Direct Seamless Parametrization—Supplement

ZOHAR LEVI, Victoria University of Wellington, New Zealand

This supplement discusses the following topics:

- Section 1 provides more details on the seamless parametrization problem that was addressed in the paper.
- Section 2 suggests what we look for in a quad mesh, how it relates to seamless parametrization, and how the two can be compared.
- Section 3 surveys the distortion measures that were used in the experiments.
- Section 4 reviews the connection between the cross field and the local frames used in the lay out stage for the applicable distortion measure. Moreover, more details are given on the main layout optimization method that was used in the paper.
- Section 5 discusses a conformal bounded distortion measure and suggests a new approach to optimizing it.
- Section 6 explains how a cross field is extracted from a given seamless parametrization.
- Section 7 explains how to extract parametrization from a quad mesh.
- Section 8 provides a way to make seamless parametrization integer.
- Section 9 suggests a recovery routine for bad fields, which can be also be used for parametrization with foldovers.
- Section 10 experiments with solving the problem with an off-the-shelf MI solver.
- Section 11 suggests how to auto-scale models to reduce numerical issues.
- Section 12 clarifies the definition of discrete seamlessness.

1 DEFINING THE SEAMLESS PARAMETRIZATION PROBLEM

Our objective is low distortion seamless parametrization. There are two aspects to consider.

Number of cones. Having fewer singular vertices is desired. For example, fields and quad meshes appear more uniform this way since fewer singular vertices usually means larger regular patches. However, similar to previous work, we choose not to limit the number of cones. Note that due to seamlessness constraints, cone angles are rounded (to $\frac{\pi}{2}$-multiple). Unless refinement is allowed, adding more cones does not necessarily reduce the distortion.

Refinement. If refinement of the triangle mesh is allowed, Myles and Zorin [2013] provide a constructive proof that the distortion can be lowered below an arbitrary threshold. The idea is the following. If the cone angles are not limited to multiples of $\frac{\pi}{2}$ (effectively omitting the seamlessness constraints), then we can create an isometric parametrization without distortion, where the angles between twin seam edges are arbitrary. This is done by treating each vertex as a cone, cutting the surface along a (dual) spanning tree of the facets, and laying out the mesh in the plane triangle by triangle. Restricting ourselves to $\frac{\pi}{2}$-multiple cones, we can approximate this parametrization and "draw" (lay out) the seam (polygon) in the plane in the spirit of Bresenham's line algorithm, using $\frac{1}{4}$ and $-\frac{1}{4}$ index cone for $\frac{\pi}{2}$ turns. The (average) error of this approximation can be made arbitrarily low by refining the triangulation.

Excessive refinement, though, is undesirable since it increases the runtime of subsequent algorithms and working with low resolution meshes is easier in general. Moreover, since we do not limit the number of cones, when the mesh resolution becomes too fine, cones can cluster close together, causing problems. For example, they can force a generated quad mesh to have a very fine resolution (which depends directly on the minimal distance between cones). Therefore, in our problem definition, we forbid mesh refinement.

Following this discussion, we define the setting of our problem. Given a triangle mesh, we generate a seamless parametrization that targets low isometric distortion. The source mesh is kept intact, and the number of cones is not limited.

2 QUAD MESH QUALITY

Since quad meshing is a key application of seamless parametrization, we address the question of how to measure the quality of a quad mesh Q generated from a given triangle mesh M. We consider the following attributes.

Number of singular vertices. We mentioned that it is desirable to have fewer singular vertices (large regular patches). From the visual aspect consideration, sometimes their placement is more important than their quantity. We also note that in general, a small number of singular vertices does not necessarily imply a simple base complex [Tarini et al. 2011]. As discussed, the paper does not target this property.

Approximating the source mesh. One can measure how close Q is to M, e.g. in terms of Hausdorff distance. If minimizing the distance between the meshes is the primary objective, then to generate Q, one can simply perform a single step of Catmull-Clark-like subdivision on M. That is, performing the combinatorial changes of the subdivision but setting the geometry as follows: the original vertices of M are kept, and the new vertices in Q are interpolated inside the triangles of M. This process generates a quad mesh with zero Hausdorff distance. Usually, though, one is interested in well

Author's address: Zohar Levi, Victoria University of Wellington, New Zealand.

0730-0301/2020/1-ART1 $15.00
https://doi.org/10.1145/3439828

shaped quads while keeping a reasonable distance from the original mesh.

We require that the quad mesh converges to the surface defined by M under refinement, i.e., when finer and finer resolutions are specified for the quad mesh generation algorithm. This is one of the properties of parametrization-based quad meshing.

Quad shape. Well shaped quads can be defined as being nearly square and having relatively uniform size across the mesh.

Before proposing a measure for quad shape quality, as illustration, we consider first the simpler problem of how to measure the quality of a triangle mesh. One desired attribute is regularity. Second, uniform size equilateral triangles are preferred. To measure how close a facet is to an equilateral triangle, one can measure angles, ratios between edge lengths, and scaling and decide how to weigh them. A different approach is to measure the distortion of the linear map from a facet to an equilateral triangle. We suggest a similar measure for quad meshes.

