

# Global Parametrization by Incremental Flattening

Ashish Myles\* and Denis Zorin†  
New York University

## Abstract

Global parametrization of surfaces requires singularities (cones) to keep distortion minimal. We describe a method for finding cone locations and angles and an algorithm for global parametrization which aim to produce seamless parametrizations with low metric distortion. The idea of the method is to evolve the metric of the surface, starting with the original metric so that a growing fraction of the area of the surface is constrained to have zero Gaussian curvature; the curvature becomes gradually concentrated at a small set of vertices which become cones. We demonstrate that the resulting parametrizations have significantly lower metric distortion compared to previously proposed methods.

**CR Categories:** I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—[Geometric algorithms, languages, and systems];

**Keywords:** geometric modeling, parametrization

**Links:** [DL](#) [PDF](#)

## 1 Introduction

Global parametrization, i.e. a map from a surface to the plane, can be continuous and everywhere locally injective only for domains of disk topology; even for these domains, [Kharevych et al. 2006] observed that singularities (cones) and cuts may be necessary to reduce the distortion to an acceptable level. More generally, a global parametrization can be defined as a flat metric on the surface, with isolated cones where all Gaussian curvature is concentrated. If the surface is cut to a disk with the cut going through all cones, the metric determines a mapping to the plane up to a rigid transform.

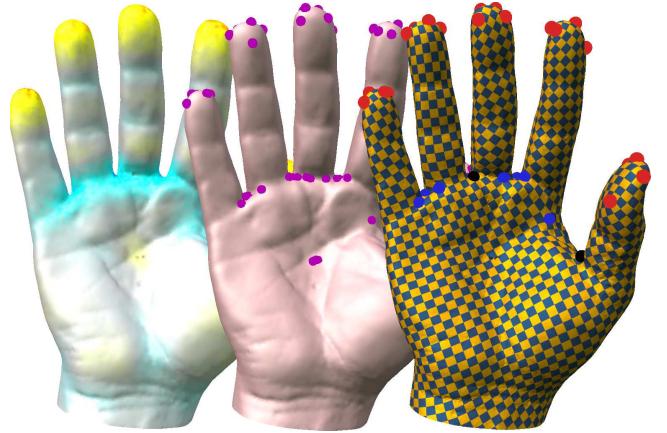
One of the common goals for surface parametrization is to minimize a measure of metric distortion. For a non-singular parametrization of a disk, this amounts to minimizing a smooth energy measuring distortion. However, if cones can be added to the parametrization, the problem becomes far more complicated: one needs to find cone *positions* and the integral Gaussian curvature (cone angle) concentrated at each cone. As shown in [Bunin 2008] for conformal maps, this problem is related to a type of inverse Poisson problem, which is generally ill-posed and expensive to solve.

We propose a method, which we call *incremental flattening*, for placing cones on the surface. The idea of the method is to evolve the metric of the surface, starting with the embedding metric so that a growing fraction of the area of the surface is constrained to have zero Gaussian curvature. The Gaussian curvature becomes

concentrated in isolated shrinking areas, which eventually give rise to cones.

In this paper we focus on *seamless* global parametrizations, for which parametric lines in the plane continue smoothly across seams (the concept is defined more precisely in Section 3). Seamless parametrizations are most valuable for applications, as they allow to generate quadrangulations and tile the surface seamlessly with texture maps. We note that our definition of seamless parametrization is adequate for generating *fine* quadrangulations and texture mapping, but additional constraints on parametric positions of cones are needed for *coarse* quadrangulations (cf. [Bommes et al. 2009] and discussion in Section 3).

A seamless parametrization has constraints on cone angles (they have to be multiples of  $\pi/2$ ) and the rotation of the parametrization gradient along noncontractible loops. These requirements substantially change the nature of the problem: if arbitrary cone angles are allowed, it is clear that the mesh parametrization can be made perfectly isometric by inserting cones at all vertices; on the other hand, with  $k\pi/2$  constraints on cone angles, having cones at all vertices is clearly suboptimal. Another consequence is that the problem becomes a mixed-integer problem which can be NP-hard (Section 5).



**Figure 1: Algorithm stages.** Left-to-right: original surface curvature; result of conformal flattening concentrating curvature at cone candidates. Seamless parameterization after iterative rounding of cone angles and collapsing nearby cones.

Our main contributions include the following:

- We describe a distortion-minimizing incremental procedure for flattening and cone angle determination; we show that it can be implemented efficiently with incremental linear solves by using a conformal map as a proxy for the distortion-minimizing map.
- We extend the *as-rigid-as-possible* (ARAP) parametrization of disk-topology surfaces to general surfaces, and demonstrate how a cross-field optimization can be used for efficient ARAP computation.
- We demonstrate that the cones determined by our procedure, followed by the generalized ARAP parametrization, yield a substantially lower parametrization distortion compared to existing methods.

\*e-mail: amyles@cs.nyu.edu

†e-mail: dzorin@cs.nyu.edu

## 2 Related work

Our work builds on ideas from four sources: (1) feature-aligned parametrization and quadrangulation, based on fitting parametrization gradients to a field, primarily [Bommes et al. 2009], but also [Ray et al. 2006; Kälberer et al. 2007]; (2) as-rigid-as-possible parametrization [Liu et al. 2008]; (3) conformal parametrization/metric computation [Jin et al. 2004; Ben-Chen et al. 2008; Springborn et al. 2008]; and (4) cross-field and N-symmetry/rotational symmetry field and connection constructions [Hertzmann and Zorin 2000; Palacios and Zhang 2007; Ray et al. 2008; Ray et al. 2009; Lai et al. 2009; Crane et al. 2010]. We discuss the relationship in Sections 5 and 6 in more detail.

We focus on the work offering automatic cone placement; broader reviews of parametrization can be found in [Hormann et al. 2007; Sheffer et al. 2006].

Many techniques based on the construction of a base domain using simplification offer good control over the number of cones, and typically result in a relatively even distribution [Eck et al. 1995; Lee et al. 1998; Khodakovsky et al. 2003; Marinov and Kobbett 2005; Daniels II et al. 2009; Daniels et al. 2009; Pietroni et al. 2009; Tarini et al. 2010]. However, the distortion is often suboptimal for the number of cones, and difficult to control. A number of methods [Gu and Yau 2003; Dong et al. 2006; Tong et al. 2006; Ben-Chen et al. 2008; Springborn et al. 2008] use global harmonic or conformal parametrizations with cones; among these, [Dong et al. 2006] and [Ben-Chen et al. 2008] and [Springborn et al. 2008] present automatic cone placement. The Morse complex constructed in [Dong et al. 2006] is not directly tied to parametrization distortion. The cone placement techniques of [Jin et al. 2004], [Ben-Chen et al. 2008] and [Springborn et al. 2008] follow a “top-down” approach, going back to [Gu et al. 2002]: e.g. for disk topology, start with a parameterization with no cones, and add cones in areas of maximal distortion. This approach works well for placing a small number of cones, but as the number grows, it become less reliable (Section 7). [Jin et al. 2008; Lai et al. 2009] parameterize a surface by flattening the metric by uniformization using *Ricci flow*, concentrating curvature at user-specified cone locations. Our metric evolution produces cone positions and curvatures automatically. [Sorkine et al. 2002] introduces discontinuities into the texture map based on maintaining bounds on distortion. Similarly to our work, the flattened areas are grown in a greedy way so that the increase in distortion is minimized at each step.

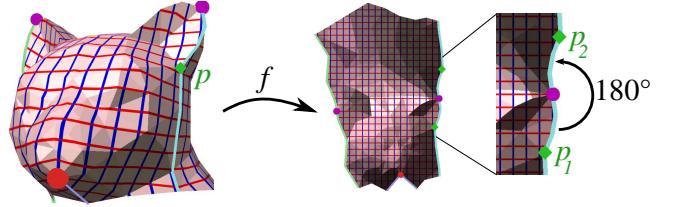
## 3 Global parametrizations and metrics

In this section, we summarize definitions and basic properties of global parametrizations and related cone metrics that we need to explain our algorithm. Many of these facts appear in the literature in different forms, cf. [Springborn et al. 2008; Ben-Chen et al. 2008; Lai et al. 2009].

A 3D mesh  $M$  can be cut to a topological disk  $M_c$ , by removing a set of simple *seam* edge chains from the surface. Away from vertices where several seam chains join, each seam point  $p$  corresponds to two points  $p_1$  and  $p_2$  in  $M_c$  (Figure 2). A piecewise linear map  $f$  from  $M_c$  to the plane is a *global parametrization* of the mesh. In general, it depends on the choice of the seam, and the points of the seam get two distinct parametric values. For a choice of coordinates on each triangle the differential of  $f$  can be represented by a  $2 \times 2$  Jacobian matrix  $J(p)$ .

A *cone metric*  $g$  on a mesh  $M$  is a metric with zero discrete Gaussian curvature everywhere except at a set  $C$  of vertices (*cones*)  $C = \{c_i\}$  with cone angles  $\alpha_i > 0$ . In a neighborhood of a cone point, the surface is isometric to a cone with cone angle  $\alpha_i$  with respect to the metric  $g$ . The Gaussian curvature at the cone  $c_i$  is the delta function  $K_i \delta(p - c_i)$ , with  $K_i = 2\pi - \alpha_i$ .

