Injective Mapping under Constraints

Xingyi Du
Washington University in St. Louis

Follow this and additional works at: https://openscholarship.wustl.edu/eng_etds

Recommended Citation
https://openscholarship.wustl.edu/eng_etds/870

This Dissertation is brought to you for free and open access by the McKelvey School of Engineering at Washington University Open Scholarship. It has been accepted for inclusion in McKelvey School of Engineering Theses & Dissertations by an authorized administrator of Washington University Open Scholarship. For more information, please contact digital@wumail.wustl.edu.
WASHINGTON UNIVERSITY IN ST. LOUIS
McKelvey School of Engineering
Department of Computer Science & Engineering

Dissertation Examination Committee:
  Tao Ju, Chair
  Quo-Shin Chi
  Brendan Juba
  Ulugbek Kamilov
  Danny M. Kaufman

Injective Mapping under Constraints
  by
  Xingyi Du

A dissertation presented to
the McKelvey School of Engineering
of Washington University in
partial fulfillment of the
requirements for the degree
of Doctor of Philosophy

August 2023
St. Louis, Missouri
# Table of Contents

List of Figures ................................................................. v
List of Tables ................................................................. xii
Acknowledgments ............................................................. xiii
Abstract .................................................................................. xvi

Chapter 1: Introduction ......................................................... 1
  1.1 Motivation ........................................................................ 1
  1.2 Simplicial meshes and simplicial maps .............................. 2
  1.3 Injectivity criteria ........................................................... 4
    1.3.1 Orientation of a simplex .............................................. 4
    1.3.2 Local injectivity ....................................................... 5
    1.3.3 Global injectivity .................................................... 7
  1.4 Problem statement .......................................................... 7
  1.5 Related works ................................................................... 8
    1.5.1 Tutte's embedding .................................................... 9
    1.5.2 Inversion-free and locally injective maps ...................... 9
    1.5.3 Globally injective maps .......................................... 11
    1.5.4 Remeshing methods ................................................. 11
  1.6 Goals ............................................................................... 12

Chapter 2: Mapping into a Domain with Fixed Boundary .......... 15
  2.1 Introduction ..................................................................... 15
  2.2 Problem statement and Overview ...................................... 18
  2.3 Energy ............................................................................ 19
    2.3.1 Total Unsigned Area ............................................... 19
    2.3.2 Total Lifted Content ............................................... 20
  2.4 Algorithm ......................................................................... 26
  2.5 Results ............................................................................ 28
    2.5.1 Parameter choices ................................................... 28
    2.5.2 A benchmark problem set ....................................... 29
    2.5.3 Benchmark comparisons ........................................ 31
2.5.4 Comparison with other methods ........................................... 37
2.5.5 Impact of initial embedding .................................................. 38
2.5.6 Initializing distortion minimization ......................................... 39

2.6 Discussion .............................................................................. 40

Chapter 3: Mapping with Positional Constraints .......................... 42
3.1 Introduction ............................................................................ 42
3.2 Problem statement and Overview ............................................ 45
3.3 Energy .................................................................................... 46
  3.3.1 Excess Area ....................................................................... 46
  3.3.2 Smooth Excess Area ............................................................ 48
3.4 Algorithm ............................................................................... 52
3.5 Results ................................................................................... 54
  3.5.1 Benchmark ......................................................................... 55
  3.5.2 SEA optimization behavior .................................................. 58
  3.5.3 Impact of initialization ........................................................... 62
  3.5.4 Minimizing distortion ............................................................ 63
3.6 Discussion ............................................................................... 64

Chapter 4: Reducing Map Distortion .......................................... 67
4.1 Introduction ............................................................................ 67
4.2 Method ................................................................................... 70
  4.2.1 Preliminaries ....................................................................... 70
  4.2.2 Isometric TLC .................................................................... 72
  4.2.3 Isometric SEA ..................................................................... 75
  4.2.4 Optimization ....................................................................... 77
4.3 Results ................................................................................... 77
  4.3.1 Fixed-boundary mapping benchmark .................................... 79
  4.3.2 Free-boundary mapping benchmark ...................................... 79
  4.3.3 Performance ........................................................................ 82
  4.3.4 Initializing map optimization ............................................... 83
  4.3.5 Failure cases of IsoTLC ......................................................... 84
  4.3.6 Handling poor triangulations ............................................... 86
4.4 Discussion ............................................................................... 87

Chapter 5: Conclusion ................................................................. 89
5.1 Summary ............................................................................... 89
5.2 Future Directions .................................................................... 90
  5.2.1 Robust injective mapping under constraints ........................... 90
  5.2.2 Generalizing our work ............................................................ 91

References .................................................................................. 93
## Appendix A  Total Lifted Content

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1</td>
<td>Lifted Content and Dirichlet energy</td>
<td>99</td>
</tr>
<tr>
<td>A.1.1</td>
<td>The 2D case</td>
<td>99</td>
</tr>
<tr>
<td>A.1.2</td>
<td>The 3D case</td>
<td>101</td>
</tr>
<tr>
<td>A.2</td>
<td>Injectivity of global minima</td>
<td>102</td>
</tr>
<tr>
<td>A.2.1</td>
<td>The 2D case</td>
<td>102</td>
</tr>
<tr>
<td>A.2.2</td>
<td>3D</td>
<td>107</td>
</tr>
<tr>
<td>A.3</td>
<td>Gradient and Hessian of TLC</td>
<td>111</td>
</tr>
<tr>
<td>A.3.1</td>
<td>Gradient</td>
<td>112</td>
</tr>
<tr>
<td>A.3.2</td>
<td>Hessian</td>
<td>113</td>
</tr>
</tbody>
</table>

## Appendix B  Smooth Excess Area

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.1</td>
<td>Excess, overlap, and inverted areas</td>
<td>114</td>
</tr>
<tr>
<td>B.2</td>
<td>Continuity and smoothness of occupancy</td>
<td>115</td>
</tr>
<tr>
<td>B.3</td>
<td>SEA and excess area</td>
<td>120</td>
</tr>
<tr>
<td>B.4</td>
<td>Injectivity at SEA minima</td>
<td>122</td>
</tr>
</tbody>
</table>

## Appendix C  Isometric Energies

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.1</td>
<td>Singular value formula for lifted content</td>
<td>128</td>
</tr>
<tr>
<td>C.2</td>
<td>Minimizer of residue functions</td>
<td>129</td>
</tr>
<tr>
<td>C.3</td>
<td>Injectivity of IsoTLC minimizer</td>
<td>132</td>
</tr>
<tr>
<td>C.4</td>
<td>Injectivity of IsoSEA minimizer</td>
<td>136</td>
</tr>
<tr>
<td>C.5</td>
<td>Implementation details</td>
<td>140</td>
</tr>
<tr>
<td>C.5.1</td>
<td>IsoTLC</td>
<td>140</td>
</tr>
<tr>
<td>C.5.2</td>
<td>IsoSEA</td>
<td>146</td>
</tr>
</tbody>
</table>
### List of Figures

Figure 1.1: Left: simplicial map is affine within each simplex of the source mesh. Right: the source mesh is mapped to another simplicial mesh, called the target mesh. ................................................................. 3

Figure 1.2: Left: a map preserves orientation if the target simplex has positive signed volume. Right: orientation is inverted if the target simplex has negative signed volume (shown in red). ................................................................. 5

Figure 1.3: A locally injective map must be free of three configurations in the target mesh: (a) degenerate triangle (b) inconsistently oriented triangles shared by an edge (c) overwound vertex (gray vertex bellow). .............................. 6

Figure 2.1: Injectively mapping a complex surface mesh (Lucy, 48K vertices) to a non-convex boundary (letter “G”, with zoom-ins), on the left, and mapping a tetrahedral mesh (Armadillo, 6K vertices) to a highly deformed target surface, on the right, as a result of minimizing our novel energy. These two examples are part of our new benchmark data set. ............................ 16

Figure 2.2: Lifting a triangle \( t \) with vertices \( \{x_i, y_i\} \) \( (i = 1, 2, 3) \) to a triangle \( \tilde{t} \) in 4D via the \( \sqrt{\alpha} \)-scaled auxiliary triangle \( \hat{t} \) with vertices \( \{u_i, v_i\} \). ................................................................. 20

Figure 2.3: Given a triangle \( t \) and an equilateral auxiliary triangle \( \tilde{t} \) (top), (a,b) plot the lifted content of \( t \), \( E_{\tilde{t},\alpha}(t) \), at different \( \alpha \) values as the red vertex in \( t \) moves along the blue dotted line in (a) and the green dotted line in (b). (c) plots \( E_{\hat{t},\alpha}(t) \) as the red vertex moves over the plane. ................................................................. 22

Figure 2.4: Left: Rest mesh (Bunny) and MIPS mapping into a target boundary (outline of letter “A”). Middle: Embeddings minimizing uniform-TLC (top) and rest-TLC (bottom) energy at various values of \( \alpha \). Right: Tutte and harmonic mapping of the rest mesh into the same boundary. Inverted triangles are colored red. Note the similarity between the two embeddings highlighted in boxes with the same color. ................................................................. 25
Figure 2.5: Far-left: List of hand-made examples in the 2D parameterization category, each indicated by an arrow that goes from the rest mesh to the target domain. Middle: Three examples (top, middle, and bottom rows) from the list (highlighted in dotted lines), each showing Tutte embedding, result of minimizing TLC (stopped at injectivity), minimizing TLC (stopped at convergence), minimizing total unsigned area (TUA), and results of LBD, SA and FF. All inverted triangles and their vertices are colored red, and the total number of inverted triangles is marked in red.

Figure 2.6: Top: rest meshes and letter-like boundaries used to create the embedding problems in the 2D parameterization category. Bottom: results of various methods on embedding Lucy into letter “G”, including our QN solver at different iterations. The zoom-in takes a close look at the inverted triangles in the result of FF (all close to being collinear).

Figure 2.7: Four examples in the 2D parameterization category derived from [47], where methods FF and LBD failed to find injective embeddings.

Figure 2.8: Three examples from the 3D parameterization category, each mapping a rest tetrahedral mesh into a sphere (top), smooth surface (middle), and a polycube (bottom). Each example is a failure case for one of the three methods, FF, LBD and SA. Inverted tetrahedra are colored in red, and the numbers of inversion are marked in red.

Figure 2.9: One example in the 3D deformation category (a twisting rod). All methods succeeded on this sequence.

Figure 2.10: Two example in the 3D deformation category (a twisting cube) where both SA and LBD failed to reach injectivity. The graph in the top-right shows the number of inverted tetrahedra for each of the 118 frames of the deformation sequence (ellipses indicate the frames from which the two examples were taken).

Figure 2.11: Three example in the 3D deformation category (a twisting armadillo) where FF, SA and LBD all failed to reach injectivity. The graph in the top-right shows the number of inverted tetrahedra for each of the 600+ frames of the deformation sequence (ellipses indicate the frames from which the three examples were taken).
Figure 2.12: Top: number of iterations taken by our PN or QN solvers over all benchmark examples in 2D (a) and 3D (b), where the horizontal axis is the ratio of the number of inverted simplices in the initial (Tutte) embedding over the total number of simplices (both axes are in log-scale). In the legend, “all pass” means that FF, LBD, and SA (3D) all succeed on that example; “some pass” means that at least one of these methods fails (our method succeeds on all examples). Bottom: running time (in seconds) of all methods in 2D (c) and 3D (d), where the horizontal axis is the number of vertices in the rest mesh (running time is in log scale).  

Figure 2.13: Mapping a triangulated Hele-Shaw polygon (from [63]) to a square. Tutte embedding contains 46 inverted or co-linear triangles due to numerical errors (they are all along the boundary). Our method (QN solver) successfully restores the injectivity in 65 iterations.  

Figure 2.14: Mapping a Bunny shape (same as Figure 2.4) into a cross with three different initial embeddings, Tutte embedding (top), harmonic map (middle), and one in which all interior vertices are collapsed onto a single point (bottom). Showing results of our method after terminating upon injectivity (middle) and after the energy has converged (right).  

Figure 2.15: Mapping a multi-boundary mesh using an initialization that has all vertices collapsed to a single point. The QN solver fails to reach injectivity within allowed number of iterations, while PN succeeds.  

Figure 2.16: Minimizing isometric distortion (right) starting from an injective mapping of Lucy into the letter “P” provided by our method (left).  

Figure 3.1: Given a non-injective initial parameterization of a surface mesh (left) with inverted triangles (red), boundary intersections (orange dots), and over-wound vertices (magenta dots), our method recovers a globally injective map (right) while keeping the constraints (blue points) in place. The inserts zoom in on one region in the initial map with many boundary intersections and inverted triangles (green box) and another region with an overwound vertex (cyan box).  

Figure 3.2: Winding numbers for different curves. Occupancy of each curve is the area of the gray regions.  

Figure 3.3: Top: a non-injective map with overlapping area (yellow) around a boundary vertex (green) transitions to an injective map, going through a configuration where two incident boundary edges overlap (middle). Bottom: the same transition but the straight edges are replaced with their arc-edges; note that the arcs do not overlap during the process.
Figure 3.4: An arc-edge (left) and two overlapping arc-edges (right). .......................... 50

Figure 3.5: The area of a region bounded by arcs (a) is the sum of the signed area of the oriented chords (b) and the signed areas bounded by each pair of oriented arc and its chord in reverse orientation (c). Positive and negative areas in (b,c) are colored green and red. .......................... 54

Figure 3.6: Benchmark creation process: starting from an open surface mesh $M$ (a) with a globally injective map $\Phi$ to the plane (b) from the [14] dataset, we randomly select vertices in $\Phi$ to be constraints (blue dots in (b)) and compute a new map $\Phi'$ from $M$ using ARAP that meets those constraints (c). If (as here) the resulting map $\Phi'$ is not injective it is added to our benchmark. .......................... 56

Figure 3.7: Three successful examples from the benchmark: initial maps (first column), maps produced by our method (second column), which are all globally injective, and maps produced by LBD (third column) and SA (last column), none of which are locally or globally injective. .......................... 57

Figure 3.8: Two failure examples from the benchmark: initial maps (first column) and non-globally-injective maps produced by our method (second column). .......................... 59

Figure 3.9: Performance: (a) log running time in seconds (vertical axis) versus vertex count (horizontal) for all benchmark examples. Each dot represents one mesh and is colored by number of QN iterations (gray dots are fails). (b) Ratio of arrangement computation time over the remaining time (vertical axis) versus ratio of squared number of boundary vertices over total number of vertices (horizontal axis) for all successful examples. (c,d) The initial and optimized maps in our slowest successful example (largest red dot in (a)). .......................... 60

Figure 3.10: (a): A mesh with two constraint vertices. (b): An initial map containing inverted triangles. (c,d): Results of optimizing for SEA with occupancy vs. with arc-occupancy. (e): Energy plot of the two energy during optimization. .......................... 61

Figure 3.11: Intermediate results of optimizing SEA energy for the example in Figure 3.10 at $\theta = 1.0$. Arc-edges are colored green. The energy converges at iteration 763 and produces the same map as in Figure 3.10 (d). .......................... 62

Figure 3.12: Two initial maps with overwound vertices (top) and the injective maps produced by our method (bottom). .......................... 63
Figure 3.13: From left to right: input meshes (with one or two boundaries), non-injective initial maps, globally injective maps produced by our method (SEA), and results of LBD and SA. Both LBD and SA have removed most or all inverted triangles, but the results are neither locally or globally injective due to overwound vertices and boundary intersections.

Figure 3.14: The “crossing-arm” configuration: initial maps with increasing extent of crossing (top row, from left to right), and maps produced by our method (middle row) where only the first two are globally injective. The energy plots are shown at the bottom.

Figure 3.15: Changing initialization type: results of our method on two alternate initializations of the hand mesh in Figure 3.7 while applying the same constraints. Injectivity is achieved starting from the top initial map but not for the bottom one (green boxes highlight the inverted triangles).

Figure 3.16: Left: output of an SEA-optimized injective map for the hand mesh in Figure 3.7. Right: further minimizing the SEA-generated map with the Symmetric Dirichlet (SD) distortion energy via the method of [35], which preserves global injectivity and point constraints. Triangles are colored by their SD distortion.

Figure 4.1: When used to parameterize meshes into the 2D plane, our isometric variants of injectivity energies empirically tend to yield injective results with low isometric distortion. Left: an initial map of Lucy into letter S, which contains many inverted triangles (red), is optimized while holding its boundary fixed, using the TLC energy [14], the Fold-Free Mapping (FFM) method of [23], and our energy - IsoTLC. Right: an initial map, which contains inverted triangles (red), overwinding vertices (purple) and global overlaps (boxed), is optimized while fixing only a set of sparse positional constraints (blue) without constraining the boundary, by minimizing either the SEA energy of [14] or our energy, IsoSEA. In both cases, our method leads to the injective map with the lowest isometric distortion (max(σ₁, 1/σ₂)) of the triangles, as shown by the histograms and the color map on the mesh.

Figure 4.2: Visualizing R(t) (a), R^{iso}(t) at different α values (b,c), and the penalty energy of [23] at different ε (d,e,f) as functions of the singular values σ₁, σ₂ of the linear transform L from ̃t to t (σ₁ is given the sign of det(L)). Observe that R(t) is minimized by a similarity transform (σ₁ = σ₂), R^{iso}(t) is minimized by an isometry (σ₁ = σ₂ = 1) regardless of α, and the minimizer of the penalty energy drifts away from isometry as ε increases.
Figure 4.3: Optimizing an initial map of the Bunny mesh into a non-convex boundary by minimizing (till convergence) the TLC energy and our IsoTLC variant (at different $\alpha$). The histograms show the triangle distortions using the measure $\max(\sigma_1, 1/\sigma_2)$ (assuming $\sigma_1 \geq \sigma_2$). ........................................ 75

Figure 4.4: Histograms (in log-log scale) of maximum (top) and average (bottom) distortion of maps computed by different methods on the fixed-boundary 2D (a) and 3D (b) benchmarks and the free-boundary benchmark (c) (only including examples where both SEA and IsoSEA succeeded in recovering injectivity). ........................................ 78

Figure 4.5: Comparing maps computed by TLC [14], FFM [23], and our method (IsoTLC) on several 2D examples in the fixed-boundary benchmark of [14]. Each example consists of a rest mesh and an initial map containing inverted triangles (red). Histograms of per-element distortion are shown using the distortion measure $\max(\sigma_1, 1/\sigma_2)$. ........................................ 80

Figure 4.6: Examples from the fixed-boundary benchmark where SEA succeeded in recovering injectivity from the initial map but IsoSEA failed (top) or vice versa (bottom). Top inserts: a region where IsoSEA produces boundary intersections (orange). Bottom inserts: a region where SEA results in inverted triangles (red). ........................................ 81

Figure 4.7: Comparing maps computed by SEA [15] and our method (IsoSEA) on several examples in the free-boundary benchmark of [15]. Each example consists of a rest mesh, a set of constrained vertices (blue), and an initial map containing inverted triangles (red), overwound vertices (purple), and global overlaps. Histograms of per-element distortion are shown using the distortion measure $\max(\sigma_1, 1/\sigma_2)$. ........................................ 82

Figure 4.8: Running time (seconds in log scale) versus mesh size (number of vertices) for all examples in the fixed-boundary 2D and 3D benchmarks and the free-boundary benchmark. Orange dots are times of the entire algorithm while blue dots are times of the first stage (computing injective maps). 83

Figure 4.9: Comparing maps produced by minimizing TLC and IsoTLC (mapping Lucy to letter G) as initialization for constrained optimization in BCQN [78] that minimizes the Symmetric Dirichlet energy. With the TLC map as the starting point, BCQN terminates in 4 iterations and the map remains highly distorted. ........................................ 84
Figure 4.10: Two fixed-boundary mapping examples (top and bottom) taken from [75] where an injective map does not exist for the given target boundary (without changing the mesh structure). Our IsoTLC method cannot produce injective maps for such problems. 85

Figure 4.11: Two fixed-boundary examples (top and bottom), mapping the “Lucy” mesh to the outline of letter “P”, initialized with two random maps. The top example is taken from [23]. IsoTLC succeeds in the top example but fails to produce an injective map within 10000 iterations (using either QN or PN) on the bottom one. TLC fails on both examples. 86

Figure 4.12: Mapping a tetrahedral mesh bounded by two cubical surfaces after rotating the inner cube by various degrees (clockwise as viewed from the top). IsoTLC produces injective maps for rotations by 45°, 90° and 180° but fails for 135°. TLC succeeds only on 45° and 90°. We note the number of solver iterations (and solver type) for each method (iterations for the two phases of optimizing IsoTLC are shown separately). 87

Figure 4.13: Fixed-boundary mapping of meshes with poorly shaped triangles from Thingi10K. Both TLC and IsoTLC succeeded in producing injective maps, while IsoTLC achieved much lower distortion. 88

Figure A.1: Illustration for Lemma 27. 103

Figure A.2: Illustration for proof of Lemma 16. 104

Figure A.3: Illustration for proof of Lemma 20. 109

Figure B.1: Comparing singular and non-singular maps. 117

Figure B.2: Illustration for the proof. Left: the mapped curves (dotted lines) and the corresponding boundaries of positively winding regions (solid lines) at different parameters (distinguished by shades of blue). Middle and right: the regions swept by the regular and non-regular subsets of $B_g$, shaded respectively in green and red, as $g$ increases from 0 to $h$. We will show that the (signed) area of both green and red regions admits a well-defined derivative at $h = 0$. 118

Figure B.3: Illustration for the proof. 124
List of Tables

Table 2.1: Number of failed examples for each compared method in each category of the benchmark problem set. ............................. 34

Table 3.1: Statistics on number of benchmark examples (out of 1791) that each method succeeds in achieving respectively (per row) an inversion-free, locally injective, or globally injective map. ............................. 58
Acknowledgments

My PhD journey has been a rewarding and transformative experience, and I am deeply grateful to everyone who has contributed to my growth and success.

I would like to express my profound gratitude to my advisor, Professor Tao Ju, whose research work inspired me even before I began my PhD journey. His enthusiasm for research and hands-on approach served as a role model that has deeply influenced my own approach to academic work.

Professor Ju’s research approach has provided me with a strong foundation for tackling complex problems. He taught me the importance of studying basic examples, being sensitive to experimental phenomena, forming verifiable hypotheses, and developing algorithms using a prototyping approach. His optimism and belief in obtaining provable results have been truly inspiring.

I am incredibly grateful for the guidance Professor Ju has provided, from offering detailed feedback on research projects and academic presentations to sharing career advice and helping me develop essential research skills. His support and mentorship have been instrumental in my growth as a researcher, and I am truly fortunate to have had the opportunity to learn from him.

I would also like to express my heartfelt appreciation to my collaborators, including Danny M. Kaufman, Qingnan Zhou, Noam Aigerman, Shahar Kovalsky, Nathan Carr and Yajie Yan, who have supported me in various aspects throughout my PhD journey, including brainstorming research ideas, providing feedback, offering valuable suggestions, and connecting me with other researchers. Their passion and enthusiasm for graphics research have always encouraged me to pursue high targets.

I am grateful to Danny for his deep insights on numerical optimization, physics-based animation, and valuable career advice. His suggestions have been instrumental in overcoming challenging research problems.
My sincere thanks go to James, who has been an exceptional mentor during my internships and a supportive collaborator in coding, research, and job searching. His career advice, research philosophy, and assistance in connecting with other researchers have been invaluable.

I am thankful to Noam for his expertise in the mapping research area, his open-hearted feedback, and his focus on impactful research directions. His support has helped me address important and challenging aspects of my research.

My gratitude also extends to Shahar for his innovative ideas that have helped me tackle some of the most difficult problems in my research, and to Nathan for his high-level insights and valuable feedback.

I would like to express special thanks to Yajie for being a supportive peer and friend throughout my PhD journey. Our open conversations about life and research have been incredibly helpful, and his assistance in running experiments, exploring new research possibilities, and providing career advice has been indispensable.

I would like to express my gratitude to my former and present lab mates, including Yajie Yan, Zhiyang Huang, Hang Dou, Dan Zeng, Huayi Zeng, Hang Yan, Chunyuan Li, Yiwen Ju, and Gustavo Gratacos Hernandez. Their mentorship, invaluable advice on research and career, and friendship have been instrumental in my growth as a researcher.

I would like to give a special mention to my friends, whose friendship has made my PhD journey truly enjoyable and left me with innumerable sweet memories. Jingyi Jiang has believed in my abilities and supported me during the publication of my first paper. Zhe Wang, Chenfeng Zhao, Shane Chu, Xiaoxiao Zhou, and Zehao Dong have been there for me through thick and thin, offering comfort and sharing my joy for achievements. Hansong Gao and Chunhui Yang have been my dear friends since I first came to US, and we have shared our ups and downs together. I am also grateful to Yu Sun and Zhihao Xia for valuable career advice and for being role models as a peer researchers.

Lastly, I am grateful to my family for instilling in me a love of learning, encouraging me to pursue my dreams, and providing unwavering support throughout my PhD journey. To all
who have supported me, I cannot thank you enough for your contributions to my success as a researcher. Thank you all for being part of this incredible journey.

Xingyi Du

Washington University in St. Louis
August 2023
Abstract of the Dissertation

Injective Mapping under Constraints

by

Xingyi Du

Doctor of Philosophy in Computer Science

Washington University in St. Louis, 2023

Professor Tao Ju, Chair

Injective mapping of meshes is a fundamental yet long-standing problem in computer graphics. The problem is particularly challenging in the presence of constraints, as an injective initial map is often not available. In this dissertation, we propose methods for computing injective maps that satisfy constraints without the need for an injective initial map. Our key contribution is a family of novel energies that are smoothly defined for arbitrary maps (injective or non-injective) and that promote injectivity when minimized. We first introduce the Total Lifted Content (TLC) energy for mapping 2D and 3D meshes into target domains with constrained boundaries. Next, we present the Smooth Excess Area (SEA) energy for computing globally injective maps of triangle meshes with general positional constraints. Both TLC and SEA have desirable theoretical properties for injectivity and are simple to optimize using standard gradient-based solvers. The two methods proved highly successful in practice when tested on large-scale benchmarks for the injective mapping problem. Finally, we develop variants of these energy forms to reduce isometric distortion, which is often desirable in graphics applications. The proposed isometric variants retain the desirable traits and achieve a similar level of success in recovering injectivity as the original energies, but with significantly lower isometric distortion.
Chapter 1

Introduction

1.1 Motivation

Functions between geometric domains are usually called mappings. Concretely, given a source and a target domain, a mapping assigns to each point in the source, a point in the target. The corresponding target point of the source point is also called its image point.

Computing a proper mapping lies at the center of many problems related to geometry. Examples include surface parameterization - mapping a surface to the 2D plane, surface registration - mapping a template to raw data, shape matching - inter-surface mappings, animation/morphing - deforming a shape using a mapping with low distortion, and physical simulation - modeling developable material or elasticity by mappings with desirable properties. More generally, mappings are useful for the following reasons:

- Mappings allow to transfer information between geometric domains. For example, in computer graphics, to enhance the visual appearance, detailed textures are stored in 2D arrays (images) and are transferred from the 2D plane to the surface using texture mapping [5, 50]. Similar applications include normal mapping, displacement mapping, and volumetric texture for volumetric rendering [56, 58].

- Mappings between shapes provide a way to quantify their similarity, which is crucial to shape understanding and analysis. For example, in biomedicine, many diseases can be diagnosed by comparing the shape of the organ with a normal organ, and their similarity can be quantified by the mapping between them [49].

- Mapping is a powerful way to represent shapes. In mathematics, an $n$-dimensional manifold is defined as a collection of mappings on $\mathbb{R}^n$. In geometric modeling, parametric surfaces can be viewed as mappings from the 2D plane to the 3D space. Recently,
representing 3D surfaces by mappings on 2D is a popular way to apply deep learning to 3D data [52, 66, 51].

- Many complicated tasks on the original domain can be simplified by mapping the domain to a simpler one. For example, complex geometric shapes, such as the human brain, can be more easily visualized and compared by mapping them to a canonical domain, such as a sphere [33, 27].

Various applications pose different requirements for mappings, e.g., texture mapping usually needs to preserve distance or angle, handle-driven deformation requires the mapping to satisfy user-given positional constraints. In almost all the applications, the mapping needs to be injective (also called one-to-one), which means it must map any two distinct points in the source to two different points in the target domain. An injective mapping guarantees the existence of an inverse mapping and ensures that the data transferred between domains is well preserved. Mathematically, there always exists a continuous one-to-one mapping (called homeomorphism) between two domains with the same topology. However, computationally finding such a mapping, possibly under extra user-given constraints, is still a challenging task. Therefore, developing practical algorithms to compute injective mappings is of vital importance to many applications in computer graphics, vision, and geometry processing.

1.2 Simplicial meshes and simplicial maps

In order to compute the mapping, geometric domains need to be represented in a way that is easy for computers to process. One of the most common representations of geometric shapes is simplicial mesh, namely, triangle mesh or tetrahedron mesh. Mathematically, a \( k \)-simplicial mesh is a pure simplicial \( k \)-complex, where \( k \) is 2 for triangle mesh or 3 for tetrahedron mesh. A simplicial mesh is represented by a pair \((V, T)\), where \( V \) is a list of points in Euclidean space, which we call vertices, and \( T \) is a list of \( k \)-simplices, where each simplex is an ordered list of \( k + 1 \) vertices in \( V \). We emphasize that the order of vertices in the simplex matters, as we will need it to define the orientation of a simplex later. Such simplices are sometimes called ordered simplices [40].

The natural class of mappings for simplicial meshes is obtained by asking the mapping to be locally affine within each simplex. This piecewise linear map is called a simplicial map.
For the simplicial map to be well-defined, every simplex of the source mesh must have a non-zero volume. From now on, we assume source meshes always satisfy this condition. Simplicial map is always continuous and it is completely determined by the location of the image points of vertices of the mesh. Hence, the degree of freedom of a simplicial map is $d*|V|$, where $|V|$ is the number of vertices of the mesh, and $d$ is the dimension of space where the target domain is embedded. Since affine mapping always maps a simplex to a simplex, the image of the source mesh is another simplicial mesh in the target domain, which we call the target mesh. As the simplicial map is completely determined by the vertices of the target mesh, computing a simplicial map is equivalent to computing a target mesh, i.e., finding the location of target vertices.

![Source Mesh](image1.png) ![Target Mesh](image2.png)

Figure 1.1: Left: simplicial map is affine within each simplex of the source mesh. Right: the source mesh is mapped to another simplicial mesh, called the target mesh.

In this work, we further focus on the case where $k$-simplicial meshes are mapped to Euclidean space $\mathbb{R}^k$. That is, the target domain is the 2D plane for a triangle mesh and the 3D space for a tetrahedron mesh. This setting is adopted in most applications, and is the foundation of more complex mapping problems, e.g., inter-surface mappings, as general mappings can usually be decomposed into the special mappings considered here.
1.3 Injectivity criteria

In our setting of simplicial maps, there is a set of simple rules to determine whether a map is injective, which we call the injectivity criteria. These criteria not only tell us when a map is injective, but also provide guidance for developing algorithms to find injective mappings. To introduce injectivity criteria, we begin with a concept that is central to injectivity of simplicial maps, that is, the orientation of a $k$-simplex in $\mathbb{R}^k$.

1.3.1 Orientation of a simplex

Consider a $k$-simplex with vertices $(v_1, v_2, \ldots, v_{k+1})$, where $v_i \in \mathbb{R}^k$. Thinking $v_i$s as column vectors, we have a $k \times k$ matrix $(v_2 - v_1, v_3 - v_1, \ldots, v_{k+1} - v_1)$, which we denote as $X$. The volume of the simplex is given by $|\det X| / k!$, where $\det$ stands for matrix determinant. We define $\det X / k!$ to be the signed volume of the simplex.

Suppose $s = (v_1, v_2, \ldots, v_{k+1})$ is a simplex in the source mesh, and $t = (p_1, p_2, \ldots, p_{k+1})$ is the corresponding target simplex under a simplicial map, where $v_i$ is mapped to $p_i$. We say the map preserves the orientation of the source simplex if the target simplex $t$ has positive signed volume. See figure 1.2 for an example in 2D. Accordingly, we say a target simplex is proper, degenerate or inverted if its signed volume is positive, zero or negative, respectively. Two target simplices are said to have the same orientation if they are both proper or both inverted.

It is well known that some non-orientable manifold cannot be embedded in 2D or 3D. The most famous examples are Möbius strip (no embedding in 2D) and Klein bottle (no embedding in 3D). Therefore, we only consider the case where the source mesh is an orientable manifold. Without loss of generality, we also assume that the order of vertices in source simplices are such that there exists an injective simplicial map that preserves the orientation, i.e., all target simplices under this map have positive signed volume.

Now, we are ready to state the first criterion for injectivity. Since an injective affine mapping never maps a region with non-zero area to a region with zero area, we obtain a necessary condition for a simplicial map to be injective: the target mesh must have no degenerate simplex if the simplicial map is injective. It is clear that this condition is not sufficient.
for injectivity. In the following, we will introduce other conditions to fully characterize injectivity. For the sake of simplicity, we restrict our discussion to triangle meshes. However, the injectivity criteria can be easily extended to tetrahedron meshes.

### 1.3.2 Local injectivity

A necessary condition for injectivity is local injectivity [62]. A map is locally injective at an interior point $p$ of the source mesh, if there is some local neighborhood of $p$ within which the map is injective. The map is called locally injective if it is locally injective at all interior points of the source mesh. As a weaker notion of injectivity, local injectivity are more tractable and are useful on its own in many applications [62].

To describe local injectivity, we need to introduce the concept of signed angles of a triangle. For a non-degenerate triangle in $\mathbb{R}^2$, we define its signed angles to have the magnitude of its (unsigned) angles and the sign of its signed area. The angle sum of an interior vertex $v$ in the target mesh is the sum of all signed angles at $v$ in its incident triangles. This sum is always a multiple of $2\pi$. We say the vertex $v$ is overwound if the absolute value of its angle sum is greater than $2\pi$. (e.g., the gray vertex in the bottom of figure 1.3 (c))
For triangle meshes, there are simple rules to check whether a simplicial map is local injective. These rules can be deduced by considering points at different locations of the source mesh:

- For an interior point of a source triangle, the map is locally injective if the corresponding target triangle is not degenerate.
- For a point in the interior of an interior edge of the source mesh, the map is locally injective if the two neighboring triangles of the edge are mapped to two target triangles with the same orientation.
- For an interior vertex, the map is locally injective if the absolute value of the angle sum of the target vertex is $2\pi$. Suppose we also know that all target triangles have the same orientation, this is equivalent to ask that there are no overwound vertices in the target mesh.

![Diagram](image)

Figure 1.3: A locally injective map must be free of three configurations in the target mesh: (a) degenerate triangle (b) inconsistently oriented triangles shared by an edge (c) overwound vertex (gray vertex bellow).

Although a locally injective map can have a fully inverted target mesh, such a map is usually not desirable in practice. Hence, we further ask the map to be free of inverted target triangles. In summary, we have the following characterization of local injectivity for simplicial maps on triangle meshes.

**Proposition 1** (Local Injectivity Criteria). *A simplicial map on a triangle mesh is locally injective if and only if the following conditions hold for the target mesh,*
1. The triangles are all proper.

2. There are no overwound vertices.

1.3.3 Global injectivity

To distinguish from local injectivity, we say a map is globally injective when the map is injective in the global sense, i.e., different source points are mapped to different target points. For simplicial maps, Lipman [46] showed that global injectivity amounts to asking that (1) the target triangles are either all proper or all inverted and (2) the boundary of the target mesh does not self-intersect.

Similar to locally injectivity, we further require a globally injective map to be free of inverted triangles. Now, we arrive at the following characterization of injective simplicial maps on triangle meshes.

Proposition 2 (Global Injectivity Criteria). A simplicial map on a triangle mesh is (globally) injective if and only if the following conditions hold for the target mesh,

1. The triangles are all proper.

2. The boundary of the target mesh does not self-intersect.

1.4 Problem statement

As we have seen, computing a simplicial map is equivalent to finding a target mesh, and the injective requirement amounts to asking the target mesh to satisfy a set of injectivity criteria.

To formally state the injective mapping problem, we further details the representation of a simplicial mesh. A $k$-simplicial mesh embedded in $d$-dimensional space is represented by a pair $(V, T)$, where $V = (v_1, v_2, ..., v_n) \in \mathbb{R}^{d \times n}$ represents $n$ vertices in $d$-dimensional Euclidean space, and $T = (t_1, t_2, ..., t_m) \in \{1, 2, ..., n\}^{(k+1) \times m}$ represents $m$ simplices, where each $k$-simplex is specified by an ordered list of $k + 1$ indices of vertices in $V$. The injective simplicial mapping problem can now be formally stated:
Definition 1 (Injective Simplicial Mapping Problem). Given an input source $k$-simplicial mesh $(V, T), V \in \mathbb{R}^{d \times n}$, find target vertices $U \in \mathbb{R}^{k \times n}$ such that the target mesh $(U, T)$ satisfies all the injectivity criteria.

