -\sin \theta_y & 0 \\ 0 & 1 & 0 & 0 \\ \sin \theta_y & 0 & \cos \theta_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{T}_{R_z} = \begin{pmatrix} \cos \theta_z & -\sin \theta_z & 0 & 0 \\ \sin \theta_z & \cos \theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(A 2D formulation of rotation matrices is illustrated in Figure 3.4.) Representing rotations by matrices makes it hard to manipulate and interpolate between rotations. Alternatives to the matrix representation include Euler angles and quaternions. Quaternions are discussed in Section 3.4.3.

Scaling \mathbf{T}_S and shearing \mathbf{T}_{Sh} matrices are easier to represent and manipulate in matrix representation:

$$\mathbf{T}_S = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \mathbf{T}_{Sh} = \begin{pmatrix} 1 & sh_{xy} & sh_{xz} & 0 \\ sh_{yx} & 1 & sh_{yz} & 0 \\ sh_{zx} & sh_{zy} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

A complete transformation matrix is created by multiplying together different transformation matrices.

A xkcd comic strip featuring a mathematical equation. The equation shows a 2x2 matrix with cosines and sines of 90 degrees multiplied by a vector [a1, a2] resulting in a vector [0, 0]. The matrix is:

$$\begin{bmatrix} \cos 90^\circ & \sin 90^\circ \\ -\sin 90^\circ & \cos 90^\circ \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Figure 3.4: 2D rotation matrices. Source: xkcd.com

3.4.2 Transformation Decomposition

To create a deformation guidance field, transformations need to be propagated from the handle regions to the rest of the mesh. The propagation step involves interpolating between different general transformations, which may contain both translational, rotational, scaling and shearing components.

Scaling transformations can be interpolated linearly, but rotations are non-commuting and nonlinear, and therefore require a different interpolation method.

Given a general transformation matrix recovered from the user interaction, it needs to be decomposed into different components that can be manipulated separately. Decomposing the transformation matrix into semantically meaningful transformations allows better control of the transformation propagation on the mesh. Extracting the translational component is simple, and this results in 3×3 transformation matrix and a translation only 3-vector, but separating the scaling and rotational components requires matrix decomposition.

Shoemake and Duff [SD92] analyze different matrix decomposition schemes for transformation matrices. They argue that polar decomposition is the best way to decompose a matrix into meaningful components. Let \mathbf{T} be a general transformation matrix. The polar decomposition of \mathbf{T} is $\mathbf{T} = \mathbf{QS}$, where \mathbf{Q} is an orthogonal factor and \mathbf{S} is a symmetric positive definite factor. This decomposition can be done using e.g. a Newton algorithm. \mathbf{Q} is the closest orthogonal matrix to the transformation matrix \mathbf{T} in the least square sense, given by the Frobenius norm:

$$\|\mathbf{Q} - \mathbf{T}\|_F^2 = \sum_{ij} (Q_{ij} - T_{ij})^2 \quad (3.14)$$

When \mathbf{T} has a positive determinant, \mathbf{Q} will be pure rotation and \mathbf{S} is a scaling matrix, which is diagonal in some coordinate system. Figure 3.5 visualizes this physical interpretation of the decomposition. Given the need to use different interpolation schemes for scaling and rotation operations, the choice for this decomposition is obvious.

After the decomposition, the transformation can be interpolated independently using the harmonic field over the mesh domain. The scaling component can be interpolated linearly, but for interpolating between rotations quaternions are needed.

3.4.3 Rotation Representation Using Quaternions

Propagating deformation constraints from multiple handle regions requires to interpolate between two or more rotations. Ideally, one could interpolate between rotations matrices or Euler angles linearly $\mathbf{R}_t = \mathbf{T}_f = (1-t)\mathbf{R}_1 + t\mathbf{R}_2$. But the rotation space is not linear, making this interpolation incorrect. Additionally, Matrix and Euler angles representation of rotations are hard to manipulate.

A quaternion is a four dimensional complex number. Originally, quaternions were developed as a generalization of the complex number system to three dimensions, to represent rotations in space. The generalization was proved as not closed to multiplication, requiring an extension to a fourth dimension.

A quaternion $\mathbf{q} \in \mathbb{R}^4$, which is a four dimensional vector space over real numbers. The basis functions of this space are denoted $\mathbf{i}, \mathbf{j}, \mathbf{k}$. The quaternion $\mathbf{q} = (q_w; q_x, q_y, q_z) = q_w + q_x \cdot \mathbf{i} + q_y \cdot \mathbf{j} + q_z \cdot \mathbf{k}$ has a scalar

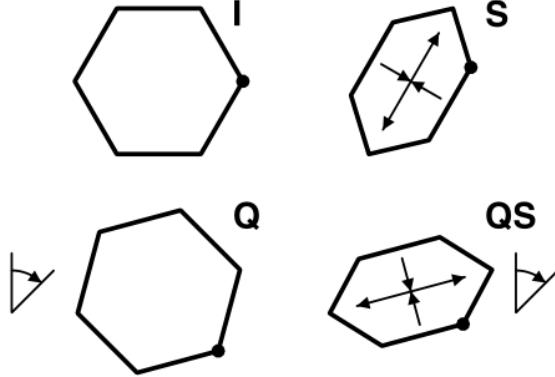


Figure 3.5: Polar decomposition for transformation matrices. **I** is the identity transformation. **Q** is the rotational component. **S** is a scaling matrix, in some orthogonal coordinate system. The transformation **QS** rotates and scales the vertices. Source: Shoemake and Duff [SD92]

component q_w which denotes the angle of rotation and a vector component, $\mathbf{q}_v = (q_x, q_y, q_z)^\top$, which defines the rotation axis. This is a more natural way to represent rotations than Euler angles, which require defining three different angles of rotation applied in a specific order.

Any three-dimensional rotation around a unit vector \mathbf{q}_v can be represented by a unit quaternion. Intuitively, this means that any combination of rotations around unit axis can be represented by a single quaternion. This is a more intuitive approach than other representations. Another benefit for using quaternions is that there exist a straightforward method to interpolate between quaternions, namely SLERP.

3.4.4 SLERP

Spherical Linear Interpolation was introduced into computer graphics by Ken Shoemake in 1985 [Sho85]. Unit quaternions define a sphere in four dimensional space. There is an infinite number of paths that transition from one quaternion to another. Using linear interpolation creates a path in this space, but it is not the straightest or shortest path. Only one such path exists for any two quaternions.

SLERP has three parameters: two quaternions $\mathbf{q}_0, \mathbf{q}_1$ and a path parameter where $t \in [0, 1]$. Ω is the angle subtended by the arc between the two quaternions on the unit sphere:

$$\text{SLERP}(\mathbf{q}_0, \mathbf{q}_1, t) = \frac{\sin[(1-t)\Omega]}{\sin \Omega} \mathbf{q}_0 + \frac{\sin[t\Omega]}{\sin \Omega} \mathbf{q}_1. \quad (3.15)$$

SLERP interpolation creates a geodesic path [DKL98] between the two quaternions, which guarantees constant speed and shortest path interpolation between the quaternions along this path. Using SLERP to interpolate between the rotation on the handle region and the identity rotation on the fixed region ensures the creation of a smooth deformation guidance field.

3.5 Evaluating Surface Deformations

To evaluate the effects of linear energy regularization on surface deformations I will use both qualitative and quantitative analysis. Although the aim of this work is to create plausible deformations, a qualitative property which cannot be measured directly, quantitative evaluation can give useful insight into the effects of energy regularization on the deformations.

The local shape of a surface can be described by a 2×2 matrix known as the first fundamental form. Given a surface which is a continuous function $\mathbf{f} : \Omega \rightarrow \mathbb{R}^3$ where Ω is a domain parametrized by (u, v) , the first fundamental form is given by:

$$\mathcal{I}_I = \begin{pmatrix} f_u \cdot f_u & f_u \cdot f_v \\ f_v \cdot f_u & f_v \cdot f_v \end{pmatrix} \quad (3.16)$$

where f_u, f_v are partial derivatives.

Similarly, a deformation is an affine mapping between a source surface \mathcal{S} to a target surface \mathcal{S}' , $\Phi(\mathbf{x}) : \mathcal{S} \rightarrow \mathcal{S}' \in \mathbb{R}^3$. The deformation gradient tensor field is defined as $\nabla\Phi = (\Phi_u, \Phi_v)$. This tensor field can be used to express the first fundamental form [FH05][HLS⁺07][Mar13]:

$$\mathcal{I}_I = \nabla\Phi^\top \nabla\Phi \quad (3.17)$$

Discrete Formulation

Given a source mesh \mathcal{S} and a target mesh \mathcal{S}' embedded in \mathbb{R}^3 , containing the same number of triangles with the same connectivity, the *deformation gradient* of the triangle \mathbf{t}_i is given by the Jacobian matrix \mathbf{J}_i of the affine mapping Φ of the triangle $\mathbf{t}_i \in \mathcal{S} \rightarrow \mathbf{t}'_i \in \mathcal{S}'$:

$$\Phi_i(\mathbf{t}_i) = \mathbf{T}_i \mathbf{t}_i + \Delta\mathbf{x}_i \quad (3.18)$$

where \mathbf{t}_i is a 3×3 matrix containing the triangle vertices positions $(\mathbf{x}'_i, \mathbf{x}'_j, \mathbf{x}'_k)$ of the target mesh, \mathbf{T}_i is a 3×3 transformation matrix containing the rotational, scaling and shearing components of the mapping, and $\Delta\mathbf{x}_i$ is the translational component of the mapping [SZGP05]. This formulation assumes that the mapping is between two continuous surfaces which is an injection (no two points on the source surface are mapped to the same point on the target surface).

The per-triangle affine transformation \mathbf{T}_i is linear in the triangle vertex position. Thus, the Jacobian \mathbf{J}_i of the per-triangle mapping $\mathbf{J}_i(\Phi_i) = \mathbf{T}_i$. This Jacobian can be calculated directly using the mesh discrete gradient operator [Mar13]:

$$\mathbf{J}(\Phi) = \mathbf{G}\mathbf{x}' \quad (3.19)$$

where \mathbf{G} is the discrete gradient operator on the source surface, and \mathbf{x}' is the deformed surface vertex coordinates vector. This creates a vector of stacked 3×3 matrices of per-triangle deformation tensors \mathbf{J} .

Metric Distortion Error Measurements

Analyzing the first fundamental form can give insight into the changes of the local surface parametrization. The eigenvalues λ_1, λ_2 of \mathcal{I}_I are used to analyze the distortion introduced by the mapping from the source to target mesh surface. Since the first fundamental form can be expressed as the deformation gradient, there is also a relationship between their eigenvalues. Therefor it is sufficient to analyze the per-triangle deformation tensor \mathbf{J}_i and use the relation between the singular values of \mathbf{J}_i and the eigenvalues of \mathcal{I}_I of the mapping between the triangles.

The relationship between the transformation gradient and the first fundamental form as depicted in Equation 3.17 exists also for their discrete counterparts [Mar13]. Therefor, it is sufficient to analyze the discrete formulation of the deformation gradient to define the discrete quantitative error measurements on the mesh. The eigenvalues of \mathcal{I}_I are used to analyze the properties of the deformation.

To measure the isometric and conformal errors of the mapping, each per-triangle deformation gradient tensor \mathbf{J}_i can be singular value decomposed: $\mathbf{J}_i = \mathbf{U}\Sigma\mathbf{V}^\top$ where Σ is a 3×3 matrix with entries: $\sigma_1, \sigma_2, \sigma_3$. The third eigenvalue σ_3 should always be degenerate, since the surface is a 2-manifold.

\mathbf{J}_i is assumed to be orientation preserving, \mathbf{U} and \mathbf{V}^\top are purely rotational matrices with no reflection. Intuitively, the rotation \mathbf{V} aligns (u, v) with the original patch parameterization. σ_1, σ_2 are scaling factors, which scale the parameterized patch to match the new surface. The rotation \mathbf{U} rotates the parametrized patch into the new triangle.

The singular values of \mathbf{J}_i are the square roots of the eigenvalues of \mathcal{I}_I :

$$\sigma_i = \sqrt{\lambda_i} \quad (3.20)$$

From this relation, it is possible to derive the relation between the properties of the mapping and its singular values:

$$\begin{aligned} \Phi \text{ is isometric} &\iff \mathcal{I}_I = \mathbb{I} & \iff \sigma_1 = \sigma_2 = 1 &\iff \lambda_1 = \lambda_2 = 1 \\ \Phi \text{ is conformal} &\iff \mathcal{I}_I = s \cdot \mathbb{I} & \iff \sigma_1 = \sigma_2 &\iff \lambda_1 = \lambda_2 \\ \Phi \text{ is equiareal} &\iff \det(\mathcal{I}_I) = \mathbb{I} & \iff \sigma_1 \sigma_2 = 1 &\iff \lambda_1 \lambda_2 = 1 \end{aligned}$$

Now that the link between the deformation gradient affect the parametrization of the triangle domain is explained, it is possible to define error measurements:

Isometric Error The isometric error measures the change in the lengths of the surface. For a 2-manifold embedded in 3D space, the per-triangle isometric error is:

$$e_{\text{iso}}^i = (\sigma_1 - 1)^2 + (\sigma_2 - 1)^2 \quad (3.21)$$

The total normalized deformation isometric error is the integration of the isometric error over the surface of the mesh. Since the error term is triangle constant, this is equivalent to multiplying e_{iso}^i by the area of the triangle t_i :

$$E_{\text{iso}} = \sum_{i=1}^{|\mathcal{T}|} e_{\text{iso}}^i \cdot \text{area}(t_i) \quad (3.22)$$

This error measurement is enough to compare the total isometric error for different deformations of the same source mesh. To compare between different meshes, it needs to be normalized relative to the original surface area \mathcal{S} :

$$E_{\text{iso}}^N = \frac{E_{\text{iso}}}{\text{area}(\mathcal{S})} \quad (3.23)$$

Conformal Error The conformal error measures the changes in angles for the surface patch. This is related to relative factor change between σ_1 and σ_2 . Scaling them by different values distort the angles on the parametrized triangle surface. The per-triangle conformal errors is:

$$e_{\text{conf}}^i = \frac{1}{2}(\sigma_1 - \sigma_2)^2 \quad (3.24)$$

The total conformal error and the normalized conformal error are defined in the same fashion as their isometric error counterparts:

$$E_{\text{conf}} = \sum_{i=1}^{|\mathcal{T}|} e_{\text{conf}}^i \cdot \text{area}(t_i) \quad (3.25)$$

$$E_{\text{conf}}^N = \frac{E_{\text{conf}}}{\text{area}(\mathcal{S})} \quad (3.26)$$

Total Normalized Area Change The total normalized change between the source and target shapes' surface area is:

$$E_{\text{area}}^N = \frac{\text{area}(\mathcal{S}') - \text{area}(\mathcal{S})}{\text{area}(\mathcal{S})} \quad (3.27)$$

Looking at the normalized change can be deceiving, because the total surface area can only shrink by 100%, but expand without limitations. Nonetheless, this error measure is easier to analyze than the absolute change in surface area.

Chapter 4

Harmonic Surface Deformation

Harmonic surface deformation is a gradient-manipulation surface deformation scheme. It is a linear deformation technique which combines differential surface representation, gradient domain shape editing and harmonic fields for transformation propagation to create smooth, physically plausible deformations.

The deformation process has three distinct stages: creating the differential representation of the input surface, modifying this representation according to the deformation constraints and reconstructing a new surface from the modified differential representation.

Harmonic surface manipulation was proposed by Zayer et al. [ZRKS05] The novelty of this method is to use a harmonic field to propagate deformation constraints through the mesh, thereby creating smooth deformations. It has certain properties that are associated with a larger class of deformation methods.

4.1 Differential Surface Representation

The surface is represented by its differential properties. The aim of this representation is to create a smooth deformation which preserves these differential properties.

Harmonic surface deformation represents the surface using its differential properties. Manipulating the surface is done in the gradient domain.

4.2 Linear Methods

Harmonic surface deformation is a linear deformation method. The deformation is linear in a sense that the energies on the mesh are quadratic, and the main deformation step is to solve a linear system. This linear system is the minimizer of a global quadratic variational minimization problem, which is formulated as an energy functional. The term energy indicates a scalar quantity which measures some property of the deformation on each point in the domain and is integrated over the domain.

Linear methods have favorable properties. First, the variational problem has a unique global minimum given proper boundary conditions [BS08].

Second, many optimized solvers are available, easing the implementation. In many cases, the linear system has to be factored only once, and can be reuse for different deformations of the same mesh. This allows the deformation to be done interactively for large meshes. Moreover, all vertex positions are solved for in a single step for each dimension.

Finally, the global minimization step guarantees at least C^0 smoothness. Depending on the method, this does not prevent artifacts such as protruding triangles or distortion of small details.

4.3 Gradient Domain Mesh Manipulation

Gradient editing for mesh deformation was inspired by Poisson methods for image processing [YZX⁺04]. Deformation constraints are propagated through the mesh and are used to locally modify the mesh gradients.

Deformation constraints are set by the user on a small set of fixed and handle vertices. Fixed vertices' positions remain unchanged throughout the deformation. All vertices in a single handle region transform uniformly. The vertices can be translated, rotated and scaled. They act as the boundary conditions.

The rotation and scaling transformations are propagated on the mesh, defining local transformations of the mesh gradients. The new gradient field acts as a *guidance field* for the deformation, under the positional constraints from the fixed and handle vertices.

This is formulated as an energy minimization problem for the vertex positions. The deformation energy functional is minimized in the least-squares sense:

$$\mathbf{x}' = \underset{\mathbf{x} \in \mathbb{R}^{3|\mathcal{V}|}}{\operatorname{argmin}} E(\mathbf{x}) \quad (4.1)$$

$$E(\mathbf{x}) = \int_{\Omega} \|\nabla \mathbf{x} - \mathbf{g}\|^2 d\mathbf{x} \quad (4.2)$$

for Ω the mesh domain, \mathbf{x} mesh vertex positions, \mathbf{x}' new vertex positions and \mathbf{g} deformation guidance field.

4.4 Transformation Propagation using Harmonic Fields

Many deformation methods differ by the approach to creating the deformation guidance field. Harmonic surface deformation use harmonic fields to propagate the local gradient transformations on the mesh. Other methods set an influence region for each handle [YZX⁺04].

The harmonic field h is used to interpolate between the deformation constraints on the mesh. Harmonic functions fulfill the steady-state discrete heat $\Delta h = 0$. On the mesh domain, its discrete formulation is $\mathbf{L}h = \mathbf{0}$, where \mathbf{L} is the discrete Laplace-Beltrami operator. Handle vertices are treated as sources, with a value of 1. Fixed vertices are treated as sinks, with a value of 0. The field is solved on the mesh vertices, its value is interpolated at the triangle centers. The solution is computed in a single step.

For harmonic surface deformation with no energy regularization, the factorization of the Laplace-Beltrami operator is done once, and is used for solving both the harmonic field and for the new vertex positions. Solving for the harmonic field is a cheap operation, in a sense that it only requires solving the same system with a different guidance field. Reusing this factorization reduces the complexity of the deformation process, as only a single system is required setup.

Using a harmonic field for transformation propagation ensures smooth interpolation between rotation and scaling imposed by the handle regions and the identity transformation imposed on the fixed vertices.

Properties of the harmonic field, such as the minimum-maximum property and the mean value properties, ensure smoothness of the deformation guidance field, and prevent the creation of new local and global extrema. Consequently, the interpolated local transformation for the guidance are limited to the interval whose endpoints are the deformation handles. See Figure 4.1 for visualization of the harmonic field on a mesh.