Given a quad mesh, we lay it out in the UV domain to create an integer seamless parametrization as described in Section 7. The resulting map is locally injective. In fact, if Q was generated from an integer seamless parametrization of M, the layout of Q in the plane would recover the parametrization (keeping in mind that a seamless parametrization is invariant to the seam location). We suggest to measure the quality of the quad mesh as the distortion of this mapping. This enables us to use parametrization distortion measures on quad meshes.

We note that this measurement is appropriate only in the right context, where the quad mesh was generated by an algorithm that targeted a similar objective. The literature offers other objectives that we did not address such as flatness or how close angles are to 90° [Bommes et al. 2013].

3 PARAMETRIZATION DISTORTION MEASURES

To preserve angles and distances in the mapping as well as possible, we aim to minimize a foldovers-free isometric distortion. Most distortion measures can be optimized using our framework, and we survey a few popular ones that were used in our experiments.

Most distortion measures can be expressed in terms of the singular values of the mapping. Given a triangle $t \in T$ and its mapping f, let σ_1, σ_2 ($\sigma_1 \geq \sigma_2$) be the *signed* singular values (the smallest singular value becomes negative if the triangle is flipped [Aigerman and Lipman 2013]) of df. We will mention a few popular isometric distortion measures.

Aigerman et al. [2014] used the distortion measure

$$\tilde{\tau}(f) = \begin{cases} max(\sigma_1, \frac{1}{\sigma_2}) & \sigma_2 > 0 \\ \infty & else \end{cases} , \tag{1}$$

where $\tilde{\tau} = \infty$ if the triangle is flipped or collapsed. Apart from penalizing foldovers, this metric is not new and was considered, for example, in [Sorkine et al. 2002].

One technicality to note about this measure is that scaling the mapping could improve the measure. More specifically, if $\sigma_1 \neq \frac{1}{\sigma_2}$, then we can uniformly scale f to lower $\tilde{\tau}$. Scaling up the mapping would increase σ_1 and decrease $\frac{1}{\sigma_2}$, and scaling down would decrease σ_1 and increase $\frac{1}{\sigma_2}$. This tradeoff is settled by the optimal scaling factor:

$$s^* = \arg\min_{s \in \mathbb{R}} \tilde{\tau}(sf) = \frac{1}{\sqrt{\sigma_1 \sigma_2}} , \tag{2}$$

which ensures $\sigma_1 = \frac{1}{\sigma_2}$. When a whole mesh is concerned, we consider a single global optimal scaling factor for the whole mapping. Based on that, we define τ, a new variation of the distortion measure, which is invariant to uniform scaling (or a global similarity transform). It takes the form of the geometric mean of σ_1^{max} and $\frac{1}{\sigma_2^{min}}$:

$$\tau = \begin{cases} \sqrt{\frac{\sigma_1^{max}}{\sigma_2^{min}}} & \sigma_2^{min} > 0 \\ \infty & else \end{cases} , \tag{3}$$

where σ_1^{max} and σ_2^{min} are the maximal and minimal singular values over all mesh triangles. For a single triangle in the mesh, we define:

$$\tau_t = \tilde{\tau}(s_M^* f_t) ,$$

where $s_M^* = \frac{1}{\sqrt{\sigma_1^{max} \sigma_2^{min}}}$ is the optimal global scaling for the whole mesh mapping. Note that $\tau(f) \leq \tilde{\tau}(f)$, where equality holds if $\sigma_1^{max} = \frac{1}{\sigma_2^{min}}$, which happens when the map is optimally scaled. When optimizing the measure, there is no need to optimize τ directly, and it is equivalent to optimize the simpler distortion measure $\tilde{\tau}$ instead (Section 4.2).

Another popular distortion measure is based on the so-called *As-Rigid-As-Possible* parametrization energy [Liu et al. 2008], which is defined for a triangle as

$$\min_{R \in SO(2)} \|\mathrm{D}f - R\|_F^2 = (\sigma_1 - 1)^2 + (\sigma_2 - 1)^2 . \tag{4}$$

For a whole mesh, usually the L_2-norm variant is used, where the measure is integrated over the mesh. Levi and Zorin [2014] offered an L_∞-norm variant; more details in Section 4.2.

Recently, the symmetric Dirichlet energy [Rabinovich et al. 2017; Shtengel et al. 2017; Smith and Schaefer 2015] has gained popularity. It is defined per triangle as

$$\sigma_1^2 + \frac{1}{\sigma_1^2} + \sigma_2^2 + \frac{1}{\sigma_2^2} \tag{5}$$

and integrated over the whole mesh.

4 USING THE CROSS FIELD AS LOCAL FRAMES

In this section, we are going to recall and summarize the relation between the cross field and the local frames that are used in the layout optimization. In the context of 2D or 3D (surface or volumetric) mesh deformation, in order to get naturally plausible results, the objective is to generate a mapping close to isometric (subject to given constraints). Surface mesh parametrization can be considered a 3D to 2D deformation.

