# Least Squares Affine Transitions for Global Parameterization

Ana Maria Vintescu
LTCI - Télécom ParisTech - Institut Mines-Telecom
75013 Paris, France
vintescu@telecom-paristech.fr

Florent Dupont
Université de Lyon, CNRS
Université Lyon 1, LIRIS UMR 5205
69622 Villeurbanne, France
florent.dupont@liris.cnrs.fr

Guillaume Lavoué
Université de Lyon, CNRS
INSA-Lyon, LIRIS UMR 5205
69621 Villeurbanne, France
glavoue@liris.cnrs.fr

Pooran Memari
LIX UMR 7161, CNRS, École Polytechnique, Université Paris Saclay
91128 Palaiseau Cedex - France
memari@lix.polytechnique.fr

Julien Tierny
Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606
75005 Paris, France
julien.tierny@lip6.fr

## ABSTRACT

This paper presents an efficient algorithm for a global parameterization of triangular surface meshes. In contrast to previous techniques which achieve global parameterization through the optimization of non-linear systems of equations, our algorithm is solely based on solving at most two linear equation systems, in the least square sense. Therefore, in terms of running time the unfolding procedure is highly efficient. Our approach is direct – it solves for the planar UV coordinates of each vertex directly – hence avoiding any numerically challenging planar reconstruction in a post-process. This results in a robust unfolding algorithm. Curvature prescription for user-provided cone singularities can either be specified manually, or suggested automatically by our approach. Experiments on a variety of surface meshes demonstrate the runtime efficiency of our algorithm and the quality of its unfolding. To demonstrate the utility and versatility of our approach, we apply it to seamless texturing. The proposed algorithm is computationally efficient, robust and results in a parameterization with acceptable metric distortion.

### 0.1 Keywords

surface parameterization, geometry processing,triangular mesh, mesh unfolding

## 1 INTRODUCTION

Surface parameterization represents a main topic in geometry processing and computer graphics fields. It is defined as a one-to-one mapping between a surface and typically a 2D plane, where geometrical tasks can be carried out more efficiently. The most important application of surface parameterization are texture mapping, texture synthesis, re-meshing, and morphing. In order to unfold a surface to the plane, it must have a disk topology; for a closed surface this requirement implies cutting it into a single or multiple disk topology charts. Cutting can result in visual artifacts due to the discontinuities across the boundaries of the charts. To this extent, methods for *global parameterization* of triangulated surfaces have been proposed. Within this framework, the global parameterization of a surface with disk topology can be defined as a homeomorphism from the surface to a subset of the plane, such that the discrete Gaussian curvature, i.e. the difference between $2\pi$ and the incident triangles' sum of angles at a vertex, is zero everywhere except for a few vertices called *cone singularities*. These can be thought of distortion absorbers, being chosen as vertices of the mesh where large area distortion can be predicted prior to the actual parameterization, [Kha05].



Several approaches based on metric scaling have been proposed in the past to address global parameterization [Jin08, Yan09]. However, these methods mostly rely on non-linear solvers and are hence computationally expensive. Linearized approximations, although computationally attractive, are imprecise (the target metric is only approximated and therefore is not guaranteed to be flat). More importantly, the planar coordinates of the surface vertices (the actual output) are not the variables that are optimized by this family of techniques. We will refer to those as *indirect methods* [Ben08]. Indeed

they focus on the surface metric, i.e. the edge lengths, and later reconstruct the planar coordinates in a post-process. However, this reconstruction post-process may be computationally expensive and, more importantly, numerically challenging. This paper addresses these two issues by presenting a global parameterization technique which is fast by employing linear solvers, which minimizes angular distortion through imposed conformality, and reduces the area distortion through the use of cone singularities. Our method is *simple* and *direct*. It directly solves for the 2D coordinates. Thus, it does not suffer from numerical instabilities due to angle-to-uv or scaling factors-to-uv conversions, as found with indirect approaches, [She06]. In contrast to more computationally expensive techniques based on non-linear solvers, the computational speed of our approach makes it a good candidate for interactive applications, such as user-driven parameterization improvement for instance, where the users could interactively adjust the number and locations of the cones.

**Contributions**

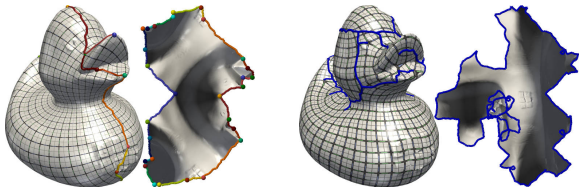
This paper makes the following new contributions:

1. **A fast and robust global parameterization algorithm:** Our method is direct (hence robust), non-iterative and only relies on the solving of at most two linear systems.
2. **Automatic curvature prescription:** Given a list of cone singularities, we present a fast algorithm to automatically evaluate relevant curvature prescriptions at the cone singularities.

The next section presents related work. Next, we introduce the method and its preliminaries. Sec. 4 and 5 present the proposed global parameterization algorithm in detail. The modeling of the linear systems is described in Sec. 6, while experimental results are reported in Sec. 7. To demonstrate the utility and versatility of our technique, we present its application to seamless texturing in Sec. 8 and finally, Sec. 9 concludes the paper.

## 2 RELATED WORK

In the following, we will only focus on surface parameterization techniques that are related to our work. We refer the reader to survey articles [Flo05, She06] for further reading. Most existing parameterization methods focus on *conformal* parameterizations (where angle distortion is minimized). Several methods [Des02, Lev02, Liu08, Ray03] focus on parameterizing surfaces of disk topology while reducing angular distortion. These methods employ linear solvers for the minimization of energy functions (that are discrete analogous to Laplace and Cauchy-Riemann equations) defined in terms of the 2D coordinates of the vertices in the mesh. These are therefore *direct* methods. They allow a free boundary setting but pin two vertices to avoid a non-trivial solution (a more recent approach removes such necessity through a spectral embedding [Mul08]).