This problem is challenging because the search space is huge and it varies with the input source mesh. For most real-world meshes, the search space $\mathbb{R}^{k \times n}$ can have thousands of dimensions. Moreover, the injectivity criteria become highly non-convex and non-linear constraints when expressed in target vertex coordinates. For example, requiring a triangle $(v_1, v_2, v_3)$ to be proper is asking $\det(v_2 - v_1, v_3 - v_1) > 0$, which is a non-convex quadratic constraint. Thus, asking all triangles to be proper introduces thousands of such constraints. Furthermore, the requirement of a non-intersecting boundary poses a more global constraint.

Many applications require the map to satisfy certain positional constraints. This means certain source vertices must map to specific points in the target domain. Positional constraints are commonly used in texture mapping to specify landmark correspondence [42], or to drive the deformation of shapes as handles [34]. This problem can be formally stated:

Definition 2 (Constrained Injective Simplicial Mapping Problem). Given an input source $k$-simplicial mesh $(V, T), V \in \mathbb{R}^{d \times n}$, an ordered list of constrained vertex indices $C$, and their corresponding target points $P \in \mathbb{R}^{k \times |C|}$, find target vertices $U \in \mathbb{R}^{k \times n}$ such that the target mesh $(U, T)$ satisfies all the injectivity criteria, and for each index $i \in \{1, 2, ..., |C|\}$, $U_{C_i} = P_i$.

The injective mapping problem is even harder with positional constraints. In general, even determining the existence of injective maps under such constraints is largely an open problem.

1.5 Related works

Injective mapping of a simplicial mesh is a classic and long-studied problem in geometry processing, see [18, 32] for surveys.
1.5.1 Tutte’s embedding

One of the earliest known algorithms for injective mappings is Tutte’s embedding [72], which up to this day, is one of the only methods that guarantee an injective map to a 2D domain. Besides the injective guarantee, Tutte’s embedding is popular because it is efficient to compute, which amounts to solving a sparse linear system. Although several works [17, 25, 3] extended it to other specific classes of mappings, its essential limitations remain: it can only map injectively to a prescribed convex boundary in 2D, without any interior constraints. Furthermore, its 3D extensions do not yield injective mappings, even in trivial cases.

1.5.2 Inversion-free and locally injective maps

As we have seen, an injective map requires the target mesh to have no inverted simplices. As a step towards injectivity, many recent methods focus on computing inversion-free maps that preserve simplex orientation.

**Inversion-free optimization** It is known that if we start from a (locally or globally) injective mapping and continuously deform the mapping without introducing any inversion, the mapping will remain (locally or globally) injective. Based on this knowledge, many methods require an injective initialization and then proceed to optimize distortion measures while preventing any inversion during optimization.

A few works addressed the problem of inversion-free mappings in 2D and 3D, via the use of “barrier”-type energies, in which the objective function includes terms that grow asymptotically as a simplex becomes degenerate. All of these methods require an injective initializer, as the barrier term is unlikely (and in most cases can’t) recover from non-injective states. Hence they initialize from the identity map or from Tutte’s embedding. Locally Injective Mappings [62] suggested incorporating a barrier term, using the log of the determinant. [48] followed a similar path by solving a sequence of convex programs. [68] tailored a line search optimization that uses a maximal step size that avoids inversions. Instead of using an auxiliary injectivity barrier, several methods directly optimize distortion metrics that explode near degeneracies. One of the first works to suggest such an energy was MIPS [31], which was used in inversion-free optimization in [21]. [59] use ARAP as a pseudo-majorizer for other
inversion-free energies. Other methods explored computational speedups of optimization of these energies via majorization [65], preconditioning [12], or modifying the quasi-Newton algorithm [78]. [47] compute a parameterization by constructing a sequence of bounded distortion maps, without setting any positional constraints. In addition to these, methods like [53], [41], [35], and [70] also adopt the injective initialization paradigm. Soft constraints with penalty energies can be added into these methods, although constraint-satisfaction is not guaranteed [35].

Inversion-free initialization An alternate range of methods enable initialization from inverted configurations and then optimize towards an inversion-free map – possibly satisfying additional positional constraints. Algorithmic strategies vary widely and include projection techniques [2, 38, 69], deconstructed domains [20], and nonconforming meshes [75]. Likewise, alternate approaches compute parameterization via a sequence of bounded distortion maps (without setting any positional constraints) [47], or adaptively partition vertices into blocks for more efficient optimization [54].

A different approach is taken in [74], which is based on the observation that the extremal quasiconformal maps (maps minimizing the maximal conformal distortion), in the continuous setting, are guaranteed to be locally injective and can conform to given positional constraints. However, the injectivity guarantees do not translate well to a discrete approximation, and inversion of triangles often happen at concave corners [75].

Another work [76] directly minimizes the total unsigned area (TUA). However, as we will detail in Section 2.3.1, TUA in fact optimizes over the closure of the space of injective maps, whose boundary contains degenerate triangles with zero area. Furthermore, TUA suffers from derivative discontinuities and vanishing gradients. Therefore, minimizing TUA often gets stuck in non-injective minima.

The method of [30] uses a subspace for locally injective harmonic maps and can adhere to periodic cone conditions, but cannot map injectively into arbitrary boundary constraints. Several methods consider enabling the boundary vertices’ images to slide across the target boundary while ensuring the resulting assignment still induces a bijective mapping into the target domain. [2] map tetrahedral meshes into polycubes while allowing each vertex to slide on its assigned flat face of the polycube. [3] extend Tutte and show that for some target convex polygons, vertices may be allowed to slide on the boundary of the target polygon.
Embeddings with sliding boundaries have been theoretically studied in [46], where local injectivity is shown to suffice for global injectivity even when the boundary is allowed to slide.

### 1.5.3 Globally injective maps

As mentioned earlier, in general, inversion-free on its own is only necessary but not sufficient to guarantee local injectivity, which in turn is a weaker requirement than global injectivity. An inversion-free map is only globally injective when the target mesh has a non-intersecting boundary [46].

To compute globally injective maps, barrier-based methods for 2D parameterization have been further extended [35, 70, 68] by augmenting distortion energies with additional terms that diverge when boundaries touch. Optimization of these energies then continuously preserves injectivity. These methods have been shown to be highly effective for 2D tasks, but they require an inversion- and overlap-free initial embedding (again e.g. Tutte) to start optimization with, analogous to their inversion-free counterparts. However, no such initializer is readily available with positional constraints. While it can sometimes be feasible to incrementally drag constrained points towards their targets, finding such a path can be NP-hard or may not exist at all. Alternately, physics-based, iterative collision-response methods [7, 29] have also been adapted for injective deformation tasks [60, 8, 28]. However, these methods generally do not resolve inversions, have no guarantees of convergence and, in practice, often cannot and do not resolve all overlaps [43].

In summary, when a traversed optimization path towards satisfying positional boundary conditions can not be continued due to nearly overlapping or nearly inverted states, all such methods for finding global injectivity will halt and so fail.

### 1.5.4 Remeshing methods

It is possible that an injective map satisfying the given positional constraints may not exist without altering the mesh structure. A different line of approach computes maps that cannot be defined solely in terms of the input mesh (i.e., the mapping changes the mesh...
1.6 Goals

Most existing methods formulate mapping problems as an optimization problem and use tailored objective functions (referred to as energies) to promote injectivity. However, current energies possess certain drawbacks that limit their application for computing injective mappings. Barrier-type energies, for example, increase to infinity when a simplex approaches degeneracy. Although effective in preserving injectivity, they require injective maps to initialize, which are generally not available for tetrahedron meshes or when positional constraints are present. Another group of energies are well-defined and bounded for all target meshes, and they are designed to eliminate inversion. However, being inversion-free is only a necessary condition and is not sufficient to guarantee local injectivity, let alone global injectivity. In addition to injectivity, map quality is another important consideration for graphics applications, such as texture mapping. It is usually desired that the map preserves the geometry (lengths, angles, areas) across the source and the target mesh. Several works have proposed improving map quality by optimizing energies that capture the distortion of the map. However, these methods often require an injective initial map to begin with.
In this dissertation, we adopt the optimization approach and strive to design suitable energies whose minimization can reliably produce injective maps with low distortion, while also satisfying user-given positional constraints.

Ideally, an energy $E$ for injectivity optimization should possess the following properties:

- $E$ is well-defined for any target mesh, allowing the optimization to start from any non-injective initial map and satisfy positional constraints from the beginning.

- $E$ captures the non-injectivity of maps. The most compelling evidence that an energy captures non-injectivity is the guarantee that its minima are injective. Alternatively, assessing the energy on numerous examples serves as a practical approach to determine whether the energy captures non-injectivity.

- $E$ captures map distortions, ensuring that the map exhibits low distortion when the energy is minimized. Similar to non-injectivity, this property can be verified either theoretically or empirically.

- Computationally, $E$ ought to be easy to optimize. This typically implies that the energy and its derivatives should be straightforward to evaluate, and that $E$ should exhibit sufficient smoothness.

- The number of parameters involved in $E$ should be minimal, and they should be relatively unaffected by specific inputs, thereby eliminating the need for users to adjust them for different use cases.

Designing an energy that possesses all desirable properties is a challenging task. Therefore, we break it down into three objectives. First, we design an energy for the special case where the boundary of the target mesh is fixed and intersection-free. Next, we generalize the energy to handle cases with positional constraints but not necessarily a fixed target boundary. Finally, we enhance the energy so that its minimization not only promotes injectivity but also reduces map distortion. The following provides more details on these three objectives.

**Goal 1 Injective mapping with a fixed boundary:** In this part, we assume the boundary of the target mesh is given as input and has no self-intersection. According to the injectivity criteria, in this specific scenario, global injectivity equates to requiring all target simplices to be proper. Our aim is to design a novel energy for this case and demonstrate its theoretical
benefits. Additionally, we intend to construct a benchmark dataset of mapping problems to evaluate our method and compare it with existing methods.

**Goal 2** *Injective mapping with positional constraints:* We generalize our energy to handle the more general case when the boundary of the target mesh is not fixed. In this situation, being inversion-free is insufficient to guarantee either local or global injectivity. We will leverage novel observations of global injectivity and design a new energy for this general case. The energy will provide us with a method to compute injective mappings under positional constraints. We aim to establish theoretical guarantees for our energy and assess its performance on a benchmark dataset.

**Goal 3** *Reducing map distortion:* Lastly, we want to investigate the capability of our methods to generate maps with low distortion. We aim to achieve low distortion without compromising the robustness of computing injective maps. We hope to demonstrate theoretical properties related to both injectivity and map distortion. Moreover, we plan to evaluate our method on large-scale benchmarks.

The remainder of the dissertation is organized as follows. Chapter 2 introduces our work on Goal 1. Chapter 3 focuses on our work on Goal 2. Chapter 4 presents our work on Goal 3. The final chapter summarizes our work and discusses potential future directions.
Chapter 2

Mapping into a Domain with Fixed Boundary

2.1 Introduction

Computing constrained mappings between domains is a fundamental task, performed across a wide range of geometric and physical applications ranging from parameterization and UV-mapping, to deformation modeling and the simulation of elastica. In all of these applications, it is in most cases critical to generate a one-to-one, injective mapping. This ensures that the inverse map exists and that the correspondence between domains is well-defined. Injectivity is critical for various applications, such as painting textures in UV space, co-analyzing shapes based on correspondences, obtaining good-looking deformations, and generating physically correct simulations of materials, to name just a few.

Most of the time, (local) injectivity is formulated computationally as preservation of all mesh simplices’ orientation, i.e., no triangle or tetrahedron is inverted. Unfortunately, the injectivity constraint is not only highly non-convex, but also an open set, making any optimization involving it non-trivial. As a result, many mapping and deformation algorithms focus on preserving triangle’s orientation while improving the map’s quality, i.e., by minimizing distortion measures [59, 78] that also act as a barrier that pushes them away from degenerating triangles on the closure of the locally-injective set. This in turn entails that they require a feasible embedding - one that is locally injective and satisfies all given constraints - as initialization to begin the minimization process. Indeed, if the input is non-injective, most distortion metrics that act as a barrier to prevent triangles from inverting also fight against un-inverting the initially-inverted triangles.
To find an un-inverted initializer for the above methods while exactly satisfying the given boundary constraints, one can opt to use one of two approaches: either (1) limit themselves to embeddings from the only known method with injectivity guarantee – Tutte’s embedding [72], whose guarantee is restricted to only 2D convex domains, or (2) use one of a number of recent methods [74, 2, 38, 20, 69] that have been developed to produce injective and low-distortion mappings for given constraints, \textit{without} requiring an injective initialization.

Unfortunately, the latter class of methods is in general not guaranteed to succeed in finding an injective mapping, and in practice they often fail on examples where injective mappings do exist (see Section 2.5). The main reason for that is that they focus on a much larger task, of computing \textit{low-distortion} maps, which entails they are not tailor-made for injectivity. Indeed, for many methods, the low-distortion paradigm is engrained in their approach. Counter-intuitively, attempting to require a \textit{less} strict distortion bound so as to optimize only injectivity often leads to deterioration of success rates in such methods instead of increasing them.

To address these issues, we focus solely on local injectivity, \textit{i.e.}, the correct orientation of elements, without considering their geometric distortion. We revisit the idea of computing an injective mapping via a variational principal of minimizing some energy. Instead of a barrier energy, we devise a new energy tailor-made for recovering injectivity from a given non-injective embedding while satisfying positional constraints. We refer to this injectivity energy as \textit{Total Lifted Content} (TLC). Intuitively, TLC measures the total content (2D area or 3D volume) of a mesh after \textit{lifiting} the simplicies of the mesh to a higher dimension. We
demonstrate two properties of our energy that shed light on our energy’s efficacy in enforcing injectivity:

1. TLC is a well-defined and smooth function over the entire embedding space, regardless of injectivity. This contrasts barrier energies (e.g., MIPS or Symmetric Dirichlet), which become undefined upon non-injectivity.

2. More importantly, the \textit{global minimum} of TLC is \textit{only} achieved by an injective embedding, if such an embedding exists.

We know of no existing energy in either 2D or 3D that possesses these properties. Additionally, we show the connection between minimizers of TLC and several known distortion-minimizing maps, including MIPS and harmonic maps. Unlike existing injectivity-recovery methods that rely on sophisticated and custom-made solvers to impose injectivity constraints, our energy can be easily minimized using standard solvers such as quasi-Newton and projected Newton.

To demonstrate the efficacy of our method, we introduce in Section 2.5 a benchmark set comprising of existing and many new challenging examples (including those in Figure 2.1) to extensively compare results of our method with state-of-the-art injective mapping methods. While our theoretical guarantee of injectivity only applies to global minima, it is not guaranteed we will achieve it in practice since our energy is \textit{not} convex. However, we show empirically that converging to a global minimum is unnecessary for achieving injectivity. In fact, simply terminating the optimization upon reaching an injective embedding obtains a success rate of 100\% on all examples in the benchmark, whereas existing methods only have varied success. Importantly, we demonstrate that the output of our method then can be used to bootstrap standard distortion minimization methods to improve the distortion of the injective mapping. This enables injective distortion minimization for challenging examples that were previously not possible due to the unavailability of a starting, constraint satisfying, injective map.
2.2 Problem statement and Overview

We address the problem of injectively mapping an input simplicial (triangles or tetrahedra) mesh into a fixed boundary. We start with a 2D or 3D rest mesh \( M \) and a target mapping boundary \( B \) that is in one-to-one correspondence with the boundary of \( M \). We then seek a (locally) injective embedding \( T \) of \( M \) into \( B \), such that each simplex of \( T \) is positively oriented. Note that an injective embedding may not exist for certain choices of \( M \) and \( B \). An example is when \( M \) has a single interior vertex and it is connected to all boundary vertices, and \( B \) is not a star-shape. As our goal is to find an injective mapping, we will assume that such a mapping exists for the given \( M \) and \( B \).

We address this embedding task variationally by solving an energy minimization over the space of all possible embeddings. This embedding space has dimension \( d \times n \) where \( d = 2, 3 \) is the dimension of the embedding and \( n \) is the number of interior vertices in the triangulation. The crux then is in forming an appropriate energy \( E : T \to \mathbb{R}_+ \) to minimize. A desirable energy should satisfy the following three criteria necessary for minimization to gain an injective embedding:

1. \( E \) is well-defined for all possible embedding \( T \);
2. \( E \) is at least \( C^2 \) over the embedding space; and
3. All global minima of \( E \) are injective embeddings.

Criteria (1) allows minimization to start from easily available, non-injective initial embeddings, e.g., as obtained by Tutte; (2) enables the effective application of efficient, gradient-descent based minimization; and (3) is necessary for minimizers of \( E \) to generate injective maps. Note that (3) alone may not be sufficient in practice; for example, the existence of non-injective, local minima can present significant challenges to gradient-based solvers. However, to our knowledge, no energy satisfying even these basic criteria has been previously proposed.

The rest of the chapter is organized as follows. In Section 2.3 we introduce our new energy (Total Lifted Content), and detail not only how it meets the criteria above but also its connection with existing distortion energies. We discuss the minimization of the energy in
Section 2.4, and present extensive experimental results on a new benchmark data set in Section 2.5. We conclude in Section 2.6 with discussions on venues of future research.

2.3 Energy

We turn to constructing an energy $E$ that satisfies all three criteria enumerated in Section 2.2. Satisfying these criteria is not trivial, and we are unaware of any existing energy that meets all criteria. The construction of $E$ stems from a central energy related to injectivity – the total unsigned area.

2.3.1 Total Unsigned Area

Xu et al. [76] observed that when $T$ is an injective triangular mesh, it minimizes the sum of the \textit{unsigned} triangle areas among all embeddings into the target domain. This is due to the total unsigned area (TUA) being an upper bound to the sum of \textit{signed} areas, which is constant for a fixed boundary $B$, and equal if all triangles have positive area.

However, TUA fails several criteria as mentioned above. First, it is not smooth, considered as a function of the embedding. In particular, TUA exhibits a $C^1$ discontinuity as a vertex moves across the supporting line of its opposite edge in a triangle. Second, while any injective embedding achieves the global minimum of TUA, the inverse is not true: a global minimum of TUA can also be achieved by a non-injective embedding where the only triangles that have non-positive areas are those having zero areas (\textit{i.e.}, degenerate triangles). We call such an embedding, which lies on the closure of the space of injective embeddings, a \textit{pseudo-injective} embedding.

In addition to these deficiencies, another problem of TUA is its vanishing gradient. Note that TUA has zero gradient with respect to any vertex surrounded by a ring of consistently oriented triangles. In practice, we have observed that minimizing TUA can easily get stuck in “plateaus”, or local minimizers of TUA with vanishing gradients, that are far from being injective (see Figure 2.5).
Figure 2.2: Lifting a triangle \( t \) with vertices \( \{x_i, y_i\} \) \((i = 1, 2, 3)\) to a triangle \( \hat{t} \) in 4D via the \( \sqrt{\alpha} \)-scaled auxiliary triangle \( \tilde{t} \) with vertices \( \{u_i, v_i\} \).

### 2.3.2 Total Lifted Content

To address the shortcomings of TUA, we propose to lift the triangles (or tetrahedra) to a higher-dimensional space, and consider their total area (or volume). The lifting is designed so that the total content (area or volume) of the lifted simplices is a smooth energy over the entire embedding space, and that every global minimum of the energy is achieved by an injective embedding. We call this energy Total Lifted Content (or TLC).

Specifically, to lift a \( d \)-dimensional \((d = 2, 3)\) simplex \( t \), we make use of another \( d \)-dimensional, non-degenerate auxiliary simplex \( \tilde{t} \) and a positive scalar \( \alpha \), both of which are fixed during embedding optimization. We construct a \( 2d \)-dimensional lifted simplex \( \hat{t} \) by concatenating the vertex coordinates of \( t \) with the corresponding coordinates of \( \tilde{t} \) scaled by \( \sqrt{\alpha} \). An illustration for lifting a 2-dimensional triangle is shown in Figure 2.2. To lift a \( d \)-dimensional simplicial mesh \( T \), we use a set of auxiliary simplices, one for each simplex of \( T \). TLC is defined as the sum of the area (or volume) of the lifted simplices.

Before diving into more details, we give some intuition as to how TLC avoids the drawbacks of TUA. Recall that TUA of a mesh \( T \) is smooth except when a simplex becomes degenerate (having zero content). Since the content of the lifted simplex has contributions from both the
simplex of \( T \) and the auxiliary simplex (scaled by \( \sqrt{\alpha} \), the lifted simplex is \textit{never degenerate} if the auxiliary simplex is chosen to be non-degenerate and \( \alpha > 0 \). Hence TLC remains smooth even when some simplices of \( T \) become degenerate. Also, when \( \alpha = 0 \), TLC reduces to TUA and shares the same set of global minima as TUA (including the pseudo-injective embeddings). However, as we will show, there is a range of sufficiently small (but positive) \( \alpha \) such that the global minimum of TLC remains a global minimum of TUA but has no degenerate simplices (\textit{i.e.}, it is injective).

In the following, we first derive an explicit formula of TLC in terms of the geometric quantities of the input mesh and the auxiliary simplices. We then prove the two key properties of TLC, namely smoothness and injectivity at global minimum. We conclude this section by shedding light on the impact of the auxiliary simplices on the \textit{shape} of the energy-minimizing embeddings.

**Formula**

We give a general formula for TLC in any dimension \( d \). Consider a simplex \( t \) with auxiliary simplex \( \tilde{t} \). Let \( X \) (respectively \( \tilde{X} \)) be a \( d \times d \) matrix whose column vectors are the edge vectors from one vertex of the simplex \( t \) (respectively \( \tilde{t} \)) to the other \( d \) vertices of the simplex. The \( 2d \times d \) matrix of the edge vectors of the lifted simplex \( \tilde{t} \), denoted by \( \tilde{X} \), is therefore defined as

\[
\tilde{X} = \begin{pmatrix} X \\
\sqrt{\alpha} \ast \tilde{X} \end{pmatrix}
\]  

(2.1)

Consider the \( d \)-dimensional subspace of the \( 2d \)-dimensional lifted space that contains \( \tilde{t} \), and pick any orthonormal basis of this subspace. We can express each (column) edge vector of \( \tilde{X} \) as a length-\( d \) vector in this basis, yielding another \( d \times d \) matrix \( Y \). Note that \( Y^T Y = \tilde{X}^T \tilde{X} = X^T X + \alpha \tilde{X}^T \tilde{X} \). Using the volume formula of a \( d \)-dimensional simplex, the content of \( \tilde{t} \) is \( \|\text{Det}(Y)\|/d! \), where \( \text{Det} \) is the matrix determinant. By the multiplicativity of determinants,

\[
E_{\tilde{t}, \alpha}(t) = \frac{1}{d!} \|\text{Det}(Y)\| = \frac{1}{d!} \sqrt{\text{Det}(Y^T Y)} = \frac{1}{d!} \sqrt{\text{Det}(X^T X + \alpha \tilde{X}^T \tilde{X})}
\]  

(2.2)
Note that the lifted content reduces to the unsigned area (or volume) of $t$, $\|\text{Det}(X)\|/d!$, when $\alpha = 0$.

The TLC of a $d$-dimensional simplicial mesh $T$, for a given (and fixed) set of auxiliary simplices $\tilde{T}$ and a scaling $\alpha$, is

$$E_{\tilde{T},\alpha}(T) = \sum_{t \in T} E_{\tilde{t},\alpha}(t)$$

(2.3)

We ask that each auxiliary simplex in $\tilde{T}$ has non-zero content, but they do not need to form a connected mesh. For example, $\tilde{T}$ can be made up of equilateral triangles or tetrahedra of the same size. Again, note that TLC becomes TUA when $\alpha = 0$.

![Diagram](image)

Figure 2.3: Given a triangle $t$ and an equilateral auxiliary triangle $\tilde{t}$ (top), (a,b) plot the lifted content of $t$, $E_{\tilde{t},\alpha}(t)$, at different $\alpha$ values as the red vertex in $t$ moves along the blue dotted line in (a) and the green dotted line in (b). (c) plots $E_{\tilde{t},\alpha}(t)$ as the red vertex moves over the plane.

**Smoothness**

We first consider the lifted content of one simplex $t$. As a concrete example, consider the triangle $t$ shown in the Figure 2.3 (top-left), and let $\tilde{t}$ be an equilateral triangle (top-right). Observe from the plots in (a,b) that, as the red vertex of $t$ moves along the vertical (blue) or horizontal (green) lines, the lifted content $E_{\tilde{t},\alpha}(t)$ of $t$ is smooth and greater than the
unsigned area of $T$ (i.e., $E_{t,0}(t)$) for all $\alpha > 0$. We will confirm these observations below, for any dimension $d \geq 2$.

**Proposition 3.** The following holds for any simplex $t$, given a non-degenerate auxiliary simplex $\tilde{t}$ and positive $\alpha$:

1. $E_{t,\alpha}(t) > E_{\tilde{t},0}(t) \geq 0$.
2. $E_{t,\alpha}(t)$ is differentiable to any order with respect to $t$’s vertices.

**Proof.** 1. Since $t$ is the orthogonal projection of the lifted simplex $\tilde{t}$ to the first $d$ dimensions, the content of $\tilde{t}$ (i.e., $E_{t,\alpha}(t)$) is no smaller than the unsigned content of $t$ (i.e., $E_{t,0}(t)$). The equality holds only when $\tilde{t}$ lies in a subspace parallel to the first $d$ dimensions. This means that the projection of $\tilde{t}$ in the remaining $d$ dimensions has no content, contradicting that $\tilde{t}$ is non-degenerate.

2. The derivative of $E_{t,\alpha}(t)$ with respect to vertices of $t$, to any order, is a summation of rational terms each with powers of $E_{t,\alpha}(t)$ on the denominator and polynomials in $t$’s vertices on the numerator (see Appendix A.3). Since $E_{t,\alpha}(t) > 0$, $E_{t,\alpha}(t)$ has finite derivatives.

The above properties easily carry over to a simplicial mesh:

**Corollary 4.** The Total Lifted Content, $E_{\tilde{T},\alpha}(T)$, is strictly positive, greater than the total unsigned areas or volumes of $T$, and differentiable to any order over the embedding space of $T$, given any set of non-degenerate auxiliary simplices $\tilde{T}$ and positive $\alpha$.

**Injectivity**

We show that $E_{\tilde{T},\alpha}$ has only injective global minima for sufficiently small values of $\alpha$. Recall that $E_{\tilde{T},0}$ reduces to the TUA energy, whose global minima are attained by injective and pseudo-injective embeddings. Note that an injective mapping lies in a region of the embedding space where TUA is constant, since any small perturbation to non-constrained vertices keeps the mapping injective and hence TUA is equal to the target domain’s area.
This means that the embedding lies in the flat plateau of TUA which is comprised of all injective embeddings, as well as some pseudo-injective embeddings that can be perturbed to become injective. On the other hand, a pseudo-injective embedding lies at a $C^1$ discontinuity of TUA (due to presence of degenerate triangles). However, for any $\alpha > 0$ $E_{\tilde{T},\alpha}$ is smooth everywhere and strictly above TUA (Corollary 4). To accomplish this transition from non-smoothness to smoothness as alpha increases, and for a small enough range of $\alpha$, a pseudo-injective embedding must see a greater rise in energy than any injective embedding in order to “round off” the sharp bottom of the valley (see the illustration in the inset), and hence it is no longer a global minimizer of $E_{\tilde{T},\alpha}$.

We make a precise statement in the following proposition, which states that any injective embedding would have lower energy than all non-injective embeddings for some range of $\alpha$, and the range depends solely on that injective embedding and the auxiliary simplices. The proof is given in Appendix A.2.

**Proposition 5.** Let $T_0$ be some $d$-dimensional ($d = 2, 3$) injective embedding into the target boundary and $\tilde{T}$ be a set of non-degenerate auxiliary simplices. Then there exists some $\beta > 0$ such that $E_{\tilde{T},\alpha}(T) > E_{T_0,\alpha}(T_0)$ for any non-injective embedding $T$ and $\alpha < \beta$.

**Shape control**

It is conceivable that there may exist other energies that also have the desired properties (being smooth and having only injective minima). A unique feature of TLC is that the choice of auxiliary simplices $\tilde{T}$ offers additional control over the *shape* of the energy-minimizing embedding. As we shall see, when $\alpha$ takes on very small or large values, TLC converges to one of a few well-known energies.

As shown above, the global minimum of $E_{\tilde{T},\alpha}$ as $\alpha$ approaches 0 are attained only by injective embeddings. Since all injective embeddings have the same energy when $\alpha = 0$ (*i.e.*, the content enclosed by the target boundary), the global minimum of TLC as $\alpha$ infinitesimally
Figure 2.4: Left: Rest mesh (Bunny) and MIPS mapping into a target boundary (outline of letter “A”). Middle: Embeddings minimizing uniform-TLC (top) and rest-TLC (bottom) energy at various values of \( \alpha \). Right: Tutte and harmonic mapping of the rest mesh into the same boundary. Inverted triangles are colored red. Note the similarity between the two embeddings highlighted in boxes with the same color.

increases from zero is the injective embedding that sees the least instantaneous rise in energy. This instantaneous rise for a given simplex \( t \), with auxiliary simplex \( \tilde{t} \), is just the derivative of \( E_{t,\alpha}(t) \) with respect to \( \alpha \) evaluated at \( \alpha = 0 \). This derivative has a surprisingly simple expression (see proof in Appendix A.1):

**Proposition 6.** Let \( t, \tilde{t} \) be two non-degenerate \( d \)-dimensional (\( d = 2, 3 \)) simplices, and \( D(t) \) the Dirichlet energy of linearly transforming \( t \) to \( \tilde{t} \). Then

\[
\frac{\partial E_{\tilde{t},\alpha}(t)}{\partial \alpha} |_{\alpha=0} = D(t)
\]  

(2.4)

In other words, the injective embedding \( T \) that minimizes TLC as \( \alpha \to 0 \) converges to minimize the Dirichlet energy from \( T \) to the auxiliary simplices \( \tilde{T} \). The latter energy, in two-dimensions, is in fact the MIPS energy [31] from \( \tilde{T} \) (the “surface triangles”) to \( T \) (the “parameter triangles”).

At the other end of the spectrum, as \( \alpha \) approaches \( \infty \), \( E_{t,\alpha}(t) \) converges to infinity as well for a given pair of \( t, \tilde{t} \). However, its gradient \( \nabla E_{t,\alpha}(t) \) with respect to the simplex \( t \) has a well-defined limit up to a constant multiplier (see proof in Appendix A.1):
Proposition 7. Let $t, \tilde{t}$ be two $d$-dimensional ($d = 2, 3$) simplices with $\tilde{t}$ non-degenerate, and let $\tilde{D}(t)$ be the Dirichlet energy of the linear map from $\tilde{t}$ to $t$. Then
\begin{equation}
\lim_{\alpha \to \infty} \alpha^{1 - \frac{d}{2}} \nabla E_{t, \alpha}(t) = \nabla \tilde{D}(t)
\end{equation}

As a result, the embedding $T$ that minimizes TLC as $\alpha \to \infty$ tends to minimize the Dirichlet energy from the auxiliary simplices $\tilde{T}$ to $T$. This minimizer is otherwise known as the Harmonic embedding from $\tilde{T}$ to $T$.

In sum, the embedding that minimizes TLC straddles between the minimizer of the Dirichlet energy (for large $\alpha$) and the MIPS energy (for small $\alpha$). We visually depict this relation in two-dimensions in Figure 2.4 for two choices of $\tilde{T}$, either equilateral triangles of uniform size (top row) or triangles in the rest mesh (bottom row). We call the resulting energy with these two choices uniform-TLC and rest-TLC. Observe that as $\alpha$ increases, the minimizer of uniform-TLC approaches Tutte’s embedding of the rest mesh into the target boundary (blue boxes in Figure 2.4), which minimizes the Dirichlet energy of the map from the set of equilateral triangles. On the other hand, the embedding minimizing rest-TLC approaches the MIPS mapping of the rest mesh for small $\alpha$ (magenta boxes), and the harmonic mapping of rest mesh for large $\alpha$ (red boxes).

2.4 Algorithm

The TLC energy’s simplicity and smoothness enable efficient closed-form computation of its gradient and Hessian (see Appendix A.3). We thus are able to explore minimization of TLC via a range of higher-order, nonlinear optimization methods. We begin by testing TLC optimization with both quasi-Newton and Newton-type strategies. For our quasi-Newton (QN) method we employed a standard, off-the-shelf limited-memory BFGS solver [55]. Alternately, to exploit second-order information via Newton-type minimization more care is needed as the TLC Hessian can be indefinite. Rather than applying global offsets [55] (as is standard in optimization packages) we follow recent developments in distortion optimization [71, 44] and ensure positive-definiteness of the global Hessian matrix by projecting per-simplex Hessians.
Motivated by this observation, we employ a two-stage solving method as follows (we use $N = 10000$):

to positive-definite prior to assembly. The resulting projected-Newton (PN) method employs standard back-tracking line search and a direct solver [11] for each linear system solution.

Our theoretical guarantee applies only at global minima, and at potentially impractically small $\alpha$. Given the nonlinearity and non-convexity of TLC, for a given choice of $\alpha$ optimizers may take a long time to reach convergence and could converge to non-injective local minima.

Since our goal is solely to find an injective configuration, we choose to stop optimization as soon as the optimization encounters an injective solution along its search path. In turn, differing optimization methods can take dramatically different search paths that vary across examples so that their ability to reach injectivity differs per example.

In this regard we empirically observe that QN and PN are complementary. Due to its efficiency QN is faster at finding injectivity than PN on most examples, while PN then easily resolves the few examples where QN struggles to reach injectivity - albeit with increased computational cost. Motivated by this observation, we employ a two-stage solving method as follows (we use $N = 10000$):
1. Run QN solver for at most $N$ iterations. If an injective mesh is found during the process, return this mesh as result. If the solver converges but the mesh is non-injective, report failure (i.e., a non-injective local minimum). If neither has happened after $N$ iterations, then:

2. Re-solve the starting problem by running PN solver for at most $N$ iterations. As above, either return the first encountered injective mesh or report failure if the solver converges to a non-injective mesh. If neither happens after $N$ iterations, report failure as well.

Initial configurations also have a significant impact on the search path taken by the solver, which again impact the ability of the optimization to reach injectivity. Ideally we would like to pick an initial configuration that is close to the global minimum when possible. A natural candidate is then Tutte embedding. While Tutte embedding cannot produce injective configurations for non-convex boundaries, we observe that the number of flipped elements are often small and concentrated near the boundary. In addition, as we discuss in the previous section, Tutte embedding is closely related to TLC when the auxiliary simplices are chosen to be uniformly sized regular elements. We evaluate the performance of our method under varying initializations in Section 2.5.5.

2.5 Results

In this section, we evaluate our method on an extensive set of embedding problems for both triangular and tetrahedral meshes. We plan to distribute both our code and the data sets on the authors’ websites.

2.5.1 Parameter choices

Our energy is controlled by two sets of parameters, the auxiliary simplices $\widetilde{T}$ and the scaling $\alpha$. While the theoretical properties of our energy, smoothness and injective global minima, hold regardless of the choice of $\widetilde{T}$, different $\widetilde{T}$ result in not only different energy-minimizing embeddings (as studied in Section 2.3.2) but also different energy landscapes, and hence
varied success rate for specific solvers. We have found that the QN solver in NLopt is generally more successful in reaching injectivity when \( \bar{T} \) consists of uniformly sized equilateral triangles or tetrahedra (\textit{i.e., uniform-TLC}) than when \( \bar{T} \) is set to be the rest mesh \( M \) (\textit{i.e., rest-TLC}), for the same \( \alpha \) values. In this work, we report the performance of uniform-TLC in both 2D and 3D, and leave a thorough investigation into the effect of \( \bar{T} \) on the performance of different solvers for future work.

We observed that smaller \( \alpha \) generally lead to higher chance of reaching injectivity by energy minimization (\textit{e.g.}, see Figure 2.4). This is in line with our theoretical guarantee of injectivity, which applies to only sufficiently small \( \alpha \). However, a too-small \( \alpha \) results in a TLC energy that is very close to TUA and thus having close-to-vanishing gradients, which can significantly slow down the progress of gradient-based solvers. In our experiments, we found that choosing \( \alpha \) such that the total content of \( \bar{T} \) is \( 10^{-6} \) times of the content of the target domain strikes a good balance between success rate and efficiency, in both 2D and 3D.

### 2.5.2 A benchmark problem set

While existing injective embedding methods have all been tested on non-trivial examples, these examples are often different, making it challenging to perform direct comparisons between different methods. One of this work’s contribution is developing a extensive problem set for fixed-boundary embedding in both 2D and 3D. The set includes as many examples as possible from existing works that we have access to, plus hundreds of new examples that we created. We hope that our problem set offers a benchmark for future research in this area.