Following [Pinkall and Polthier 1993], let $\mathcal{P}$ be a source triangle mesh and $\mathcal{Q}$ a deformed version of it. Consider two corresponding triangles T_p and T_q in the two meshes, as well as a canonical triangle

T_e given in the unit base $\{e_1, e_2\} \subset \mathbb{R}^2$. Define the maps $p\colon T_e \to T_p$, $q\colon T_e \to T_q$, and $f\colon T_p \to T_q$, $f := q \circ p^{-1}$. The mapping Jacobians are $\mathrm{D}\,p$, $\mathrm{D}\,q$, and $\mathrm{D}\,f = \mathrm{D}\,q \cdot \mathrm{D}\,p^{-1}$. Earlier work on (direct) mesh deformation was guided by preserving local differential quantities and encoding local details [Botsch and Sorkine 2008; Sorkine 2006]. In particular, there is the *gradient editing* energy (that aims at preserving the mapping Jacobian)

$$E(T_p, T_q) = \|\mathrm{D}\,q - \mathrm{D}\,p\|^2 \ .$$

Integrating this energy over the mesh and solving for x, the vertex coordinates of Q, results in a linear system of the Poisson equation $\Delta x = b$. The main issue with these differential quantities is that they are not rotation invariant. Therefore, there was a need to introduce a local orientation term $R \in SO(n)$ to encode the local frame:

$$E(T_p, T_q, R) = \|\mathrm{D}\,q - R \cdot \mathrm{D}\,p\|^2 \ . \tag{6}$$

Related work strives to approximate or solve for R efficiently. One approach is to solve for a smooth frame field over the mesh [Zayer et al. 2005], where the motivation is propagating the user handle orientation in a reasonable manner, as well as preserving local rigidity by trying to keep a patch of facets oriented uniformly.

Bommes et al. [2009] used a similar concept for seamless parametrization. Here, due to seamlessness constraints, the frame field becomes a cross field, and its treatment is done more rigorously. Field concepts, such as singularities, field index, and period jumps [Ray et al. 2008], are utilized, and the field smoothness energy is formulated as a connection. After a set of frames is acquired, the layout step becomes solving the same Poisson equation that results from (6).

One paradigm to note is solving for optimal local frames. This motivates the *As-Rigid-As-Possible* energy:

$$E(T_p, T_q) = \min_{R \in SO(d)} \|\mathrm{D}\,q - R \cdot \mathrm{D}\,p\|^2 \ . \tag{7}$$

Another way to view this energy is through (4), which is identical up to weighting. In that view, we measure how far the mapping Jacobian is from a rotation (from being a rigid transformation).

4.1 The Local-Global Method

An efficient approach, the so-called *local-global method*, to minimize (4) was offered in [Sorkine and Alexa 2007]. The idea is to alternate between a local and a global step minimizing the energy in a block-coordinate-descent manner until convergence. Given the coordinates x, the local step solves efficiently for the optimal local frame R using the polar decomposition $\mathrm{D}\,f = RY$. Given a set of local frames, the global step solves the Poisson equation that results from (6). The initial frames that are used are normally the identity. Alternatively, if a plausible deformed mesh is given as initialization (e.g. [Sorkine 2006]), then the initial frames can be extracted using the local step.

The idea was applied successfully to mesh deformation and general mesh parametrization [Levi and Gotsman 2015; Liu et al. 2008; Sorkine and Alexa 2007]. Myles and Zorin [2012] applied the same idea to seamless parametrization. Similar to [Bommes et al. 2009], they first solve for the cross field, but in the layout step, instead of minimizing (6) in a single iteration, they use the frame field as initialization to the local-global method to minimize (7).

Chao et al. [2010] use a second-order Newton method to optimize (4), which requires less iterations (but they are more costly). While this method works for simple energies, it is less practical in the presence of non-linear constraints that arise, for example, when using L_∞-norm-based energy (Section 4.2).

4.2 Bounded Distortion and Local Frames

Instead of integrating (4) over the mesh (using $L2$-norm), getting a Dirichlet-like energy that targets the average distortion (which results in the linear Poisson equation in the global step), [Levi and Zorin 2014] observed that it can be used in formulating a Second Order Cone Programming (SOCP) to minimize (or bound) the maximal point-wise distortion over the mesh (using L_∞-norm):

$$\min_{x \in \mathbb{R}^{2\times n}} \quad \max_{t \in T} \ \min_{R_t \in SO(2)} \ \mu \tag{8a}$$

$$\text{s.t.} \qquad \|\mathrm{D}\,f_t - R_t\|_F^2 \le \mu \ , \tag{8b}$$

where $\mu \in \mathbb{R}$ is a global distortion bound (an auxiliary variable to bound τ, which is used in the standard approach to minimize an L_∞ objective), R_t is the optimal frame of triangle t, and df_t is the Jacobian of the mapping of triangle t.

One thing to note is that if μ turns out to be strictly less than one, then the mapping would be locally injective. This follows directly from (4):

$$(\sigma_1 - 1)^2 + (\sigma_2 - 1)^2 < 1 \ ,$$

which leads to

$$1 - \sigma_2 < 1 \ .$$

Similar to the L_2 version, the local-global method is used to minimize this L_∞-norm-based energy. The local step remains the same. In the global step, instead of solving a linear system, we solve the SOCP above, which can be done in a reasonable time using off-the-shelf solvers [ApS 2015]. We will refer to energies of this type as L_p-ARAP.

A global bound on the distortion measure in (1) can also be minimized efficiently, effectively minimizing our main distortion measure in (3). We provide a summary of the details, following [Aigerman et al. 2014; Lipman 2012].

