Optimizing Global Injectivity for Constrained Parameterization

XINGYI DU, Washington University in St. Louis, USA
DANNY M. KAUFMAN and QINGNAN ZHOU, Adobe Research, USA
SHAHAR Z. KOVALSKY, Duke University, USA
YAJIE YAN, Facebook, USA
NOAM AIGERMAN, Adobe Research, USA
TAO JU, Washington University in St. Louis, USA

Injective parameterizations of triangulated meshes are critical across applications but remain challenging to compute. Existing algorithms to find injectivity 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. Here we present, to our knowledge, the first algorithm for recovering a globally injective parameterization from an arbitrary non-injective initial mesh subject to stationary constraints. These initial meshes can be inverted, wound about interior vertices and/or overlapping. Our algorithm in turn enables globally injective mapping for meshes with arbitrary positional constraints. Our key contribution is a new energy, called *smooth excess area* (SEA), that measures non-injectivity in a map. This energy is well-defined across both injective and non-injective maps and is smooth almost everywhere, making it readily minimizable using standard gradient-based solvers starting from a non-injective initial state. Importantly, we show that maps minimizing SEA are guaranteed to be locally injective and *almost* globally injective, in the sense that the overlapping area can be made arbitrarily small. Analyzing SEA's behavior over a new benchmark set designed to test injective mapping, we find that optimizing SEA successfully recovers globally injective maps for 85% of the benchmark and obtains locally injective maps for 90%. In contrast, state-of-the-art methods for removing triangle inversion obtain locally injective maps for less than 6% of the benchmark, and achieve global injectivity (largely by chance as prior methods are not designed to recover it) on less than 4%.

CCS Concepts: • **Computing methodologies → Mesh models**.

Additional Key Words and Phrases: Parameterization, mapping, bijective

ACM Reference Format:
Xingyi Du, Danny M. Kaufman, Qingnan Zhou, Shahar Z. Kovalsky, Yajie Yan, Noam Aigerman, and Tao Ju. 2021. Optimizing Global Injectivity for Constrained Parameterization. *ACM Trans. Graph.* 40, 6, Article 1 (December 2021), 17 pages. https://doi.org/10.1145/3478513.3480556

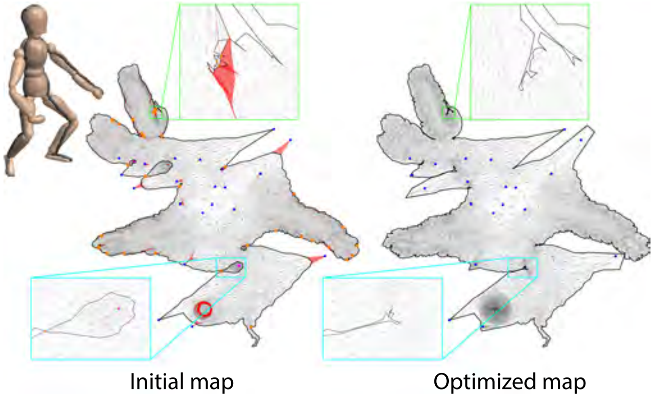

Fig. 1. Given a non-injective initial parameterization of a surface mesh (left) with inverted triangles (red), boundary intersections (orange dots), and overwound vertices (magenta dots), our method recovers a globally injective map (right) while keeping the constraints (blue points) in place. The inserts 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).

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 a comparison between global and local injectivity in Figure 2). Towards this goal, a range of recent methods [Aigerman and Lipman 2013; Du et al. 2020; Fu and Liu 2016; Kovalsky et al. 2015; Su et al. 2019; Weber et al. 2012] 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 [Lipman 2014]. 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

Authors' addresses: Xingyi Du, Washington University in St. Louis, USA, du.xingyi@wustl.edu; Danny M. Kaufman; Qingnan Zhou, Adobe Research, USA; Shahar Z. Kovalsky, Duke University, USA; Yajie Yan, Facebook, USA; Noam Aigerman, Adobe Research, USA; Tao Ju, Washington University in St. Louis, USA.

0730-0301/2021/12-ART1
https://doi.org/10.1145/3478513.3480556

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.

For free boundary problems, computing a globally injective map remains a significant challenge. Harmonic maps into convex domains (e.g. via Tutte's embedding [Tutte 1963]) 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 [Jiang et al. 2017; Smith and Schaefer 2015; Su et al. 2020]. 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) [Jiang et al. 2017], or else to programmatically drag constrained points from an initial, feasible embedding towards their final target positions while continuously preserving injectivity [Schüller et al. 2013]. 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 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) energy proposed in [Du et al. 2020], 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 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.

2 RELATED WORK

Globally injective mapping of a triangulation to the plane is a classic and long-studied problem in geometry processing [Floater and Hormann 2005; Hormann et al. 2007]. Tutte's embedding [Tutte 1963] is the earliest and remains one of the few methods that *guarantees* an injective map to a 2D domain. While extensions have been proposed [Aigerman and Lipman 2015; Floater 2003; Gortler et al. 2006], these maps remain limited to a small class of largely convex target domains and cannot enforce interior constraints.

2.1 Inversion-free and locally-injective maps.

As a step towards injectivity, many recent methods focus on computing non-inverting maps that preserve triangle orientation.

A broad class of barrier-type methods have been designed to compute inversion-free maps. They require an inversion-free initialization (e.g. Tutte's) and then proceed to optimize distortion measures that either implicitly include [Fu et al. 2015; Hormann and Greiner 2000; Smith and Schaefer 2015] or else are explicitly augmented with [Liu et al. 2016; Schüller et al. 2013] barrier terms that diverge as triangle areas collapse. As these energies are stiff, and so challenging to minimize, a diverse range of customized methods have been recently developed to optimize them [Claici et al. 2017; Rabinovich et al. 2017; Shtengel et al. 2017; Zhu et al. 2018].

An alternate range of methods enable initialization from inverted configurations and then optimize towards an orientation-preserving map – possibly satisfying additional positional contraints. Algorithmic strategies vary widely and include projection techniques [Aigerman and Lipman 2013; Kovalsky et al. 2015; Su et al. 2019], deconstructed domains [Weber and Zorin 2014], and nonconforming meshes [Fu and Liu 2016]. Energies to promote non-inversion during optimization have also been designed [Xu et al. 2011] and recently improved to avoid degeneracies (i.e., triangles mapped to a line) [Du et al. 2020]. Likewise, alternate approaches compute parameterization via a sequence of bounded distortion maps (without setting any positional constraints) [Liu et al. 2018], or apply subspace modeling [Hefetz et al. 2019] to efficiently obtain inversion-free maps, albeit with just a limited set of possible positional constraints.

In some cases, changing the topology of the starting triangulation is acceptable. Here remeshing can then be applied to untangle and avoid inversions [Agarwal et al. 2008; Shen et al. 2019], or to avoid singularities during flow [Gu et al. 2018]. Similarly Weber et al. [Weber and Zorin 2014] maps source and target domains to a convex polygon and then remesh to obtain a one-to-one correspondence.

Further afield, changing representation, e.g., to foliations [Campen et al. 2016], enables computation of injective maps to some domains, while quasiconformal maps [Weber et al. 2012] are promisingly locally-injective – however this only holds in the smooth setting.

2.2 Globally injective maps.

As mentioned earlier, in general, non-inversion 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 it has a non-intersecting boundary [Lipman 2014].

To compute globally injective maps, barrier-based methods for 2D parameterization have been further extended [Jiang et al. 2017; Smith and Schaefer 2015; Su et al. 2020] 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 [Bridson et al. 2002; Harmon et al. 2008] have also been adapted for injective deformation tasks [Brochu and Bridson 2009; Harmon et al. 2011; Sacht et al. 2015]. However, these methods generally do not resolve inversions, have no guarantees of convergence and, in practice, often cannot and do not resolve all overlaps [Li et al. 2020].

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. Here we propose a first energy and algorithm that can pass through non-injective intermediate states, as necessary, and so progress towards a final, globally injective solution.