A global parametrization  $f$  defines a metric on the mesh with constant metric tensor  $g = \nabla f^T \nabla f$  on each triangle; this metric is flat



**Figure 2: Global parameterization.** Each seam edge chain on the surface (left) maps to two seam curves related by a rigid transformation on the parametric domain (middle and right) – in this case, a rotation of  $180^\circ$  about the cone. A surface point  $p$  on a seam maps to two points  $p_1$  and  $p_2$  on the domain related by the same transformation.

away from the seam. We additionally require that the metric is flat at all points of the seam, excluding the endpoints of the seam curve. Flatness of metric on seam curves implies that two images of a seam in the parametric domain are related by a rigid transformation (Figure 2). Conversely, a flat metric on  $M \setminus C$ , and a seam passing through all points of  $C$  defines a parametrization  $f$  uniquely, up to a composition with rigid transformations of the plane.

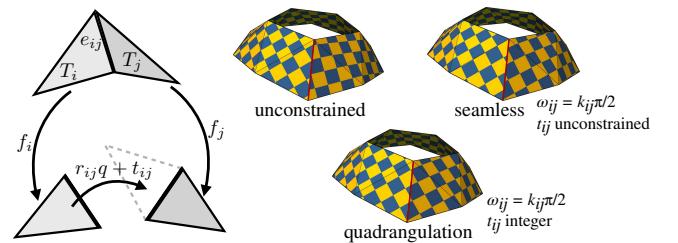
We are interested in a special type of global parametrization and cone metric (a) which does not depend on the choice of seams (up to rigid transformations), but only on the choice cone points; (b) and whose isoparametric lines continue smoothly across seams. Conditions for a parametrization to be seamless can be formulated in terms of the corresponding metric.

If the metric is flat, two important facts follow from the Gauss-Bonnet theorem:

**Seamless parametrizations.** We say that a parametrization  $f$  is *seamless* if it has the following property: if  $f_i$  is the linear parametrization of a triangle  $T_i$ , then for each point  $p$  on the shared edge of  $T_i$  and  $T_j$ , the differentials  $J_i$  and  $J_j$  of the maps  $f_i$  and  $f_j$  acting on the common edge vector  $e_{ij}$  of  $T_i$  and  $T_j$  agree up to a rotation  $r_{ij}$  by the *seam rotation angle*  $\omega_{ij} = k_{ij}\pi/2$ , where  $k_{ij}$  is an integer (Figure 3):

$$J_j e_{ij} = r_{ij} J_i e_{ij} \quad (1)$$

For edges not corresponding to seams,  $\omega_{ij} = 0$  and  $r_{ij}$  is the identity.



**Figure 3: Unconstrained vs. seamless parametrization.** Satisfying (1) across cuts results in a seamless parametrization. Additionally rounding cone parametric positions and seam translations  $t_{ij}$  makes it suitable for quadrangulation.

A quadrangulation additionally requires constraining translational differences of parametric values across seams to be integer (Figure 3). Computing a sufficiently fine quadrangulation from a seamless parametrization by rounding translations does not significantly change the distortion (cf. Figure 11). We thus restrict our attention to seamless parametrizations, while noting that minimizing distortion for very coarse mesh generation requires taking into account distortion due to cone parametrization rounding.

An invariant definition for a seamless parametrization not relying on the choice of a seam, but directly on the metric, can be given in

terms of *holonomy angles*. A metric  $g$  on the mesh can be defined as an assignment of edge lengths, satisfying triangle inequality, from which the angles of all triangles in the parametric domain can be computed.

Consider, as in Figure 4, a closed chain  $\gamma$  of  $N$  triangles  $T_m$  of the mesh, such that  $T_m$  and  $T_{m+1}$  share an edge  $e_{m,m+1}$ , where  $m+1$  is computed modulo  $N$ . Define a loop consisting of dual edges orthogonal to the mesh edges (*in the flat metric  $g$*  in Figure 4, *middle*), and the discrete geodesic curvature at the endpoints of dual edges equal to the signed angle between the normals of the dual-edge, i.e. the angle between corresponding primal edges. The *discrete holonomy angle* indicates how much the sum of triangle angles  $\beta_m$  between  $e_{m-1,m}$  and  $e_{m,m+1}$ , with signs  $d_m = \pm 1$  chosen depending on the direction of rotation, deviates from the angle sum of a closed loop in the parametric domain.

$$A_g^d(\gamma) = 2\pi - \sum_{m=1}^N d_m \beta_m. \quad (2)$$

The *holonomy* of the closed chain is the corresponding rotation. This is a discrete version of the standard continuous definition of holonomy by parallel transport of a unit vector around a loop. For a counterclockwise loop around a vertex, the holonomy angle is just the sum of angles at the vertex.

If the metric is flat, two important facts directly follow from the discrete version of the Gauss-Bonnet theorem:

**Proposition 1** (1) Any two homotopic chains in  $M \setminus C$  have the same holonomy angle.

(2) For any counterclockwise chain  $\gamma$  encircling a single cone  $c_i$ ,

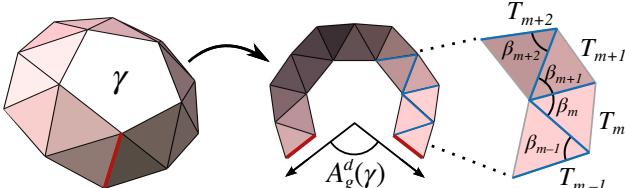
$$A_g(\gamma) = K_i.$$

As a consequence, the holonomy of a cone metric is completely described by a finite set of values: it is sufficient to define the holonomy angles on a set of representatives  $\gamma_i$  of all homotopy classes of loops on  $M \setminus C$ . This set can be chosen to include a set of homology loops  $\gamma_i^h$  in  $M$ , and a single ring of triangles  $\gamma_i^c$  around each cone  $c_i$ . We call a cone metric  $g$  *seamless*, if for all  $\gamma_i$ ,

$$A_g(\gamma_i) = \frac{k_i \pi}{2}, \quad (3)$$

where  $k_i$  is an integer.

**Remark 1.** A similar condition appeared in [Lai et al. 2009] where turning numbers of an  $N$ -RoSy field are constrained to be a multiples of  $\frac{2\pi}{N}$  by similarly constraining the holonomy angles of the related metric.



**Figure 4: Loop holonomy.** A loop  $\gamma$  unfolded isometrically to a plane with its holonomy angle  $A_g^d(\gamma)$ . Within triangle  $T_m$ , the loop  $\gamma$  turns an angle of  $\beta_m$ , computed as the angle between adjacent edges  $e_{m-1,m}$  and  $e_{m,m+1}$  shown in blue. For interior edges like the ones highlighted in blue,  $\omega_{m,m+1} = 0$  and  $r_{m,m+1}$  is the identity. Meanwhile,  $\omega_{m,m+1} = A_g^d(\gamma)$  across the red seam edge at the bottom and (4) holds.

The sum for the discrete holonomy angle of a loop can also be directly expressed in terms of the Jacobian matrices of the linear maps

$f_i$  as the sum of rotation angles  $\omega_{m,m+1}$  between  $J_m e_{m,m+1}$  and  $J_{m+1} e_{m,m+1}$  for all edges in a chain of triangles:

$$A_g^d(\gamma) = \sum_{m=1}^N \omega_{m,m+1}. \quad (4)$$

As a consequence, while individual rotations depend on the choice of seams and maps  $f_i$ , their sums along closed loops do not.

Recall that we assume that  $f_i$  agree at non-seam edges, so  $\omega_{m,m+1}$  is zero there with  $r_{m,m+1}$  being identity.

**Proposition 2** A cone metric  $g$  is seamless if and only if any global parametrization  $f$  with this metric  $g$  is seamless.

For a seamless parametrization, rotation angles  $\omega_{m,m+1}$  on seam edges, are uniquely determined by holonomy angles  $A_g^d(\gamma_i)$  for a complete set of loops of triangles representing homotopy classes in  $M \setminus C$ , and conversely, rotation angles on seam edges are determined by the holonomy angles. The relationship between seam rotation angles and holonomy angles is linear.

The proposition is proved in the appendix.

## 4 Overview of the method

**As-rigid-as-possible energy for a metric.** As our method aims to minimize parametrization distortion, we need to choose a distortion measure. We use the *as-rigid-as-possible* (ARAP) energy:

$$E^{ARAP} = \frac{1}{2} \sum_T \min_{R \in SO(2)} \|\nabla f_T - R\|^2 A_T \quad (5)$$

where the summation is over all triangles of the mesh. As it is shown in [Liu et al. 2008], the integrand can be expressed directly in terms of *signed* singular values  $\sigma_1$  and  $\sigma_2$  of  $\nabla f$ , as  $(\sigma_1 - 1)^2 + (\sigma_2 - 1)^2$ . If  $f$  preserves orientations,  $\sigma_1^2$  and  $\sigma_2^2$  are the singular values of the metric  $g = \nabla f^T \nabla f$  associated with  $f$ . Assuming we choose local isometric coordinates on the surface (e.g., an orthonormal coordinate system on each triangle), the initial metric  $g_0 = I$ , and

$$E(g_0, g) = \frac{1}{2} \sum_T (\sigma_1 - 1)^2 + (\sigma_2 - 1)^2 A_T \quad (6)$$

Ideally, we want to compute a metric  $g$  globally minimizing the distortion energy subject to the constraint (3), with unknown integers  $k_i$ , unknown number of cones and unknown cone positions  $c_i$ . This is a difficult problem: in Section 5 we discuss in more detail why it is likely to be NP-hard. As a consequence, we design a heuristic approach to find a low-distortion parametrization, and validate it experimentally.

