Eurographics Symposium on Geometry Processing 2024
R. Hu and S. Lefebvre
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 5

Mesh Parameterization Meets Intrinsic Triangulations

Koray Akalin[1,2†] Ugo Finnendahl[1†] Olga Sorkine-Hornung[2] Marc Alexa[1]

[1]TU Berlin [2]ETH Zürich

Abstract

A parameterization of a triangle mesh is a realization in the plane so that all triangles have positive signed area. Triangle mesh parameterizations are commonly computed by minimizing a distortion energy, measuring the distortions of the triangles as they are mapped into the parameter domain. It is assumed that the triangulation is fixed and the triangles are mapped affinely. We consider a more general setup and additionally optimize among the intrinsic triangulations of the piecewise linear input geometry. This means the distortion energy is computed for the same geometry, yet the space of possible parameterizations is enlarged. For minimizing the distortion energy, we suggest alternating between varying the parameter locations of the vertices and intrinsic flipping. We show that this process improves the mapping for different distortion energies at moderate additional cost. We also find intrinsic triangulations that are better starting points for the optimization of positions, offering a compromise between the full optimization approach and exploiting the additional freedom of intrinsic triangulations.

CCS Concepts

• ***Computing methodologies*** → ***Computer graphics; Mesh models; Mesh geometry models;***

1. Introduction

A *parameterized surface* is given by a map $\mathbf{p} : \Omega \subset \mathbb{R}^2 \mapsto \mathcal{S} \subset \mathbb{R}^3$. The inverse $\mathbf{p}^{-1}$ is commonly called a *parameterization* of the surface $\mathcal{S}$. Assuming $\mathbf{p}$ is continuously differentiable, the Jacobian $\mathbf{J_p}$ captures how the surface locally shears and stretches. If the (two non-trivial) singular values of $\mathbf{J_p}$ are identical, the mapping is *conformal*; if their product is equal to 1, the mapping is *area-preserving* (or *authalic*). If the mapping is both conformal and area-preserving, it is *isometric* – this is ideal for applications, since properties defined over the surface can be modeled in the plane, as the mapping introduces no distortion. For surfaces with non-zero Gaussian curvature, however, the mapping cannot be isometric. The quality of a parameterization is commonly measured as the deviation from isometry. While there are many different quality measures, most of them are described in terms of $\mathbf{J_p}$, and in particular, its singular values [LZX*08, AL13, SS15, RPPSH17]. The total quality of the mapping $\mathbf{p}$ is an integral over the the surface $\mathcal{S}$ (or the domain Ω) – commonly called a *distortion energy*.

A triangulated surface with vertex positions $\mathbf{x}_i \in \mathbb{R}^3$ and triangles $t_j \in \mathbb{N}^3$ is naturally parameterized. An important practical problem is to find a parameterization that is continuous across interior edges. To accommodate this, we assume the mesh is manifold with boundary, and triangles adjacent in the triangulated surface are also adjacent in the parameter domain. Then the parameterization

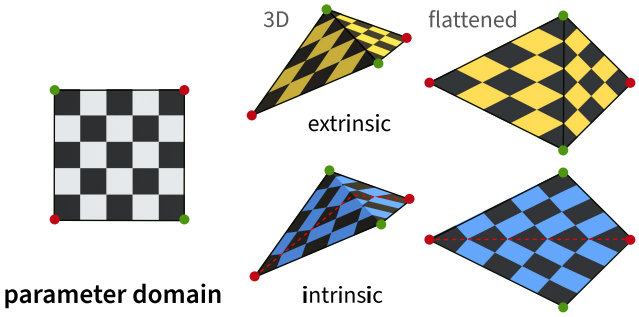

Figure 1: *Four points on the surface with a given parameterization yield two piecewise linear interpolating mappings.*

$\mathbf{p}^{-1}$ of the surface $\mathcal{S}$ may be specified by only specifying the parameter locations $\mathbf{u}_i \in \mathbb{R}^2$ of the vertices and linear interpolation on edges and inside triangles. This procedure implies that the Jacobian $\mathbf{J_p}$ is constant in each triangle. As a practical benefit, the distortion energy can be computed per triangle (see Section 2 for more details).

As natural as this approach may seem, a consequence is that computing a parameterization by minimizing a distortion energy not only depends on the geometry of the surface, but also its triangulation. In fact, basing the quality measure on the Jacobian $\mathbf{J_p}$ is *intrinsic*, but by assuming it is constant for each triangle in the given triangulation we lose this property. The dependence on the

† equal contribution

triangulation for discrete differential operators whose continuous counterparts are intrinsic has motivated the exploration of *intrinsic triangulations* [BS07]. A piecewise linear (PL) surface is intrinsically flat everywhere except (possibly) in the vertices. This suggests that the vertex set on the PL surface could be triangulated in different ways, with *intrinsically* flat triangles. Concretely, a triangle in an intrinsic triangulation may consist of several flat pieces that are not in a common plane in 3D, but can always be unfolded across the original edges into a planar triangle. One way to explore the different intrinsic triangulations of a given PL surface is by starting from the original triangulation and then flipping an edge in an intrinsically flat quadrilateral formed by two adjacent triangles (if the quadrilateral is intrinsically convex) [FSSB07, SC20, SSC19]. It is important to note here that *every* quadrilateral created by two intrinsically flat triangle is intrinsically flat, so the flipping operation is not restricted to planar areas of the surface. As every intrinsic triangulation can be flipped into the intrinsic Delaunay triangulation, intrinsic triangulations of PL surfaces are *flip-connected*, just like triangulations in the plane [Law72, DGR93, OB08]. This means that all intrinsic triangulations can be explored by flipping, starting from the given triangulation of the surface.

It remains true for intrinsic triangles that the parameterization can be specified using the locations of the vertices in the plane and interpolating linearly across the triangle, as can be seen in Fig. 1. The only caveat is that the intrinsic triangle may be difficult to recover from the vertices alone and is better stored using appropriate data structures [FSSB07, SSC19, GSC21]. The Jacobian $\mathbf{J_p}$ is only piecewise constant in this case, but because the triangle is intrinsically flat, linear interpolation leads to constant singular values across the triangle. In other words, while the different orientations of the flat pieces of an intrinsic triangle lead to different Jacobians, they only differ by an orthogonal transformation accommodating the different orientation, while the singular values are all the same. This means that distortion energies based on the singular values of the Jacobian can still be computed per complete (intrinsic) triangle, without the need to decompose the intrinsic triangle into pieces of extrinsic geometry. All we need are edge lengths, and this is exactly what data structures for intrinsic triangulations readily provide.

Our central observation is that when parameterizing a piecewise linear surface, one should minimize a distortion energy not only over the possible parameter locations $\mathbf{u}_i$ in the plane, but also over the space of intrinsic triangulations. While the former space is typically explored by making small adjustments to the locations, the latter discrete space can be explored by intrinsic edge flipping. This leads to the idea of additionally flipping the edges during minimization of the energies, resulting in an alternating discrete/continuous minimization approach. In Section 4 we discuss the details of intrinsic flipping in this context.

We evaluate the resulting optimization approaches in Section 5. As expected, considering the discrete space of intrinsic triangulations leads to generally better parameterizations. Based on these observations, we additionally analyze the idea of starting the optimization with the intrinsic Delaunay triangulation (see Section 6), which turns out to be optimal for the linear energies and less suitable for the non-linear energies. We discuss these results and further possibilities for exploration in Section 7.

2. Background

We begin with an overview of well-established distortion energies for parameterizations to which we will later apply the method. Since the method also assumes a basic knowledge of intrinsic triangulations we refer the unfamiliar reader to the respective course material [SGC21]. But in the interest of completeness, we will briefly explain a simple storage option of intrinsic triangulations and how to perform a basic operation on it.

2.1. Distortion Energies

Assume a triangle mesh is given as a set of triangles $\mathcal{T} = \{(i,j,k) \in \mathbb{N}^3\}$ and vertex positions $\mathbf{X} = \{\mathbf{x}_i \in \mathbb{R}^3\}$. For simplicity and compatibility with the methods below we assume the mesh has disk topology. A parameterization is given by an assignment of parameter locations $\mathbf{P} = \{\mathbf{p}_i \in \mathbb{R}^2\}$ and using linear interpolation inside the fixed triangles $\mathcal{T}$. The parameter locations are commonly computed by minimizing a distortion energy, possibly constraining some of the parameter locations to avoid degenerate minimizers.