3 PRELIMINARIES

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 ∂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*, $\Phi : M \longrightarrow R^2$, 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 Φ to denote both the map and the mapped image $\Phi(M)$. Similarly, $\partial\Phi$ denotes both the map Φ restricted to the boundary ∂M and the image of this map $\Phi(\partial M)$.

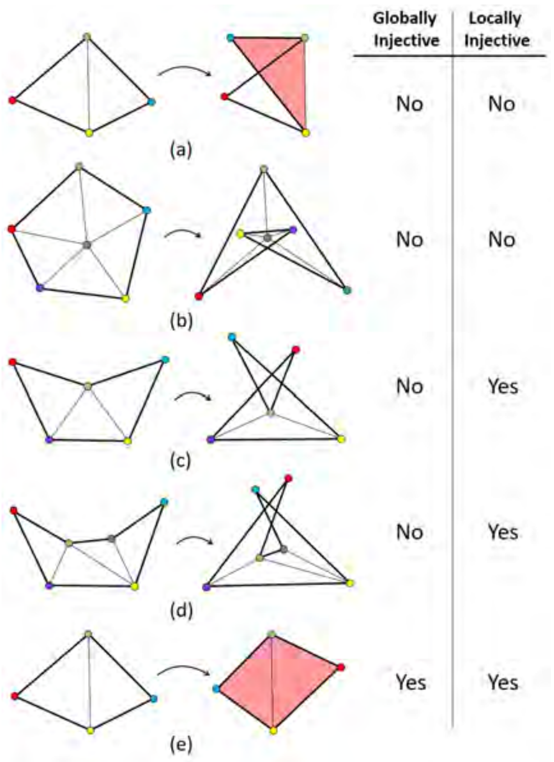

Fig. 2. Simplicial maps and their injectivity. Red triangles are inverted.

We say that a triangle in Φ is *proper*, *degenerate* or *inverted* if its signed area is positive, zero or negative, respectively. The *angle sum* of an interior vertex v in Φ is the sum of all signed angles at v in its incident triangles. This sum is always a multiple of 2π. We call a vertex *overwound* if its angle sum is greater than 2π (e.g., the gray vertex on the right of Figure 2 (b)).

A map is *globally injective*, or simply *injective*, if two distinct points in M are mapped to two different points in the plane. For simplicial maps Φ, Lipman [Lipman 2014] showed that injectivity amounts to asking that (1) the triangles in Φ are either all proper or all inverted and (2) the boundary $\partial\Phi$ does not self-intersect. A map is *locally injective* if, for any point in the interior of M, there exists some local neighborhood of that point within which the map is injective. A simplicial map Φ is locally injective if (1) the triangles in Φ are either all proper or all inverted (so that Φ is injective near each interior edge), and (2) Φ has no overwound vertices (so that Φ is injective near each interior vertex).

We illustrate the difference between global and local injectivity in Figure 2. The simplicial maps in (a,b) fail to be locally injective, due to either the presence of a mixture of proper and inverted triangles (a) or an overwound interior vertex (b). The maps in (c,d) are locally injective, but they are not globally injective due to the self-intersection of the boundary. Finally, the map in (e) is both locally and globally injective, even though all triangles are inverted. In general, a desirable map should be both injective and free of inverted triangles (or *inversion-free*).

4 PROBLEM AND OVERVIEW

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 (e.g., we wish to avoid injective maps like Figure 2 (e)). 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* $\Phi : M \rightarrow R^2$ *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 5, and then we present our optimization algorithm in Section 6.

5 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 5.1). 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 5.2).

5.1 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 shows examples of curves, winding numbers, and regions contributing to occupancy (colored gray).

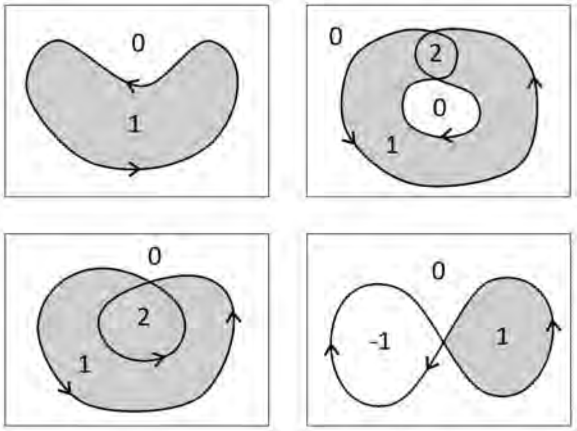

Fig. 3. Winding numbers for different curves. Occupancy of each curve is the area of the gray regions.

We define the *excess area* of a simplicial map Φ as the difference between the total unsigned area of all triangles of Φ, denoted by $A(\Phi)$, and the occupancy of the boundary curve $\partial\Phi$:

$$A_{excess}(\Phi) = A(\Phi) - O(\partial\Phi) \tag{1}$$

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

Proposition 5.1. *For any simplicial map* Φ *of a triangular mesh,*

(1) $A_{excess}(\Phi) \geq A_{overlap}(\Phi)$.
(2) $A_{excess}(\Phi) \geq A_{invert}(\Phi)$.
(3) $A_{excess}(\Phi) \leq A_{overlap}(\Phi) + A_{invert}(\Phi)$.

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 [Chazelle and Edelsbrunner 1992], 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

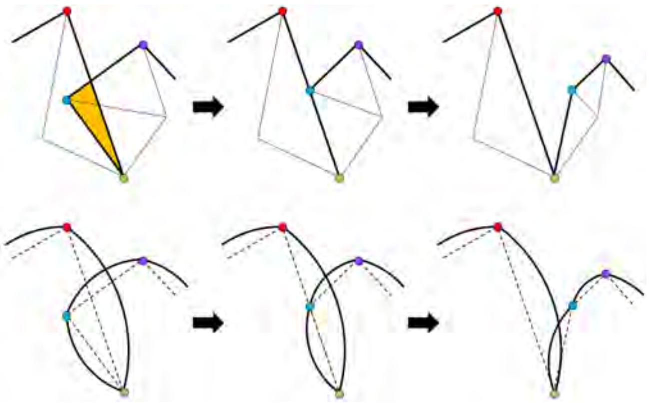

Fig. 4. Top: a non-injective map with overlapping area (yellow) around a boundary vertex (green) transitions to an injective map, going through a configuration where two incident boundary edges overlap (middle). Bottom: the same transition but the straight edges are replaced with their arc-edges; note that the arcs do not overlap during the process.

area can be computed more efficiently than computing all pairwise intersections between triangles.

5.2 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. [2020], the unsigned area $A(\Phi)$ is only C^0 when Φ contains a degenerate triangle. Second, as we show in Appendix B, 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 4 top). In fact, our preliminary attempts at directly optimizing the excess area lead to frequent locking at precisely such degenerate, unavoidable configurations (see Figure 11).

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.

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. [2020]. 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 α 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 $\hat{t}$ is constructed as $\{x_i, y_i, \sqrt{\alpha}\tilde{x}_i, \sqrt{\alpha}\tilde{y}_i\}$. Here, α moderates the contribution from the auxiliary triangle, so that the area of the lifted triangle approaches that of the original triangle as $\alpha \longrightarrow 0$. 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 Φ for a given α as $A_\alpha(\Phi)$. As shown in [Du et al. 2020], for any positive α, $A_\alpha(\Phi)$ is differentiable to any order for all simplicial maps Φ – including those with degenerate triangles.

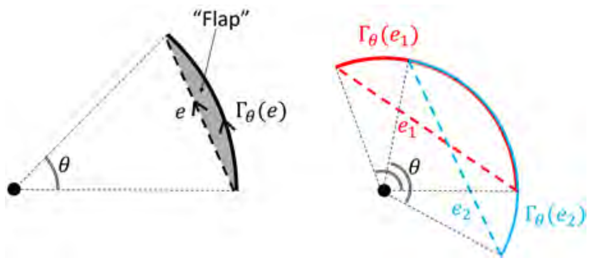

Fig. 5. An arc-edge (left) and two overlapping arc-edges (right).