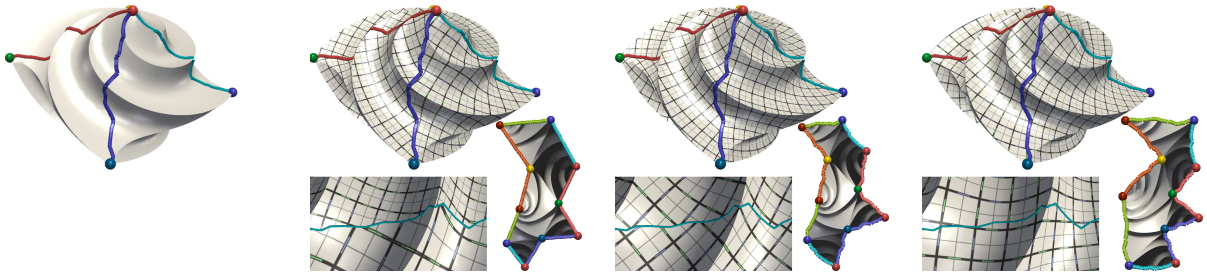
(a) *LSGP* (b) *MIQ*

Figure 1: Comparison between our approach (a) and Mixed-Integer Quadrangulation (MIQ) [Bom09] (b). On this example, the MIQ approach generates many boundary self-intersections, see the unfolded blue boundary.

*Indirect* methods [She05, She00, Zay07] aim at minimizing the difference between the initial angles of the 3D mesh and the final ones. The methods are computationally expensive (for advances see linearized version [Zay07]) and suffer from numerical instability when converting the obtained angles to actual 2D coordinates.

Indirect global parameterization methods [Ben08, Spr08, Kha05] determine the necessary edge lengths to parameterize the mesh to the plane before cutting it along a set of cut-paths to disk topology. Moreover, they also make use of cone singularities to absorb the curvature (i.e. the angle deficits), resulting in a global parameterization where the scaling of the surface is continuous across each cut-path.

While expensive non-linear solvers are usually employed [Jin08, Kha06, Spr08, Yan09], Ben Chen et al. [Ben08] approximate the solution through a Finite Element discretization of the Poisson equation, yielding better computational complexity at the expense of metric accuracy. Some methods [Spr08, Myl12, Myl13] additionally provide the possibility of obtaining seamless parameterizations by iteratively quantizing the cone angle deficits to multiples of $\pi/2$ and rectifying the cone positions to integer locations. Myles and Zorin [Myl12] compute seamless parameterizations, by employing linear solvers with linear constraints in an iterative manner for the first two steps of their algorithm (cone detection and curvature prescription). However, the last step consists in optimizing the non-linear as-rigid-as-possible (ARAP) energy function. Even though the first two steps solve linear systems, they do so in an iterative fashion and many iterations may be required. Moreover, the last step still requires the use of a time-consuming non-linear solver. Quadrangulation techniques based on structure-aligned parameterizations [Ray06, Ton06, Kal07, Bom09, Cam15, Myl14] are also related to global parameterization (for a more



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

Figure 2: Algorithm overview: (a) Given an input triangular mesh and cone singularities, the mesh is cut through the cones in order to obtain a disk topology. The cones are shown by colored spheres and conic cuts by colored cylinders. (b) Next, the mesh is conformally parameterized by concentrating the entire curvature at the cones. (c) Given the cone angles resulting from the previous step, the surface is globally parameterized. (d) Optionally, the mesh can be seamlessly parameterized.

detailed description see [Bom12]). Bommes et al. [Bom09] obtain quadrangulations by solving two mixed integer problems, one for the computation of a direction-aware cross field [Ray08, Pan12, Kno13] and one for the global parameterization, with additional similar solves in case of singularity relocation. The transition functions across cut-paths that we employ are similar to the ones used by Bommes et al. [Bom09] and Myles et al. [Myl12]. However, we formulate such constraints in a different optimization setting, which is based on faster, linear solvers.

Although the Mixed-Integer Quadrangulation (MIQ) method [Bom09] generates high quality output quadrangulations, it can yield many boundary self-intersections in the planar domain, Fig. 1 - parameterization using the MIQ method obtained by Ebke et al. [Ebk13], which may challenge their systematic usage for sub-sequent applications (such as surface cross parameterization for instance). An extension of (MIQ) [Bom09] has been proposed to address this boundary domain intersection problem [Bom13] but at the expense of increasing further computation times.

# 3 METHOD OVERVIEW

## 3.1 Preliminaries

The input surface $M$ is given as a mesh made of vertices $(V)$, edges $(E)$ and triangles $(T)$. Their number is noted with $|V|$, $|E|$ and $|T|$ respectively. The geometry of $M$ is given as the 3D coordinates of the vertices $X_v = (v_x, v_y, v_z), \forall v \in V$. The output parameterization is represented with 2D coordinates for each vertex $U_v = (u, v)$. The length of an edge is given by $e_{i,j} = \|X_{v_i} - X_{v_j}\|_2$ in 3D or $e_{i,j} = \|U_{v_i} - U_{v_j}\|_2$ in 2D. An angle in a triangle $t$ is given by: $\alpha^t_{v_i} = \arccos\left(\frac{e^2_{i,j}+e^2_{i,k}-e^2_{j,k}}{2e_{i,j}e_{i,k}}\right)$, where $v_i, v_j$ and $v_k$ are the vertices of $t$.

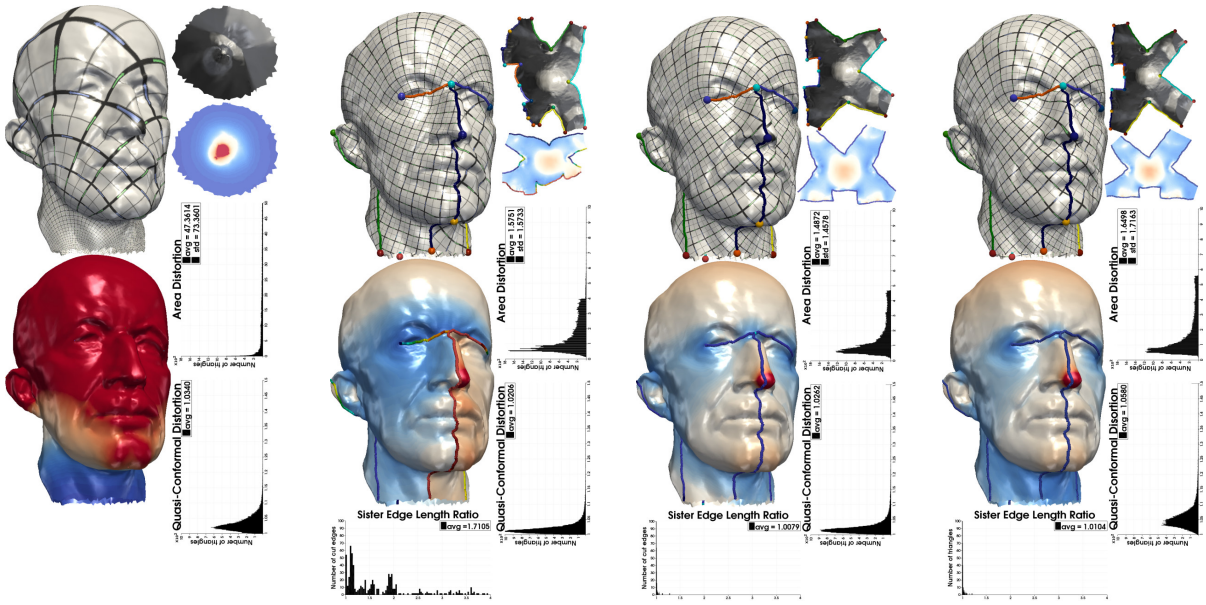
The discrete Gaussian curvature is given by $K = \left\{ k_{v_i} = \begin{cases} 2\pi - \sum_{t\in T_{v_i}}(\alpha^t_{v_i}), \text{for an interior vertex} \\ \pi - \sum_{t\in T_{v_i}}(\alpha^t_{v_i})\}, \text{for a boundary vertex} \end{cases} \right\}$, where $T_{v_i}$ represents the set of incident triangles to the vertex $v_i$.