In the following we briefly recap widely used distortion energies and their relation to the singular values of the Jacobian. More detailed treatment and analysis can be found in Rabinovich et al. [RPPSH17] and, partly, the original works cited below. Distortion energies, conceptually, are modeled as integrals of a function of the first-order derivatives, the Jacobian matrix $\mathbf{J}$. Assuming linear (barycentric) interpolation from the parameter domain to the surface inside the triangles, the Jacobian in each triangle and we write $\mathbf{J}_t$ for the Jacobian in triangle t. When computing the parameterization it is more convenient to work with the inverse $\mathbf{J}_t^{-1}$, since the mapping from a non-degenerate triangle t of a given triangle mesh to the parameter domain is linear in the parametric vertex positions $\mathbf{p}_i$, so it can be computed efficiently for any parameterization $\mathbf{P}$. Concretely, for triangle $t = (i,j,k)$, the inverse Jacobian is

$$\mathbf{J}_t^{-1} = (\mathbf{p}_j - \mathbf{p}_i,\ \mathbf{p}_k - \mathbf{p}_i)(\mathbf{x}_j - \mathbf{x}_i,\ \mathbf{x}_k - \mathbf{x}_i)^+, \tag{1}$$

showing the linear dependence on the parameter locations; the pseudo-inverse can be pre-computed for the fixed geometry $\mathbf{X}$ of the triangle mesh.

The singular value decomposition $\mathbf{J}_t^{-1} = \mathbf{U}_t\mathbf{\Sigma}_t^{-1}\mathbf{V}_t^\mathsf{T}$, where $\mathbf{\Sigma}_t^{-1}$ is a 2×2 matrix, allows us to formulate distortion energies as

$$\mathcal{E}(\mathbf{P}) = \sum_t a_t \mathcal{D}(\mathbf{\Sigma}_t^{-1}), \tag{2}$$

where a_t is the area of t in the input mesh and $\mathcal{D}$ is the distortion function. The energies differ based on the definition of $\mathcal{D}$. We consider several common choices, as discussed in the following.

Harmonic energy. The harmonic energy of a function, also called Dirichlet energy, is the integral of the squared gradient of that function. For PL functions over triangulations it can be computed using the *cotan*-Laplacian [PP93]. Minimizing the harmonic energy for parameterization leads to *discrete harmonic maps* [EDD*95]. In our setting it can be written as

$$\mathcal{D}_\mathrm{D} = \frac{1}{2}\|\mathbf{\Sigma}_t^{-1}\|_F^2 = \frac{1}{2}\|\mathbf{J}_t^{-1}\|_F^2. \tag{3}$$

The harmonic energy scales with area, so it vanishes for a triangle whose vertices collapse into a point. For minimization, it is necessary to constrain the boundary of the mesh to a fixed polygon. With fixed boundary, the energy can be minimized by solving a linear system based on the cotan-Laplacian. In general, the minimizer is not guaranteed to map all triangles to planar triangles with positive area. The following energies are related to harmonic energy, but improve in terms of requiring a fixed boundary and/or inverted triangles.

Conformal energy. The undesirable scaling behavior of the harmonic energy can be rectified by considering the areas of the triangles, leading to what has been referred to as *conformal* energy. In the discrete case this can be done simply by subtracting an area term for the triangle [LPRM02, DMA02]:

$$\mathcal{D}_{\mathrm{C}} = \frac{1}{2}\|\mathbf{\Sigma}_t^{-1}\|_F^2 - \det(\mathbf{\Sigma}_t^{-1}). \tag{4}$$

Now the only degrees of freedom are similarity transformations, which can be removed by constraining some parameter values, typically two vertices on the boundary. The problem in this form is quadratic and remains convex even in more general variants [DMA02]. An optimal boundary in the sense of the energy can be computed by solving a generalized eigenvalue problem [MTAD08]. We use the quadratic version, which only requires the solution of a linear system. The resulting parameterization may not be bijective, because the boundary can intersect itself. Additionally, the energy scales linearly with the area of the resulting parameterization. To compare different parameterizations of the same mesh we divide the energy by the area of the parameterization.

Symmetric Dirichlet energy. To avoid degenerate triangles in the parameter domain, one can consider the mapping in both directions [SS15]:

$$\mathcal{D}_{\mathrm{S}} = \begin{cases} \frac{1}{2}\|\mathbf{\Sigma}_t^{-1}\|_F^2 + \frac{1}{2}\|\mathbf{\Sigma}_t\|_F^2 & \det(\mathbf{\Sigma}_t^{-1}) > 0 \\ \infty & \text{else} \end{cases}. \tag{5}$$

This guarantees that all parameterized triangles are positively oriented. The energy is non-linear, and is commonly minimized using a descent scheme starting from a bijective parameterization (for example computed using Tutte's embedding [Tut60]).

As-rigid-as-possible (ARAP) energy. One may argue that the ideal Jacobian only rotates, and measure the deviation from rotation in the Frobenius norm [SA07], leading to the following energy in the context of parameterizations [LZX*08]:

$$\mathcal{D}_{\mathrm{R}} = \frac{1}{2}\|\mathbf{\Sigma}_t^{-1} - \mathbf{I}\|_F^2. \tag{6}$$

In our context we could write the energy also as

$$\mathcal{D}_{\mathrm{L}} = \frac{1}{2}\|\mathbf{\Sigma}_t^{-1}\|_F^2 - \mathrm{tr}(\mathbf{\Sigma}_t^{-1}), \tag{7}$$

which is equivalent up to an irrelevant constant term.

The symmetric Dirichlet and ARAP energies are non-linear and non-convex and are minimized by iterative descent approaches. This means the solution is a local minimum of the energy and no guarantee can be made about its distance to the global minimum.

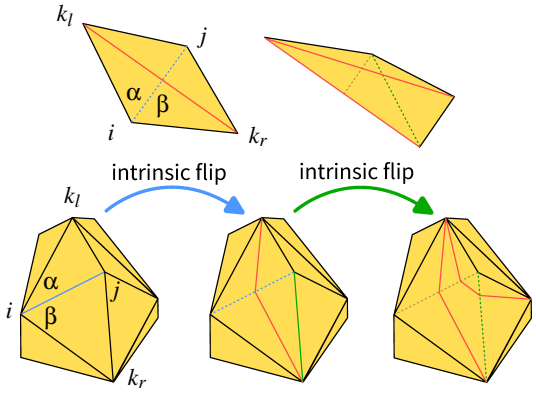

Figure 2: *Intrinsic flips create piecewise linear edges on the mesh surface. Red lines are the intrinsic edges, the dotted lines are the input edges that were flipped. To flip edge (i,j) to (k_l,k_r) we need to calculate the new intrinsic edge length $\|(k_l,k_r)\|$. This operation can be chained. Here we first flip the blue edge then the green one.*

2.2. Exploring Intrinsic Triangulations

Assume a 2-manifold triangle mesh $M = (\mathcal{T},\mathbf{X})$. We can store the intrinsic geometry of M by $\mathcal{S} = (\mathcal{T},\mathbf{L})$ where $\mathbf{L}$ contains all edge lengths in $\mathcal{T}$. We can explore the space of all intrinsic triangulations just by flipping edges in $\mathcal{S}$. This can be done without further knowledge about $\mathbf{X}$. To perform a flip we first need to check if the flip is possible. Then we need to update the edge lengths and the triangulation.

Consider an (intrinsic) edge $e = (i,j)$, the surrounding triangles $t_l = (i,j,k_l)$ and $t_r = (j,i,k_r)$ and their respective edge lengths. Since the triangles are intrinsically flat we can map them into the plane using just the edge lengths as a representation of the local intrinsic space. An intrinsic flip can then only be performed if the edge is not on the boundary and the two neighboring triangles t_r and t_l form a convex quadrilateral. Given that the angles of the triangles can be calculated using the cosine law, we only need to check that no angle of the quadrilateral is greater than π. Note that the angle incident to i and j consist of two triangle angles. To perform the flip we just need to update the connectivity e in $\mathcal{T}$ to $e = (k_l,k_r)$ and calculate the new edge length in $\mathbf{L}$. As we already calculated the angles α and β that are incident to i, we can simply calculate the new edge length of the other diagonal of the quadrilateral. For a complete derivation we refer to the literature (e.g. [FSSB07]). This process can be continued with other (including intrinsic) edges to explore the space of intrinsic triangulations. An illustration is provided in Fig. 2.