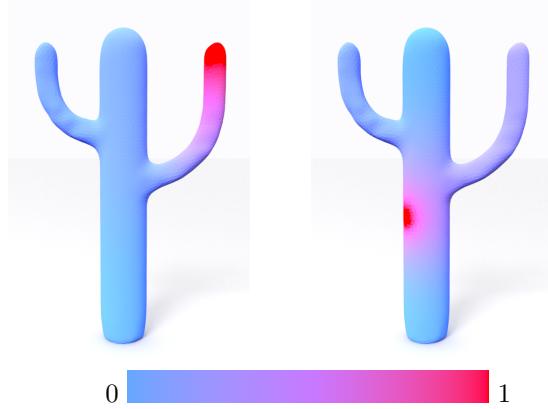


Figure 4.1: Harmonic fields on a mesh. Within each triangle, the values are interpolated linearly. The values of the harmonic field are limited to the $(0, 1)$ range. The colors indicate the value of the harmonic field. *Left:* the base of the cactus is fixed, a single handle is specified on the right spike. *Right:* the base and top of the cactus are constrained, a few vertices are set as a deformation handle on the cactus' trunk.

4.5 Solving for Vertex Positions

To minimize the guidance field energy functional [4.2], it is sufficient to solve the corresponding Euler-Lagrange equations of the problem. This leads to a second Poisson problem:

$$\Delta \mathbf{x} = \nabla^T \mathbf{g} \quad (4.3)$$

The discretization of the energy functional in Equation 4.2 is the integrated squared difference in the divergence of the new gradient field and the deformation guidance field. Since the mesh gradients are defined to be triangle-constant, the discretization of this integral is straightforward. It is an area-weighted sum of the quadratic difference:

$$E(\mathbf{x}) = \|\mathbf{Gx} - \mathbf{g}\|_{\mathbf{A}}^2 \quad (4.4)$$

$$= (\mathbf{Gx} - \mathbf{g})^T \mathbf{A} (\mathbf{Gx} - \mathbf{g}) \quad (4.5)$$

where \mathbf{G} is the discrete gradient operator, \mathbf{g} is the discrete deformation guidance field, and \mathbf{A} is the triangle-area weighted norm, as defined for Equation 3.13. Differentiating the energy functional w.r.t. the deformed coordinates leads to the equation:

$$\frac{d}{dx} E(\mathbf{x}) = 2\mathbf{G}^T \mathbf{A} \mathbf{Gx} - 2\mathbf{G}^T \mathbf{Ag} \stackrel{!}{=} 0 \quad (4.6)$$

Finding vertex positions \mathbf{x} is formulated as a linear system for:

$$\mathbf{G}^T \mathbf{A} \mathbf{Gx} = \mathbf{G}^T \mathbf{Ag} \quad (4.7)$$

4.6 Limitations

Harmonic surface manipulation can create very smooth, plausible deformations, but has limitations.

First, depending on the size of the handle regions, different artifacts may appear. For small handle regions and strong deformations, flipped triangles and surface intersections are common. For larger

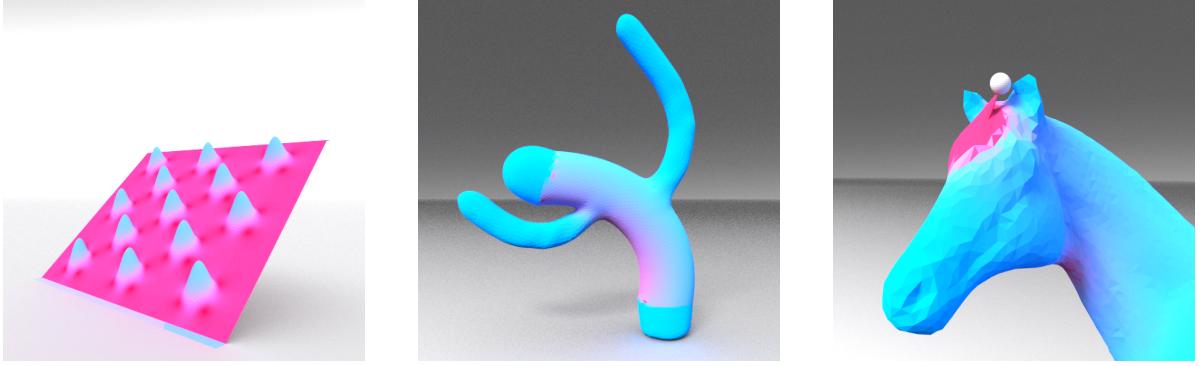


Figure 4.2: Limitations of harmonic surface deformation. *Left to right:* Translation insensitivity, local geometry distortion near boundary between handle and free mesh regions, protruding triangles near handles.

handle regions, there might be a sharp transition which changes the shape of the mesh on the boundary between constrained and unconstrained regions. Other related artifacts include protruding elements close to handle regions.

Second, this method contains no mechanisms to prevent global surface intersection.

Third, the linear model does not an accurate model of the bending and shearing energy of a real surface. For strong rotation it might create unintuitive results.

Moreover, the modeler is required to find a good balance between rotational constraints and positional constraints. This means that creating good deformations is not always straightforward.

Finally, being a global method, depending on the distribution of handles on the mesh, the harmonic field might deform the mesh in an unexpected way, since deformation constraints are propagated over the entire mesh. See Figure 4.2 for examples of some of these limitations. In this work, I propose a method to solve some of these problem.

Chapter 5

Energy Regularization

Linear deformation methods are efficient for creating plausible, physical-like deformations. However, their results are not necessarily artifact free. Possible artifacts include degenerate triangles, protruding elements and self-intersection. These artifacts invalidate the resulting deformation. Strong deformations are especially vulnerable to protruding elements or sharp transitions between constrained and unconstrained mesh regions. Since harmonic surface deformations is a linear method, it is susceptible to these artifacts.

Modeling constraints such as positive, non-zero triangle area for all triangles and the absence of protruding triangles is complex. A common approach is to use non-linear terms to model these constraints in a iterative scheme [Lip12] [SKPSH13]. Other artifacts, such as global self-intersections, are even harder to detect and avoid. There is no straightforward formulation to model these constraints as linear problems.

Energy regularization has been used extensively in signal and image processing, mostly to solve ill-posed inverse problems. Linear deformation is not necessarily ill-posed, but introducing energy regularization might prove beneficial to deal with deformation artifacts, as they are associated with high variance in the energies on the domain [MRT14].

In this chapter I will first examine how energy regularization is used in image and signal processing, I will then explain why this approach is not suitable for geometry processing and present an alternative approach to energy regularization for surface deformations.

5.1 Tikhonov Regularization

Tikhonov regularization is a classic approach to linear energy regularization. It is used to find solutions with different qualities for ill-posed linear systems $\mathbf{Ax} = \mathbf{b}$ where \mathbf{b} is not unique. This is modeled as a β weighted energy functional with two components, a data term E_D and a smoothness term E_S , both minimized in the least square sense using the Frobenius norm, over the domain Ω :

$$E_R(\mathbf{x}) = (1 - \beta) \underbrace{\int_{\Omega} \|\mathbf{Ax} - \mathbf{b}\|^2 d\mathbf{x}}_{E_D \text{ data term}} + \beta \underbrace{\int_{\Omega} \|\Gamma\mathbf{x}\|^2 d\mathbf{x}}_{E_S \text{ smoothness term}} \quad (5.1)$$

Classical choices for the Tikhonov smoothness term $\|\Gamma\mathbf{x}\|$ include:

- $\|\mathbf{x}\|$ Total magnitude of the solution. This is used mainly for signal processing, and irrelevant for geometry processing. For geometry processing, the function x is the mesh positions. Minimizing the magnitude of this will cause changes to the shape.
- $\|\nabla\mathbf{x}\|$ Spatial variation of the solution, the solution smoothness. This regularization term might be valid for very specific deformation problems, such as smoothing noisy meshes, but cannot be used for general deformations where one of the goals is to preserve the local geometry.

- $\|\Delta \mathbf{x}\|$ Minimizing the Laplace operator on the surface. Again this approach might be useful for some very specific geometry processing tasks, but it is not suitable for any task where preservation of the local mesh geometry is important.

These regularization terms are not suitable for the purpose of general geometric processing tasks. In all three cases the smoothness term affects the local geometry of the mesh. In many tasks, especially for surface deformation tasks, it is important to preserve the local mesh geometry.

There are even more inherit problems with using Tikhonov regularizes for geometry processing and specifically, mesh deformations. In many deformation methods, the energy functional measures very specific qualities of the deformation. Simply weighting these with a geometric smoothness term is unlikely to improve the deformation, as the smoothness term is independent on the specific energies being optimized. Generic Tikhonov regularizes cannot guarantee that a deformation is free from degenerate triangles, local and global intersections or protruding triangles.

Due to these reasons, Tikhonov regularization is not suitable as a regularization strategy for harmonic surface deformation.

5.2 Energy Smoothness Regularization

In contrast to the Tikhonov regularization which is independent of the specific problem and its data term, Martinez Esturo et al. [MRT14] propose a novel regularization approach that is coupled with the deformation energy. Their method is generic, and creates problem-specific regularization. It is motivated by an observation that artifacts are correlated to large variations in the *energy* on the domain around their location.

An energy functional of linear deformation methods measures problem-specific energies of a feasible solution $\mathbf{x} : \Omega \rightarrow \mathbb{R}^3$ on each point of the domain. An energy regularization term measures the spatial variation of the energy over the domain.

For an energy functional where the solution \mathbf{x}' minimizes the energy \mathbf{e}_f on the mesh domain Ω :

$$E_D(\mathbf{x}) = \int_{\Omega} \|\mathbf{e}_f(\mathbf{x})\|^2 d\mathbf{x} \quad (5.2)$$

The energy smoothness term is defined as:

$$E_S(\mathbf{x}) = \int_{\Omega} \|\nabla \mathbf{e}_f(\mathbf{x})\|_F^2 d\mathbf{x} \quad (5.3)$$

This is a generic approach that creates an energy regularization term which is both problem specific and strongly coupled with the problem specific energy. For this formulation to be valid, the deformation energy and energy functional has to fulfill the following requirements:

- The energies \mathbf{e}_f measure a specific scalar quality of the solution at the point on the domain (locally).
- The energy term is spatially differentiable.
- The energy \mathbf{e}_f term is linear in \mathbf{x} .
- The energy functional is quadratic in \mathbf{e}_f .
- The solution has a unique solution which is found by solving a linear system of equations.
- The energy operator is global.

There are two distinct benefits for using this formulation of an energy smoothness term:

First, the regularization term is an energy smoothness term, directly relates to main energy term, the data term. It is strongly coupled with the energies being optimized.

Second, the energy smoothness term is problem-specific. The regularization term measures the changes in the problem-specific energy operator which was designed to minimize for certain qualities.

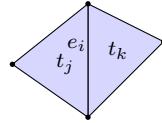
Other approaches to regularization discussed in Section 2.4 also use problem specific energies. However, this approach is generic and directly relates to the energy being optimized. It is also simpler to implement. The solution can be computed in a single-step.

5.3 Implementation for Triangle Constant Energies on Surface Meshes

For a differentiable energy function on a triangle mesh $\mathcal{S} = (\mathcal{V}, \mathcal{T}, \mathcal{E})$, the energy smoothness term is a differential operator on the energy field. Many deformation methods, including harmonic surface manipulation, create triangle-constant energy terms. In these cases, the discretization of the differential energy operator is straightforward. Since energies are triangle constant, $\|\nabla \mathbf{e}_f(\mathbf{x})\|$ vanishes on all triangle-interior points. The regularization operator differs from zero only across triangle boundaries, in the direction perpendicular to the edge direction. Integrating the energy smoothness operator over the mesh domain is equivalent to integrating the value of the differential operator along the mesh edges.

This translates into a neighbor relationship between the triangles. Triangles are neighbors if both are incident to the same edge. Edges which are incident to two triangles are called *interior edges*, and $|\mathcal{E}_i|$ is the cardinality of the set of the mesh interior edges. The neighbor relationship can be expressed in matrix form as a sparse discrete differences matrix \mathbf{D} . It has a size of $|\mathcal{E}_i| \times |\mathcal{T}|$. For interior edge i and triangle t :

$$D_{it} = \begin{cases} 1 & t \text{ is left neighbor of } i \\ -1 & t \text{ is right neighbor of } i \\ 0 & \text{else} \end{cases} \quad (5.4)$$



For the example above, $D_{ij} = 1$, $D_{ik} = -1$.

The energy smoothness operator can be formulated from \mathbf{D} . The discrete energy differential operator $\mathbf{D}_n = \mathbf{D} \otimes \mathbb{I}_n$, creates the global energy smoothness operator for the entire mesh when applied to the mesh energy term. \mathbb{I}_n is an identity matrix of size n , n is the dimension of \mathbb{R}^n and \otimes represent the Kronecker product of the matrices. For a two-manifold in \mathbb{R}^3 , the resulting operator has dimensions of $3|\mathcal{E}_i| \times 3|\mathcal{T}|$.

Let \mathbf{B} be a diagonal matrix with all interior edge lengths, its dimensions are $|\mathcal{E}_i| \times |\mathcal{E}_i|$. Then $\mathbf{B}_n = \mathbf{B} \otimes \mathbb{I}_n$ is a $n|\mathcal{E}_i| \times n|\mathcal{E}_i|$ matrix which is used as the norm for the energy regularization term. It performs the integration of the differential operator over the mesh edges. For a two-manifold in \mathbb{R}^3 , the operator size is $3|\mathcal{E}_i| \times 3|\mathcal{E}_i|$.

For gradient domain surface deformation, the resulting energy smoothness term is:

$$E_{\mathcal{S}}(\mathbf{x}) = \|\mathbf{D}_n(\mathbf{E}\mathbf{x} - \mathbf{g})\|_{\mathbf{B}_n}^2 \quad (5.5)$$

Formulation as a Norm

The energy smoothness operator can be incorporated into the norm of the data term, creating a new energy formulation:

$$\begin{aligned}
E_S(\mathbf{x}) &= \|\mathbf{D}_n(\mathbf{Ex} - \mathbf{g})\|_{\mathbf{B}_n}^2 \\
&= (\mathbf{D}_n(\mathbf{Ex} - \mathbf{g}))^\top \mathbf{B}_n (\mathbf{D}_n(\mathbf{Ex} - \mathbf{g})) \\
&= (\mathbf{Ex} - \mathbf{g})^\top \underbrace{\mathbf{D}_n^\top \mathbf{B}_n \mathbf{D}_n}_{\mathbf{D}_n^\top \mathbf{B}_n \mathbf{D}_n} (\mathbf{Ex} - \mathbf{g}) \\
&= \|(\mathbf{Ex} - \mathbf{g})\|_{\mathbf{D}_n^\top \mathbf{B}_n \mathbf{D}_n}^2
\end{aligned} \tag{5.6}$$

Since the structure of the energy smoothness term is similar to the unregulated energy functional, the two terms can be combined into a *single* term with a β -weighted norm:

$$\begin{aligned}
E_R(\mathbf{x}) &= (1 - \beta) E_D(\mathbf{x}) + \beta E_S(\mathbf{x}) \\
&= (1 - \beta) \|\mathbf{Ex} - \mathbf{g}\|_{\mathbf{A}}^2 + \|\mathbf{Ex} - \mathbf{g}\|_{\mathbf{D}_n^\top \mathbf{B}_n \mathbf{D}_n}^2 \\
&= \|\mathbf{Ex} - \mathbf{g}\|_{\mathbf{W}_\beta}^2
\end{aligned} \tag{5.7}$$

where

$$\mathbf{W}_\beta = (1 - \beta)\mathbf{A} + \beta \mathbf{D}_n^\top \mathbf{B}_n \mathbf{D}_n \tag{5.8}$$

Implementing this approach has several benefits. First, the problem formulation remains unchanged. Adding regularization simply requires changing the norm of the energy operator. Second, only a single linear system needs to be solved for each deformation. Finally, both the solution and regularization terms are computed in a single, global step. However, there is a drawback to this regularization approach: changing the value of β requires to re-factorize the system. Fortunately, this can be done in interactive run time for relatively large meshes.

Chapter 6

Enhancing Harmonic Surface Deformation using Energy Regularization

Previous work on gradient domain mesh deformations [YZX⁺04] [SP04] [ZRKS05] does not discuss regularizing energies for harmonic surface deformation or suppressing common artifacts associated with this deformation method.

Although harmonic surface deformation creates smooth results, very strong deformation constraints, such as strong positional or rotational constraints, can and do create artifacts. Protruding elements and local surface intersection are common around handles for strong deformation constraints. Strong deformations can also distort the local surface geometry on the boundaries between constrained and unconstrained regions. All of these artifacts are usually coupled with strong variation of the triangle constant energies on the mesh, making them suitable candidates for energy smoothness regularization.

In harmonic surface deformation, the energy term of the energy functional is defined as the residual between the deformation guidance field and the surface gradient. The guidance field is created independently for each triangle. It is a set of non-conforming transformations. The gradient field of the deformed mesh is a given conforming.

Equation 4.2 defined the energy functional for harmonic surface deformation:

$$\mathbf{e}_f = \nabla \mathbf{x} - \mathbf{g} \quad (6.1)$$

$$E(\mathbf{x}) = \int_{\Omega} \|\mathbf{e}_f(\mathbf{x})\|^2 d\mathbf{x} \quad (6.2)$$

where \mathbf{x} is the vertex positions function and \mathbf{g} is the deformation guidance field.

Poisson methods for gradient domain surface manipulation, and specifically, harmonic surface manipulation, are suitable for linear energy regularization. The energy \mathbf{e}_f is linear w.r.t. \mathbf{x} and is spatially differentiable, due to the linearity of the operator, on a piecewise linear domain. The energy functional $E(\mathbf{x})$ is global, quadratic to \mathbf{e}_f , and it has a unique solution which is found by a linear system of equations. Thus all requirements are fulfilled.

The mesh gradient field is triangle constant, thus it yields a single constant value for each triangle on the mesh. Using the discrete formulation in Equation 5.7, the discrete energy functional for regularized harmonic surface deformation is:

$$E(\mathbf{x}) = \|\mathbf{G}\mathbf{x} - \mathbf{g}\|_{\mathbf{W}_\beta}^2 \quad (6.3)$$

where \mathbf{W}_β a the β weighed norm.

6.1 Discussion and Drawbacks

Introducing energy regularization into harmonic surface deformation slightly increases the computational complexity of the method. One of the benefits of harmonic surface deformation is that a single system is factored and used to solve for the harmonic field and all deformations. Once a β weighted norm is introduced, the system needs to be re-factored each time the β value changes. The solver for the harmonic field cannot be reused, only for $\beta = 0$. Today, run time and solver reuse are not as major concern as they were when the code was first published, but some of the elegance of the method is lost.

There is also an increased runtime cost due to setting up the neighbor relation operator. However, this needs to be done only once for a each model. There is also a slight cost increase due to multiplying larger sparse systems, but it only increases linearly with the size of the mesh and can be neglected.

Energy smoothness does not guarantee artifact free deformation. It contains no mechanisms that prohibit these artifacts, merely reducing them. It does not allow direct control on how these artifacts are fixed. And, since the operator is global, it might affect areas where no changes are desired.