$A := \mathrm{D}\,f$ can be decomposed into $A = B + C$, where $B := \frac{1}{2}(A - A^T + tr(A)I)$ is a similarity matrix, and $C := \frac{1}{2}(A + A^T - tr(A)I)$ is an anti-similarity matrix. In this notation, the singular values of A can be expressed as:

$$\sigma_1 = \frac{1}{\sqrt{2}}\Big(\|B\|_F + \|C\|_F\Big) \ , \ \sigma_2 = \frac{1}{\sqrt{2}}\Big|\|B\|_F - \|C\|_F\Big| \ .$$

Then, given a set of (orthogonal) frames $\{R_t\}_{t \in T}$, the convexified SOCP version of the layout problem (without seamlessness constraints for clarity) becomes:

$$\min_{x \in \mathbb{R}^{2\times n}} \quad \mu_1 \tag{9a}$$

$$\text{s.t.} \qquad \sigma_1^t \le \mu_1, \ \forall t \in T, \tag{9b}$$

$$0 \le \mu_2 \le \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}} tr(R_t^T B_t) - \|C_t\|_F\right), \ \forall t \in T, \tag{9c}$$

$$1 \le \mu_1\,\mu_2 \ , \tag{9d}$$

where σ_1^t, σ_2^t are the singular values of df_t. $\mu_1, \mu_2 \in \mathbb{R}$ are two global (upper and lower) bounds on all the singular values. (9d) is a rotated quadratic cone (or a restricted hyperbolic constraint), which can be written as a second-order cone (SOC) constraint:

$$\left\| \begin{bmatrix} 2 \\ \mu_1 - \mu_2 \end{bmatrix} \right\| \leq \mu_1 + \mu_2 \; .$$

(9c) is the convexified version of $\mu_2 \leq \sigma_2^t$ [Lipman 2012]. The shape of its solution space is a maximal carved chunk from the original constraint space, which is a cone shape that can be parametrized by a range of rotation angles. (In other words, in order to make the constraint convex, we ignore some of the possible solutions. A solution can be expressed using a rotation angle, and limiting the range of rotation angles is tantamount to ignoring the corresponding solutions.) The range of rotation angles in the solution space depends on R_t, and in fact, the range is centered around it.

Following that, if we would like the (convexified) constraint feasible region to be centered around f_t, then we extract R_t from $\mathrm{D}\, f_t$ using the local step (polar decomposition). This allows us to utilize the local-global method as before. The local step remains the same, and in the global step we solve the SOCP in (9) [Lipman 2012]. Alg. 1 summarizes the layout algorithm. In our experiments, this proved to be the fastest method to optimize the distortion measure τ in (3).

Algorithm 1: The layout algorithm for bounded distortion

```
Input  : A set of frames $\{R_t\}_{t \in T}$.
Output : $UV$ coordinates $x$.
while not converged do
    Global step: solve the SOCP in (9).
    Local step: solve for a new set of frames $\{R_t\}$ using polar decomposition.
```

4.3 Extracting Frames from a Mapping

Given a mapping f, a surface parametrization in our case, we can use the local step to extract a set of frames from all the faces. As mentioned previously, these frames can be used to initialize the local-global method. Furthermore, we can extract a fully-prescribed cross field—one that defines matchings and singularities—from any such mapping; see Section 6. Therefore, even if a surface parametrization method does not explicitly use cross field concepts, a cross field can be extracted from the result. For other uses of a cross field, see [Vaxman et al. 2016].

A cross field can also be extracted from a quad mesh. First, we extract a mapping f (see Section 7), then we extract a field from the mapping.

5 QUASICONFORMAL MAPPING

Quasiconformal mappings are important in geometry processing [Lipman 2012; Sawhney and Crane 2017; Weber et al. 2012]. We will discuss two bounded conformal distortion measures.

Similar to ARAP, a quasiconformal version was suggested, the As-Similar-As-Possible (ASAP) energy [Liu et al. 2008], which is defined for a triangle as

$$\min_{\substack{R \in SO(2) \\ s \in \mathbb{R}^+}} \|\mathrm{D}\, f - sR\|_F^2 = (\sigma_1 - s)^2 + (\sigma_2 - s)^2 \; . \tag{10}$$

Here, we would like the mapping Jacobian to be close to a similarity matrix that is represented by a rotation matrix times a uniform scale. ASAP can be minimized in the local-global framework similar to ARAP, with a modification to the local step that solves for the best uniform scale

$$s = \frac{\sigma_1 + \sigma_2}{2} \; .$$

A bounded distortion version ASAP-L_∞ was considered in [Levi and Zorin 2014]. Over the mesh, it is defined similar to (8). To weigh the terms properly (w.r.t. the maximal distortion bound) and maintain the condition that $\mu < 1$ guarantees local injectivity, we use the following version of the energy:

$$\min_{\substack{R \in SO(2) \\ s \in \mathbb{R}^+}} \left\| \frac{1}{s}\, \mathrm{D}\, f - R \right\|_F^2 = (\frac{\sigma_1}{s} - 1)^2 + (\frac{\sigma_1}{s} - 1)^2 \; . \tag{11}$$

When using an arbitrary frame instead of a cross, e.g. for anisotropy, a similar consideration is applied:

$$\min_{R \in SO(2)} \|A^{-1}\, \mathrm{D}\, f - R\|_F^2 \; ,$$

where A is a linear transformation that is applied to a cross and represents a frame [Panozzo et al. 2014].