Recall that the (inverse) Jcaobian $\mathbf{J}_t$ can be computed from thee edges of triangle t (cf. Eq. (1)). Moreover, the singular values are independent of rotations, so the edge lengths are sufficient to compute the singular values of $\mathbf{J}_t$. make that we only need the edge lengths to compute the (inverse) Jacobian. This means that a data structure that only stores the edge lengths of the intrinsic triangles (as mentioned above) is enough to compute and optimize all distortion energies we consider. On the other hand, for visualizing the

intrinsic triangulation and the parameterization results, we need to be able to map a surface point to its corresponding intrinsic triangle. This requires more information, as we need to compute the common subdivision of the intrinsic triangles and the underlying surface triangulation. There are different approaches in the literature [FSSB07, SSC19, GSC21] – in our implementation we use the one of Sharp et. [SSC19].

3. Related Work

In the following we briefly relate our method to energy minimizing methods in parameterization, energy minimizing flipping, and the use of intrinsic triangulations in the context of the ARAP energy.

Distortion minimizing 2D parameterizations. The range of distortion-minimizing parameterization methods is too large for a detailed review. The basic setup and most useful energies have been discussed in detail in excellent overview papers [FH05, HLS07, SPR06]. Recent research in this domain mostly focuses on efficient, smooth and inversion-free maps with arbitrary boundaries, but is still largely based on the same energies.

Efficient optimization usually either relies on minimizing a convex energy function [Tut60, Flo03, LPRM02, DMA02] or exploits the structure of the energy [CBSS17, RPPSH17, LZX*08, BN21, SS15]. Although many methods are guaranteed to be injective if they reach a minimum in real numbers, this cannot be guaranteed in floating point arithmetic, yet carefully penalizing inverted triangles seems to work well in practice [GKK*21]. Strict guarantees appear to come at the expense of giving up the minimization of a distortion energy [LYNF18, FBRCA23]. The combination of natural non-convex boundaries and guarantees on injectivity are generally non-trivial [WZ14, LYNF18, DAZ*20].

Our method is orthogonal to these goals. In fact, we rely on the successful past development of efficient, robust energy minimization as a framework and enhance the space of possible mappings of the given piecewise linear surface to the plane by considering more than the given triangulation of the input.

Harmonic energies in triangulations. Given a set of points and associated function values, one may define a piecewise linear (PL) function by triangulating the points. In this setting we could optimize over the set of different triangulations to minimize a measure of variation. As shown by Rippa [Rip90], the Delaunay triangulation leads to minimal harmonic energy *regardless of the function values at the vertices*. This property carries over to the intrinsic triangulations of piecewise linear surfaces immersed in 3D [BS07] and led to the definition of an intrinsic Laplace-Beltrami operator for PL surfaces. This operator has turned out to be beneficial in a variety of applications [CWW17, SGC21]. For dimensions higher than two, Rippa's theorem does not hold, yet it may still be beneficial to perform flips that minimize the harmonic energy and combine this with optimization of vertex locations [Ale19].

Our main idea is related in that we also approach an energy minimization problem by including the exploration of different triangulations through flipping. However, our method rather considers the image of the function when exploiting flipping.

Measuring ARAP intrinsically. Finnendahl et al. [FSA23] suggest to considering the ARAP energy of the intrinsic Delaunay triangulation rather than the extrinsic triangulation. Their motivation is that the ARAP energy as suggested by Sorkine & Alexa [SA07] is non-negative only for Delaunay triangles, and using the intrinsic Delaunay triangulation guarantees positive energy while retaining the original PL geometry. Since the ARAP energy requires computing local transformations of the embedding, they need to explicitly construct the subdivision of explicit and intrinsic triangles, as the transformations vary across triangles.

While we similarly measure energies on intrinsic triangulations, all the energies are defined intrinsically and can be computed based on the edge lengths of triangles. Concretely, we never need to intersect intrinsic against extrinsic edges for the energy minimization. We only compute the intersection for visualization purposes, as this allows exploiting the standard graphics pipeline for texturing.

Refining Finite Elements. A parameterization of a surface is a function over the surface. This function can be discretized using a finite element basis. Typically, a parameterization of a PL surface maps the vertices to the plane and assumes that the interior is mapped by linear interpolation of the values at the vertices of each triangle on the surface. Thus, a discrete parameterization can be thought of as coefficients for the standard *"hat" basis*, i.e. *piecewise linear* basis functions defined on the triangles.

Any given discretization can generally only approximate the minimizer of continuously formulated problem, and the error in this approximation depends on the choice of basis. It is common in finite elements to reduce the error by *refinement* methods, either refining the number of elements, i.e., subdividing the given triangles (so-called *h*-refinement); or increasing the polynomial degree of the basis functions on the elements (*p*-refinement); or both (*hp* refinement) [BS94]. Where and how to refine can be decided either *a priori*, before calculating the approximation, or *a posteriori*, after the approximation. Such approaches are extensively studied and used in engineering problems, and have also helped to deal with badly triangulted surfaces in geometry processing [SHD*18].

For parameterizations, higher order basis function have only recently been considered [MC20]. In this context we also like to mention parameterizations of subdivision surfaces [DKT98, HSH10, dGDMD16], although this is slighlty different in that the mapping assumes a change of the underlying geometry alongside the refinement.

Our approach can be viewed as a search for a new piecewise linear finite element basis over the surface, using intrinsic instead of the given explicit triangle as the supported regions of the basis functions. Unlike *p*- and *h*-refinement, we improve the approximation of the optimal mapping without adding degrees of freedom, since we still rely on a "hat" function per vertex – and the number of vertices stays *constant*. Applying the same perspective to previous work based on intrinsic triangulations shows that so far only the hat functions induced by the *intrinsic Delaunay triangulation* [BS07], or a-priori *h*-refined versions of it [SSC19], have been considered. We extend these approaches by exploring a larger space of intrinsic triangulations a-posteriori.

4. Method: Alternating Energy Minimization

Similar to the setup in Section 2, we assume a triangle mesh with disk topology is given as the pair $(\mathcal{T},\mathbf{X})$ and we want to compute a parameterization by minimizing a distortion energy. The important change is that, in addition to the parameter locations, we now optimize over the combinatorial space of intrinsic triangulations admitted by the input mesh. This means the output of the optimization is the pair $(\mathcal{S},\mathbf{P})$, specifying the triangulation in the parameter domain. To perform our energy minimization, S only needs to store the edge lengths and connectivity of the intrinsic triangulation, as this information is sufficient to perform intrinsic edge flips and compute the Jacobians [BS07, FSSB07]. Although extensive data structures [FSSB07, SSC19, GSC21] are not required for the optimization we still recommend to use a data structure that can compute the common subdivision between the surface edges and the intrinsic mesh (see Section 7 for a brief discussion). The mapping is then given by combining the linear maps from the (straight-edge) triangles in the parameter domain to the unfolded, straight-edge intrinsic triangles of the input geometry. An example with two triangles is given in Fig. 1.

Intrinsic triangulations can be explored by flipping. Similarly to minimizing a distortion energy w.r.t. the parameter locations iteratively in a descent scheme, we want to perform intrinsic edge flips to decrease the distortion energy. For the minimization we suggest to alternate between optimization of $\mathbf{P}$ and $\mathcal{S}$. We start with $\mathcal{S}_0=\mathcal{T}$ and compute (locally) optimal parameter locations $\mathbf{P}_0$ using any of the established methods. Then, for fixed geometry $\mathbf{P}_0$, triangulation $\mathcal{S}_1$ is computed by performing flips that decrease the energy. This may enable further optimization of the parameter locations, leading to $\mathbf{P}_1$. The steps are repeated until there are no more flips that decrease the energy, i.e. $\mathcal{S}_{i+1}=\mathcal{S}_i$. Pseudo-code for this approach can be found in Algorithm 1.

Admissible, energy decreasing flips. Each flip has to be considered in both the input geometry and the current triangulation. We call the flip of edge $e=(i,j)$ *admissible* if the two triangles $t_l=(i,j,k_l)$ and $t_r=(j,i,k_r)$ incident on e form a convex quadrilateral – in the parameter domain as well as in the intrinsic triangulation in 3D. The quadrilateral is convex, if all its interior angles are smaller than π. The quadrilateral is composed of two triangles, so it suffices to check the angles of the triangles opposite of the common edge; and the sum of the interior triangle angles at the shared vertices i and j. Note that this check in the parameter domain also ensures that the triangulation of the parameterization has no self intersections, so necessarily remains planar. In particular, self-loops or interior vertices with degree less than three, which are possible in intrinsic triangulations [BS07, SGC21], would create angles larger than π in the parameterization and are avoided by the convexity condition.