Our approach consists of three main steps.

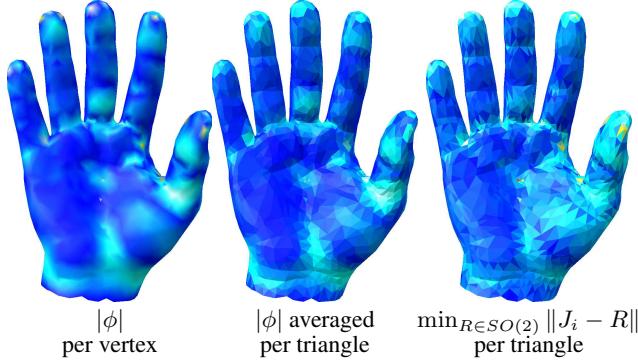
(1) *Flattening* (Section 5) determines cone positions (but not angles). This is done by a greedy approach, which starts from low-curvature areas of the surface and gradually concentrates all curvature at isolated cone points, continuously evolving the metric.

(2) *Rounding* sequentially rounds the angles for cones so that the increase in the energy is minimized at each step. Holonomy constraints for homology loops are fixed by simultaneous rounding, with possible additional optimization at the ARAP optimization step.

(3) *Global ARAP parametrization* with fixed holonomy angles completes the process. We use a cross-field initialization for ARAP, for a substantial speedup (Section 6).

Performing nonlinear optimization would be prohibitively expensive for flattening and rounding, so we use a highly restricted class of metrics (metrics conformally equivalent to  $g_0$ ) as a proxy for optimization over all possible metrics. For this class of metric, the optimization problem that needs to be solved at every step is quadratic

(Section 5). The conformal map metric distortion approximates the minimum of the ARAP energy as long as the maps remain close to isometric (Figure 5).



**Figure 5: Comparison of curvature distortion measures.** Even with the majority of surface vertices constrained to be flat, the conformal distortion  $|\phi|$  of the metric (introduced in Section 5) serves as a good proxy for the ARAP distortion  $\min_{R \in SO(2)} \|J_i - R\|$  of the resulting parameterization. Distortion is visualized varying from 0 (blue) to 0.5 (red).

## 5 Flattening and rounding

Both flattening and rounding perform a constrained minimization of  $E(g_0, g)$ . Figure 7 shows the evolution of the metric distortion during the flattening and rounding phase.

**Flattening.** Our goal is to define a new metric  $g$ , for which the Gaussian curvature  $K[g]$  is zero almost everywhere, except at a set of isolated *cone candidates*  $c_i$ , which are assigned concentrated curvatures  $K_i$ . At the flattening stage we do *not* require the metric to be seamless, and place no constraints on the holonomy angles. We start with areas of low curvature and gradually increase the area where the curvature has to be zero, recomputing the metric at every step. As the constrained areas are expanded, the curvature concentrates in the unconstrained areas (Figure 6). Once only isolated points remain unconstrained, the process stops. The flattening procedure can be summarized as follows:

```

1: Flattening
2:  $\epsilon := \epsilon_0$  {Initialize curvature threshold}
3:  $g := g_0$  {Initialize metric}
4: while  $M \setminus \Omega$  contains non-isolated vertices do
5:    $\Omega :=$  set of all vertices  $p$  on  $M$  with discrete Gaussian curvature  $K[g](p) < \epsilon$ .
6:   Compute new metric  $g$ , minimizing  $E(g_0, g)$  with constraint  $K[g] = K_i = 0$  on  $\Omega$ .
7:   Increment  $\epsilon$ .
8: end while
```

In its general form, the algorithm remains prohibitively expensive, as it requires a constrained nonlinear energy optimization at each step. For metrics conformally equivalent to the original metric, the problem is considerably simplified: the metric tensor has the form  $e^{2\varphi} I$ , where  $\varphi$  is a scalar function, and the energy needs to be optimized with respect to a scalar function  $\varphi$  rather than a tensor  $g$ .

Even more significantly, the energy  $E^{ARAP}(g)$  can be approximated by a *quadratic* energy and the nonlinear constraint on  $K[g]$  can be replaced by a *linear* constraint on  $\varphi$  (cf. [Jin et al. 2004; Ben-Chen et al. 2008]). By using  $\varphi$  as our variables, we entirely avoid computing the actual parametrization until the last stage of the algorithm.

Indeed, observe that for a conformal metric  $e^{2\varphi} I$  and small scale



**Figure 6: Curvature evolution:** An example of curvature evolution during flattening. Flattened areas are in white; negative curvatures increase in magnitude from cyan to blue; positive curvatures increase from yellow to red.

factors  $\varphi$ , the as-rigid-as-possible energy density is

$$\frac{1}{2} ((\sigma_1 - 1)^2 + (\sigma_2 - 1)^2) = (e^\varphi - 1)^2 \approx \varphi^2$$

i.e. is quadratic in the scale factor  $\varphi$ . As Figure 5 illustrates, this approximation serves as a good proxy even at the end of the flattening stage, when all but a few isolated vertices have been flattened. Another well-known fact about conformal metrics is that if the original metric has Gaussian curvature  $K^0$ , and the new metric is flat with cones angles  $\alpha_i = 2\pi - K_i$  at points  $c_i$ , then  $\varphi$  in the smooth case satisfies the Poisson equation

$$\Delta\varphi(p) = \sum_i K_i \delta(p - c_i) - K^0(p)$$

at every point  $p \in M$  (cf. [Bunin 2008]). We observe that this makes constraints in the optimization procedure linear. We end up with a significantly simpler problem in terms of the logarithmic scale factor  $\varphi$  for redistributing the metric during the optimization for flattening:

*Minimize  $\int_M \varphi^2 dA$  with constraint  $\Delta\varphi(p) = -K^0(p)$  for  $p \in \Omega$ .*

**Rounding.** The metric  $g$  produced by the flattening algorithm, is not seamless. At the second stage, we fix the cone position candidates  $c_i$  determined by flattening, and impose conditions on holonomy angles to make the parametrization seamless. We treat the conditions for cones and homology loops separately. First, we find the rounded curvatures  $K_i$  for the cones. Ideally, we need to minimize  $E(g_0, g)$  with respect to  $g$ , so that  $K[g]$  is zero everywhere except at cone candidates  $c_i$  where it has concentrated curvature  $K_i = k_i\pi/2$ , with  $k_i$  unknown integers.

While in principle this problem can be solved by complete enumeration if the range of rounded cone angles is restricted, the complexity is too high even for a moderate number of candidates. Instead, we again opt for a greedy procedure. As before, we iteratively fix the cone angle closest to a target value (*closest* integer multiple of  $\pi/2$  in this case). Just as in the case of rounding, we stay within the domain of conformal metrics, yielding the following procedure:

```

1: Rounding
2: Place all cone candidates  $c_i$  on a priority queue, with  $|K_i - \frac{\pi}{2} [2K_i/\pi]|$  as the key.
3: while priority queue not empty do
4:   Remove the first cone  $p$  from the queue, and add it to the list  $F$  of fixed cones.
5:    $K_i := \frac{\pi}{2} [2K_i/\pi]$ 
6:   Recompute the metric  $g$  and the cone angles for cones  $\notin F$  by minimizing  $\int_M \varphi^2 dA$  subject to constraint  $\Delta\varphi(p) = \sum_{c_i \in F} K_i \delta(p - c_i) - K^0(p)$  for all  $p \in F \cup \Omega$ .
7: end while
```

**Rounding homology loop holonomy angles.** The conformal flattening and rounding stages produce only cone angles, not holonomy

angles for the homology loops.

Once the rounded cone values are fixed, the conformal metric  $g$  is defined uniquely by the constraint  $\Delta\varphi(p) = \sum_{c_i \in F} K_i \delta(p - c_i) + K_i(p)$ , up to a scale factor that can be added to  $\varphi$ . This means that there may be no conformal metric for which the holonomy angles for homology loops are integer multiples of  $\pi/2$  as needed for a seamless metric, and the distortion can no longer be optimized by solving a quadratic problem. Computing holonomy angles for the conformal parametrization via (2) and rounding them to the nearest  $k\pi/2$  value yields an initial approximation.

For low genus, the angles can be further optimized in a brute-force manner by rounding each homology loop angle to each of the two nearest holonomy angles and computing the full metric with each rounded combination. However, in all of our experiments up to genus 4, this resulted in the same rounded holonomies as direct rounding.

**Discretization.** We use a standard linear finite element discretization for the Poisson equation for  $\varphi$ , as in [Ben-Chen et al. 2008].

At each step of the flattening algorithm, the set  $\Omega$  consists of vertices where we impose zero curvature constraint. The metric  $g$  is represented by vertex values  $\phi_i$ , determined at each step by minimizing  $\sum_i A_i \varphi_i^2$ , where  $A_i$  is the vertex area, subject to the constraint

$$\sum_j L_{ij} \phi_j = -K_i^d \quad (7)$$

for each flattened vertex  $i$ , where  $L$  is the standard Laplace matrix with cotangent weights,  $\phi$  is the vector of per-vertex values of  $\varphi$  and where  $K_i^d$  is the integral curvature over the vertex area of vertex  $i$ . The discrete curvature at a vertex  $i \notin \Omega$  in metric  $g$  is compared with threshold  $\epsilon_K$  to decide if it should be included in  $\Omega$  on the next step.