Another bounded conformal distortion measure was offered in [Lipman 2012]. It is based on the same convexification approach to τ in (3), and it is defined as the large dilatation

$$K = \frac{\sigma_1}{\sigma_2} \; .$$

Unlike τ, though, which can be used as a minimization objective in a SOCP with the suggested convexification, K can only be bounded by a given constant. Then, to optimize K, Lipman [2012] resorts to using the bisection method. This needs to be combined with the local-global approach that optimizes the frames, e.g. as an additional outer loop.

Instead of that, we offer a new method in the spirit of the ASAP optimization method. We use the same local-global method that we used to minimize τ in Section 4.2, but we modify the local step to solve for the optimal scale as well. The optimal scale for a triangle is given in (2). The problem becomes:

$$\begin{aligned} \min_{\substack{x \in \mathbb{R}^{2 \times n} \\ s \in \mathbb{R}^{|T|}}} \quad & \mu_1 & \text{(12a)} \\ \text{s.t.} \quad & s_t \sigma_1^t \leq \mu_1, \; \forall t \in T, & \text{(12b)} \\ & 0 \leq \mu_2 \leq s_t \sigma_2^t, \; \forall t \in T, & \text{(12c)} \\ & 1 \leq \mu_1 \, \mu_2 \; , & \text{(12d)} \end{aligned}$$

where we introduced a uniform scale s_t per triangle. The constraints are convexified as in (9).

6 FIELD EXTRACTION

A cross field has various applications [Vaxman et al. 2016], and it can be extracted from a seamless parametrization as described in the following steps:

- An orthogonal frame for a triangle is extracted from the polar decomposition of its mapping Jacobian.
- Cone angles are extracted by computing every vertex angle in the parametric domain and converted to field indices $I_V \in \mathbb{R}^{|V|}$.
- Homology (generator) loop indices $I_L \in \mathbb{R}^{|L|}$ are extracted from the parametrization.
- Matchings (modulo 4) are extracted from the transition functions between twin seam edges in the layout and set to 0 for all the rest. These matchings are determined up to integer multiples of 4, which need to be found. From these matchings, we calculate the field index set $\bar{I} \in \mathbb{R}^{|V|+|L|}$ of the homology loops L and vertices over the surface [Bommes et al. 2009]. Given their desired field indices $I = I_V \frown I_L \in \mathbb{R}^{|V|+|L|}$ (where $\frown$ stands for sequence concatenation) that were extracted from the parametrization, we solve for a set of integer field indices $k \in \mathbb{Z}^{|E|}$ to be added to the matchings on all the edges (as $4k$). This results in the following MILP (Mixed Integer Linear Programming) problem:

$$\min_{k \in \mathbb{Z}^{|E|}} \quad 1 \tag{13a}$$

$$\text{s.t.} \quad I_v - \bar{I}_v = \sum_{e \in \mathrm{adj}(v)} k_e, \quad \forall v \in V, \tag{13b}$$

$$I_\ell - \bar{I}_\ell = \sum_{e \in \ell} k_e, \quad \forall \ell \in L\,. \tag{13c}$$

(13b) relates the difference between vertex field indices and matchings, where adj(v) stands for the set of edges adjacent (incident) to the vertex. In (13c), ℓ is a loop consisting of a set of edges. The problem is solved under a second with [Gurobi 2018], and we did not encounter any failures. This is not surprising since there is an algorithm solvable in polynomial time for this problem that results in a homogeneous linear system. Our approach, though, is easier to implement, and we can expect from a MILP solver to succeed and solve it in a timely fashion.

7 QUAD MESH LAYOUT

A quad mesh can be laid out in the parametric domain to define a locally injective mapping. First, cut the mesh into a disk. Then, lay out all quad facets in the plane: starting from a seed facet and proceeding with its neighbors (in a BFS-like manner), map each quad facet to a unit square with integer vertex coordinates. In effect, we endow the quad mesh with a metric and lay it out accordingly. The prescribed metric sets each quad vertex angle to $\frac{\pi}{2}$, and each quad edge length to 1. As an alternative to the incremental layout approach, we can construct a linear system and solve for the coordinates, e.g. LSCM [Lévy et al. 2002]. In our experiments, that method, though, proved to be less accurate.

Fig. 1 shows an example of laying out [Fang et al. 2018]'s quad mesh result (and triangulating it) that generates integer seamless parametrization. We used this map for texture mapping. When no global rotation is applied to the map, the texture is similar to the quad mesh. We also illustrate how the texture looks like after a global rotation of the map. Compare this result to the teaser image in the paper, where an arbitrary global scale, translation, and rotation were applied to the map (in the figures, an arbitrary similarity transform was applied to the texture mapping). While a global similarity transform of the mapping affects a seamless parametrization such that it may not be integer anymore, a distortion measure, e.g. (3), and the seam choice should be invariant to it.