For an admissible flip we can compute the change in distortion energy. Since the energy is the sum of per-triangle energies (Eq. 2) and a flip has no effect on triangles other than t_l,t_r, it suffices to compare the per-triangle energies of t_l,t_r with the two triangles $t_i=(i,k_l,k_r)$ and $t_j=(j,k_r,k_l)$ generated by the flip. This means a flip is energy decreasing if

$$\Delta\mathcal{E}_{ij}=a_{t_i}\mathcal{D}\left(\Sigma_{d_i}^{-1}\right)+a_{t_j}\mathcal{D}\left(\Sigma_{d_j}^{-1}\right)-a_{t_l}\mathcal{D}\left(\Sigma_{d_l}^{-1}\right)-a_{t_r}\mathcal{D}\left(\Sigma_{d_r}^{-1}\right) \tag{8}$$

is negative. This expression can be calculated using only the connectivity and the edge lengths of the intrinsic triangulation. Note that such a flip does not necessarily reduce the energy for each of the four incident triangles of the common subdivision, as can be seen in Fig. 5.

ALGORITHM 1: Optimize embedding

```
Input  : T
Output: P_i, S_i
S_0 := T
P_0 := Parameterize(S_0, E)          // Minimizes E(S_0)
i := 0
q := Queue()                         // Create an empty queue
for e ∈ T do
    Enqueue(q, e)                    // Enqueue all edges
end
decreased := true
while decreased do
    decreased = false
    while q ≠ ∅ do
        e := Pop(q)  // Gets and removes the first
         e in q
        S_tmp := IntrinsicFlipIfPossible(S_i, e)
        if E(S_tmp) < E(S_i) then
            S_{i+1} = S_tmp
            for d ∈ D_e do
                /* D_e contains all other edges of
                   the diamond around e.        */
                Enqueue(q, d)
            end
            decreased = true
        end
    end
    if decreased then
        P_{i+1} = Parameterize(S_{i+1}, E)
        i = i +1
    end
end
```

Flipping order. In general, for a given parameterization $(\mathcal{S}_i,\mathbf{P}_i)$, more than one flip can decrease the energy. On the other hand performing a flip may (1) invalidate other admissible, energy decreasing flips and (2) enable additional ones that were inadmissible before. So the question is whether certain orders for the flips lead to better local minima.

An obvious choice for prioritizing flips is the decrease in energy, i.e., the magnitude of $\Delta\mathcal{E}_{ij}$. More complicated heuristics are possible, potentially looking into the values of invalidated/enabled flips. On the other hand, prioritizing the flips requires maintaining a priority queue (ideally with mutable priorities because of the changes mentioned above), incurring extra costs. Without prioritizing the flips, we can just walk over the edges in any convenient order, flip if admissible and decreasing the energy, until no such flips are left.

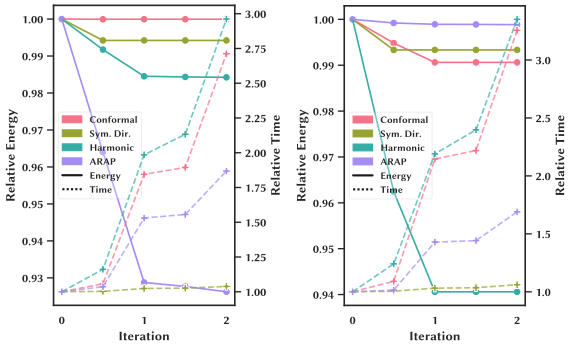

Figure 3: *The relative energies and relative times it took for two meshes to converge using our method optimizing different standard energies. The used meshes are #106620 (left) and #222747 (right) of the Thingi10k dataset [ZJ16].*

Table 1: *Energy decrease and computation time for alternating optimization. Energies and times are relative to optimization without flipping. Statistics (mean, median, and percentiles) are based on the relative numbers and are taken over the data set of 756 triangulated surfaces. The 95% percentile for the relative energy and the 5% percentile for the relative time are close to 1 and omitted.*

	Relative energy			Relative time		
	avg	med	5%	avg	med	95%
Harmonic	0.912	0.995	0.481	2.8	2.7	4.4
Conformal	0.945	0.995	0.746	1.4	1.4	2.0
Sym. Dir.	0.993	1.000	0.995	1.0	1.0	1.1
ARAP	0.821	0.934	0.137	2.2	1.5	5.3

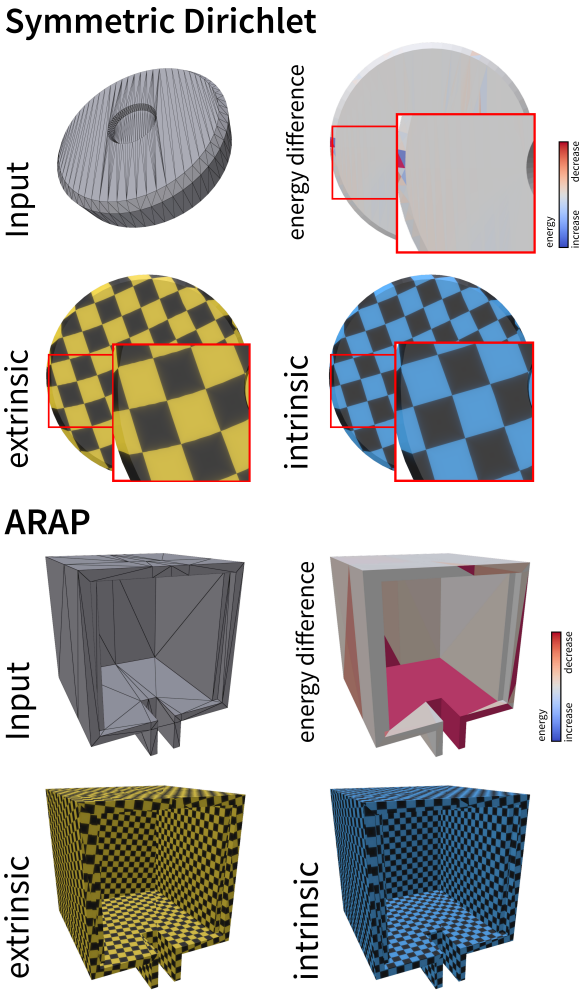

Figure 4: *Two meshes (#103357 and #49902 from Thingi10K [ZJ16] in grey) are parameterized using symmetric Dirichlet (top) and ARAP energy (bottom). Although the energy decreases by a factor of only 0.999 at the top we can clearly see the improved straight lines in the planar region. On the bottom, the energy has decreased by a factor of 0.074, but besides some overall improved alignment, the individual triangles do not seem to have improved that much.*

5. Exploration

In the following we explore the idea laid out in the previous section on a set of example meshes. This set is the result of querying Thingi10K [ZJ16] for (a) manifold triangulated topological disks, and (b) manifold topological spheres, which were cut using the libigl [JP*18] implementation of the mixed-integer quadrangulation cut graph [BZK09]. We filtered these meshes further such that every mesh has no flipped triangle after applying any of our parameterization methods. Together this results in above 756 test cases.

Our implementation of parameterization with intrinsic flipping is based on libigl [JP*18] and geometry central [SC*19]. The latter supports mapping between the intrinsic and extrinsic triangulation [SGC21], which we use for visualizing the resulting parameterizations based on regular textures.

In the exploration we collect data for the basic idea of alternating energy minimization for the four energies described in Section 2 and address the following aspects:

Energy: How much energy decrease is possible with intrinsic flipping?

Timing: How do the execution times compare between minimizing only the positions and additionally alternating between flipping and further decreasing the energy?

Visual differences: How does the parameterization look like? Are the differences between the reference method and our improvement visible?

Visualization overhead: By what factor is the mesh size increased when adding vertices required for expression of the common subdivision? How does the time compare to optimization time?

Alternation strategy: Is it better to always exhaust one of the two ways to decrease the energy? Or should we prioritize flipping or modifying the parameter locations?