To provide fair and practically relevant evaluations, the data set is built with two criteria in mind. First, the problem should be feasible, meaning that an injective mapping satisfying the boundary constraints exists. If such mapping does not come with a problem, we ran all available methods (ours included) and include the problem in our data set if \textit{any} method succeeded (that is, we did not exclude any example that our method failed while some other methods succeeded). Second, the problem should not be trivial. In particular, we discard all examples where Tutte’s embedding is already injective. Our data set consists of three broad categories that correspond to different use scenario of injective embedding:
Figure 2.6: Top: rest meshes and letter-like boundaries used to create the embedding problems in the 2D parameterization category. Bottom: results of various methods on embedding Lucy into letter “G”, including our QN solver at different iterations. The zoom-in takes a close look at the inverted triangles in the result of FF (all close to being collinear).

- 2D parameterization: This category contains a few hand-made toy examples (Figure 2.5 far left), including two (in the middle) from previous works [75], and 30 examples that embed five surfaces (between 20K-50K vertices each) into the outline of six letters S,G,R,A,P,H (Figure 2.6 top; also in teaser). The bulk of this category comes from the impressive dataset put together by [47], which includes more than twenty thousand open surface meshes. For each mesh, we created an intersection-free target boundary using the method of [35], and included it in our data set if it is not solved by Tutte embedding (Figure 2.7).
• 3D parameterization: This category includes 116 polycube embedding problems from
[2, 19], 20 spherical mapping and 40 free-surface mapping problems from [69] (Figure 2.8).

• 3D deformation: Simulating physical deformation often yields severely distorted meshes
for which it is challenging to avoid inversion. Hence they provide an excellent test set
for a method’s ability to recover from inverted elements. We capture frames from
non-inverting simulations of twisting, generated by [43], starting from a rest shape and
then twisting it to introduce increasing levels of distortion. For each simulation we
embed the starting mesh to the boundary surface of each successive frame, obtaining
examples with generally increasing challenge. This category includes 728 examples
from deforming a rod (Figure 2.9), a cube (Figure 2.10), and the armadillo (Figure
2.11; also in teaser).

2.5.3 Benchmark comparisons

We compare our method (TLC) to the three most, to our knowledge, competitive methods
for injective embedding into a target domain without the need of an injective initialization:
Large-scale Bounded Distortion Mappings (LBD) [38], Simplex Assembly (SA) [20] and
Foldover-Free Volumetric Mapping (FF) [69]. LBD needs to set the upper bound \( K \)
on the distortion. While in theory an extremely large \( K \) is more permissive from a small
one, in practice large values prevent convergence of the method [38]. To choose \( K \), we
compute the maximal distortion between the rest mesh and the known injective solution
(either coming with the example or produced by one of other methods), and set \( K \) to be
double the maximal distortion. This guarantees the existence of an embedding with lower
distortion than the upper bound. For SA and FF, we use the default parameters suggested
by the authors. SA is not tested on examples that map a surface mesh to a planar boundary,
since the corresponding code is not available (after confirmation with the authors of SA).
For consistency, we feed all methods with the (non-injective) Tutte embedding as the initial
map.

We are primarily interested in how often a method succeeds. Here, success is defined as
producing an embedding without any inverted elements. As shown in Table 2.1, while every
other method exhibits varied success across different categories, our method consistently
solved all examples in each category. For our method, we also report (in parenthesis) the number of examples (if there are any) that require the second stage of the algorithm, which uses the PN solver. In the following, we present and discuss specific instances of each category of the benchmark.

**2D parameterization** The hand-made examples shown in Figure 2.5 are designed to test an embedding algorithm’s ability to deal with large boundary deformations (e.g., convex-to-concave deformations, as in top and middle) and transformation of inner boundaries (e.g., the bottom example, where the inner square is rotated by 180 degrees). We found that both SA and (particularly) LBD tend to fail in these scenarios. Note that since our method
Figure 2.8: Three examples from the 3D parameterization category, each mapping a rest tetrahedral mesh into a sphere (top), smooth surface (middle), and a polycube (bottom). Each example is a failure case for one of the three methods, FF, LBD and SA. Inverted tetrahedra are colored in red, and the numbers of inversion are marked in red.

stops at soon as (local) injectivity is obtained, the resulting embedding may contain many triangles with small angles. While reducing distortion is not our objective, for illustrative purposes, we show that the shape of such triangles significantly improves if optimization continues (see “TLC (converged)” column). This echoes our observations in Section 2.3.2 about the connection between the global minima of TLC and distortion-minimization maps. For these examples, we also show the results of minimizing the total unsigned area (TUA) energy, by setting $\alpha = 0$ in our method. Observe that the optimization easily gets stuck in local minima, where the gradient of the TUA vanishes.

The second group of examples, which map a surface mesh to a letter outline (Figure 2.6 top), prove challenging for all methods. Due to the highly detailed surfaces and the non-convex boundaries, achieving injectivity in these examples necessarily comes at the cost of
Figure 2.9: One example in the 3D deformation category (a twisting rod). All methods succeeded on this sequence.

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>TLC</th>
<th>FF</th>
<th>SA</th>
<th>LBD</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D (Param.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simple</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>Letters</td>
<td>30</td>
<td>0 (17)</td>
<td>7</td>
<td>N/A</td>
<td>26</td>
</tr>
<tr>
<td>[47]</td>
<td>10706</td>
<td>0 (19)</td>
<td>1292</td>
<td>N/A</td>
<td>267</td>
</tr>
<tr>
<td>3D (Param.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Polycube</td>
<td>116</td>
<td>0</td>
<td>3</td>
<td>29</td>
<td>2</td>
</tr>
<tr>
<td>Sphere</td>
<td>20</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>13</td>
</tr>
<tr>
<td>Surface</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>3D (Deform)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rod</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Cube</td>
<td>119</td>
<td>0</td>
<td>0</td>
<td>54</td>
<td>32</td>
</tr>
<tr>
<td>Armadillo</td>
<td>601</td>
<td>0</td>
<td>75</td>
<td>177</td>
<td>378</td>
</tr>
</tbody>
</table>

Table 2.1: Number of failed examples for each compared method in each category of the benchmark problem set.

significant triangle distortions. As shown in Figure 2.6 (bottom) for the example of mapping Lucy into the letter “G” (same as in Figure 2.1), Tutte embedding contains a large region of inverted triangles. Both LBD and FF failed, and the QN solver in the first stage of our algorithm failed to achieve injectivity within the maximum number of iterations. While it appears that QN might eventually reach injectivity if given more iterations to run, the PN solver in the second stage of our algorithm successfully finds an injective mesh with only 66 iterations (see result in Figure 2.1).

In contrast to the surface-to-letter problems, the examples from [47] map a complex surface to a domain that is already adapted to the shape of the surface (created by a distortion-minimizing parameterization algorithm [35]). As a result, Tutte embedding tends to create only small inverted triangles close to the target boundary (Figure 2.7). However, we observed that both FF and LBD often make the initial embedding worse by creating many more inverted triangles in the interior of the domain. In contrast, our method removes inversions without significantly impacting the interior triangulation.
Figure 2.10: Two example in the 3D deformation category (a twisting cube) where both SA and LBD failed to reach injectivity. The graph in the top-right shows the number of inverted tetrahedra for each of the 118 frames of the deformation sequence (ellipses indicate the frames from which the two examples were taken).

3D parameterization The embedding problem becomes significantly harder in 3D due to the added dimension. Tutte embedding, which is no longer guaranteed to be injective for convex 3D domains, generates inverted elements much more often than in 2D and in greater quantity. We found that existing methods are fairly effective in removing the majority of the inverted tetrahedra in this category (see Table 2.1). However, no method was able to resolve all inversions in all examples, except for ours. A few examples that compare our method with LBD, FF or SA in their failure cases are shown in Figure 2.8.

3D deformation For examples arising from deforming sequences, existing methods struggle with many frames, particularly on more complex meshes and towards the end of the deforming sequence where there is a large amount of twisting (see Figures 2.10 and 2.11).
Figure 2.11: Three example in the 3D deformation category (a twisting armadillo) where FF, SA and LBD all failed to reach injectivity. The graph in the top-right shows the number of inverted tetrahedra for each of the 600+ frames of the deformation sequence (ellipses indicate the frames from which the three examples were taken).

On the other hand, our method successfully found injective mappings for all frames in the three deformation sequences.

**Performance** We visualize the number of iterations taken by our QN solver or PN solver (for examples that QN reaches $N = 10000$ iterations) to converge for every example in our benchmark in Figure 2.12 (a,b). Observe that our method generally requires more iterations when the initial (Tutte) embedding contains more inverted elements. On the other hand, examples where at least one other method (FF, SA, or LBD) fails generally do not pose particular challenges for our method, although they tend to require more iterations in 3D. We also show in Figure 2.12 (c,d) the wall-clock running time of all methods being compared (TLC, FF, SA, LBD) on the subset of the benchmark where all methods succeed. Observe
Figure 2.12: Top: number of iterations taken by our PN or QN solvers over all benchmark examples in 2D (a) and 3D (b), where the horizontal axis is the ratio of the number of inverted simplices in the initial (Tutte) embedding over the total number of simplices (both axes are in log-scale). In the legend, “all pass” means that FF, LBD, and SA (3D) all succeed on that example; “some pass” means that at least one of these methods fails (our method succeeds on all examples). Bottom: running time (in seconds) of all methods in 2D (c) and 3D (d), where the horizontal axis is the number of vertices in the rest mesh (running time is in log scale).

that our method is comparable with FF but more expensive than LBD in 2D, while on-par with both FF and LBD but faster than SA in 3D. All experiments are performed on a Windows PC with Intel core-i7-4770 CPU at 3.40 GHz and 32GB of memory.

2.5.4 Comparison with other methods

We further compare our method with a few other injective mapping methods in 2D. Unlike methods compared above (FF, SA, LBD, and ours), these methods either may modify the mesh structure or are limited to convex target domains. We first compare with the method of [75], which uses an intermediate domain for co-parameterization and occasionally requires
refinement of the input mesh. Our method (with QN) successfully found injective maps for all examples in that paper (provided by the authors), except those that do not admit feasible solutions (e.g., Figures 1 and 9 in their paper). Two examples were included in the “2D parameterization” category of our benchmark (middle row of Figure 2.5). For these two examples, [75] needs to add 2 and 9 new vertices to achieve injectivity. We next consider the recent method of [63], which restores injectivity to Tutte embeddings without mesh refinement but only for convex target domains. Our method can serve the same purpose but is not limited to convex domains. Figure 2.13 shows our result on a complex example from [63] (mapping a Hele-Shaw polygon to a square).

Figure 2.13: Mapping a triangulated Hele-Shaw polygon (from [63]) to a square. Tutte embedding contains 46 inverted or co-linear triangles due to numerical errors (they are all along the boundary). Our method (QN solver) successfully restores the injectivity in 65 iterations.

2.5.5 Impact of initial embedding

Since our energy is non-convex, the choice of the initial mesh could significantly affect the solver’s behavior. The effect can manifest in two ways, (1) time to reach injectivity, and (2) the shape of the embedding when injectivity is reached. We demonstrate these effects in Figures 2.14 and 2.15. In Figure 2.14, we ran our method on the same set of input but starting from three different types of initial embeddings, which result in rather different injective maps. Interestingly, the solver produces very similar embeddings if we let it run until convergence of the energy (instead of terminating at injectivity). Figure 2.15 shows another example with a multi-connected target domain and a pathological initial embedding (note that the Tutte embedding is already injective). The QN solver failed to find an injective
Figure 2.14: Mapping a Bunny shape (same as Figure 2.4) into a cross with three different initial embeddings, Tutte embedding (top), harmonic map (middle), and one in which all interior vertices are collapsed onto a single point (bottom). Showing results of our method after terminating upon injectivity (middle) and after the energy has converged (right).

embedding within the maximum number of iterations, and while PN solver succeeded in reaching injectivity, it took nearly 5000 iterations.

2.5.6 Initializing distortion minimization

Many distortion-minimization algorithms can preserve injectivity if given a feasible injective initial map. Combining such an algorithm with our method allows to find injective and distortion-minimizing embeddings for challenging inputs that are challenging for existing approaches. We demonstrate such an example in Figure 2.16, where we compute an embedding of the Lucy model into the outline of letter “P” by first finding an injective embedding using our method (left), followed by minimizing an isometric energy (Symmetric Dirichlet) using a standard method [78] (right). Observe from the zoom-ins that the isometric distortion is significantly reduced from the initial mapping produced by our method.
2.6 Discussion

There are a number of promising avenues of future research. First, while we demonstrated success using only standard off-the-shelf solvers (quasi-Newton and projected Newton) for energy minimization, developing customized solvers has the potential to significantly improve convergence rate of the energy and hence reducing the time needed to reach injectivity. A second and related direction is exploring different types of auxiliary simplices, which can have a significant impact on the energy landscape and hence the search paths of the solver.

Last but not least, so far our theoretical analysis has only considered the injectivity of global minima of TLC. An even more interesting and practically relevant subject is characterizing the local minima and their injectivity. Interestingly, we did not observe any non-injective
local minima in the benchmark data set (since our solver never converged to a non-injective embedding). Further theoretical investigation might lead to a better understanding of the energy landscape and in turn more effective solving techniques.
Chapter 3

Mapping with Positional Constraints

3.1 Introduction

Mapping triangulated meshes onto the plane is a fundamental task in computer graphics, geometry processing and physical modeling. It is essential for many applications, such as UV parameterization, 2D deformation, simulation and inter-surface mapping, to name just a few. A key property often required of these maps is to be globally injective – meaning the map is one-to-one. Without global injectivity, maps are often unusable in many of the aforementioned applications (consider for example texture-mapping or packing). In many cases it is also required that injectivity is upheld along with user-given Dirichlet boundary conditions, that is, positional constraints that designate 2D target positions for a subset of the mesh vertices.

Many works focus on the weaker requirement of local injectivity, which enforces injectivity only within the local neighborhood of an interior mesh vertex or edge (see Section 1.3.2). Towards this goal, a range of recent methods [14, 74, 2, 38, 20, 69] focus on computing maps that are free of inverted triangles (and hence are locally injective around each edge). These methods have been successfully applied to fixed-boundary mapping problems, with possibly additional constraints, since an inversion-free map is both locally and globally injective if the boundary does not intersect itself [46]. However, having no inverted triangle is far from being sufficient to ensure an injective map, locally or globally, when the boundary self-intersects. An intersecting boundary may give rise to overlap between triangles sharing a common vertex (making the vertex overwound) and/or between non-adjacent triangles. Predetermining a non-intersecting boundary that admits a constraint-satisfying, injective map is often a computational crux in the first place.
Figure 3.1: Given a non-injective initial parameterization of a surface mesh (left) with inverted triangles (red), boundary intersections (orange dots), and overwound vertices (magenta dots), our method recovers a globally injective map (right) while keeping the constraints (blue points) in place. The inserts zoom in on one region in the initial map with many boundary intersections and inverted triangles (green box) and another region with an overwound vertex (cyan box).

For free boundary problems, computing a globally injective map remains a significant challenge. Harmonic maps into convex domains (e.g. via Tutte’s embedding [72]) are the only tractable class of maps known to be globally injective. For results beyond Tutte, recent methods employ barrier-type strategies that evolve an embedding from a given, globally injective starting configuration (via Tutte or otherwise given) while preserving injectivity and minimizing distortion [35, 68, 70]. However, for arbitrary positional constraints, no such starting configuration is generally available. An alternate strategy for these methods then is to either add soft constraints (penalty energies) [35], or else to programatically drag constrained points from an initial, feasible embedding towards their final target positions while continuously preserving injectivity [62]. The former strategy has no guarantee of constraint satisfaction, while the latter is often locked by choice of the path (and is more generally an instance of the path planning problem).

Hence, existing algorithms for computing injective maps either require initialization from an injective starting state, which is currently only possible without positional constraints, or
else can only prevent triangle inversion, which is insufficient to ensure injectivity. Finding globally injective maps that satisfy positional constraints thus requires a method to be able to recover from, or even pass through, non-injective configurations on its way towards a constraint satisfying, injective solution.

We present, to our knowledge, the first method for recovering globally injective parameterizations from arbitrary non-injective initial meshes while meeting positional constraints. These initial meshes can be inverted, wound about interior vertices, and/or overlapping (e.g., Figure 3.1 left). While our focus is on injectivity, the resulting maps can then be used to bootstrap distortion minimizing methods that require a feasible initialization.

Our core contribution is a new energy that reliably measures the degree of non-injectivity in a mapping. Our energy builds on the total lifted content (TLC) [14] energy proposed in 2, which is a smooth variant of unsigned triangle areas. While TLC addresses local injectivity within a fixed boundary, in order to tackle global injectivity with arbitrary positional constraints, in this work we introduce a new term whose subtraction from TLC captures the areas of both overlapping and inverted triangles in a given map. This energy, which we call the smooth excess area (SEA), is continuously defined for all maps - injective and non-injective alike - and is smooth almost everywhere. This allows SEA to be trivially minimized by off-the-shelf gradient-based optimization tools, initializing from any given non-injective state while maintaining “hard” positional constraints. As a key theoretical result that supports the use of SEA in promoting injectivity, we show that maps that minimize SEA are guaranteed to be locally injective and almost globally injective, in the sense that the area of overlap between non-adjacent triangles can be made arbitrarily small.

Since SEA is non-linear and non-convex, reaching a global minimum cannot be guaranteed in practice. Also, our theoretical guarantee does not cover global injectivity. Nonetheless, we show that our algorithm significantly outperforms state-of-the-art methods: it achieves a 85% success rate in recovering globally injective maps across a large benchmark set (one example is shown in Figure 3.1), and a 90% success rate for local injectivity. In contrast, existing methods for removing inverted triangles, which are not expected to achieve either local or global injectivity, largely fail on this data set, obtaining globally and locally injective maps on no more than 4% and 6% of the examples.
3.2 Problem statement and Overview

We consider a triangular mesh $M$ as a 2-dimensional simplicial complex whose faces are open balls in $d$-dimensions and are called vertices ($d = 0$), edges ($d = 1$) and triangles ($d = 2$). We denote the set of faces on the boundary of $M$ as $\partial M$. Two triangles are said to be edge-connected if they share a common edge, and an edge-connected component is a maximal group of triangles where any pair of triangles in the group are connected by a path of edge-connected triangles. A mesh may consist of one or more edge-connected components of triangles, and it may have one or more boundaries.

A simplicial map, $\Phi : M \to \mathbb{R}^2$, from $M$ to the plane is a continuous map that is affine when restricted to each face of $M$. This map is completely determined by the image of each vertex of $M$. For convenience of discussion, and since $M$ is typically given as input, we will use $\Phi$ to denote both the map and the mapped image $\Phi(M)$. Similarly, $\partial \Phi$ denotes both the map $\Phi$ restricted to the boundary $\partial M$ and the image of this map $\Phi(\partial M)$.

Our goal is to parameterize a triangular mesh to the plane in a globally injective manner while adhering to user-prescribed positional constraints. In addition, we seek parameterizations where all triangles are not inverted. Specifically, our input consists of a triangular mesh $M$, a subset $S$ of the vertices of $M$ that are labelled as constrained, and a set of target locations $T$ in the plane, one for each constrained vertex. We wish to find a globally injective, inversion-free simplicial map $\Phi : M \to \mathbb{R}^2$ such that $\Phi(S) = T$.

A feasible solution for a given input $\{M, S, T\}$ may not exist. For example, the constraint set $S$ may require that three vertices of a triangle are mapped so that the triangle inverts its orientation. An ideal parameterization algorithm would produce a feasible map whenever such map exists and report failure otherwise.

We start with initial maps that only satisfy positional constraints. These maps may be given by application (e.g., deformed) or can be directly computed via one of the numerous constraint-based parameterization methods. Our method optimizes our tailored energy, $E(\Phi)$, starting from a non-injective initial map, to seek an injective and inversion-free map while keeping the constraints unchanged. Next we introduce the energy $E$ and discuss its properties in Section 3.3, and then we present our optimization algorithm in Section 3.4.
3.3 Energy

We introduce an energy that measures the extent to which a simplicial map is not globally injective. In contrast to prior measures, our energy accounts for global injectivity (and not only inverted triangles) and is well-behaved (i.e., continuous and almost everywhere smooth) across all injective and non-injective maps.

The key idea behind our energy is to minimize the area of overlap between triangles, in addition to the area of inverted triangles. We first introduce the excess area measure, which captures both the overlapping and inverted triangle areas in a computationally efficient manner (Section 3.3.1). This measure is not smooth, and we show there exist certain degenerate configurations from which gradient-based optimization methods cannot recover. We analyze these conditions and introduce a smoother variant as our energy, called smooth excess area (SEA), which bounds the excess area from above, hence upholding smoothness properties as well as injectivity guarantees (Section 3.3.2).

3.3.1 Excess Area

A necessary condition of global injectivity is that the triangles do not overlap in their interior. As a result, the overlapping area between triangles is a proxy of the degree of non-injectivity. While we can explicitly compute this overlapping area by considering all pairs of triangles, the computational cost could be prohibitive for larger meshes. Furthermore, the pair-wise overlap does not capture a complete inversion of the mesh. We instead build an efficient-to-compute measure that simultaneously captures both overlap and inversion.

Consider a closed, oriented, and possibly self-intersecting curve $C$ in the plane. We define the occupancy of the curve, denoted by $O(C)$, as the total area of the plane where the winding number is positive. The winding number of $C$ around a point is the integer number of times that $C$ travels around that point. If $C$ is not self-intersecting and has a counter-clockwise orientation, the winding number is 0 everywhere outside the curve and 1 inside the curve. Otherwise, the winding number can assume other integer values, and each region of the plane partitioned by the curve has the same winding number. Figure 3.2 shows examples of curves, winding numbers, and regions contributing to occupancy (colored gray).
We define the *excess area* of a simplicial map $\Phi$ as the difference between the total unsigned area of all triangles of $\Phi$, denoted by $A(\Phi)$, and the occupancy of the boundary curve $\partial \Phi$:

$$A_{\text{excess}}(\Phi) = A(\Phi) - O(\partial \Phi) \quad (3.1)$$

As the next proposition shows, the excess area $A_{\text{excess}}(\Phi)$ is closely related to both the overlapping area among triangles and the area of the inverted triangles. We denote by $A_{\text{overlap}}(\Phi)$ the total unsigned area $A(\Phi)$ minus the area of the plane covered by $\Phi$, and $A_{\text{invert}}(\Phi)$ the total unsigned area of inverted triangles in $\Phi$. We prove in Appendix B.1 that:

**Proposition 8.** *For any simplicial map $\Phi$ of a triangular mesh,*

1. $A_{\text{excess}}(\Phi) \geq A_{\text{overlap}}(\Phi)$.
2. $A_{\text{excess}}(\Phi) \geq A_{\text{invert}}(\Phi)$.
3. $A_{\text{excess}}(\Phi) \leq A_{\text{overlap}}(\Phi) + A_{\text{invert}}(\Phi)$.
An immediate corollary of these inequalities is that the excess area is zero if and only if the mesh has neither overlapping nor inverted triangles. This property, together with the inequalities, makes the excess area a promising energy for promoting injectivity.

As to the computational cost, the most expensive part of computing the occupancy is computing the partitioned regions (known as the arrangement) of a polygon. For a polygon with \( n \) edges, the structure of the arrangement has complexity \( O(n^2) \) in the worst case, and so is the computational time. With a more sophisticated algorithm [10], this complexity can be further reduced to \( O(n \log n + k) \), where \( k \) is the number of edge intersections. Since the number of boundary edges of a mesh is typically much smaller than the number of triangles, the excess area can be computed more efficiently than computing all pairwise intersections between triangles.

### 3.3.2 Smooth Excess Area

The main drawback of the excess area as an energy to optimize is its lack of smoothness. First, as covered by Du et al. [14], the unsigned area \( A(\Phi) \) is only \( C^0 \) when \( \Phi \) contains a degenerate triangle. Second, as we show in Appendix B.2, while the boundary occupancy \( O(\partial \Phi) \) is continuous for all simplicial maps, it is only \( C^0 \) when the boundary map \( \partial \Phi \) is singular; that is, when two segments of \( \partial \Phi \) with non-zero length completely overlap. Although these non-smooth configurations are geometrically degenerate (e.g., three or more points are collinear), during optimization they generally cannot be avoided when maps transition from a non-injective state to an injective state. For example, flipping the orientation of an inverted triangle requires passing through a degenerate state of that triangle. Likewise, resolving overlapping triangles at a boundary vertex necessarily overlaps its two incident boundary edges (and so makes the map singular) during the process (Figure 3.3 top). In fact, our preliminary attempts at directly optimizing the excess area lead to frequent locking at precisely such degenerate, unavoidable configurations (see Figure 3.10).

To address these limitations, we construct our energy as a smoother variant that provides an upper bound to the excess area. While our energy is still \( C^0 \) in certain degeneracies, these configurations are even more degenerate than the \( C^0 \) configurations of excess area and easier to avoid during transitions from non-injectivity to injectivity. In our experiments, we found that optimization of our energy seldom gets stuck due to its non-smoothness.
Figure 3.3: Top: a non-injective map with overlapping area (yellow) around a boundary vertex (green) transitions to an injective map, going through a configuration where two incident boundary edges overlap (middle). Bottom: the same transition but the straight edges are replaced with their arc-edges; note that the arcs do not overlap during the process.

Our energy replaces each term of excess area, $A(\Phi)$ and $O(\partial \Phi)$, by a smooth(er) variant. First, the unsigned area $A(\Phi)$ is replaced with the total lifted content (TLC), introduced for these purposes by Du et al. [14]. TLC computes the sum of the unsigned area of all triangles after lifting each triangle to a 4-dimensional space. Specifically, consider a triangle $t \in \Phi$ whose vertex coordinates are $\{x_i, y_i\}$ for $i = 1, 2, 3$. Given a positive constant $\alpha$ and a non-degenerate auxiliary triangle $\tilde{t}$ with vertex coordinates $\{\tilde{x}_i, \tilde{y}_i\}$ for $i = 1, 2, 3$, both of which are specified by the user, the vertex coordinates of the 4D lifted triangle $\tilde{t}$ is constructed as $\{x_i, y_i, \sqrt{\alpha} \tilde{x}_i, \sqrt{\alpha} \tilde{y}_i\}$. Here, $\alpha$ moderates the contribution from the auxiliary triangle, so that the area of the lifted triangle approaches that of the original triangle as $\alpha \to 0$. Following the TLC framework, we chose the equilateral triangle with a unit area as the auxiliary triangle $\tilde{t}$ for every $t \in \Phi$. We denote the TLC of $\Phi$ for a given $\alpha$ as $A_\alpha(\Phi)$. As shown in [14], for any positive $\alpha$, $A_\alpha(\Phi)$ is differentiable to any order for all simplicial maps $\Phi$ – including those with degenerate triangles.

Next, we improve the smoothness of the occupancy $O(\partial \Phi)$. To avoid non-smooth states like that in Figure 3.3 (top), we compute the occupancy of a modified boundary curve. In this new curve, straight edges of $\partial \Phi$ are replaced with curved segments so that they are less likely to overlap with each other. Specifically, for each oriented edge $e \in \partial \Phi$, we consider the circular arc with $e$ as its chord and whose center angle is some constant $\theta > 0$. The arc, denoted by $\Gamma_\theta(e)$, is located on the right side of $e$ and shares the same orientation as $e$. See
Figure 3.4: An arc-edge (left) and two overlapping arc-edges (right).

Figure 3.4 (left) for illustration. We call $\Gamma_\theta(e)$ the *arc-edge* of $e$ and the curve consisting of all arc-edges the *arc-boundary* of $\partial \Phi$, denoted by $\Gamma_\theta(\partial \Phi)$.

It can be verified that, for two arc-edges to overlap (Figure 3.4 right), all of the following conditions have to hold for their straight edges $e_1, e_2$: (1) $\|e_1\| = \|e_2\|$ (so that their arc-edges have the same radius); (2) the end points of $e_1, e_2$ are co-circular and the radius of the circle is $\|e_1\| \arcsin(\theta/2)$; (3) the center of this common circle lies on the left of both $e_1$ and $e_2$; and (4) The intersection of the interior of $e_1, e_2$ is not empty. If we fix $e_1$, these conditions leave only one degree of freedom for $e_2$ so that their arc-edges overlap. In contrast, the straight edge $e_2$ has two degrees of freedom to stay overlapped with $e_1$. So the overlapping of arc-edges is less likely than the overlapping of straight edges. Importantly, if $e_1, e_2$ share a common boundary vertex, the four conditions cannot hold simultaneously, and hence their arc-edges never overlap. As a result, the arc-boundary never becomes singular during the common transition depicted at the top of Figure 3.3, as shown at the bottom of the figure.

We define the *arc-occupancy* of the boundary $\partial \Phi$ as the occupancy of the arc-boundary $\Gamma_\theta(\partial \Phi)$ minus the additional areas introduced by the arcs. We call the region bounded by each edge $e$ and its arc-edge a “flap” (see Figure 3.4 left). Let $B_\theta(\partial \Phi)$ be the sum of all flap areas. The arc-occupancy, $O_\theta(\partial \Phi)$, is then

$$O_\theta(\partial \Phi) = O(\Gamma_\theta(\partial \Phi)) - B_\theta(\partial \Phi).$$

(3.2)

As shown in Proposition 25 in Appendix B.2, the occupancy of a curve undergoing a piecewise smooth deformation is $C^1$ continuous except where the curve is singular. Also, since each flap area is proportional to the square of the corresponding edge length, which is a smooth
function of $\partial \Phi$, so is $B_\theta(\partial \Phi)$. We conclude that the arc-occupancy is a $C^1$ continuous function of $\partial \Phi$ except when two arcs of the arc-boundary $\Gamma_\theta(\partial \Phi)$ overlap.

Finally, our full energy is constructed by replacing the unsigned area and occupancy in the excess area (Equation 3.1) by TLC and arc-occupancy, respectively:

$$E_{\alpha,\theta}(\Phi) = A_\alpha(\Phi) - O_\theta(\partial \Phi).$$  \hspace{2cm} (3.3)

We refer to this energy as smooth excess area (or SEA). SEA is $C^1$ continuous for all simplicial maps $\Phi$ whenever $\partial \Phi$ has no overlapping arc-edges. In the special case where the entire boundary $\partial \Phi$ is constrained, minimizing $E_{\alpha,\theta}(\Phi)$ is then equivalent to minimizing $A_\alpha(\Phi)$, as the arc-occupancy is a constant. In this sense, SEA is an extension of TLC to free-boundary mapping.

It remains to show that SEA, just like excess area, promotes injectivity. We will establish several theoretical results in this regard with corresponding experimental evidence in Section 3.5. We first show that SEA is an upper bound of excess area, which in turn is an upper bound of both the overlapping and inverted triangle areas (by Proposition 8). Furthermore, SEA is zero when the map is globally injective, $\alpha = 0$, and $\theta$ is sufficiently small. More precisely (see proof in Appendix B.3):

**Proposition 9.** For any simplicial map $\Phi$ of a triangular mesh, $E_{\alpha,\theta}(\Phi) \geq A_{\text{excess}}(\Phi)$ for all $\alpha \geq 0, \theta > 0$. Furthermore, if $\Phi$ is globally injective and inversion-free, there exists some $\theta_0 > 0$ such that $E_{0,\theta}(\Phi) = 0$ for all $\theta < \theta_0$.

The statement above shows that the SEA energy generally promotes injectivity. As our main result, we offer a more precise characterization of injectivity at the global minima of the energy. The following proposition shows that, if a globally-injective map exists for the given input, any map $\Phi$ achieving the global minimum of the energy is guaranteed to be not only locally injective but also arbitrarily close to being globally injective in the following sense: the total amount of overlap between triangles, if there is any, can be made arbitrarily small by choosing sufficiently small $\alpha$ and $\theta$ (see proof in Appendix B.4):

**Proposition 10.** Let $\Phi_0$ be an injective, inversion-free simplicial map of a triangular mesh $M$, and $S$ a subset of vertices of $M$ such that $S$ includes at least two vertices from each edge-connected component of triangles of $M$. For any $\lambda > 0$, there exists some $\alpha_0 > 0$ and $\theta_0 > 0$ such that for any $\alpha \in (0, \alpha_0)$ and $\theta \in (0, \theta_0)$, $E_{\alpha,\theta}(\Phi_0) < E_{\alpha,\theta}(\Phi)$ for any simplicial
map $\Phi$ that satisfies $\Phi(S) = \Phi_0(S)$ but is not locally injective, or not inversion-free, or $A_{overlap}(\Phi) > \lambda$.

This result extends the local-injectivity guarantee given in [14] for TLC, which is limited to fixed-boundary mapping, to the more general setting of arbitrary positional constraints and additionally offers bounds on the overlap area. Similar to the injectivity guarantee for TLC, the above guarantee for SEA considers only the global minima and may require impractically small parameter values ($\alpha$ and $\theta$). Nevertheless, as our experimental results will show, SEA is generally effective in promoting injectivity in practice.

We make one final remark on the shape of the map promoted by SEA. [14] shows that, as $\alpha$ approaches 0, a locally injective map $\Phi$ that minimizes TLC tends to also minimize a conformal distortion measure, namely the sum of the Dirichlet energy of the linear transformation from each triangle $t \in \Phi$ to its auxiliary triangle $\tilde{t}$. It is easy to see that the same property holds for globally injective maps that minimize SEA. In particular, Proposition 9 shows that $E_{0,\theta}(\Phi) = 0$ for all globally injective $\Phi$ and sufficiently small $\theta$. Hence the map that achieves the minimal SEA as $\alpha$ increases from zero minimizes the partial derivative, $\partial E_{\alpha,\theta}/\partial \alpha$ at $\alpha = 0$. The derivative is the same as $\partial A_\alpha/\partial \alpha$, which was shown in [14] to be the conformal measure mentioned above. As our auxiliary triangles are equilateral, as in [14], SEA has the tendency to promote equilateral triangles.

### 3.4 Algorithm

The SEA energy is $C^1$ continuous almost everywhere, and as a result it can be optimized using standard gradient-based methods. We adopt the quasi-Newton (QN) method employing the standard limited-memory BFGS solver [55].

Despite our theoretical guarantees, descent-based solvers like QN have no guarantee of reaching a global minima of SEA, as the energy is non-convex. Furthermore, an applied choice of SEA parameters ($\alpha$ and $\theta$) can be larger than that required by our guarantee. As our objective is to find an injective map, we stop our QN solver when either (1) a globally injective and inversion-free map $\Phi$ is found, or (2) the optimization converges, or (3) a maximum number of iterations is reached (when not otherwise specified we use 10,000). To check for
(1), we adopt the criteria given in [46], namely that $\Phi$ contains no degenerate or inverted triangles and the boundary $\partial \Phi$ is free of intersections.

Optimization requires evaluation of the SEA energy $E_{\alpha,\theta}(\Phi)$ as well as its gradient in $\Phi$. The energy is the sum of three terms, the TLC energy $A_{\alpha}(\Phi)$, the total flap area $B_{\theta}(\partial \Phi)$, and the (negative) occupancy of the arc-boundary $\partial \Phi$. The formula for TLC and its gradient can be found in [14], while the flap area has a simple expression as a summation over all boundary edges $e \in \partial \Phi$,

$$B_{\theta}(\partial \Phi) = \sum_{e \in \partial \Phi} \frac{||e||^2(\theta - \sin \theta)}{4(1 - \cos \theta)}.$$

To compute the occupancy of the arc-boundary $\Gamma_{\theta}(\partial \Phi)$, we compute its arrangement and the winding number of each region in the arrangement. The occupancy is the sum of area of all regions with a positive winding number. To compute the arrangement, we first perform pairwise intersections of all arc-edges on $\Gamma_{\theta}(\partial \Phi)$. As mentioned earlier, the quadratic complexity of this step can be made output-sensitive using a more sophisticated algorithm [10]. The intersections break $\Gamma_{\theta}(\partial \Phi)$ into closed loops of arc segments. If the mesh has multiple boundaries, some region of the arrangement may be surrounded by multiple arc loops. We identify loops that bound the same region based on the containment relation between each pair of loops. Given two loops, their containment relation is obtained by computing the winding number of one loop around a point on the other loop. Finally, the winding number of each region with respect to $\Gamma_{\theta}(\partial \Phi)$ is obtained by propagating from the exterior region (whose winding number is zero) to adjacent regions while incrementing or decrementing by 1 depending on the orientation of the arcs on the common boundary between the regions.

The area of a region bounded by a loop of arcs can be computed as the sum of the area of the polygon formed by the arcs’ chords and the areas of the flaps bounded by each pair of arc and its chord. Specifically, denote the region as $R$ and the sequence of arcs bounding the region as $\{a_1, \ldots, a_n\}$. Note that each arc is either a complete arc-edge of $\Gamma_{\theta}(\partial \Phi)$ or a portion of it (due to intersection with other arc-edges). We ignore the orientation of the arc-edge that each $a_i$ lies on and assign $a_i$ a new orientation so that $R$ is on its left. Let $c_i$ be the chord of the arc $a_i$, and assign it the same orientation as $a_i$. See Figure 3.5 (a) for an illustration. The area of $R$ is the signed area of the polygon $\{c_1, \ldots, c_n\}$ (Figure 3.5 (b)) plus the sum of the signed areas bounded by each arc $a_i$ and its chord $c_i$ in the reverse
orientation (Figure 3.5 (c)). Both signed areas can be expressed as functions of the end
points of the arcs, each of which in turn either is a vertex of $\partial \Phi$ or, if it is the intersection
of two arc-edges, can be expressed as a function of the four vertices of $\partial \Phi$ defining those
arc-edges. The gradient of $R$’s area with respect to vertices of $\partial \Phi$ can then be derived using
the chain rule.