In the rounding algorithm, each cone  $c_i$  with rounded integral curvature  $K_i = k_i \frac{\pi}{2}$  contributes the additional constraint

$$\sum_j L_{ij} \phi_j = k_i \frac{\pi}{2} - K_i^d \quad (8)$$

Appendix A describes an efficient way to update the metric during the iterative flattening and rounding procedures, which allows the per-step metric to be recomputed efficiently.



**Figure 7: Distortion evolution.** Left to right: Initial surface with no distortion; distortion after flattening stage; distortion during iterative rounding; final distortion with all vertices rounded. Distortion varies from 0 (blue) to 0.5 (red) and is computed in triangle  $i$  as  $\min_{R \in SO(2)} \|J_i - R\|$  after minimizing the ARAP energy (9).

**Computational complexity.** The discrete conformal formulation provides further insight into the intrinsic complexity of the problem of finding optimal cone positions and rounded angles. If the problem is restricted to the space of discrete conformal mappings, it reduces to minimizing  $\sum_i A_i \varphi_i^2$ , with constraint  $\sum_j L_{ij} \phi_j = \pi k_i / 2 - K_i^d$ , for all  $i$ , and  $k_i$  (unknown) integers, i.e. to a standard mixed-integer problem. Mixed-integer problems are known to be NP-hard in general, and with no assumptions on the mesh connectivity, it is unlikely that the complexity of the problem can be reduced.

**Comparison to related techniques.** Our approach is motivated by the ideas of the mixed-integer solver described in [Bommes et al. 2009]. When enforcing integer constraints in a greedy manner, the solver proceeds by rounding variables already close to integers first, gradually progressing to variables further away from integers. This is the overall strategy we follow for curvatures.

**Flattening.** The flattening stage of our algorithm finds a conformal flat cone metric minimizing isometric distortion, determining cone positions and non-rounded angles. A related problem is solved in [Springborn et al. 2008] and [Ben-Chen et al. 2008]. In both of these papers, the starting point is a high-distortion conformal flat metric with no cones for a surface of disk topology. Cones are added in locations with highest distortion in a greedy way, decreasing the distortion energy, until a desired number of cones is added, or the distortion is below a threshold.

In contrast, our method starts with the original (non-flat) surface metric, which can be regarded as having cones at all vertices, and gradually expands the domain where vertices have zero curvature. Starting a greedy optimization from a metric with low rather than high distortion leads to lower distortion in the final map, for the same number of cones (see Figure 12). In our case, there are no stopping criteria, as the method naturally stops when all cone candidates are isolated. At the same time, if desired, we can control the maximal number of cones by forcing zero curvature at candidate cones until the number of cones is less than the desired number. [Ben-Chen et al. 2008] uses a Markov process to distribute curvature to the cones each time a cone is added, rather than directly minimize a distortion energy to determine curvature. Similarly to [Springborn et al. 2008], we use an energy directly. [Springborn et al. 2008] uses the Dirichlet energy of  $\phi$  to measure distortion rather than the  $L^2$ -norm of  $\|\phi\|$ . Due to its direct connection to the ARAP energy,  $\|\phi\|$  produces maps with a much lower fraction of points with high area distortion. Our approach is more suitable for surfaces of arbitrary topology, as it does not require an initial cone-free map, which cannot be constructed for surfaces other than disks.

We compare the performance of our algorithm against other methods for adding cone vertices to conformal maps in Section 7.

**Rounding.** In [Springborn et al. 2008], seamless parametrizations of domains of disk topology are produced by simultaneous rounding of all cones to the nearest multiple of  $\frac{\pi}{2}$ . Compared to our more expensive greedy strategy, this results in much higher distortion. Another strategy in the context of a top-down method would be to round each cone as it is added and update the metric. But this produces worse results as after a few successful cone insertions, new cones are consistently rounded to zero, preventing the addition of more cones needed to reduce distortion. Seamless metrics are not considered in [Ben-Chen et al. 2008], but using the rounding strategies above with their cones still produced higher distortion than our approach (cf. Section 7).

## 6 ARAP parametrization

At the last stage of our algorithm, we assume that all holonomy angles are fixed, both for cones and homology loops. We no longer restrict the parametrization to be conformal. Rather, we minimize the ARAP energy over all possible metrics with given holonomy angles.

### 6.1 Discrete ARAP energy for surfaces of arbitrary genus

As we discussed in Section 3, holonomy angles uniquely determine rotations at seams. Once rotations  $r_{ij}$  at all seam edges are fixed, the ARAP energy of [Liu et al. 2008] naturally generalizes to arbitrary surfaces. We use a set of affine maps  $f_i$ , one per triangle of the cut mesh  $M_c$ , to define the energy. The maps are defined by specifying values at vertices (multiple values at seam vertices, one for each consecutive pair of seam curves).

The Jacobian  $J_i$  of each map can be used to define the energy per triangle in the usual way:

$$E_i^{ARAP} = A_i \min_{R \in SO(2)} \|J_i - R\|^2, \quad (9)$$

with  $A_i$  denoting the triangle's area, subject to the rotation constraint on the seam. If the seam edge  $e_{ij}$  is shared by triangles  $T_i$ , then the constraint is  $J_j = r_{ij} J_i$  where  $r_{ij}$  is the rotation by angle  $\omega_{ij}$ , which is 0 on non-seam edges and a multiple of  $\pi/2$  across seams in a seamless parameterization.

There are two known ways to optimize this energy: the local-global method of [Liu et al. 2008], and the Newton procedure of [Chao et al. 2010]. Both methods are iterative, and the behavior strongly depends on the choice of the starting point. For surfaces of disk topology, [Liu et al. 2008] and [Chao et al. 2010] use the LSCM parametrization of [Lévy et al. 2002] as the starting point. However, an LSCM initialization results in a relatively large number of iterations for both methods (Figure 14, top).

Instead, we start with *rotations*  $R$ , and compute the initial parametrization by minimizing the energy with fixed  $R: \sum_i A_i \|J_i - R\|^2$ , i.e. performing the global step of the local-global iteration. On subsequent steps either Newton or local-global iterations can be used. As compared to an LSCM initialization, we found a 3- to 6-fold decrease in the number of iterations needed for iterations to converge as discussed in greater detail in Section 7.

To explain how a suitable initial rotation field is computed, we establish the relation between the rotations and *cross-fields* on the surface, and seam rotation angles of the parametrization and the rotation field.

## 6.2 Rotation field of a parametrization

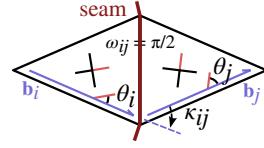
We start with explaining the idea for smooth surfaces, as it is simpler in the smooth case. For a given global parametrization  $f$ , we define the *best-fit rotation field* on  $M_c$  as  $R(J) = \operatorname{argmin}_{R \in SO(2)} \|J - R\|^2$  for  $J = \nabla f$ . The columns of  $R^T = [\mathbf{q}_1, \mathbf{q}_2]$  define orthonormal vectors in the tangent plane of  $M_c$ . If we fix a reference frame field  $\mathbf{b}$  on the surface,  $R$  can be represented by the angles  $\theta$  between  $\mathbf{q}_1$  and  $\mathbf{b}$ , and the four angles  $\theta + k\pi/2$ ,  $k \in \{0, 1, 2, 3\}$ , correspond to  $\pm \mathbf{q}_m$ ,  $m \in \{1, 2\}$ . In this way, the rotations  $R$  define a *cross-field* on  $M_c$ .

Conversely, if a cross-field has no singularities on  $M_c$ , which is a topological disk, it can be split globally into four unit vector fields  $\mathbf{q}_i(p)$ ,  $i \in \{1, 2, 3, 4\}$ . These vector fields are not continuous across the seam. If  $p_1$  and  $p_2$  are two points of  $M_c$  corresponding to a point  $p$  on the seam, then the angle between  $\mathbf{q}_i(p_1)$  and  $\mathbf{q}_i(p_2)$  is of the form  $k\pi/2$  for an integer  $k$ . We call this angle a *seam rotation angle of the field* and it is constant along each seam curve by the continuity of the vector field on each side. The integer  $k$  is called *matching* (cf. [Ray et al. 2008]). If we define the rotation field by two orthogonal vector fields  $\mathbf{q}_1$  and  $\mathbf{q}_2$ , so that  $R^T = [\mathbf{q}_1, \mathbf{q}_2]$ , then  $R(p_2) = rR(p_1)$  at a seam point  $p$ . This observation leads to the following proposition, proved in the Appendix.

**Proposition 3** *For an orientation-preserving parametrization, the seam rotation  $r(p)$  of the cross-field defined by  $R(J)$  at a seam point  $p$  coincides with the seam rotation of  $J$ .*

**Initializing rotations using cross-field optimization.** Although we do not know  $J$ , the flattening and rounding stage of the algorithm yields all holonomy angles, and therefore also the seam rotation angles. Proposition 3 suggests initializing the local-global iteration using a smooth cross-field with seam rotation angles constrained to be equal to the one obtained from holonomy angles.