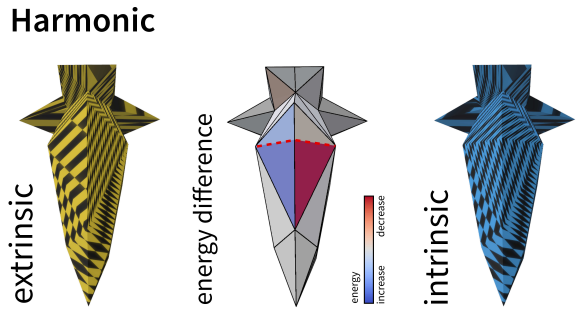

Figure 5: *We minimize the harmonic energy of the mesh #89912 of the Thingi10k dataset [ZJ16]. The edge hinge on top got flipped (dashed line). The overall energy decreased (red) although it increased in some areas (blue).*

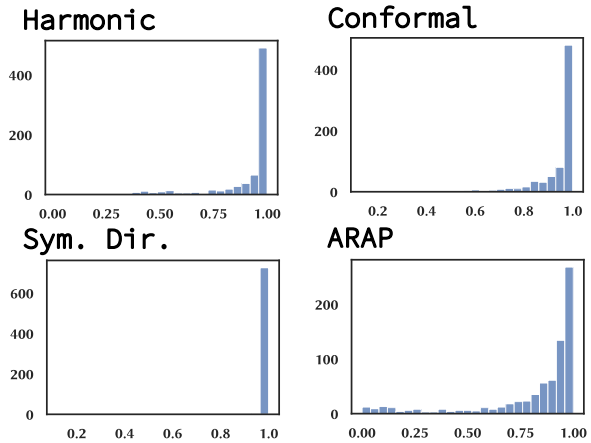

Figure 6: *Extension of Table 1. Histogram of the ratio distribution of intrinsic energy to extrinsic energy for all meshes. Symmetric Dirichlet energy shows the least improvement.*

Flipping orders: Does preferring flips that decrease the energy most lead to larger total energy decrease?

Energy and time. In our standard setting, we optimize the parameter positions until convergence and then exhaust energy decreasing flips, without ordering the flips in any way. We perform this optimization over the set of triangle meshes and compare energy and time to optimizing positions for the starting extrinsic triangulation. A typical decrease in energy is shown in Fig. 3 and the reduced texture distortion is visualized in Fig. 7. The improvement is measured as the ratio of resulting intrinsic energy and initial extrinsic energy. So, smaller ratios are better; for example, a ratio of 0.912 means the energy improved by 8.8%. A short summary of the statistics provided in Table 1 is visualized as histograms in Fig. 6.

We find that the energy can be decreased by flipping in almost all cases, more significantly for harmonic and ARAP energy, and marginally for symmetric Dirichlet energy. Interestingly, despite the nominally small decrease in energy, the differences are clearly visible also for symmetric Dirichlet, as shown using checkerboard textures in Fig. 9 and Fig. 4 (top). This could be explained by the fact that a flip, although it decreases the overall energy, usually also increases the energy of neighbouring triangles as can be seen in Fig. 5. On the other hand, we also noticed that the decrease in energy may be large, but the improvement, while visible, is moderate at best – Fig. 4 (bottom) shows an example.

Table 2: *Vertex count ratios of the common subdivision to the input mesh, and computation times for constructing the subdivided mesh relative to the alternating optimization. The 5%-percentiles are omitted, since they correspond to cases where there are very few flips, so the vertex ratios are 1 and relative times are close to 0.*

	Vertex Ratio			Relative time		
	avg	med	95%	avg	med	95%
Harmonic	1.4	1.2	2.7	0.6	0.3	1.5
Conformal	1.4	1.1	2.8	0.4	0.3	0.2
Sym. Dir.	1.5	1.1	3.8	0.0	0.0	0.0
ARAP	2.3	1.5	6.3	0.0	0.0	0.1

Table 3: *Absolute time (in seconds) for selected meshes of the intrinsic optimization (Opt.) and creating the common subdivision (CS) using the resulting intrinsic triangulation.*

	Har.		Con.		Sym. Dir.		ARAP	
Mesh ID	Opt.	CS	Opt.	CS	Opt.	CS	Opt.	CS
471990	0.9	0.4	0.8	0.3	306	0.6	101	0.9
79193	0.4	0.1	0.4	0.2	132	0.1	33	0.2
100388	0.7	0.3	0.7	0.3	269	0.2	74	0.4
133568	3.4	1.1	4.2	1	1226	0.9	603	1.3
518083	11	3	6	2.6	5849	2.7	312	3.3

For harmonic energy, the relative increase in time is significant, because each iteration in our implementation requires a new linear solve. For minimizing conformal energy, we use the Gauss-Seidel method to approximate the result of the linear system after the first minimization w.r.t. extrinsic geometry, making the relative increase in time lower than for harmonic energy. We also tried to use Gauss-Seidel for harmonic energy, but we observed that the time to convergence was worse than using CHOLMOD [CDHR08]. We found a simpler way to reduce the extra cost as discussed in the following section for these energies. For ARAP the increase in time is moderate, and for symmetric Dirichlet almost negligible, as the bulk of the optimization time is spent on computing the initial local minimum starting from a Tutte embedding.

Prioritization. Based on the results for the standard setup we tried different strategies for prioritizing parameter location optimization and flipping. As expected, it is best to optimize the energy until completion. On average, applying flipping after 50 iterations of location optimization for ARAP results in an energy difference smaller than 10^{-4}, and for symmetric Dirichlet of about 0.05. Likewise, prioritizing flips that have larger $\Delta\mathcal{E}$ (in magnitude) is not significantly affecting the results (the resulting average and median relative energy differences are smaller than 10^{-3}), so we suggest to simply flip in the most convenient order. In Fig. 8, we show a his-

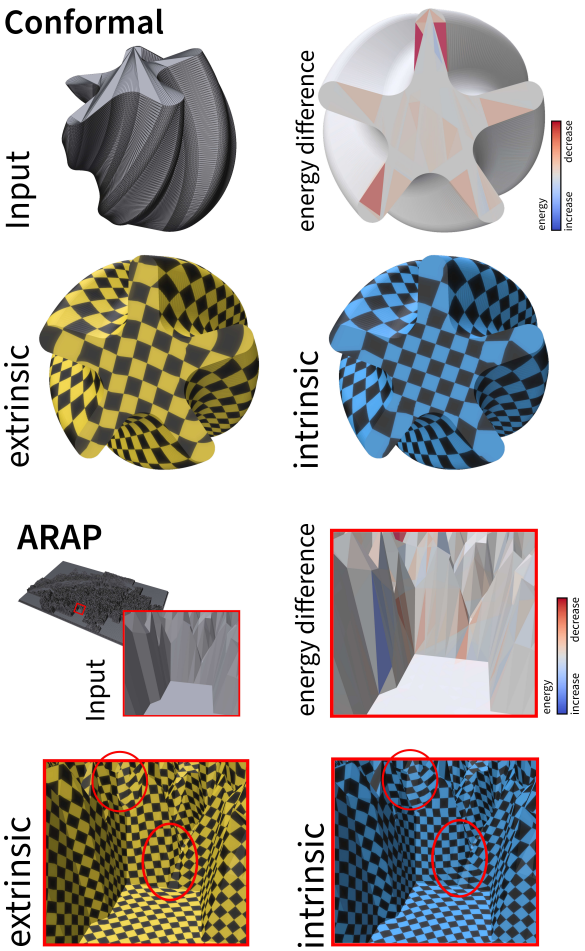

Figure 7: *Two meshes (#222694 and #106620 from Thingi10K [ZJ16] in grey) are parameterized using conformal (top) and symmetric Dirichlet energy (bottom). The left parameterization is the standard result. Intrinsic flipping further optimizes mapping, apparent especially in planar regions.*

togram of resulting energies over different random orders of flips. We also find that the majority of energy reduction occurs in the first two iterations. The effect of additional iterations is not noticeable in numbers and visualizations, see Fig. 9.

Common subdivision. To visualize the texture, we need to intersect the intrinsic triangulation resulting from the optimization with the extrinsic triangulation of the input mesh. To do this, we compute the common subdivision of the input mesh and the intrinsic mesh. The number of additional vertices introduced by the common subdivision depends on the number of intrinsic flips. The time for computing the subdivisions are negligible compared to the optimization times for ARAP and symmetric Dirichlet energies. For minimizing conformal and harmonic energies, the relative time varies depending on the number of vertices in the common subdivision. For our data set, computing the common subdivision required on average 10% of the time for optimization, see Table 2.