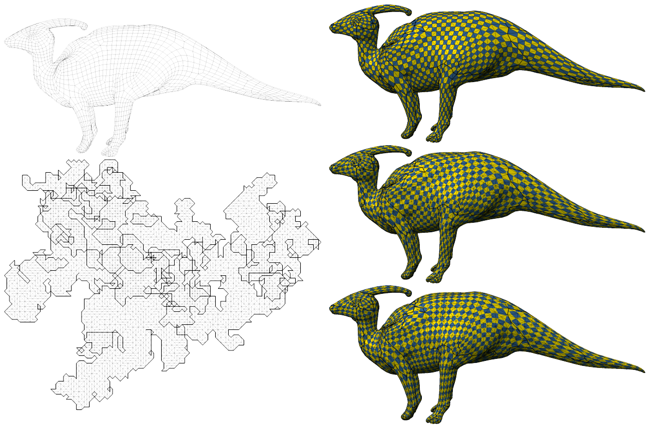

Fig. 1. Quad layout. *Top-left*: Fang18's result. *Bottom-left*: quad layout. *Right*: texture mapping, where the map is globally rotated in the domain by 0°, 22°, and 45° from *top* to *bottom*.

8 GENERATING INTEGER SEAMLESS PARAMETRIZATION

Given a seamless mapping $\bar{f}$, we describe how to generate a similar integer seamless mapping f such that twin edge translations, cone positions, and mapped feature isolines are integers with respect to a desired grid resolution. We begin by calculating the Jacobian for each triangle $\bar{J}_t = d\bar{f}_t$. Then, using the rounding method suggested in [Bommes et al. 2009], we minimize the energy:

$$\sum_{t \in T} A_t \| df_t - \bar{J}_t \|_F^2\,,$$

where A_t denotes the triangle area. The energy aims at preserving the Jacobian of the given mapping. The energy is quadratic, and [Bommes et al. 2009] solves this mixed-integer problem greedily and efficiently.

9 MATCHING CONSISTENCY

Matchings are usually rounded individually (even if rounded simultaneously), e.g. the naive approach of rounding greedily to the nearest integer. In challenging cases, the matchings result may not be consistent with the mesh topology. We offer a procedure to remedy such cases and make the matchings consistent. While matching consistency is a necessary condition to generate an injective mapping, it is not sufficient. Nevertheless, when experimenting with naive rounding (without the local estimation step), our procedure fixed all the problematic fields in our benchmark, resulting in injective mappings. This procedure can be applied to any field, including one generated by other methods.

Necessary consistency conditions:

- The field satisfies the Poincare–Hopf theorem.
- The sum of angles in a triangle is π.
- Triangle angles are positive.

Our matching consistency procedure consists of solving the following MIQP problem. Given a vector of (domain) triangle angles $\bar{\alpha} \in \mathbb{R}^{3|T|}$, vertex field indices $\bar{I} \in \mathbb{R}^{|V|}$, and matchings $\bar{r} \in \mathbb{R}^{|S|}$, we solve for new integer matchings r:

$$\min_{\substack{r\in\mathbb{Z}^{|S|}\\ \alpha\in\mathbb{R}^{3|T|}\\ I\in\mathbb{R}^{|V|}}} \quad \|\alpha - \bar{\alpha}\|^2 \tag{14a}$$

$$\text{s.t.} \quad I_v - \bar{I}_v = \sum_{e\in\mathrm{adj}(v)} \frac{r_i - \bar{r}_i}{4}, \quad \forall v \in V, \tag{14b}$$

$$I_v < 1, \quad \forall v \in V, \tag{14c}$$

$$\alpha_i \geq 0, \quad i = 1 \ldots 3|T|, \tag{14d}$$

$$\sum_{\alpha_i\in\mathrm{cor}(t)} \alpha_i = \pi, \quad \forall t \in T, \tag{14e}$$

$$\sum_{\alpha_i\in\mathrm{cor}(v)} \alpha_i = 2\pi(1 - I_v), \quad \forall v \in V, \tag{14f}$$

$$\sum_{v\in V} I_v = \chi(M)\ . \tag{14g}$$

The minimized objective keeps the solution angles close to the input. α is a vector of new domain angles, and I are new vertex field indices. (14b) relates the difference between vertex field indices and matchings, where $\mathrm{adj}(v)$ stands for the set of edges adjacent (incident) to the vertex. (14e) ensures that the sum of angles in a triangle remains π, where the notation $\mathrm{cor}(t)$ stands for the set of three triangle (corner) angles. (14f) relates field indices to angles, where the notation $\mathrm{cor}(v)$ stands for the set of (corner) angles incident to a vertex. (14g) enforces the Poincare–Hopf theorem to hold. For more background about the field-related constraints, see [Ray et al. 2008]. We solved this MIQP with [Gurobi 2018], initializing with a warm start $r = \bar{r}$.

We incorporated this procedure into the main algorithm as follows. In each iteration, if the layout step fails (the solver cannot find a solution that satisfies the constraints), we roll back to the state at the end of the previous iteration and redo the current iteration in the following way:

- Perform the usual matching rounding step and derive $\bar{\alpha}$ from the source mesh and $\bar{I}$, $\bar{r}$ from the current rounded matchings.
- Feed these as input to the matching consistency procedure to acquire new consistent rounded matchings. Note that when implementing problem (14), only the matchings that are rounded in the current iteration are set to integer variables. That is, the set of variables r is divided into 3 sets:
  i) Constant matchings that were already rounded.
  ii) Integer matchings that are going to be rounded in the current iteration.
  iii) Real (unrounded) matchings that remain free.
- Continue the main algorithm as usual.