Next, we improve the smoothness of the occupancy $O(\partial\Phi)$. To avoid non-smooth states like that in Figure 4 (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 5 (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 5 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 4, 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 5 left). Let $B_\theta(\partial\Phi)$ be the sum of all flap areas. The arc-occupancy, $O_\theta(\partial\Phi)$, is then

$$O_\theta(\partial\Phi) = O(\Gamma_\theta(\partial\Phi)) - B_\theta(\partial\Phi). \tag{2}$$

As shown in Proposition B.2 in Appendix B, 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 1) by TLC and arc-occupancy, respectively:

$$E_{\alpha,\theta}(\Phi) = A_\alpha(\Phi) - O_\theta(\partial\Phi). \tag{3}$$

We refer to this energy as *smooth excess area* (or SEA). SEA is C^1 continuous for all simplicial maps Φ 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 7. 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 5.1). Furthermore, SEA is zero when the map is globally injective, $\alpha = 0$, and θ is sufficiently small. More precisely (see proof in Appendix C):

Proposition 5.2. *For any simplicial map Φ of a triangular mesh, $E_{\alpha,\theta}(\Phi) \geq A_{excess}(\Phi)$ for all $\alpha \geq 0, \theta > 0$. Furthermore, if Φ 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 Φ 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 α and θ (see proof in Appendix D):

Proposition 5.3. *Let Φ_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 Φ 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 [Du et al. 2020] 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 (α and θ). 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. [Du et al. 2020] shows that, as α approaches 0, a locally injective map Φ 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 5.2 shows that $E_{0,\theta}(\Phi) = 0$ for all globally injective Φ and sufficiently small θ. Hence the map that achieves the minimal SEA as α 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 [Du et al. 2020] to be the conformal measure mentioned above. As our auxiliary triangles are equilateral, as in [Du et al. 2020], SEA has the tendency to promote equilateral triangles.

6 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 [Wright and Nocedal 1999].

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 (α and θ) 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 Φ 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 [Lipman 2014], namely that Φ 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 Φ. 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 $-O(\Gamma_\theta(\partial\Phi))$. The formula for TLC and its gradient can be found in [Du et al. 2020], 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 [Chazelle and Edelsbrunner 1992]. 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 6 (a) for an illustration. The area of R is the *signed* area of the polygon $\{c_1, \ldots, c_n\}$ (Figure 6 (b)) plus the sum of the *signed* areas bounded by each arc a_i and its chord c_i in the *reverse* orientation (Figure 6 (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.

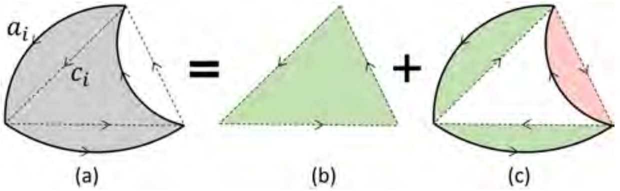

Fig. 6. The area of a region bounded by arcs (a) is the sum of the signed area of the oriented chords (b) and the signed areas bounded by each pair of oriented arc and its chord in reverse orientation (c). Positive and negative areas in (b,c) are colored green and red.

7 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 implementation currently employs an off-the-shelf, limited-memory BFGS quasi-Newton implementation (NLOpt [Johnson [n.d.]]) for energy optimization, with energy and gradient evaluation and assembly implemented 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: α for defining TLC and the center angle θ of the arc-edges. As expected we generally see that smaller values of α are more successful in reaching injectivity, with the concurrent cost of slower convergence for stiffer energy. This is consistent with the behavior for α in TLC [Du et al. 2020]. Similarly, we find that while larger θ (and hence more "bulgy" arc-edges) enable easier optimization steps, this must be balanced against the improving likelihood of reaching injectivity with smaller θ. 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 7.1 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 α 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.

7.1 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 [2020] dataset, which in turn is derived from [Liu et al. 2018]. 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 Φ, 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., Φ). Next, we ignore Φ and compute a new map Φ' from M to the plane using ARAP [Igarashi et al. 2005] while satisfying the same constraints (i.e., $\Phi'(S) = T$). If Φ' is *not* injective, it is included in our benchmark as an initial map. This process is illustrated in Figure 7. Observe that the ARAP map Φ', as in (c), can differ significantly from the globally injective map Φ that comes with the mesh, as in (b).

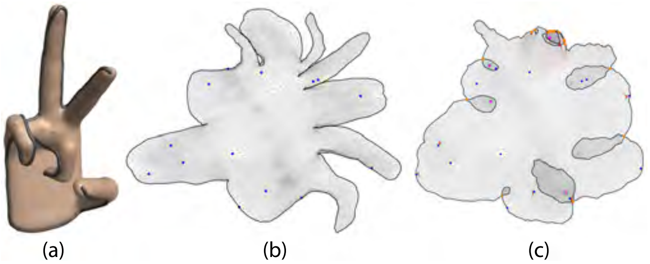

Fig. 7. Benchmark creation process: starting from an open surface mesh M (a) with a globally injective map Φ to the plane (b) from the [Du et al. 2020] dataset, we randomly select vertices in Φ to be constraints (blue dots in (b)) and compute a new map Φ' from M using ARAP that meets those constraints (c). If (as here) the resulting map Φ' is *not* injective it is added to our benchmark.

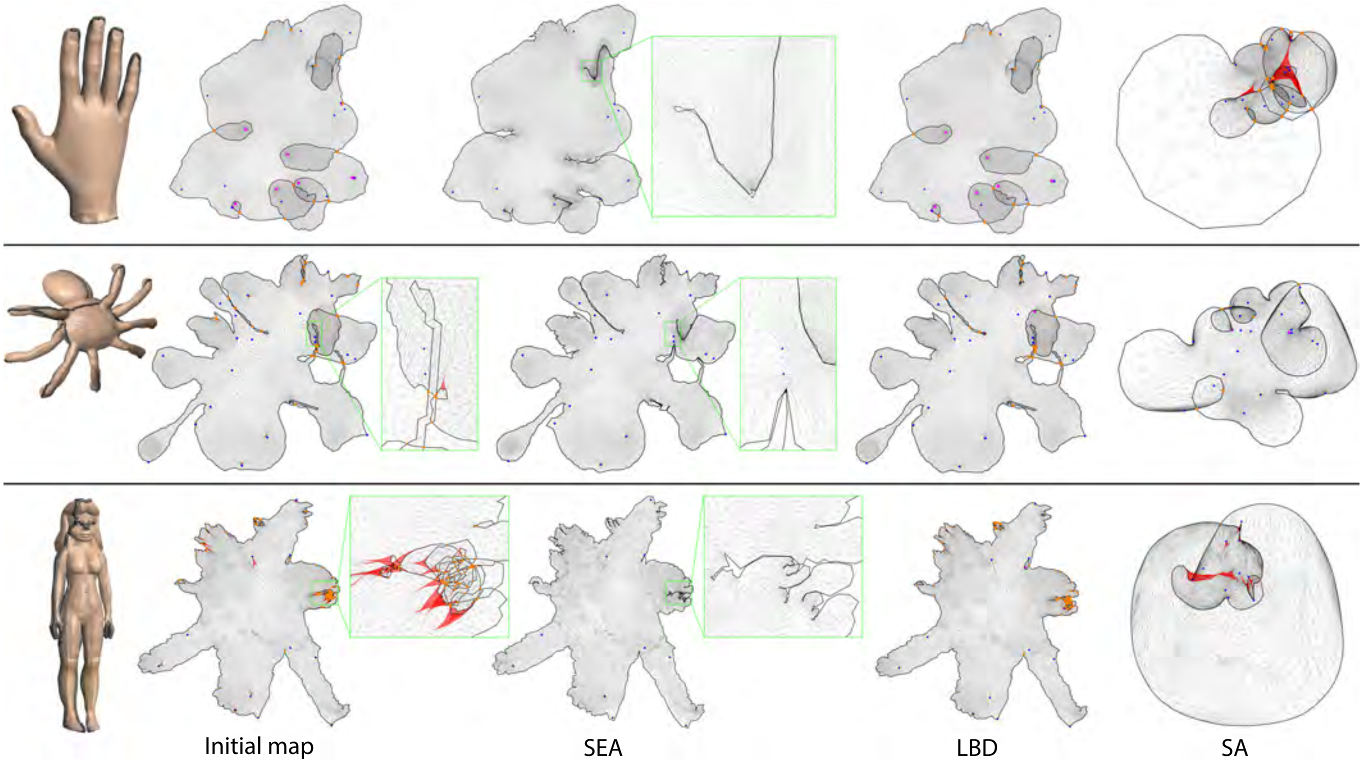

Fig. 8. Three successful examples from the benchmark: initial maps (first column), maps produced by our method (second column), which are all globally injective, and maps produced by LBD (third column) and SA (last column), none of which are locally or globally injective.