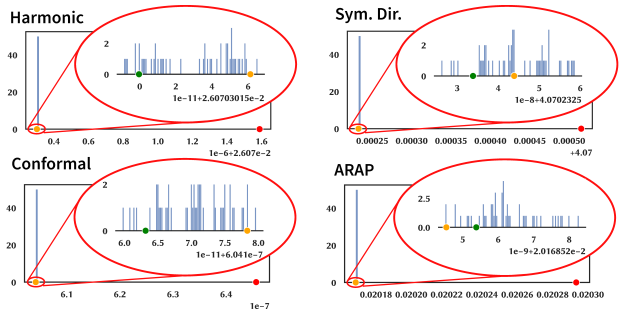

Figure 8: *Instead of flipping in some fixed order in each iteration of alternating minimization (orange point), we run 50 optimization with randomized order of flips. The histograms show the distribution of the energy for Mesh #1411702 from Thingi10K [ZJ16]. Red points correspond to optimization without flipping. Green points are the resulting energies when flips are prioritized based on energy decrease. The order of flipping appears to have negligible impact on the effect of flipping.*

6. Initialization

In our exploration we made an important observation: For harmonic and conformal energy, every energy decreasing flip is an intrinsic Delaunay flip, i.e. turns an edge that is not intrinsically Delaunay into an intrinsic Delaunay edge. Based on this observation we verified that, conversely, an intrinsic Delaunay flip never increases the energy. We will prove this observation, but perhaps more importantly, it suggests that we could have performed energy decreasing flips independent of the parameterization. Concretely, for harmonic and conformal energy it may seem promising to first establish an intrinsic Delaunay triangulation of the input and use this as the initial triangulation. Then we would know that no further flips during the minimization will be necessary, and in this case a single pass of optimizing the parameter locations suffices to find the optimal solution. Alas, the situation is not as easy, because we have to restrict the triangulations $\mathcal{S}$ to topological disks, more specifically, to 3-connected planar triangle graphs. Intrinsic Delaunay flips are known to create self loops around small structures such as cones [BS07, SGC21]; and in the process they might also reduce vertex degrees below three.

For other deformation energies it is not even clear how flips could be characterized that decrease the energy independently of the parametric realization, or at least for 'expected' parameterizations. We can nonetheless ask: what flipping strategy / heuristic could be used to generate an *initial* intrinsic triangulation that leads to faster decrease in energy? Such a different starting point might lead to faster convergence for the optimization of parameter locations, or might result in offering a good compromise when using the parameterization result after a single step of the procedure.

Intrinsic Delaunay flips. We observed that intrinsic Delaunay flips are energy decreasing for harmonic and conformal energy. To see why this has to be true, consider the parameter locations $\mathbf{P}$ as a rectangular matrix with the two rows $\mathbf{p}_u, \mathbf{p}_v$. Then harmonic energy

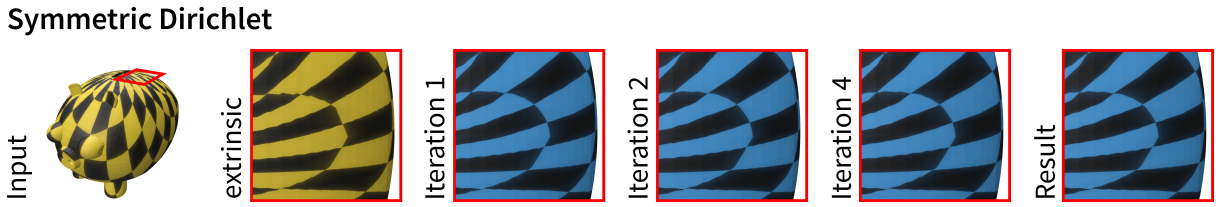

Figure 9: *Given the input mesh (#222747 from Thingi10K [ZJ16]) we optimize for the symmetric Dirichlet energy [SS15]. The difference after the first iteration are barley visible.*

(plugging in Eq. 3 into Eq. 2) can be written as

$$\mathcal{E}(\mathbf{P}) = \mathbf{p}_u^\mathsf{T}\mathbf{L}_\mathcal{T}\mathbf{p}_u + \mathbf{P}_v^\mathsf{T}\mathbf{L}_\mathcal{T}\mathbf{P}_v, \tag{9}$$

where $\mathbf{L}_\mathcal{T}$ is the cotan-Laplacian of the (intrinsic) triangulation $\mathcal{T}$. Now consider a triangulation $\mathcal{T}'$ that arises from $\mathcal{T}$ by a single flip. If the flip that generates $\mathcal{T}'$ from $\mathcal{T}$ is a Delaunay flip, then necessarily $\mathbf{f}^\mathsf{T}\mathbf{L}_{\mathcal{T}'}\mathbf{f} \leq \mathbf{f}^\mathsf{T}\mathbf{L}_\mathcal{T}\mathbf{f}$ for any vector of function values $\mathbf{f}$ [Rip90, BS07]. Using this observation for the energy in Eq. 9 above, i.e. using $\mathbf{p}_u$ and $\mathbf{p}_v$ as $\mathbf{f}$, shows that intrinsic Delaunay flips never increase the harmonic parameterization energy.

For conformal energy we have to additionally consider the area term $\det(\Sigma_t^{-1})$ for the triangles (see Eq. 4). Note that this expresses the change in area from the intrinsic triangulation to the parameter domain. Since this term is multiplied by a_t in Eq. 2, it measures the area of the triangle in the parameter domain. For a flip we sum up over the two triangles t_l, t_r, so in total this measures the area of the quadrilateral spanned by the two triangles. The same is true after the flip, so this part of the energy is irrelevant for flipping – and it follows that intrinsic Delaunay flips are also optimal for minimizing the conformal energy.

Intrinsic Delaunay triangulations as initialization. Knowing that we can perform intrinsic Delaunay flips and this generally decreases harmonic energy for any realization in the parameter domain suggests performing such flips a-priori. As mentioned, in our context there is still a topological restriction: the flips have to preserve the planarity of the triangle graph and maintain a vertex degree of at least three. We augment Delaunay flipping with this restriction and call the result a *planar* intrinsic Delaunay triangulation, denoted $\mathcal{S}^\circ$.

While intrinsic Delaunay triangulations (assuming non-degenerate input) for a given PL surface of arbitrary topology are unique [BS07], the restricted planar version $\mathcal{S}^\circ$ is not: consider the space of (intrinsic) triangulations of $(\mathcal{T}, \mathbf{X})$ as the nodes of a connected graph. Two nodes are connected by an edge in this graph if they are related by a flip (this is the so-called flip-graph of triangulations over a fixed point set, see [DLRS10]). Now orient each edge based on the Delaunay property. Bobenko and Springborn [BS07] showed that this oriented flip graph is strongly connected and has a unique sink. In our context we need to restrict the flips to maintain planarity and 3-connectedness. This removes edges from the flip graph, creating at least one connected component of planar triangulations, because the initial triangulation is planar. We know that this component may be disconnected from the global minimum of harmonic energy, as the intrinsic Delaunay

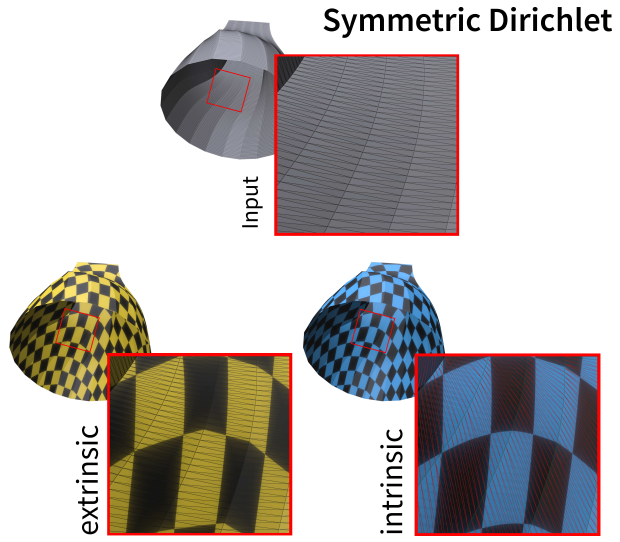

Figure 10: *Mesh #222690 from Thingi10K [ZJ16] parameterized with symmetric Dirichlet energy. Energy decreasing flipping improved the mapping (note the difference in the isolines), but the resulting intrinsic triangulation is not Delaunay.*

triangulation (the global minimum) is not necessarily planar or 3-connected. In general, there is no reason that the connected component containing the initial planar triangulations to have a unique sink. The harmonic energy of different sinks may depend on function values, in our case the parameter locations of the vertices. This means the optimal triangulation may also depend on the realization in the plane. As we show in the following experiments, generating a planar intrinsic Delaunay triangulation by greedy Delaunay flipping is nonetheless a better starting point than the extrinsic triangulation (in most cases).