![Figure 3.5: The area of a region bounded by arcs (a) is the sum of the signed area of the
oriented chords (b) and the signed areas bounded by each pair of oriented arc and its chord
in reverse orientation (c). Positive and negative areas in (b,c) are colored green and red.](image)

### 3.5 Results

We evaluate our method on a benchmark test set of real-world examples and then analyze its
behavior in detail on a set of challenging, hand-crafted, stress-test cases. Our implementa-
tion currently employs an off-the-shelf, limited-memory BFGS quasi-Newton implementation
(NLOpt [36]) for energy optimization, with energy and gradient evaluation and assembly im-
plemented in C++. We use Eigen for matrix operations and OpenMP for parallelizing the
pairwise intersection of arc-edges (for computing arc-occupancy), which is the most time-
consuming step in our method.

Our SEA energy is parameterized with two terms: $\alpha$ for defining TLC and the center angle
$\theta$ of the arc-edges. As expected we generally see that smaller values of $\alpha$ are more successful
in reaching injectivity, with the concurrent cost of slower convergence for stiffer energy. This
is consistent with the behavior for $\alpha$ in TLC [14]. Similarly, we find that while larger $\theta$
(and hence more “bulgy” arc-edges) enable easier optimization steps, this must be balanced
against the improving likelihood of reaching injectivity with smaller $\theta$. To set parameters
for the evaluation, we first apply a parameter sweep across a set of 50 test examples. All
tuning examples are created via the same method we employ to create the benchmark itself – see Section 3.5.1 below. Of these 50 tuning examples, 7 are then re-used as part of our final 1791 example benchmark. We find best success obtained with $\theta = 0.1$ and $\alpha$ set to $10^{-4}$ the average of unsigned triangle area in the given initial map (assuming that each auxiliary triangle in TLC is an equilateral triangle of unit area). We use these settings in all of the following experiments.

In all subsequent figures, constrained vertices are colored blue, intersection points between boundary edges are colored orange, overwound vertices are colored magenta, and degenerate or inverted triangles are colored red. Recall that a locally injective map has no degenerate or inverted triangles or overwound vertices, and global injectivity further requires that the boundary has no self-intersection. Please see the accompanying video for animations showing optimization sequences for many of these examples.

3.5.1 Benchmark

We create a benchmark of 1791 examples for evaluating free-boundary injective mapping with positional constraints. Each example in our benchmark includes a non-injective initial map and a set of constrained vertices. For all examples we ensure that a globally injective map satisfying imposed constraints always exists. Our examples are randomly sampled from Du et al.’s [14] dataset, which in turn is derived from [47]. The dataset consists of over ten thousand 3D surface meshes, each associated with a globally injective map to the plane. For each sampled mesh $M$ and its associated map $\Phi$, we randomly pick up to 20 vertices of $M$ as the constraint set $S$, and set their target locations as $T = \Phi(S)$. This ensures that a constraint-satisfying, globally injective map exists (i.e., $\Phi$). Next, we ignore $\Phi$ and compute a new map $\Phi'$ from $M$ to the plane using ARAP [34] while satisfying the same constraints (i.e., $\Phi'(S) = T$). If $\Phi'$ is not injective, it is included in our benchmark as an initial map. This process is illustrated in Figure 3.6. Observe that the ARAP map $\Phi'$, as in (c), can differ significantly from the globally injective map $\Phi$ that comes with the mesh, as in (b).

Table 3.1 reports the success rate of our SEA-based method on the benchmark. For each example, our method is successful in achieving global injectivity if the algorithm terminates with a globally injective map, and successful in achieving local injectivity if a locally injective map is found at any iteration before the algorithm terminates. We observe that our algorithm
achieves global injectivity for a large majority of all examples (85%). We also see that the QN solver reaches our pre-set, maximum number of iterations (10,000) for the vast majority of examples where global injectivity is not achieved, indicating that the optimization is converging too slowly. After running for an additional 10,000 iterations on these failure cases, we see that the success rate improves slightly to 88%.

We also consider, see Table 3.1, the behavior of both the SA [20] and LBD [38] methods on our benchmark. Both methods seek to compute inversion-free mappings. For each we use the code provided by their respective authors with default settings. Here, the SA code requires a 2D injective “rest” mesh. In our tests we provide a Tutte embedding of the 3D surface mesh. The LBD code requires setting an upper bound, $K$, on distortion. We (in consultation with the LBD authors) set $K$ to twice the maximum distortion between the 3D surface mesh and the known injective map. This ensures that a feasible solution with respect to the bound exists. We see that, across examples in the benchmark, SA and LBD successfully achieve inversion-free mappings on just a small fraction, respectively 9.9% and 13.5%, of the benchmark examples. They are only able to achieve locally injective mappings on an even smaller proportion of examples, respectively 3.6% and 6%. This is in contrast to SEA’s 90% success rate for local injectivity. Finally, both SA and LBD achieve global injectivity (largely by chance as they are not designed to find it) on an even smaller set of examples.
Figure 3.7: Three successful examples from the benchmark: initial maps (first column), maps produced by our method (second column), which are all globally injective, and maps produced by LBD (third column) and SA (last column), none of which are locally or globally injective.

To more closely examine the behavior of these methods, we first take a closer look at some of the specific benchmark example results. Figure 3.7 demonstrates several representative examples where our optimization of SEA succeeds in recovering global injectivity. Here we observe that initial maps often contain large areas of overlaps, complex and highly wound boundaries, and many inverted triangles. For the same examples, SA and LBD do not achieve locally injective maps.

Figure 3.8 then demonstrates two representative failure cases for our method. We observe that the majority of failures with boundary intersections in our results are caused by a “crossing-arm” configuration, as highlighted by the green boxes, where one part of the shape crosses over another part. In the following section we analyze this mode further (see Figure 3.14). Likewise, some optimization results can terminate with extremely skinny, inverted triangles, as highlighted by the cyan boxes, which is another source of slowed or stalled convergence.
<table>
<thead>
<tr>
<th>Inversion free</th>
<th>SEA (Our method)</th>
<th>SA</th>
<th>LBD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1629 (91.0%)</td>
<td>179 (9.9%)</td>
<td>243 (13.5%)</td>
</tr>
<tr>
<td>Local injectivity</td>
<td>1618 (90.3%)</td>
<td>65 (3.6%)</td>
<td>108 (6.0%)</td>
</tr>
<tr>
<td>Global injectivity</td>
<td>1537 (85.8%)</td>
<td>56 (3.1%)</td>
<td>20 (1.1%)</td>
</tr>
</tbody>
</table>

Table 3.1: Statistics on number of benchmark examples (out of 1791) that each method succeeds in achieving respectively (per row) an inversion-free, locally injective, or globally injective map.

Figure 3.9 (a) visualizes the runtime of our method on our benchmark examples as a scattered plot. These statistics are collected on a Intel Core i9 CPU at 3.7GHz with 64 GB memory. We see that the majority of successful examples finish within one or two minutes, with the remaining taking up to ten or twenty minutes. Likewise, failure examples (gray dots) are, of course, most time consuming as they utilize the full maximum number of allowed iterations. Running time for examples increases both with the size of the mesh (the horizontal axis) and the number of solver iterations (the coloring of dots). We take a closer look at the slowest successful example in the entire benchmark in (c,d), which corresponds to the large red dot in (a). This mesh has 31,444 vertices, and optimization took 8,852 iterations and 893 seconds to obtain an injective map.

A significant part of our computation is spent in obtaining the arrangement of the arc-boundary. Due to performing pair-wise intersections between arc-edges, arrangement computation time is quadratic to the number of boundary vertices. We note that this certainly could be significantly optimized in future implementations. The remainder of the algorithm, such as evaluating TLC and performing a QN solve, runs in time roughly linear to the total number of vertices. This analysis is confirmed in the plot of Figure 3.9 (b), which shows a linear correlation between two ratios - the ratio of the two components of running time (arrangement versus the rest), and the ratio of the squared number of boundary vertices over the number of all vertices. Observe that, for meshes where the second ratio is less than 50, which are most examples in the benchmark, arrangement computation usually takes no longer than the rest of the algorithm.

### 3.5.2 SEA optimization behavior

Next we further illustrate and analyze the behavior of our method on a set of hand-crafted, small-scale stress-test examples. To better study the energy and convergence behavior of
Figure 3.8: Two failure examples from the benchmark: initial maps (first column) and non-globally-injective maps produced by our method (second column).

our solver, and solely for these following examples, we continue to run our QN solver (even after it reaches a globally injective map) until either the solver stagnates or else a maximum number of iterations (10,000) is reached.

A key design choice that we made for our SEA energy is to replace the occupancy of the boundary, \( O(\partial \Phi) \), by the smoother arc-occupancy measure, \( O_{\theta}(\partial \Phi) \). In Figure 3.10 we compare results of our SEA energy with one that uses occupancy instead of arc-occupancy; i.e., \( O_{\theta}(\partial \Phi) \) in Equation 3.3 is replaced by \( O(\partial \Phi) \). For this example we set two constraints and the initial map in (b) has two inverted triangles. When using occupancy rather than arc-occupancy, the solver stops at 59 iterations with a line-search error (insufficient decrease) in the solver – indicative of encountering a sharp energy transition. Correspondingly we confirm that indeed, at this iteration, two adjacent boundary edges nearly overlap (highlighted by the green oval in (c)), which means that the solver is close to a \( C^0 \) transition of the occupancy energy. At this point the resulting map, shown in (c), is not yet globally injective due to a
Figure 3.9: Performance: (a) log running time in seconds (vertical axis) versus vertex count (horizontal) for all benchmark examples. Each dot represents one mesh and is colored by number of QN iterations (gray dots are fails). (b) Ratio of arrangement computation time over the remaining time (vertical axis) versus ratio of squared number of boundary vertices over total number of vertices (horizontal axis) for all successful examples. (c,d) The initial and optimized maps in our slowest successful example (largest red dot in (a)).

boundary intersection. In contrast, optimization with SEA using arc-occupancy is able to continue until the energy converges to a globally injective map (see (d)). Again, observe that triangles are close to equilateral when SEA converges, due to our use of equilateral auxiliary triangles.

In Figure 3.11 we visualize iterations of the SEA optimization process for the same example. Here, with our default parameter setting of $\theta = 0.1$, the arc-edges would appear very close to the straight edges. For a better visualization, in Figure 3.11 we set $\theta = 1.0$ so that the arc-edges are easier to see. Even at this large $\theta$, SEA converges to the same globally injective map as $\theta = 0.1$ (Figure 3.10 (d)), but at a slower rate (requiring 763 iterations instead of 259 iterations at $\theta = 0.1$). Observe that the map becomes globally injective very quickly (around 10 iterations), and the energy continues to reduce during the remainder of the iterations.
The optimization processes for both $\theta$ values (0.1, 1.0) can be found in the accompanying video.

We next consider the two examples in Figure 3.12, where the initial maps have intersecting boundaries but no inverted triangles. In both maps, one of the constrained vertices is also overwound with an angle sum of respectively $4\pi$ (top) and $6\pi$ (bottom). Our method is able to achieve global injectivity in both cases. Note that methods that seek only non-inverted triangles would not correct the input in these examples.

Figure 3.13 shows two stress-test examples that are even more challenging for recovering injectivity. We create each example by taking an injective (rest) map and dragging a few constraint vertices to faraway target locations. Note that the mesh in the second example has two boundaries. Our method succeeds in recovering global injectivity from both non-injective initial maps. While methods like SA [20] and LBD [38] can remove (most) triangle inversion in these examples, they leave behind other types of non-injectivity such as boundary intersections and overwound vertices.
Figure 3.11: Intermediate results of optimizing SEA energy for the example in Figure 3.10 at \( \theta = 1.0 \). Arc-edges are colored green. The energy converges at iteration 763 and produces the same map as in Figure 3.10 (d).

As discussed in the last section, our optimization of the SEA energy may converge very slowly or even stall for certain geometric configurations. One such configuration, which we find represents the majority of our failure cases in our benchmark (e.g., Figure 3.8), is the “crossing-arm” case illustrated in Figure 3.14. In each of the three examples (a,b,c), one arm of the shape crosses over the other arm in the initial map (top). As the extent of crossing increases from the first example to the last, the convergence of the SEA energy slows, until the solver reaches 10,000 iterations without converging or producing a globally injective map. For the last example (c), we let the solver continue to run and found it converging at nearly 70,000 iterations with a non-injective map (see the accompanying video for the optimization processes). This behavior can be explained by recalling that SEA penalizes the area of overlapping triangles, which in these examples are where the two arms cross. However, shrinking the crossing region (e.g., in (c)) creates a bottleneck that prevents the arms from making larger moves that are necessary for reaching global injectivity.

### 3.5.3 Impact of initialization

As our energy is non-convex, the behavior of a descent-based solver like QN may vary significantly for different initial configurations. As expected, our method is generally more
Figure 3.12: Two initial maps with overwound vertices (top) and the injective maps produced by our method (bottom).

successful in reaching injectivity for initial maps that are closer to an injective state and free from the crossing-arm configurations. Figure 3.15 shows two different initial maps for the hand mesh in Figure 3.7 with the same set of constraints, one using harmonic mapping instead of ARAP (top), and a more extreme initialization created by first mapping the hand mesh into a circle using Tutte embedding and then dragging the constraint vertices to their target locations (bottom). While our method succeeds in finding a globally injective mapping in the first case, it fails in the second case, where the result contains skinny and inverted triangles (highlighted in green boxes) similar to the failure case shown in Figure 3.8.

3.5.4 Minimizing distortion

Besides injectivity, minimizing mapping distortion is another key consideration for parameterization quality. While our method is primarily designed for recovering injectivity, the resulting map can then serve as a starting point for existing, injectivity preserving, distortion-minimization methods that require initialization with a starting injective embedding [68, 35, 16]. As an example, starting from the SEA-optimized injective map of the hand mesh in Figure 3.7, Figure 3.16 shows the result of minimizing the Symmetric Dirichlet energy [68] using the simplicial augmentation framework of [35]. This method continuously deforms the mesh to reduce distortion while preserving both the injectivity of the map and the point constraints. Observe that the total distortion of the map is significantly reduced, and features of the hand (e.g., the fingers) are recovered, while the entire map remains globally injective. Our method is particularly useful in this constrained mapping scenario, as injective initial mappings cannot be obtained through the classical Tutte embedding.
Figure 3.13: From left to right: input meshes (with one or two boundaries), non-injective initial maps, globally injective maps produced by our method (SEA), and results of LBD and SA. Both LBD and SA have removed most or all inverted triangles, but the results are neither locally or globally injective due to overwound vertices and boundary intersections.

3.6 Discussion

We have presented, to our knowledge, the first method to recover global injectivity from an non-injective map subject to arbitrary positional constraints. To do so we have constructed the SEA energy, a new joint measure of overlap and inversion, sufficiently smooth to directly support gradient-based optimization. SEA comes equipped with a guarantee that maps minimizing it will be locally injective with a bounded area of overlap between non-adjacent triangles. SEA can then be simply and directly minimized with existing, off-the-shelf optimization codes, without extra customization. Results then demonstrate that doing so significantly outperforms state-of-the-art methods in achieving local injectivity while, at the same time, recovering global injectivity with a high success rate.

SEA is clearly just a first step towards robust global injectivity recovery. The ”crossing-arm” modes we discuss above remain challenging in our current optimization. One direction to explore then is constructing higher-order, Newton-type optimizers for improved convergence in these and other slower cases. Alternately, further energy modifications encouraging gradients to better resolve such overlapped regions are also promising alternatives to consider. Likewise, as we do not require inversion-free initialization, scaffolding-type solutions are an
Figure 3.14: The “crossing-arm” configuration: initial maps with increasing extent of crossing (top row, from left to right), and maps produced by our method (middle row) where only the first two are globally injective. The energy plots are shown at the bottom.

interesting possibility to better incorporate the complement space. Furthermore, while our energy currently fixes $\alpha$ and $\theta$ for the entire map, adapting them to individual triangles and boundary edges could open up new opportunities in improving convergence and boosting success rate.

Along with improved convergence, extending our method to 3D is also an important next step. Here, for example, we can consider generalizing our occupancy definition to 3D, as our TLC term is already suitable for 3D. To do so we require a comparable 3D proxy for boundary faces to optimize occupancy smoothly. Analogously to our current arc-based solution, it is tempting to subtend a spherical patch over each boundary face. However, interfaces formed by intersecting spheres among adjacent boundary faces are more complex when compared to arcs in 2D, making this initial strategy potentially much less practical. A comparable method for injectivity recovery in 3D remains an exciting avenue for future exploration.
Figure 3.15: Changing initialization type: results of our method on two alternate initializations of the hand mesh in Figure 3.7 while applying the same constraints. Injectivity is achieved starting from the top initial map but not for the bottom one (green boxes highlight the inverted triangles).

Figure 3.16: Left: output of an SEA-optimized injective map for the hand mesh in Figure 3.7. Right: further minimizing the SEA-generated map with the Symmetric Dirichlet (SD) distortion energy via the method of [35], which preserves global injectivity and point constraints. Triangles are colored by their SD distortion.
Chapter 4

Reducing Map Distortion

4.1 Introduction

This chapter concerns recovering injective, low-distortion mappings of triangular and tetrahedral meshes under given constraints. The ability to compute such maps is crucial in a wide range of applications in computer graphics and geometry processing, such as texture mapping [32], remeshing [4], deformation [78], shape correspondence [39] and physical simulation [13].

Such tasks often involve three different requirements of the given map: 1) it usually needs to satisfy *positional constraints*, such as mapping a specific vertex of the mesh to a specific location. For instance, in texture mapping, a surface mesh may need to be mapped to a domain with a prescribed boundary, and additionally some internal vertices are required to align with corresponding locations in the texture image; 2) The map should have *low distortion*, preserving the shape of the original mesh as much as possible; 3) the map is one-to-one, i.e., *injective*, meaning that the mapped mesh does not have elements that overlap one another and each element has a positive area (or volume).

Unfortunately, satisfying all three properties simultaneously proves to be quite difficult: on one hand, there is no known method which is guaranteed to produce injective mappings for the given constraints. On the other hand, distortion optimization techniques usually require a *feasible initialization* (i.e., an initial map which is injective and satisfies the constraints). In some cases, even a feasible initial map may not be a good initializer for optimization methods, as high-distortion triangles lead to numerical issues which hinder convergence.

A few recent methods show great success in recovering injectivity by minimizing energies that are specifically designed to promote injectivity. Du et al. [14] proposed to minimize the Total
Figure 4.1: When used to parameterize meshes into the 2D plane, our isometric variants of injectivity energies empirically tend to yield injective results with low isometric distortion. Left: an initial map of Lucy into letter S, which contains many inverted triangles (red), is optimized while holding its boundary fixed, using the TLC energy [14], the Fold-Free Mapping (FFM) method of [23], and our energy - IsoTLC. Right: an initial map, which contains inverted triangles (red), overwinding vertices (purple) and global overlaps (boxed), is optimized while fixing only a set of sparse positional constraints (blue) without constraining the boundary, by minimizing either the SEA energy of [14] or our energy, IsoSEA. In both cases, our method leads to the injective map with the lowest isometric distortion \((\max(\sigma_1, 1/\sigma_2))\) of the triangles, as shown by the histograms and the color map on the mesh.

*Lifted Content* (TLC) energy to compute injective mapping with a fixed boundary. TLC is a smooth approximation of the total unsigned area (or volume) of the mapped elements. This energy was generalized to enable mapping triangular meshes to the plane with arbitrary positional constraints [15]. The generalized energy, called *Smooth Excess Area* (SEA), is a smooth proxy of the total inverted and overlapping area. Both TLC and SEA are readily minimized by standard gradient-based solvers. However, a key limitation of both energies is that their minimization may result in significant isometric distortions (see Figure 4.1).

Recently, Garanzha et al. [23] introduced a new energy for recovering inversion-free, constraint-satisfying maps that also have low isometric distortions. The authors apply the penalty technique of [22] on a barrier energy to obtain a smooth function that heavily penalizes inverted elements while reducing both angle and area distortions of non-inverted elements. Using a customized solver, minimizing the energy yields maps with much lower distortions than [14] while retaining its robustness in restoring injectivity (see Figure 4.1 left).
However, the penalty energy does not consider overlaps between triangles, and hence it cannot be used as-is to recover injectivity when the boundary is not fully constrained.

Concurrent to our work, Wang et al. [73] presented a technique for free-boundary mapping of 2D meshes under positional constraints. Instead of proposing a new energy, the authors combine several existing methods (e.g., [35, 69, 6]) into a practically effective pipeline for achieving global injectivity with low distortions.

As reviewed above, while the variational approach has been extensively used for recovering injectivity, energies that promote injective, low-distortion, and constraint-satisfying maps remain scarce. This work makes another step towards filling this gap.

We propose a modification to the energies introduced in TLC [14] and SEA [15], which augments them from solely inducing injectivity to also reducing distortion. The modified energies, which we call Isometric TLC (IsoTLC) and Isometric SEA (IsoSEA), inherit the desirable traits of TLC and SEA: they are well-defined for both injective and non-injective maps, readily minimized (with a fixed parameter) using standard gradient-based solvers, and equipped with provable properties at global energy minima (even though such minima are unlikely to be reached in practice due to the non-convexity of the energies).

Our energies are evaluated on both fixed-boundary and free-boundary mapping benchmarks, and the resulting maps exhibit significantly lower isometric distortion than TLC and SEA (see Figure 4.1) while maintaining similarly high success rates in recovering injectivity. Furthermore, maps minimizing IsoTLC in the fixed-boundary benchmark typically have lower distortion than those produced by [23], most notably in 2D. Lastly, we show that our method can produce initial maps that facilitate distortion-optimization algorithms to achieve better convergence than starting from other initializers.

Our method can be used on its own in tasks which require low distortion maps, or otherwise can be used in tandem with other map optimization techniques where our result can serve as a good initializer.
4.2 Method

4.2.1 Preliminaries

**Problem statement.** We assume to be given a rest mesh $M$ whose elements are $d$-dimensional simplices (e.g., triangles, tetrahedra, etc.), embedded in $\mathbb{R}^n$ for $n \geq d$. We assume that every element of $M$ is positively oriented (i.e., having a positive $d$-dimensional volume) and no two elements overlap in their interior. The boundary of $M$, $\partial M$, may consist of one or multiple connected components. Additionally, we are given a set of positional constraints, described as pairs of a vertex index and its desired target position in $\mathbb{R}^d$.

Our output is a piecewise-linear map $T : M \rightarrow \mathbb{R}^d$, i.e., a map which is linear over each element of $M$. $T$ is represented via an assignment of new coordinates to each vertex of $M$, and we alternate between referring to $T$ as both the map and the mapped mesh. We aim to output a map $T$ that meets the following criteria, if it exists:

1. **Constraint-satisfying:** The positional constraints are all met.

2. **Globally injective (and non-inverting):** The mesh $T$ has only positively oriented elements, and no two elements of $T$ overlap in their interior.

3. **Low-distortion:** The map $T$ should minimize both angle and area distortions.

Our approach modifies the injectivity-inducing energies TLC [14] and SEA [15] to also encourage low distortion. The core observation used in the derivation of both energies is that a piecewise-linear map $T$ is injective if and only if all its elements are positively oriented and the boundary map $\partial T$ is injective [46]. Hence, these energies are formulated so as to ensure triangles have correct orientation. The boundary map can either be set to be injective via positional constraints, leading to the TLC energy, or otherwise be optimized along with the triangles’ orientation, leading to the SEA energy.

**TLC for fixed-boundary mapping** TLC assumes a given, fixed target boundary $\partial T$, possibly with additional interior constraints, and creates a smooth, robust proxy for the sum of unsigned volumes of all elements. The proxy is constructed by lifting the vertex coordinates of each $d$-dimensional simplex $t$ of $T$ to $2d$ dimensions and measuring the volume of
the lifted simplex. Specifically, lifting is controlled by two parameters, a positively oriented
$d$-dimensional auxiliary simplex $\tilde{t}$ and a non-negative scalar $\alpha$. Each vertex of the lifted sim-
plex has coordinates $\{x_1, \ldots, x_d, \sqrt{\alpha} \bar{x}_1, \ldots, \sqrt{\alpha} \bar{x}_d\}$, where $\{x_1, \ldots, x_d\}$ are the coordinates
of a vertex of $t$ and $\{\bar{x}_1, \ldots, \bar{x}_d\}$ are the coordinates of the corresponding vertex of $\tilde{t}$. The
volume of the lifted simplex is called the lifted content of $t$ and denoted by $A_{\tilde{t},\alpha}(t)$. Du et al.
[14] show that, for any $\alpha > 0$, $A_{\tilde{t},\alpha}(t)$ is always positive and smooth, even if $t$ is degenerate or
inverted [14]. The Total Lifted Content (TLC) of a mesh $T$, given a set of auxiliary simplices $\tilde{T}$, one for each element of $T$, is then defined as the sum,

$$A_{\tilde{T},\alpha}(T) = \sum_{t \in T} A_{\tilde{t},\alpha}(t)$$

Du et al. [14] prove that, for $d = 2, 3$ and assuming an injective map exists for the given boundary map, the minimizer of $A_{\tilde{T},\alpha}(T)$ is injective for any choice of auxiliary simplices $\tilde{T}$ and sufficiently small values of $\alpha$.

**SEA for free-boundary mapping.** SEA [15] tackles cases where a triangular mesh ($d = 2$) is to be mapped injectively but without necessarily constraining its boundary curve. They construct an energy that smoothly approximates the inverted and overlapping triangle areas. They define the excess area of $T$ as

$$\sum_{t \in T} |A(t)| - O(\partial T),$$

where $A(t)$ is the signed area of a triangle $t$, and $O(C)$ is the occupancy of a closed curve $C$ defined as the total area of the plane where the winding number is positive (the winding number of $C$ around a point is the number of times that $C$ travels around the point). The excess area is zero if and only if there is zero inverted or overlapping triangle area, which is equivalent to injectivity except for the presence of degenerate triangles.

A smooth proxy is devised by replacing the first term of the excess area with TLC and the second term with the arc-occupancy $O_\theta(\partial T)$, defined on a new curve that replaces each edge of $\partial T$ with an arc of center angle $\theta$, leading to Smooth Excess Area (SEA),

$$A_{\tilde{T},\alpha,\theta}(T) = A_{\tilde{T},\alpha}(T) - O_\theta(\partial T).$$
Du et al. [15] proved that, for sufficiently small $\alpha$ and $\theta$, SEA is minimized by a locally injective map with bounded total overlapping area, where a map $T$ is \emph{locally injective} if it has only positively oriented elements and no two \emph{vertex-adjacent} elements overlap.

We now begin deriving the modifications to the above energies in order to make them distortion-reducing.

### 4.2.2 Isometric TLC

\textbf{Distortion analysis.} We first analyze the relation between the TLC energy [14] and the map’s distortion when the boundary of the map is fully constrained. To do so, we introduce a singular-value form of the lifted content, $A_{t,\alpha}(t)$. Denote $L$ the matrix that transforms the edge vectors of $\tilde{t}$ to the corresponding vectors of $t$, and $\{\sigma_1, \ldots, \sigma_d\}$ its singular values. Let $A_{\tilde{t}}$ be the volume of the auxiliary simplex $\tilde{t}$ (note that $A_{\tilde{t}} > 0$). We show in the Appendix C.1 that,

$$ A_{t,\alpha}(t) = A_{\tilde{t}} \sqrt{\prod_{i=1}^{d} (\sigma_i^2 + \alpha)} . \quad (4.4) $$

![Image showing R(t), R^{iso}(t), and penalty energy](image)

Figure 4.2: Visualizing $R(t)$ (a), $R^{iso}(t)$ at different $\alpha$ values (b,c), and the penalty energy of [23] at different $\epsilon$ (d,e,f) as functions of the singular values $\sigma_1, \sigma_2$ of the linear transform $L$ from $\tilde{t}$ to $t$ ($\sigma_1$ is given the sign of $\det(L)$). Observe that $R(t)$ is minimized by a similarity transform ($\sigma_1 = \sigma_2$), $R^{iso}(t)$ is minimized by an isometry ($\sigma_1 = \sigma_2 = 1$) regardless of $\alpha$, and the minimizer of the penalty energy drifts away from isometry as $\epsilon$ increases.

It is helpful to examine the \emph{residue} of the lifted content of $t$ after subtracting its signed volume,

$$ R(t) = A_{t,\alpha}(t) - A(t) $$
$$ = A_{\tilde{t}} \left( \sqrt{\prod_{i=1}^{d} (\sigma_i^2 + \alpha)} - \det(L) \right) . \quad (4.5) $$
Since the lifted content \( A_{t,\alpha}(t) \) approximates the unsigned volume \(|A(t)|\), the residue \( R(t) \) approximates the inverted volume of \( t \). Note that \( R(t) \) is a smooth function in \( t \), and its sum over all elements of \( T \) differs from TLC only by the total signed volumes, which is a constant if the boundary is fully constrained. As a result, the sum of \( R(t) \) over all elements shares the same energy minimizer as TLC.

We next show that \( R(t) \) is minimized by a similarity transformation in two dimensions and by a singular map in higher dimensions (see proof in Appendix C.2):

**Proposition 11.** For any \( \alpha > 0 \), \( R(t) \geq \alpha^2 A_{t} \). Equality holds when either of the following holds:

1. \( d = 2 \), \( \sigma_1 = \sigma_2 \) and \( \det(L) \geq 0 \).
2. \( d > 2 \) and \( \sigma_1 = \ldots = \sigma_d = 0 \).

We visualize the residue \( R(t) \) for \( d = 2 \) in Figure 4.2 (a) as a function of \( \sigma_1, \sigma_2 \) (\( \sigma_1 \) is given the sign of \( \det(L) \)). Observe that the function is smoothly defined even when \( t \) is inverted (\( \det(L) < 0 \)) or degenerate (\( |\det(L)| = \sigma_1 \sigma_2 = 0 \)), and it reaches its minimum when \( \sigma_1 = \sigma_2 \). Note that such minimizers are scale-invariant, which explains the extremely small triangles after minimizing TLC (see Figure 4.1 left).

**Energy modification.** To reduce the area distortions created when minimizing TLC, we introduce another smooth measure that approximates the unsigned volume \(|A(t)|\). Unlike the lifted content, the residue of this measure after subtracting \( A(t) \) is minimized by an isometry. The measure, which we call the **isometric lifted content**, has the form:

\[
A_{t,\alpha}^{iso}(t) = \sqrt{A(t)^2 + \frac{\alpha}{2^{d-1}} A_{t,1}^{2} + \alpha^2 A_{t}^{2}}. \tag{4.6}
\]

Inside the square root is a weighted sum of squares of the volume of the simplex \( (A(t)) \), the lifted content at unit scale \( (A_{t,1}(t)) \), and the volume of the auxiliary simplex \( (A_{t}) \). Using the singular-value form of the lifted content (Equation 4.4), we obtain an alternative form of isometric lifted content,

\[
A_{t,\alpha}^{iso}(t) = A_{t} \sqrt{\prod_{i=1}^{d} \sigma_i^2 + \frac{\alpha}{2^{d-1}} \prod_{i=1}^{d} (\sigma_i^2 + 1) + \alpha^2}. \tag{4.7}
\]

73
It is easy to see that, like the lifted content, the isometric lifted content is smoothly defined over $t$ and greater than $|A(t)|$ for any $\alpha > 0$. We now analyze its residue after subtracting $A(t)$,

$$
R^{\text{iso}}(t) = A^{\text{iso}}_{T,\alpha}(t) - A(t) = A_t\left(\sqrt{\prod_{i=1}^{d} \sigma_i^2 + \frac{\alpha}{2^{n-1}} \prod_{i=1}^{d} (\sigma_i^2 + 1)} + \alpha^2 - \det(L)\right). \tag{4.8}
$$

As shown below, $R^{\text{iso}}(t)$ is minimized in any dimension $d$ only when $t, \tilde{t}$ are congruent (see proof in Appendix C.2):

**Proposition 12.** For any $\alpha > 0$, $R^{\text{iso}}(t) \geq \alpha A_t$, and equality holds only when $\sigma_1 = \ldots = \sigma_d = 1$ and $\det(L) > 0$.

We visualize $R^{\text{iso}}(t)$ for $d = 2$ in Figure 4.2 (b,c) for two different $\alpha$ values (0.1,1). Observe that $R^{\text{iso}}(t)$, like $R(t)$, is smooth for all $t$, but it reaches minimum at $\sigma_1 = \sigma_2 = 1$. We further compare with the penalty energy of [23] defined over a single triangle. This energy is a weighted sum of angle and area distortion measures (we set weight $\lambda = 1$ as suggested in the paper), and it is controlled by a parameter $\epsilon$. As $\epsilon$ decreases, the energy approaches a barrier function that is infinite when the triangle is degenerate or inverted. We visualize their energy in Figure 4.2 (d,e,f) for three different values of $\epsilon = 0.1, 0.5, 1$. Observe that the penalty energy is smoothly defined and penalizes triangle inversion. However, its energy minimum drifts away from isometry ($\sigma_1 = \sigma_2 = 1$) towards the singular map ($\sigma_1 = \sigma_2 = 0$) as $\epsilon$ increases. In contrast, the minimum of $R^{\text{iso}}(t)$ stays at isometry regardless of $\alpha$.

We define the *Isometric TLC* (IsoTLC) of a mesh $T$ as the sum of isometric lifted content of all elements of $T$,

$$
A^{\text{iso}}_{T,\alpha}(T) = \sum_{t \in T} A^{\text{iso}}_{T,\alpha}(t). \tag{4.9}
$$

A key property of TLC for $d = 2, 3$, as mentioned earlier, is the injectivity of maps attaining the global minimum, if the problem is feasible (again, we note that reaching the global minimum in practice is unlikely due to the non-convexity of the energy). IsoTLC shares the same property for any $d \geq 2$ (see Appendix C.3):

**Proposition 13.** Let $T_0$ be an injective map with a fully constrained boundary and possible interior constraints. Then there exists some $\alpha_0 > 0$ such that $A^{\text{iso}}_{T,\alpha}(T) > A^{\text{iso}}_{T,\alpha}(T_0)$ for any $\alpha < \alpha_0$ and any non-injective map $T$ satisfying the same constraints.
We compare the results of minimizing TLC and IsoTLC in Figure 4.3. The rest mesh $M$ is used as the auxiliary elements $\tilde{T}$ for both energies. Observe that minimizing IsoTLC at different $\alpha$ results in injective maps with similar appearances, all with much lower distortions than maps minimizing TLC.

![Figure 4.3](image)

Figure 4.3: Optimizing an initial map of the Bunny mesh into a non-convex boundary by minimizing (till convergence) the TLC energy and our IsoTLC variant (at different $\alpha$). The histograms show the triangle distortions using the measure $\max(\sigma_1, 1/\sigma_2)$ (assuming $\sigma_1 \geq \sigma_2$).

### 4.2.3 Isometric SEA

**Distortion analysis.** To see how the SEA energy [15] is related to map distortion, consider a map $T$ with a non-intersecting boundary $\partial T$. According to Lemma C.1 in [15], the arc-occupancy $O_{\theta}(\partial T)$ reduces to the occupancy $O(\partial T)$ for any $\theta < \theta_0$ where $\theta_0$ is a positive constant determined by $\partial T$. Since $\partial T$ is non-intersecting, $O(\partial T)$ is simply the area bounded by the curve, which equals the sum of signed triangle areas of $T$. So, for such a pair $\{T, \theta\}$,
SEA reduces to a measure of the inverted triangle areas,

\[ A_{\tilde{T},\alpha,\theta}(T) = A_{\tilde{T},\alpha}(T) - \sum_{t \in \tilde{T}} A(t) = \sum_{t \in T} R(t). \]  

(4.10)

Here \( R(t) \) is the residue defined in Equation 4.5, which is minimized by a similarity transformation from the auxiliary simplex \( \tilde{t} \). As a result, minimizing SEA on a mesh with an intersection-free boundary tends to suppress angle distortion from the auxiliary simplicies \( \tilde{T} \), but at the cost of possibly significant area distortions (see Figure 4.1 right).

**Energy modification.** To penalize isometric distortions, we modify SEA by replacing the TLC term with the proposed IsoTLC,

\[ A_{\tilde{T},\alpha,\theta}^{\text{iso}}(T) = A_{\tilde{T},\alpha}^{\text{iso}}(T) - O(\partial T). \]

(4.11)

We call this variant the *Isometric SEA* (IsoSEA). Following the reasoning above, for any map \( T \) with a non-intersecting boundary and some range of small \( \theta \), IsoSEA reduces to the sum of residues \( R^{\text{iso}}(t) \) over all elements \( t \in T \). Since \( R^{\text{iso}}(t) \) is minimized by an isometry from \( \tilde{t} \), minimizing IsoSEA has the effect of reducing both angle and area distortions from \( \tilde{T} \).