Table 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 1, the behavior of both the SA [Fu and Liu 2016] and LBD [Kovalsky et al. 2015] 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.

Table 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.

	SEA (Our method)	SA	LBD
Inversion free	**1629** (91.0%)	179 (9.9%)	243 (13.5%)
Local injectivity	**1618** (90.3%)	65 (3.6%)	108 (6.0%)
Global injectivity	**1537** (85.8%)	56 (3.1%)	20 (1.1%)

To more closely examine the behavior of these methods, we first take a closer look at some of the specific benchmark example results. Figure 8 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 9 then demonstrates two representative failure cases for our method. We observe that the majority of failures with boundary

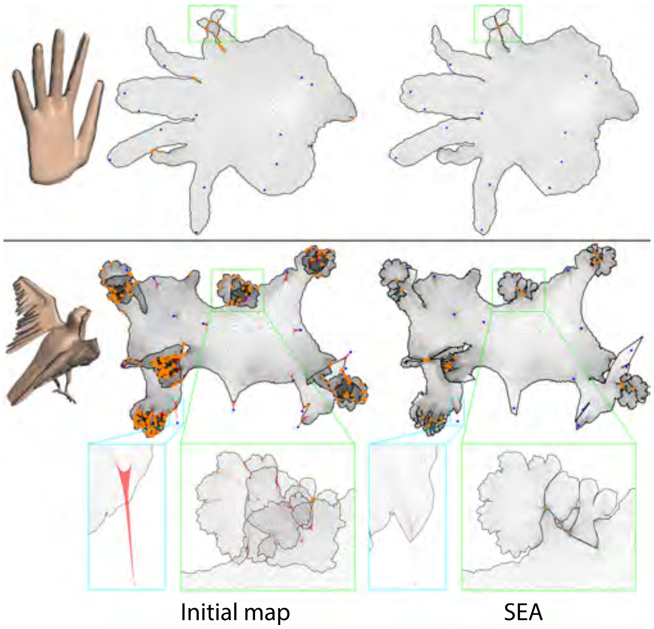

Fig. 9. Two failure examples from the benchmark: initial maps (first column) and non-globally-injective maps produced by our method (second column).

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 15). 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.

Figure 10 (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 10 (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.

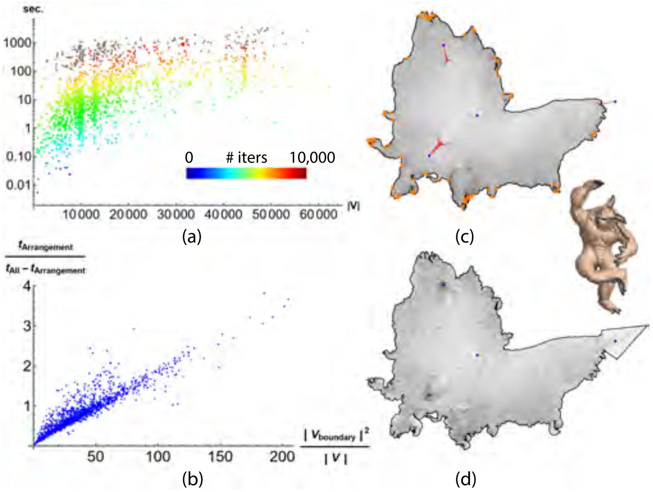

Fig. 10. Performance: (a) log running time in seconds (vertical axis) versus vertex count (horizontal) for all benchmark examples. Each dot represents one mesh and is colored by number of QN iterations (gray dots are fails). (b) Ratio of arrangement computation time over the remaining time (vertical 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)).

7.2 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 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 11 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 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 boundary intersection. In contrast, optimization with SEA using arc-occupancy is able to continue until the energy converges to a globally injective map

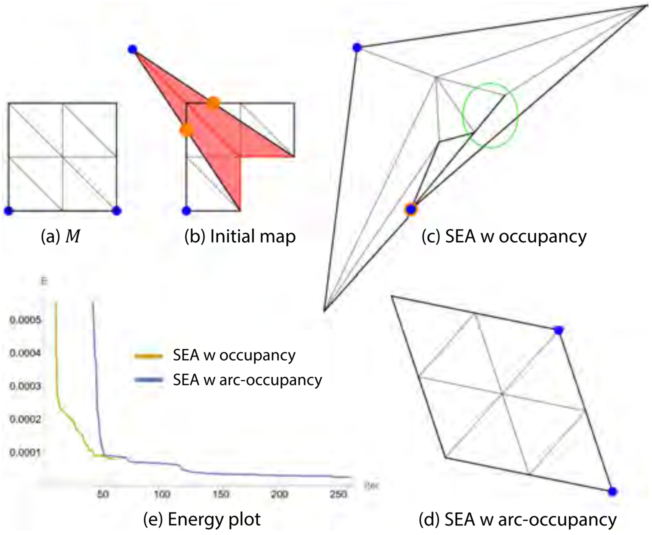

Fig. 11. (a): A mesh with two constraint vertices. (b): An initial map containing inverted triangles. (c,d): Results of optimizing for SEA with occupancy vs. with arc-occupancy. (e): Energy plot of the two energy during optimization.

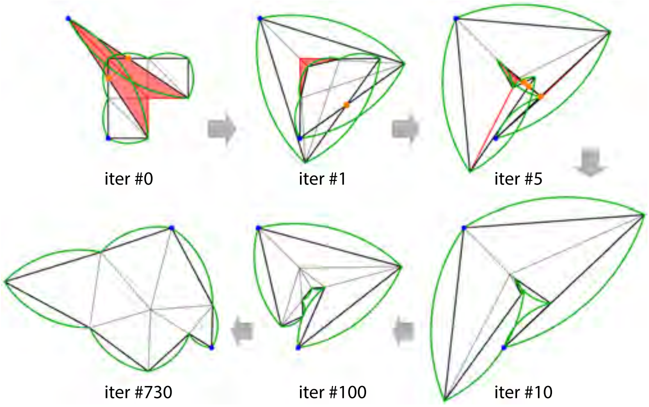

Fig. 12. Intermediate results of optimizing SEA energy for the example in Figure 11 at $\theta = 1.0$. Arc-edges are colored green. The energy converges at iteration 763 and produces the same map as in Figure 11 (d).

(see (d)). Again, observe that triangles are close to equilateral when SEA converges, due to our use of equilateral auxiliary triangles.

In Figure 12 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 12 we set $\theta = 1.0$ so that the arc-edges are easier to see. Even at this large θ, SEA converges to the same globally injective map as $\theta = 0.1$ (Figure 11 (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 θ values (0.1, 1.0) can be found in the accompanying video.