6.2 Approach Details

This chapter discusses how the different stages of harmonic surface manipulation were approached. The process can be divided into five distinct steps and one optional interaction:

1. Creating matrix representations of the mesh and its operators.
2. Recovering deformation constrains from the user.
3. Solving a Poisson problem for a harmonic field.
4. Constructing a guidance field.
5. Solving for new mesh positions.
6. Adjusting the regularization weight

To deform a mesh, the user must create at least two deformation constraints. In the case of a single handle, they have to choose at least one handle vertex and at least one fixed vertex. In this case, these are sufficient conditions to solve for the harmonic field. In case of multiple deformation handles, the user must set at least two handle regions, see Section 6.3.

The mesh is converted into matrix representation, while building the mesh gradient and Laplace operator, see Section 6.4.

The handles define a Poisson problem, which is solve for a harmonic field. In case of a single manipulation handle, the harmonic field is solved only once. In case of multiple deformation handles, the harmonic field directly represent the quaternion components and it is solved for after each handle manipulation, see Section 6.5).

For each triangle, a single rotation is interpolated. For a single handle, the value of the harmonic field in the triangle barycenter is used for SLERP interpolation between the the handle rotation matrix and the identity rotation (see section 3.4.4 for further details in SLERP). For multiple handles, the rotation quaternion is interpolated directly from the harmonic field on the mesh. The triangle gradient is transformed. This creates the *deformation guidance field*, see Section 6.6).

Next, a single step weighted variational problem is solved. This leads to a global optimization problem of new mesh gradients which fit the deformation gradients in the least-squares sense, while fulfilling the positional constraints on handle vertices, see Section 6.7.

The user can choose to change the energy regularization weight. This requires setting up a new system, see Section 6.8).

Finally I will discuss the resources that were used to implement this approach (Section 7.10).

6.3 User Interaction

After loading a (2-manifold) mesh, the user can select vertices and assign them to modeling regions. A modeling region is a set of vertices which transform rigidly. Ultimately, this is a single vertex or a set of connected vertices. There are no restrictions on choosing fixed vertices.

In case of a single deformation handle, vertices can be labeled as either *handle* or *fixed*. In the case of multiply handles, all modeling regions are treated as source, creating positional constraints. Each vertex can belong to at most one handle. Vertices which do not belong to any modeling region are known as *free vertices*.

The user can translate and rotate handles using a manipulator. For the purpose of this work, rotations were sufficient, but harmonic surface manipulation also supports scaling and shearing operations.

6.4 Discrete Mesh Representation and Operators

Two aspects of the mesh are represented: its topology and its geometry. The original mesh coordinates and gradient operator are stored and used for all deformations.

In this step, all discrete operators are created using the source mesh geometry: \mathbf{G} the discrete gradient operator and \mathbf{A} the associated triangle area weighted norm. The discrete mesh Laplacian is then $\mathbf{L} = \mathbf{G}^\top \mathbf{A} \mathbf{G}$.

The triangle differential operator \mathbf{D}_3 is also created, which creates a $3|\mathcal{E}_i| \times 3|\mathcal{T}|$ sparse matrix. This operator is used with an interior-edges lengths weighted norm \mathbf{B}_3 , which is a diagonal matrix of dimensions $3|\mathcal{E}_i| \times 3|\mathcal{E}_i|$.

Multiplying them creates an edge-weighted differential energy operator: $\mathbf{D}_3^\top \mathbf{B}_3 \mathbf{D}_3$ with size $3|\mathcal{T}| \times 3|\mathcal{T}|$ which is dimensionally compatible with the triangle-area weighted norm \mathbf{A} . The β weight norm $\mathbf{W}_\beta = (1 - \beta)\mathbf{A} + \beta\mathbf{D}_3^\top \mathbf{B}_3 \mathbf{D}_3$ is of a size compatible to be a norm.

6.5 Constructing the Harmonic Field

To construct the guidance fields on the mesh, I follow the approach used by Zayer et al. [ZRKS05]. The handle transformations are recovered from the user interaction, and a polar decomposition scheme is used to separate the rotational transformation from any scaling operation introduced by the user. This rotation is represented as a quaternion. For the purpose of this work, any other transformations are ignored. Interpolating using a harmonic field is done differently for single and multiple handles.

Single Handle

In the case of a single handle, a single harmonic field is constructed for the mesh. The harmonic field on the mesh is the solution to the Laplace equation on the mesh. In a discrete field setting, this corresponded to

$$\mathbf{L}\mathbf{h} = 0 \quad (6.4)$$

$$\mathbf{h} = \begin{cases} 0 & \text{vertex is fixed} \\ 1 & \text{vertex is handle} \end{cases} \quad (6.5)$$

where \mathbf{L} is the mesh Laplace-Beltrami operator, \mathbf{h} is the harmonic function on the mesh. This defines a Dirichlet problem which is analogous to the steady-state heat equation.

Using the Laplace-Beltrami operator ensures that the gradient flow of the harmonic field respects the intrinsic surface geometry.

The harmonic field is solved for once only, and it can be used for any deformation as long as the configuration of handle and source vertices remain unchanged.

Multiple Handles

The harmonic field is handled differently in the case of multiple deformation handles. Handles are no longer defined as sources and sinks. All handles are sources of different transformation. The harmonic field needs to be resolved every time the user rotates any of the handles.

The transformation of each handle is decomposed into a rotational and scaling component. The rotational component is represented as a quaternion. For the purpose of this work, any scaling operation is ignored. Rather than a single harmonic field, we solve for four independent harmonic fields.

The rotational component is represented as a quaternion $q = (q_w; q_x, q_y, q_z)$. A single solver is used to solve for four harmonic fields on the mesh, one for each component of the quaternion. The quaternion values on the handle vertices $\{h_j\}$ act as boundary constraints for the problem:

$$\begin{aligned} \mathbf{L}\mathbf{q}_i &= \mathbf{0}, i \in (w, x, y, z) \\ \mathbf{q}_i(h_j) &= q_i^{h_j} \end{aligned} \quad (6.6)$$

where $q_i^{h_j}$ is the value of the quaternion component $i \in (w, x, y, z)$ at the handle $h_j \in (1, \# \text{ of handles})$. By default all handles are sources of the identity transformation.

Solving these system results in explicit quaternion values on each vertex of the mesh and explicit scaling matrices for all mesh triangles.

6.6 Constructing the Guidance Field

For single handles, the value of the harmonic field is used for SLERP (see Section 3.4.4) interpolation between the source and identity transformation to get a rotation matrix \mathbf{R}_i , $i \in (1, |\mathcal{T}|)$. For multiple handles, the rotation matrix is created from the explicit quaternion values solved for in Equation 6.6.

The mesh gradients $\mathbf{y} = \mathbf{G}\mathbf{x}$ creates a $3|\mathcal{T}| \times 3$ vector of stacked triangle gradients. Each triangle's gradient matrix is a 3×3 block in this vector. Each triangle gradient is rotated using the rotation \mathbf{R}_i :

$$\mathbf{g}_i = \mathbf{y}_i \mathbf{R}_i^\top \quad (6.7)$$

Scaling transformation are handled similarly. For single deformation handle, the value of the harmonic field is directly used to interpolate linearly between the scaling at the source and fixed vertices. For multiple deformation handles, the harmonic field of the scaling factors gives the scaling factor for each dimension.

All transformed mesh gradients are stacked into \mathbf{g} of size $3|\mathcal{T}| \times 3$.

6.7 Solving for Vertex Positions

The final problem that needs to be solved is:

$$\mathbf{G}^\top \mathbf{W}_\beta \mathbf{G} \mathbf{x}' = \mathbf{G}^\top \mathbf{W}_\beta \mathbf{g} \quad (6.8)$$

The matrix $\mathbf{G}^\top \mathbf{W}_\beta \mathbf{G}$ is factorized using a sparse Cholesky solver. Each dimension is solved for independently, solving for three discrete position functions over the domain: $(\mathbf{x}_x, \mathbf{x}_y, \mathbf{x}_z)$. The positions of the all handles and fixed vertices act as Dirichlet boundary constraints for this Poisson problem.

6.8 Changing Regularization Weight

Changing the β value requires updating the \mathbf{W}_β norm. The deformation guidance field \mathbf{g} remains unchanged, but both the left and right hand terms in Equation 6.8 needs to be updated. This requires factorizing a new linear system. The factorization can be reused for all deformation with the same handle configuration and β weight.

Chapter 7

Results

I tested the proposed approach to energy regularization on meshes with 3k-90k vertices. Deformations were performed using a single and multiple handles, varying sizes of the handle regions and the regularization weight. The deformations were assessed both qualitatively and quantitatively.

This chapter is structured as follows: the quantitative and qualitative measures are introduced (Section 7.1), and the choices for visualization methods are explained (Section 7.2). Regularization was tested on benchmark deformations for linear methods (Section 7.3). Then, I tested what the effects of varying the size of the handles are and how regularization affects these deformations (Section 7.4) and whether regularization can be useful for dealing with translation only deformations (Section 7.5). I tested energy regularization as a method for suppressing triangle flips, a common artifact in planar deformations, which can also appear in specific cases on surface meshes (Section 7.6). Next, regularization was tested on a variety of models, using a mixture of small and large handle regions. Quantitative and qualitative analysis was performed for effective regularization (Sections 7.7) and for failure cases (Section 7.8). Finally, I discuss performance and implementation details (Sections 7.9 and 7.10).

7.1 Analysis Criteria

The aims of introducing energy regularization is to improve the plausibility of the deformations, by suppressing artifacts. Plausibility is not a well-defined property, but there are several quantitative and qualitative properties that are well-defined, that can be used to assess the quality of the deformation.

7.1.1 Quantitative Analysis

For the quantitative analysis, the following error measurements were used:

Total deformation energy $E(\mathbf{x})$, as defined in Equations 4.4.

This measurement is going to the lowest for $\beta = 0$ by definition. The difference in the total deformation energy between the total deformation energy of the regularized deformation and the naive deformation indicates how far these solutions deviate in the solution space.

Total normalized isometric and conformal errors $E_{\text{iso}}^N, E_{\text{conf}}^N$ respectively, as defined in Equations 3.23 and 3.26. I expect both error measures to be strongly coupled with the total deformation energy, i.e. a rise in the β value and deformation energy is expected to cause this error measurement to rise as well.

Total normalized area change E_{area}^N as defined in Equation 3.27. Minimizing the change to the total surface area relates to minimizing the stretching energy.

Maximum isometric and conformal errors, $\max(e_{\text{iso}}^i)$ and $\max(e_{\text{conf}}^i)$, $1 \leq i \leq |\mathcal{T}|$ on a single triangle. Locally high isometric and conformal errors are related to artifacts such as protruding elements or surface self-intersections. I expect regularization to reduce these maximal errors.

Maximum local energy $\max((e_f^i)^2)$, $1 \leq i \leq |\mathcal{T}|$ on a single triangle. This is the energy term which is being smoothed by the regularization. Just in the case of the maximum isometric and conformal errors, I expect regularization to decrease the maximal value, but increase the number of high energy triangles.

7.1.2 Qualitative Analysis

Qualitative analysis was performed using the following guidelines:

Local shape preservation Does regularization affects the distortion of small details? Mesh regions close to the handles are susceptible to distortion.

Global shape preservation Does energy regularization affect the global shape of the mesh, or are the effects limited to localized regions?

Flipped triangles and local surface intersection Are any flipped triangles or local surface intersections in the naive deformation suppressed by regularization?

Protruding elements Does regularization effectively suppress protruding elements?

Creation of new artifacts Does introducing regularization create new artifacts?

7.2 Visualization

7.2.1 Energy Visualization

For all deformations in this chapter, the triangle constant energies $(e_f^i)^2$ are visualized. These are the local energies that are being optimized for. The aim of the visualization is to show these local energies change with the regularization weight, without creating bias. Deformation of the same model with different regularization weight should be visually comparable. The colors indicate the value of the energy, light blue indicate low energies and pink-red colors indicate high energies.

Energies in a single deformation can range over ten or twenty orders of magnitude. After trying various methods of compacting this to a meaningful interval, two methods proved effective:

Linear Mapping This process was done independently for each series of deformations with varying regularization weight. The energy range of $(e_f^i)^2$ for the naive deformation was normalized using $\max((e_f^i)^2)$, then mapped into the color space in Figure 7.1. The rest of the deformation in this series are visualized using the same color space.

Consequently, the highest energy visualized is the maximal energy value of the naive deformation. If higher energy values are created on the regularized deformations, they are truncated to the maximal value of the first deformation. This allows to visually compare the changes in energy distribution on different deformations. For deformations where this approach was chosen, the range of energies was compact enough that the visualization of low energy values is meaningful.

This was used for the Botsch and Sorkine benchmark deformation in Figures 7.2,7.3.

Truncated Linear Mapping (TLM) A similar approach is used to visualize the energies in the rest of deformations in this chapter. However, instead of using $\max((e_f^i)^2)$ as a normalization factor, I chose an endpoint value which was greater than approximately 95%. For many deformations, a few triangles had energies which were two or three orders of magnitudes larger than $\max((e_f^i)^2)$ for any regularized deformation. Visualizing normalized energies using the $\max((e_f^i)^2)$ for $\beta = 0$ resulted in uniform mesh coloring.

Using logarithmic mapping did not create better visualizations, therefor the truncated linear mapping was chosen.

The colors for this visualization are shown in Figure 7.1. This approach was used from most other figures in the chapter.



Figure 7.1: Energy visualization color maps.

7.2.2 Visualizing Deformation Constraints

Small constrained regions and single constrained vertices are indicated by spheres. Larger constrained regions are described in the figure captions, to avoid occluding the model or the energy visualization.

7.2.3 Sampling the regularization weight

The effects of changing the regularization weight are subtle, any in many case, can be indistinguishable above a certain β value. For most deformations, the biggest changes were observed when the value of $0 \leq \beta \leq 0.25$, and were less apparent for higher values. Choosing a few β values to sample the entire space was sufficient to show the variation of the deformations. For other deformations I only sampled lower β values, which already proved effective for suppressing artifacts.

7.3 Comparison to Benchmark Deformations

The effectiveness of energy regularization was tested on a series of benchmark deformations for linear deformation methods, created by Botsch & Sorkine [BS08]. These benchmarks are four high quality meshes with well-designed deformations, three of which already create plausible deformations without regularization. All deformations are defined on large constrained regions. The results for original and regularized deformation appear in Figures 7.2 and 7.3. Quantitative analysis of the results is shown in Figure 7.4 and graphed in Figure 7.5.

Since three of these deformations already create good results without regularization, i.e. these are best case scenarios for harmonic surface deformation, I examined the effects of energy regularizations on each one in detail:

Plane: this deformation is not well suited for gradient domain deformation methods, as it only constrains translation. Introducing energy regularization did not improve the results. Very high values of $\beta > 0.95$ introduce sine-wave like distortion into the plane. The deformation for the Plane is not well-suited for gradient domain deformations, and regularization does not improve the results for this deformation.

Cylinder: this deformation creates plausible results with no regularization. Regularization improves the local shape in the transition between the constrained and unconstrained regions. Values of β between 0.25 and 0.5 create very smooth deformations. For $\beta = 0.99$ the results are overly smooth.

Bar: the effects of the energy regularization are subtle. There are some very high energy triangles close to the base of the bar, on the boundary between the constrained and unconstrained region. Introducing regularization removes them. The energy distribution near the boundary between the constrained and free mesh regions is smoother, but their impact on mesh geometry is very subtle.

Cactus: With no regularization, there are a few isolated triangles with very high energy near the boundary between constrained and free regions. For $\beta = 0$ there are two red triangles, one near top of the cactus, the other near the bottom, both close to the boundary between the constrained and free region. For $\beta = 0.25$ these local errors are removed and replaced by a smoother energy distribution, resulting in smoother local shape. For $\beta = 0.5$ and onwards, there is a significant rise in the concentration of high local energies at the center of the free mesh region.

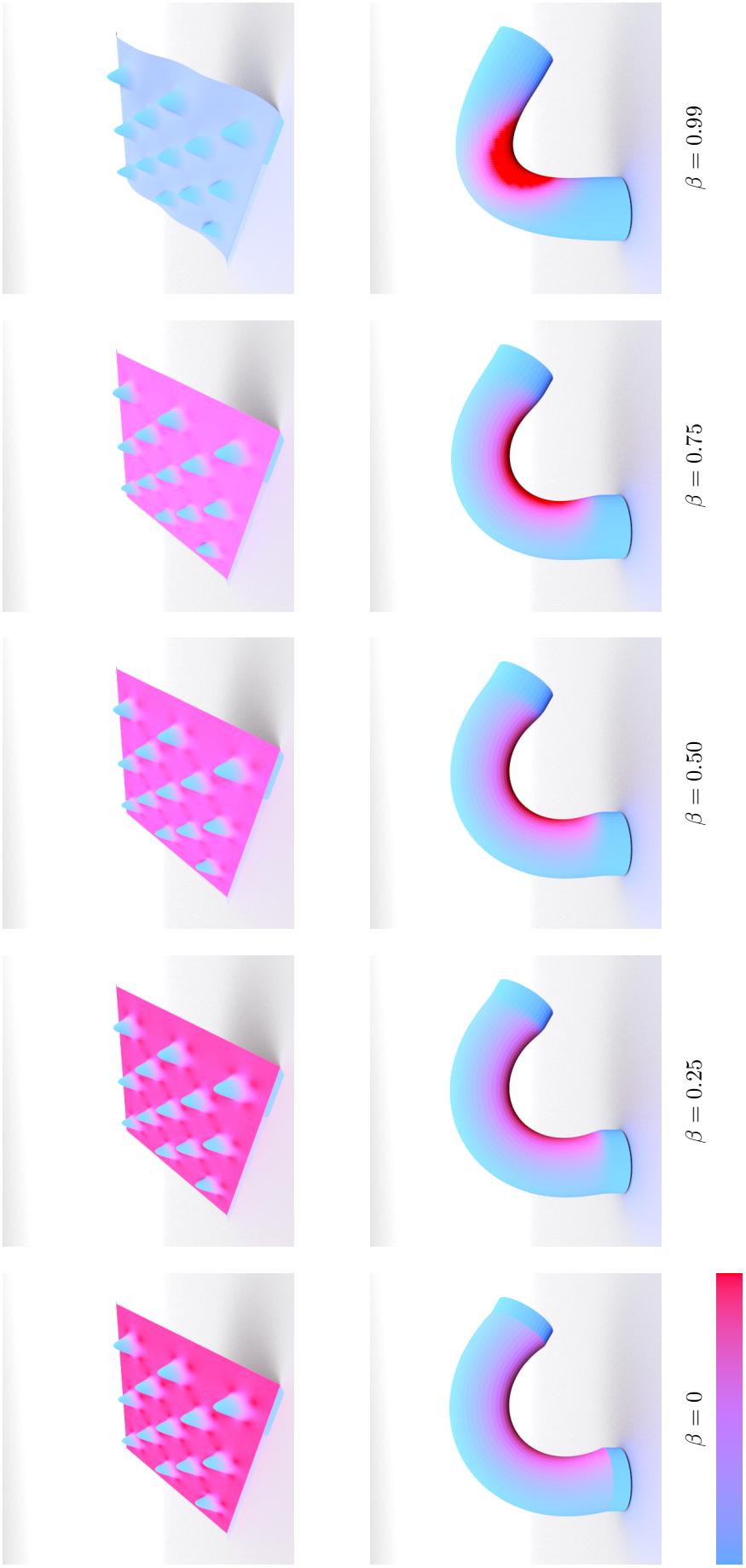


Figure 7.2: Regularization results for benchmark deformations from Botsch and Sorkine's survey [BS08]. *Top row: Plane.* The edges of the plane are constrained, one edge was translated upwards. *Bottom row: Cylinder.* The bottom and top regions of cylinder were constrained. The top handle was translated and rotated.