IsoSEA preserves the key properties of SEA. Since IsoTLC, like TLC, is smooth over the space of all maps, IsoSEA maintains the smoothness of SEA. Furthermore, we can show that IsoSEA inherits the same theoretical property as SEA: assuming an injective constraint-satisfying map exists, and for sufficiently small \( \alpha \) and \( \theta \), IsoSEA is minimized only by a locally injective map with a bounded total overlapping area. The property is formally stated as follows (see proof in Appendix C.4):

**Proposition 14.** Let \( T_0 \) be an injective, triangular map satisfying the given constraints. For any \( \lambda > 0 \), there exists some \( \alpha_0 > 0 \) and \( \theta_0 > 0 \) such that, for any \( \alpha < \alpha_0, \theta < \theta_0 \), \( A_{\tilde{T},\alpha,\theta}^{\text{iso}}(T) > A_{\tilde{T},\alpha,\theta}^{\text{iso}}(T_0) \) for any map \( T \) that is not locally injective or whose overlapping area is greater than \( \lambda \).
4.2.4 Optimization

Our variants of TLC and SEA, like the original energies, can be readily minimized using standard gradient-based methods. We implemented a quasi-Newton (QN) method using an off-the-shelf BFGS solver [55] for both IsoTLC and IsoSEA. Since IsoTLC has higher-order smoothness, we adopted the projected-Newton (PN) method in [14] that ensures positive-definiteness of the global Hessian matrix by projecting per-simplex Hessians. We derived analytical expressions for the gradient and (projected) Hessian of IsoTLC from the singular-value form of the isometric lifted content (Equation C.15) using the technique of [67] (see Appendix C.5, for details).

Despite our theoretical analysis of map injectivity at the minima of IsoTLC and IsoSEA, gradient-based solvers have no guarantee of reaching the global minimum of these non-convex energies. In practice, we adopt a two-step approach that first attempts to find an injective map and, if successful, then lowers its distortion:

1. Computing injective maps: Following the same strategy in [14, 15], we first run QN until an injective map is found, or the energy has converged, or a maximum $N$ iterations is reached (we use $N = 10000$). If IsoTLC is being minimized and an injective map is not found, we repeat the same process with PN from the initial map.

2. Lowering distortion: If the previous step produces an injective map, we then run PN (for IsoTLC) or QN (for IsoSEA) to continue minimizing the energy for another $N$ iterations or until the energy converges. If the resulting map is non-injective, we output the last injective map obtained during the iterations.

4.3 Results

We evaluate our IsoTLC and IsoSEA energies on existing benchmarks for both fixed-boundary and free-boundary mapping, and we compare the injectivity and distortion of the resulting maps with existing methods. We implemented the optimization strategy in Section 4.2.4 in C++. Eigen was used for matrix operations. Evaluating the arc-occupancy term in SEA and IsoSEA requires computing the pairwise intersection of circular arcs. We follow [15]
Figure 4.4: Histograms (in log-log scale) of maximum (top) and average (bottom) distortion of maps computed by different methods on the fixed-boundary 2D (a) and 3D (b) benchmarks and the free-boundary benchmark (c) (only including examples where both SEA and IsoSEA succeeded in recovering injectivity).

and used OpenMP to parallelize this step. More implementation details are provided in Appendix C.5.

The only parameters in our method are scalars $\alpha, \theta$. We observed that while a larger $\alpha$ generally leads to smoother energy landscapes and faster convergence, optimizing with a smaller $\alpha$ is more likely to reach an injective map (which is consistent with Propositions 13, 14). We found that setting $\alpha = 10^{-4}$ in both IsoTLC and IsoSEA and $\theta = 0.1$ in IsoSEA maximizes the success rate in recovering injectivity within the allowed number of iterations.

We use the rest meshes $M$ provided in the benchmark as the auxiliary simplices $\tilde{T}$, which enables both IsoTLC and IsoSEA to reduce isometric distortions from $M$. In contrast, equilateral triangles (and tetrahedra) were used as auxiliary simplices for minimizing TLC and SEA in [14, 15]. The choice was made to boost the success rate in recovering injectivity, as these energies are not concerned with map distortion.

To measure isometric distortion, we consider $\max(\sigma_1, 1/\sigma_2)$ in 2D and $\max(\sigma_1, 1/\sigma_3)$ in 3D. This measure reaches the minimum of 1 only when $t$ is congruent with $\tilde{t}$. We define the maximum and average distortion of a map as the maximum and mean per-element distortion over all elements.
4.3.1 Fixed-boundary mapping benchmark

We first evaluated the IsoTLC energy on the benchmark data set in [14], which consists of more than 10,000 2D examples and more than 900 3D examples of fixed-boundary mapping. Most examples were taken from real-world parameterization and deformation problems, but some of them are hand-crafted stress tests such as mapping a complex mesh into the outline of a letter (e.g., Figure 4.1 left). Each example comes with a rest mesh and an initial non-injective map into a fixed boundary, and all examples are known to have feasible (injective) solutions.

Our method achieved 100% success rate in recovering injectivity on this benchmark. To the best of our knowledge, only the original TLC method [14] and the method of [23] (which we call FFM) have passed this benchmark with complete success. Furthermore, the maps produced by minimizing IsoTLC exhibit much lower isometric distortion than both of these methods. As seen in the histograms Figure 4.4 (a,b), both the maximum and average distortion of 2D maps minimizing IsoTLC are a few orders of magnitude lower than those minimizing TLC or produced by FFM. Similarly observations can be made for 3D maps, except in the case of average distortion where IsoTLC and FFM are similar.

Figure 4.1 left shows one example from the 2D benchmark and compares the results of the three methods. Observe from the histograms that the TLC-minimizing map contains many elements with high distortion – up to seven orders of magnitude higher than elements in the IsoTLC-minimizing map or that produced by FFM. On the other hand, minimizing IsoTLC produces more elements with lower distortion than FFM.

Figure 4.5 shows several 2D examples from the fixed-boundary benchmark. The initial maps in these examples contain tens to hundreds inverted triangles. While TLC, FFM and our IsoTLC method successfully recover injectivity in all examples, IsoTLC achieves the lowest isometric distortion among the three methods.

4.3.2 Free-boundary mapping benchmark

We next evaluate the IsoSEA energy on the benchmark data set of [15], which consists of nearly 1800 examples of mapping triangular meshes onto the plane with arbitrary positional
Figure 4.5: Comparing maps computed by TLC [14], FFM [23], and our method (IsoTLC) on several 2D examples in the fixed-boundary benchmark of [14]. Each example consists of a rest mesh and an initial map containing inverted triangles (red). Histograms of per-element distortion are shown using the distortion measure $\max(\sigma_1, 1/\sigma_2)$.

constraints. Each example comes with a rest mesh (a triangular mesh in 3D), a constraint set of up to 20 vertices and their designated locations in the plane, and an initial map that satisfies those constraints but contains inverted or overlapping triangles. As in the fixed-boundary benchmark, each example in this benchmark has a feasible (injective and constraint-satisfying) solution.

Our method successfully recovered injectivity for 82% of examples in this challenging benchmark. This rate is slightly lower than the original SEA method [15], which was successful on 85% of examples. Figure 4.6 examines two examples, one on which SEA succeeded but IsoSEA failed, and one on which IsoSEA succeeded but SEA failed. In the former case, the IsoSEA failure was caused by two parts of the mesh deeply crossing each other, which is a typical situation of slow convergence for both IsoSEA and SEA (see an illustrative example in Figure 15 of [15]). In the latter case, the SEA failure was caused by several extremely small
inverted triangles, a consequence of high isometric distortions. In contrast, by promoting
isometry, IsoSEA successfully resolved all inverted triangles in this region in the initial map.
Note that both methods were significantly more successful than previous methods designed
to suppress only inverted triangles, such as [20] and [38], which recovered injectivity for less
than 4% of examples in this benchmark as reported in [15].

![Diagram showing comparison between Init, SEA, and IsoSEA](image)

Figure 4.6: Examples from the fixed-boundary benchmark where SEA succeeded in recover-
ing injectivity from the initial map but IsoSEA failed (top) or vice versa (bottom). Top
inserts: a region where IsoSEA produces boundary intersections (orange). Bottom inserts:
a region where SEA results in inverted triangles (red).

For those benchmark examples where both SEA and IsoSEA successfully produced injective
maps, maps minimizing our IsoSEA energy exhibited significantly reduced isometric disor-
tion than those minimizing SEA, as shown in the histograms of Figure 4.4 (c). Figure 4.1
right visually compares the results of the two methods on one example from the benchmark.
Observe from the histograms that the SEA-minimizing map contains elements with up to
two orders of magnitude higher distortion than elements in the IsoSEA-minimizing map.
Figure 4.7: Comparing maps computed by SEA [15] and our method (IsoSEA) on several examples in the free-boundary benchmark of [15]. Each example consists of a rest mesh, a set of constrained vertices (blue), and an initial map containing inverted triangles (red), overwound vertices (purple), and global overlaps. Histograms of per-element distortion are shown using the distortion measure $\max(\sigma_1, 1/\sigma_2)$.

Figure 4.7 shows several examples from the free-boundary benchmark. Each example contains up to 20 constrained vertices, and the initial map has inverted triangles, overwound vertices, and large areas of overlap between triangles. While both SEA and our IsoSEA method successfully recover injectivity in all examples, IsoSEA achieves significantly lower isometric distortions.

### 4.3.3 Performance

We report the running time of our algorithm on the fixed-boundary and free-boundary benchmarks as a function of mesh size in Figure 4.8. The timing was recorded on a Intel Core i9 CPU at 3.7GHz with 64 GB memory. We also separately report the timing for the first stage of our algorithm (computing an injective mesh).
Figure 4.8: Running time (seconds in log scale) versus mesh size (number of vertices) for all examples in the fixed-boundary 2D and 3D benchmarks and the free-boundary benchmark. Orange dots are times of the entire algorithm while blue dots are times of the first stage (computing injective maps).

Observe that the complexity of our algorithm generally grows with the mesh size. IsoSEA takes much longer to minimize than IsoTLC, as it needs to compute the arrangement of the boundary curve for the arc-occupancy term. These timings are similar to those of minimizing TLC and SEA as reported in [14, 15] and are comparable to those of [23] on the fixed-boundary benchmark. Finally, note that the second stage of our algorithm (lowering distortion of an already injective map) usually takes much longer than the first stage, since it only terminates at energy convergence or the maximum number of iterations.

4.3.4 Initializing map optimization

With the ability to produce injective maps satisfying positional constrains, our method as well as [14, 15, 23] can serve as the starting point for many existing constrained map optimization methods that require an injective initial mesh. Most of these optimization methods require computing the gradient or Hessians of some non-linear energy. Such computation may encounter numerical issues if the elements are too small, which in turn leads to slow or stalled convergence. As a result, the lower isometric distortions offered by our method can improve the convergence of these optimization methods.

As an example, we compare TLC and IsoTLC as initializers for fixed-boundary map optimization in Figure 4.9. Minimization TLC from an initially non-injective map (from Lucy to letter G) recovers an injective mesh with many extremely small triangles (as seen in the histogram of per-element distortion). Further optimizing this map by minimizing the
Figure 4.9: Comparing maps produced by minimizing TLC and IsoTLC (mapping Lucy to letter G) as initialization for constrained optimization in BCQN [78] that minimizes the Symmetric Dirichlet energy. With the TLC map as the starting point, BCQN terminates in 4 iterations and the map remains highly distorted.

Symmetric Dirichlet energy [68] using a standard method [78], while keeping the boundary fixed, terminates after just a few iterations due to divergent energy and fails to improve upon the TLC result. Similar failures were observed on 6 out of the 30 examples from the benchmark of [14] that map 3D surfaces to letter-like 2D domains. In contrast, IsoTLC minimization yields an injective map with much lower distortions, and further optimization of the Symmetric Dirichlet energy was able to converge with an even lower isometric distortion.

4.3.5 Failure cases of IsoTLC

Even though our IsoTLC method successfully produced injective maps for all examples in the fixed-boundary benchmark of [14], there is no guarantee that it (or the original TLC method) will always succeed. Here we investigate several reasons that may cause IsoTLC to fail to produce injective maps.
Figure 4.10: Two fixed-boundary mapping examples (top and bottom) taken from [75] where an injective map does not exist for the given target boundary (without changing the mesh structure). Our IsoTLC method cannot produce injective maps for such problems.

The first reason is that the mapping problem does not have an injective solution for the given mesh structure and boundary (and possibly other positional) constraints. Figure 4.10 shows two such examples taken from [75], where an injective solution is known to not exist for the respective target boundary (if the mesh structure is not allowed to change). Our method cannot produce injective maps for either problem, and the resulting maps still contain flipped triangles.

The second reason is that the energy converges slowly or the solver gets stuck at a non-injective local minimum. Slow or stalled convergence often arises when the initial map is too far from an injective map. We show several such failure examples in Figures 4.11 and 4.12, both taken from [23] (except for the bottom one in Figure 4.11). In Figure 4.11, the “Lucy” mesh is mapped to the outline of letter “P” but initialized with two different randomized maps (instead of Tutte embedding as done in our main experiment on the benchmark). Optimizing TLC (using either QN or PN) fails to produce an injective map within the maximum number of iterations (10,000) starting from either initial state. Optimizing IsoTLC produces an injective map starting from one of the initial maps but fails on the other. In Figure 4.12, we took a tetrahedral mesh sandwiched between an outer cube and an inner cube and rotated those vertices on the inner cube by various degrees. Our IsoTLC method
Figure 4.11: Two fixed-boundary examples (top and bottom), mapping the “Lucy” mesh to the outline of letter “P”, initialized with two random maps. The top example is taken from [23]. IsoTLC succeeds in the top example but fails to produce an injective map within 10000 iterations (using either QN or PN) on the bottom one. TLC fails on both examples.

produces an injective map for rotations by 45°, 90° and 180° but fails for 135° (TLC succeeds only on 45° and 90°). The FFM method of [23] succeeds in producing injective maps for almost all examples in these two figures, except for the second random initialization of Figure 4.11.

4.3.6 Handling poor triangulations

Finally, we demonstrate our IsoTLC method on meshes containing poorly shaped elements in Figure 4.13. These meshes were taken from Thingi10K [77] (id: 662115), and the triangulations are highly anisotropic. To make sure that the problem is feasible, for each mesh $M$, we first employed the algorithm of [45] to find a cut as well as a globally injective map $T$ to the plane. To test IsoTLC, we fixed the boundary $\partial T$ and created the initial non-injective map by re-embedding $M$ into $\partial T$ using Tutte. We observed that both TLC and IsoTLC
succeeded in producing injective maps for all tested examples, while IsoTLC achieved much lower distortion.

4.4 Discussion

While we have improved the distortion aspect of existing energies, TLC [14] and SEA [15], our energies retain some of their limitations. First, the rate of convergence of our energies may be strongly affected by the choice of the initial map, and our method may fail to produce injective maps within the allotted number of iterations for pathological initializations (see the TLC failure in Figure 5 of [23] and SEA failure in Figure 16 of [15]. Second, like SEA, our IsoSEA energy exhibits slow convergence when mesh parts deeply cross each other (Figure 4.6), which accounts for majority of our failure cases in the free-boundary benchmark. Third, our methods cannot produce injective maps for problems that do not have a feasible solution. Some potential directions for improving the convergence rate include adapting the parameters $\alpha, \theta$ to either the mesh elements or the stage of optimization (as done in [23]) and designing higher-order variants of the arc-occupancy term in IsoSEA.
Figure 4.13: Fixed-boundary mapping of meshes with poorly shaped triangles from Thingi10K. Both TLC and IsoTLC succeeded in producing injective maps, while IsoTLC achieved much lower distortion.
Chapter 5

Conclusion

5.1 Summary

In this dissertation, we address the problem of injective mapping with positional constraints in three steps.

First, we focus on the special case where the target mesh’s boundary is fixed and has no self-intersections. We propose a new method that can reliably generate injective maps under fixed-boundary constraints. Our primary contribution is an energy called Total Lifted Content (TLC), whose minimization encourages injective mappings. The TLC energy is smoothly defined for all maps and has global minima that are injective (if such maps exist and an energy parameter is sufficiently small). In practice, the TLC energy can be optimized using gradient-based solvers without requiring any special treatment. Our method achieves a 100% success rate on a large-scale benchmark of fixed-boundary mapping in both 2D and 3D.

Next, we handle arbitrary positional constraints. Prior to our work, injective mapping with arbitrary positional constraints remained largely an open problem. We propose the first method for this problem to capture non-injectivity in the general case. Our core contribution is an energy called Smooth Excess Area (SEA), which is an augmented version of the TLC energy with a new term called arc occupancy. This allows the new SEA energy to capture both inversion and overlapping of the map. The SEA energy has several theoretical properties, such as being an upper bound for overlapping and inversion, smooth (almost everywhere), and having global minima that are locally injective with small overlapping (if such maps exist and energy parameters are sufficiently small). In practice, SEA energy can be optimized using off-the-shelf quasi-Newton solvers. By optimizing SEA, our method demonstrates a high success rate in recovering injective maps from non-injective initial maps
while satisfying positional constraints, as verified on a large-scale free-boundary mapping benchmark of triangle meshes.

Finally, we focus on reducing map distortion. Although the TLC and SEA energies are not designed for producing low-distortion maps, we found that their minima possess theoretical properties that favor angle-preserving maps. Furthermore, we propose a simple modification of the energies to promote isometric maps, resulting in IsoTLC and IsoSEA energies. These modified energies retain the injectivity properties and high success rate in practice, while also enabling the reduction of isometric distortions.

5.2 Future Directions

Our work presents several opportunities for further investigation and improvement in the context of injective mapping under constraints. We identify the following potential areas for future exploration.

5.2.1 Robust injective mapping under constraints

Our methods significantly improve the robustness of injective mapping under constraints. However, they may still fail to recover an injective map due to several reasons.

Feasibility Problem: Constraints may prevent the existence of an injective map, making it impossible to recover one. This is known as the feasibility problem. Addressing this challenge involves tackling two problems:

- Determining if the given constraints are feasible: Our TLC and SEA energies have (locally) injective global minima when the constraints are feasible. Optimizing these energies to global minima could help us check feasibility. However, these energies are non-convex, making this type of check less practical. Nevertheless, our injectivity proofs provide a first step towards a theoretical understanding of the feasibility problem. It is useful to investigate the complexity class of the determination problem. If we could show that the determination problem is NP-complete, then finding a convex
energy with an injective guarantee at minimum would be unlikely. This would provide a simple way to check feasibility despite the problem being NP-complete.

- Coping with potentially infeasible constraints in practice: Several strategies can be employed to address infeasible constraints, such as relaxing the constraints by removing or changing positional constraints, using soft constraints instead of hard constraints, or allowing for changing mesh structure. Some previous works have explored mesh refinement to improve the robustness of injective mapping, but these solutions have limitations [9, 75, 63]. Generalizing our methods to allow for changing mesh structure could improve robustness in the presence of infeasible constraints.

**Failure to produce an injective map:** Even if the problem is feasible, failure to produce an injective map may still occur due to several reasons:

- Non-injective local minimum: Our TLC and SEA energies are non-convex and could have local minima that are not injective. Developing a strategy to adapt energy parameters to different scenarios, or modifying our energies to include injectivity guarantees at local minima (in addition to global minima), may improve the success rate of recovering injective maps.

- Slow convergence rate: Addressing the slow convergence rate of the optimization process can involve modifying the energy to have higher-order smoothness, employing customized solvers to increase efficiency, designing better initialization methods using data-driven approaches, or applying coarse-to-fine strategies to jump out of undesirable local minima while improving efficiency.

**Other approaches for injective mapping with positional constraints:** Exploring alternative methods for injective mapping, such as applying our TLC and IsoTLC energies in scaffold-based methods for 2D mapping [35, 73], or generalizing our methods and scaffold-based techniques to 3D mapping, could offer promising results.

### 5.2.2 Generalizing our work

In addition to improving the robustness of injective mapping under constraints, our work can be generalized to other contexts, offering several avenues for future research. One possible
direction for future research is to investigate mapping methods for different shape representations, such as polygonal meshes (e.g., quad meshes), point clouds, or signed distance fields, which would broaden the applicability of our approach to a wider range of scenarios in graphics and related fields.

Another area of interest is exploring non-Euclidean target domains. Inter-surface mapping, where the target domain is another surface mesh, has applications in texture transfer and shape analysis [61]. Mapping surfaces to spheres could also be considered as an alternative to mapping a closed surface to the 2D plane, avoiding undesirable cut seams in certain applications [64].

Lastly, modifying our energy formulation to allow for bounded distortion [2] in the resulting maps could lead to more accurate and visually appealing results, making our approach even more versatile and useful in a variety of applications.
References

[1] P. K. Agarwal, B. Sadri, and H. Yu. Untangling triangulations through local explo-
rations. In Proceedings of the twenty-fourth annual symposium on Computational ge-


[4] P. Alliez, G. Ucelli, C. Gotsman, and M. Attene. Recent advances in remeshing of

texture mapping. Proceedings of the 18th annual conference on Computer graphics and

Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry pro-

friction for cloth animation. In Proceedings of the 29th annual conference on Computer


supernodal sparse cholesky factorization and update/downdate. ACM Transactions on

Wiley Online Library, 2017.


Appendix A

Total Lifted Content

A.1 Lifted Content and Dirichlet energy

We shall prove Propositions 6 and 7 separately in 2D and 3D. In the process, we will derive alternative expressions of the lifted content of a simplex that reveal the connections to Dirichlet energy. These expressions lead to derivative formula that are used in the proofs of both this section and the next (on injectivity).

A.1.1 The 2D case

We first re-write the lifted content for a single triangle \( t \) given an auxiliary triangle \( \tilde{t} \). Let \( X, \tilde{X}, \tilde{X} \) be the edge vectors of \( t, \tilde{t}, \tilde{t} \) as defined in Section 2.3.2, and \( L \) be the linear transformation from \( t \) to \( \tilde{t} \) (i.e., \( \tilde{X} = LX \)). We obtain

\[
\text{Det}(X^TX + \alpha \tilde{X}^T \tilde{X}) = \text{Det}(X^TX + \alpha X^T L^T L X) \\
= \text{Det}(X^T(I + \alpha L^T L)X) \\
= \text{Det}(X)^2 \text{Det}(I + \alpha L^T L) \\
= \text{Det}(X)^2(1 + \alpha \text{Tr}(L^T L) + \alpha^2 \text{Det}(L^T L)) \\
= \text{Det}(X)^2 + \alpha \text{Det}(X)^2 \text{Tr}(L^T L) + \alpha^2 \text{Det}(\tilde{X})^2
\]  \hspace{1cm} (A.1)

where \( I \) is the identify matrix and \( \text{Tr} \) denotes the matrix trace. The third and fifth identities are due to the multiplicativity of determinants (i.e., \( \text{Det}(AB) = \text{Det}(A)\text{Det}(B) \)), and the fourth identity can be verified by hand (note that \( L^T L \) is a \( 2 \times 2 \) matrix).

Let \( A(t) \) be the unsigned area of \( t \), \( A_{\tilde{t}} \) be the unsigned area of \( \tilde{t} \), and \( D(t) \) be the Dirichlet energy from \( t \) to \( \tilde{t} \). Note that \( D(t) = \text{Det}(X)\text{Tr}(L^T L)/4 \) [57]. Substituting equation (A.1)
into (2.2) gives the following equivalent expression for $E_{t,\alpha}$,

$$E_{t,\alpha}(t) = \sqrt{A(t)^2 + 2\alpha A(t)D(t) + \alpha^2 A_t^2}$$

(A.2)

By swapping $t$ with $\tilde{t}$ and repeating the same derivation, we get a symmetric expression using the Dirichlet energy $\tilde{D}(t)$ from $\tilde{t}$ to $t$:

$$E_{\tilde{t},\alpha}(t) = \sqrt{A(t)^2 + 2\alpha A_t \tilde{D}(t) + \alpha^2 A_t^2}$$

(A.3)

We now prove both Propositions 6 and 7 using the two expressions above:

**Proof.** Using expression (A.2), the derivative of $E_{\tilde{t},\alpha}(t)$ with respect to $\alpha$ has the form:

$$\frac{\partial E_{\tilde{t},\alpha}(t)}{\partial \alpha} = \frac{2A(t)D(t) + 2\alpha A_t^2}{2\sqrt{A(t)^2 + 2\alpha A(t)D(t) + \alpha^2 A_t^2}}$$

(A.4)

At $\alpha = 0$, and since $A(t) > 0$ by assumption of Proposition 6, we have:

$$\frac{\partial E_{\tilde{t},\alpha}(t)}{\partial \alpha}|_{\alpha=0} = \frac{A(t)D(t)}{\sqrt{A(t)^2}} = D(t)$$

(A.5)

which proves Proposition 6.

Next, using expression (A.3), we find the gradient $\nabla E_{\tilde{t},\alpha}(t)$ with respect to $t$ as

$$\nabla E_{\tilde{t},\alpha}(t) = \frac{2A(t)\nabla A(t) + 2\alpha A_t \nabla \tilde{D}(t)}{2\sqrt{A(t)^2 + 2\alpha A(t)\tilde{D}(t) + \alpha^2 A_t^2}}$$

$$= \frac{\frac{\partial}{\partial t}A(t)\nabla A(t) + 2\alpha A_t \nabla \tilde{D}(t)}{2\sqrt{\frac{1}{\alpha^2}A(t)^2 + \frac{2}{\alpha}A_t \tilde{D}(t) + A_t^2}}$$

As $\alpha \to \infty$, we evaluate the limit of the gradient as

$$\lim_{\alpha \to \infty} \nabla E_{\tilde{t},\alpha}(t) = \frac{A_t \nabla \tilde{D}(t)}{\sqrt{A_t^2}} = \nabla \tilde{D}(t)$$

(A.6)

which proves Proposition 7 for $d = 2$. 

[100]
A.1.2 The 3D case

Let $X, \bar{X}, \tilde{X}$ be the edge vectors of tetrahedra $t, \tilde{t}, \tilde{t}$ as defined in Section 2.3.2, and $L, \bar{L}$ be the linear transformations from $t$ to $\tilde{t}$ and from $\tilde{t}$ to $t$ (i.e., $\bar{L} = L^{-1}$). Using a derivation similar to that in Equation A.1, we obtain the identity:

$$
Det(X^T X + \alpha \bar{X}^T \bar{X}) = Det(X)^2 + \alpha \bar{X}^T \bar{X}$$

Let $V(t)$ be the unsigned volume of $t$, $V_{\tilde{t}}$ be the unsigned volume of $\tilde{t}$, $D(t)$ be the Dirichlet energy from $t$ to $\tilde{t}$, and $\bar{D}(t)$ be the Dirichlet energy transforming $\tilde{t}$ to $t$. Note that $D(t) = Det(X)Tr(L^T L)/12$ and $\bar{D}(t) = Det(\bar{X})Tr(\bar{L}^T \bar{L})/12$ [57]. Substituting these identifies into equation (A.7) and then into (2.2) gives the following equivalent expression for $E_{\tilde{t},\alpha}(t)$,

$$
E_{\tilde{t},\alpha}(t) = \sqrt{V(t)^2 + 2\alpha V(t)D(t) + 2\alpha^2 V_{\tilde{t}} \bar{D}(t) + \alpha^3 V_{\tilde{t}}^2}
$$

Note that there are two equivalent expressions in 2D (A.2,A.3), each using the Dirichlet energy in one direction ($D$ or $\bar{D}$), whereas the 3D expression above contains the Dirichlet energy in both directions.

We now prove both Propositions 6 and 7 using the two expressions above:

**Proof.** We first obtain the derivative of $E_{\tilde{t},\alpha}(t)$ with respect to $\alpha$ as

$$
\frac{\partial E_{\tilde{t},\alpha}(t)}{\partial \alpha} = \frac{2V(t)D(t) + 4\alpha V_{\tilde{t}} \bar{D}(t) + 3\alpha^2 V_{\tilde{t}}^2}{2\sqrt{V(t)^2 + 2\alpha V(t)D(t) + 2\alpha^2 V_{\tilde{t}} \bar{D}(t) + \alpha^3 V_{\tilde{t}}^2}}
$$

At $\alpha = 0$, and since $V(t) > 0$ by assumption of Proposition 6, we obtain

$$
\frac{\partial E_{\tilde{t},\alpha}(t)}{\partial \alpha} |_{\alpha=0} = \frac{V(t)D(t)}{\sqrt{V(t)^2}} = D(t)
$$

which proves Proposition 6.
Next, we obtain the gradient $\nabla E_{t,\alpha}(t)$ with respect to $t$ as

$$
\nabla E_{t,\alpha}(t) = \frac{2V(t)\nabla V(t) + 2a\nabla V(t)D(t) + 2aV(t)\nabla D(t) + 2a^2V(t)\nabla D(t)}{2\sqrt{V(t)^2 + 2aV(t)D(t) + 2a^2V(t)D(t) + \alpha^3V(t)^2}}
$$

As $\alpha \to \infty$, the limit of the gradient after scaling by $\alpha^{-1/2}$ is

$$
\lim_{\alpha \to \infty} \alpha^{-1/2} \nabla E_{t,\alpha}(t) = \frac{V(t)\nabla \tilde{D}(t)}{\sqrt{V(t)^2}} = \nabla \tilde{D}(t)
$$

which proves Proposition 7 for $d = 3$. \hfill \square

### A.2 Injectivity of global minima

We shall prove the main result, Proposition 5, separately for dimensions $d = 2, 3$. In the discussions below, we assume that the auxiliary simplices $\tilde{T}$ are fixed and non-degenerate (i.e., having positive content). For notational simplicity, and since we are concerned with a specific embedding $T$, we use subscripts in place of the functional forms been used so far. That is, we use $A_t$ (or $V_t$) to denote the unsigned area (or volume) of a triangle (or tetrahedron) $t \in T$, $D_t$ the Dirichlet energy from $t$ to its auxiliary triangle (or tetrahedron) $\tilde{t}$, and $\tilde{D}_t$ the Dirichlet energy from $\tilde{t}$ to $t$.

#### A.2.1 The 2D case

We will introduce several lemmas before proving the proposition. We start with some simple, but useful properties of the Dirichlet energy (see illustration in Figure A.1):

**Lemma 15.** Let $e$ be one edge of $t$, $h$ be the distance to $e$ from the vertex opposite to $e$, and $\tilde{h}$ be the corresponding distance in $\tilde{t}$. Then,

1. $D_t \geq A_t(\tilde{h}/h)^2/2$

2. $h/|e| \geq \tilde{h}^2/4D_t$

[102]
Proof. Since the Dirichlet energy is invariant under rotation and translation, we first rotate and translate both $t, \tilde{t}$ so that $e$ and its corresponding edge $\tilde{e}$ of $\tilde{t}$ are both aligned with the X axis of the plane (see Figure A.1). Note that $D_t = A_t \text{Tr}(L^T L)/2$, where $L$ is the linear transformation matrix from $t$ to $\tilde{t}$ (as defined in Appendix A.1). With the aligned $t, \tilde{t}$, the bottom row of $L$ is $\{0, \tilde{h}/h\}$. Hence $\text{Tr}(L^T L) \geq (\tilde{h}/h)^2$, proving the inequality in (1). The inequality in (2) can be derived from (1) by noting that $A_t = |e|h/2$. 

The following lemma shows that if an embedding contains a sufficiently small triangle, it will contain some triangle $t$ whose Dirichlet energy $D_t$ (from $t$ to $\tilde{t}$) is sufficiently large:

**Lemma 16.** For any $\delta > 0$, there exists some $\epsilon > 0$ such that if an embedding $T$ contains a triangle whose unsigned area is smaller than $\epsilon$, then $T$ must contain a triangle $t$ such that $D_t > \delta$ and $A_t D_t > \epsilon \delta$.

*Proof.* We first show that if $D_t \leq \delta$ for all $t \in T$, then all edge lengths in $T$ are lower bounded by some positive constant (invariant to the choice of $T$). For any edge $e_0$ in $T$, we find a sequence of triangles $\{t_1, \ldots, t_k\} \subseteq T$ such that $e_0$ is incident to $t_1, t_i$ and $t_{i+1}$ share a common edge (denoted by $e_i$) for $i \in [1, k-1]$, and $t_k$ contains an edge (denoted by $e_k$) on the target boundary $B$. See Figure A.2 for an illustration. Consider triangle $t_k$. Let $h$ be the height with respect to base edge $e_k$, and $\tilde{h}$ be the corresponding height in the auxiliary triangle $\tilde{t}_k$. Since $|e_{k-1}| \geq h$, $D_{tk} \leq \delta$, and due to Lemma 27(2),

$$\frac{|e_{k-1}|}{|e_k|} \geq \frac{h}{|e_k|} \geq \frac{\tilde{h}^2}{4D_{tk}} \geq \frac{\tilde{h}^2}{4\delta}$$

[103]
Since $e_k$ (being on the fixed boundary) has a fixed length invariant to $T$, and $\widetilde{h}$ is fixed, $|e_{k-1}|$ is lower-bounded by some constant. Iteratively applying this inequality to $|e_{k-2}|/|e_{k-1}|, \ldots, |e_0|/|e_1|$ shows that $|e_0|$ is lower-bounded by some constant that depends only the target boundary $B$, the auxiliary triangles $\widetilde{T}$, and the combinatorial sequence of triangles $\{t_1, \ldots, t_k\}$. Call a sequence simple if any triangle appears at most once in the sequence. Let $\sigma_{e_0}$ be the minimal lower bound of $|e_0|$ over all possible simple triangle sequences, and let $\sigma$ be any value smaller than the minimum of $\sigma_{e_0}$ over all edges $e_0$. Then $\sigma$ lower-bounds the length of any edge in any embedding $T$ where $D_t \leq \delta$ for all $t \in T$.

Next, consider an embedding $T$ where $D_t \leq \delta$ for all $t \in T$, and consider any triangle $t \in T$. Let $e$ be an edge of $t$, $h$ be the height with respect to $e$, and $\widetilde{h}$ be the corresponding height in $\widetilde{T}$. We can bound $t$'s unsigned area as follows (using Lemma 27(2), $|e| > \sigma$, and $D_t \leq \delta$):

$$A_t = \frac{|e|^2\widetilde{h}}{2|e|} \geq \frac{|e|^2\widetilde{h}^2}{8D_t} > \frac{\sigma^2\widetilde{h}^2}{8\delta}$$

Define $\epsilon = \sigma^2 \lambda^2/8\delta$, where $\lambda$ is the minimum height in all auxiliary triangles $\widetilde{T}$. Due the inequality above, if an embedding contains a triangle whose unsigned area is less than $\epsilon$, then the embedding must contain a triangle $t$ such that $D_t > \delta$.

Finally, we will show that if an embedding $T$ contains some triangle $t$ such that $D_t > \delta$, then it must contain some triangle $s$ that satisfies both $D_s > \delta$ and $A_sD_s > \epsilon\delta$. If one of $t$’s edges, $e$, has length greater than $\sigma$, and let $h$ be the height with respect to $e$, then we obtain (using Lemma 27(1)):

$$A_tD_t \geq \frac{A_t^2\widetilde{h}^2}{2h^2} = \frac{|e|^2\widetilde{h}^2}{8} > \frac{\sigma^2\lambda^2}{8} = \epsilon\delta$$
Hence $t$ is the desired triangle $s$. Otherwise, all edges of $t$ are no longer than $\sigma$. Consider the triangle sequence $\{t_1, \ldots, t_k\} \subseteq T$ defined above that links one edge $e_0$ of $t$ to the boundary of $T$. Since $|e_0| \leq \sigma$, there must exist some triangle $t_i$ in the sequence such that $D_{t_i} > \delta$. Let $t_i$ be the last such triangle in the sequence (i.e., with the highest index $i$). Since all triangles $t_j$ for $j \in [i+1, k]$ satisfy $D_{t_j} \leq \delta$, the length of edge $e_i$ must be greater than $\sigma$. Note that $e_i$ is incident to triangle $t_i$, and so applying the inequality above for $t = t_i$ and $e = e_i$ shows that $t_i$ is the desired triangle $s$.

Building on the previous lemma, the next lemma shows that if an embedding contains a sufficiently small triangle, then the rate of increase in TLC as $\alpha$ increases from 0 is sufficiently large for a non-trivial range of $\alpha$:

**Lemma 17.** For any $\delta > 0$, there exists some $\epsilon > 0$ and $\beta > 0$ such that if an embedding $T$ contains a triangle whose unsigned area is smaller than $\epsilon$, then for any $\alpha < \beta$, $\partial E_{T,\alpha}(T)/\partial \alpha > \delta$.

**Proof.** By Lemma 16, there exists some $\epsilon > 0$ such that if $T$ contains some triangle whose unsigned area is smaller than $\epsilon$, then $T$ also contains a triangle $t$ such that $D_t > 2\delta$ and $A_t D_t > 2\epsilon \delta$. We will focus on this specific triangle $t$, and show that there is some constant $\beta > 0$ (invariant to the choice of $T$) such that $\partial E_{t,\alpha}(t)/\partial \alpha > \delta$ for any $\alpha < \beta$.