Intrinsic Delaunay flips are generally not decreasing symmetric Dirichlet or ARAP energy. One might speculate that they it nonetheless useful to use $\mathcal{S}^\circ$ as initialization. Note that both energies contain Dirichlet energy as one of the terms (Eqs. 5, 7). For symmetric Dirichlet, the second term is minimized exactly if an edge is Delaunay also in the parameterization domain. Assuming an energy minimized state is almost conformal, one may argue that the angles should be similar in both realizations, and it is likely that an edge that is intrinsically Delaunay in the input is also extrinsically Delaunay in the parameterization.

Table 4: *Comparison of energy starting from planar intrinsic Delaunay triangulations rather than the extrinsic triangulation, for optimization of the parameter locations and no further flipping as well as alternating optimization until convergence. Compare also to Table 1. The average values are not included due to very large symmetric Dirichlet energies caused by Tutte embedding with planar intrinsic Delaunay triangulations.*

	No further flipping			Alternating		
	med	5%	95%	med	5%	95%
Sym. Dir.	1.000	0.999	2.705	1.000	0.998	2.446
ARAP	0.995	0.873	1.027	0.970	0.441	0.998

We have performed experiments, similar to the previous section, optimizing symmetric Dirichlet and ARAP energy starting from $\mathcal{S}^\circ$. We compare the results of the initial parameterization with the alternating optimization approach that searches for further flips after each parameterization step. In Table 4 we show the results after computing the parameterization. We have excluded the mean as it was affected by several outliers in the energy optimization. Interestingly, the statistics indicate that it is not preferable to start from the intrinsic Delaunay triangulations for these energies as starting from the given triangulation yields lower energies (c.f. Table 1). Indeed, in most cases the optimized triangulation contains many non-Delaunay edges – this is illustrated in Fig. 10.

7. Discussion

The idea of optimizing mesh parameterization by intrinsic flipping works. The difference are often clearly visible in the resulting textures, while showing only marginally in the energies.

For visualization, we need to map from points on the surface to the parameter domain for lookup of the texture values. In the standard graphics pipeline this is done by interpolation of texture coordinates stored in vertices. If one wants to exploit this technique, it is necessary to triangulate the intersection of the extrinsic and intrinsic mesh, and compute texture coordinates for the vertices resulting from edge-edge intersections by barycentric interpolation along the intrinsic edges (which are the edges in the parameterization). Depending on the application, the additional computation for processing the resulting additional vertices may not be worth the improved quality in texture mapping. An example with a significant amount of added vertices is shown in Fig. 11.

Another perspective on our method is that the piecewise linear finite element basis we use to approximate the optimal mapping has different supports. It would be interesting to see how increasing the degree of the basis functions would affect the mapping. Our approach offers a natural recursive refinement on the number of elements as well: optimize using intrinsic flipping, insert vertices at the intersection of intrinsic and extrinsic edges, triangulate, and then repeat the process with the refined mesh. Notice that choice of diagonals when triangulating non-triangular faces is irrelevant, because we allow intrinsic flipping in the next iteration of this process. We also want to note that the consideration of intrinsic flipping is not restricted to toplogical disks but could be applied to more general settings (such as spheres [AL15] or orbifolds [GGS03]), as well.

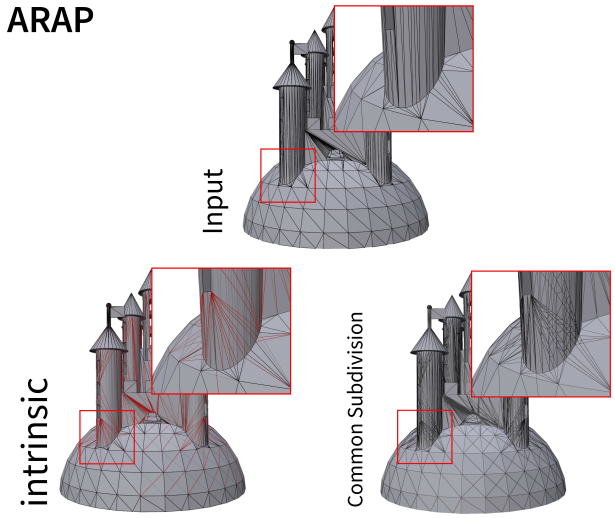

Figure 11: *Mesh #47089 from Thingi10K [ZJ16] parameterized with ARAP energy. Representing the intrinsic parametrization as an ordinary piecewise linear mapping from the plane requires the common subdivisions of the input mesh and the intrinsic triangulation. Intrinsic edges are colored in red. For this mesh the number of vertices quadrupled, from 2919 to 11604.*

Acknowledgements

This work was funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant agreement No. 101055448, ERC Advanced Grand EMERGE).

References

[AL13] AIGERMAN N., LIPMAN Y.: Injective and bounded distortion mappings in 3d. *ACM Trans. Graph. 32*, 4 (jul 2013). doi:10.1145/2461912.2461931. 1

[AL15] AIGERMAN N., LIPMAN Y.: Orbifold Tutte embeddings. *ACM Trans. Graph. 34*, 6 (nov 2015). doi:10.1145/2816795.2818099. 10

[Ale19] ALEXA M.: Harmonic triangulations. *ACM Trans. Graph. 38*, 4 (jul 2019). doi:10.1145/3306346.3322986. 4

[BN21] BROWN G. E., NARAIN R.: Wrapd: weighted rotation-aware admm for parameterization and deformation. *ACM Trans. Graph. 40*, 4 (jul 2021). doi:10.1145/3450626.3459942. 4

[BS94] BABUVSKA I., SURI M.: The p and h-p versions of the finite element method, basic principles and properties. *SIAM Review 36*, 4 (1994), 578–632. doi:10.1137/1036141. 4

[BS07] BOBENKO A. I., SPRINGBORN B. A.: A discrete Laplace–Beltrami operator for simplicial surfaces. *Discrete & Computational Geometry 38*, 4 (2007), 740–756. doi:10.1007/s00454-007-9006-1. 2, 4, 5, 8, 9

[BZK09] BOMMES D., ZIMMER H., KOBBELT L.: Mixed-integer quadrangulation. *ACM Trans. Graph. 28*, 3 (jul 2009). doi:10.1145/1531326.1531383. 6

[CBSS17] CLAICI S., BESSMELTSEV M., SCHAEFER S., SOLOMON J.: Isometry-aware preconditioning for mesh parameterization. *Computer Graphics Forum 36*, 5 (2017), 37–47. doi:10.1111/cgf.13243. 4

[CDHR08] CHEN Y., DAVIS T. A., HAGER W. W., RAJAMANICKAM S.: Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. *ACM Trans. Math. Softw. 35*, 3 (oct 2008). doi:10.1145/1391989.1391995. 7

[CWW17] CRANE K., WEISCHEDEL C., WARDETZKY M.: The heat method for distance computation. *Commun. ACM 60*, 11 (Oct. 2017), 90–99. doi:10.1145/3131280. 4

[DAZ*20] DU X., AIGERMAN N., ZHOU Q., KOVALSKY S. Z., YAN Y., KAUFMAN D. M., JU T.: Lifting simplices to find injectivity. *ACM Trans. Graph. 39*, 4 (aug 2020). doi:10.1145/3386569.3392484. 4

[dGDMD16] DE GOES F., DESBRUN M., MEYER M., DEROSE T.: Subdivision exterior calculus for geometry processing. *ACM Trans. Graph. 35*, 4 (jul 2016). URL: https://doi.org/10.1145/2897824.2925880, doi:10.1145/2897824.2925880. 4

[DGR93] DYN N., GOREN I., RIPPA S.: Transforming triangulations in polygonal domains. *Computer Aided Geometric Design 10*, 6 (1993), 531–536. doi:10.1016/0167-8396(93)90029-3. 2

[DKT98] DEROSE T., KASS M., TRUONG T.: Subdivision surfaces in character animation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1998), SIGGRAPH '98, Association for Computing Machinery, p. 85–94. doi:10.1145/280814.280826. 4