The *Gauss-Bonnet* Theorem states that the integral of the curvature is a constant, which depends on the topology of $M$: $\sum K = 2\pi\chi$, where $\chi$ represents the Euler characteristic of $M$ ($\chi = |V| - |E| + |T|$). Given a mesh with disk topology, a *global* parameterization is a homeomorphism to a subset of the plane, such that the discrete Gaussian curvature is zero everywhere except at a set of selected vertices $C$, called *cones*.

## 3.2 Algorithm Description

Given an input triangular mesh, our algorithm first cuts the mesh open through a set of cut-paths that connect cone singularities. We call those *conic cuts*. Such singularities will *absorb* the area distortion of the parameterization, as showcased in Fig. 3. The second step consists in parameterizing the mesh, while minimizing angular distortion and imposing zero curvature everywhere except at the cones. This is achieved by introducing straightness conditions for the entire boundary, Fig. 2(b). By employing only conformality and boundary straightness conditions, the two sides of a conic cut might have different lengths in the plane, Fig. 2(b). To ensure that the two resulting sides of a conic cut are scaled similarly, we additionally enforce rotations and translations between each side of a conic cut, Fig. 2(c) note the continuity of the distortion across the cuts.

The rotation angles are either provided by the user or detected as the angles between the conic cuts in the parameterization that resulted from the previous step, i.e. the curvature prescription..




(a) *LSCM [Lev02]* (b) *LSCM [Lev02], conic cuts* (c) *Our approach* (d) *Our approach - Seamless*

Figure 3: Comparison between Least-Squares Conformal Maps (LSCM) [Lev02] without (a) and with conic cuts (b) and our approach (c), (d). From top to bottom: *(i)* textured surface with cones (colored spheres) and cuts (colored cylinders), as well as planar unfolding, *(ii)* area distortion (color map: blue (0) to red (10)) and sister edge length distortion on the cuts (rainbow color map), *(iii)* histograms of sister edge length, quasi-conformal and area distortions (ideal values: 1). Introducing cones (b) drastically reduces area distortion, while minimizing sister edge length distortion (c) yields a global parameterization . By imposing positional and rotational constraints on cones, and respectively on conic cuts, our approach can be used to generate seamless parameterizations (d).

# 4 GLOBAL PARAMETERIZATION WITH ROTATIONAL CONSTRAINTS

Given a set of cone singularities and their corresponding curvature prescriptions (either provided by the user or computed automatically, see Sec. 5), a global parameterization can be defined as an angle preserving homeomorphism from the input triangular surface to a subset of a plane, such that the discrete curvature is zero everywhere except at the cones. In this section, we present an algorithm that computes such a mapping, by minimizing angular distortion and penalizing the deviation from the target curvature in the least-squares sense. First, the surface is cut open along *conic cuts* to become homeomorphic to a disk. Second, the surface is unfolded and the target curvature is enforced by imposing affine transition functions across conic cuts.

## 4.1 Mesh Cutting

To be unfolded to the plane, we require the input surface to have a disk topology. For surfaces with a sphere topology, this can be obtained by introducing a boundary component, by cutting the mesh along the shortest paths that connect the cones. We detail this process hereafter. Variants of this strategy can be derived for surfaces with different genus.

The shortest path between each possible pair of cones is first computed with Dijkstra's algorithm. Next, a minimum spanning tree is constructed on a graph where the nodes denote the cones and where each edge is weighted by the geodesic distance between its cones (i.e. the length of their shortest paths). The edges of the spanning tree then correspond to the shortest paths along which the surface is cut open and that we call *conic cuts*. The valence of a cone corresponds to the number of conic cuts incident to it. The actual cutting process involves the duplication of all the surface edges found on the paths. Given an edge initially present on a shortest path, its copy is called its *sister edge*. Similarly, the copy of a conic cut is called its *sister conic cut*. Throughout the paper, sister conic cuts will be represented by curves with matching colors (see Fig. 2 for instance). Note that after the cutting, a cone may have a high valence. Also, similarly to Springborn et al. [Spr08], each boundary component is treated as a cone.

## 4.2 Least-Squares Conformal Maps with Rotational Constraints

Once the mesh is cut open, we unfold it to the plane with a new algorithm that minimizes angular distortion and penalizes the deviation from the target curvature.

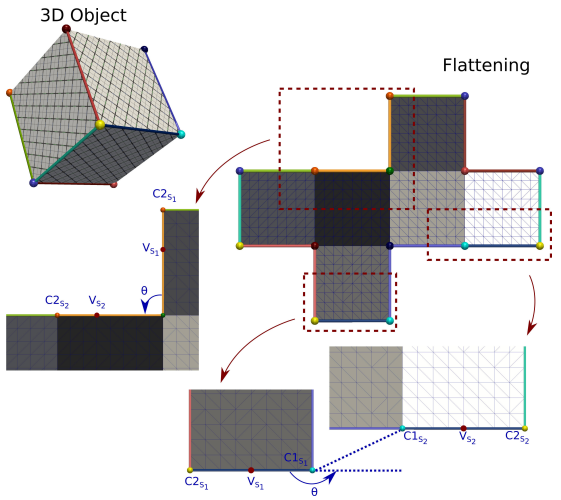



Figure 4: Rotational transformation relations between simple and complex sister conic cuts of a cube.