Recall Equation (A.4),

$$\frac{\partial E_{t,\alpha}(t)}{\partial \alpha} > \frac{2A_t D_t + 2\alpha A_t^2}{2\sqrt{A_t^2 + 2\alpha A_t D_t + \alpha^2 A_t^2}}$$

Let $C = 2A_t D_t + 2\alpha A_t^2$. Note that $A_t, D_t$ are non-negative, we have:

$$\frac{\partial E_{t,\alpha}(t)}{\partial \alpha} > \frac{2\sqrt{A_t^2 + \alpha C}}{\alpha} > \frac{\sqrt{\frac{C}{2}}}{\sqrt{\frac{2A_t^2 C}{\alpha} + 2\alpha}}$$

The first inequality is due to $C > 2A_t D_t + \alpha A_t^2$, and the last inequality is due to $C > 2A_t D_t$.

Let $\beta = 3\epsilon/4\delta$, we will next show that $\partial E_{t,\alpha}(t)/\partial \alpha > \delta$ for any $\alpha < \beta$. We will separately examine the cases when $A_t \geq \epsilon$ and when $A_t < \epsilon$. Suppose $A_t \geq \epsilon$, and noting that $D_t > 2\delta$,
we obtain for all $\alpha < \beta$:

$$\frac{\sqrt{A_tD_t}}{\sqrt{\frac{A_t}{D_t} + 2\alpha}} = \frac{\sqrt{D_t}}{\sqrt{\frac{1}{D_t} + \frac{2\alpha}{A_t}}} > \frac{\sqrt{2\delta}}{\sqrt{\frac{1}{2\delta} + \frac{3\alpha}{2\delta}}} = \delta$$

Now suppose $A_t < \epsilon$, and noting that $D_t > 2\delta$ and $A_tD_t > 2\epsilon\delta$, we again obtain for all $\alpha < \beta$:

$$\frac{\sqrt{A_tD_t}}{\sqrt{\frac{A_t}{D_t} + 2\alpha}} > \frac{2\epsilon\delta}{\sqrt{\frac{\epsilon}{2\delta} + \frac{3\alpha}{2\delta}}} = \delta$$

Hence we conclude that $\partial E_{t,\alpha}(t)/\partial \alpha > \delta$ for any $\alpha < \beta$. The proof is completed by noting that

$$\frac{\partial E_{\tilde{T},\alpha}(T)}{\partial \alpha} = \sum_{s \in T} \frac{\partial E_{s,\alpha}(s)}{\partial \alpha} \geq \frac{\partial E_{t,\alpha}(t)}{\partial \alpha} > \delta$$

\[\square\]

Finally, we prove Proposition 5 for $d = 2$ using the previous lemma:

**Proposition 18 (5 in 2D).** Given any injective 2D embedding $T_0$, there exists some $\beta > 0$ such that $E_{\tilde{T},\alpha}(T) > E_{\tilde{T},\alpha}(T_0)$ for any non-injective embedding $T$ and $\alpha < \beta$.

**Proof.** Since $T_0$ is injective, the derivative $\partial E_{\tilde{T},\alpha}(T_0)/\partial \alpha$ is bounded. Pick an arbitrary but small positive value $\tau$, and define $\delta$ as the maximum derivative for all $\alpha < \tau$.

Now suppose $T$ has a triangle whose unsigned area is smaller than $\epsilon$, which is found by Lemma 17 for $\delta$. By that Lemma, there exists some $\gamma > 0$ such that for any $\alpha < \gamma$, $\partial E_{\tilde{T},\alpha}(T)/\partial \alpha > \delta$. As a result, the following holds for all $\alpha < \min(\tau, \gamma)$:

$$\frac{\partial(E_{\tilde{T},\alpha}(T) - E_{\tilde{T},\alpha}(T_0))}{\partial \alpha} = \frac{\partial E_{\tilde{T},\alpha}(T)}{\partial \alpha} - \frac{\partial E_{\tilde{T},\alpha}(T_0)}{\partial \alpha} > \delta - \delta = 0$$

Since $E_{\tilde{T},\alpha}(T) \geq E_{\tilde{T},\alpha}(T_0)$ (as $T_0$ achieves the least total unsigned areas), we conclude $E_{\tilde{T},\alpha}(T) > E_{\tilde{T},\alpha}(T_0)$ for any $\alpha \in (0, \min(\tau, \gamma))$.

Otherwise, suppose $T$ has no triangle whose unsigned area is smaller than $\epsilon$. Since $T$ is non-injective, it must contain at least one inverted triangle whose unsigned area is no smaller
than $\epsilon$. Due to Corollary 4, for any $\alpha > 0$,
\[
E_{T,\alpha}(T) > E_{T,0}(T) \geq E_{T,0}(T_0) + 2 * \epsilon
\]
Since the derivative $\partial E_{T,\alpha}(T_0)/\partial \alpha$ is bounded, there exists some $\kappa > 0$ such that, for all $\alpha < \kappa$,
\[
E_{T,\alpha}(T_0) < E_{T,0}(T_0) + 2 * \epsilon \leq E_{T,\alpha}(T)
\]
The proof is completed by letting $\beta = \min(\tau, \gamma, \kappa)$. \hfill \square

### A.2.2 3D

The proof in 3D proceeds similarly as in 2D, with an essentially same set of lemmas and similar proofs. Most differences appear in extending Lemmas 27 and 16 to 3D.

We start by extending the properties of the 2D Dirichlet energy in Lemma 27 to 3D, while adding a new property (3):

**Lemma 19.** Let $f$ be one face of $t$, $h$ be the distance to $f$ from the vertex opposite to $f$, $\tilde{h}$ be the corresponding distance in $\tilde{t}$, and $D_f$ be the Dirichlet energy from $f$ to its corresponding face $\tilde{f}$ in $\tilde{t}$. Then,

1. $D_t \geq V_t(\tilde{h}/h)^2/2$
2. $h/A_f \geq \tilde{h}^2/6D_t$
3. $D_t \geq D_fh/3$

**Proof.** We first rotate and translate both $t, \tilde{t}$ so that $f$ and $\tilde{f}$ are both on the XY plane, with one corner at the origin. Note that $D_t = V_tr(L^TL)/2$, where $L$ is the linear transformation matrix from $t$ to $\tilde{t}$ (as defined in Section A.1.2). After the rotation and translation, $L$ has the form:

\[
L = \begin{pmatrix}
L_f & v \\
0 & \tilde{h}/h
\end{pmatrix}
\]
where $L_f$ is the linear transformation matrix from $f$ to $\tilde{f}$. As a result, $\text{Tr}(L^T L) \geq (\tilde{h}/h)^2$, which leads to properties (1) and (2) as shown in the proof of Lemma 27, and $\text{Tr}(L_f^T L_f) \geq \text{Tr}(L^T L)$, which yields property (3) due to $D_f = A_f \text{Tr}(L_f^T L_f)/2$ and $V_i = A_f h/3$. □

Extending Lemma 16 to 3D, we argue that if an embedding contains a sufficiently small tetrahedron, it will contain some tetrahedron $t$ whose Dirichlet energy to $\tilde{t}$, $D_t$, is sufficiently large:

**Lemma 20.** For any $\delta > 0$, there exists some $\epsilon > 0$ such that if an embedding $T$ contains a tetrahedron whose unsigned volume is smaller than $\epsilon$, then $T$ must contain a tetrahedron $t$ such that $D_t > \delta$ and $V_i D_t > \epsilon \delta$.

**Proof.** We first show that if $D_t \leq \delta$ for all tetrahedra $t \in T$, then all triangle areas in $T$ are lower bounded by some positive constant (invariant to the choice of $T$). For any triangle $f_0$ in $T$, we find a sequence of tetrahedra $\{t_1, \ldots, t_k\} \subseteq T$ such that $f_0$ is incident to $t_1$, $t_i$ and $t_{i+1}$ share a common face (denoted by $f_i$) for $i \in [1, k - 1]$, and $t_k$ contains a face (denoted by $f_k$) on the target boundary $B$.

Consider tetrahedron $t_k$. Let $h$ be the height with respect to base face $f_k$, $\tilde{h}$ be the corresponding height in the auxiliary tetrahedron $\tilde{t}_k$, $e$ be the common edge shared by $f_k$ and $f_{k-1}$, and $h'$ be the height of $f_{k-1}$ with respect to base $e$. See Figure A.3 for an illustration. Since $h' \geq h$, $D_{t_k} \leq \delta$, and due to Lemma 19(2),

$$\frac{A_{f_{k-1}}}{A_{f_k}} = \frac{|e|h'}{2A_{f_k}} \geq \frac{|e|h}{2A_{f_k}} \geq \frac{|e|\tilde{h}^2}{12D_{t_k}} \geq \frac{|e|\tilde{h}^2}{12\delta}$$

Since $f_k$ is on the fixed boundary and so is $e$, both $A_{f_k}$ and $|e|$ are invariant to $T$. Since $\tilde{h}$ is also fixed, $A_{f_{k-1}}$ is lower-bounded by some constant.

We hope to lower-bound $A_{f_0}$ by iteratively applying the above inequality to the triangle sequence $A_{f_{k-2}}, \ldots, A_{f_0}$, as we have done in 2D. However, the rightmost term of the inequality also involves the length of an edge $|e|$ of $f_k$. To bound $A_{f_{k-2}}$, for example, we need to additionally show that the length of any other edge of $f_{k-1}$ (which might be incident to $f_{k-2}$) is also lower-bounded by some constant invariant to $T$. Consider some other edge $e'$ of $f_{k-1}$ (see Figure A.3). Let $\tilde{h}', \tilde{h}''$ be the corresponding heights of $h', h''$ in $\tilde{t}_k$, and $D_{f_{k-1}}$ be the Dirichlet energy from $f_{k-1}$ to its corresponding face $\tilde{f}_{k-1}$ in $\tilde{t}$. Due to Lemmas 27 (2)
and 19 (2,3),

\[
\frac{|e'|}{|e|} \geq \frac{h'}{|e|} \geq \frac{\tilde{h}'^2}{4D_{f_{k-1}}} \geq \frac{\tilde{h}'^2h''}{12D_{t_h}} \geq \frac{A_{f_{k-1}}\tilde{h}'^2\tilde{h}''^2}{72D^2_{t_h}} \geq \frac{A_{f_{k-1}}\tilde{h}'^2\tilde{h}''^2}{72\delta^2}
\]

Since $|e|, \tilde{h}', \tilde{h}''$ are fixed, and $A_{f_{k-1}}$ is lower-bounded by some constant (as shown above), $|e'|$ is lower-bounded by some constant as well. Now we can iteratively apply the two inequalities above to $A_{f_{k-2}}/A_{f_{k-1}}, \ldots, A_{f_0}/A_{t_1}$, and to the shared edges between successive triangles $f_i, f_{i-1}$ for $i \in [k - 1, 1]$, to lower-bound $A_{f_0}$ by some constant that depends only the target boundary $B$, the auxiliary tetrahedra $\tilde{T}$, and the combinatorial sequence of tetrahedra $\{t_1, \ldots, t_k\}$. Call a sequence simple if any tetrahedron appears at most once in the sequence. Let $\sigma_{f_0}$ be the minimal lower bound of $A_{f_0}$ over all possible simple tetrahedra sequences, and let $\sigma$ be any value smaller than the minimum of $\sigma_{f_0}$ over all triangles $f_0$. Then $\sigma$ lower-bounds the area of any triangle in any embedding $T$ where $D_t \leq \delta$ for all $t \in T$.

The rest of the proof proceeds like the proof of Lemma 16. Consider an embedding $T$ where $D_t \leq \delta$ for all $t \in T$, and consider any tetrahedron $t \in T$. Let $f$ be an face of $t$, $h$ be the height with respect to $f$, and $\tilde{h}$ be the corresponding height in $\tilde{T}$. We can bound $t$’s unsigned
volume as follows (using Lemma 19(2), $A_f > \sigma$, and $D_t \leq \delta$):

$$V_t = \frac{A_f^2 h}{3A_f} \geq \frac{A_f^2 \tilde{h}^2}{18D_t} > \frac{\sigma^2 \tilde{h}^2}{18\delta}$$

Define $\epsilon = \sigma^2 \lambda^2/18\delta$, where $\lambda$ is the minimum height in all auxiliary tetrahedra $\tilde{T}$. Due the inequality above, if an embedding contains a tetrahedron whose unsigned volume is less than $\epsilon$, then the embedding must contain a tetrahedron $t$ such that $D_t > \delta$.

Finally, we will show that if an embedding $T$ contains some tetrahedron $t$ such that $D_t > \delta$, then it must contain some tetrahedron $s$ that satisfies both $D_s > \delta$ and $V_s D_s > \epsilon \delta$. If one of $t$’s face, $f$, has unsigned area greater than $\sigma$, and let $h$ be the height with respect to $f$, then we obtain (using Lemma 19(1)):

$$V_t D_t \geq \frac{V_t^2 \tilde{h}^2}{2h^2} = \frac{A_f^2 \tilde{h}^2}{18} > \frac{\sigma^2 \lambda^2}{18} = \epsilon \delta$$

Hence $t$ is the desired tetrahedron $s$. Otherwise, all faces of $t$ have unsigned areas no greater than $\sigma$. Consider the tetrahedra sequence $\{t_1, \ldots, t_k\} \subseteq T$ defined above that links one face $f_0$ of $t$ to the boundary of $T$. Since $A_{f_0} \leq \sigma$, there must exist some tetrahedron $t_i$ in the sequence such that $D_{t_i} > \delta$. Let $t_i$ be the last such tetrahedron in the sequence (i.e., with the highest index $i$). Since all tetrahedra $t_j$ for $j \in [i+1, k]$ satisfy $D_{t_j} \leq \delta$, the unsigned are of triangle $f_i$ must be greater than $\sigma$. Note that $f_i$ is a face of tetrahedron $t_i$, and so applying the inequality above for $t = t_i$ and $f = f_i$ shows that $t_i$ is the desired tetrahedron $s$.

Building on the previous lemma, and extending Lemma 17 to 3D, the next lemma shows that if an embedding contains a sufficiently small tetrahedron, then the rate of increase in TLC as $\alpha$ increases from 0 is sufficiently large for a non-trivial range of $\alpha$:

**Lemma 21.** For any $\delta > 0$, there exists some $\epsilon > 0$ and $\beta > 0$ such that if an embedding $T$ contains a tetrahedron whose unsigned volume is smaller than $\epsilon$, then for any $\alpha < \beta$, $\partial E_{\tilde{T}, \alpha}(T)/\partial \alpha > \delta$.

**Proof.** By Lemma 20, there exists some $\epsilon > 0$ such that if $T$ contains some tetrahedron whose unsigned volume is smaller than $\epsilon$, than $T$ also contains a tetrahedron $t$ such that
$D_t > 2δ$ and $A_t D_t > 2εδ$. We will focus on this specific tetrahedron $t$, and show that there is some constant $β > 0$ (invariant to the choice of $T$) such that $∂E_{t,α}(t)/∂α > δ$ for any $α < β$.

Recall Equation (A.9),

$$\frac{∂E_{t,α}(t)}{∂α} = \frac{2V_t D_t + 4αV_t \bar{D}_t + 3α^2 V_t^2}{2V_t^2 + 2αV_t D_t + 2α^2 V_t \bar{D}_t + α^3 V_t^2}$$

Let $C = 2V_t D_t + 4αV_t \bar{D}_t + 3α^2 V_t^2$. Note that $V_t, V_t, D_t, \bar{D}_t$ are non-negative, we have:

$$\frac{∂E_{t,α}(t)}{∂α} \geq \frac{C}{2V_t^2 + αC} = \frac{\sqrt{C}}{2\sqrt{\frac{V_t^2}{C} + α}} \geq \frac{\sqrt{V_t D_t}}{\sqrt{\frac{V_t^2}{C} + α}}$$

Proceeding exactly as the rest of the proof of Lemma 17 (replacing $A_t$ with $V_t$), we can show that $∂E_{t,α}(t)/∂α > δ$ for any $α < β$ where $β = 3ε/4δ$.

Finally, we prove Proposition 5 for $d = 3$ using the previous lemma:

**Proposition 22 (3 in 3D).** Given any injective 3D embedding $T_0$, there exists some $β > 0$ such that $E_{\tilde{T},α}(T) > E_{\tilde{T},α}(T_0)$ for any non-injective embedding $T$ and $α < β$.

**Proof.** The proof proceeds exactly as the proof of Proposition 18, after making the dimensionality-specific replacements: “triangle” by “tetrahedron”, “area” by ” volume”, and “Lemma 17” by “Lemma 21”.

---

### A.3 Gradient and Hessian of TLC

We provide explicit expressions for the gradient and hessian of TLC. As TLC is accumulative, it suffices to consider a single simplex $t$ with an auxiliary simplex $\tilde{t}$, whose TLC is $E_{\tilde{T},α}(t)$ (which we shall shorthand as $E$). Denote the vertices of the $d$-dimensional simplex $t$ as $v_1, \ldots, v_{d+1}$, and similarly $\tilde{v}_i, \hat{v}_i$ for simplices $\tilde{t}, \hat{t}$. Here, we only consider the cases of $d = 2, 3$.

To simplify the expressions, we use an alternative formula for the volume of a simplex known as the Cayley-Menger determinant. The formula calculates the volume $E$ of the
lifted simplex, \( \hat{t} \), as \( E = \sqrt{D}/c \), where \( c \) is a dimension-dependent constant and \( D \) is the determinant of a matrix involving squared edge lengths of \( \hat{t} \). For a triangle \( t \), \( c = 4 \) and

\[
D = \text{Det} \left( \begin{array}{ccc}
2\hat{d}_{12} & \hat{d}_{12} + \hat{d}_{13} - \hat{d}_{23} & 2\hat{d}_{13} \\
\hat{d}_{12} + \hat{d}_{13} - \hat{d}_{23} & 2\hat{d}_{13} & \hat{d}_{13} + \hat{d}_{14} - \hat{d}_{34} \\
\hat{d}_{12} + \hat{d}_{14} - \hat{d}_{24} & \hat{d}_{13} + \hat{d}_{14} - \hat{d}_{34} & 2\hat{d}_{14}
\end{array} \right)
\]

where \( \hat{d}_{ij} \) is the squared edge length between vertices \( \hat{v}_i, \hat{v}_j \) (note that \( \hat{d}_{ij} = d_{ij} + \hat{d}_{ij} \), where \( d_{ij}, \hat{d}_{ij} \) are squared lengths of the corresponding edges in \( t, \hat{t} \)). For a tetrahedron \( t \), \( c = 12\sqrt{2} \) and

\[
D = \text{Det} \left( \begin{array}{ccc}
2\hat{d}_{12} & \hat{d}_{12} + \hat{d}_{13} - \hat{d}_{23} & \hat{d}_{12} + \hat{d}_{14} - \hat{d}_{24} \\
\hat{d}_{12} + \hat{d}_{13} - \hat{d}_{23} & 2\hat{d}_{13} & \hat{d}_{13} + \hat{d}_{14} - \hat{d}_{34} \\
\hat{d}_{12} + \hat{d}_{14} - \hat{d}_{24} & \hat{d}_{13} + \hat{d}_{14} - \hat{d}_{34} & 2\hat{d}_{14}
\end{array} \right)
\]

### A.3.1 Gradient

It suffices to derive the gradient of \( E \) with respect to one vertex \( v_i \), which is a length-\( d \) vector. Differentiating \( E = \sqrt{D}/c \) yields:

\[
\frac{\partial E}{\partial v_i} = \frac{\partial D}{\partial v_i} / 2c^2 E. \tag{A.12}
\]

Applying the chain rule to \( \partial D/\partial v_i \) yields:

\[
\frac{\partial D}{\partial v_i} = \sum_{j \neq i} \frac{\partial D}{\partial \hat{d}_{ij}} \frac{\partial \hat{d}_{ij}}{\partial v_i}.
\]

We provide expressions for the terms on the right-hand side. First,

\[
\frac{\partial \hat{d}_{ij}}{\partial v_i} = \frac{\partial d_{ij}}{\partial v_i} = 2(v_i - v_j). \tag{A.13}
\]

Next, if \( t \) is a triangle, and \( k \) is the vertex index other than \( i, j \), we get

\[
\frac{\partial D}{\partial \hat{d}_{ij}} = 2(\hat{d}_{ik} + \hat{d}_{jk} - \hat{d}_{ij}). \tag{A.14}
\]
Finally, if \( t \) is a tetrahedron, and \( k, l \) are the vertex indices other than \( i, j \), we get
\[
\frac{\partial D}{\partial d_{ij}} = \begin{cases} 
2(\tilde{d}_{il} - \tilde{d}_{ik})(\tilde{d}_{jl} - \tilde{d}_{ik}) + 2 \tilde{d}_{kl}(\tilde{d}_{ik} + \tilde{d}_{il} + \tilde{d}_{jk} + \tilde{d}_{jl} - \tilde{d}_{kl} - 2\tilde{d}_{ij}), & i = j, k \neq l \\
2(\tilde{d}_{ij} + \tilde{d}_{il} + \tilde{d}_{jk} + \tilde{d}_{kl} - 2\tilde{d}_{ik} - 2\tilde{d}_{jl}), & \text{otherwise}.
\end{cases}
\]

(A.15)

### A.3.2 Hessian

It suffices to derive the hessian of \( E \) with respect to two vertices \( v_i, v_j \), which is a \( d \times d \) matrix. Differentiating equation (A.12) yields:
\[
\frac{\partial^2 E}{\partial v_i \partial v_j} = \left( \frac{\partial^2 D}{\partial v_i \partial v_j} - 2c^2 \frac{\partial E}{\partial v_j} \left( \frac{\partial E}{\partial v_i} \right)^T \right) / 2c^2 E.
\]

As we already have expression for the gradient \( \partial E / \partial v_i \), we only need to provide expression for \( \partial^2 D / \partial v_i \partial v_j \). Application of the chain rule yields:
\[
\frac{\partial^2 D}{\partial v_i \partial v_j} = \sum_{k \neq i} \left( \left( \sum_{l \neq j} \frac{\partial^2 D}{\partial \tilde{d}_{ik} \partial \tilde{d}_{jl}} \frac{\partial \tilde{d}_{ij}}{\partial v_i} \right) \left( \frac{\partial \tilde{d}_{ik}}{\partial v_i} \right)^T + \frac{\partial D}{\partial \tilde{d}_{ik}} \frac{\partial^2 \tilde{d}_{ik}}{\partial v_i \partial v_j} \right).
\]

Expressions for first-order terms on the right-hand side are given in equations (A.13,A.14,A.15). We next provide expressions for the remaining second-order terms. Let \( I \) denote the identity matrix, and define \( \delta_{pq} \) as 1 if \( p = q \) and 0 otherwise. We have:
\[
\frac{\partial^2 \tilde{d}_{ik}}{\partial v_i \partial v_j} = 2(\delta_{ij} - \delta_{jk})I.
\]

Next, if \( t \) is a triangle, we have:
\[
\frac{\partial^2 D}{\partial \tilde{d}_{ik} \partial \tilde{d}_{jl}} = \begin{cases} 
-2, & i = j, k = l \\
2, & \text{otherwise}.
\end{cases}
\]

Finally, if \( t \) is a tetrahedron, and let \( p, q \) be vertex indices other than \( i, j, k, l \), we can derive:
\[
\frac{\partial^2 D}{\partial \tilde{d}_{ik} \partial \tilde{d}_{jl}} = \begin{cases} 
-4\tilde{d}_{pq}, & i = j, k = l \\
2(\tilde{d}_{pk} + \tilde{d}_{pl} - \tilde{d}_{lk}), & i = j, k \neq l \\
2(\tilde{d}_{ij} + \tilde{d}_{il} + \tilde{d}_{jk} + \tilde{d}_{kl} - 2\tilde{d}_{ik} - 2\tilde{d}_{jl}), & \text{otherwise}.
\end{cases}
\]

[113]
Appendix B

Smooth Excess Area

B.1 Excess, overlap, and inverted areas

Here we prove Proposition 8 in Section 3.3.1, which relates the excess area to the area of overlapping or inverted triangles. We start with a lemma on winding numbers, which extends a previous result (Theorem 4 of [46]) to meshes with both proper and inverted triangles:

**Lemma 23.** Let $\Phi$ be a simplicial map of a triangular mesh, and $z$ a point in the plane that is not on any vertex or edge of $\Phi$. The winding number of $\partial \Phi$ around $z$ is the number of proper triangles of $\Phi$ that cover $z$ minus the number of inverted triangles of $\Phi$ that cover $z$.

**Proof.** We recall the property of the winding number that it is 0 at any point $z$ outside the curve, and it increments (resp. decrements) by 1 as $z$ moves across the oriented curve from its right to left (resp. from its left to right). Let $pos(z), neg(z)$ denote the number of proper and inverted triangles of $\Phi$ that cover $z$. As the location of $z$ changes, the two functions change as follows:

1. If $z$ does not cross $\partial \Phi$, $pos(z)$ and $neg(z)$ either do not change, or simultaneously increase or decrease by some integer $k \geq 1$. The latter happens when $z$ crosses an interior edge or vertex of $\Phi$ that is incident to both proper and inverted triangles.

2. When $z$ crosses $\partial \Phi$ from right to left, either $pos(z)$ increases by 1 or $neg(z)$ decreases by 1. Conversely, when $z$ crosses $\partial \Phi$ from left to right, either $pos(z)$ decreases by 1 or $neg(z)$ increases by 1.

In both cases, the difference $pos(z) - neg(z)$ increases (resp. decreases) by 1 when $z$ crosses the boundary $\partial \Phi$ from right to left (resp. from left to right). Since $pos(z) = neg(z) = 0$
when z is at infinity, \( pos(z) - neg(z) \) has the same value as the winding number of \( \partial \Phi \) around z for any z on the plane that is not on the vertices or edges of \( \Phi \).

We now prove the proposition using the lemma above:

**Proof of Proposition 8:** Lemma 23 shows that a point \( z \) contributes to the occupancy \( O(\partial \Phi) \) if and only if \( pos(z) > neg(z) \). As a result, \( z \) must be covered by at least one proper triangle. Let \( A_{\text{cover}}(\Phi) \) be the total area of the plane covered by \( \Phi \), and \( A_{\text{proper}}(\Phi) \) the total area of proper triangles of \( \Phi \). Then,

\[
A_{\text{cover}}(\Phi) \geq O(\partial \Phi) \\
A_{\text{proper}}(\Phi) \geq O(\partial \Phi)
\] (B.1)

On the other hand, the overlap and inverted areas are related to \( A_{\text{cover}}(\Phi) \) and \( A_{\text{proper}}(\Phi) \) by:

\[
A_{\text{overlap}}(\Phi) = A(\Phi) - A_{\text{cover}}(\Phi) \\
A_{\text{invert}}(\Phi) = A(\Phi) - A_{\text{proper}}(\Phi)
\] (B.2)

Substituting the Equations B.1 into Equations B.2 yields the first two inequalities of the proposition. To prove the last inequality, note that points covered by \( \Phi \) but not contributing to the occupancy of \( \partial \Phi \) must be covered by inverted triangles of \( \Phi \). So we have:

\[
A_{\text{cover}}(\Phi) - O(\partial \Phi) \leq A_{\text{invert}}(\Phi).
\] (B.3)

Substituting Equation B.3 into the first equation of Equations B.2 yields:

\[
A_{\text{overlap}}(\Phi) \geq A(\Phi) - (O(\partial \Phi) + A_{\text{invert}}(\Phi)) \\
= A_{\text{excess}}(\Phi) - A_{\text{invert}}(\Phi).
\] (B.4)

\[\Box\]

**B.2 Continuity and smoothness of occupancy**

We will show that the occupancy of a curve undergoing piecewise smooth deformation is not only continuous, but also \( C^1 \) smooth except at some well-defined degenerate (singular) configurations.

[115]
Consider a 1-dimensional cell complex $M$ that consists of 1-cells (edges) and 0-cells (vertices) that form a closed loop, and a map $\Psi : \mathbb{R}^n \times M \to \mathbb{R}^2$ that takes $M$ to a closed curve in the plane under a set of parameters $X = \{x_1, \ldots, x_n\}$. As an example, if $\Psi$ is a simplicial map, the map parameters are the 2D coordinates of the mapped vertices of $M$. However, $\Psi$ is not limited to simplicial maps, and $\Psi$ may map each edge of $M$ into a smooth curve (e.g., the arcs used in our SEA energy).

Denote the occupancy of the mapped curve $\Psi(X, M)$ as a function of $X$, $O(X) = O(\Psi(X, M))$. We first show that $O(X)$ is continuous:

**Proposition 24.** If $\Psi(X, p)$ is continuous in both $X$ and $p \in M$, then $O(X)$ is continuous in $X$.

*Proof.* The winding number of a closed and oriented curve $C$ around a point $z$ can be computed as $\frac{1}{2\pi}$ of the sum of signed angles spanned by $z$ and each infinitesimal oriented segment of $C$. As $C$ changes continuously, each (infinitesimal) signed angle changes continuously except when $z$ lies on $C$. Since the winding number at $p$ is always an integer, it stays the same as long as $C$ does not pass through $p$. In other words, points whose winding numbers change (which include all those that contribute to the change in $C$’s occupancy) are restricted to the region “swept” by $C$. Since $\Psi(X, M)$ is continuous in $X$, an infinitesimal change of $X$ leads to an infinitesimal area swept by $\Psi(X, M)$, and hence the change in $O(X)$ is also infinitesimal. As a result, $O(X)$ is continuous in $X$. \hfill \Box

To analyze the smoothness of occupancy, we introduce the concept of singular maps. Let $\Psi_X(p) = \Psi(X, p)$. A point $q$ on the mapped curve $\Psi_X(M)$ is called a *singular point* if $|\Psi_X^{-1}(q)| > 1$; that is, $q$ has multiple pre-images in $M$. Intuitively, a singular point is where two segments of $\Psi_X(M)$ intersect (e.g., Figure B.1 middle). The map $\Psi$ is said to be *singular at $X$* if $\Psi_X(M)$ contains an infinite number of singular points. This happens when two segments of $\Psi_X(M)$ completely overlap (e.g., Figure B.1 right).

We show that $O(X)$ is $C^1$ continuous except at the singularities:

**Proposition 25.** If $\Psi_X(p)$ is differentiable in both $X$ and $p$ over each edge of $M$, $O(X)$ is differentiable at all $X$ where $\Psi$ is not singular.

[116]
**Figure B.1:** Comparing singular and non-singular maps.

**Proof.** Let $A = \{a_1, \ldots, a_n\}$ be a set of parameters at which $\Psi$ is not singular. Our goal is to show that the partial derivative of $O(X)$ in each parameter exists at $X = A$. The partial derivative for the $i$-th parameter is defined as the limit,

$$\frac{\partial O}{\partial x_i}(A) = \lim_{h \to 0} \frac{O(A + h\delta_i) - O(A)}{h},$$

where $\delta_i$ is a length-$n$ vector with 0s except a 1 in the $i$-th place. We will show that this limit exists.

We first derive the limit as $h$ approaches 0 from the positive side (the other side can be treated in a symmetric manner). Let $C_g = \Psi_{A+g\delta_i}(M)$ be the mapped curve for any $g \geq 0$, and $B_g \subseteq C_g$ the boundary of the regions with positive winding numbers. $B_g$ is oriented such that these positively winding regions are on the left side of the boundary. Following the same argument in the proof of Proposition 24, the change in the occupancy, $O(A + h\delta_i) - O(A)$, is the total area swept by the sequence of curves $B_g$ as $g$ increases from 0 to $h$. The sweep area is signed, so that a point that is swept from the left (resp. right) contributes positively (resp. negatively) to the sweep area. A point that is swept multiple times contributes to the sweep area with the corresponding multiplicity. Our idea is to decompose this sweep area into the sum of two type of areas, one swept by the regular subsets of $B_g$ (to be defined later) and the other swept by the remainder of $B_g$, and show that each type of area admits a well-defined one-sided derivative at $h = 0$. See Figure B.2 for an illustration.

We start with the following observation: for any point $p \in M$, its image $p_g = \Psi_{A+g\delta_i}(p)$, as $g$ changes, switches from being not on $B_g$ to being on $B_g$ (or vice versa) only at values of
Figure B.2: Illustration for the proof. Left: the mapped curves (dotted lines) and the corresponding boundaries of positively winding regions (solid lines) at different parameters (distinguished by shades of blue). Middle and right: the regions swept by the regular and non-regular subsets of $B_g$, shaded respectively in green and red, as $g$ increases from 0 to $h$. We will show that the (signed) area of both green and red regions admits a well-defined derivative at $h = 0$.

$g$ where $p_g$ is a singular point of $C_g$. This is because the winding numbers of points on the two sides of $C_g$ around $p_g$ do not change until $p_g$ is passed by another segment of $C_g$. Note that the change of winding numbers around $p_g$ does not always switch the membership of $p_g$ in $B_g$; but such a switch always happens at singular points.

Consider the function $f(q)$ that gives, for every point $q \in C_0$, the smallest $g$ for which a pre-image of $q$ on $M$ is mapped to a singular point on $C_g$. Let $S_g$ be the set of singular points of $C_g$, and $\Pi_g$ the composite map $\Psi_{A+g\delta} \circ \Psi_A^{-1}$ that takes points on $C_0$ to points on $C_g$. Note that $\Pi_g$ is only one-to-one at non-singular points of $C_0$. We define:

$$f(q) = \begin{cases} 
0, & q \in S_0 \\
\arg\min_{g > 0} \Pi_g(q) \in S_g, & q \in C_0 \setminus S_0
\end{cases}$$

By this definition, the set of points for which $f$ evaluates to zero is precisely the singular points $S_0$, which is a finite set because $\Psi$ is not singular at $A$.

We define the regular subset of the boundary $B_0$, denoted by $R_h$, as points where $f$ evaluates to be greater than $h$. In other words, $R_h$ consist of non-singular points of $B_0$ whose image under $\Pi_g$ remain non-singular for all $g \in [0, h]$. Based on the observation above, the image of the regular subset $\Pi_g(R_h)$ remains on the boundary $B_g$ for all $g \in [0, h]$, which we also call the regular subsets of $B_g$. See Figure B.2 (middle and right) for an example.
As \(g\) increases from 0 to \(h\), we call the area swept by the regular subsets \(\Pi_g(R_h)\) the \textit{regular sweep} (denoted by \(RS_h\)), and the area swept by the remaining subsets \(B_g \setminus \Pi_g(R_h)\) the \textit{non-regular sweep} (denoted by \(NS_h\)). We can re-write the limit in Equation B.5 as:

\[
\lim_{h \to 0} \frac{O(A + h\delta_t) - O(A)}{h} = \lim_{h \to 0} \frac{RS_h + NS_h}{h} = \lim_{h \to 0} \frac{RS_h}{h} + \lim_{h \to 0} \frac{NS_h}{h}, \tag{B.6}
\]

We next derive the limits for the regular and non-regular sweeps:

- **Regular sweep:** Let \(CC(R_h)\) denote the connected components of \(R_h\) after removing the image of the vertices of \(M\). Since \(\Psi_X(p)\) is differentiable along each edge of \(M\), and each connected component \(r \in CC(R_h)\) is non-singular, the map \(\Pi_g(q)\) is differentiable over \(g \in [0, h]\) and \(q \in r\). Hence the regular sweep can be expressed as the following integral,

\[
RS_h = \sum_{r \in CC(R_h)} \int \int_0^h \text{det}(\nabla \Pi_g(q)) \, dq \, dg
\]

where \(\nabla \Pi_g(q)\) is the Jacobian of \(\Pi\) at \(g\) and \(q\), and \(\text{det}\) is the determinant operator. As \(h \to 0\), \(R_h\) approaches the entire boundary \(B_0\) minus the (finite) set of singular points \(S_0\), and the ratio \(RS_h/h\) has a well-defined limit,

\[
\lim_{h \to 0} \frac{RS_h}{h} = \sum_{r \in CC(B_0 \setminus S_0)} \int \int \text{det}(\nabla \Pi_0(q)) \, dq \tag{B.7}
\]

- **Non-regular sweep:** We shall first bound the \textit{unsigned} area swept by the non-regular subsets of \(B_g\). This area is no more than the product of (1) the maximum length of the non-regular subset on \(B_g\) for any \(g \in [0, h]\), and (2) the maximal distance travelled by any point on the non-regular subsets as \(g\) increases from 0 to \(h\). We can bound each quantity as follows:

  - We first argue that, for any \(q \in C_0\) such that \(\Pi_g(q)\) (which may consist of more than one point if \(q\) is singular) intersects the non-regular subset of \(B_g\) for some \(g \in [0, h]\), \(f(q) \leq h\). To see why, suppose \(q \in B_0\), then \(q\) cannot be in the regular subset \(R_h\), and hence \(f(q) \leq h\). Otherwise, since \(q \notin B_0\) but \(\Pi_g(q) \cap B_g \neq \emptyset\), then by the observation made earlier, one of \(\Pi_{g'}(q)\) must be a singular point on \(C_{g'}\) for some \(g' \leq g\), implying that \(f(q) \leq g' \leq h\). Let \(T_h\) be the subset of \(C_0\) where \(f\)
evaluates to be no greater than $h$, and $l_h$ be the maximum length of $\Pi_g(T_h)$ over all $g \in [0, h]$. By the argument above, $l_h$ is an upper bound of (1).