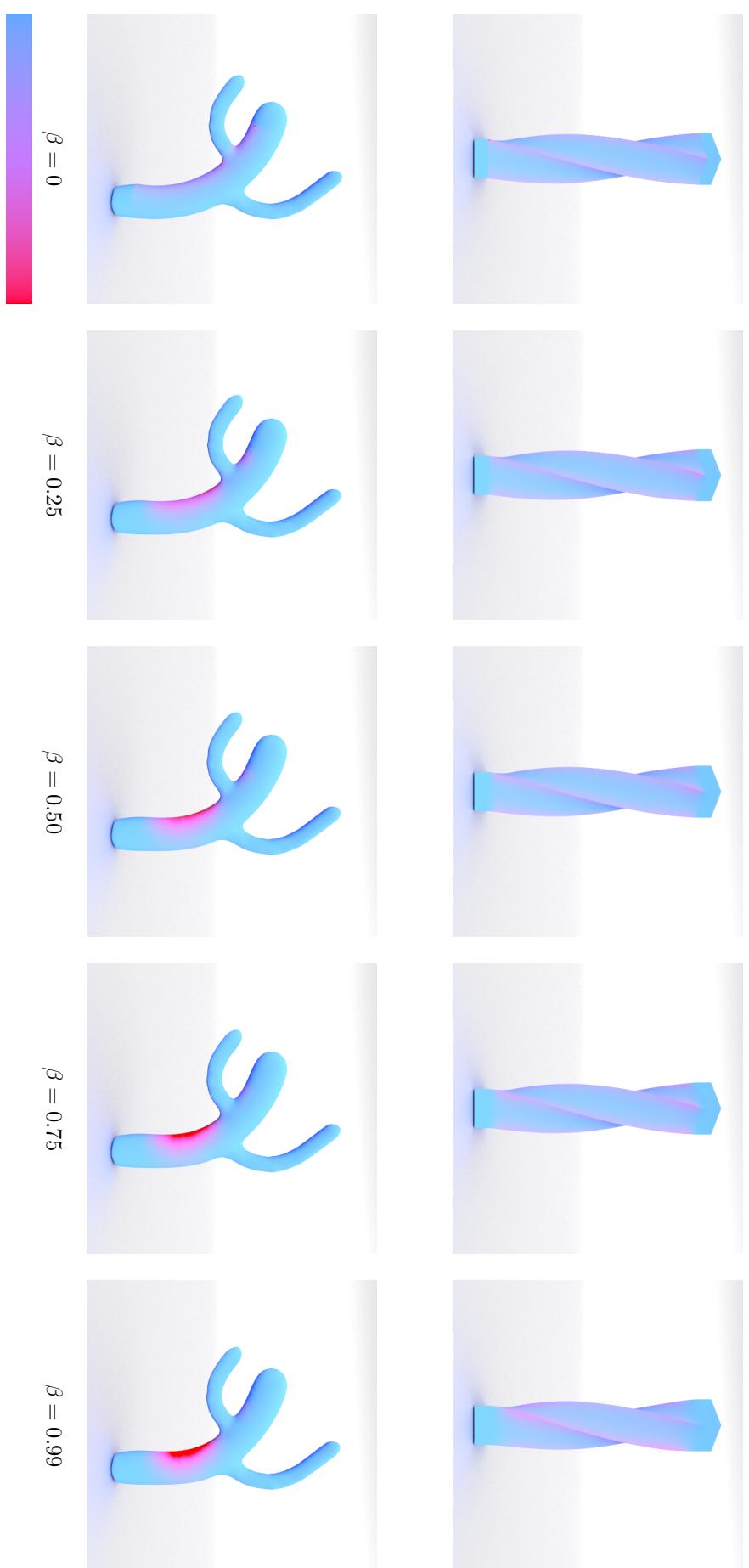


Figure 7.3: Regularization results for benchmark deformations from Botsch and Sorkine's survey [BS08]. *Top row:* Bar. The top and bottom or the bar were constrained, the top was rotated. *Bottom row: Cactus.* The bottom and the top of the cactus were constrained. The top was translated and rotated.

Model	β value	$E(\mathbf{x})$	E_{iso}^N	E_{conf}^N	E_{area}^N	$\max(e_{\text{iso}}^i)$	$\max(e_{\text{conf}}^i)$	$\max((e_f^i)^2)$
Plane	0.0	5102	0.1610	0.0867	31.08	0.4002	0.2244	0.6617
	0.25	5112	0.1612	0.0870	31.37	0.3767	0.2189	0.6764
	0.50	5136	0.1622	0.0877	31.87	0.3815	0.2169	0.7048
	0.75	5198	0.1653	0.0893	32.93	0.3961	0.2116	0.7647
	0.90	5330	0.1732	0.0931	34.60	0.4176	0.2268	0.9061
	0.99	5879	0.2059	0.1086	37.78	0.5613	0.2814	1.4902
Cylinder	0.0	8.596	0.07942	0.04001	-22.83	0.2612	0.1305	0.2613
	0.25	8.679	0.08103	0.04001	-23.12	0.2698	0.1348	0.2699
	0.50	8.778	0.08280	0.04010	-23.43	0.2782	0.1390	0.2784
	0.75	8.957	0.08615	0.04059	-23.94	0.2949	0.1474	0.2952
	0.90	9.242	0.09180	0.04213	-24.61	0.3277	0.1639	0.3283
	0.95	9.489	0.09670	0.04384	-25.09	0.3601	0.1802	0.3609
	0.99	9.966	0.10574	0.04736	-25.88	0.4223	0.2114	0.4233
Bar	0.0	183.1	0.01864	0.01817	-3.170	0.10928	0.06168	0.1591
	0.25	184.1	0.01880	0.01831	-3.160	0.04388	0.03572	0.1074
	0.50	185.3	0.01900	0.01849	-3.185	0.03616	0.03566	0.1080
	0.75	187.4	0.01935	0.01877	-3.288	0.03620	0.03582	0.1097
	0.90	190.5	0.01987	0.01916	-3.542	0.03696	0.03676	0.1129
	0.99	198.7	0.02106	0.01993	-4.274	0.04285	0.04271	0.1286
Cactus	0.0	0.007097	0.01208	0.006272	-6.522	0.1038	0.05236	0.1043
	0.25	0.007795	0.01260	0.006554	-6.450	0.0890	0.04461	0.0966
	0.50	0.008187	0.01280	0.006746	-6.358	0.0904	0.04516	0.1041
	0.75	0.008502	0.01300	0.006921	-6.281	0.0927	0.04631	0.1127
	0.90	0.008662	0.01312	0.007014	-6.243	0.0946	0.04714	0.1175
	0.99	0.008749	0.01318	0.007066	-6.222	0.0959	0.04778	0.1202

Figure 7.4: Quantitative analysis of the deformations in Figures 7.2, 7.3.

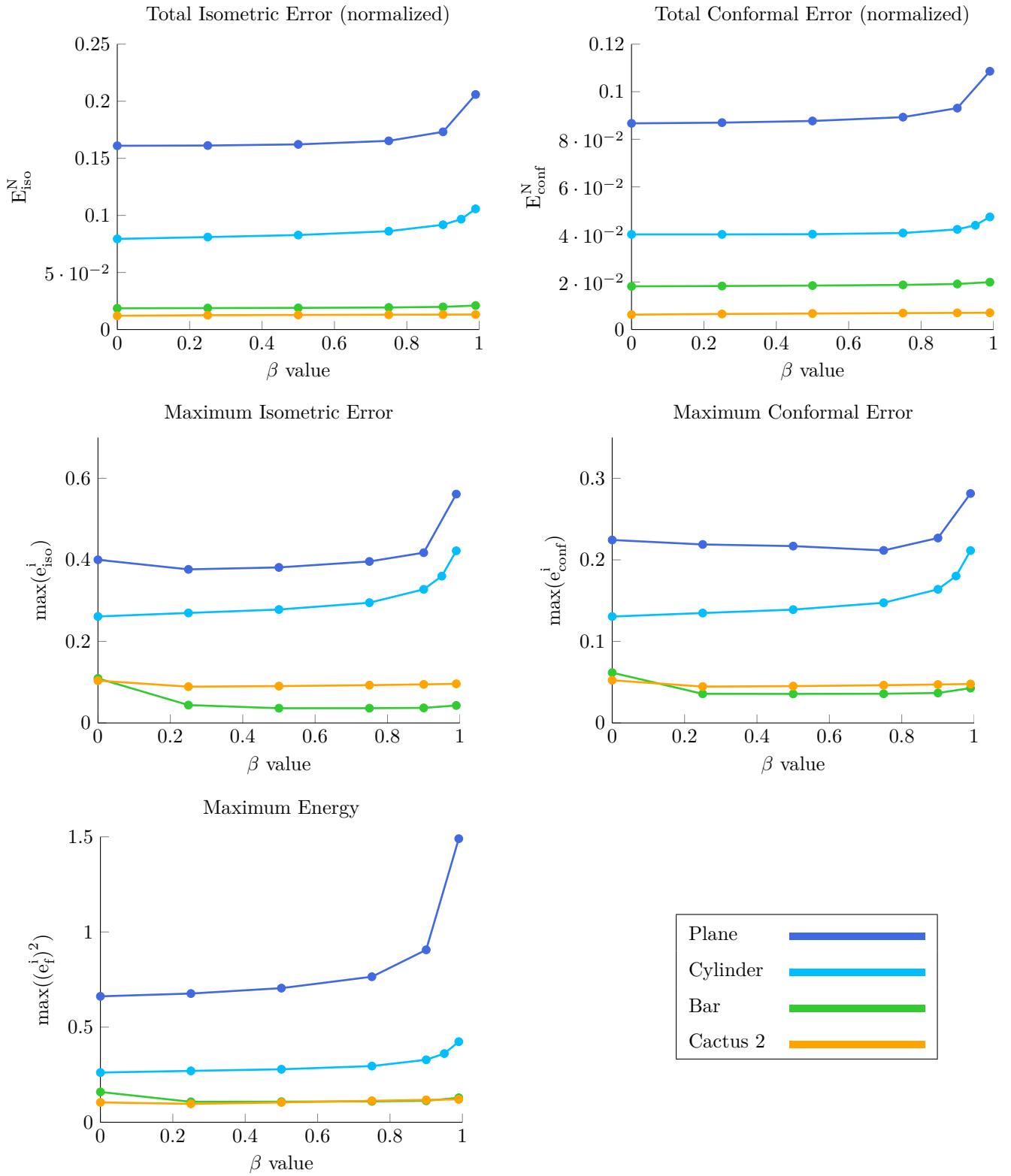


Figure 7.5: Graphs for error measures of the Botsch and Sorkine benchmarks.

7.4 Point-wise Constraints vs. Constrained Regions

Varying the size of the constraint region significantly changes the results of the deformation and the types of artifacts that commonly occur. Figure 7.7 shows the results of performing the same transformation on the same mesh, while varying the size of the handle.

In both of deformations, the entire base of the cactus is fixed. In the first set of deformations, the entire top of cactus was set as a handle region. In the second set, a single vertex acts as a handle. The results of the deformations are different, and create different artifacts. For a handle region, the shape of the top of the cactus is preserved, but there is a visible and sharp transition between the constrained and free mesh regions. Adding energy regularization helps smooth the transition between the constrained and unconstrained region.

For a handle vertex, the naive deformation results in local surface intersections. Introducing moderate regularization fixes this, but the deformed cactus has a different shape from the same deformation with a large constrained region.

For both deformations, there is little difference between the results for $\beta = 0.25$ and $\beta = 0.99$, suggesting that moderate use of regularization is enough to avoid deformation artifacts. The quantitative analysis of these deformations appears in Figure 7.9.

Generally, I observed that deformations with large constrained regions results in local shape distortion around the boundary region between the constrained and unconstrained regions, and smaller modeling regions result in artifacts such as protruding or intruding triangles.

7.5 Translation Only Deformations

Gradient domain surface manipulation is not well suited for deformations which are composed of a translational component alone. Differential surface representation is by definition translation invariant. To improve the results with a naive method, the user needs to specify rotations for surface elements near the translated vertices.

I tested energy regularization as a method to handle this inherent limitation of gradient domain surface deformations. Figure 7.8 shows two deformations: the first deformation is only a translation, the second deformation has composed of a large translation and a small rotation. Without regularization, both suffer from a few triangles protruding from the surface to fulfill the hard positional constraints, but the global shape of the the mesh remains unaffected. The local energies on these protruding triangles are high. Regularization propagates high energy values to a larger mesh region, resulting in a more natural looking deformation, change in the global mesh shape and suppression of these localized artifacts. A quantitative analysis of these deformations appears in Figure 7.10.

A contrary example is the *Plane* deformation from Figure 7.2, which is another translation only deformation. For that deformation regularization fails. This might have to do with the size and orientation of the handle regions.

7.6 Triangles Flips

As mentioned previously in this work, energy regularization cannot guarantee artifact suppression. Degenerate and flipped triangles are a common artifact in planar meshes. While not a common problem in surface meshes embedded in \mathbb{R}^3 , they are a possibility.

Figure 7.6 shows a planar mesh, to demonstrate how regularization affects these artifact. This is a challenging example, with a lot of hard positional constraints and a large translation. All vertices on the mesh boundary are set as fixed, and the central vertex is a deformation handle. It was translated downwards in the same plane as the mesh, creating flipped triangles, which are a common artifact for 2D meshes. The same deformation method was used as in the surface meshes, as this is a 2-manifold surface mesh.

Solving a regularized problem with moderate values of β reduces the number of the flipped triangles. Low to moderate values of $\beta < 0.5$, does not prevent all flipped triangles. Very high values of $\beta > 0.9$ start introducing new triangle flips close to the mesh border.

$\beta = 0.75$ successfully suppresses these flipped triangles without creating new artifacts. Finding the optimal β value is challenging in this case.

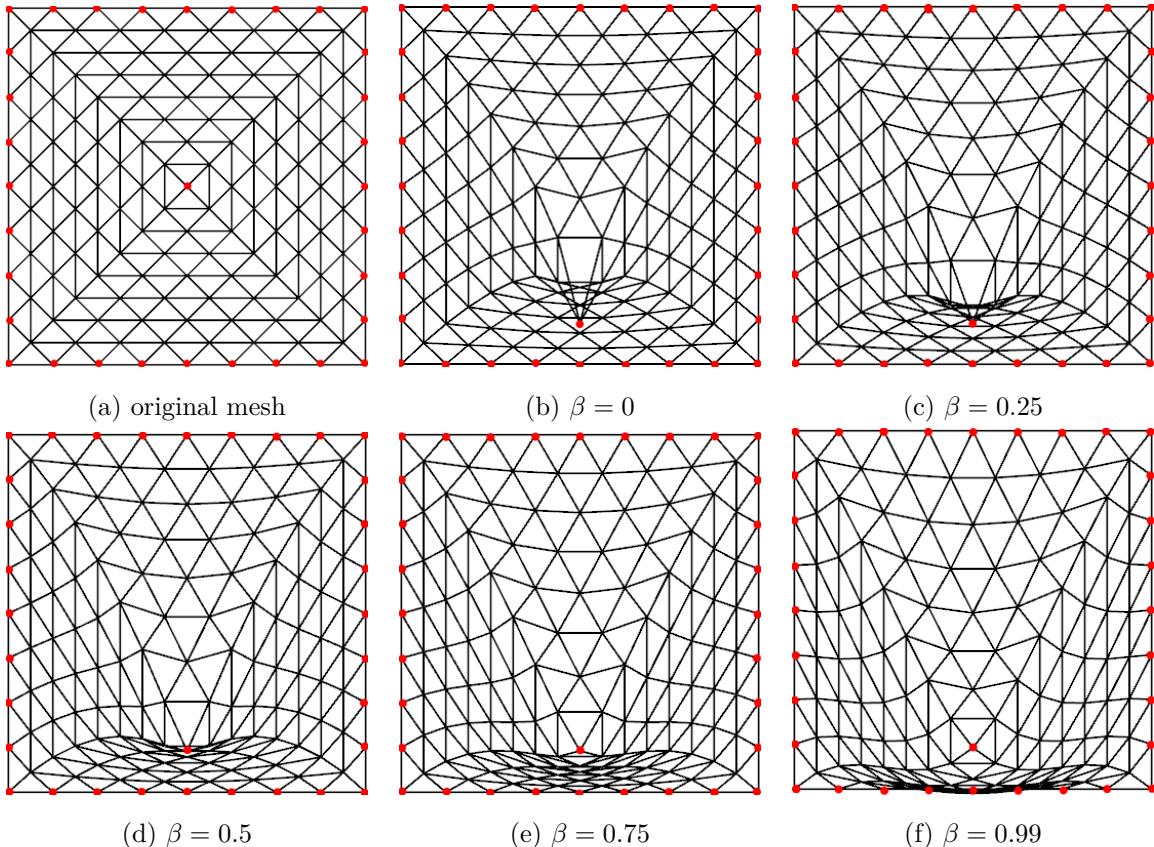


Figure 7.6: Planar input mesh. All red vertices are constrained. The number of flipped triangles decreases until about $\beta = 0.75$ but for $\beta = 0.99$ the solution is worse. The number of flipped triangles increases, in unexpected places. For this example, $\beta \in (0.5, 0.75)$ produces the best results.

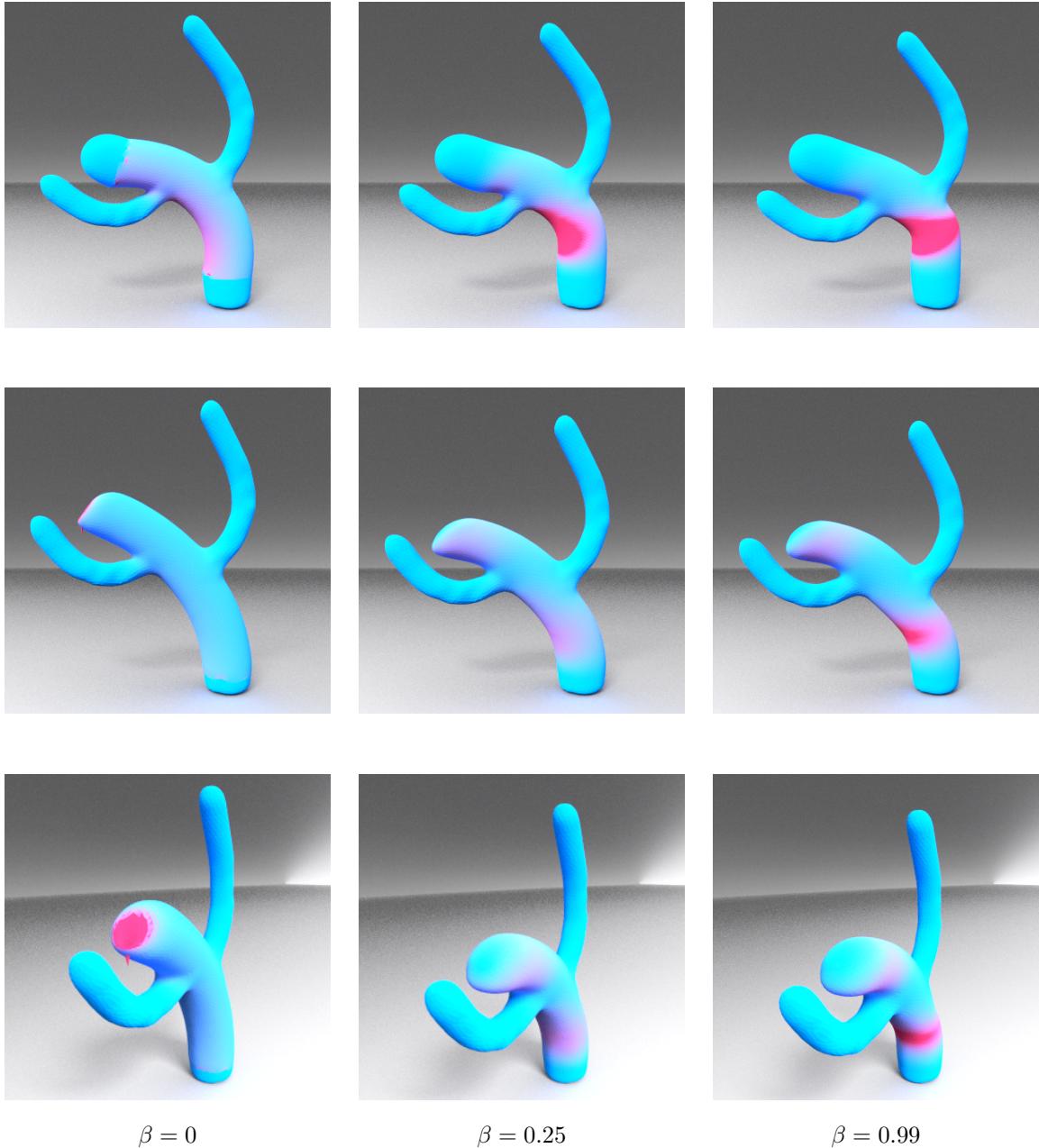


Figure 7.7: Varying the size of the handle regions creates different deformations. Both models were visualized with TLM at 0.3, for comparison. *Top row:* The base of the cactus is fixed, and the entire cap of the cactus is used as a handle region. *Middle row:* Same transformation, but a single vertex acts as the deformation handle. *Bottom row:* Top view of the deformation with a single handle vertex.