We represent the discrete cross-field on each triangle  $T_i$  by an angle  $\theta_i$  with respect to a frame  $\mathbf{b}_i$  fixed per-triangle. In addition, seam rotation angles  $\omega_{ij} = k_{ij}\pi/2$  are specified on seam edges  $e_{ij}$  and are presumed to be zero in the interior. With  $\kappa_{ij}$  being the angle



**Figure 8: Cross-field representation.** The angle-based cross-field representation can be transported from facet  $i$  to  $j$  as  $\theta_i + \kappa_{ij} + \omega_{ij}^C$  for direct comparison with  $\theta_j$ , where  $\kappa_{ij}$  is the difference in angle between the per-triangle reference frames  $\mathbf{b}_i$  and  $\mathbf{b}_j$ .

between  $\mathbf{b}_i$  and  $\mathbf{b}_j$  (Figure 8), we minimize the quadratic energy

$$\min_{\theta} \sum_{ij} (\theta_i - \theta_j + \kappa_{ij} + \omega_{ij})^2. \quad (10)$$

The minimizer is unique if  $\theta_i$  is fixed for one triangle. The same energy is used in [Ray et al. 2008; Bommes et al. 2009] and is equivalent to the energy used in [Crane et al. 2010]. Once the angles  $\theta_i$  are computed, they are converted to rotation matrices

$$R_i = \left[ \frac{\mathbf{b}_i}{|\mathbf{b}_i|} \cos \theta, \frac{\mathbf{b}_i^\perp}{|\mathbf{b}_i|} \sin \theta \right]^T \quad (11)$$

used to initialize the local-global iteration.

**Remark 2.** We note that Proposition 3 can be also formulated in terms of the *turning numbers* of the cross-field (which can be expressed in terms of holonomy angles of the metric) [Lai et al. 2009], as well as in terms of holonomy angles of the principal connection associated with the cross-field [Crane et al. 2010]. We use the less invariant notion of seam rotations as it yields an algorithm for computing an initial field more directly.

**Remark 3.** Algorithms of [Bommes et al. 2009] and [Kälberer et al. 2007] construct their parametrizations starting with a cross-field, which is optimized to follow feature lines. The directions of the cross-field are used as target directions for the parametrization gradients minimizing the energy

$$E^{\text{feature}} = \sum_i A_i (\|\nabla u - \mathbf{u}_i\|^2 + \|\nabla v - \mathbf{v}_i\|^2) \quad (12)$$

We observe that  $[\nabla u \ \nabla v]^T$  is exactly the Jacobian matrix of the parametrization of a triangle, and  $[\mathbf{u} \ \mathbf{v}]^T$  is a rotation matrix. Per triangle, the energy has the form  $\|J_i - R_i\|^2$ , i.e. the energy minimized at the global step of the ARAP iteration.

## 6.3 Final algorithm

To summarize, our algorithm proceeds as follows:

1. **Cone angles ( $k_i$ ):** Given an input surface  $M$ , perform the **Flattening** and **Rounding** subalgorithms listed in Section 5.
  - (a) **Flattening:** Iteratively evolve the surface metric by flattening all points whose curvatures lie within an increasing threshold  $\epsilon$ . The logarithmic metric scaling factor  $\varphi$  discretized using vertex values  $\phi_i$  is evolved by constraining every flattened vertex  $i$  by

$$\sum_j L_{ij} \phi_j = K_i - K_i^d$$

with  $K_i = 0$  and minimizing the distortion  $\int_M \varphi^2 dA$ . Flattening completes once all curvatures are concentrated at isolated vertices which serve as cone candidates for the Rounding phase.

- (b) **Rounding:** Continue evolving the metric by iteratively rounding the curvatures of cone candidates, prioritizing those that are closer to multiples of  $\pi/2$ , until there are

no more vertices remaining. Each cone  $c_i$  with curvature rounded to  $K_i = k_i\pi/2$  contributes an additional constraint on  $\phi$  as in the flattening stage, but with a non-zero curvature.

This yields for each vertex  $i$  on the surface an integral curvature  $K_i$  which is used to constrain the holonomy  $A_g^d(\gamma_i^c) = K_i$  of loops  $\gamma_i^c$  around each vertex  $i$ .

2. **Holonomy angles for homology loops:** (Section 5) Compute rotation angles  $A_g^d(\gamma_i^h)$  for a homology basis via (2) using the new metric, and round these angles to multiples of  $\pi/2$ .
3. **Seam rotation angles  $\omega_{ij}$ :** (Section 3, Proposition 2) Given all holonomy angles from steps 1 and 2, solve for the unique compatible  $\omega_{ij}$ , constrained by (4). Note that  $\omega_{ij} = 0$  along non-seam edges.
4. **Initial  $R_i$  for ARAP:** (Section 6.2) Compute  $\theta_i$  from  $\omega_{ij}$  by minimizing (10). Compute  $R_i$  from  $\theta_i$  via (11).
5. **ARAP:** (Section 6.1) With  $\omega_{ij}$  fixed, and initial per-triangle rotations  $R_i$  specified, initialize  $f = (u, v)$  by minimizing  $E^{\text{feature}}$  (12). From this initialization, the ARAP energy can be minimized using either
  - the local-global iterations of [Liu et al. 2008], or
  - the Newton method of [Chao et al. 2010].

Either approach yields a final global parameterization  $f$ .

## 7 Results and Discussion

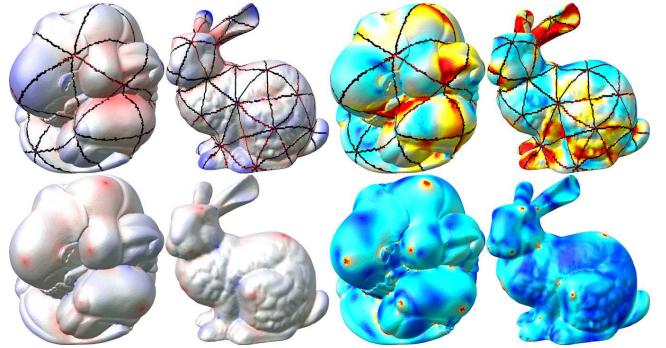
We evaluate the results of our algorithm in several ways. The overall algorithm is compared to recent parametrization methods optimizing isometry. As the flattening and rounding stages use conformal parametrizations as proxies for the actual parametrization, we compare the results of this stage with previously proposed techniques for adding cones to conformal parametrizations. For the global ARAP parametrization, we compare initialization following [Liu et al. 2008], with our rotation field initialization, and demonstrate a 6-fold speedup for the Newton method, and a 3-fold decrease in the number of local-global iterations to reach within 1% of the final energy. We explore the dependence of the results on the main parameter of the method (curvature threshold increment), and discuss the limitations at the end.

In all cases, other than the area distortion in Figure 9, *left*, the distortion visualized in the figures in this section is computed on each triangle  $i$  as the ARAP deviation from isometry  $\min_{R \in SO(2)} \|J_i - R\|$ .

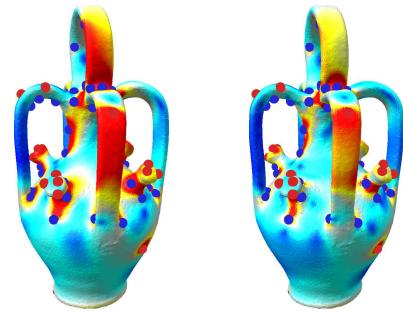
**Comparison with methods optimizing isometry.** There are relatively few global parametrization methods directly optimizing a measure of deviation from isometry. The almost-isometric parametrization algorithm of [Pietroni et al. 2009] uses a completely different approach to global parametrization, based on constructing a domain by mesh simplification, following [Eck et al. 1995; Lee et al. 1998; Khodakovsky et al. 2003]. The method optimizes a combination of quality factors different from ours, so comparisons on our metric may be somewhat misleading (Figure 9, right). While our method clearly outperforms that method in terms of overall distortion, the area error distribution is more concentrated at singularities for our approach.

While [Bommes et al. 2009] does not state isometry optimization as its goal, as we have shown in Section 6, it is very close to a global version of ARAP parametrization we propose. It is not surprising that in terms of isometric distortion it outperforms global parametrization methods based on conformal and harmonic parametrization. Its isometric distortion is qualitatively similar to the related methods of [Ray et al. 2006] and [Kälberer et al. 2007], so we focus on comparisons with this method.

The distortion reduction compared to [Bommes et al. 2009] is due to two factors: the choice of cone positions and angles, and as-



**Figure 9:** Upper row: (left) The almost-isometric parametrization of [Pietroni et al. 2009] minimizes the area distortion of large triangular patches, but (right) deviates highly from isometry. Lower row: Our method performs well on both measures. Area scale color varies logarithmically from scale 1/2 (blue) to 2 (red). The isometric distortion varies from 0 (blue) to 0.5 (red).

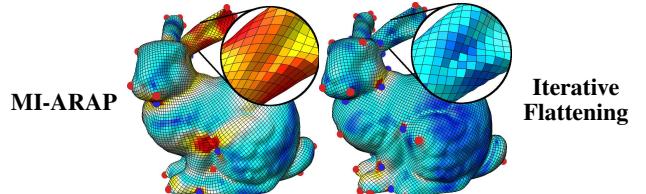


**Figure 10: Distortion reduction due to ARAP.** Left: Parametrization of [Bommes et al. 2009]. Right: Global ARAP parametrization initialized with the same cones. The isometric distortion varies from 0 (blue) to 0.5 (red).

rigid-as-possible parametrization for fixed cones. Figure 10 compares the behavior of the mixed-integer parametrization and ARAP parametrization with the same cones, showing a significantly lower distortion. However, most of the distortion reduction is due to cone placement. To isolate the effects of cone placement, rather than comparing to the mixed-integer parametrization in pure form, we apply our ARAP optimization to the cones produced by [Bommes et al. 2009].