In our experiments, when using naive rounding (without the local estimation step) on our benchmark models (without feature alignment), four models required the matching consistency step. Table 1 gives statistics of how many times it needed to be run for each model, and the average time spent on each run. For the results in the paper, this procedure was not used, and using our rounding strategy (based on local estimation) prevented these problems in the first place.

model	#tri	1 it(sec)	#it
armadillo	100K	7	1
dragon	105K	16	3
robocat	8K	1	1
greek_sculpture	50K	4	1

Table 1. Matching consistency procedure. Columns: #tri, number of triangles; 1 it, the average time in seconds to run the procedure; #it, number of times the procedure was called.

The procedure can be useful in general to recover consistent (valid) matchings (and cones) from any given field, including a parametrization with foldovers. If no parametrization is supplied, then surface angles can be used. In the case of a parametrization with foldovers, it might be better to use ℓ_1-norm in the objective in (14), which promotes sparsity—concentrating the error in sparse locations, e.g. at angles of flipped triangles. For example, the robocat model result of [Bommes et al. 2009] contains 1-field-index cones. We applied the procedure to it to correct the field, solved for a new smooth cross-field (without changing the matchings), and used it as local frames in the layout optimization. This resulted in a foldover-free mapping.

10 USING A MIXED-INTEGER SOLVER

While the procedure in Section 9 can resolve consistency issues, success is not guaranteed since the forced conditions are necessary but not sufficient for local injectivity. Moreover, while its objective is to keep the metric intact, it does not directly minimize the distortion energy. One option is to employ an off-the-shelf MISOCP solver to directly solve the main problem that optimizes UV coordinates and integer matchings with convexified bounded distortion constraints. Unfortunately, there are too many integer variables, which makes it impractical. To make the problem more manageable, we incorporate this solver into the main algorithm and optimize the integer variables in batches of size b as follows.

Similar to our approach in Section 9, if the layout step fails, we perform a rollback to the previous step and define as integer variables at most b of the new matchings to be rounded (if more are left, we do not increase the rounding error threshold for the next iteration). This problem is solved using an MISOCP solver. In our experiments, we used $b = 800$ (a large group size that still runs in a reasonable time), which took a couple of minutes on average for each iteration using [Gurobi 2018]. Since this approach is slow and generally performs worse than our rounding strategy (with local estimation), we suggest it as a last resort. While we experimented with it (instead of our rounding strategy), we did not use it for the results in the paper. Nevertheless, it was an obvious direction that was needed to be explored (or compared with) before coming up with a more specialized rounding method to the problem.

11 AUTO SCALING THE MODELS

Due to limited precision, the optimization may run into numerical issues for some of the models. Often, these can be resolved with globally scaling the models. Considering our optimized distortion measure, we suggest an optimal global scale for a triangle mesh $\mathcal{P}$. Similar to Section 4, define the map p from the canonical triangle to a mesh facet. Similar to Section 3, define σ_1^{max} and σ_2^{min} as the maximal and minimal singular values of $\mathrm{D}\,p$ over all mesh triangles. Then, the optimal scale is similar to (2):

$$\sqrt{\frac{\sigma_1^{max}}{\sigma_2^{min}}}\ .$$

In the evaluation, we scaled the heptoroid, seahorse, and vh_skin models. Otherwise, due to numerical issues, the resulting map had significantly higher distortion.

12 DISCRETE SEAMLESSNESS

In the smooth case, for seamlessness, we require the Jacobians of the mapping at a seam point to agree on both sides of the seam up to a $90r^\circ$ rotation [Myles and Zorin 2012] (in the context of a symmetric grid pattern texture). In the discrete case, we only require that vectors of twin seam half-edges agree up to a $90r^\circ$ rotation. More specifically, the Jacobians of the two triangles sharing a seam edge do not necessarily agree. As a result, in a coarse triangulation, a checkers pattern across the seam will not be smooth (e.g. with nice, parallel, straight grid lines) across the seam—like in the smooth case—even though seamlessness constraints hold (then, verification of the constraints cannot be done visually). This effect can be seen in the checkers pattern across the seam of coarse models such as the robocat.

Some may find the definition of seamless somewhat misleading, but we need to keep in mind that unless the Jacobians agree across the seam, the seam would be visible. In the discrete case, the triangle Jacobians rarely agree across the seam, and the mapping is almost never smooth as in the smooth case. Therefore, without better use in the discrete case, previous work (coined and) used the term seamless parametrization in the context of quad meshing, where the lines of a grid pattern connect across the seam (but only continuous), and a valid quad mesh can be extracted. Besides that, in the refinement limit of the source mesh, the Jacobians across the seam would coincide, and the discrete and smooth definition would agree.

REFERENCES

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

Aigerman Noam, Poranne Roi, and Lipman Yaron. 2014. Lifted bijections for low distortion surface mappings. *TOG* 33, 4 (2014), 69.