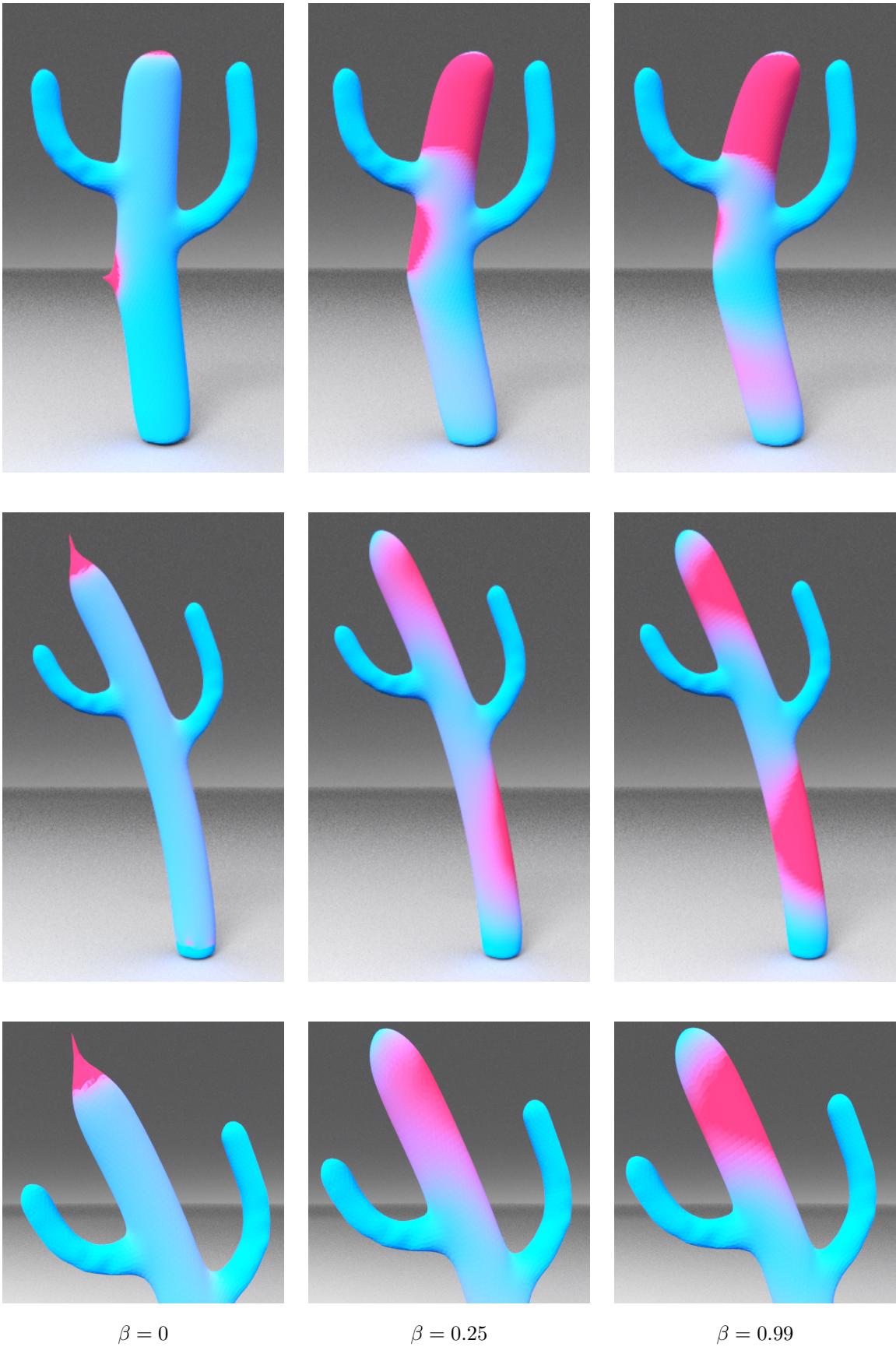


Figure 7.8: Effects of energy regularization on protruding elements. *First row:* TLM at 0.05. A few vertices at the top of the cactus and the entire base of the cactus were fixed. A single vertex was set as a handle and translated outwards. *Second and third row:* TLM at 0.5. The base of the cactus was fixed, and a single vertex at the top was translated upwards and rotated.

Model	β value	$E(\mathbf{x})$	E_{iso}^N	E_{conf}^N	E_{area}^N	$\max(e_{\text{iso}}^i)$	$\max(e_{\text{conf}}^i)$	$\max((e_f^i)^2)$
Handle Vertex	0.0	0.04295	0.06972	0.03856	-18.25	22.329	11.359	23.8573
	0.10	0.05215	0.08632	0.04732	-22.11	0.273	0.143	0.2908
	0.25	0.05410	0.08977	0.04900	-22.40	0.296	0.148	0.3211
	0.50	0.05576	0.09325	0.05061	-22.61	0.337	0.168	0.3671
	0.75	0.05677	0.09570	0.05170	-22.75	0.364	0.182	0.4023
	0.99	0.05746	0.09757	0.05251	-22.86	0.386	0.193	0.4282
Handle Region	0.0	0.03916	0.07210	0.03713	-19.3	0.2884	0.1447	0.3927
	0.25	0.04277	0.08256	0.04146	-19.8	0.4011	0.2006	0.4025
	0.50	0.04466	0.08815	0.04392	-19.9	0.4775	0.2386	0.4785
	0.75	0.04616	0.09236	0.04581	-20.1	0.5315	0.2653	0.5323
	0.99	0.04733	0.09554	0.04726	-20.1	0.5701	0.2844	0.5706

Figure 7.9: Quantitative analysis of the deformations in Figure 7.7.

Model	β value	$E(\mathbf{x})$	E_{iso}^N	E_{conf}^N	E_{area}^N	$\max(e_{\text{iso}}^i)$	$\max(e_{\text{conf}}^i)$	$\max((e_f^i)^2)$
Cactus Long	0.0	0.07885	13909	5395	30.16	60936	30483	46.7165
	0.25	0.09938	14915	5802	36.16	66586	33329	0.5665
	0.50	0.10240	14982	5830	36.34	66737	33409	0.6582
	0.75	0.10415	15027	5848	36.43	66798	33442	0.7181
	0.90	0.10492	15049	5857	36.46	66820	33453	0.7456
	0.99	0.10531	15061	5862	36.48	66831	33459	0.7602
Cactus Side	0.0	0.00661	0.005589	0.003983	0.891	2.9333	1.5011	3.7775
	0.25	0.01203	0.008572	0.007177	1.466	0.0411	0.0410	0.0895
	0.50	0.01317	0.009174	0.007855	1.334	0.0523	0.0523	0.1104
	0.75	0.01390	0.009577	0.008295	1.250	0.0603	0.0603	0.1275
	0.90	0.01424	0.009766	0.008498	1.214	0.0640	0.0640	0.1354
	0.99	0.01442	0.009866	0.008605	1.196	0.0660	0.0659	0.1396

Figure 7.10: Quantitative analysis of the deformations in Figure 7.8.

7.7 Qualitative Analysis of Freestyle Deformations

For this part of the work, I tested energy regularization on many models, with various configurations of handles. I chose to only test low β values, because using high regularization weight is not how regularization should be used in the context of general modeling. The quantitative analysis of the models below appears in numerical form in Figure 7.18 and graphed in 7.23, along with the analysis of the failure cases presented in Section 7.8.

Horse 1. Figure 7.11. The original deformation suffers from protruding elements on the head and back foot. The global shape is distorted - the back leg is elongated and there is an unnatural curve in the horse's back. Regularization suppresses the local artifacts and improves the global shape. The stretch of the back leg is minimized, the curvature of the back looks more natural.

Hand. Figure 7.12. This original deformation suffers from localized artifacts near the small handle regions. All handle regions result in either protruding or intruding elements. Low regularization weight suppresses these artifacts and creates a smooth deformation.

Dragon. Figure 7.13. The original deformation suffers from local shape distortion. The affected region is larger than in the case of a protruding element. Regularization reduces the local shape distortion near the handles. The global shape is subtly stretched and elongated instead.

Horse 2. Figure 7.14. The model suffers from volume loss after the foot was rotated and local shape distortion around the handle on the head of the horse. Regularization recovers the lost volume in the foot and reduces the distortion near the head handle.

Cow 1. Figure 7.15. The model suffers from local shape distortion in the face and volume loss in the horns due to rotations. Regularization solves these problems without affecting the global shape.

Cow 2. Figure 7.16. The back two feet are visibility elongated, distorting the global shape of the cow. Regularization propagates high energy values to larger mesh regions, improving the global shape. This is one of a few examples where even the global shape is visibly affected changing β from 0.5 to 0.75.

XYZ Dragon. Figure 7.17. This is the largest model tested with 90k vertices. It suffers from overall volume loss and shape distortion on the boundary between the fixed and free regions. The effects of regularization are subtle, but the local shape distortion is reduced close to the boundary between the handle and free mesh regions with the increase in β values.

7.8 Failure Cases

I categorized three deformations as failure cases: *Elephant*, *Armadillo* and *Bunny*.

The *Elephant* (Figure 7.19) is an example of a deformation where regularization is ineffective. Regularization does not significantly improve the deformation, nor does it create new artifacts.

In the case of *Armadillo* (Figure 7.20), regularization could have significantly improved this deformation, by changing the global shape of the model, but this is not observed. The local shape remains distorted near the handle on the hand of the model. Only two results are shown, but the deformation with high β values are nearly indistinguishable from the results for $\beta = 0.25$.

The third deformation, *Bunny* (Figure 7.21), is a failure case. Although regularization successfully suppresses the protruding element near the foot, the global shape is visibly distorted.

The quantitative analysis of these deformations appears in Figure 7.22 and graphed together with the deformations from the previous section in Figure 7.23.

7.9 Performance

Deformation were computed on a Intel Core i7 Quad, at 2.20 GHz. Pre-calculations, which include setting up the discrete operators on the mesh, can take up to a few minutes for very large meshes ($> 50k$ vertices). Creating regularization specific matrix structures for the energy regularization account for approximately 30% of the run time of this step.

The systems for meshes of 5-15K were factored and solved in under 0.1 second. The modeling can be performed interactively. For very large meshes of 30-90k vertices, it takes up to 0.5 second.

The performance bottleneck of this method is creating the deformation guidance field, which took up to 120 seconds for meshes with 90k vertices. For meshes of this size some acceleration structure needs to be implemented. This is a limitation of many Poisson methods. The impact of adding regularization is merely the addition of a factorization step whenever the β weight changes.

7.10 Implementation Details

I used OpenMesh, an open source mesh data structure, created by Botch et al. [BSBK02]. Their mesh data structure is based on the half-edge data structure. For user interaction and visualization, I use OpenFlipper, an open-source geometry processing and rendering framework, created by Möbius and Kobbelt [MK12]. I use Eigen SimplicialLLT sparse Cholesky solver [GJ⁺10] with fill-in reducing reordering, implementation courtesy of Janick Martinez Esturo.

7.11 Mesh Sources

The *Armadillo*, *Bunny*, *Dragon* and *XYZ Dragon* models are from the Stanford Scanning Scanning Directory. *Horse*, *Hand*, and *Cow* models are from the AIM @ Shape project. The *Plane*, *Cylinder*, *Bar* and *Cactus* models were taken from Botsch and Sorkine [BS08]. The *Elephant* model is courtesy of Sumner et al. [SP04].

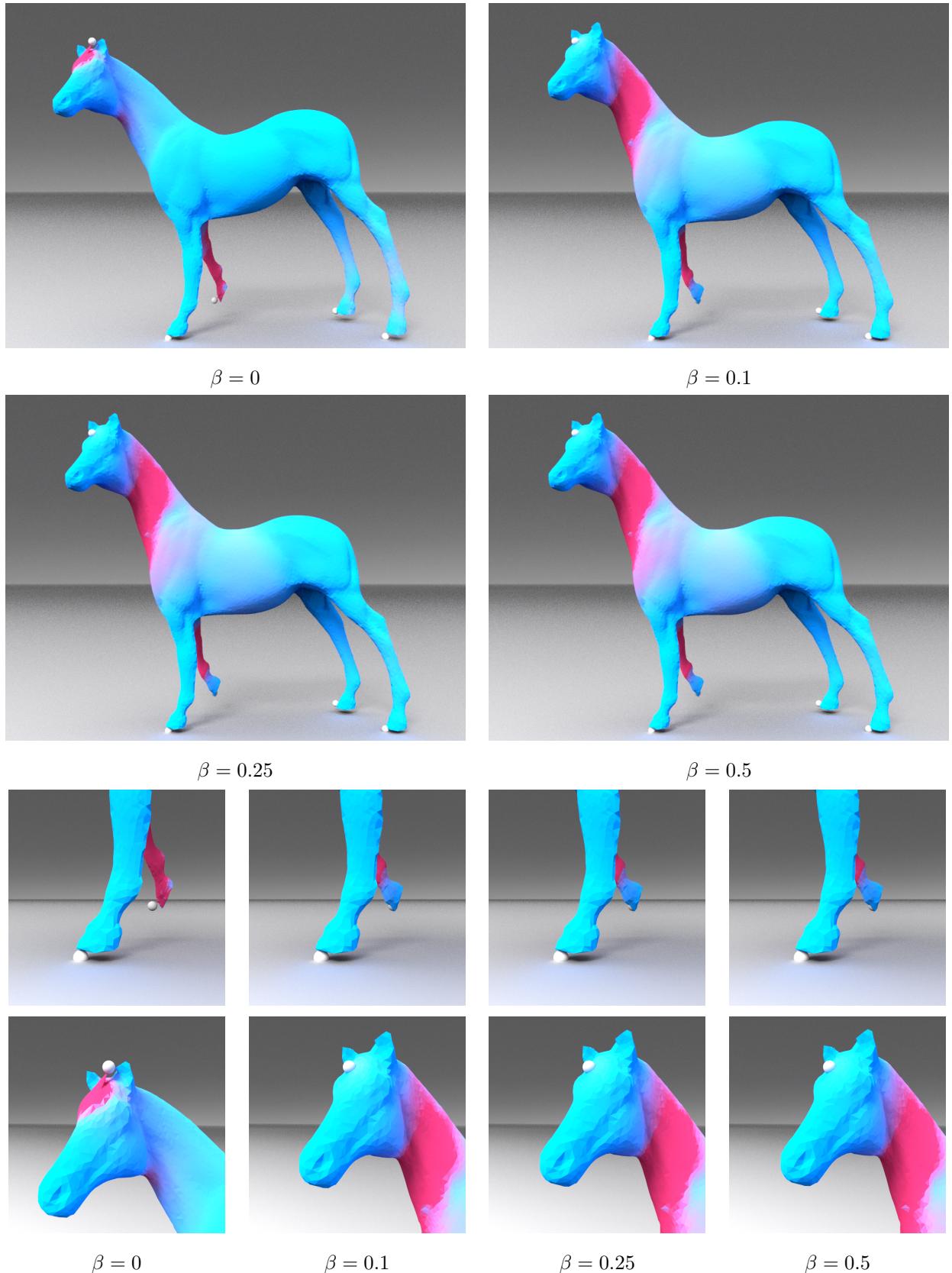


Figure 7.11: *Horse 1*. TLM 0.2. One vertex was constrained on each horse foot and one vertex on the head. The handle on the head and on the front leg were translated and rotated. The spheres represent the constrained vertices. *Top*: Regularization improves the global shape of the horse. The curve of the back is more natural looking, the back leg is less stretched. *Middle row*: Regularization suppresses the protruding triangle on the back foot. The sphere is harder to see in the regularized deformations due to the changes to the mesh. *Bottom row*: The protruding triangle is suppressed by regularization.