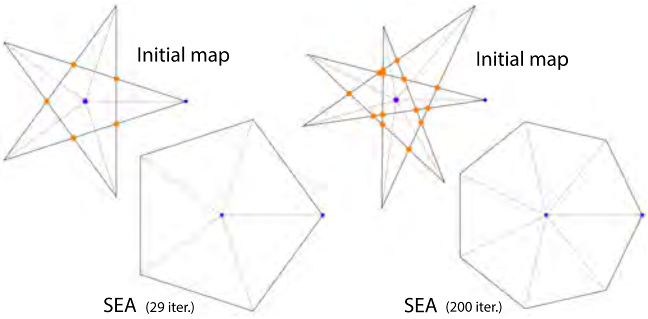

Fig. 13. Two initial maps with overwound vertices (top) and the injective maps produced by our method (bottom).

We next consider the two examples in Figure 13, 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 4π (top) and 6π (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 14 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 [Fu and Liu 2016] and LBD [Kovalsky et al. 2015] can remove (most) triangle inversion in these examples, they leave behind other types of non-injectivity such as boundary intersections and overwound vertices.

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 9), is the "crossing-arm" case illustrated in Figure 15. 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.

7.3 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 successful in reaching

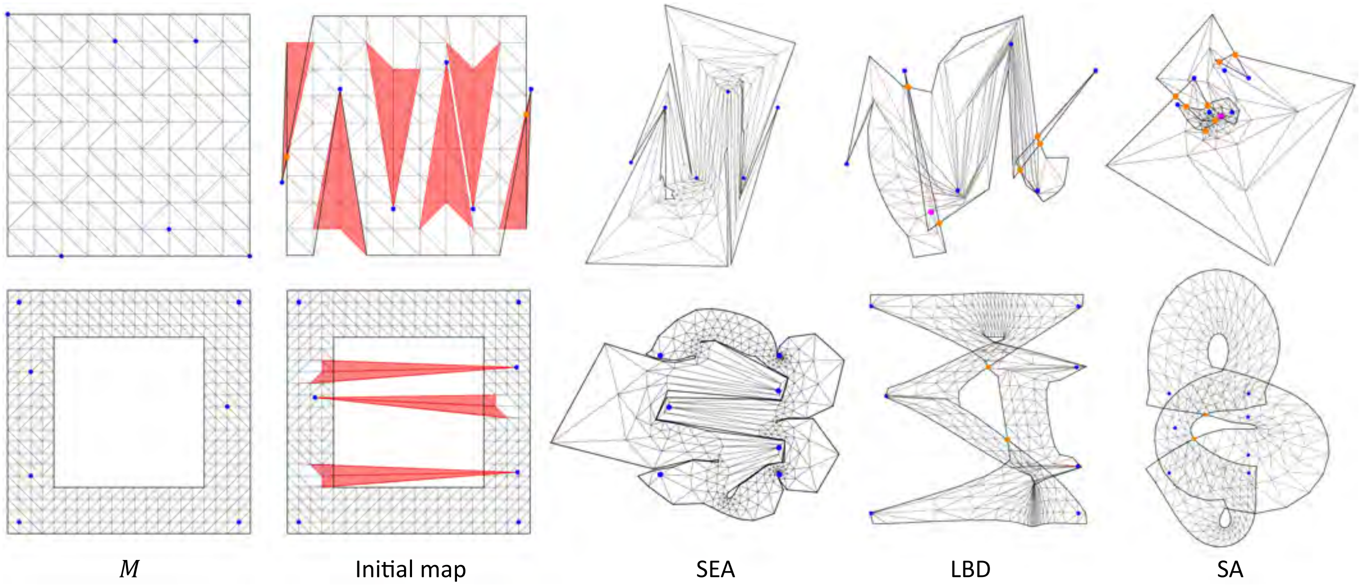

Fig. 14. From left to right: input meshes (with one or two boundaries), non-injective initial maps, globally injective maps produced by our method (SEA), and results of LBD and SA. Both LBD and SA have removed most or all inverted triangles, but the results are neither locally or globally injective due to overwound vertices and boundary intersections.

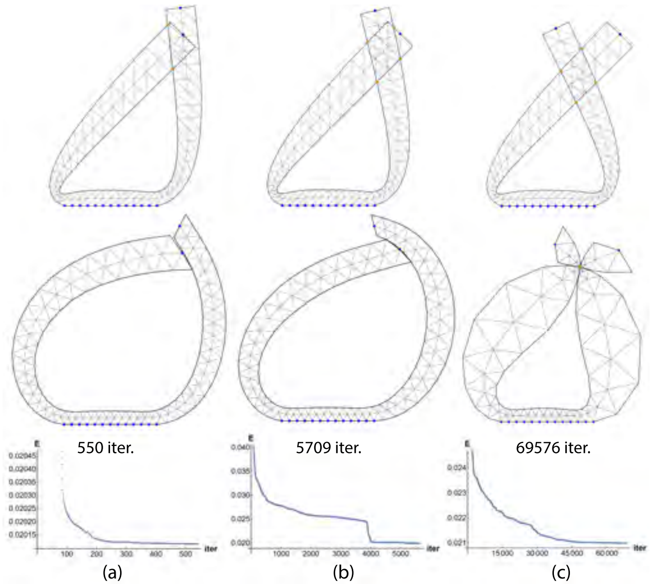

Fig. 15. The "crossing-arm" configuration: initial maps with increasing extent of crossing (top row, from left to right), and maps produced by our method (middle row) where only the first two are globally injective. The energy plots are shown at the bottom.

injectivity for initial maps that are closer to an injective state and free from the crossing-arm configurations. Figure 16 shows two different initial maps for the hand mesh in Figure 8 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 9.

7.4 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 [Fang et al. 2021; Jiang et al. 2017; Smith and Schaefer 2015]. As an example, starting from the SEA-optimized injective map of the hand mesh in Figure 8, Figure 17 shows the result of minimizing the Symmetric Dirichlet energy [Smith and Schaefer 2015] using the simplicial augmentation framework of [Jiang et al. 2017]. 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.

8 DISCUSSION

We have presented, to our knowledge, a first method to recover global injectivity from an non-injective map subject to arbitrary

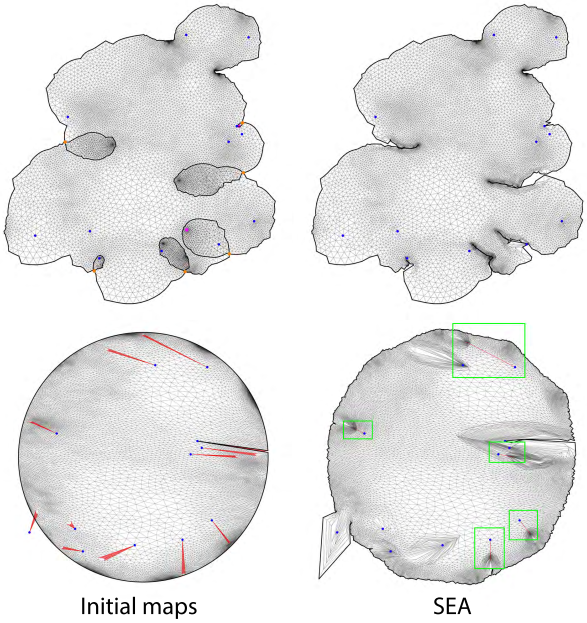

Fig. 16. Changing initialization type: results of our method on two alternate initializations of the hand mesh in Figure 8 while applying the same constraints. Injectivity is achieved starting from the top initial map but not for the bottom one (green boxes highlight the inverted triangles).

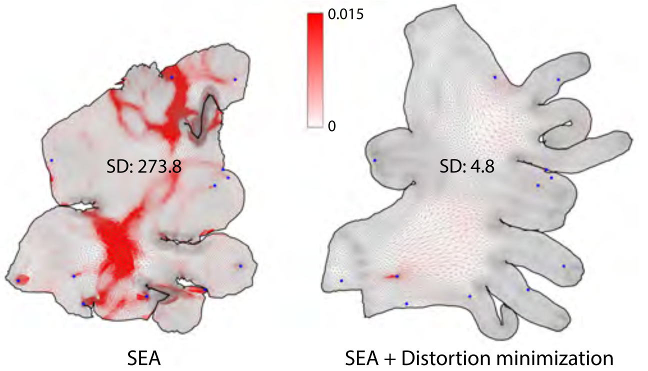

Fig. 17. Left: output of an SEA-optimized injective map for the hand mesh in Figure 8. Right: further minimizing the SEA-generated map with the Symmetric Dirichlet (SD) distortion energy via the method of [Jiang et al. 2017], which preserves global injectivity and point constraints. Triangles are colored by their SD distortion.