Figure 17 shows a comparison of the distributions of distortion on a number of models. We observe that we obtain an improvement in all instances. Further details on the models are listed in Table 1.

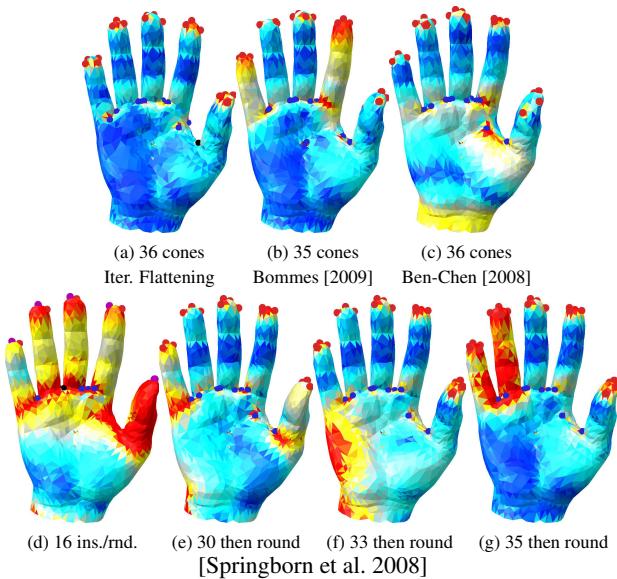
Figure 11 compares a quadrangulation generated from ARAP parametrizations on Mixed Integer fields and ours, obtained after rounding seam translations and cone parametric coordinates. The visualized distortion  $\min_{R \in SO(2)} \|J - R\|$  is computed by interpreting each quad as a bilinear map from a unit quad and evaluating the Jacobian at the center.



**Figure 11: Comparison of quadrangulations from ARAP with Mixed Integer fields and our method.** Nearby cones were automatically collapsed for both methods and cone positions were rounded. The distortion visualization is scaled from 0 (red) to 0.5 (blue).

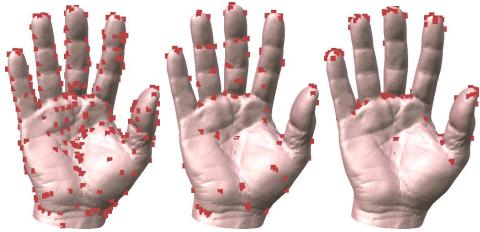
|             | faces  | $N_{MI}$ | $N_{final}$ | $T_{flat}$ | $T_{rnd}$ | $D_{MI}$ | $D_{final}$ |
|-------------|--------|----------|-------------|------------|-----------|----------|-------------|
| Fandisk     | 14454  | 29       | 28          | 6.52       | 6.03      | 0.161    | 0.0829      |
| Dancer      | 49996  | 71       | 87          | 20.9       | 64.5      | 0.216    | 0.106       |
| Elephant    | 49918  | 133      | 118         | 22.4       | 95.7      | 0.197    | 0.113       |
| Screwdriver | 54300  | 20       | 26          | 33.1       | 40.4      | 0.244    | 0.095       |
| Hand        | 65524  | 42       | 35          | 36.4       | 43.4      | 0.223    | 0.0757      |
| Botijo      | 82332  | 80       | 89          | 74.6       | 188       | 0.245    | 0.0890      |
| Horse       | 96966  | 111      | 70          | 63.4       | 211       | 0.234    | 0.105       |
| Omotondo    | 100000 | 98       | 68          | 90.4       | 301       | 0.192    | 0.095       |
| Bunny       | 111364 | 36       | 44          | 91.1       | 133       | 0.248    | 0.0844      |

**Table 1:** Column titles:  $N_{MI}$  and  $N_{final}$  are, respectively, the number of cones in the MI field, and our parametrization after cone collapse; the  $N_{cand}$  column denotes the number of cone candidates;  $T_{flat}$  and  $T_{rnd}$  are the run times, in seconds, of the flattening and rounding procedures, respectively;  $D_{MI}$  and  $D_{final}$  denote, respectively, the average triangle distortions of the MI-ARAP procedure and on our final model.

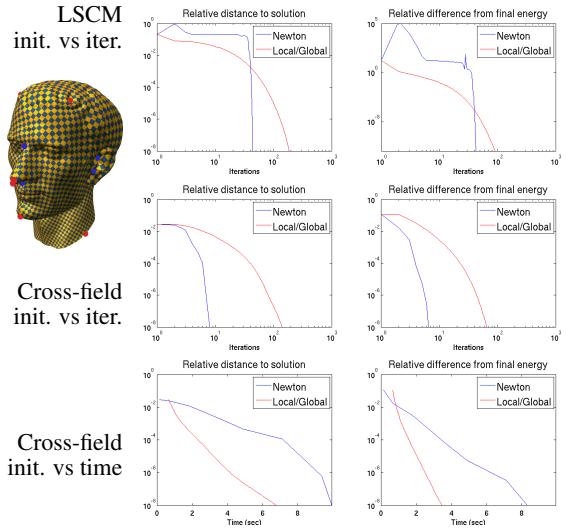


**Figure 12: Conformal insertion methods.** (a) Iterative flattening (b) Mixed Integer [Bommes et al. 2009] (c) the cone insertion [Ben-Chen et al. 2008] with cone curvatures rounded at the end. [Springborn et al. 2008]: (d) iterative cone insertion of [Springborn et al. 2008] (e–g) simultaneous cone insertion followed by rounding all at once. In all cases, ARAP parameterization was computed using the prescribed cones, and the scale is 0 (red) to 0.5 (blue).

**Comparison with cone insertion methods for conformal maps.** A related problem, addressed by [Springborn et al. 2008; Ben-Chen et al. 2008], minimizes the metric distortion while allowing no more than  $k$  cones with arbitrary cone angles. The greedy procedure suggested by both methods starts with a conformally flattened surface and iteratively inserts cone singularities at vertices of maximal  $|\phi|$ . Springborn et al. [2008] suggest two ways to round cone curvatures: inserting and rounding cones in sequence, or inserting a fixed number of cones and rounding them all at once. In the example of Figure 12(d), the former method stops at 16 cones – far too few to lower distortion – before rounding the curvatures of subsequent cones to 0. The latter, trial-and-error approach of guessing either the termination threshold of [Ben-Chen et al. 2008] or the the number of cones to insert with [Springborn et al. 2008] (Figure 12(c,e–g)) gives better results but is cumbersome and still yields a greater energy than the cones of [Bommes et al. 2009] and our iterative flattening approach shown in Figure 12(a,b). The maximal  $\phi$ -difference termination threshold [Ben-Chen et al. 2008] of 0.9 was used for Figure 12(c) so that four cones were produced on every finger.



**Figure 13:** Left and middle: local maxima on successively smoothed curvature yield many randomly-placed cone candidates; Right: our iterative flattening procedure concentrates curvature in more meaningful locations.



**Figure 14: Convergence of Newton vs local-global iterations.** Initialization with LSCM yields slower convergence for both procedures as compared to the cross-field-based initialization proposed in Section 6. Newton converges to high accuracy in far fewer iterations than the local-global procedure (middle row), but at a greater computational cost per iteration (bottom row).

**Cones at local curvature extrema.** We observe that in simple cases, incremental flattening is likely to push curvature to existing curvature extrema. For completeness, we compare to the algorithm of placing cone candidates at the extrema of Gaussian curvature, suggested in [Tong et al. 2006]. Figure 13 shows the results of this approach for two levels of smoothing of Gaussian curvature.

**Rotation field vs. harmonic field initialization of ARAP.** An alternative to using local-global iterations is to use a Newton solver, which was shown to exhibit much faster convergence [Chao et al. 2010]. As an example, Figure 14 illustrates the convergence rates of the solution and the energy of both procedures on a Julius head model with 10K triangles and 15 cones produced by our flatten procedure. We compare the Newton and the local-global solvers with different initializations: an LSCM parameterization [Lévy et al. 2002], and a global step from the cross-field  $R_i$  in Section 6. As Figure 14, top, shows, LSCM is a poorer initializer for both procedures, especially Newton. In contrast, 10 local-global iterations from the cross-field initialization decrease the energy to within 1% of the minimum, which may be sufficient for some applications.

While the Newton solver initialized by the method of Section 6 seems the clear winner in terms of the number of iterations, the cost per iteration is very high. The time is dominated by the conjugate gradient iterations used to solve for the step size from the Hessian and the gradient; the procedure requires nearly 10 seconds for the 7 Newton iterations in this example. In contrast, the matrix of the global step remains unchanged throughout the local-

global procedure. The overall cost is thus dominated by a one-time Cholesky factorization of the matrix (0.654 seconds) and a much cheaper back-substitution (0.044 seconds) per iteration. Thus, 200 local-global can be performed in the same time as the 7 Newton iterations above. Depending on the application, just 10 local-global iterations may suffice, requiring less than a second in this example. Both algorithms were implemented in Matlab for a fair comparison.

**Dependence on the rate of increase in curvature threshold.** Next, we consider the dependence of the result on the rate of increase in the curvature threshold, determining how far the domain expands at each step of the flattening stage. Figure 15 shows that the dependence is moderate for a broad range of values. However, choosing a very large threshold increment flattens the surface too aggressively on fine meshes where the angle deficit around a vertex is low in the initial metric.