Our approach relies on the conformality criterion introduced by Lévy et al. [Lev02], as it enables a direct and fast optimization. We impose similar conditions as Aigerman et al. [Aig15], but in contrast to them we are not restricted to four cone configurations, allowing a general framework. Given some prescribed target curvatures for each cone, our approach consists in imposing this angle deficit by constraining combinations of translations and rotations between sister conic cuts, which is achieved as follows. Sister conic cuts can be classified into two categories:

1. *Simple conic cuts* - cuts connected to a cone of valence 1 (see the orange cuts connected to the green cone of valence 1 in Fig. 4);
2. *Complex conic cuts* - cuts not connected to a cone of valence 1 (see the dark blue cut connected in Fig. 4).

Note that simple sister cuts will be adjacent in the plane while complex ones will not. Given a cone of valence 1, we enforce its prescribed angle deficit $\theta$ by constraining its incident sister cuts to be related by a rotation of angle $\theta$ (Fig. 4, left inset zoom). For complex sister cuts, we first translate them to the origin (translation $T2$), apply the required rotation of angle $\theta$ (rotation $R$) and translate them back to their original location (translation $T1$):

$$\begin{bmatrix} V_{s_2u} \\ V_{s_2v} \\ 1 \end{bmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & C1_{s_2u} \\ 0 & 1 & C1_{s_2v} \\ 0 & 0 & 1 \end{pmatrix}}_{T1} \cdot \underbrace{\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{R} \cdot \underbrace{\begin{pmatrix} 1 & 0 & -C1_{s_1u} \\ 0 & 1 & -C1_{s_1v} \\ 0 & 0 & 1 \end{pmatrix}}_{T2} \begin{bmatrix} V_{s_1u} \\ V_{s_1v} \\ 1 \end{bmatrix} \tag{1}$$

where $(V_{s_1u}, V_{s_1v})$ and $(V_{s_2u}, V_{s_2v})$ are the unknown $(u,v)$ coordinates of the cut vertex $V_{s_1}$ and its sister cut vertex $V_{s_2}$ (Fig. 4), and where $(C1_{s_1u}, C1_{s_1v})$ and $(C1_{s_2u}, C1_{s_2v})$ are the unknown $(u,v)$ coordinates of the cone $C1$ on the conic cut $s_1$ and of its duplicate on the sister conic cut $s_2$ (Fig. 4). For simple conic cuts, the latter two cones will coincide (the cone being of valence one). Therefore, for each vertex along a conic cut, we add the following two equations to the least-squares conformal map system:

$$\begin{aligned} V_{s_2u} &= V_{s_1u} \cdot \cos(\theta) - V_{s_1v} \cdot \sin(\theta) - C1_{s_1u} \cdot \cos(\theta) \\ &\quad + C1_{s_1v} \cdot \sin(\theta) + C1_{s_2u} \\ V_{s_2v} &= V_{s_1u} \cdot \sin(\theta) + V_{s_1v} \cdot \cos(\theta) - C1_{s_1u} \cdot \sin(\theta) \\ &\quad - C1_{s_1v} \cdot \cos(\theta) + C1_{s_2v} \end{aligned} \tag{2}$$

## 5 CURVATURE PRESCRIPTION ESTIMATION BY STRAIGHTNESS CONSTRAINTS

So far, we assumed that the target curvatures, i.e. the corresponding target $\theta$ angles, were provided by the user. We describe in this section a new, fast algorithm for the automatic evaluation of relevant curvature prescriptions for a set of input cone singularities. The cone singularities can be either user-provided or automatically extracted with an existing technique ([Ben08, Spr08, Myl12]). The key idea of our algorithm is to unfold the input surface while minimizing in the least-squares sense angle distortion as well as the deviation from zero curvature, everywhere except at the cones. With this strategy, cone angles will self-adjust to provide a good balance between cone curvature absorption and angular distortion. This procedure can be interpreted as a redistribution of the surface curvature onto the cones in a least-squares sense. As described below, an appealing aspect of this method is that it only requires a single linear solving. Hence, it is very efficient in terms of computation time. To penalize the deviation from zero curvature, we enforce straightness constraints. First, for each vertex located on a conic cut, we evaluate its (3D) arc-length parameterization along the conic cut. This parameterization will be used as barycentric coordinates to enforce the alignment of the conic cut in 2D, as follows. We denote the original edge lengths vectors along a conic cut as $L_{CP} = [e_{V_{C_1},V_s^1}, ..., e_{V_s^i,V_s^{i+1}}, ..]$, Fig. 5, the total conic cut lengths as $L_{CP}^{Tot} = \sum_{e \in CP} L_{CP}[e]$; the relative edge lengths will be: $r_{CP}[e] = [\frac{L_{CP}[e]}{L_{CP}^{Tot}}, \forall e \in CP$.

We calculate the cumulative sum for the ratio vector and obtain: $R_{CP} = [0, r_{CP}^1, r_{CP}^1 + r_{CP}^2, ..., 1]$.

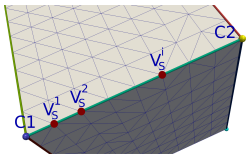


Figure 5: Conic cut.

To enforce the straightness of the conic cuts in the plane, we impose that each vertex along a cut is placed at a location which is dictated by the linear interpolation between the ending points of the cut, with factors $r^i_{CP}$. This yields two new equations that we add to the least-squares conformal map system:

$$\begin{aligned} V^i_{s\,u} &= C1_u \cdot (1 - r^i_{CP}) + C2_u \cdot (r^i_{CP}) \\ V^i_{s\,v} &= C1_v \cdot (1 - r^i_{CP}) + C2_v \cdot (r^i_{CP}) \end{aligned} \tag{3}$$

where $(V^i_{s\,u}, V^i_{s\,v})$, $(C1_u, C1_v)$ and $(C2_u, C2_v)$ stand for the unknown $(u,v)$ coordinates of the cut vertex $V_s$ and the cones $C1$ and $C2$ respectively, Fig. 5. The result of this least-squares solution is illustrated in Fig. 2(b), where the conic cuts have been straightened in the plane. From there, our algorithm collects for each cone

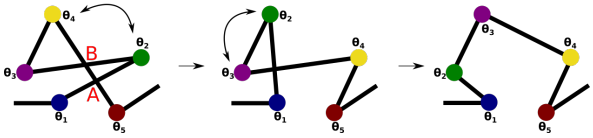


Figure 6: Possible cut intersections are recursively resolved by cone swapping. Here, the intersection $A$ between segments $[\theta_1, \theta_2]$ and $[\theta_4, \theta_5]$ is first solved instead of intersection $B$ because it counts the most intersections between its two extremities $\theta_1$ and $\theta_5$.

the angle between its incident conic cuts and provides this value as a curvature prescription for the rest of the pipeline, Fig. 2(c). This approach is non-iterative and simple to implement, requiring only solving one linear system. By applying these straightness conditions for all paths, it is possible that such paths intersect. If such a configuration is encountered, we simply recursively swap the positions of their extremity cones. In particular, this swapping procedure processes intersections in decreasing order of the number of remaining intersections between their path extremities, Fig. 6.