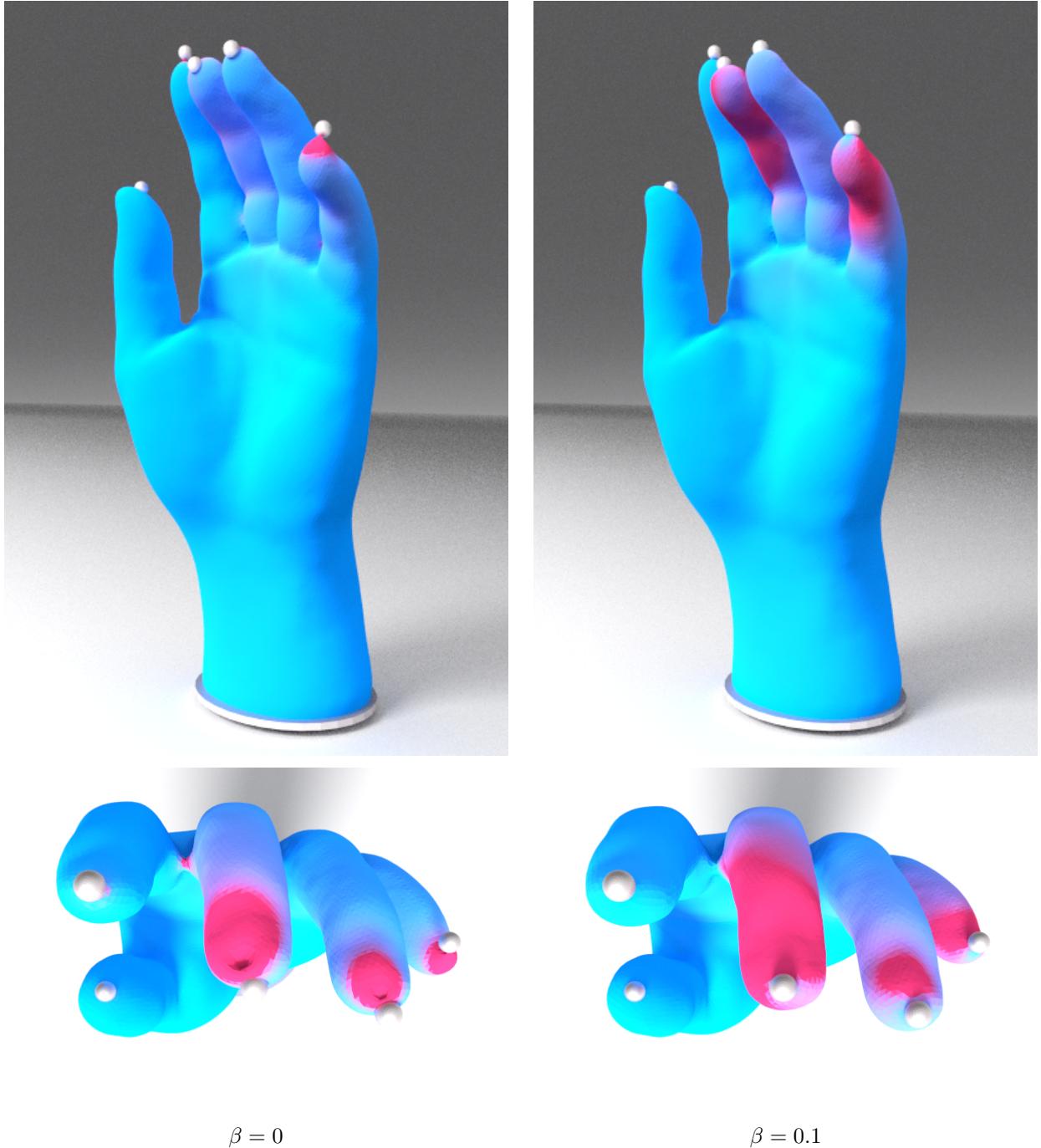


Figure 7.12: *Hand*. TLM at 0.1. A single vertex was constrained on each finger. The base of the hand was fixed. The constrained vertices were transformed by a rotation and a small translation. *Top left*: The deformation has many protruding and intruding elements. *Top right*: Regularization distributed higher energy values over larger mesh regions, creating a smooth surface with no protruding or intruding elements. *Bottom row*: view from above. *Bottom left*: The constrained region intrudes into the model. *Bottom right*: Regularization creates a smooth, artifact free deformation.

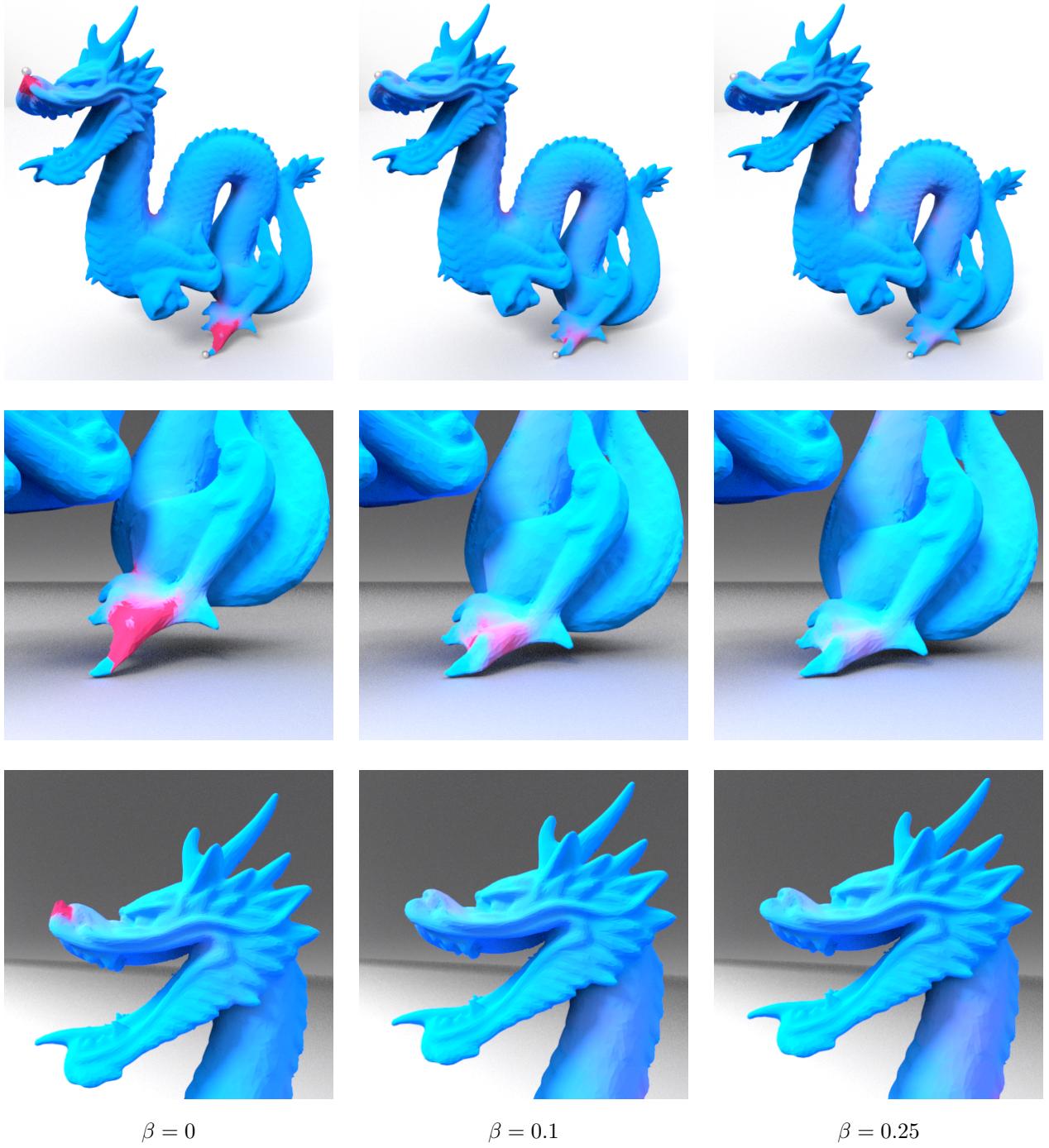


Figure 7.13: *Dragon*. TLM at 0.4. A single vertex on the foot was fixed, the tip of the head was translated upwards and rotated. Regularization improves both the artifacts on the head and foot, and the global shape.

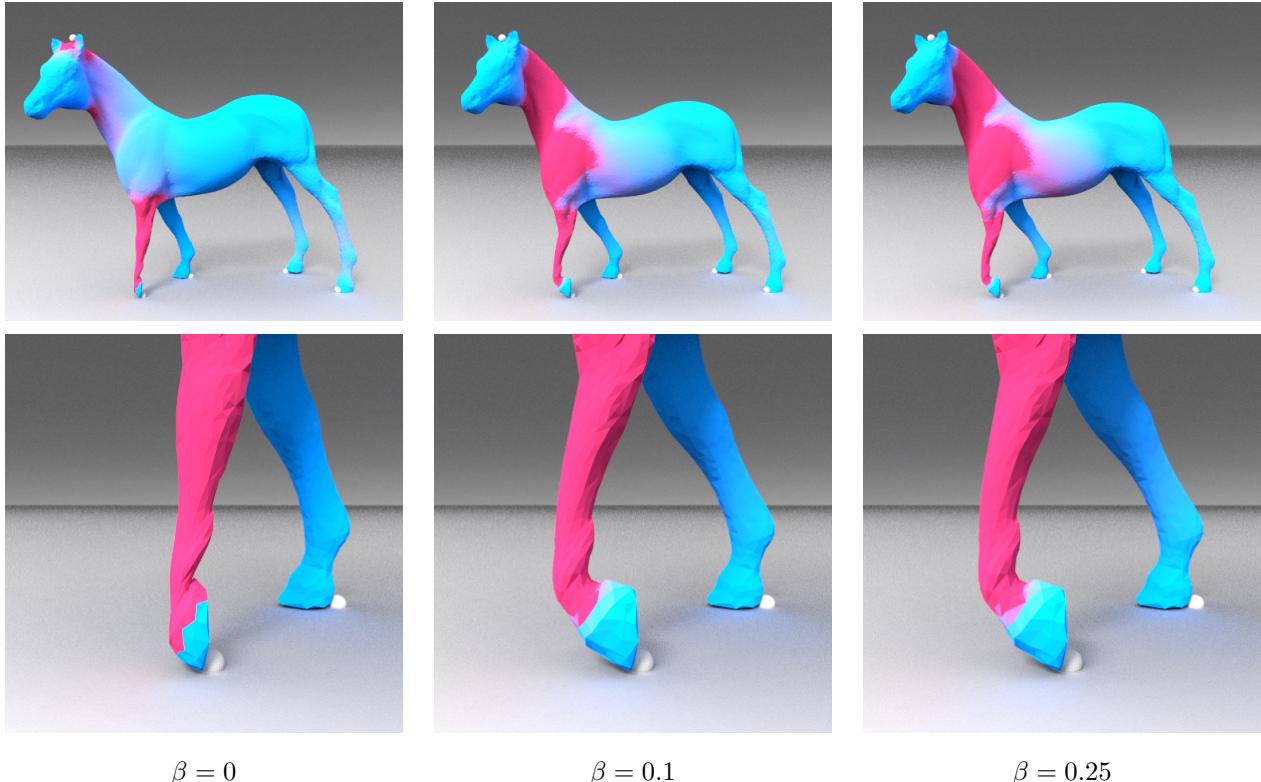

 $\beta = 0$
 $\beta = 0.1$
 $\beta = 0.25$

Figure 7.14: *Horse 2*. TLM at 0.1. The spheres represent the constrained vertices. The handles on the head and on the front foot were rotated. Regularization improves both the artifacts on the head and the entire body position.

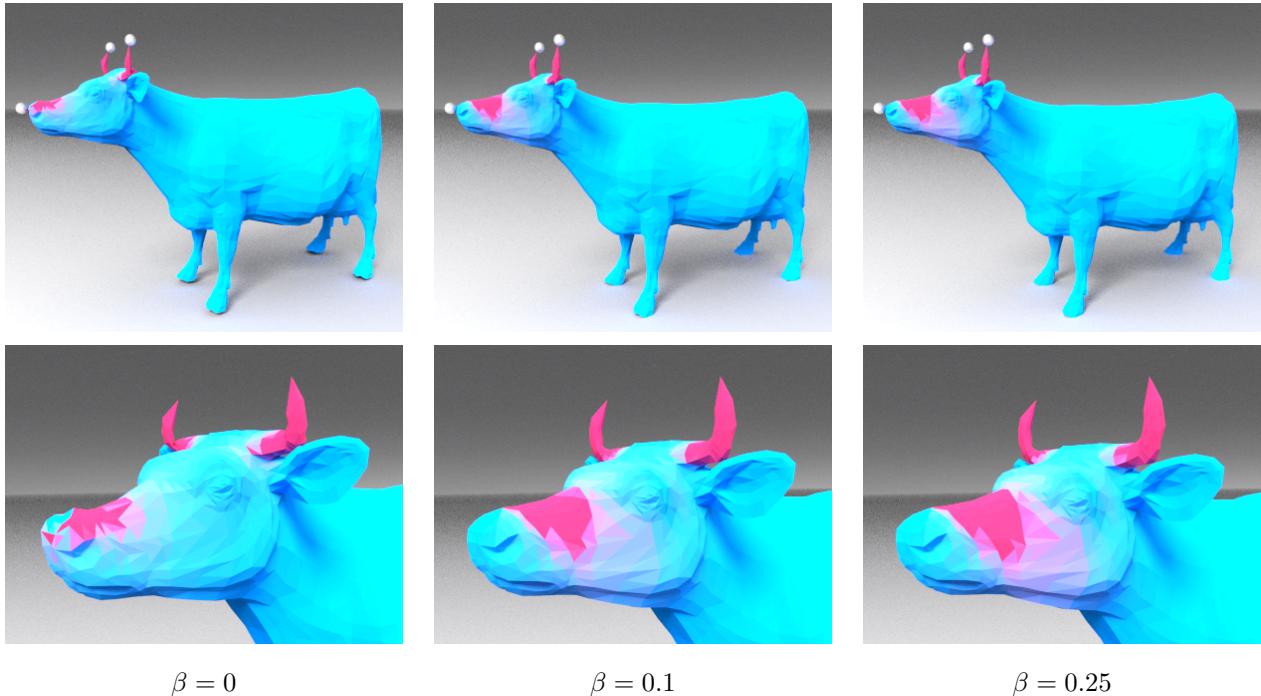
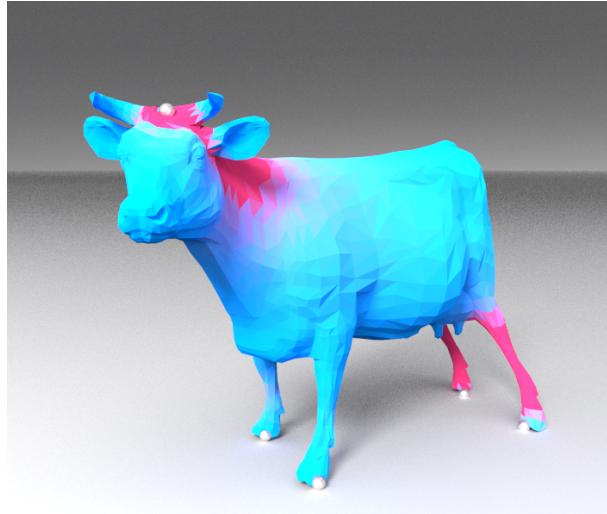
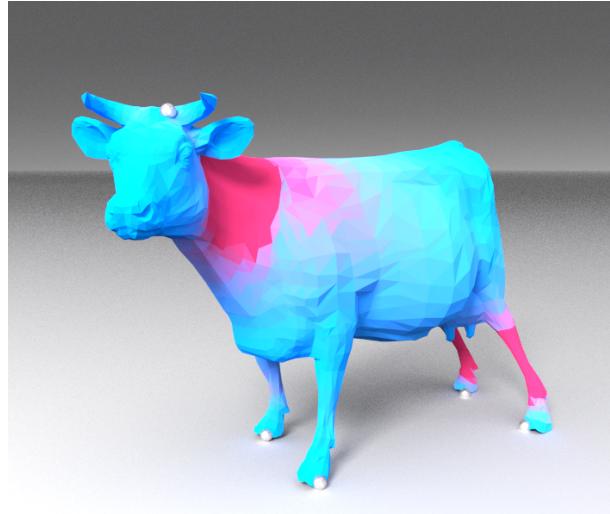

 $\beta = 0$
 $\beta = 0.1$
 $\beta = 0.25$

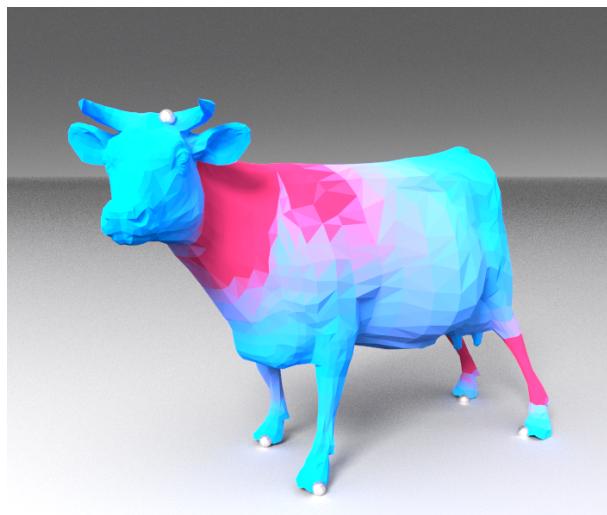
Figure 7.15: *Cow 1*. TLM at 0.2. A single vertex was constrained on each one of horns and the nose. The horns were rotated upwards.



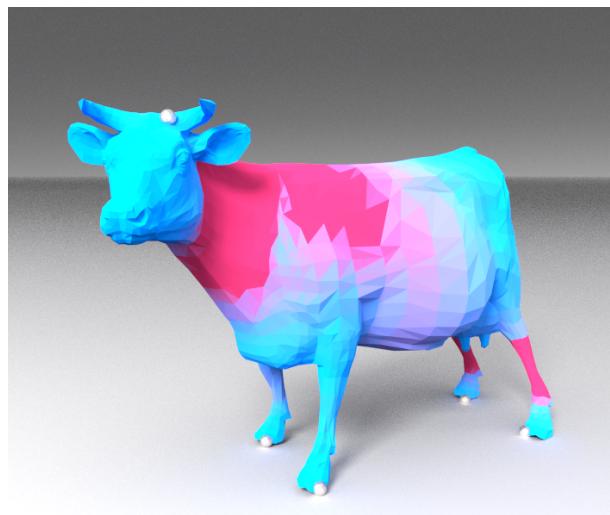
$$\beta = 0$$



$$\beta = 0.1$$



$$\beta = 0.25$$



$$\beta = 0.5$$

Figure 7.16: *Cow 2*. TLM at 0.2. The handle on the head was translated forward and rotated. Regularization improves the global shape by propagating high energies to larger regions and decreasing the surface stretch in the back legs.