**Limitations.** An important limitation of our method is that it is ignoring mesh features. We make no attempt to align the isoparametric lines with high-curvature or sharp regions as it is done in [Bommes et al. 2009], or to adapt the surface metric to features as in [Kovacs et al. 2011]. The latter is straightforward, as it only requires assigning different edge lengths, while the first limitation is more fundamental, and requires substantial changes in our framework. As Figure 16 shows, while our iterative flattening approach produces lower distortion than mixed integer fields, it does not support feature alignment, which we leave for future work.

For quadrangulation, in addition to the holonomy constraints we describe, it is essential to ensure that cone parametric coordinates and seam translations are integers (Figure 3). Our distortion minimization procedure does not account for this constraint. However, if the quadrangulation is sufficiently fine as on the Bunny in Figure 11, the distortion of the quadrangulation with the required additional integer constraints does not deviate from the distortion of the original parameterization. For applications like generating coarse subdivision control meshes, very coarse rounding of cones positions and seam translations is clearly desired and additional effort is required to integrate this rounding into the process. We note that for rectangular texture atlases, multi-chart geometry images [Carr et al. 2006], and T-mesh constructions [Myles et al. 2010], rounding is not essential or rounding by small distances is sufficient.



**Figure 15: Choice of threshold increment:** Left: Too large an increment flattens far too many vertices, especially on fine meshes where the integrated curvature (angle deficit) around a vertex is small in the initial metric. Right: Too small an increment can be unnecessarily expensive, taking about 3.5 minutes for flattening in this example. Middle:  $\frac{\pi}{200}$ , taking about 40 seconds, serves as a good compromise.

## A Incremental solution for $\phi$

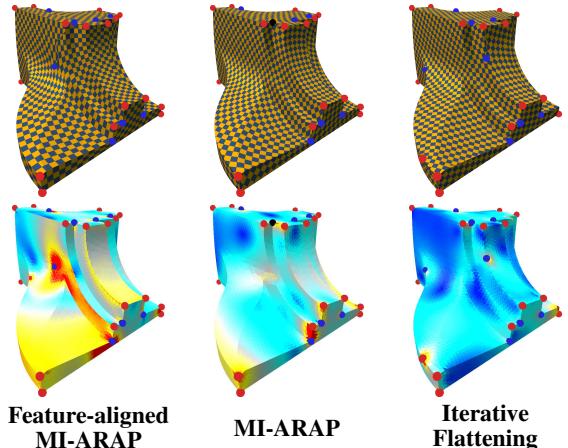
With the  $\phi$  denoting the vector of all  $\phi_i$ , we minimize

$$\phi^t D \phi = \|D^{1/2} \phi\|^2 \quad \text{constrained by } C \phi = d,$$

where  $D$  is the matrix with vertex areas  $A_i$  on the diagonal, and  $C$  and  $d$  enumerate those constraints (7) and (8) where  $k_i$  are fixed. The pseudoinverse yields a direct formula

$$\phi = D^{-1} C^t (C D^{-1} C^t)^{-1} d, \quad (13)$$

which can be computed by solving  $(C D^{-1} C^t) y = d$  for the vector  $y$  using a Cholesky factorization and evaluating  $\phi = D^{-1} C^t y$ .



**Figure 16: Comparison to feature-aligned mixed-integer parametrization.** While iterative flattening exhibits far less distortion, it is not yet sensitive to features, which is left for future work. The distortion visualization is scaled from 0 (red) to 0.5 (blue).

As more curvatures are fixed, the number of rows of the Laplacian matrix  $C$  increases, which is not amenable to utilizing an incremental Cholesky factor update [Chen et al. 2008]. Therefore, we pre-compute the matrix  $B = LD^{-1}L^t$ , where  $L$  is the Laplacian matrix of the complete mesh. With  $N_{V_i}$  denoting the number of interior vertices, we define the  $N_{V_i} \times N_{V_i}$  matrix  $\tilde{B}$  and the  $N_{V_i}$ -vector  $\tilde{d}$  as

$$\tilde{B}_{ij} = \begin{cases} B_{ij} & \text{if } k_i \text{ and } k_j \text{ fixed} \\ \delta_{ij} & \text{otherwise} \end{cases} \quad \tilde{d}_i = \begin{cases} \frac{\pi}{2} k_i - A_i^d & \text{if } k_i \text{ fixed} \\ 0 & \text{otherwise} \end{cases}$$

where  $\delta_{ij}$  is the Kronecker delta function. Then,  $\phi$  may be computed with

$$\tilde{y} = \tilde{B}^{-1} \tilde{d}, \quad \phi = (D^{-1} L^t) \tilde{y}.$$

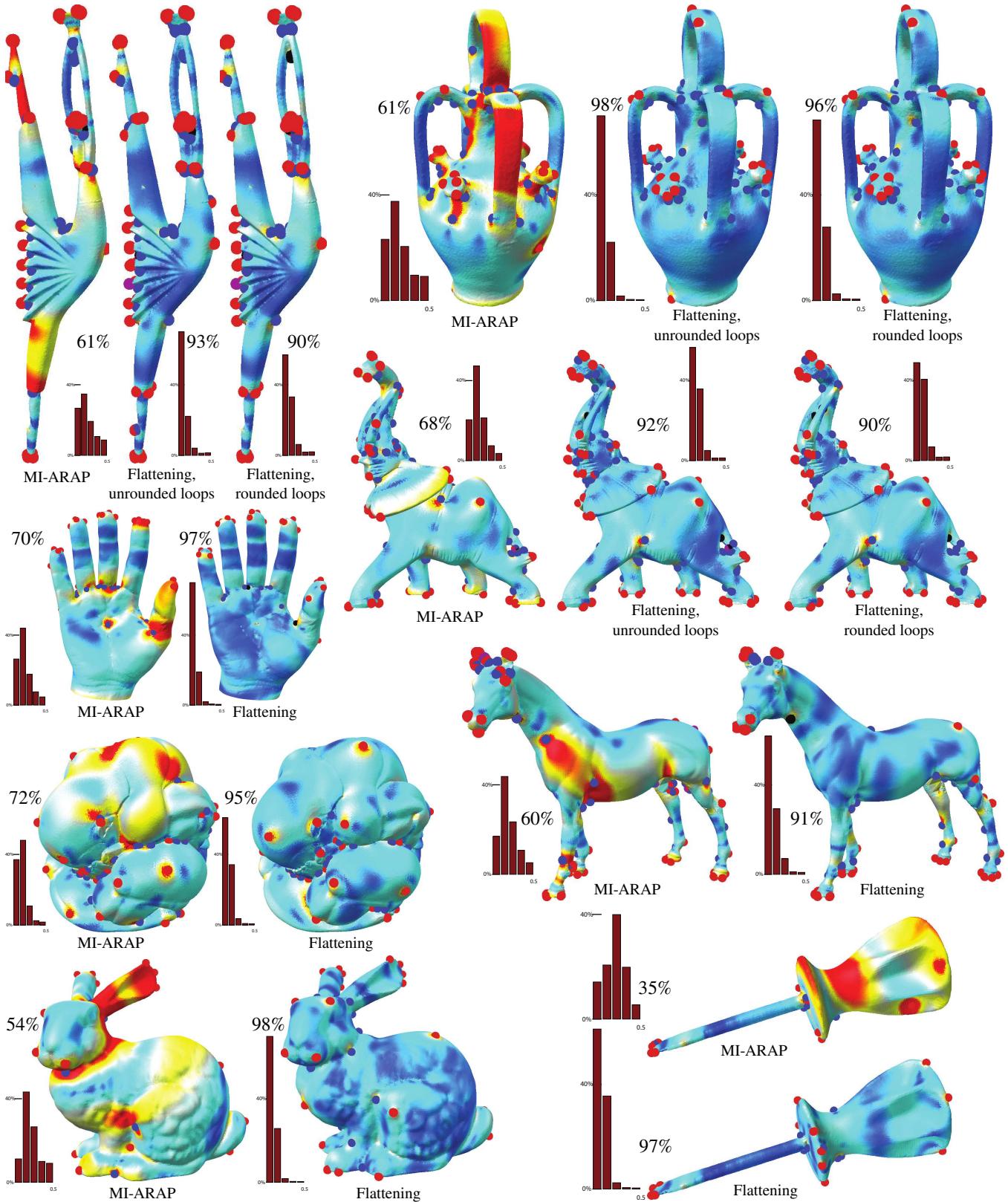
This system is conducive to incremental Cholesky factor updates since  $\tilde{B}$  is symmetric positive-definite and has constant size.

## B Proofs of propositions

**Proposition 2.** Suppose a parametrization  $f$  has a seamless metric, and let  $M_c$  be the cut mesh. Construct a dual spanning tree  $\mathcal{T}$  of triangles of the mesh with edges in the interior of  $M_c$ . Consider a seam edge  $e$  and two triangles incident at the edge. For each of the two triangles, there is a path to the root of the tree consisting of edges dual to interior edges  $M_c$ . As all seam rotation angles are zero at the interior edges of  $M_c$ , the only nonzero rotation angle on the loop formed by the two paths and the edge  $\tilde{e}$  dual to  $e$  is the angle on  $e$ . By (4), this angle is equal to the homology angle of the loop. The loop can be represented as a combination of homology basis loops, which, by assumption, have rotation angles  $k\pi/2$ , where  $k$  is integer, so we conclude that all seam rotation angles are of the same form.