## 6 FORMULATING THE LINEAR SYSTEM

In this section, we detail how the equations discussed in the previous sections can be integrated in the least-squares conformal map system. The least-squares conformal map (LSCM) method [Lev02] defines the conformality of the mapping in terms of its gradients: the gradient vectors inside a triangle should be orthogonal and have the same norm. Thus, the authors propose the minimization of the following energy $E_{LSCM}$:

$$E_{LSCM} = \sum_{T_j \subset T} A_{T_j} \left\| \nabla v - \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \nabla u \right\|^2 \tag{4}$$

where $A_{T_j}$ represents the area of a triangle $T_j$ and $\nabla u$ and $\nabla v$ stand for the gradient of the $(u,v)$ coordinates within the triangle $T_j = \{p_1, p_2, p_3\}$:

$$\begin{aligned} \nabla u = (&X_{p_1} \cdot (v_{p_2} - v_{p_3}) + X_{p_2} \cdot (v_{p_3} - v_{p_1}) \\ &+ X_{p_3} \cdot (v_{p_1} - v_{p_2}))/(2 \cdot A_{T_j}) \end{aligned} \tag{5}$$

Similarly, for $\nabla v$. In order to obtain a non-trivial solution, the authors fix the 2D coordinates of two vertices $(V_p)$, leaving the rest of the vertices evolve freely $(V_f)$, $|V_p| + |V_f| = |V|$. The objective function will have the form $E(\mathbf{x}) = \|\mathbf{A} \cdot \mathbf{x} - \mathbf{b}\|^2$, where $\mathbf{x}$ represents the vector of 2D coordinates of the free vertices of the mesh ($\mathbf{x} \in \mathbb{R}^{2|V_f|}$), $A$ is a sparse matrix containing the conformality conditions ($\mathbf{A} \in \mathbb{R}^{2|T| \times 2|V_f|}$). $\mathbf{b} \in \mathbb{R}^{2|T|}$ is the vector that introduces the 2D coordinates of the fixed vertices into the system. Considering that the system is defined in terms of 2D coordinates, other positional constraints can be easily added to it, the result being a trade-off between the conformality of the mapping and the imposed constraints. By adding the rotation equations (2) that impose affine transformations for sister cuts (Fig. 4), the minimization energy will become: $E_{LSGP} = E_{LSCM} + E_{Rot}$, where:

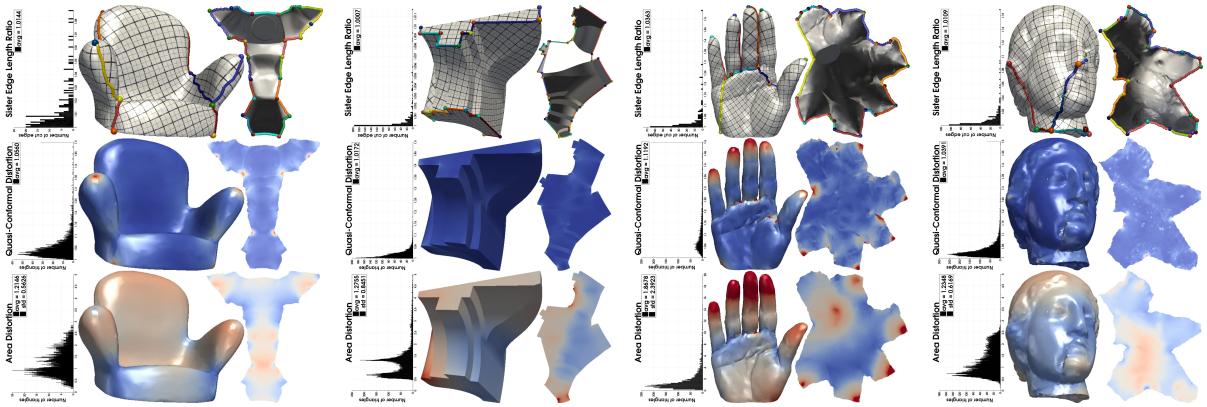
$$E_{Rot} = \sum_{CP_i \subset CP} \left( \sum_{V \subset CP_i} \left\| U_{V_{S_2}} - (T1 \cdot R \cdot T2)\, U_{V_{S_1}} \right\|^2 \right) \tag{6}$$

The number of additional equations will be equal to the number of duplicated vertices $V_d$ from each conic cut $CP_i$ multiplied by two (one equation for each of the two planar coordinates - $u$ and $v$). Therefore in the linear system, the number of equations will increase ($\mathbf{A} \in \mathbb{R}^{(2|T|+2|V_d|) \times 2|V_f|}$), while the number of variables remains the same.

In the case of straightness conditions (Sec. 5), the employed minimization energy is: $E_{Curvature\_Precription} = E_{LSCM} + E_{Str}$, where:

$$E_{Str} = \sum_{P_i \subset P} \left( \sum_{V_j \subset P_i} \left\| U_{V_j} - U_{C1} \cdot \left(1 - r^j_{CP}\right) - U_{C2} \cdot (r^i_{CP}) \right\|^2 \right) \tag{7}$$

where $P$ represents the entire set of paths resulted after the cutting. Therefore the total number of equations will be $2|T|$ to which we add $2\left(\sum_{P_i \subset P}(|P_i| - 2)\right)$ straightness equations, corresponding to the total number of vertices along paths, except their ending points; which is equal to $(|V_{bdry}| - 2|P|)$, where $V_{bdry}$ represents the total number of boundary vertices after the mesh has been cut. To remove the need for positional constraints of at least two vertices as in LSCM [Lev02], we employ the spectral approach described by Mullen et al. [Mul08].



(a) *Armchair* (b) *Fandisk* (c) *Hand* (d) *Hygeia*

Figure 7: Global parameterization examples obtained with our algorithm. From top to bottom: *(i)* textured surface with cones and cuts (with planar domain inset), *(ii)* quasi-conformal distortion, *(iii)* area distortion. Inset histograms on the left indicate, from top to bottom: sister edge length, quasi-conformal and area distortions (ideal values: 1). Our approach yields global conformal parameterizations with low area distortion.

## 7 EXPERIMENTS

Experiments were performed with a C++ implementation of our approach (using Eigen, Spectra and Boost libraries), on a laptop with a 2.50GHz i7-4710HQ CPU. Our test data-sets were taken from the AIM@SHAPE repository, [AIM].