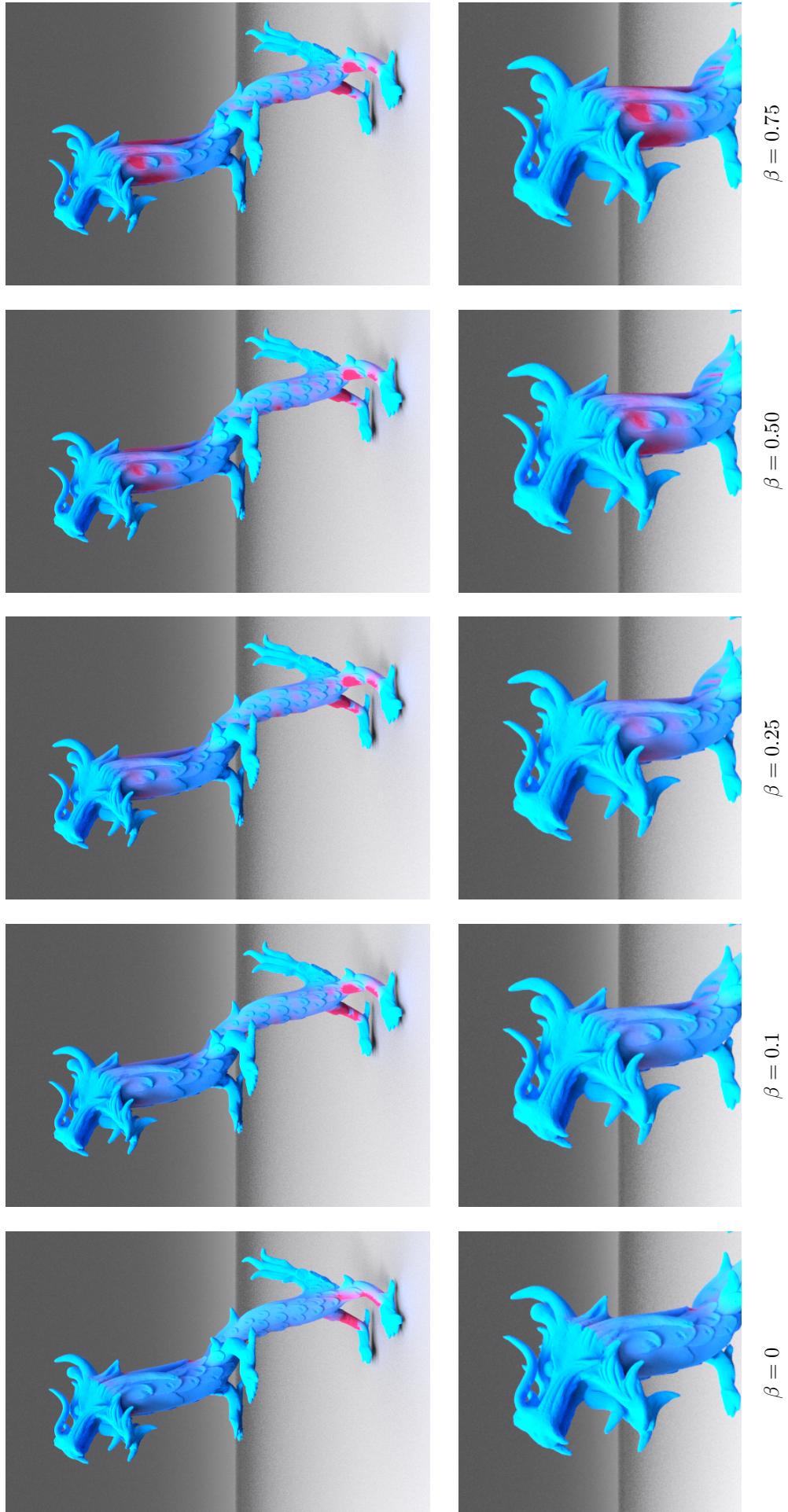
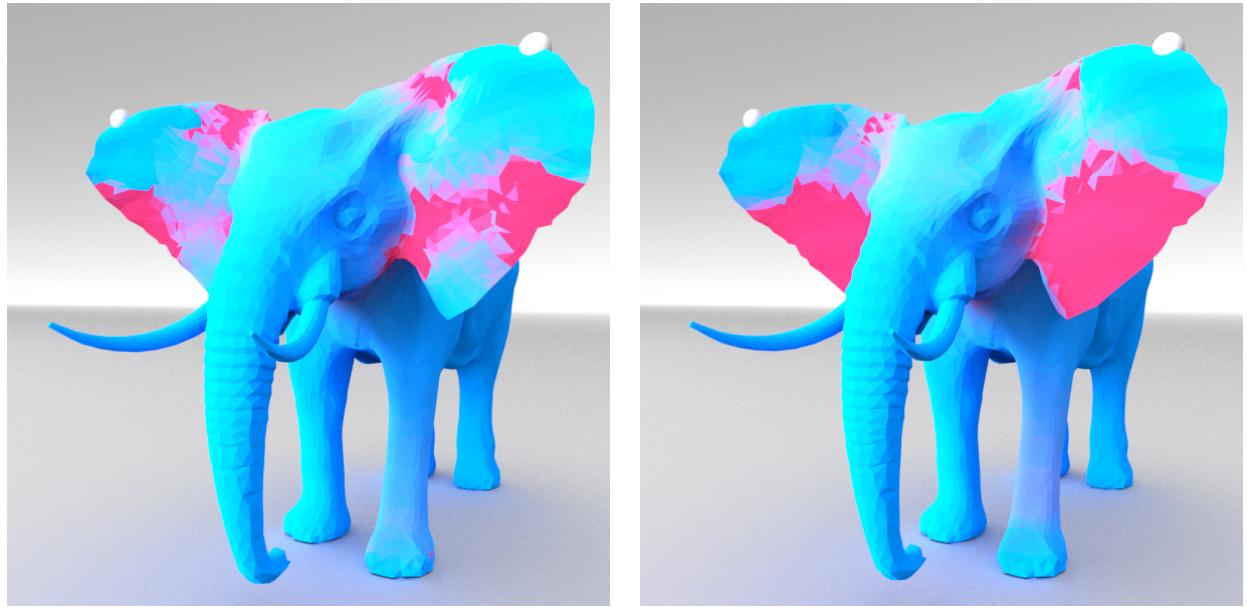


Figure 7.17: *XYZ Dragon*. TLM at 2.5. This is the largest mesh tested with 90K vertices. Both feet were fixed, the entire head was translated and rotated. The boundary between the constrained head vertices and the free vertices is apparent for $\beta = 0$.

Model	β value	$E(\mathbf{x})$	E_{iso}^N	E_{conf}^N	E_{area}^N	$\max(e_{\text{iso}}^i)$	$\max(e_{\text{conf}}^i)$	$\max((e_f^i)^2)$
Horse 1	0.0	0.001643	0.02113	0.01455	1.7	5.4959	3.9605	9.0734
	0.10	0.002416	0.02309	0.01817	2.1	0.4425	0.2838	0.6335
	0.25	0.002612	0.02376	0.01885	2.3	0.4247	0.2708	0.6056
	0.50	0.002713	0.02413	0.01919	2.4	0.4158	0.2644	0.5919
Hand	0.0	0.05438	0.005479	0.003968	-2.5	7.2680	3.6551	12.3276
	0.10	0.06560	0.006093	0.004747	-2.8	0.0987	0.0717	0.3252
	0.25	0.06938	0.006293	0.004984	-2.8	0.0867	0.0823	0.2456
	0.50	0.07320	0.006527	0.005233	-2.7	0.0962	0.0928	0.2050
Dragon	0.0	0.2038	0.021408	0.013200	1.6	10.9326	5.5823	11.2178
	0.10	0.2364	0.027511	0.016635	1.7	0.3505	0.2214	0.4941
	0.25	0.2545	0.030804	0.018573	1.7	0.3727	0.1893	0.4032
Horse 2	0.0	0.001465	0.01662	0.01216	-0.2	1.1066	0.8346	2.0769
	0.10	0.001906	0.01774	0.01412	-0.1	0.4798	0.3452	0.7667
	0.25	0.002040	0.01850	0.01476	0.2	0.4471	0.3295	0.7353
Cow 1	0.0	0.0467	0.003782	0.002435	-0.0	2.6997	2.0470	8.1532
	0.10	0.0565	0.003607	0.002510	-0.3	0.5828	0.5452	3.6573
	0.25	0.0590	0.003773	0.002730	-0.3	0.5810	0.5459	3.6573
Cow 2	0.0	0.1889	0.02078	0.01683	-1.9	1.7137	1.0694	3.5524
	0.10	0.2183	0.01998	0.01696	-2.1	0.4191	0.3979	0.8001
	0.25	0.2412	0.02095	0.01814	-2.1	0.3751	0.3219	0.6517
	0.50	0.2639	0.02227	0.01954	-2.1	0.3747	0.3335	0.6753
	0.75	0.2780	0.02318	0.02048	-2.1	0.3727	0.3392	0.6826
XYZ Dragon	0.0	29.49	0.2264	0.1546	8.583	26.803	13.440	50.676
	0.10	31.10	0.2400	0.1631	8.897	6.408	3.213	6.427
	0.25	32.22	0.2526	0.1700	9.191	5.412	2.707	5.413
	0.50	33.28	0.2660	0.1771	9.470	4.677	2.352	5.029
	0.75	33.94	0.2749	0.1817	9.642	4.325	2.420	5.405

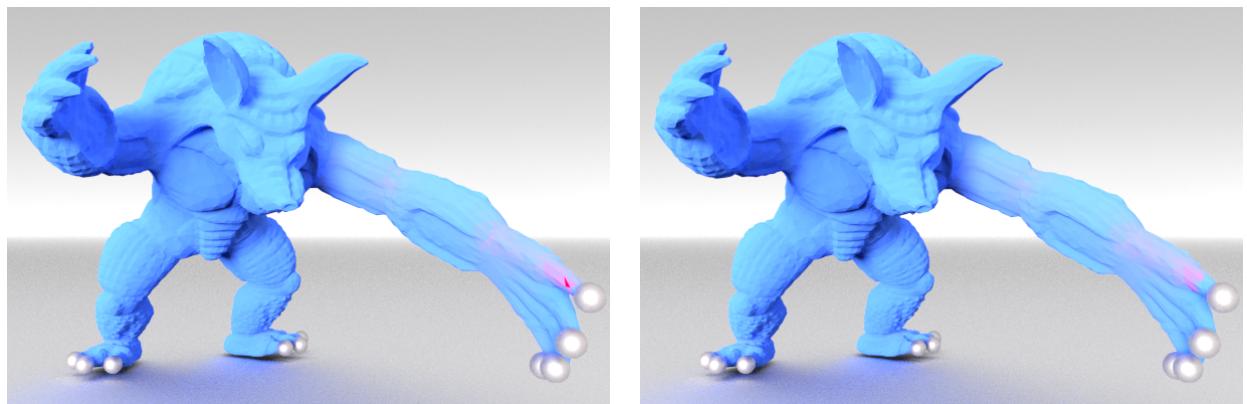
Figure 7.18: Quantitative analysis for Figures 7.11, 7.12, 7.13, 7.14, 7.15, 7.16, 7.17.



$$\beta = 0$$

$$\beta = 0.25$$

Figure 7.19: *Elephant*. Failure case. TLM at 0.1. A few vertices were constrained near each sphere. They were translated and rotated.



$$\beta = 0$$

$$\beta = 0.1$$

Figure 7.20: *Armadillo*. Failure case. Linear mapping. The spheres represent the constrained vertices. The feet were fixed, and the hand was translated and rotated.

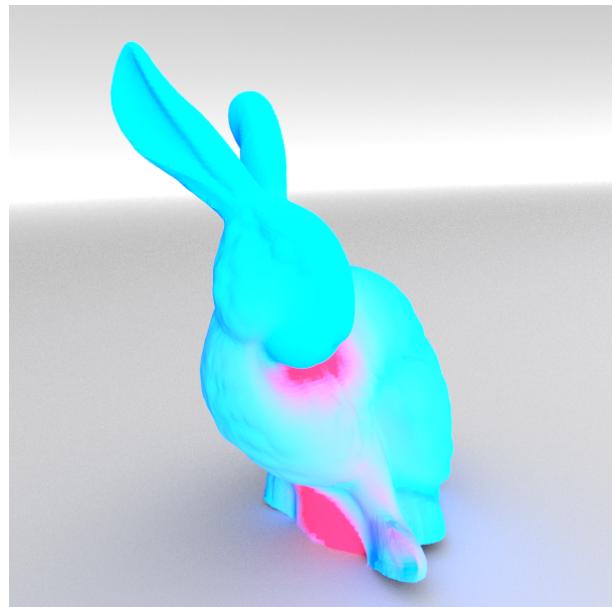
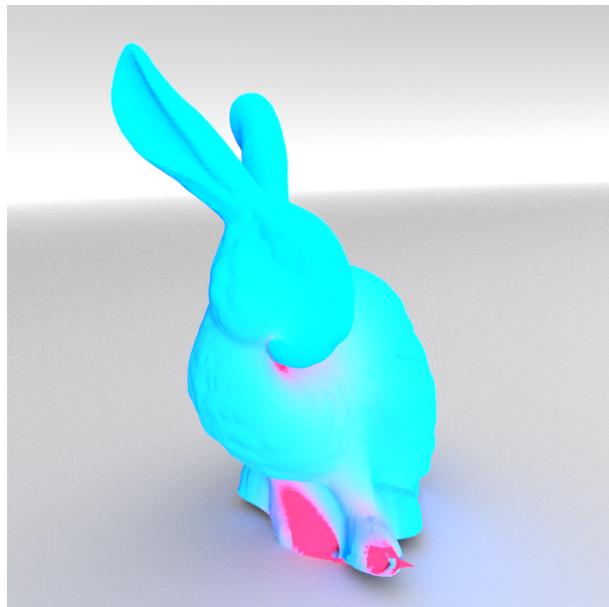
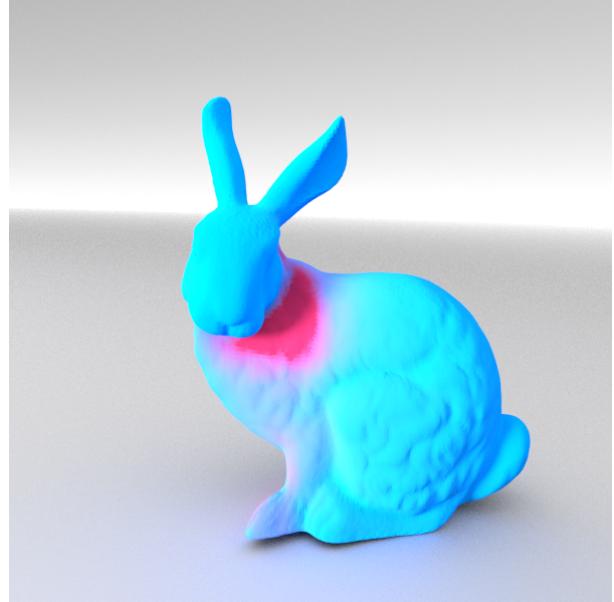
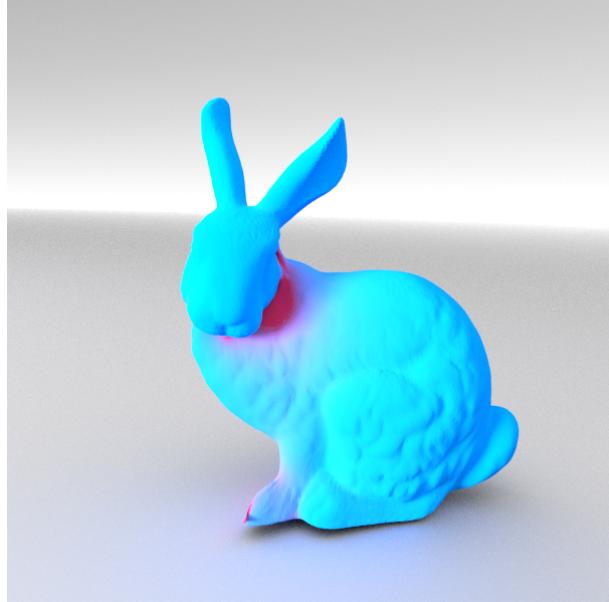
 $\beta = 0$ $\beta = 0.1$

Figure 7.21: *Bunny*. Failure case. TLM at 1.0. The tip of the visible foot as fixed, the heard was set as a deformation handle and rotated. *Top row*: front view. *Bottom row*: side view. The protruding triangle is repressed when regularization is introduced. However, the entire mesh is strongly distorted.

Model	β value	$E(\mathbf{x})$	E_{iso}^N	E_{conf}^N	E_{area}^N	$\max(e_{\text{iso}}^i)$	$\max(e_{\text{conf}}^i)$	$\max((e_f^i)^2)$
Elephant	0.0	0.04256	0.00737	0.005560	-0.3	0.7493	0.6080	1.0275
	0.25	0.05682	0.01017	0.008219	-0.1	0.1719	0.1270	0.3326
	0.75	0.06204	0.01123	0.009132	-0.2	0.1833	0.1532	0.3715
Armadillo	0.0	3103	1800	1119	3.6	2.191	1.1097	2.627
	0.10	3151	1814	1132	3.5	1.426	0.7338	1.501
	0.25	3173	1820	1138	3.5	1.213	0.6194	1.270
	0.50	3205	1829	1148	3.5	1.073	0.5401	1.081
	0.75	3268	1847	1165	3.6	0.977	0.4921	1.010
	0.90	3371	1880	1195	3.8	0.908	0.4564	0.930
Bunny	0.0	0.5393	0.05388	0.03943	-1.8	76.2355	38.2820	83.5667
	0.10	0.6416	0.05855	0.04581	-1.3	0.8990	0.6740	2.2928
	0.25	0.6938	0.06294	0.04925	-1.1	0.9013	0.4978	1.7060
	0.50	0.7488	0.06730	0.05269	-1.0	0.7179	0.4007	1.4464

Figure 7.22: Quantitative analysis for Figures 7.19, 7.20, 7.21.

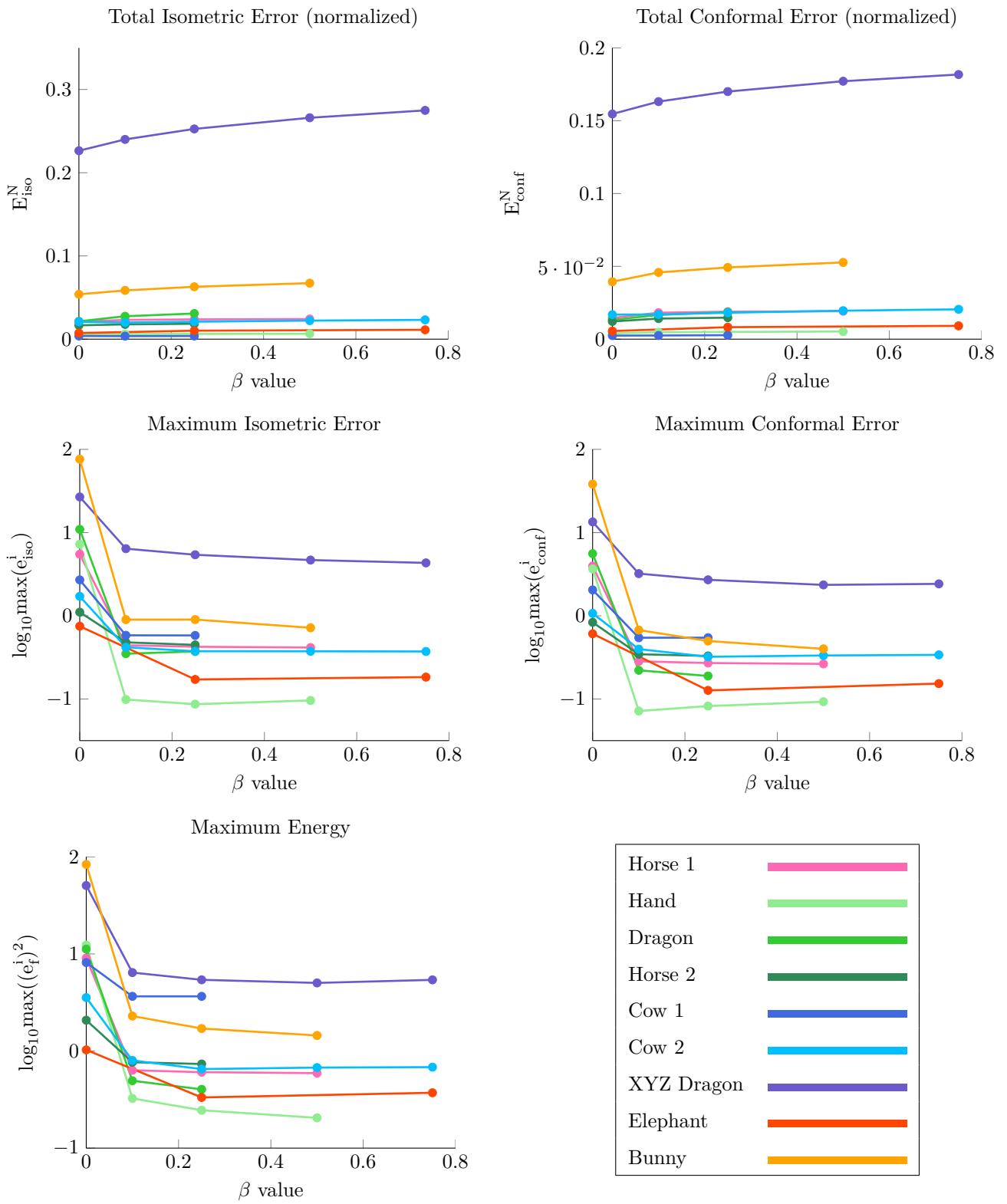


Figure 7.23: Graphs of error measures in Figure 7.18

Chapter 8

Discussion

This chapter is structured as follows: Sections 8.1 and 8.2 discuss how regularization affects the quantitative and qualitative error measures defined previously (Section 7.1). Section 8.3 discusses issues caused by high regularization weight. Section 8.4 discusses how regularization affects artifacts related to handle region size. Section 8.5 compares linear energy regularization to other regularization methods. The advantages and disadvantages of linear energy regularization are summarized in Section 8.6.