[DLRS10] DE LOERA J. A., RAMBAU J., SANTOS F.: *Triangulations: Structures for Algorithms and Applications*, 1st ed. Springer Publishing Company, Incorporated, 2010. 9

[DMA02] DESBRUN M., MEYER M., ALLIEZ P.: Intrinsic parameterizations of surface meshes. *Computer graphics forum 21*, 3 (2002), 209–218. doi:10.1111/1467-8659.00580. 3, 4

[EDD*95] ECK M., DEROSE T., DUCHAMP T., HOPPE H., LOUNSBERY M., STUETZLE W.: Multiresolution analysis of arbitrary meshes. In *Proc. SIGGRAPH* (New York, NY, USA, 1995), Association for Computing Machinery, p. 173–182. doi:10.1145/218380.218440. 2

[FBRCA23] FINNENDAHL U., BOGIOKAS D., ROBLES CERVANTES P., ALEXA M.: Efficient embeddings in exact arithmetic. *ACM Trans. Graph. 42*, 4 (jul 2023). doi:10.1145/3592445. 4

[FH05] FLOATER M. S., HORMANN K.: Surface parameterization: a tutorial and survey. In *Advances in Multiresolution for Geometric Modelling* (Berlin, Heidelberg, 2005), Dodgson N. A., Floater M. S., Sabin M. A., (Eds.), Springer Berlin Heidelberg, pp. 157–186. 4

[Flo03] FLOATER M. S.: One-to-one piecewise linear mappings over triangulations. *Math. Comput. 72*, 242 (apr 2003), 685–696. doi:10.1090/S0025-5718-02-01466-7. 4

[FSA23] FINNENDAHL U., SCHWARTZ M., ALEXA M.: ARAP revisited: Discretizing the elastic energy using intrinsic Voronoi cells. *Computer Graphics Forum 42*, 6 (2023), e14790. doi:10.1111/cgf.14790. 4

[FSSB07] FISHER M., SPRINGBORN B., SCHRÖDER P., BOBENKO A. I.: An algorithm for the construction of intrinsic Delaunay triangulations with applications to digital geometry processing. *Computing 81*, 2 (2007), 199–213. doi:10.1007/s00607-007-0249-8. 2, 3, 4, 5

[GGS03] GOTSMAN C., GU X., SHEFFER A.: Fundamentals of spherical parameterization for 3d meshes. In *ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), SIGGRAPH '03, Association for Computing Machinery, p. 358–363. doi:10.1145/1201775.882276. 10

[GKK*21] GARANZHA V., KAPORIN I., KUDRYAVTSEVA L., PROTAIS F., RAY N., SOKOLOV D.: Foldover-free maps in 50 lines of code. *ACM Trans. Graph. 40*, 4 (jul 2021). doi:10.1145/3450626.3459847. 4

[GSC21] GILLESPIE M., SHARP N., CRANE K.: Integer coordinates for intrinsic geometry processing. *ACM Trans. Graph. 40*, 6 (dec 2021). doi:10.1145/3478513.3480522. 2, 4, 5

[HLS07] HORMANN K., LÉVY B., SHEFFER A.: Mesh parameterization: theory and practice. In *ACM SIGGRAPH 2007 Courses* (New York, NY, USA, 2007), SIGGRAPH '07, Association for Computing Machinery. doi:10.1145/1281500.1281510. 4

[HSH10] HE L., SCHAEFER S., HORMANN K.: Parameterizing subdivision surfaces. In *ACM SIGGRAPH 2010 Papers* (New York, NY, USA, 2010), SIGGRAPH '10, Association for Computing Machinery. doi:10.1145/1833349.1778857. 4

[JP*18] JACOBSON A., PANOZZO D., ET AL.: libigl: A simple C++ geometry processing library, 2018. https://libigl.github.io/. 6

[Law72] LAWSON C. L.: Transforming triangulations. *Discrete mathematics 3*, 4 (1972), 365–372. 2

[LPRM02] LÉVY B., PETITJEAN S., RAY N., MAILLOT J.: Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph. 21*, 3 (jul 2002), 362–371. doi:10.1145/566654.566590. 3, 4

[LYNF18] LIU L., YE C., NI R., FU X.-M.: Progressive parameterizations. *ACM Trans. Graph. 37*, 4 (jul 2018). doi:10.1145/3197517.3201331. 4

[LZX*08] LIU L., ZHANG L., XU Y., GOTSMAN C., GORTLER S. J.: A local/global approach to mesh parameterization. In *Proc. Symposium on Geometry Processing* (2008), SGP '08, Eurographics Association, p. 1495–1504. 1, 3, 4

[MC20] MANDAD M., CAMPEN M.: Efficient piecewise higher-order parametrization of discrete surfaces with local and global injectivity. *Computer-Aided Design 127* (2020), 102862. doi:10.1016/j.cad.2020.102862. 4

[MTAD08] MULLEN P., TONG Y., ALLIEZ P., DESBRUN M.: Spectral conformal parameterization. In *Proceedings of the Symposium on Geometry Processing* (Goslar, DEU, 2008), SGP '08, Eurographics Association, p. 1487–1494. 3

[OB08] OSHEROVICH E., BRUCKSTEIN A.: All triangulations are reachable via sequences of edge-flips: an elementary proof. *Computer Aided Geometric Design 25*, 3 (2008), 157–161. doi:10.1016/j.cagd.2007.07.002. 2

[PP93] PINKALL U., POLTHIER K.: Computing discrete minimal surfaces and their conjugates. *Experim. Math. 2* (1993), 15–36. 2

[Rip90] RIPPA S.: Minimal roughness property of the Delaunay triangulation. *Comput. Aided Geom. Des. 7*, 6 (Oct. 1990), 489–497. doi:10.1016/0167-8396(90)90011-F. 4, 9

[RPPSH17] RABINOVICH M., PORANNE R., PANOZZO D., SORKINE-HORNUNG O.: Scalable locally injective mappings. *ACM Trans. Graph. 36*, 4 (jul 2017). doi:10.1145/3072959.2983621. 1, 2, 4

[SA07] SORKINE O., ALEXA M.: As-rigid-as-possible surface modeling. In *Proc. Symposium on Geometry Processing* (2007), SGP '07, Eurographics Association, p. 109–116. 3, 4

[SC*19] SHARP N., CRANE K., ET AL.: Geometrycentral: A modern c++ library of data structures and algorithms for geometry processing. 6

[SC20] SHARP N., CRANE K.: You can find geodesic paths in triangle meshes by just flipping edges. *ACM Trans. Graph. 39*, 6 (nov 2020). doi:10.1145/3414685.3417839. 2

[SGC21] SHARP N., GILLESPIE M., CRANE K.: Geometry processing with intrinsic triangulations. In *ACM SIGGRAPH 2021 Courses* (New York, NY, USA, 2021), SIGGRAPH '21, Association for Computing Machinery. doi:10.1145/3450508.3464592. 2, 4, 5, 6, 8

[SHD*18] SCHNEIDER T., HU Y., DUMAS J., GAO X., PANOZZO D., ZORIN D.: Decoupling simulation accuracy from mesh quality. *ACM Trans. Graph. 37*, 6 (dec 2018). doi:10.1145/3272127.3275067. 4

[SPR06] SHEFFER A., PRAUN E., ROSE K.: Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis. 2*, 2 (jan 2006), 105–171. doi:10.1561/0600000011. 4

[SS15] SMITH J., SCHAEFER S.: Bijective parameterization with free boundaries. *ACM Trans. Graph. 34*, 4 (jul 2015). doi:10.1145/2766947. 1, 3, 4, 9

[SSC19] SHARP N., SOLIMAN Y., CRANE K.: Navigating intrinsic triangulations. *ACM Trans. Graph. 38*, 4 (jul 2019). doi:10.1145/3306346.3322979. 2, 4, 5

[Tut60] TUTTE W. T.: Convex representations of graphs. *Proceedings of the London Mathematical Society s3-10*, 1 (1960), 304–320. doi:10.1112/plms/s3-10.1.304. 3, 4

[WZ14] WEBER O., ZORIN D.: Locally injective parametrization with arbitrary fixed boundaries. *ACM Trans. Graph. 33*, 4 (jul 2014). doi:10.1145/2601097.2601227. 4

[ZJ16] ZHOU Q., JACOBSON A.: Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797* (2016). 6, 7, 8, 9, 10