### 7.1 Quality Estimations

We evaluated the quality of our approach with respect to the following quantitative measures. They should ideally be all equal to 1.

1. *Quasi-conformal distortion [Kha06]:* ratio between the largest and smallest eigenvalues of the metric tensor of the parameterization. This indicates a violation of the conformality condition. The color code map depicted in Fig. 7 and Fig. 9 ranges from blue (1) to red (1.5).

2. *Area distortion:* ratio between the normalized area of a triangle in 3D and in 2D (the normalized area refers to the proportion between the area of a triangle and the total area of the mesh). This indicates how much the surface needs to be stretched to be unfolded. The color code map depicted in Fig. 7 and Fig. 9 ranges from blue (0) to red (5).

3. *$10^{th}$ and $90^{th}$ Percentile of Area distortion:* area distortion values below which (and above which) are located the top 10% triangles that have been scaled down (and scaled up respectively) the most after parameterization.

4. *Sister edge length distortion*: ratio between the planar lengths of a conic cut edge and its sister's. This indicates the violation of the continuity of the global parameterization.

5. *L2Stretch:* measure of distance preservation, computed as in [San01].

Tab. 1 reports a comparison, with respect to these measures, between our technique and Least-Squares Conformal Maps (LSCM) [Lev02], to which conic cuts have been applied in a pre-process (as in Fig. 3(b)), for the sake of a fair comparison. Although slightly higher than those of LSCM, the quasi-conformal distortion measures obtained by our approach are reasonable, Fig. 7: in all our experiments, the worst quasi-conformal distortion is 1.119, Fig. 7(c). For all the remaining criteria - area distortion, sister edge length distortion, L2stretch - our approach outperforms LSCM with conic cuts for all surface examples but one. Although our approach balances conformality for globality, it still produces a quasi-conformal distortion that is on par with Least-Squares Conformal Maps (LSCM), while improving on the area distortion, Fig. 3. The sister edge length distortion is very close to 1.0 on all models, demonstrating the good globality of our parameterization. This is further exemplified in Fig. 3(c), where the area distortion is indeed continuous across the conic cuts (in contrast to LSCM with conic cuts Fig. 3(b)).

### 7.2 Time Requirement

Tab. 2 presents the running times for the different steps of our approach on the surfaces shown in the paper. This table also presents the total runtime of our approach and the runtime of the LSCM approach [Lev02] with conic cuts, including mesh cutting, as well as the speedup. As illustrated, the most expensive steps of our approach are the linear solvers for the curvature prescription and the final unfolding. Note that even combined, these two steps are still faster than the original LSCM approach. We suspect this performance

| Model | $\|T\|$ | $\|C\|$ | Distortion | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Quasi Conformal | | Areal | | $10^{th}$ Percentile Areal | | $90^{th}$ Percentile Areal | | Sister Edge Length | | L2Stretch | |
| | | | Our Approach | LSCM [Lev02] | Our Approach | LSCM [Lev02] | Our Approach | LSCM [Lev02] | Our Approach | LSCM [Lev02] | Our Approach | LSCM [Lev02] | Our Approach | LSCM [Lev02] |
| Octa-flower (Fig. 2) | 16K | 6 | 1.016 | 1.016 | **0.965** | 1.089 | **0.808** | 0.661 | **1.166** | 1.428 | **1.002** | 1.294 | **1.010** | 1.063 |
| Planck (Fig. 3) | 47K | 8 | 1.026 | **1.021** | **1.488** | 1.575 | 0.516 | **0.525** | 3.370 | **3.295** | **1.007** | 1.711 | 1.273 | 1.273 |
| Fandisk (Fig. 7(b)) | 13K | 25 | **1.017** | 1.018 | **1.276** | 3.601 | **0.606** | 0.536 | **2.068** | 8.748 | **1.001** | 2.609 | **1.132** | 1.926 |
| Armchair (Fig. 7(a)) | 5K | 12 | 1.056 | **1.042** | **1.215** | 1.560 | **0.633** | 0.525 | **1.961** | 3.156 | **1.014** | 1.626 | **1.099** | 1.252 |
| Hand (Fig. 7(c)) | 5K | 11 | 1.119 | **1.085** | **1.868** | 4.166 | **0.451** | 0.350 | **4.134** | 8.445 | **1.036** | 1.365 | **1.455** | 2.010 |
| Hygeia (Fig. 7(d)) | 16.5K | 14 | 1.039 | **1.037** | **1.235** | 1.521 | **0.618** | 0.462 | **2.123** | 3.013 | **1.011** | 1.448 | **1.111** | 1.247 |

Table 1: Comparison of distortion measures between our approach and Least Squares Conformal Maps [Lev02] with conic cuts. For each criterion (ideal values: 1), the best measure of the two techniques is displayed in bold.

| Model | $\|T\|$ | $\|C\|$ | Runtime [s] | | | | | | | Speedup of our approach vs. LSCM [Lev02] |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Mesh Cutting (Sec. 4.1) | Curvature Prescription (Sec. 5) | | Rotationally Constrained Unfolding (Sec. 4.2) | | Our approach Total | LSCM [Lev02] Total | |
| | | | | Setup | Solve | Setup | Solve | | | |
| Octa-flower (Fig. 2) | 16K | 6 | 0.144 | 0.051 | 0.028 | 0.079 | 0.014 | 0.316 | 2.085 | 6.6 |
| Planck (Fig. 3) | 47K | 8 | 1.096 | 0.214 | 0.130 | 0.477 | 0.091 | 2.008 | 16.760 | 8.35 |
| Fandisk (Fig. 7(b)) | 13K | 25 | 0.010 | 0.053 | 0.032 | 0.088 | 0.029 | 0.212 | 1.551 | 7.32 |
| Armchair (Fig. 7(a)) | 5K | 12 | 0.017 | 0.014 | 0.007 | 0.019 | 0.007 | 0.064 | 0.324 | 5.06 |
| Hand (Fig. 7(c)) | 5K | 11 | 0.017 | 0.015 | 0.008 | 0.017 | 0.008 | 0.065 | 0.235 | 3.62 |
| Hygeia (Fig. 7(d)) | 16.5K | 14 | 0.160 | 0.067 | 0.033 | 0.090 | 0.024 | 0.374 | 2.399 | 6.41 |

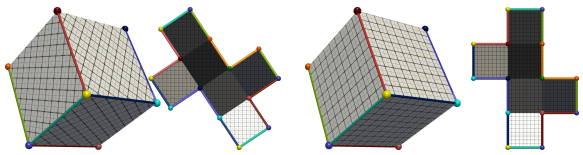
Table 2: Computation times for each step of our approach in seconds.

gain is due to the fact that the original LSCM method uses an indirect method (Conjugate Gradients) to solve the least-squares problem, while we employ the spectral method described by Mullen et al. [Mul08]. The average speedup of our method compared to LSCM, [Lev02] is 6.23.