ApS MOSEK. 2015. *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28).* http://docs.mosek.com

Bommes David, Lévy Bruno, Pietroni Nico, Puppo Enrico, Silva Claudio, Tarini Marco, and Zorin Denis. 2013. Quad-Mesh Generation and Processing: A Survey. *CGF* 32, 6 (2013), 51–76.

Bommes David, Zimmer Henrik, and Kobbelt Leif. 2009. Mixed-integer Quadrangulation. *ACM Trans. Graph.* 28, 3 (2009), 77:1–77:10.

Botsch Mario and Sorkine Olga. 2008. On linear variational surface deformation methods. *TVCG* 14, 1 (2008), 213–230.

Chao Isaac, Pinkall Ulrich, Sanan Patrick, and Schröder Peter. 2010. A simple geometric model for elastic deformations. *ACM Trans. Graph.* 29, 4 (2010), 38:1–38:6.

Fang Xianzhong, Bao Hujun, Tong Yiying, Desbrun Mathieu, and Huang Jin. 2018. Quadrangulation Through Morse-parameterization Hybridization. *ACM Trans. Graph.* 37, 4 (2018), 92:1–92:15.

Gurobi . 2018. Gurobi Optimizer Reference Manual. http://www.gurobi.com

Levi Zohar and Gotsman Craig. 2015. Smooth Rotation Enhanced As-Rigid-As-Possible Mesh Animation. *IEEE Trans. Vis. Comput. Graph.* 21, 2 (2015), 264–277.

Levi Zohar and Zorin Denis. 2014. Strict minimizers for geometric optimization. *ACM Transactions on Graphics (TOG)* 33, 6 (2014), 185.

Lévy Bruno, Petitjean Sylvain, Ray Nicolas, and Maillot Jérome. 2002. Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics* 21, 3 (2002), 362–371.

Lipman Yaron. 2012. Bounded distortion mapping spaces for triangular meshes. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 108.

Liu Ligang, Zhang Lei, Xu Yin, Gotsman Craig, and Gortler Steven J. 2008. A local/global approach to mesh parameterization. In *Computer Graphics Forum*, Vol. 27. 1495–1504.

Myles Ashish and Zorin Denis. 2012. Global parametrization by incremental flattening. *TOG* 31, 4 (2012), 109.

Myles Ashish and Zorin Denis. 2013. Controlled-distortion constrained global parametrization. *TOG* 32, 4 (2013), 105.

Panozzo Daniele, Puppo Enrico, Tarini Marco, and Sorkine-Hornung Olga. 2014. Frame fields: anisotropic and non-orthogonal cross fields. *ACM Trans. Graph.* 33, 4 (2014), 134:1–134:11.

Pinkall Ulrich and Polthier Konrad. 1993. Computing discrete minimal surfaces and their conjugates. *Experimental mathematics* 2, 1 (1993), 15–36.

Rabinovich Michael, Poranne Roi, Panozzo Daniele, and Sorkine-Hornung Olga. 2017. Scalable Locally Injective Mappings. *ACM Trans. Graph.* 36, 4 (2017).

Ray Nicolas, Vallet Bruno, Li Wan Chiu, and Lévy Bruno. 2008. N-symmetry direction field design. *ACM Transactions on Graphics* 27, 2 (2008), 10.

Sawhney Rohan and Crane Keenan. 2017. Boundary first flattening. *ACM Transactions on Graphics* 37, 1 (2017), 1–14.

Shtengel Anna, Poranne Roi, Sorkine-Hornung Olga, Kovalsky Shahar Z., and Lipman Yaron. 2017. Geometric Optimization via Composite Majorization. *ACM Trans. Graph.* 36, 4 (2017).

Smith Jason and Schaefer Scott. 2015. Bijective Parameterization with Free Boundaries. *ACM Trans. Graph.* 34, 4 (2015), 70:1–70:9.

Sorkine Olga. 2006. Differential Representations for Mesh Processing. *Computer Graphics Forum* 25, 4 (2006), 789–807.

Sorkine Olga and Alexa Marc. 2007. As-rigid-as-possible Surface Modeling. In *SGP*. 109–116.

Sorkine Olga, Cohen-Or Daniel, Goldenthal Rony, and Lischinski Dani. 2002. Bounded-distortion Piecewise Mesh Parameterization. In *Proceedings of the Conference on Visualization*. IEEE Computer Society, 355–362.

Tarini Marco, Puppo Enrico, Panozzo Daniele, Pietroni Nico, and Cignoni Paolo. 2011. Simple quad domains for field aligned mesh parametrization. In *Proceedings of SIGGRAPH Asia*. 1–12.

Vaxman Amir, Campen Marcel, Diamanti Olga, Panozzo Daniele, Bommes David, Hildebrandt Klaus, and Ben-Chen Mirela. 2016. Directional Field Synthesis, Design, and Processing. *CGF* (2016).

Weber Ofir, Myles Ashish, and Zorin Denis. 2012. Computing Extremal Quasiconformal Maps. *CGF* 31, 5 (2012), 1679–1689.

Zayer Rhaleb, Rössl Christian, Karni Zachi, and Seidel Hans-Peter. 2005. Harmonic guidance for surface deformation. In *Computer Graphics Forum*, Vol. 24. 601–609.