8.1 Effects of Regularization on Error Measures

From the deformations shown in Chapter 7, the following patterns emerged for different error measures:

Total deformation energy The total deformation energy always rises with the regularization weight. This is expected, increasing the value of β deviates the regularized system away from the naive system. The energy being minimized is different, and the further the systems deviate from each other, the further the solutions are.

Total normalized isometric and conformal errors For all deformations, the behavior of these two error measures was very similar. Both increased slightly with the value of β .

The regularized deformation energy deviates from minimizing the energy term related to these errors by also optimizing for energy smoothness, thereby increasing the total normalized isometric and conformal errors.

The deformation guidance field is constructed by rigid rotations of the original mesh gradients, making it an isometric and conformal mapping. Solving for this field in the least squares sense introduces an initial residual error. Regularization exaggerates this the deviation from the isometric and conformal mapping.

Total normalized area change For almost all deformation, the total normalized area change slightly increased with β . This does not happen only in four deformations - *Cactus* (Figure 7.3), *Horse 2* (Figure 7.14) where the absolute values of this number decreases, and *Elephant* (Figure 7.19) and *Bunny* (Figure 7.21) where the total normalized area change decreases. In all of these deformations the change in this error measure is insignificant.

Maximum isometric and conformal errors For almost all deformations, these error measures follow the same pattern of an initially strong drop off with the introduction of low regularization weight, then either stabilizing or decreasing slowly as the β value increases. For very high values of $\beta > 0.9$ there might be an increase in this number.

Very high, local and isolated maximum isometric and conformal errors were associated with strong artifacts such as protruding triangles or surface self-intersections. The drop of these errors was associated with artifact suppression. A drop off of up to two or three orders of magnitude for these errors was common, without a significant rise in the total deformation energy.

Maximum local energy Very low use of regularization significantly reduces $\max((e_f^i)^2)$ in all deformations. However, this error measure may rise with high values of β , as the smoothness term overtakes the energy minimization term.

Discussion Overall, the deformation energy was not significantly increased by regularization. The rise in the total deformation energy was within the same order of magnitude of the original deformation.

The total normalized isometric and conformal errors and the total deformation energy rise with β , but the maximum local isometric and conformal errors and the local energy are lower for a certain range of β values.

For deformations with very high local isometric and conformal errors and high local energies on singular triangles, linear energy regularization improves the associated artifacts and reduces these local error measures and high energies without a significant rise in the deformation energy.

The changes in these error measures could be used to find an optimal value for β .

8.2 Effects on Common Deformation Artifacts

Section 1.3 lists common deformation artifacts. Linear energy regularization helps to suppress many of them:

Distortion of Local Geometry In almost all deformations, energy regularization reduced the distortion of local geometric details. This distortion is especially common around boundaries between constrained and free mesh regions. Regularization helps reduce it. Examples include *XYZ Dragon* (Figure 7.17) and the Cactus deformation (Figure 7.7, first row).

Preservation of Topology Global topology was not affected negatively by regularization in comparison to the naive solutions. By reducing triangle flips and local surface self-intersections (see below), regularization helps preserve the mesh topology.

Degenerate and Flipped Triangles Energy regularization can suppress flipped triangles (Figure 7.6). The weight of the regularization depend on how strong the deformation is. For smaller translations of vertices within the plane, lower values of β suppressed these artifacts. Very high values of β actually make these artifacts worse.

Protruding Elements Harmonic deformation proved effective in suppressing protruding and intruding faces. These artifacts appear for translational constraints. Examples include *Cactus Long* (Figure 7.8), *Horse* (Figure 7.11) and *Hand* (Figure 7.12).

Local Self-Intersections These artifacts appear most often around small modeling regions, Figure 7.7 demonstrates how energy regularization is effective for dealing with these artifacts.

Global Self-Intersections Harmonic deformation has no mechanisms to detect and penalize global self intersections. Regularization does not add a mechanism which affects it either. Although none of the deformations displayed here have global self intersections, my experiments show that these artifacts are not suppressed or affected by regularization.

Volume Loss Energy regularization effectively prevents local volume loss. See *Horse 2* (Figure 7.14) and *Cow 1* (Figure 7.15). Volume loss on the global scale was not prevented by linear energy regularization, and in fact, might increase for high values of β (see *XYZ Dragon*, Figure 7.17 and *Cylinder*, Figure 7.2).

Global Shape Preservation Moderate use of energy regularization improves the global shape. Two good example are the *Horse* (Figure 7.11) and *Cow 2* (Figure 7.16). Except for one failure case (*Bunny*, Figure 7.21), the global shape was not negatively affected by introducing regularization.

Discussion Energy regularization effectively suppress many deformation artifacts, both on the local and global scale. Low energy regularization weight did not create new artifacts, but in singular cases suppressing local artifacts came at the price of increasing the global shape distortion. Both global surface self-intersections and global volume loss are not enhanced by energy regularization.

8.3 High Regularization Weight

Very high regularization weight can change the energy distribution in unexpected ways, magnifying local errors and create new artifacts. Examples of the effects of very high regularization weights can be seen in Figures 7.2 and 7.3.

In the *Plane* deformation, where very high value of β introduce sine-wave like distortion into the plane (this was observed for $\beta > 0.95$). The *Cylinder* deformation also changes dramatically for very high values of β . In Figure 7.6, very high regularization actively increase the number of flipped triangles. However, for many other deformations, (Figure 7.8 for example), very high values of β differ little from the deformation for $\beta = 0.25$.

The effects of very high regularization weight depends on the energy distribution on the mesh. It can create unexpected results and increase both the local maximal metric distortion and the global metric distortion, since it changes the dominating factor of the energy functional.

8.4 Handle Region Size

The size of the handle regions affects the types of artifacts seen on the mesh. Naive harmonic surface deformation is not suitable for working with small deformation handles. Common artifacts include local surface intersections (Figure 7.7), protruding elements (Figures 7.11 and 7.12) and local volume loss (Figures 7.16 and 7.14). However, these artifacts are effectively suppressed by low regularization weights.

Large handle regions create globally smoother deformations, but introduce strong changes to the mesh curvature near the boundary between the constrained and free mesh regions (see *Cylinder* in Figure 7.2 and *XYZ Dragon* in Figure 7.17). These are harder to suppress - the boundary is often apparent for high β values. Nonetheless, in many cases regularization reduces the magnitude of this boundary artifact.

By introducing low regularization weight, using small regions as deformation handle becomes a viable option for harmonic surface deformation. Linear energy regularization can also improves the transition between constrained and constrained mesh regions for large deformation handles.

8.5 Comparison to Other Regularization Methods

Regularization methods for geometry processing were discussed in Section 2.4. These methods differ from linear energy regularization in by several ways.

Linear energy regularization can be performed with little impact on the deformation runtime. Lipman [Lip12] and Aigerman and Lipman [AL13] both do not run interactively. Schüller et al. [SKPSH13] can deform meshes with up to 10k vertices interactively, but achieve this by using an iterative optimization scheme.

All three of these method guarantee deformation free from certain artifacts. Linear energy regularization cannot guarantee these properties, and in some cases, fails to generate artifact-free deformations. Schüller et al. [SKPSH13] have better results for flipped triangle suppression on small planar meshes.

However, due to being strongly coupled with an energy term, it can be used for 2-manifolds embedded in \mathbb{R}^3 . The above regularization methods were carefully designed to be applied directly to the elements of the objects, requiring volumetric elements for regularization in \mathbb{R}^3 .

Depending on the purpose of the deformation, linear energy regularization can offer a “cheaper” regularization solution. In case of surface meshes, and specifically, harmonic surface deformation, none of the methods above is suitable.

8.6 Summary

The discussion is summarized by listing the advantages and disadvantages of linear energy regularization for harmonic surface deformation:

Advantages

- Linear energy regularization proved very effective for reducing a variety of artifacts common to harmonic surface deformation, both local and global, with little impact on the run time, and without significant rise to the deformation.
- Most significant reduction of artifacts was observed for $\beta < 0.25$, when energy regularization is first introduced. Some deformation benefited from higher regularization weight, but for many deformations, a low regularization weight was enough to gain most of the benefits without any negative effects.
- Energy regularization can effectively suppress many deformation artifacts associated with small handle regions, thereby giving modelers more freedom in creating deformations. Naive harmonic surface deformation does proved to be ill-suited method for designing deformation with small handle regions.
- Even when artifacts were not completely removed, their magnitude decreases, improving the overall quality of the deformation.
- Failure cases are rare. Even the deformations categorized as failure cases are not straight-out failures, and are better categorized as examples of where the deformation is ineffective. Only one case was observe where low regularization weights worsened the deformation, and even in this case the categorization as failure is disputed, as a strong artifact is suppressed by the regularization.
- Global. There are no restrictions on how the energy is distributed on the mesh. In most cases the global shape was improved by energy regularization.
- Little impact on the deformation run time. Although changing the β weight requires re-factorizing the linear system, this factorization time is insignificant in comparison to the time it takes to construct the deformation guidance field.
- These results make a good case for energy regularization as an effective approach to overcoming some of the inherent limitations of linear gradient domain methods, and specifically, harmonic surface manipulation. Small handles and translation only deformations are two main limitations

of harmonic surface deformations - creating plausible looking deformations which involve them, without regularization, is hard. Even low regularization weight was enough to suppress many artifacts. Adding constant low-weighted regularization to existing, user facing software, can improve the results of these methods without adding complexity.

Disadvantages

- Global. Linear energy regularization does not allow fine control on artifact suppression. Fixing local artifacts can creates global changes to the surface. This also means that sometime it is necessary to use very high regularization weight to get the desired results, because the energy smoothness over the entire domain acts as an energy term. This was especially obvious for deformations with large constrained regions, where achieving a smooth transition between the constrained and free mesh regions requires high regularization weight.
- Linear energy regularization cannot guarantee artifact suppression. While it has proved to be very effective method, it cannot in fact guarantee this. It is unsuitable for deformations where artifacts such as element inversion must be avoided at all costs.
- Choosing very high values of β can introduce artifacts and errors. For singular examples with very strong deformation constraints, choosing moderate to high β values is necessary to suppress artifacts, thereby increasing the possibly of creating new artifacts elsewhere.
- Not all artifacts can be suppressed using this approach. This highly depends on the size of the mesh, the configuration of the deformation handles.
- Choosing an optimal β value might not be straightforward in the case of strong artifacts such as flipped triangles.

Chapter 9

Conclusions

In this work I examined a linear energy regularization method for harmonic surface deformation. I demonstrated that it is a computationally cheap, easy-to-implement approach to regularization, which can substantially improve deformation quality. In most cases, it suppresses deformation artifacts to create plausible, natural looking deformations. At worst, it does very little. Even carefully-designed deformations benefited from some energy regularization.

Although this method cannot make any guarantees on creating artifact free deformations, many artifacts associated with harmonic surface deformation are effectively suppressed. Additionally, energy regularization allows to create artifact free deformations even when constrained mesh regions are small, which previously was a complex task.

If desired, this approach be embedded into harmonic surface deformation by choosing a low regularization weight ($0.1 \leq \beta \leq 0.25$) and applying it uniformly to all deformations, without exposing users to even a single parameter, with little chance of this negatively affecting any deformation results.

As an outlook for future work in this field, the following topics could benefit from further research:

Automatic regularization parameter selection Designing a cost function which uses error measure which are correlated with artifacts, such as $\max(e_{\text{iso}}^i), \max(e_{\text{conf}}^i)$ to find an optimal regularization weight. The parameter space for β can be sampled at intervals to estimate the optimal value.

Regularizing other Poisson methods for geometry processing Many Poisson methods which suffer from artifacts might benefit from linear energy regularization. In particular, MeshIK proposed by Sumner et al. [SZGP05] is a good candidate for such regularization, as it suffers from artifacts such as sharp changes in mesh curvature, as was shown in Figure 2.3.

Local regularization Since most of the artifacts in harmonic surface deformation are localized to areas around the constrained mesh regions, it might be beneficial to introduce local regularization operator, restricted to a mesh region around the deformation handles.

This could give the user a very strong control on the surface created, will allow to use different regularization weights on different mesh regions, and allow to use very high localized regularization weight, without creating unwanted artifacts on the rest of the mesh.

Bibliography

- [ACOL00] Marc Alexa, Daniel Cohen-Or, and David Levin. As-rigid-as-possible shape interpolation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 157–164. ACM Press/Addison-Wesley Publishing Co., 2000.
- [AL13] Noam Aigerman and Yaron Lipman. Injective and bounded distortion mappings in 3d. *ACM Transactions on Graphics*, 32:106, 2013.
- [Ale02] Marc Alexa. Linear combination of transformations. In *ACM Transactions on Graphics*, 2002.
- [Ale03] Marc Alexa. Differential coordinates for local mesh morphing and deformation. *The Visual Computer*, 19(2):105–114, 2003.
- [BBM04] Charles Bloom, J Blow, and C Muratori. Errors and omissions in Marc Alexa’s Linear combination of transformations, 2004.
- [BKP⁺10] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. *Polygon mesh processing*. CRC press, 2010.
- [BPGK06] Mario Botsch, Mark Pauly, Markus Gross, and Leif Kobbelt. Primo: coupled prisms for intuitive surface modeling. In *Symposium on Geometry Processing*, volume 256, pages 11–20, 2006.
- [BS08] Mario Botsch and Olga Sorkine. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics*, 14:213–230, 2008.
- [BSBK02] Mario Botsch, Stephan Steinberg, Stephan Bischoff, and Leif Kobbelt. Openmesh -a generic and efficient polygon mesh data structure. 2002.
- [BSPG06] Mario Botsch, Robert Sumner, Mark Pauly, and Markus Gross. Deformation transfer for detail-preserving surface editing. In *Vision, Modeling & Visualization*, 2006.
- [CL96] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996.
- [DKL98] Erik B Dam, Martin Koch, and Martin Lillholm. *Quaternions, interpolation and animation*. Datalogisk Institut, Københavns Universitet, 1998.
- [FH05] Michael S. Floater and Kai Hormann. Surface parameterization: a tutorial and survey. In *Advances in multiresolution for geometric modelling*, pages 157–186. Springer, 2005.
- [GJ⁺10] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. <http://eigen.tuxfamily.org>, 2010.
- [Haz02] Michiel Hazewinkel. *Encyclopaedia of mathematics*. Springer-Verlag, Berlin New York, 2002.
- [HLS⁺07] Kai Hormann, Bruno Lévy, Alla Sheffer, et al. Mesh parameterization: Theory and practice. 2007.
- [JDD03] Thouis R. Jones, Frédo Durand, and Mathieu Desbrun. Non-iterative, feature-preserving mesh smoothing. In *ACM Transactions on Graphics*, volume 22, pages 943–949. ACM, 2003.

- [JK07] Neel Joshi and David J Kriegman. Shape from varying illumination and viewpoint. In *ICCV*, volume 7, pages 1–7, 2007.
- [KBH06] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006.
- [KCVS98] Leif Kobbelt, Swen Campagna, Jens Vorsatz, and Hans-Peter Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 105–114. ACM, 1998.
- [KS12] Ladislav Kavan and Olga Sorkine. Elasticity-inspired deformers for character articulation. *ACM Transactions on Graphics*, 31(6):196:1–196:8, 2012.
- [LC87] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM Siggraph Computer Graphics*, volume 21, pages 163–169. ACM, 1987.
- [Lip12] Yaron Lipman. Bounded distortion mapping spaces for triangular meshes. *ACM Transactions on Graphics*, 31:108, 2012.
- [LSLCO05] Yaron Lipman, Olga Sorkine, David Levin, and Daniel Cohen-Or. Linear rotation-invariant coordinates for meshes. In *ACM Transactions on Graphics*, volume 24, pages 479–487. ACM, 2005.
- [Mar13] Janick Martinez Esturo. *Shapes in Vector Fields*. PhD thesis, Otto-von-Guericke Universitat Magdeburg, 2013.
- [MK12] Jan Mobius and Leif Kobbelt. Openflipper: An open source geometry processing and rendering framework. In *Curves and Surfaces*. Springer, 2012.
- [MRT14] Janick Martinez Esturo, Christian Ross, and Holger Theisel. Smoothed quadratic energies on meshes. *ACM Transactions on Graphics*, 2014.
- [MSRT13] Janick Martinez Esturo, Maik Schulze, Christian Ross, and Holger Theisel. Poisson-based tools for flow visualization. In *Proc. IEEE Pacific Vis Symposium*, 2013.
- [PGB03] Patrick Perez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Transactions on Graphics*, 22:313–318, 2003.
- [PR05] Yehuda Pinchover and Jacob Rubinstein. *An introduction to partial differential equations*, volume 10. Cambridge University Press, 2005.
- [SA07] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, 2007.
- [SCOL⁺04] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Ross, and H-P Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184. ACM, 2004.
- [SD92] Ken Shoemake and Tom Duff. Matrix animation and polar decomposition. In *Proc. of the Conference on Graphics interface*, 1992.
- [Sho85] Ken Shoemake. Animating rotation with quaternion curves. *ACM Transactions on Graphics*, 19:245–254, 1985.
- [SKPSH13] Christian Schuller, Ladislav Kavan, Daniele Panozzo, and Olga Sorkine-Hornung. Locally injective mappings. In *Computer Graphics Forum*, 2013.
- [Sor06] Olga Sorkine. Differential representations for mesh processing. In *Computer Graphics Forum*, 2006.
- [SP86] Thomas W Sederberg and Scott R Parry. Free-form deformation of solid geometric models. In *ACM Siggraph Computer Graphics*, volume 20, pages 151–160. ACM, 1986.
- [SP04] Robert W Sumner and Jovan Popovi. Deformation transfer for triangle meshes. *ACM Transactions on Graphics*, 23:399–405, 2004.

- [SZGP05] Robert W Sumner, Matthias Zwicker, Craig Gotsman, and Jovan Popović. Mesh-based inverse kinematics. In *ACM Transactions on Graphics*, volume 24, pages 488–495. ACM, 2005.
- [TL94] Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 311–318. ACM, 1994.
- [TPBF87] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. In *ACM Siggraph Computer Graphics*, volume 21, pages 205–214. ACM, 1987.
- [XZWB05] Dong Xu, Hongxin Zhang, Qing Wang, and Hujun Bao. Poisson shape interpolation. In *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pages 267–274. ACM, 2005.
- [YZX⁺04] Yizhou Yu, Kun Zhou, Dong Xu, Xiaohan Shi, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Mesh editing with poisson-based gradient field manipulation. In *ACM Transactions on Graphics*, volume 23, pages 644–651. ACM, 2004.
- [ZRKS05] Rhaleb Zayer, Christian Rössl, Zachi Karni, and Hans-Peter Seidel. Harmonic guidance for surface deformation. In *Computer Graphics Forum*, 2005.
- [ZSS97] Denis Zorin, Peter Schröder, and Wim Sweldens. Interactive multiresolution mesh editing. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 259–268. ACM, 1997.