In comparison to the approach by Myles and Zorin [Myl12], who report a timing of 12.55 seconds for the Fandisk mesh for only the first two steps of their algorithm (cone detection and curvature prescription), our method requires only 0.328 seconds overall.

## 8 SEAMLESS TEXTURES APPLICATION

As described previously, our approach computes global parameterizations, where area distortion is continuous across conic cuts, Fig. 8(a). For example, this facilitates texture design. Artists want to paint across cuts without noticing distortions. However, for specific texturing tasks such as procedural texturing with periodic patterns, it is additionally beneficial to enforce planar coordinate alignment across the cuts, to guarantee the alignment of the periodic pattern. Such a parameterization is called *seamless* and it is illustrated on a simple cube in Fig. 8(b), where the repeating checker board pattern is indeed well aligned across conic cuts. Seamless parameterizations are also useful for re-meshing as pure quadrangulations can readily be extracted from them. Seamless texturing requires the usage of specific transition functions across conic cuts: translations and rotations by multiples of $\pi/2$. Additionally, cone singularities must be located at integer texture coordinates.

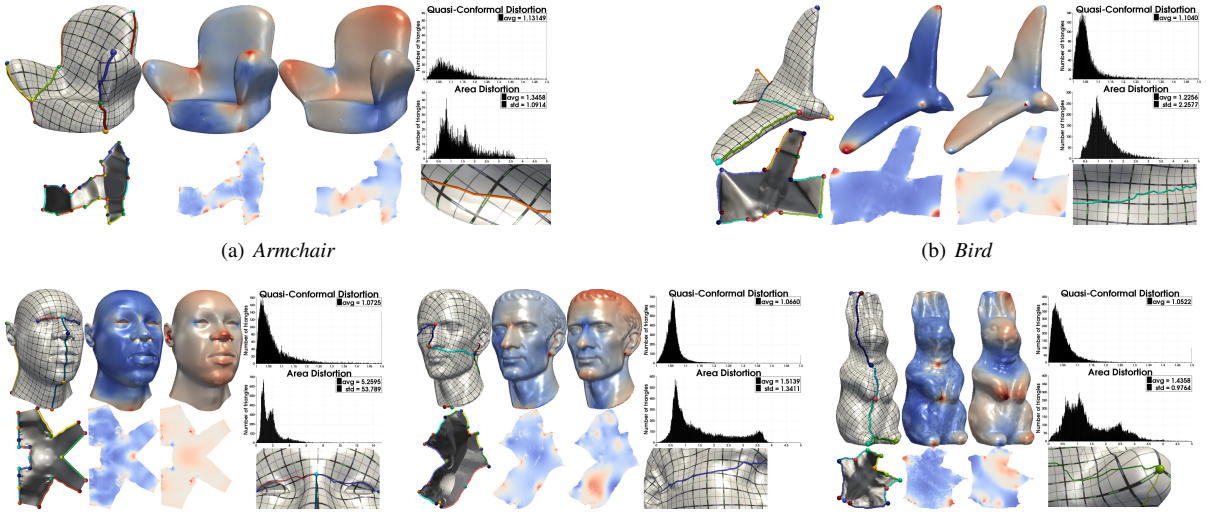


(a) *LSGP* (b) *Seamless LSGP*

Figure 8: Cube unfolding with our approach (a); our approach - seamless (b).

These two constraints can easily be integrated in our approach.

Given input cones as well as initial curvature prescriptions computed automatically by our approach, we start by rounding the curvature prescriptions to the nearest multiples of $\pi/2$ while respecting the Gauss-Bonnet theorem, similarly to Springborn et al. [Spr08]. In particular if the sum of prescribed angles is different from the allowed sum, we decrease them in descending order of their rounding error. The resulting angles are then prescribed in the reminder of the proposed algorithm. Next, we snap each cone to the nearest integer location. The resulting integer locations are then added as hard constraints to the linear system described in Sec. 6 and the rest of the algorithm is executed as is. In the extreme case where several cones are quantized to the same $(u,v)$ coordinates, we relax the seamless constraints by not pinning such cones, but rather adding them to the system as soft constraints. This case appears rarely in practice and it is either caused by the proximity of the cones in 3D, the high cone number or the low resolution of the texture space.




(a) *Armchair* (b) *Bird*

(c) *Head* (d) *Julius* (e) *Rabbit*

Figure 9: Seamless global parameterization obtained with our algorithm. From left to right: *(i)* textured surface with cones and cuts (with planar domain inset), *(ii)* quasi-conformal distortion, *(iii)* area distortion, *(iv)* quasi-conformal and area distortion histograms (ideal values: 1). Our approach yields global conformal parameterizations with low area distortion and seamless texture transitions across conic cuts (inset zooms).

Fig. 2(d), Fig. 1(a), Fig. 3(d), Fig. 8(b) and Fig. 9 provide examples of seamless global parameterizations obtained with our approach. The initial curvature prescription, before rounding, has been automatically evaluated by our method for all surfaces. As showcased in these examples, based on the integration of seamless constraints, our approach provides rapidly seamless global parameterizations with low area distortion.

## 9 CONCLUSION

We have presented a fast and efficient method for the global parameterization of triangular surfaces.

For modeling the transition functions between pairs of sister conic cuts, we introduced linear equations which account for translations and rotations with given cone angles. Also we have provided an automatic method to compute such angles. Extensive experimental results demonstrate the time efficiency of our algorithm which performs better than standard, non-global parameterization algorithms [Lev02]. The average speedup of our method compared to LSCM is 6.23. The quality of our parameterizations has been illustrated by examining accepted distortion measures. We demonstrated the interest of the computational speed of our approach in the seamless texturing application, which requires slight modifications to our algorithm.

In future work, we want to extend the seamless texturing application. In particular, the proposed method for detecting integer positions for the cones is not guaranteed to find solutions for the entire set of cones, the quantization can be partial, but we show a number of examples where the application provides satisfactory results. A more robust but still efficient quantization of the cones remains an open problem that we will address in the future. Although our procedure for the resolution of conic cut intersection in the planar domain has worked successfully in our experiments, we would like to further investigate theoretical guarantees regarding the bijective property of the maps computed by our approach. Also, since our method handles only disk or sphere topology, another future work direction lies in the extension of our algorithm to surfaces of non trivial topology, by applying loop computation algorithms [Dey08, Dey13].