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 interesting possibility to better incorporate the complement space. Furthermore, while our energy currently fixes α and θ 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.

ACKNOWLEDGMENTS

This work is supported in part by a gift from Adobe Research. We would like to thank authors of several papers for providing code, data, and help with our experiments.

REFERENCES

Pankaj K Agarwal, Bardia Sadri, and Hai Yu. 2008. Untangling triangulations through local explorations. In *Proceedings of the twenty-fourth annual symposium on Computational geometry*. ACM, 288–297.

Noam Aigerman and Yaron Lipman. 2013. Injective and bounded distortion mappings in 3D. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 106.

Noam Aigerman and Yaron Lipman. 2015. Orbifold Tutte embeddings. *ACM Trans. Graph.* 34, 6 (2015), 190–1.

Robert Bridson, Ronald Fedkiw, and John Anderson. 2002. Robust treatment of collisions, contact and friction for cloth animation. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. 594–603.

Tyson Brochu and Robert Bridson. 2009. Robust topological operations for dynamic explicit surfaces. *SIAM Journal on Scientific Computing* 31, 4 (2009), 2472–2493.

Marcel Campen, Cláudio T Silva, and Denis Zorin. 2016. Bijective maps from simplicial foliations. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 74.

Bernard Chazelle and Herbert Edelsbrunner. 1992. An optimal algorithm for intersecting line segments in the plane. *Journal of the ACM (JACM)* 39, 1 (1992), 1–54.

Sebastian Claici, Mikhail Bessmeltsev, Scott Schaefer, and Justin Solomon. 2017. Isometry-Aware Preconditioning for Mesh Parameterization. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 37–47.

Xingyi Du, Noam Aigerman, Qingnan Zhou, Shahar Z Kovalsky, Yajie Yan, Danny M Kaufman, and Tao Ju. 2020. Lifting simplices to find injectivity. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 120–1.

Yu Fang, Minchen Li, Chenfanfu Jiang, and Danny M. Kaufman. 2021. Guaranteed Globally Injective 3D Deformation Processing. *ACM Trans. on Graphics (SIGGRAPH 2021)* (2021).

Michael Floater. 2003. One-to-one piecewise linear mappings over triangulations. *Math. Comp.* 72, 242 (2003), 685–696.

Michael S Floater and Kai Hormann. 2005. Surface parameterization: a tutorial and survey. In *Advances in multiresolution for geometric modelling*. Springer, 157–186.

Xiao-Ming Fu and Yang Liu. 2016. Computing inversion-free mappings by simplex assembly. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 216.

Xiao-Ming Fu, Yang Liu, and Baining Guo. 2015. Computing locally injective mappings by advanced MIPS. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 71.

Steven Gortler, Craig Gotsman, and Dylan Thurston. 2006. Discrete one-forms on meshes and applications to 3D mesh parameterization. *Computer Aided Geometric Design* (2006).

Xianfeng Gu, Ren Guo, Feng Luo, Jian Sun, Tianqi Wu, et al. 2018. A discrete uniformization theorem for polyhedral surfaces II. *Journal of differential geometry* 109, 3 (2018), 431–466.

David Harmon, Daniele Panozzo, Olga Sorkine, and Denis Zorin. 2011. Interference-aware geometric modeling. *ACM Transactions on Graphics (TOG)* 30, 6 (2011), 1–10.

David Harmon, Etienne Vouga, Rasmus Tamstorf, and Eitan Grinspun. 2008. Robust treatment of simultaneous collisions. In *ACM SIGGRAPH 2008 papers*. 1–4.

Eden Fedida Hefetz, Edward Chien, and Ofir Weber. 2019. A Subspace Method for Fast Locally Injective Harmonic Mapping. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 105–119.

Kai Hormann and Giinther Greiner. 2000. MIPS: An Efficient Global Parametrization Method. *France on 1-7 July 1999. Proceedings, Volume 1. Curve and Surface Design. F61775-99-WF068* (2000), 153.

Kai Hormann, Bruno Lévy, and Alla Sheffer. 2007. Mesh parameterization: Theory and practice. (2007).

Takeo Igarashi, Tomer Moscovich, and John F Hughes. 2005. As-rigid-as-possible shape manipulation. *ACM transactions on Graphics (TOG)* 24, 3 (2005), 1134–1141.

Zhongshi Jiang, Scott Schaefer, and Daniele Panozzo. 2017. Simplicial complex augmentation framework for bijective maps. *ACM Transactions on Graphics* 36, 6 (2017).

Steven G. Johnson. [n.d.]. The NLopt nonlinear-optimization package. http://github.com/stevengj/nlopt

Shahar Z Kovalsky, Noam Aigerman, Ronen Basri, and Yaron Lipman. 2015. Large-scale bounded distortion mappings. *ACM Trans. Graph.* 34, 6 (2015), 191–1.

Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. 2020. Incremental Potential Contact: Intersection- and Inversion-free Large Deformation Dynamics. *ACM Transactions on Graphics* 39, 4 (2020).

Yaron Lipman. 2014. Bijective Mappings of Meshes with Boundary and the Degree in Mesh Processing. *SIAM Journal on Imaging Sciences [electronic only]* 7 (04 2014). https://doi.org/10.1137/130939754

Ligang Liu, Chunyang Ye, Ruiqi Ni, and Xiao-Ming Fu. 2018. Progressive parameterizations. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 41.

Tiantian Liu, Ming Gao, Lifeng Zhu, Eftychios Sifakis, and Ladislav Kavan. 2016. Fast and Robust Inversion-Free Shape Manipulation. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 1–11.

Michael Rabinovich, Roi Poranne, Daniele Panozzo, and Olga Sorkine-Hornung. 2017. Scalable locally injective mappings. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 37a.

Leonardo Sacht, Etienne Vouga, and Alec Jacobson. 2015. Nested cages. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 1–14.

Christian Schüller, Ladislav Kavan, Daniele Panozzo, and Olga Sorkine-Hornung. 2013. Locally injective mappings. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*. Eurographics Association, 125–135.

Hanxiao Shen, Zhongshi Jiang, Denis Zorin, and Daniele Panozzo. 2019. Progressive embedding. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 32.

Anna Shtengel, Roi Poranne, Olga Sorkine-Hornung, Shahar Z Kovalsky, and Yaron Lipman. 2017. Geometric optimization via composite majorization. *ACM Trans. Graph.* 36, 4 (2017), 38–1.

Jason Smith and Scott Schaefer. 2015. Bijective parameterization with free boundaries. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 70.

Jian-Ping Su, Xiao-Ming Fu, and Ligang Liu. 2019. Practical Foldover-Free Volumetric Mapping Construction. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 287–297.

Jian-Ping Su, Chunyang Ye, Ligang Liu, and Xiao-Ming Fu. 2020. Efficient bijective parameterizations. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 111–1.

William Thomas Tutte. 1963. How to draw a graph. *Proceedings of the London Mathematical Society* 3, 1 (1963), 743–767.

Ofir Weber, Ashish Myles, and Denis Zorin. 2012. Computing extremal quasiconformal maps. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 1679–1689.

Ofir Weber and Denis Zorin. 2014. Locally injective parametrization with arbitrary fixed boundaries. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 75.

Stephen J Wright and Jorge Nocedal. 1999. *Numerical optimization*. Vol. 2. Springer New York.

Yin Xu, Renjie Chen, Craig Gotsman, and Ligang Liu. 2011. Embedding a triangular graph within a given boundary. *Computer Aided Geometric Design* 28, 6 (2011), 349–356.

Yufeng Zhu, Robert Bridson, and Danny M Kaufman. 2018. Blended cured quasi-newton for distortion optimization. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 40.