- Note that (2) is bounded by $h$ times the maximum travel speed for any point on $B_g$. Since the speed is the partial derivative of $\Psi$ in $x_i$, which is bounded, (2) is bounded by $h \cdot \gamma$ where $\gamma$ is the maximum absolute value of that partial derivative over $M$ and a sufficiently large range of $g$.

Since the absolute value of non-regular sweep is no more than the unsigned area swept by the non-regular subsets, we have

$$\|NS_h\| \leq l_h \cdot \gamma \cdot h$$

As $h \to 0$, $l_h$ approaches the length of $T_h$ (due to the continuity of $\Psi$), which in turn approaches 0 since $T_0$ becomes the (finite) singular set $S_0$. Since $\gamma$ is a constant, we conclude that

$$\lim_{h \to 0} \frac{\|NS_h\|}{h} \leq \lim_{h \to 0} l_h \cdot \gamma = 0,$$

which implies

$$\lim_{h \to 0} \frac{NS_h}{h} = 0 \quad (B.8)$$

Substituting Equations B.7 and B.8 into B.6 shows that the limit in Equation B.5 exists for $h$ approaching from the positive side, and the limit equals the righthand side of Equation B.7. The case of $h$ approaching from the negative side is completely symmetric, and the limit is identical to that in Equation B.7. This proves that the partial derivative $\frac{\partial O}{\partial x_i}(A)$ exists for any $x_i \in X$. 

\[\square\]

### B.3 SEA and excess area

Here we prove Proposition 9 in Section 3.3.2, which relates the SEA energy to the excess area and injective maps. We start with a lemma that shows that the arc-occupancy of the boundary $\partial \Phi$ is a lower bound of the boundary’s occupancy:

**Lemma 26.** Let $\Phi$ be a simplicial map of a triangular mesh and $\theta > 0$, then $O_\theta(\partial \Phi) \leq O(\partial \Phi)$. Furthermore, if $\Phi$ is globally injective and inversion-free, there exists some $\theta_0$ such that $O_\theta(\partial \Phi) = O(\partial \Phi)$ for all $\theta < \theta_0$. 

[120]
Proof. Following the same argument in the proof of Lemma 23, the winding number of the arc-boundary \(\Gamma_\theta(\partial \Phi)\) around a point \(z\) is the number of positively-oriented triangles and flaps covering \(z\) minus the number of inverted triangles and flaps covering \(z\). Since each arc-edge is on the right of its straight edge and sharing the same orientation, the flaps are always positively oriented. As a result, if the winding number of \(\partial \Phi\) around \(z\) is already positive (implying that there are more proper than inverted triangles covering \(z\)), so must be the winding number of \(\Gamma_\theta(\partial \Phi)\). Conversely, if the winding number of \(\Gamma_\theta(\partial \Phi)\) around \(z\) is positive but the winding number of \(\partial \Phi\) is not, \(z\) must be covered by some flap. As a result, the region with positive winding numbers w.r.t. \(\partial \Phi\) is covered by the union of the regions with positive winding numbers w.r.t. \(\partial \Phi\) and regions covered by all flaps. This yields the inequality,

\[
O(\Gamma_\theta(\partial \Phi)) \leq O(\partial \Phi) + B_\theta(\partial \Phi).
\]

Substituting the above into Equation 3.2 yields \(O_\theta(\partial \Phi) \leq O(\partial \Phi)\). The equality holds if (i) \(B_\theta(\partial \Phi)\) equals the area of the region covered by all the flaps, which implies that there is no overlap between the flaps, and (ii) every point \(z\) covered by some flap has non-positive winding number w.r.t. \(\partial \Phi\) but positive winding number w.r.t. \(\Gamma_\theta(\partial \Phi)\), which, in conjunction with (i), implies that the winding number of \(\partial \Phi\) around \(z\) must be zero. In summary, \(O_\theta(\partial \Phi) = O(\partial \Phi)\) if no flap overlaps any other flap or any region that has a non-zero winding number w.r.t. \(\partial \Phi\).

We next show that the condition above holds for an injective and inversion-free \(\Phi\) and all \(\theta < \theta_0\) for some positive value \(\theta_0\) that can be derived from \(\Phi\). It is obvious that the condition holds for \(\theta = 0\), in which case the arc-edges are identical to the straight edges and the flaps vanish. Let \(\theta_0\) be the smallest value of \(\theta\) such that the condition no longer holds. Since \(\Phi\) is injective and inversion-free, as \(\theta\) increases from 0, the flaps expand into the region outside the boundary \(\partial \Phi\), which has zero winding number w.r.t. \(\partial \Phi\). For the condition to fail at \(\theta = \theta_0\) and not for any \(\theta < \theta_0\), the arc-edges of two edges \(e_1, e_2 \in \partial \Phi\) must come into contact at \(\theta = \theta_0\). If these \(e_1, e_2\) do not share a common vertex, then by injectivity of \(\Phi\), \(e_1, e_2\) do not intersect, and \(\theta_0\) is the smallest \(\theta\) at which the two arc-edges \(\Gamma_\theta(e_1), \Gamma_\theta(e_2)\) are tangent. If \(e_1, e_2\) share a common vertex, then \(\theta_0\) equals the exterior angle of that vertex. In either case, \(\theta_0\) is a non-zero value that can be derived from \(\Phi\).

We shall refer to the threshold angle \(\theta_0\) in Lemma 26 as the clearance angle of an injective map \(\Phi\). Now we prove the proposition:
**Proof of Proposition 9:** We note that the TLC is an upper bound of the unsigned area, that is, $A_\alpha(\Phi) \geq A(\Phi)$ for all $\alpha \geq 0$ [14]. Since $O_\theta(\partial \Phi) \leq O(\partial \Phi)$ for all $\theta > 0$ by Lemma 26, we derive, for $\alpha \geq 0, \theta > 0$:

$$E_{\alpha,\theta}(\Phi) = A_\alpha(\Phi) - O_\theta(\partial \Phi) \geq A(\Phi) - O(\partial \Phi) = A_{\text{excess}}(\Phi) \quad (B.9)$$

Suppose $\Phi$ is injective and inversion-free, and let $\theta_0$ be its clearance angle. Lemma 26 shows that $O_\theta(\partial \Phi) = O(\partial \Phi)$ for all $\theta < \theta_0$. Since $A_0(\Phi) = A(\Phi)$, the inequality in Equation B.9 becomes equality for $\alpha = 0$ and $\theta < \theta_0$, making $E_{0,\theta}(\Phi) = A_{\text{excess}}(\Phi)$. Finally, since $\Phi$ has no overlapping or inverted triangles, Proposition 8 shows that $A_{\text{excess}}(\Phi) = 0$, which proves $E_{0,\theta}(\Phi) = 0$ for all $\theta < \theta_0$.

\[\square\]

**B.4 Injectivity at SEA minima**

Here we prove the injectivity guarantee for SEA (Proposition 10). The proof closely follows that of the injectivity guarantee for TLC in [14] (Proposition 4.3). They differ when it comes to the type of constraints (fixed boundary for TLC, versus arbitrary constraints for SEA) and the scope of guarantee (local injectivity within an intersection-free boundary for TLC, versus local injectivity with bounded overlap within an arbitrary boundary for SEA).

We start by re-stating a few useful lemmas from [14]. We shall refer to [14] for the proofs of these lemmas whenever possible and only state the necessary extensions. In the following, $A_t$ denotes the unsigned area of a triangle $t$ of a simplicial map, $\tilde{t}$ is the auxiliary triangle of $t$ for defining TLC, and $D_t$ denotes the Dirichlet energy of the linear map from $t$ to $\tilde{t}$.

The first lemma states a few useful properties of the Dirichlet energy (see proof in [14]):

**Lemma 27** (Lemma B.1 in [14]). Let $e$ be one edge of $t$, $h$ be the distance to $e$ from the vertex opposite to $e$, and $\tilde{h}$ be the corresponding distance in $\tilde{t}$. Then,

1. $D_t \geq A_t(\tilde{h}/h)^2/2$
2. $h/|e| \geq \tilde{h}^2/4D_t$
The next lemma shows that a simplicial map $\Phi$ will contain a triangle with an arbitrarily large Dirichlet energy if there is some triangle of $\Phi$ with sufficiently small unsigned area:

**Lemma 28** (Lemma B.2 in [14]). For any $\delta > 0$, there exists some $\epsilon > 0$ such that if a simplicial map $\Phi$ contains a triangle whose unsigned area is smaller than $\epsilon$, then $\Phi$ must contain a triangle $t$ such that $D_t > \delta$ and $A_t D_t > \epsilon \delta$.

**Proof.** The proof of Lemma B.2 in [14] assumes that all boundary edges of $\partial \Phi$ are fixed, and thereby having a fixed length, in order to provide a lower bound of the length of interior edges. Specifically, it constructs a sequence of triangles that link each interior edge $e$ of $\Phi$ to a boundary edge $e_0$. It shows that, if $D_t < \delta$ for all $t$ in $\Phi$, the edge length $\|e\|$ is lower-bounded by $\|e_0\|$ multiplied by a constant that depends only on the combinatorial structure of the domain mesh $M$ (and not on the map $\Phi$). This edge length lower bound is then used to complete the rest of the proof.

Without a fixed boundary, we can still provide a lower bound of (interior or exterior) edge lengths, by leveraging the fact that at least two vertices (say $p, q$) are constrained within each edge-connected component of triangles of $M$. Let $P$ be a path of edges in $M$ that links $p$ and $q$. The longest edge in $\Phi(P)$, denoted by $e_0$, is therefore no shorter than the ratio of the Euclidean distance between $\Phi(p)$ and $\Phi(q)$ over the number of edges on $P$. Since $p, q$ are constrained and $P$ is independent of $\Phi$, that ratio is a constant independent of $\Phi$, and hence the edge length $\|e_0\|$ has a constant lower bound. For every other edge $e$ (interior or exterior) of $\Phi$ in the same edge-connected component as $\Phi(p), \Phi(q)$, we can construct a sequence of triangles linking $e$ to $e_0$. This sequence, just like the edge-to-boundary triangle sequence in the proof of [14], depends only on the combinatorial structure of $M$. Following the same argument therein, $\|e\|$ is lower-bounded by $\|e_0\|$ multiplied by a constant that is independent of $\Phi$. \hfill \QED

The next lemma builds on the previous one and states that the rate of increase in TLC as $\alpha$ increases from 0 can be arbitrarily large, if $\Phi$ has a sufficiently small triangle (see proof in [14]):

**Lemma 29** (Lemma B.3 in [14]). For any $\delta > 0$, there exists some $\epsilon > 0$ and $\beta > 0$ such that if a simplicial map $\Phi$ contains a triangle whose unsigned area is smaller than $\epsilon$, then for any $\alpha < \beta$, $\partial A_\alpha(\Phi)/\partial \alpha > \delta$. 

[123]
An immediate corollary of this lemma is that if $\Phi$ contains a degenerate triangle (whose area is zero), the partial derivative of TLC, $\partial A_\alpha(\Phi)/\partial \alpha$, is unbounded at $\alpha = 0$. On the other hand, [14] shows that this partial derivative is well-defined and bounded for any map without degenerate triangles and any $\alpha \geq 0$.

Before proving the proposition, we introduce a new lemma that lower-bounds the overlapping triangle areas for a simplicial map with overwound vertices.

**Lemma 30.** Suppose a simplicial map $\Phi$ contains an interior vertex $v$ such that $v$ is incident to only proper triangles and the sum of angles around $v$ is not $2\pi$. Then $A_{\text{overlap}}(\Phi) \geq \epsilon \eta^2 \pi/2\delta$, where $\epsilon$ is the minimum of $A_t$ among all $t \in \Phi$, $\eta$ is the shortest height of any auxiliary triangle $\tilde{t}$ for $t \in \Phi$, and $\delta$ is the maximum of $D_t$ for all $t \in \Phi$.

**Proof.** Since $v$ is interior and all incident triangles of $v$ are proper, the angle sum at $v$ must be a positive multiple of $2\pi$. In this case, the multiplier is greater than 1. Consider the intersection region $I$ of the half-spaces defined by the supporting lines of edges on the 1-ring boundary of $v$ (yellow region in Figure B.3). Each point in $I$ (away from the vertices and edges of $\Phi$) is covered by at least two triangles incident to $v$. Let $A(I)$ be the area of $I$, we therefore have $A_{\text{overlap}}(\Phi) \geq A(I)$.

Note that $I$ is convex and $v$ lies inside $I$. Hence $A(I)$ can be computed by summing the areas of triangles each made up of $v$ and a segment bounding $I$. Denote the perimeter of $I$ as $L(I)$, and let $h$ be the shortest distance from $v$ to any segment bounding $I$. We have the
inequality:

\[ A(I) \geq h \times L(I)/2 \]

On the other hand, since the shape with the smallest perimeter that attains a given area is a circle, we can lower-bound \( L(I) \) by \( A(I) \) as

\[ L(I) \geq 2\sqrt{A(I)\pi}. \]

Combining the two inequalities above yields

\[ A(I) \geq h^2\pi. \]

Now consider the triangle \( t \) incident to \( v \) that has \( h \) as its height. Let \( e \) denote the edge of \( t \) opposite to \( v \), and \( \tilde{h} \) be the height of the auxiliary triangle \( \tilde{t} \) corresponding to \( h \). Since \( A_t \geq \epsilon, \tilde{h} \geq \eta, \) and \( D_t \leq \delta \), and by Lemma 27, we have:

\[ h \parallel e \parallel \geq 2\epsilon, \quad \text{and} \quad h / \parallel e \parallel \geq h^2 / 4D_t \geq \eta^2 / 4\delta \]

Multiplying these two inequalities gives \( h^2 \geq \epsilon \eta^2 / 2\delta \). As a result,

\[ A_{\text{overlap}}(\Phi) \geq A(I) \geq h^2 \pi \geq \epsilon \eta^2 \pi / 2\delta \]

Now we prove the proposition using the lemmas above:

\[ \]

Proof of Proposition 10: We closely follow the approach taken in the proof of Proposition 4.3 in [14]. We separately consider the case that \( \Phi \) has some small-area triangle (including degenerate triangles) and the case that it does not. The key difference with [14] lies in the latter case, which we further divide into three sub-cases: \( \Phi \) has some inverted triangle; \( \Phi \) has overwound interior vertices; or \( \Phi \) has an overlapping area greater than \( \lambda \) that is not caused by inverted triangles or overwound vertices. The last two sub-cases are unique to our setting.

We start with a few remarks. First, since \( \Phi_0 \) is injective, Proposition 9 shows that there exists some \( \theta_0 \) (the clearance angle of \( \Phi_0 \)) such that \( E_{0,\theta}(\Phi_0) = 0 \) for all \( \theta < \theta_0 \). In the
following, we consider some fixed $\theta$ in this range, and we shorthand the SEA energy $E_{\alpha,\theta}$ as $E_\alpha$. Second, since the occupancy term of SEA does not depend on $\alpha$, SEA shares the same partial derivative as TLC with respect to $\alpha$. That is, $\partial A_\alpha(\Phi)/\partial \alpha = \partial E_\alpha(\Phi)/\partial \alpha$ for any map $\Phi$ (injective or not) and $\alpha \geq 0$. Third, since $\Phi_0$ is injective, $\partial A_\alpha(\Phi_0)/\partial \alpha$ is always bounded, and so is $\partial E_\alpha(\Phi_0)/\partial \alpha$. We pick an arbitrary but small positive value $\tau$, and let $\delta$ be the maximum value of $\partial E_\alpha(\Phi_0)/\partial \alpha$ for all $\alpha < \tau$.

We first consider the case that $\Phi$ has a triangle whose unsigned area is smaller than $\epsilon$, which is found by Lemma 29 for $\delta$. This case is already considered in the proof of Proposition 4.3 in [14], which shows that there exists some $\beta \in (0, \tau]$ such that for any $\alpha < \beta$, $\partial E_\alpha(\Phi)/\partial \alpha > \delta > \partial E_\alpha(\Phi_0)/\partial \alpha$. On the other hand, by Proposition 9, $E_0(\Phi) \geq 0 = E_0(\Phi_0)$. Hence $E_\alpha(\Phi) > E_\alpha(\Phi_0)$ for any $\alpha < \beta$.

Otherwise, suppose $\Phi$ has no triangle whose unsigned area is smaller than $\epsilon$ (which includes degenerate triangles). We can further assume that $D_t \leq 2\delta$ for any $t \in \Phi$. Otherwise, and since $A_t \geq \epsilon$, we have $D_t > 2\delta$ and $A_tD_t > 2\epsilon\delta$. Following the arguments in the proof of Lemma B.3 in [14], the conclusion of that lemma (which is re-stated as Lemma 29 above) holds; that is, $\partial A_\alpha(\Phi)/\partial \alpha > \delta$ for any $\alpha < \beta$. This would allow us to conclude that, just like the case above, $E_\alpha(\Phi) > E_\alpha(\Phi_0)$ for any $\alpha < \beta$.

Now we consider what happens if $\Phi$ is not locally injective, not inversion-free, or $A_{overlap}(\Phi) > \lambda$. If $\Phi$ is not locally injective or inversion-free, $\Phi$ either has an inverted triangle (whose unsigned area is no smaller than $\epsilon$) or has an interior vertex whose angle sum is not $2\pi$. We separately consider these sub-cases.

- Suppose $\Phi$ has some inverted triangle, whose unsigned area is no smaller than $\epsilon$. By Propositions 8 and 9, $E_\alpha(\Phi) \geq A_{inverted}(\Phi) \geq \epsilon$ for any $\alpha \geq 0$. Since $E_0(\Phi_0) = 0$ and the partial derivative $\partial E_\alpha(\Phi_0)/\partial \alpha$ is bounded, we conclude that there exists some $\kappa_1 > 0$ such that, for all $\alpha < \kappa_1$, $E_\alpha(\Phi_0) < \epsilon \leq E_\alpha(\Phi)$.

- Suppose $\Phi$ contains only proper triangles but some vertex $v$ has an angle sum other than $2\pi$. Since $A_t \geq \epsilon$ and $D_t \leq 2\delta$ for all $t \in \Phi$, Lemma 30 shows that $A_{overlap}(\Phi) \geq \sigma = \epsilon\eta^2\pi/4\delta$ where $\eta$ is the smallest height among all auxiliary triangles. Note that $\sigma$ is independent of the map $\Phi$. By Propositions 8 and 9, $E_\alpha(\Phi) \geq A_{overlap}(\Phi) \geq \sigma$ for any $\alpha \geq 0$. Similar to the previous sub-case, there exists some $\kappa_2 > 0$ such that, for all $\alpha < \kappa_2$, $E_\alpha(\Phi_0) < \sigma \leq E_\alpha(\Phi)$.

[126]
- Suppose $A_{\text{overlap}}(\Phi) > \lambda$. By Propositions 8 and 9, $E_\alpha(\Phi) \geq A_{\text{overlap}}(\Phi) > \lambda$ for any $\alpha \geq 0$. Similar to the previous sub-cases, there exists some $\kappa_3 > 0$ such that, for all $\alpha < \kappa_3$, $E_\alpha(\Phi_0) < \lambda < E_\alpha(\Phi)$.

The proof is completed by setting $\alpha_0 = \min(\beta, \kappa_1, \kappa_2, \kappa_3)$. □
Appendix C

Isometric Energies

This appendix completes the derivations and proofs absent from chapter 4 (Sections 1-4), and provides additional implementation details (Section 5).

C.1 Singular value formula for lifted content

We will derive the alternative formula of lifted content (Equation 4.4). Given two d-
dimensional simplices \( t, \tilde{t} \) (\( \tilde{t} \) has positive d-dimensional volume) and scalar \( \alpha > 0 \), the lifted content of \( t \) is defined as:

\[
A_{t,\alpha}(t) = \frac{1}{d!} \sqrt{\det(X^T X + \alpha \tilde{X}^T \tilde{X})}
\]  

where \( X, \tilde{X} \) are the edge (column) vectors of \( t, \tilde{t} \) respectively.

Let \( L \) be the linear transformation that maps \( \tilde{t} \) to \( t \), that is, \( L = X \tilde{X}^{-1} \) (note that \( \tilde{X} \) is always invertible because \( \tilde{t} \) has positive content). Applying Sylvester’s Theorem to the determinant inside the square root yields:

\[
\det(X^T X + \alpha \tilde{X}^T \tilde{X}) = \det(\alpha \tilde{X}^T \tilde{X}) \det(I_d + \frac{1}{\alpha} X \tilde{X}^{-1} (\tilde{X}^T)^{-1} X^T)
\]

\[
= \det(\tilde{X}^T \tilde{X}) \det(\alpha I_d + X \tilde{X}^{-1} (\tilde{X}^{-1})^T X^T)
\]

\[
= \det(\tilde{X}^T \tilde{X}) \det(\alpha I_d + LL^T)
\]

where \( I_d \) is the \( d \times d \) identity matrix. Consider the singular value decomposition \( L = U \Sigma V^T \), where \( U, V \) are orthonormal matrices and \( \Sigma \) is a diagonal matrix whose diagonal entries are
the singular values \( \{\sigma_1, \ldots, \sigma_d\} \). We can simplify the last determinant in Equation C.2 as:

\[
\det(\alpha I_d + LL^T) = \det(\alpha I_d + U\Sigma V^T V \Sigma^T U^T) \\
= \det(U(\alpha I_d) U^T + U\Sigma \Sigma^T U^T) \\
= \det(U(\alpha I_d + \Sigma \Sigma^T) U^T) \\
= \det(U) \det(\alpha I_d + \Sigma \Sigma^T) \det(U^T) \\
= \det(\alpha I_d + \Sigma \Sigma^T) \\
= \prod_{i=1}^{d} (\sigma_i^2 + \alpha)
\]

(C.3)

The last equality holds because \( \alpha I_d + \Sigma \Sigma^T \) is a diagonal matrix whose diagonal values are \( \{\sigma_i^2 + \alpha, \ldots, \sigma_d^2 + \alpha\} \). Substituting Equation C.3 into Equation C.2 and then into Equation C.1 yields:

\[
A_{t,\alpha}(t) = \frac{1}{d!} \sqrt{\det(\tilde{X}^T \tilde{X}) \prod_{i=1}^{d} (\sigma_i^2 + \alpha)} \\
= A_t \sqrt{\prod_{i=1}^{d} (\sigma_i^2 + \alpha)}
\]

(C.4)

where \( A_t \) is the volume of \( \tilde{t} \).

**C.2 Minimizer of residue functions**

We will prove Propositions 11, 12 regarding the minimizer of functions \( R(t) \) and \( R^{iso}(t) \). Recall that \( R(t) \) is the residue of the lifted content \( A_{t,\alpha}(t) \) of a simplex \( t \) after subtracting the signed volume \( A(t) \), and it has the form (Equation 4.5):

\[
R(t) = A_t \left( \sqrt{\prod_{i=1}^{d} (\sigma_i^2 + \alpha)} - \det(L) \right).
\]

(C.5)

Similarly, \( R^{iso}(t) \) is the residue of the isometric lifted content \( A_{t,\alpha}^{iso}(t) \) after subtracting \( A(t) \), and it has the form (Equation 4.8):

\[
R^{iso}(t) = A_t \left( \sqrt{\prod_{i=1}^{d} (\sigma_i^2 + \alpha)} + \frac{\alpha}{2^{d-1}} \prod_{i=1}^{d} (\sigma_i^2 + 1) + \alpha^2 - \det(L) \right).
\]

(C.6)
We first prove that \( R(t) \) is minimized by similarity transformations in 2D and a singular transformation in higher dimensions:

**Proposition 31** (Proposition 11 in chapter 4). For any \( \alpha > 0 \), \( R(t) \geq \alpha^\frac{d}{2} A_t \). Equality holds when either of the following holds:

1. \( d = 2 \), \( \sigma_1 = \sigma_2 \) and \( \det(L) \geq 0 \).
2. \( d > 2 \) and \( \sigma_1 = \ldots = \sigma_d = 0 \).

**Proof.** Using the definition of \( R(t) \) in Equation C.5, and since \( A_t > 0 \), we only need to show that

\[
\sqrt{\prod_{i=1}^{d} (\sigma_i^2 + \alpha)} \geq \det(L) + \alpha^\frac{d}{2} \tag{C.7}
\]

We first consider the case of \( d = 2 \). The lhs of Equation C.7 becomes:

\[
\sqrt{(\sigma_1^2 + \alpha)(\sigma_2^2 + \alpha)} = \sqrt{(\sigma_1 \sigma_2 + \alpha)^2 + \alpha(\sigma_1^2 + \sigma_2^2 - 2\sigma_1 \sigma_2)} \\
\geq \sqrt{(\sigma_1 \sigma_2 + \alpha)^2} \\
= \sigma_1 \sigma_2 + \alpha \\
= |\det(L)| + \alpha \\
\geq \det(L) + \alpha
\]

The first inequality becomes equality when \( \sigma_1 = \sigma_2 \), and the second inequality becomes equality when \( \det(L) \geq 0 \).

Now consider \( d \geq 3 \), which we split into two cases.

1. Suppose \( \prod_{i=1}^{d} \sigma_i = 0 \) (and hence \( \det(L) = 0 \)). It follows that:

\[
\sqrt{\prod_{i=1}^{d} (\sigma_i^2 + \alpha)} \geq \sqrt{\alpha^d} = \det(L) + \alpha^\frac{d}{2}
\]

The inequality becomes equality only when \( \sigma_1 = \ldots = \sigma_d = 0 \).
2. Now suppose $\Pi_{i=1}^d \sigma_i > 0$. We can drive from the lhs of Equation C.7 that:

$$\sqrt{\Pi_{i=1}^d (\sigma_i^2 + \alpha)} \geq \left(\Pi_{i=1}^d \sigma_i^2 + \alpha^d + \alpha^{d-1} \sum_{i=1}^d \sigma_i^2 + \alpha \sum_{i=1}^d \frac{\Pi_{j=1}^d \sigma_j^2}{\sigma_i^2}\right)^\frac{1}{2}$$

$$= \left(\Pi_{i=1}^d \sigma_i^2 + \frac{\alpha^d}{2}\right)^2 - 2\alpha^d \Pi_{i=1}^d \sigma_i + \alpha^d \left(\Pi_{i=1}^d \sigma_i\right) \left(\sum_{i=1}^d \left(\frac{\alpha^{d-1}\sigma_i^2}{\Pi_{j=1}^d \sigma_j} + \frac{\Pi_{j=1}^d \sigma_j}{\alpha^{d-1}\sigma_i^2}\right) - 2\right)$$

$$\geq \left(\Pi_{i=1}^d \sigma_i^2 + \frac{\alpha^d}{2}\right)^2 + \alpha^d \left(\Pi_{i=1}^d \sigma_i\right) (2d - 2)$$

$$> \left(\Pi_{i=1}^d \sigma_i^2 + \frac{\alpha^d}{2}\right)^2$$

$$= |\text{det}(L)| + \alpha^d$$

$$\geq \text{det}(L) + \alpha^d$$

The second inequality is due to the fact that $x + \frac{1}{x} \geq 2$ for any positive $x$.

Combining the two cases, we have prove Equation C.7 for $d \geq 3$. In particular, the equality holds only in the first case, that is, when $\sigma_1 = \ldots = \sigma_d = 0$. \qed

We next prove that $R_{iso}(t)$ is minimized by an isometry in any dimensions:

**Proposition 32** (Proposition 12 in chapter 4). For any $\alpha > 0$, $R_{iso}(t) \geq \alpha A_{\tilde{t}}$, and equality holds only when $\sigma_1 = \ldots = \sigma_d = 1$ and $\text{det}(L) > 0$.

**Proof.** Using the definition of $R_{iso}(t)$ in Equation C.6, and since $A_{\tilde{t}} > 0$, we need to show that

$$\sqrt{\Pi_{i=1}^d \sigma_i^2 + \frac{\alpha}{2d-1} \Pi_{i=1}^d (\sigma_i^2 + 1) + \alpha^2} \geq \text{det}(L) + \alpha$$

(C.8)

We consider two cases for any $d \geq 2$:

1. Suppose $\Pi_{i=1}^d \sigma_i = 0$ (and hence $\text{det}(L) = 0$). The lhs of Equation C.8 becomes:

$$\sqrt{\frac{\alpha}{2d-1} \Pi_{i=1}^d (\sigma_i^2 + 1) + \alpha^2} > \alpha = \text{det}(L) + \alpha$$

[131]
Note that the inequality is strict, because the first term under the square root is always positive.

2. Now suppose \( \Pi_{i=1}^d \sigma_i > 0 \). We have the following derivation from the lhs of Equation C.8:

\[
\sqrt{\Pi_{i=1}^d \sigma_i^2 + \frac{\alpha}{2d-1} \Pi_{i=1}^d (\sigma_i^2 + 1)} + \alpha^2 = \sqrt{(\Pi_{i=1}^d \sigma_i + \alpha)^2 - 2\alpha \Pi_{i=1}^d \sigma_i + \frac{\alpha}{2d-1} \Pi_{i=1}^d (\sigma_i^2 + 1)}
\]

\[
= \sqrt{(\Pi_{i=1}^d \sigma_i + \alpha)^2 + 2\alpha \left( \Pi_{i=1}^d \sigma_i \right) \left( \frac{1}{2d} \Pi_{i=1}^d \left( \sigma_i + \frac{1}{\sigma_i} \right) - 1 \right)}
\]

\[
\geq \sqrt{(\Pi_{i=1}^d \sigma_i + \alpha)^2}
\]

\[
= \Pi_{i=1}^d \sigma_i + \alpha
\]

\[
= |\det(L)| + \alpha
\]

\[
\geq \det(L) + \alpha
\]

The second inequality is again due to the fact that \( x + \frac{1}{x} \geq 2 \) for any positive \( x \), and it becomes equality when \( \sigma_1 = \ldots = \sigma_d = 1 \), which turns the last inequality into equality as well.

We conclude from both cases that Equation C.8 holds for all \( d \geq 2 \), and it becomes equality only when \( \sigma_1 = \ldots = \sigma_d = 1 \).

\[\square\]

### C.3 Injectivity of IsoTLC minimizer

We will prove Proposition 13 in the chapter 4 on the injectivity of the energy minimizer of IsoTLC. Recall that the *isometric lifted content* of a simplex \( t \), given the auxiliary simplex \( \tilde{t} \) and scalar \( \alpha \), has the following form,

\[
A_{t,\alpha}^{iso}(t) = \sqrt{A(t)^2 + \frac{\alpha}{2d-1} A_{\tilde{t},1}(\tilde{t})^2 + \alpha^2 A_{t,1}^2},
\]

(C.9)
where $A_{t,1}(t)$ is the lifted content of $t$ at scale 1. The Isometric Total Lifted Content (IsoTLC) for a mesh $T$, given auxiliary elements $\tilde{T}$ and scalar $\alpha$, is the sum,

$$A_{T,\alpha}^{iso}(T) = \sum_{t \in T} A_{t,\alpha}^{iso}(t).$$

We start with a lemma:

**Lemma 33.** The following holds for all $\alpha \geq 0$:

1. $A_{t,\alpha}^{iso}(t) \geq |A(t)|$, and equality holds when $\alpha = 0$.
2. The derivative $\frac{\partial A_{t,\alpha}^{iso}(t)}{\partial \alpha}$ is finite and positive if either $A(t) \neq 0$ or $\alpha \neq 0$.

**Proof.** Statement (1) is straightforward from Equation C.9 by noting that the second and third terms under the square root are non-negative and zero only when $\alpha = 0$. For (2), substituting Equation C.9 into the derivative gives:

$$\frac{\partial A_{t,\alpha}^{iso}(t)}{\partial \alpha} = \frac{A_{t,1}(t)^2}{2^{d-1} \alpha} + 2\alpha A_t^2}{2\sqrt{A(t)^2 + \frac{\alpha}{2^{d-1}} A_{t,1}(t)^2 + \alpha^2 A_t^2}}$$

Assuming that either $A(t) \neq 0$ or $\alpha \neq 0$, the denominator of the rhs is non-zero, and hence the derivative is well-defined. Also, since the lifted content $A_{t,1}(t)$ is always positive, the numerator is strictly positive, and so is the derivative. \hfill \Box

We next prove a lemma that plays the same role as Lemma B.3 (2D) or B.7 (3D) in [14] in their proof of the injectivity of TLC minimizer. Unlike TLC, where the statement requires different proofs in 2D and 3D and the validity of the statement is unknown in higher dimensions, our statement on IsoTLC applies to any dimension $d \geq 2$.

**Lemma 34.** For any $\delta > 0$, there exists some $\epsilon > 0$ and $\beta > 0$ such that if a map $T$ contains an element whose unsigned volume is smaller than $\epsilon$, then for any positive $\alpha < \beta$, $\frac{\partial A_{T,\alpha}^{iso}(T)}{\partial \alpha} > \delta$. 

[133]
Proof. Consider a single element \( t \in T \). Using the lifted content formula of Equation C.4, we have

\[
A_{\overline{t},1}(t) = A_{\overline{t}} \sqrt{\prod_{i=1}^{d} (\sigma_{i}^2 + 1)} \geq A_{\overline{t}}
\]

Using this inequality, we can derive from the derivative formula of Equation C.11 for any \( \alpha > 0 \):

\[
\frac{\partial A_{\overline{t},\alpha}^{iso}(t)}{\partial \alpha} > \frac{A_{\overline{t},1}^{2}(t) - 2^{d} \sqrt{A(t)^2 + \frac{\alpha}{2^{d-1}} A_{\overline{t},1}(t)^2 + \alpha^2 A_{\overline{t}}^{2}}}{\frac{A(t)^2}{A_{\overline{t}}^{2}} + \frac{\alpha}{2^{d-1}} A_{\overline{t},1}(t)^2 + \alpha^2 A_{\overline{t}}^{2}}
\]

\[
= \frac{A_{\overline{t}}^{2}}{2^{d} \sqrt{A(t)^2 + A_{\overline{t}}^{2}(\frac{\alpha}{2^{d-1}} + \alpha^2)}}
\]

For any \( \delta > 0 \), we can find \( \epsilon > 0, \beta > 0 \) that satisfies:

\[
\epsilon^2 + A_{\overline{t}}^{2}(\frac{\beta}{2^{d-1}} + \beta^2) = \frac{A_{\overline{t}}^{4}}{2^{2d}\delta^2}
\]

If \(|A(t)| < \epsilon\) and \( \alpha \in (0, \beta) \), we conclude that:

\[
\frac{\partial A_{\overline{t},\alpha}^{iso}(t)}{\partial \alpha} > \frac{A_{\overline{t}}^{2}}{2^{d} \sqrt{A(t)^2 + A_{\overline{t}}^{2}(\frac{\alpha}{2^{d-1}} + \alpha^2)}} > \frac{A_{\overline{t}}^{2}}{2^{d} \sqrt{\epsilon^2 + A_{\overline{t}}^{2}(\frac{\beta}{2^{d-1}} + \beta^2)}} = \delta
\]

Finally, let \( t_0 \) be an element of \( T \) such that \(|A(t_0)| < \epsilon\). Combining the inequality above and Lemma 33 (2) yields the desired inequality for all \( \alpha \in (0, \beta) \),

\[
\frac{\partial A_{\overline{t},\alpha}^{iso}(T)}{\partial \alpha} = \sum_{t \in T} \frac{\partial A_{\overline{t},\alpha}^{iso}(t)}{\partial \alpha} > \frac{\partial A_{\overline{t},\alpha}^{iso}(t_0)}{\partial \alpha} > \delta
\]

\( \square \)

Finally, we prove the main result for any \( d \geq 2 \):

[134]
Proposition 35 (Proposition 13 in chapter 4). Let $T_0$ be an injective map with a fully constrained boundary and possible interior constraints. Then there exists some $\alpha_0 > 0$ such that $A_{T,\alpha}^{iso}(T) > A_{T,\alpha}^{iso}(T_0)$ for any $\alpha < \alpha_0$ and any non-injective map $T$ satisfying the same constraints.

Proof. The proof closely follows that of Proposition 4.3 in [14]. Since $T_0$ is injective, so is the boundary map $\partial T_0$ from $\partial M$. By [46], a map $T$ whose boundary is the same as $\partial T_0$ is injective if and only if $T$ has no degenerate or inverted elements. In the following, we assume that an arbitrary but fixed set of auxiliary elements $\mathcal{T}$ is used. For notational convenience, we shall drop the subscript $\mathcal{T}$ in $A_{T,\alpha}^{iso}$.

By Lemma 33 (2), and since $T_0$ is injective, the derivative $\partial A_{\alpha}^{iso}(T_0)/\partial \alpha$ is bounded. We shall pick an arbitrary but small positive value $\tau$, and define $\delta$ as the maximum derivative for all $\alpha < \tau$.