## 10 REFERENCES

[Aig15] N. Aigerman and Y. Lipman. Orbifold tutte embeddings. *ACM Trans. Graph.*, 34(6):190:1–190:12, 2015.

[AIM] Digital shape workbench. AIM@SHAPE shape repository. `http://visionair.ge.imati.cnr.it/ontologies/shapes/`

[Ben08] M. Ben-Chen, C. Gotsman, and G. Bunin. Conformal flattening by curvature prescription and metric scaling. *Comput. Graph. Forum*, 27(2):449–458, 2008.

[Bom13] D. Bommes, M. Campen, H.-C. Ebke, P. Alliez, and L. Kobbelt. Integer-grid maps for reliable quad meshing. *ACM Trans. Graph.*, 32(4):98:1–98:12, 2013.

[Bom12] D. Bommes, B. Lévy, N. Pietroni, E. Puppo, C. Silva, M. Tarini, and D. Zorin. State of the art in quad meshing. In *Eurographics STARS*, pp. 159–182.

[Bom09] D. Bommes, H. Zimmer, and L. Kobbelt. Mixed-integer quadrangulation. *ACM Trans. Graph.*, 28(3):77:1–77:10, 2009.

[Cam15] M. Campen, D. Bommes, and L. Kobbelt. Quantized global parametrization. *ACM Trans. Graph.*, 34(6):192:1–192:12, 2015.

[Des02] M. Desbrun, M. Meyer, and P. Alliez. Intrinsic parameterizations of surface meshes. *Comput. Graph. Forum*, 21(3), 2002.

[Dey13] T. K. Dey, F. Fan, and Y. Wang. An efficient computation of handle and tunnel loops via reeb graphs. *ACM Trans. Graph.*, 32(4):1–10, 2013.

[Dey08] T. K. Dey, K. Li, J. Sun, and D. Cohen-Steiner. Computing geometry-aware handle and tunnel loops in 3d models. *ACM Trans. Graph.*, 27(3):45:1–45:9, 2008.

[Ebk13] H. C. Ebke, D. Bommes, M. Campen, and L. Kobbelt. QEx:Robust Quad Mesh Extraction. *ACM Trans. Graph.*, 32(6), 2013. `http://www.rwth-graphics.de/software/libQEx`

[Flo05] M. Floater and K. Hormann. Surface parameterization:a tutorial and survey. In *Adv. Multires. Geom. Mod.* Springer, 2005.

[Jin08] M. Jin, J. Kim, and X. D. Gu. *Discrete Surface Ricci Flow: Theory and Applications*, pp. 209–232. Springer, 2007.

[Kal07] F. Kalberer, M. Nieser, and K. Polthier. Quadcover – surface paramterization using branched coverings. *Comput. Graph. Forum*, 26(3):375–384, 2007.

[Kha06] L. Kharevych, B. Springborn, and P. Schröder. Discrete conformal mappings via circle patterns. *ACM Trans. Graph.*, 2006.

[Kha05] L. Kharevych, B. Springborn, and P. Schröder. Cone singularities to the rescue: Mitigating area distortion in discrete conformal maps. In *Symp. on Geom. Processing*, ACM SIGGRAPH/Eurographics , 2005.

[Kno13] F. Knöppel, K. Crane, U. Pinkall, and P. Schröder. Globally optimal direction fields. *ACM Trans. Graph.*, 32(4):1–10, 2013.

[Lev02] B. Lévy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.*, 21(3):362–371, 2002.

[Liu08] L. Liu, L. Zhang, Y. Xu, C. Gotsman, and S. Gortler. A local/global approach to mesh parameterization. In *Proc. of the Symp. on Geom. Processing*, SGP '08, pp. 1495–1504, 2008.

[Mul08] P. Mullen, Y. Tong, P. Alliez, and M. Desbrun. Spectral conformal parameterization. In *Proc. of the Symp. on Geom. Processing*, SGP '08, pp. 1487–1494, 2008.

[Myl14] A. Myles, N. Pietroni, and D. Zorin. Robust field-aligned global parametrization. *ACM Trans. Graph.*, 33(4):1–14, 2014.

[Myl13] A. Myles and D. Zorin. Controlled-distortion constrained global parametrization. *ACM Trans. Graph.*, 32(4):1–14, 2013.

[Myl12] A. Myles and D. Zorin. Global parametrization by incremental flattening. *ACM Trans. Graph.*, 31(4):109:1–109:11, 2012.

[Pan12] D. Panozzo, Y. Lipman, E. Puppo, and D. Zorin. Fields on symmetric surfaces. *ACM Trans. Graph.*, 31(4):1–12, 2012.

[Ray08] N. Ray, B. Vallet, W. C. Li, and B. Lévy. N-symmetry direction field design. *ACM Trans. Graph.*, 27(2):10:1–10:13, 2008.

[Ray06] N. Ray, W. C. Li, B. Lévy, A. Sheffer, and P. Alliez. Periodic global parameterization. *ACM Trans. Graph.*, 25(4), 2006.

[Ray03] N. Ray and B. Lévy. Hierarchical least squares conformal map. In *Pacific Graphics*, 2003.

[San01] P. V. Sander, J. Snyder, S. J. Gortler, and H. Hoppe. Texture mapping progressive meshes. ACM SIGGRAPH, 2001.

[She06] A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis.*, 2(2):105–171, 2006.

[She05] A. Sheffer, B. Lévy, M. Mogilnitsky, and A. Bogomyakov. Abf++: Fast and robust angle based flattening. *ACM Trans. Graph.*, 24(2):311–330, 2005.

[She00] A. Sheffer and E. De Sturler. Surface parameterization for meshing by triangulation flattening. In *Proc. of IMR*, 2000.

[Spr08] B. Springborn, P. Schröder, and U. Pinkall. Conformal equivalence of triangle meshes. *ACM Trans. Graph.*, 27(3), 2008.

[Ton06] Y. Tong, P. Alliez, D. Cohen-Steiner, and M. Desbrun. Designing quadrangulations with discrete harmonic forms. In *Symp. on Geom. Processing*, pp. 201–210, 2006.

[Yan09] Y. L. Yang, R. Guo, F. Luo, S. M. Hu, and X. Gu. Generalized discrete ricci flow. *Comput. Graph. Forum*, 28(7), 2009.

[Zay07] R. Zayer, B. Lévy, and H.-P. Seidel. Linear angle based parameterization. In *Symp. on Geom. Proc.*, pp. 135–141, 2007.