Conversely, suppose the parametrization is seamless, i.e. all seam rotation angles are  $k\pi/2$ . We can find a homology basis of the same structure (two paths to the root of the spanning tree and a single additional edge [Erickson and Whittlesey 2005]); as there is only a single nonzero rotation angle on the edge, we conclude that the homology angle of the loop is equal to these rotation angles.

Note that the spanning tree construction defines a linear system relating two sets of variables (holonomy angles of a metric and seam rotation angles), and each set defines the other set uniquely. We note that effectively this is the same relation as the one described in [Ray et al. 2006] for turning numbers and matchings of a discrete



**Figure 17: Distortion comparison:** Distortion coloring and histograms show that ARAP on cross-fields generated by the Mixed Integer algorithm (MI-ARAP) yields higher distortion than ARAP on the Iterative Flattening procedure. For the flattening procedure, we show the distortion both before and after the homology loops are rounded. The percentage accompanying the histogram indicates the fraction of triangles with distortion less than 0.2. The distortion is visualized from 0 (blue) to 0.4 (red).

$N$ -symmetry field, and the zippering algorithm described in [Ray et al. 2006] can be viewed as a way to solve this linear system for rotation angles.

**Proposition 3.** Suppose points  $p_1$  and  $p_2$  of the boundary of  $M_c$  correspond to the seam point  $p$ , and suppose  $J_1 = J(p_1) = rJ(p_2) = rJ_2$ , where  $r$  is a  $k\pi/2$  rotation.  $R(J)$  can be expressed analytically as a function of  $J$ : Let  $B_i = J_i - J_i^T + (\text{Tr } J_i)I$ . Then  $R(J_i) = B_i / \det B_i$ .  $\det B = 0$  if and only if  $J$  is a similarity transformation combined with axial reflection, which is not possible if  $J$  is orientation-preserving. Then  $R(J_1) = rR(J_2)$  is easily verified, by a direct calculation, for the four possible rotation matrices  $r$ .

## References

- BEN-CHEN, M., GOTSMAN, C., AND BUNIN, G. 2008. Conformal flattening by curvature prescription and metric scaling. *Computer Graphics Forum* 27, 2, 449–458.
- BOMMES, D., ZIMMER, H., AND KOBBELT, L. 2009. Mixed-integer quadrangulation. *ACM Trans. Graph.* 28, 3, 77.
- BUNIN, G. 2008. A continuum theory for unstructured mesh generation in two dimensions. *CAGD* 25, 14–40.
- CARR, N., HOBEROCK, J., CRANE, K., AND HART, J. 2006. Rectangular multi-chart geometry images. In *Symposium on Geometry Processing*, Eurographics Association, 190.
- CHAO, I., PINKALL, U., SANAN, P., AND SCHRÖDER, P. 2010. A simple geometric model for elastic deformations. *ACM Trans. Graph.* 29, 4 (July), 38:1–38:6.
- CHEN, Y., DAVIS, T. A., HAGER, W. W., AND RAJAMANICKAM, S. 2008. Algorithm 887: CHOLMOD, supernodal sparse cholesky factorization and update/downdate. *ACM Trans. Math. Softw.* 35 (October), 22:1–22:14.
- CRANE, K., DESBRUN, M., AND SCHRÖDER, P. 2010. Trivial connections on discrete surfaces. *Computer Graphics Forum* 29, 5 (July), 1525–1533.
- DANIELS, J., SILVA, C. T., AND COHEN, E. 2009. Localized quadrilateral coarsening. *Computer Graphics Forum* 28, 5, 1437–1444.
- DANIELS II, J., SILVA, C. T., AND COHEN, E. 2009. Semiregular quadrilateral remeshing from simplified base domains. *Computer Graphics Forum* 28, 5 (July), 1427–1435.
- DONG, S., BREMER, P., GARLAND, M., PASCUCCI, V., AND HART, J. 2006. Spectral surface quadrangulation. *ACM Trans. Graph.* 25, 3, 1057–1066.
- ECK, M., DEROSSE, T., DUCHAMP, T., HOPPE, H., LOUNSBERRY, M., AND STUETZLE, W. 1995. Multiresolution analysis of arbitrary meshes. *SIGGRAPH 1995*, 173–182.
- ERICKSON, J., AND WHITTLESEY, K. 2005. Greedy optimal homotopy and homology generators. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 1046.
- GU, X., AND YAU, S.-T. 2003. Global conformal surface parameterization. In *Proc. 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP ’03, 127–137.
- GU, X., GORTLER, S., AND HOPPE, H. 2002. Geometry images. *ACM Trans. Graph.* 21, 3, 355–361.
- HERTZMANN, A., AND ZORIN, D. 2000. Illustrating smooth surfaces. In *SIGGRAPH 2000*, 517–526.
- HORMANN, K., LÉVY, B., AND SHEFFER, A. 2007. Mesh parameterization: Theory and practice. *SIGGRAPH Course Notes*.
- JIN, M., WANG, Y., YAU, S., AND GU, X. 2004. Optimal global conformal surface parameterization. In *Proc. IEEE Visualization’04*, 267–274.
- JIN, M., KIM, J., LUO, F., AND GU, X. 2008. Discrete surface ricci flow. *IEEE Trans. Visualization and Computer Graphics* 14, 1030–1043.
- KÄLBERER, F., NIESER, M., AND POLTHIER, K. 2007. QuadCover: Surface Parameterization using Branched Coverings. *Computer Graphics Forum* 26, 3, 375–384.
- KHAREVYCH, L., SPRINGBORN, B., AND SCHRÖDER, P. 2006. Discrete conformal mappings via circle patterns. *ACM Trans. Graph.* 25 (April), 412–438.
- KHODAKOVSKY, A., LITKE, N., AND SCHRÖDER, P. 2003. Globally smooth parameterizations with low distortion. *ACM Trans. Graph.* 22, 3, 350–357.
- KOVACS, D., MYLES, A., AND ZORIN, D. 2011. Anisotropic quadrangulation. *Computer Aided Geometric Design* 28, 8, 449–462. Solid and Physical Modeling 2010.
- LAI, Y., JIN, M., XIE, X., HE, Y., PALACIOS, J., ZHANG, E., HU, S., AND GU, X. 2009. Metric-driven rosy field design and remeshing. *IEEE Trans. Visualization and Computer Graphics*, 95–108.
- LEE, A., SWELDENS, W., SCHRÖDER, P., COWSAR, L., AND DOBKIN, D. 1998. MAPS: multiresolution adaptive parameterization of surfaces. In *SIGGRAPH 1998*, 95–104.
- LÉVY, B., PETITJEAN, S., RAY, N., AND MAILLOT, J. 2002. Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.* 21, 3, 362–371.
- LIU, L., ZHANG, L., XU, Y., GOTSMAN, C., AND GORTLER, S. J. 2008. A Local/Global approach to mesh parameterization. *Computer Graphics Forum* 27, 5 (July), 1495–1504.
- MARINOV, M., AND KOBBELT, L. 2005. Automatic generation of structure preserving multiresolution models. *Computer Graphics Forum* 24, 3 (Sept.), 479–486.
- MYLES, A., PIETRONI, N., KOVACS, D., AND ZORIN, D. 2010. Feature-aligned T-meshes. *ACM Trans. Graph.* 29, 4, 1–11.
- PALACIOS, J., AND ZHANG, E. 2007. Rotational symmetry field design on surfaces. *ACM Trans. Graph.* 26, 3, 55.
- PIETRONI, N., TARINI, M., AND CIGNONI, P. 2009. Almost isometric mesh parameterization through abstract domains. *IEEE Trans. Visualization and Computer Graphics* 99, RapidPosts.
- RAY, N., LI, W., LÉVY, B., SHEFFER, A., AND ALLIEZ, P. 2006. Periodic global parameterization. *ACM Trans. Graph.* 25, 4, 1460–1485.
- RAY, N., VALLET, B., LI, W., AND LÉVY, B. 2008. N-Symmetry direction field design. *ACM Trans. Graph.* 27, 2.
- RAY, N., VALLET, B., ALONSO, L., AND LEVY, B. 2009. Geometry-aware direction field processing. *ACM Trans. Graph.* 29, 1, 1–11.
- SHEFFER, A., PRAUN, E., AND ROSE, K. 2006. Mesh parameterization methods and their applications. *Foundations and Trends® in Computer Graphics and Vision* 2, 2, 171.
- SORKINE, O., COHEN-OR, D., GOLDENTHAL, R., AND LISCHINSKI, D. 2002. Bounded-distortion piecewise mesh parameterization. In *Proc. IEEE Visualization ’02*, 355–362.
- SPRINGBORN, B., SCHRÖDER, P., AND PINKALL, U. 2008. Conformal equivalence of triangle meshes. *ACM Trans. Graph.* 27 (August), 77:1–77:11.
- TARINI, M., PIETRONI, N., CIGNONI, P., PANZZO, D., AND PUPPO, E. 2010. Practical quad mesh simplification. *Computer Graphics Forum* 29, 2.
- TONG, Y., ALLIEZ, P., COHEN-STEINER, D., AND DESBRUN, M. 2006. Designing quadrangulations with discrete harmonic forms. *Symposium on Geometry Processing*, 201–210.