Now suppose $T$ has an element whose unsigned volume is smaller than $\epsilon$, which is found by Lemma 34 for $\delta$. By that lemma, there exists some $\beta > 0$ such that for any $\alpha < \beta$, $\partial A_{\alpha}^{iso}(T)/\partial \alpha > \delta$. As a result, the following holds for all $\alpha < \min(\tau, \beta)$:

$$\frac{\partial (A_{\alpha}^{iso}(T) - A_{\alpha}^{iso}(T_0))}{\partial \alpha} = \frac{\partial A_{\alpha}^{iso}(T)}{\partial \alpha} - \frac{\partial A_{\alpha}^{iso}(T_0)}{\partial \alpha} > \delta - \delta = 0$$

Furthermore, by Lemma 33 (a), $A_0^{iso}(T)$ is the total unsigned volume of $T$, which is no smaller than the total unsigned volume of $T_0$, or $A_0^{iso}(T_0)$. Thus we conclude $A_{\alpha}^{iso}(T) > A_{\alpha}^{iso}(T_0)$ for any $\alpha \in (0, \min(\tau, \beta))$.

Otherwise, suppose $T$ has no element whose unsigned volume is smaller than $\epsilon$. Since $T$ is non-injective, it must contain no degenerate element and at least one inverted element whose unsigned volume is at least $\epsilon$. Due to Lemma 33 (1), for any $\alpha > 0$,

$$A_{\alpha}^{iso}(T) > \sum_{t \in T} |A(t)| \geq \sum_{t \in T} A(t) + 2\epsilon = A_0^{iso}(T_0) + 2\epsilon$$

Since the derivative $\partial A_{\alpha}^{iso}(T_0)/\partial \alpha$ is bounded (Lemma 34), there exists some $\kappa > 0$ such that, for all $\alpha < \kappa$,

$$A_{\alpha}^{iso}(T_0) < A_0^{iso}(T_0) + 2\epsilon < A_{\alpha}^{iso}(T)$$

The proof is completed by letting $\alpha_0 = \min(\tau, \beta, \kappa)$. \qed
C.4 Injectivity of IsoSEA minimizer

Lastly, we will prove Proposition 14 in the chapter 4 on the injectivity of the energy minimizer of IsoSEA. Recall that the Isometric Smooth Excess Area (IsoSEA) of a triangular mesh $T$, given the auxiliary simplices $\tilde{T}$ and scalars $\alpha, \theta$, has the following form:

$$A_{\tilde{T},\alpha,\theta}^{iso}(T) = A_{\tilde{T},\alpha}^{iso}(T) - O(\partial T).$$

(C.12)

where $A_{\tilde{T},\alpha}^{iso}(T)$ is the IsoTLC of $T$ and $O(\partial T)$ is the arc-occupancy of the boundary $\partial T$ defined as follows. Recall that the occupancy $O(C)$ of a closed curve $C$ is the area of the plane with a positive winding number w.r.t. $C$. Let $C_{\theta}$ be a curve constructed by replacing each edge of $\partial T$ by an arc with center angle $\theta$. The arc-occupancy $O_{\theta}(\partial T)$ is defined as the occupancy of $C_{\theta}$ subtracted by the total area of the regions each bounded by an edge of $\partial T$ and its arc in $C_{\theta}$.

We first recall a few useful properties of the excess area of $T$ defined as the difference:

$$A^{excess}(T) = \sum_{t \in T} |A(t)| - O(\partial T)$$

Furthermore, let $A^{overlap}(T)$ be the total overlapping area defined as the difference between the total unsigned area of triangles in $T$ and the area of the plane covered by $T$, and let $A^{invert}(T)$ be the total area of inverted triangles in $T$. The following result from [15] shows that the excess area is a good proxy of both the overlapping and inverted area:

**Lemma 36** (Proposition 5.1 in [15]). For any map $T$,

1. $A^{excess}(T) \geq A^{overlap}(T)$
2. $A^{excess}(T) \geq A^{invert}(T)$
3. $A^{excess}(T) \leq A^{overlap}(T) + A^{invert}(T)$

Using these results, we prove a property of IsoSEA similar to that of SEA in Proposition 5.2 of [15]:

**Lemma 37.** For any map $T$, $\alpha \geq 0$ and $\theta > 0$, $A_{\tilde{T},\alpha,\theta}^{iso}(T) \geq A^{excess}(T)$. Furthermore, if $T$ is injective, there exists some $\theta_0 > 0$ such that $A_{\tilde{T},0,\theta}^{iso}(T) = 0$ for all $\theta < \theta_0$.  

[136]
Proof. The proof of Proposition 5.2 in [15] utilizes Lemma 36 and the fact that TLC is an upper bound of the total unsigned area, that is, \( A_{T,\alpha}(T) \geq \sum_{t \in T} |A(t)| \). Since the same property holds for IsoTLC (Lemma 33 (1)), the rest of the proof follows. \( \square \)

Next, we introduce a variant of Lemma 34 that concerns maps containing triangles with large angle distortions. Let \( D_t(t) \) be the Dirichlet energy of the transformation from a triangle \( t \) back to its auxiliary triangle \( \tilde{t} \), which has the form [57] (utilizing the fact that the singular values of this inverse transform are reciprocals of \( \sigma_1, \sigma_2 \), the singular values of the transform from \( \tilde{t} \) to \( t \)):

\[
D_t(t) = \frac{|A(t)|}{2} \left( \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \right) = \frac{A_t}{2} \left( \frac{\sigma_1^2 + \sigma_2^2}{\sigma_1 \sigma_2} \right)
\]

(C.13)

The second equality comes from \( |A(t)| = A_t \sigma_1 \sigma_2 \). Note that the energy is minimal when \( \sigma_1 = \sigma_2 \) (no angle distortion).

**Lemma 38.** For any \( \delta > 0 \), there exists some \( \eta > 0 \) and \( \gamma > 0 \) such that if a map \( T \) has no degenerate triangle but contains a triangle \( t \) such that \( D_t(t) > \eta \), then for any positive \( \alpha < \gamma \), \( \partial A^{iso}_{T,\alpha}(T)/\partial \alpha > \delta \).

**Proof.** Consider a single element \( t \in T \) \( (A(t) \neq 0 \) by assumption). We first derive the following inequality using equality \( |A(t)| = A_t \sigma_1 \sigma_2 \) and Equations C.4 and C.13,

\[
\frac{A^2_{t,1}(t)}{|A(t)|} = \frac{A_t^2(\sigma_1^2 + 1)(\sigma_2^2 + 1)}{A_t \sigma_1 \sigma_2} > \frac{A_t(\sigma_1^2 + \sigma_2^2)}{\sigma_1 \sigma_2} = 2D_t(t).
\]

Recall from the proof of Lemma 34 that, for any \( \alpha > 0 \):

\[
\frac{\partial A^{iso}_{T,\alpha}(t)}{\partial \alpha} > \frac{1}{2^d \sqrt{A^2_{t,1}(t) + \frac{\alpha}{2^{d-1}A_{t,1}(t)} + \frac{\alpha^2 A^2_{t,1}(t)}}}
\]

[137]
Continuing the derivation using the inequalities $\frac{A_{t,1}(t)}{|A(t)|} > 2D(t)$ and $A_{t,1}(t) > A_{t}$ yields

$$\frac{\partial A_{t,\alpha}^{iso}(t)}{\partial \alpha} > \frac{1}{2d\sqrt{\frac{1}{4D_{t}(t)} + \frac{\alpha}{2^{d-1}A_{t}^{2}}} + \frac{\alpha^{2}A_{t}^{2}}{A_{t}^{2}}} = \frac{A_{t}}{2d\sqrt{4D_{t}^{2}(t)} + \frac{\alpha}{2^{d-1}A_{t}^{2}}} + \alpha^{2}$$

For any $\delta > 0$, we can find $\eta > 0, \gamma > 0$ that satisfies:

$$\frac{A_{t}^{2}}{4\eta^{2}} + \frac{\gamma}{2^{d-1}} + \gamma^{2} = \frac{A_{t}^{2}}{2^{2d}\delta^{2}}$$

If $D_{t}(t) > \eta$ and $\alpha \in (0, \gamma)$, we conclude that:

$$\frac{\partial A_{t,\alpha}^{iso}(t)}{\partial \alpha} > \frac{A_{t}}{2d\sqrt{\frac{A_{t}^{2}}{4D_{t}^{2}(t)} + \frac{\alpha}{2^{d-1}A_{t}^{2}}} + \alpha^{2}} > \frac{A_{t}}{2d\sqrt{\frac{A_{t}^{2}}{4\eta^{2}} + \frac{\gamma}{2^{d-1}A_{t}^{2}}} + \gamma^{2}} = \delta$$

Finally, let $t_{0}$ be an element of $T$ such that $D_{t}(t_{0}) > \eta$. Combining the inequality above and Lemma 33 (2) yields the desired inequality for all $\alpha \in (0, \gamma)$,

$$\frac{\partial A_{t,\alpha}^{iso}(T)}{\partial \alpha} = \sum_{t \in T} \frac{\partial A_{t,\alpha}^{iso}(t)}{\partial \alpha} > \frac{\partial A_{t_{0},\alpha}^{iso}(t_{0})}{\partial \alpha} > \delta$$

We will also need the following result from [15]. It gives a lower bound of the overlapping area around an overwound vertex as a function of the minimum triangle area and maximum per-triangle Dirichlet energy.

**Lemma 39** (Lemma D.4 in [15]). *If map $T$ contains an interior vertex $v$ such that $v$ is incident to only triangles with positive areas and the sum of angles around $v$ is not $2\pi$, then $A_{\text{overlap}}(T) \geq \epsilon h^{2}\pi/2\eta$, where $\epsilon$ is the minimum unsigned area of any triangle in $T$, $h$ is the shortest height of any auxiliary triangle in $T$, and $\eta$ is the maximum $D_{t}(t)$ in any $t \in T$.*

We are ready to prove the main result:

[138]
Proposition 40 (Proposition 14 in chapter 4). Let $T_0$ be an injective, triangular map satisfying the given constraints. For any $\lambda > 0$, there exists some $\alpha_0 > 0$ and $\theta_0 > 0$ such that, for any $\alpha < \alpha_0, \theta < \theta_0, A_{T,\alpha,\theta}^{iso}(T) > A_{T,\alpha,\theta}^{iso}(T_0)$ for any map $T$ that is not locally injective or whose overlapping area is greater than $\lambda$.

Proof. The proof closely follows that of Proposition 5.3 in [15]. In the following, we assume that an arbitrary but fixed set of auxiliary triangles $\tilde{T}$ is used. Also, since $T_0$ is injective, by Lemma 37, there exists some $\theta_0 > 0$ such that $A_{T,\theta}^{iso}(T_0) = 0$ for all $\theta < \theta_0$. In the following, we consider some fixed $\theta$ in this range. For notational convenience, we shall shorthand $A_{T,\alpha,\theta}^{iso}$ as $A_{T}^{iso}$ (not to be confused with IsoTLC).

Since arc-occupancy does not depend on $\alpha$, IsoSEA shares the same partial derivative as IsoTLC with respect to $\alpha$, and previous results regarding the derivative of IsoTLC (e.g., Lemma 33 (2)) applies to IsoSEA too. In particular, since $T_0$ is injective, the derivative $\partial A_{T}^{iso}(T)/\partial \alpha$ is always bounded. We pick an arbitrary but small positive value $\tau$, and let $\delta$ be the maximum value of the derivative for all $\alpha < \tau$.

Now consider a map $T$ that is not locally injective, or that $A_{overlap}(T) > \lambda$. We separately consider the cases that (1) $T$ has some small-area triangle (including degenerate triangles), (2) $T$ has some triangle with large Dirichlet energy, and (3) $T$ has neither.

1. Suppose $T$ has a triangle whose unsigned area is smaller than $\epsilon$, which is found by Lemma 34 for $\delta$. This case is already considered in the proof of Proposition 35, which shows that there exists some $\beta > 0$ such that for any $\alpha < \min(\beta, \tau), \partial A_{T}^{iso}(T)/\partial \alpha > \partial A_{T}^{iso}(T_0)/\partial \alpha$. On the other hand, by Lemma 37, $A_{T}^{iso}(T) \geq 0 = A_{T}^{iso}(T_0)$. Hence $A_{T}^{iso}(T) > A_{T}^{iso}(T_0)$ for any $\alpha < \min(\beta, \tau)$.

2. Suppose all triangles in $T$ have unsigned areas no smaller than $\epsilon$, but at least one triangle has a Dirichlet energy greater than $\eta$, which is found by Lemma 38 for $\delta$. By the lemma, and similar to the case above, there exists some $\gamma > 0$ such that $A_{T}^{iso}(T) > A_{T}^{iso}(T_0)$ for any $\alpha < \min(\gamma, \tau)$.

3. Suppose the triangles of $T$ have neither small unsigned areas nor large Dirichlet energy.

We further split this case into three sub-cases:

- $T$ has some inverted triangle. Since each inverted triangle must have an unsigned area no smaller than $\epsilon$, by Lemmas 36 and 37, $A_{T}^{iso}(T) \geq A_{T}^{excess}(T) \geq A_{T}^{invert}(T) \geq \epsilon$.

[139]
ε for any α ≥ 0. Since $A_{0}^{iso}(T_0) = 0$ and the partial derivative $\partial A_{\alpha}^{iso}(T_0)/\partial \alpha$ is bounded, we conclude that there exists some $\kappa_1 > 0$ such that, for all $\alpha < \kappa_1$, $A_{\alpha}^{iso}(T_0) < \epsilon \leq A_{\alpha}^{iso}(T)$.

- $T$ has no inverted (or degenerate) triangle, but some vertex $v$ has an angle sum other than $2\pi$. By Lemma 39, $A^{\text{overlap}}(T) \geq \sigma = \epsilon h^2 \pi/4\eta$, where $h$ is the smallest height among all auxiliary triangles, and $\epsilon, \eta$ are constants found in cases (1,2) above. Note that $\sigma$, like $\epsilon$ and $\eta$, is independent of the map $T$. Following Lemmas 36 and 37, $A_{\alpha}^{iso}(T) \geq A^{\text{excess}}(T) \geq A^{\text{overlap}}(T) \geq \sigma$ for any $\alpha \geq 0$. Similar to the previous sub-case, there exists some $\kappa_2 > 0$ such that, for all $\alpha < \kappa_2$, $A_{\alpha}^{iso}(T_0) < \sigma \leq A_{\alpha}^{iso}(T)$.

- $T$ is locally injective but $A^{\text{overlap}}(T) > \lambda$. Similar to the previous sub-case, there exists some $\kappa_3 > 0$ such that, for all $\alpha < \kappa_3$, $A_{\alpha}^{iso}(T_0) < \lambda < A_{\alpha}^{iso}(T)$.

The proof is completed by setting $\alpha_0 = \min(\tau, \beta, \gamma, \kappa_1, \kappa_2, \kappa_3)$. ⊡

C.5 Implementation details

Minimizing IsoTLC and IsoSEA using gradient-based solvers requires the evaluation of the energies, their gradients (for quasi-Newton), and Hessians (for projected Newton). We provide more implementation details in this section.

C.5.1 IsoTLC

Evaluating IsoTLC amounts to summing up the per-element isometric lifted content, defined in Equation C.9, which in turn can be directly obtained from the area (or volume) of the $d$-dimensional element $t$, the auxiliary element $\tilde{t}$, and the lifted element. In particular, as derived in [14] (Section 4.2.1), the area (or volume) of the lifted element at scale $\alpha$ has the form:

$$A_{\alpha}(t) = \frac{1}{d!} \sqrt{\text{Det}(X^T X + \alpha \tilde{X}^T \tilde{X})}$$

where $X$ (respectively $\tilde{X}$) is the $d \times d$ matrix whose column vectors are the edge vectors from one vertex of the simplex $t$ (respectively $\tilde{t}$) to the other $d$ vertices of the simplex.

[140]
To derive the gradient and Hessian of IsoTLC, we resort to the general technique proposed by Smith et al. [67] for distortion energies in both 2D or 3D. We shall give details on the application of this technique to IsoTLC, and we refer the reader to the paper [67] and a comprehensive course note [37] for in-depth discussions of the general technique.

Smith’s technique assumes distortion energies of the general form

$$\Psi(T) = \sum_{i \in T} A_i \Psi_i(L)$$

Here, $A_i \Psi_i(L)$ is the per-simplex distortion $\Psi_i(L)$ weighted by the area (or volume) of the rest (auxiliary) simplex $\tilde{t}$. The distortion is defined using $L$, which is the linear transformation that maps the edge vectors $\tilde{X}$ of $\tilde{t}$ to the edge vectors $X$ of $t$. $L$ is often called the deformation gradient. Smith’s technique applies to any distortion $\Psi_i(t)$ that can be written in terms of three invariants:

$$I_1 = \sum_i \sigma_i, \quad I_2 = \sum_i \sigma_i^2, \quad I_3 = \prod_i \sigma_i$$ (C.14)

where $\sigma_i$ are the singular values of $L$.

The per-simplex term in IsoTLC is the isometric lifted content, $A^{iso}_{L,\alpha}(t)$. From its singular-value form, we obtain

$$A^{iso}_{L,\alpha}(t) = A_{L,\alpha}(L),$$ (C.15)

where

$$\Psi_{L,\alpha}(L) = \sqrt{\Pi_{i=1}^d \sigma_i^2 + \frac{\alpha}{2^{d-1}} \Pi_{i=1}^d (\sigma_i^2 + 1) + \alpha^2}. \quad (C.16)$$

We can rewrite Equation C.16 in terms of the invariants in Equations C.14 for a triangle ($d = 2$),

$$\Psi_{L,\alpha}(L) = \sqrt{(1 + \frac{\alpha}{2})I_3^2 + \frac{\alpha}{2} I_2 + \frac{\alpha}{2} + \alpha^2}, \quad (C.17)$$

and for a tetrahedron ($d = 3$),

$$\Psi_{L,\alpha}(L) = \sqrt{(1 + \frac{\alpha}{4})I_3^2 + \frac{\alpha}{4} I_2 + \frac{\alpha}{8} (I_2 - II_C) + \frac{\alpha}{4} + \alpha^2}, \quad (C.18)$$

where $II_C = ||L^TL||^2$ can also be expressed in terms of the invariants (see Appendix A of [67]),

$$II_C = \frac{1}{2} I_2^2 - \frac{1}{2} I_1^4 + I_1^2I_2 + 4I_1I_3.$$ [141]
To evaluate the gradient of the isometric lifted content, $A_{t,\alpha}^{iso}(t)$, we flatten $t$’s vertex coordinates into a vector $\mathbf{x}$ and apply the chain rule,

$$\frac{\partial A_{t,\alpha}^{iso}}{\partial \mathbf{x}} = A_t \frac{\partial \Psi_{t,\alpha} \partial L}{\partial \mathbf{x}} = A_t \left( \sum_{i=1}^{3} \frac{\partial \Psi_{t,\alpha}}{\partial I_i} \frac{\partial I_i}{\partial L} \right) \frac{\partial L}{\partial \mathbf{x}}. \quad (C.19)$$

Since $\Psi_{t,\alpha}$ is a scalar function of the three invariants, the explicit formula for its partial derivatives $\frac{\partial \Psi_{t,\alpha}}{\partial I_i}$ can be obtained using calculus. The formulas for $\frac{\partial L}{\partial L}$ are provided in [67] and Appendix B of [37]. Finally, Appendix E of [37] describes how to compute $\frac{\partial L}{\partial \mathbf{x}}$.

Optimizing IsoTLC using projected Newton requires a positive semi-definite (PSD) projection of the Hessian. One approach is first evaluating the Hessian matrix of the isometric lifted content $A_{t,\alpha}^{iso}(t)$ for each simplex, then numerically projecting it to be positive semi-definite, and finally assembling the full Hessian by accumulating per-simplex Hessians. Smith et al. [67] introduced a general routine to obtain per-simplex PSD-projected Hessian without first evaluating the original Hessian matrix, provided that the distortion energy can be written in terms of the three invariants. The routine is described in details in [67] and chapter 7 of [37]. In practice, we found that using the projected Hessian of the residual $R^{iso}(t)$ instead of $A_{t,\alpha}^{iso}(t)$ leads to a higher success rate and lower distortions on the benchmark. Recall that $R^{iso}(t)$ is the difference between the isometric lifted content $A_{t,\alpha}^{iso}(t)$ and the signed volume $A(t)$. The residual can also be written in terms of the three invariants in 2D,

$$R^{iso}(t) = A_t \left( \sqrt{(1 + \frac{\alpha}{2})I_3^2 + \frac{\alpha}{2}I_2 + \frac{\alpha^2}{2} - I_3} \right), \quad (C.20)$$

and in 3D,

$$R^{iso}(t) = A_t \left( \sqrt{(1 + \frac{\alpha}{4})I_3^2 + \frac{\alpha}{4}I_2 + \frac{\alpha}{8}(I_2^2 - II_C) + \frac{\alpha^2}{4} + \alpha^2 - I_3} \right). \quad (C.21)$$

We provide the pseudo-code to evaluate projected Hessians of the residual $R^{iso}(t)$ in 2D (Algorithm 1) and 3D (Algorithm 2) using the technique in [67].

[142]
Algorithm 1 Evaluate projected Hessian of the residual $R^{iso}(t)$ in 2D

Require: $\alpha$, coordinates $\bar{x}$ of $\bar{t}$, coordinates $x$ of $t$
1: $L \leftarrow$ deformationGradient($\bar{x}, x$) \hspace{1cm} $\triangleright$ See Appendix D of [37] for computing $L$
2: $U, \Sigma, V \leftarrow$ SVD($L$) \hspace{1cm} $\triangleright$ Rotation-variant SVD, see Appendix F of [37]
3: $\sigma_1, \sigma_2 \leftarrow$ diagonal($\Sigma$)
4: $I_1 \leftarrow \sigma_1 + \sigma_2$
5: $I_2 \leftarrow \sigma_1^2 + \sigma_2^2$
6: $I_3 \leftarrow \sigma_1 \sigma_2$
7: $\psi \leftarrow \sqrt{(1 + \frac{\alpha}{2})I_3^2 + \frac{\alpha}{2}I_2 + \frac{\alpha}{2} + \alpha^2}$ \hspace{1cm} $\triangleright$ $A_2 \psi$ is the isometric lifted content
8: $\lambda_{\text{twist}} \leftarrow \left(\frac{\alpha}{2} + (1 + \frac{\alpha}{2})I_3\right)/\psi - 1$
9: $v_{\text{twist}} \leftarrow \text{vec}(\frac{1}{\sqrt{2}}U\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}V^\top)$ \hspace{1cm} $\triangleright$ vec(.) vectorizes a matrix. See section 3 of [67]
10: $\lambda_{\text{flip}} \leftarrow \left(\frac{\alpha}{2} - (1 + \frac{\alpha}{2})I_3\right)/\psi + 1$
11: $v_{\text{flip}} \leftarrow \text{vec}(\frac{1}{\sqrt{2}}U\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}V^\top)$
12: $d_1 \leftarrow \text{vec}(U\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}V^\top)$
13: $d_2 \leftarrow \text{vec}(U\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}V^\top)$
14: $a_{11} \leftarrow \left(\frac{\alpha}{2} + \alpha^2 + \frac{\alpha}{2} \sigma_2^2\right)(\frac{\alpha}{2} + (1 + \frac{\alpha}{2}) \sigma_2^2)/\psi^3$
15: $a_{22} \leftarrow \left(\frac{\alpha}{2} + \alpha^2 + \frac{\alpha}{2} \sigma_1^2\right)(\frac{\alpha}{2} + (1 + \frac{\alpha}{2}) \sigma_1^2)/\psi^3$
16: $a_{12} \leftarrow \left(\alpha + \frac{\alpha}{2} \sigma_2^3 + \alpha^3 + (1 + \frac{\alpha}{2})\left(\frac{\alpha}{2} I_2 + (1 + \frac{\alpha}{2})I_3^2\right)\right)/\psi^3 - 1$
17: if $a_{12} = 0$ then
18: \hspace{1cm} $\lambda_{\text{scale}1} \leftarrow a_{11}$
19: \hspace{1cm} $\lambda_{\text{scale}2} \leftarrow a_{22}$
20: \hspace{1cm} $v_{\text{scale}1} \leftarrow d_1$
21: \hspace{1cm} $v_{\text{scale}2} \leftarrow d_2$
22: else
23: \hspace{1cm} $A \leftarrow \begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{bmatrix}$
24: \hspace{1cm} $\lambda_{\text{scale}1}, \lambda_{\text{scale}2} \leftarrow$ eigenValues($A$)
25: \hspace{1cm} $\beta \leftarrow (\lambda_{\text{scale}1} - a_{22})/a_{12}$
26: \hspace{1cm} $v_{\text{scale}1} \leftarrow (\beta d_1 + d_2)/\sqrt{1 + \beta^2}$
27: \hspace{1cm} $v_{\text{scale}2} \leftarrow (d_1 - \beta d_2)/\sqrt{1 + \beta^2}$
28: end if
29: $H_L \leftarrow \sum_{i \in \{\text{twist}, \text{flip}, \text{scale1}, \text{scale2}\}} \max(\lambda_i, 0)v_i v_i^\top$
30: $H_x \leftarrow \text{area}(\bar{x})(\frac{\partial L}{\partial x})^\top H_L \frac{\partial L}{\partial x}$ \hspace{1cm} $\triangleright$ See Appendix E of [37] for computing $\frac{\partial L}{\partial x}$
31: return $H_x$
Algorithm 2 Evaluate projected Hessian of the residual $R_{iso}(t)$ in 3D

Require: $\alpha$, coordinates $\tilde{x}$ of $t$, coordinates $x$ of $t$

1: $L \leftarrow \text{deformationGradient}(\tilde{x}, x)$ \Comment{See Appendix D of [37] for computing $L$}
2: $U, \Sigma, V \leftarrow \text{SVD}(L)$ \Comment{Rotation-variant SVD, see Appendix F of [37]}
3: $\sigma_1, \sigma_2, \sigma_3 \leftarrow \text{diagonal}(\Sigma)$
4: $I_1 \leftarrow \sigma_1 + \sigma_2 + \sigma_3$
5: $I_2 \leftarrow \sigma_1^2 + \sigma_2^2 + \sigma_3^2$
6: $I_3 \leftarrow \sigma_1 \sigma_2 \sigma_3$
7: $I^*_C \leftarrow \sigma_1^2 + \sigma_2^2 + \sigma_3^2$
8: $I^*_C \leftarrow \sigma_1^2 \sigma_2^2 + \sigma_1^2 \sigma_3^2 + \sigma_2^2 \sigma_3^2$
9: $\psi \leftarrow \sqrt{(1 + \frac{\alpha}{4})I_2^2 + \frac{\alpha}{4}I_2 + \frac{\alpha}{2}(I_2^2 - I^*_C) + \frac{\alpha}{4} + \alpha^2}$ \Comment{$A_2\psi$ is the isometric lifted content}
10: $c_1 \leftarrow 1 + \frac{\alpha}{4}$
11: $c_2 \leftarrow \frac{\alpha}{4}$
12: $c_3 \leftarrow \frac{\alpha}{8} + \alpha^2$
13: $c_4 \leftarrow \frac{\alpha}{4} + \alpha^2$
14: $\lambda_{twist_1} \leftarrow (c_2 + 2c_3(\sigma_1^2 + \sigma_2 \sigma_3) + c_1 \sigma_1 I_3) / \psi - \sigma_1$
15: $v_{twist_1} \leftarrow \text{vec}(\frac{1}{\sqrt{2}}U 0 0 0 0 -1 0 V^T)$ \Comment{vec(.) vectorizes a matrix. See section 3 of [67]}
16: $\lambda_{twist_2} \leftarrow (c_2 + 2c_3(\sigma_2^2 + \sigma_1 \sigma_3) + c_1 \sigma_2 I_3) / \psi - \sigma_2$
17: $v_{twist_2} \leftarrow \text{vec}(\frac{1}{\sqrt{2}}U 0 0 1 0 -1 0 V^T)$
18: $\lambda_{twist_3} \leftarrow (c_2 + 2c_3(\sigma_3^2 + \sigma_1 \sigma_2) + c_1 \sigma_3 I_3) / \psi - \sigma_3$
19: $v_{twist_3} \leftarrow \text{vec}(\frac{1}{\sqrt{2}}U 0 0 0 0 -1 0 V^T)$
20: $\lambda_{flip_1} \leftarrow (c_2 + 2c_3(\sigma_1^2 - \sigma_2 \sigma_3) - c_1 \sigma_1 I_3) / \psi + \sigma_1$
21: $v_{flip_1} \leftarrow \text{vec}(\frac{1}{\sqrt{2}}U 0 0 1 0 -1 0 V^T)$
22: $\lambda_{flip_2} \leftarrow (c_2 + 2c_3(\sigma_2^2 - \sigma_1 \sigma_3) - c_1 \sigma_2 I_3) / \psi + \sigma_2$
23: $v_{flip_2} \leftarrow \text{vec}(\frac{1}{\sqrt{2}}U 0 0 0 0 1 0 V^T)$
24: $\lambda_{flip_3} \leftarrow (c_2 + 2c_3(\sigma_3^2 - \sigma_1 \sigma_2) - c_1 \sigma_3 I_3) / \psi + \sigma_3$
25: $v_{flip_3} \leftarrow \text{vec}(\frac{1}{\sqrt{2}}U 0 0 0 0 1 0 V^T)$

[144]
26: \(d_1 \leftarrow \text{vec}(U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^\top)\)

27: \(d_2 \leftarrow \text{vec}(U \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^\top)\)

28: \(d_3 \leftarrow \text{vec}(U \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^\top)\)

29: \(a_{11} \leftarrow (c_4 + 2c_3 \sigma_2^2 \sigma_3^2 + c_2 (\sigma_2^2 + \sigma_3^2))(c_2 + c_1 \sigma_2^2 \sigma_3^2 + 2c_3 (\sigma_2^2 + \sigma_3^2))/\psi^3\)

30: \(a_{22} \leftarrow (c_4 + 2c_3 \sigma_2^2 \sigma_3^2 + c_2 (\sigma_2^2 + \sigma_3^2))(c_2 + c_1 \sigma_2^2 \sigma_3^2 + 2c_3 (\sigma_2^2 + \sigma_3^2))/\psi^3\)

31: \(a_{33} \leftarrow (c_4 + 2c_3 \sigma_2^2 \sigma_3^2 + c_2 (\sigma_2^2 + \sigma_3^2))(c_2 + c_1 \sigma_2^2 \sigma_3^2 + 2c_3 (\sigma_2^2 + \sigma_3^2))/\psi^3\)

32: \(a_{12} \leftarrow (c_2 (c_1 I_3 \sigma_3 (I_2 + \sigma_3^2) + 2c_3 \sigma_1 \sigma_2 (I_2 - \sigma_3^2)) - \psi I_c^3 \sigma_3) + 4c_3^2 \sigma_1 \sigma_2 (I_1^c - \sigma_3^4) + 2c_3 (2c_4 \sigma_1 \sigma_2 - \psi I_c^3 \sigma_3 + c_1 I_3 (I_1^c + \sigma_3^4 \sigma_3) + (c_1 I_3^2 (c_1 I_3 - \psi) - c_4 (\psi - 2c_1 I_3) ) \sigma_3 - c_2^3 \sigma_3^4 \sigma_3^3)/\psi^3\)

33: \(a_{13} \leftarrow (c_2 (c_1 I_3 \sigma_3 (I_2 + \sigma_3^2) + 2c_3 \sigma_1 \sigma_2 (I_2 - \sigma_3^2)) - \psi I_c^3 \sigma_3) + 4c_3^2 \sigma_1 \sigma_2 (I_1^c - \sigma_3^4) + 2c_3 (2c_4 \sigma_1 \sigma_3 - \psi I_c^3 \sigma_3 + c_1 I_3 (I_1^c + \sigma_3^4 \sigma_3) + (c_1 I_3^2 (c_1 I_3 - \psi) - c_4 (\psi - 2c_1 I_3) ) \sigma_3 - c_2^3 \sigma_3^4 \sigma_3^3)/\psi^3\)

34: \(a_{23} \leftarrow (c_2 (c_1 I_3 \sigma_1 (I_2 + \sigma_3^2) + 2c_3 \sigma_2 \sigma_3 (I_2 - \sigma_3^2)) - \psi I_c^3 \sigma_1) + 4c_3^2 \sigma_2 \sigma_3 (I_1^c - \sigma_3^4) + 2c_3 (2c_4 \sigma_2 \sigma_3 - \psi I_c^3 \sigma_1 + c_1 I_3 (I_1^c + \sigma_3^4 \sigma_3) + (c_1 I_3^2 (c_1 I_3 - \psi) - c_4 (\psi - 2c_1 I_3) ) \sigma_1 - c_2^3 \sigma_3^4 \sigma_3^3)/\psi^3\)

35: \textbf{if} \(a_{12} = 0\) and \(a_{13} = 0\) and \(a_{23} = 0\) \textbf{then}

36: \(\lambda_{\text{scale}_1} \leftarrow a_{11}\)

37: \(\lambda_{\text{scale}_2} \leftarrow a_{22}\)

38: \(\lambda_{\text{scale}_3} \leftarrow a_{33}\)

39: \(v_{\text{scale}_1} \leftarrow d_1\)

40: \(v_{\text{scale}_2} \leftarrow d_2\)

41: \(v_{\text{scale}_3} \leftarrow d_3\)

42: \textbf{else}

43: \(A \leftarrow \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{bmatrix}\)

44: \(\lambda_{\text{scale}_1}, \lambda_{\text{scale}_2}, \lambda_{\text{scale}_3} \leftarrow \text{eigenValues}(A)\) \(\triangleright\) normalized eigen vectors, i.e., \(\|z_i\| = 1\)

45: \(z_1, z_2, z_3 \leftarrow \text{eigenVectors}(A)\)

46: \(D \leftarrow \begin{bmatrix} d_1 & d_2 & d_3 \end{bmatrix}\) \(\triangleright\) matrix \(D\) with column vectors \(d_1, d_2, d_3\)

47: \(v_{\text{scale}_1} \leftarrow D z_1\)

48: \(v_{\text{scale}_2} \leftarrow D z_2\)

49: \(v_{\text{scale}_3} \leftarrow D z_3\)

50: \textbf{end if}

51: \(H_L \leftarrow \sum_{i \in \{\text{twist}, \text{twist}_3, \text{flip}_1, \text{flip}_2, \text{flip}_3, \text{scale}_1, \text{scale}_2, \text{scale}_3\}} \max(\lambda_i, 0) v_i v_i^\top\)

52: \(H_x \leftarrow \text{volume}(\bar{x}) (\frac{\partial L}{\partial x})^\top H_L \frac{\partial L}{\partial x}\) \(\triangleright\) See Appendix E of [37] for computing \(\frac{\partial L}{\partial x}\)

53: \textbf{return} \(H_x\)

[145]
C.5.2 IsoSEA

The IsoSEA energy is the difference between the IsoTLC energy and the arc-occupancy of the boundary of the mapped domain. This latter term was proposed in [15], which gave a detailed account of its derivation (Section 5.2) and computation (Section 6). Here we give an abbreviated account of arc-occupancy, and we refer readers to [15] for a full discussion.

Consider a triangular mesh $T$ in the plane with an oriented boundary $\partial T$ (which may consist of one or multiple curves). To compute arc-occupancy, we construct, for each edge $e \in \partial T$, a circular arc with center angle $\theta$ and $e$ as its chord. This arc, $\Gamma_{\theta}(e)$, has the same orientation as $e$ and lies on the right side of $e$. We call $\Gamma_{\theta}(e)$ the arc-edge of $e$, and the curve consisting of all arc-edges the arc-boundary of $\partial T$, denoted by $\Gamma_{\theta}(\partial T)$. Additionally, we call the area bounded by each edge $e$ and its arc-edge a “flap” (see Figure 5 left of [15]). The arc-occupancy of $\partial T$ is defined as the occupancy of the arc-boundary minus the sum of all the flap areas, denoted as $B_{\theta}(\partial T)$:

$$O_{\theta}(\partial T) = O(\Gamma_{\theta}(\partial T)) - B_{\theta}(\partial T) \tag{C.22}$$

The second term of Equation C.22, $B_{\theta}(\partial T)$, has a simple expression:

$$B_{\theta}(\partial T) = \sum_{e \in \partial T} \frac{||e||^2(\theta - \sin \theta)}{4(1 - \cos \theta)}.$$

Evaluating the first term of Equation C.22, the occupancy of the arc-boundary $\Gamma_{\theta}(\partial T)$, amounts to computing the arrangement of the arc-boundary and the winding number of each region in the arrangement. The occupancy is the sum of area of all regions with a positive winding number. Please refer to [15] (Section 6) for details of computing such arrangement, the winding numbers, and the region areas. Finally, since both terms of Equation C.22 can be expressed as functions of the locations of the vertices of $\partial T$, the gradient of $O_{\theta}(\partial T)$ can be derived using the chain rule.