A EXCESS, OVERLAP, AND INVERTED AREAS

Here we prove Proposition 5.1 in Section 5, 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 [Lipman 2014]) to meshes with both proper and inverted triangles:

Lemma A.1. *Let Φ be a simplicial map of a triangular mesh, and z a point in the plane that is not on any vertex or edge of Φ. The winding number of $\partial\Phi$ around z is the number of proper triangles of Φ that cover z minus the number of inverted triangles of Φ 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 Φ 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 Φ 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 Φ. □

We now prove the proposition using the lemma above:

Proof of Proposition 5.1: Lemma A.1 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 $A_{cover}(\Phi)$ be the total area of the plane covered by Φ, and $A_{proper}(\Phi)$ the total area of proper triangles of Φ. Then,

$$\begin{aligned} A_{cover}(\Phi) &\geq O(\partial\Phi) \\ A_{proper}(\Phi) &\geq O(\partial\Phi) \end{aligned} \tag{4}$$

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

$$\begin{aligned} A_{overlap}(\Phi) &= A(\Phi) - A_{cover}(\Phi) \\ A_{invert}(\Phi) &= A(\Phi) - A_{proper}(\Phi) \end{aligned} \tag{5}$$

Substituting the Equations 4 into Equations 5 yields the first two inequalities of the proposition. To prove the last inequality, note that points covered by Φ but not contributing to the occupancy of $\partial\Phi$ must be covered by inverted triangles of Φ. So we have:

$$A_{cover}(\Phi) - O(\partial\Phi) \leq A_{invert}(\Phi). \tag{6}$$

Substituting Equation 6 into the first equation of Equations 5 yields:

$$\begin{aligned} A_{overlap}(\Phi) &\geq A(\Phi) - (O(\partial\Phi) + A_{invert}(\Phi)) \\ &= A_{excess}(\Phi) - A_{invert}(\Phi). \end{aligned} \tag{7}$$

□

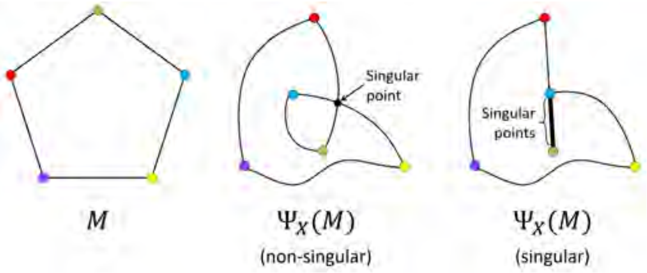

Fig. 18. Comparing singular and non-singular maps.

B 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.

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 $\Psi : R^n \times M \rightarrow R^2$ 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 Ψ is a simplicial map, the map parameters are the 2D coordinates of the mapped vertices of M. However, Ψ is not limited to simplicial maps, and Ψ 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 B.1. *If $\Psi(X, p)$ is continuous in both X and $p \in M$, then $O(X)$ in 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. □

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 18 middle). The map Ψ 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 18 right).

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

Proposition B.2. *If $\Psi_X(p)$ is differentiable in both X and p over each edge of M, $O(X)$ is differentiable at all X where Ψ is not singular.*

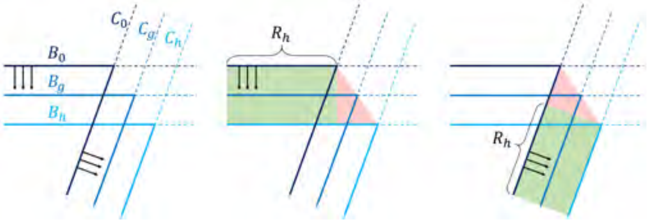

Fig. 19. Illustration for the proof. Left: the mapped curves (dotted lines) and the corresponding boundaries of positively winding regions (solid lines) at different parameters (distinguished by shades of blue). Middle and right: the regions swept by the regular and non-regular subsets of B_g, shaded 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 $h = 0$.

Proof. Let $A = \{a_1, \ldots, a_n\}$ be a set of parameters at which Ψ 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,

$$\frac{\partial O}{\partial x_i}(A) = \lim_{h \to 0} \frac{O(A + h\delta_i) - O(A)}{h}, \tag{8}$$

where δ_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 B.1, 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 19 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 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 Π_g the composite map $\Psi_{A+g\delta_i} \circ \Psi_A^{-1}$ that takes points on C_0 to points on C_g. Note that Π_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 Ψ 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 Π_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 19 (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 *regular sweep* (denoted by RS_h), and the area swept by the remaining subsets $B_g \setminus \Pi_g(R_h)$ the *non-regular sweep* (denoted by NS_h). We can re-write the limit in Equation 8 as:

$$\lim_{h\to 0} \frac{O(A + h\delta_i) - O(A)}{h} = \lim_{h\to 0} \frac{RS_h + NS_h}{h} = \lim_{h\to 0} \frac{RS_h}{h} + \lim_{h\to 0} \frac{NS_h}{h}, \tag{9}$$

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,

  $$RS_h = \sum_{r \in CC(R_h)} \int_r \int_0^h \det(\nabla \Pi_g(q))\, dg\, dq$$

  where $\nabla \Pi_g(q)$ is the Jacobian of Π at g and q, and det is the determinant operator. As $h \longrightarrow 0$, 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,

  $$\lim_{h\to 0} \frac{RS_h}{h} = \sum_{r \in CC(B_0 \setminus S_0)} \int_r \det(\nabla \Pi_0(q))\, dq \tag{10}$$

- Non-regular sweep: We shall first bound the *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 $g \in [0, h]$, $f(q) \le h$. To see why, suppose $q \in B_0$, then q cannot be in the regular subset R_h, and hence $f(q) \le h$. 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 $g' \le g$, implying that $f(q) \le g' \le h$. 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 Ψ in x_i, which is bounded, (2) is bounded by $h * \gamma$ where γ 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

  $$\|NS_h\| \le l_h * \gamma * h$$

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

  $$\lim_{h\to 0} \frac{\|NS_h\|}{h} \le \lim_{h\to 0} l_h * \gamma = 0,$$

  which implies

  $$\lim_{h\to 0} \frac{NS_h}{h} = 0 \tag{11}$$

Substituting Equations 10 and 11 into 9 shows that the limit in Equation 8 exists for h approaching from the positive side, and the limit equals the righthand side of Equation 10. The case of h approaching from the negative side is completely symmetric, and the limit is identical to that in Equation 10. This proves that the partial derivative $\frac{\partial O}{\partial x_i}(A)$ exists for any $x_i \in X$. □

C SEA AND EXCESS AREA

Here we prove Proposition 5.2 in Section 5, 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 C.1. *Let Φ be a simplicial map of a triangular mesh and $\theta > 0$, then $O_\theta(\partial\Phi) \le O(\partial\Phi)$. Furthermore, if Φ is globally injective and inversion-free, there exists some θ_0 such that $O_\theta(\partial\Phi) = O(\partial\Phi)$ for all $\theta < \theta_0$.*

Proof. Following the same argument in the proof of Lemma A.1, 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. $\Gamma_\theta(\partial\Phi)$ 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,

$$O(\Gamma_\theta(\partial\Phi)) \le O(\partial\Phi) + B_\theta(\partial\Phi).$$

Substituting the above into Equation 2 yields $O_\theta(\partial\Phi) \le O(\partial\Phi)$. 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 Φ and all $\theta < \theta_0$ for some positive value θ_0 that can be derived from Φ. 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 θ_0 be the smallest value of θ such that the condition no longer holds. Since Φ is injective and inversion-free, as θ 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 Φ, e_1, e_2 do not intersect, and θ_0 is the smallest θ 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 θ_0 equals the exterior angle of that vertex. In either case, θ_0 is a non-zero value that can be derived from Φ. □

We shall refer to the threshold angle θ_0 in Lemma C.1 as the *clearance angle* of an injective map Φ. Now we prove the proposition:

Proof of Proposition 5.2: We note that the TLC is an upper bound of the unsigned area, that is, $A_\alpha(\Phi) \geq A(\Phi)$ for all $\alpha \geq 0$ [Du et al. 2020]. Since $O_\theta(\partial\Phi) \leq O(\partial\Phi)$ for all $\theta > 0$ by Lemma C.1, we derive, for $\alpha \geq 0, \theta > 0$:

$$E_{\alpha,\theta}(\Phi) = A_\alpha(\Phi) - O_\theta(\partial\Phi) \geq A(\Phi) - O(\partial\Phi) = A_{excess}(\Phi) \quad (12)$$

Suppose Φ is injective and inversion-free, and let θ_0 be its clearance angle. Lemma C.1 shows that $O_\theta(\partial\Phi) = O(\partial\Phi)$ for all $\theta < \theta_0$. Since $A_0(\Phi) = A(\Phi)$, the inequality in Equation 12 becomes equality for $\alpha = 0$ and $\theta < \theta_0$, making $E_{0,\theta}(\Phi) = A_{excess}(\Phi)$. Finally, since Φ has no overlapping or inverted triangles, Proposition 5.1 shows that $A_{excess}(\Phi) = 0$, which proves $E_{0,\theta}(\Phi) = 0$ for all $\theta < \theta_0$. □

D INJECTIVITY AT SEA MINIMA

Here we prove the injectivity guarantee for SEA (Proposition 5.3). The proof closely follows that of the injectivity guarantee for TLC in [Du et al. 2020] (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 [Du et al. 2020]. We shall refer to [Du et al. 2020] 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 [Du et al. 2020]):

Lemma D.1 (Lemma B.1 in [Du et al. 2020]). *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 Φ will contain a triangle with an arbitrarily large Dirichlet energy if there is some triangle of Φ with sufficiently small unsigned area:

Lemma D.2 (Lemma B.2 in [Du et al. 2020]). *For any $\delta > 0$, there exists some $\epsilon > 0$ such that if a simplicial map Φ contains a triangle whose unsigned area is smaller than ϵ, then Φ 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 [Du et al. 2020] 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 Φ to a boundary edge e_0. It shows that, if $D_t < \delta$ for all t in Φ, 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 Φ). 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 Φ, that ratio is a constant independent of Φ, and hence the edge length $\|e_0\|$ has a constant lower bound. For every other edge e (interior or exterior) of Φ 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 [Du et al. 2020], 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 Φ. □

The next lemma builds on the previous one and states that the rate of increase in TLC as α increases from 0 can be arbitrarily large, if Φ has a sufficiently small triangle (see proof in [Du et al. 2020]):

Lemma D.3 (Lemma B.3 in [Du et al. 2020]). *For any $\delta > 0$, there exists some $\epsilon > 0$ and $\beta > 0$ such that if a simplicial map Φ contains a triangle whose unsigned area is smaller than ϵ, then for any $\alpha < \beta$, $\partial A_\alpha(\Phi)/\partial\alpha > \delta$.*

An immediate corollary of this lemma is that if Φ 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, [Du et al. 2020] 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 D.4. *Suppose a simplicial map Φ contains an interior vertex v such that v is incident to only proper triangles and the sum of angles around v is not 2π. Then $A_{overlap}(\Phi) \geq \epsilon\eta^2\pi/2\delta$, where ϵ is the minimum of A_t among all $t \in \Phi$, η is the shortest height of any*

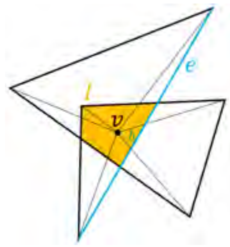

Fig. 20. Illustration for the proof.

auxiliary triangle $\tilde{t}$ for $t \in \Phi$, and δ 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 2π. 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 20). Each point in I (away from the vertices and edges of Φ) is covered by at least two triangles incident to v. Let $A(I)$ be the area of I, we therefore have $A_{overlap}(\Phi) \geq A(I)$.

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:

$$A(I) \geq h * L(I)/2$$

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 D.1, we have:

$$h * \|e\| \geq 2\epsilon$$

and

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

Multiplying these two inequalities gives:

$$h^2 \geq \epsilon\eta^2/2\delta$$

As a result,

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

□

Now we prove the proposition using the lemmas above:

Proof of Proposition 5.3: We closely follow the approach taken in the proof of Proposition 4.3 in [Du et al. 2020]. We separately consider the case that Φ has some small-area triangle (including degenerate triangles) and the case that it does not. The key difference with [Du et al. 2020] lies in the latter case, which we further divide into three sub-cases: Φ has some inverted triangle; Φ has overwound interior vertices; or Φ has an overlapping area greater than λ 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 Φ_0 is injective, Proposition 5.2 shows that there exists some θ_0 (the clearance angle of Φ_0) such that $E_{0,\theta}(\Phi_0) = 0$ for all $\theta < \theta_0$. In the following, we consider some fixed θ in this range, and we shorthand the SEA energy $E_{\alpha,\theta}$ as E_α. Second, since the occupancy term of SEA does not depend on α, SEA shares the same partial derivative as TLC with respect to α. That is, $\partial A_\alpha(\Phi)/\partial\alpha = \partial E_\alpha(\Phi)/\partial\alpha$ for any map Φ (injective or not) and $\alpha \geq 0$. Third, since Φ_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 τ, and let δ be the maximum value of $\partial E_\alpha(\Phi_0)/\partial\alpha$ for all $\alpha < \tau$.

We first consider the case that Φ has a triangle whose unsigned area is smaller than ϵ, which is found by Lemma D.3 for δ. This case is already considered in the proof of Proposition 4.3 in [Du et al. 2020], which shows that there exists some $\beta \in (0, \tau]$ such that for any $\alpha < \beta$, $\partial E_\alpha(\Phi)/\partial\alpha > \delta \geq \partial E_\alpha(\Phi_0)/\partial\alpha$. On the other hand, by Proposition 5.2, $E_0(\Phi) \geq 0 = E_0(\Phi_0)$. Hence $E_\alpha(\Phi) > E_\alpha(\Phi_0)$ for any $\alpha < \beta$.

Otherwise, suppose Φ has no triangle whose unsigned area is smaller than ϵ (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 $A_t D_t > 2\epsilon\delta$. Following the arguments in the proof of Lemma B.3 in [Du et al. 2020], the conclusion of that lemma (which is re-stated as Lemma D.3 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 Φ is not locally injective, not inversion-free, or $A_{overlap}(\Phi) > \lambda$. If Φ is not locally injective or inversion-free, Φ either has an inverted triangle (whose unsigned area is no smaller than ϵ) or has an interior vertex whose angle sum is not 2π. We separately consider these sub-cases.

- Suppose Φ has some inverted triangle, whose unsigned area is no smaller than ϵ. By Propositions 5.1 and 5.2, $E_\alpha(\Phi) \geq A_{invert}(\Phi) \geq \epsilon$ 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 Φ contains only proper triangles but some vertex v has an angle sum other than 2π. Since $A_t \geq \epsilon$ and $D_t \leq 2\delta$ for all $t \in \Phi$, Lemma D.4 shows that $A_{overlap}(\Phi) \geq \sigma = \epsilon\eta^2\pi/4\delta$ where η is the smallest height among all auxiliary triangles. Note that σ is independent of the map Φ. By Propositions 5.1 and 5.2, $E_\alpha(\Phi) \geq A_{overlap}(\Phi) \geq \sigma$ for any $\alpha \geq 0$. Similar to the previous sub-case, there exists some $\kappa_2 > 0$ such that, for all $\alpha < \kappa_2$, $E_\alpha(\Phi_0) < \sigma \leq E_\alpha(\Phi)$.
- Suppose $A_{overlap}(\Phi) > \lambda$. By Propositions 5.1 and 5.2, $E_\alpha(\Phi) \geq A_{overlap}(\Phi) > \lambda$ 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 $\alpha_0 = \min(\beta, \kappa_1, \kappa_2, \kappa_3)$. □