Seamless Global Parametrization in a Single Optimization
Guillaume Coiffier, Etienne Corman

To cite this version:
hal-03670525

HAL Id: hal-03670525
https://hal.science/hal-03670525
Preprint submitted on 17 May 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Seamless Global Parametrization in a Single Optimization

GUILLAUME COIFFIER, Université de Lorraine, CNRS, Inria, LORIA, France
ETIENNE CORMAN, Université de Lorraine, CNRS, Inria, LORIA, France

Fig. 1. We propose to solve a single optimization problem for simultaneously computing a frame field and a global seamless parametrization. This allows the singularity cones to be moved during optimization in order to minimize any differentiable distortion energy. Our results produce less distorted parametrizations than two-steps methods relying on integrating a precomputed frame field. This figure presents our parametrizations of the Rhino head model computed: (a) by integrating a smooth frame field, (b) by reducing the shear, (c) by maximizing isometric deformations, (d) by minimizing the LSCM energy [Lévy et al. 2002] effectively reducing the stretch. We represent positive $\pi/2$ cones in red and negative $-\pi/2$ cones in blue. Distortion measurements are computed as a mean over all triangles.

The challenge of computing high quality seamless parametrizations is to account for both the optimal placement of singularity cones and distortion minimization. Existing methods rely either on greedy solvers or on a simplified two steps problem, first finding the cone positions using a cross field and second optimizing the parametrization under fixed topology. Both approaches, however, loosen the link between cone placement and distortion, effectively leading to suboptimal solutions. We instead formulate the problem of global seamless parametrization as a single continuous optimization problem where singularities may appear anywhere inside triangles. To achieve this, the input mesh is subdivided into charts centered at vertices whose orientations are related by rotations along edges. We then exhibit a set of continuous constraints for the parametrization to be seamless, locally injective and aligned with feature and boundary edges. Formulated as a non-linear least-squares problem, these constraints can be easily optimized. We compare our algorithm to previous techniques on a variety of smooth and CAD models, often achieving lower distortion and more practical cone placement for subsequent quad remeshing.

CCS Concepts: • Computing methodologies → Mesh geometry models; Volumetric models.

Additional Key Words and Phrases: Seamless parametrization, quad-meshing, hex-meshing

1 INTRODUCTION

Meshes play a crucial role in numerical algorithms for geometry processing and scientific computing. They decompose the space into atomic cells allowing to store a discrete approximation of a continuous process. Depending on the application, meshes with different properties are needed. Quadrangle meshes have this unique property that they can tile space in a grid-like manner everywhere except at a few local exceptions called singularities or cones.

Singularities are bound to appear to compensate the topology of non-torus-like surfaces. These extraordinary points also have a great impact on the quad mesh distortion, i.e. how close to a perfect square are the quads. Depending on the surface curvature, the number of cones and their positions can drastically increase the mesh quality. And, on this matter, the fewer is not always the better. In fact, as singularity indices must be integer multiple of $1/4$, finding optimal positions minimizing the overall distortion is challenging and often leads to sophisticated integer programming problems.

To overcome this apparent complexity, current quad-meshing pipelines often relies on two-steps: first choosing the singularity locations using a cross field or other proxies and then compute a global seamless parametrization from which quads will be extracted. This procedure, however, loosens the link between cone placement and the distortion of the final mesh. As a consequence, quads can
stray far away from squares because the connectivity of the quad-mesh is optimized almost independently of its geometry.

This limitation is the starting point of this paper: we propose a single step seamless parametrization algorithm. In our setup, cones are positioned based on distortion minimization as part of a global optimization problem. This approach is in total opposition to greedy methods used in previous work heavily relying on local considerations. Moreover, we do not need integer variables making our system of equations solvable by any off-the-shelf continuous optimizer.

To achieve this, we rely on two key ingredients. First, we consider vertex-based cross field so that singularities can appear and move inside triangles, making the parametrization more flexible than face based frame field integration. Second, we unveil a necessary and sufficient system of equations for seamless parametrization which is independent of the cross field symmetries. This is done by using the so-called parallel transport introduced in previous work [Crane et al. 2010; Knöppel et al. 2013]. This integrability condition allows us to compute simultaneously a cross field and its parametrization while minimizing a distortion criterion. To achieve this, we rely on two key ingredients. First, we consider vertex-based cross field so that singularities can appear and move inside triangles, making the parametrization more flexible than face based frame field integration. Second, we unveil a necessary and sufficient system of equations for seamless parametrization which is independent of the cross field symmetries. This is done by using the so-called parallel transport introduced in previous work [Crane et al. 2010; Knöppel et al. 2013]. This integrability condition allows us to compute simultaneously a cross field and its parametrization while minimizing a distortion criterion.

Our results demonstrate that cone positions depend on the distortion energy and that we have lower average distortion than previous work including directly integrating a smooth frame field (see Fig. 1).

**Algorithm.** The overall algorithm takes as input a triangle mesh along with boundary and feature constraints and follows the steps:

1. Cut the input triangle mesh into disconnected charts centered at vertices (Sec. 3.2);
2. Flatten the boundary to fit the prescribed boundary and feature edge angles (Sec. 4.2);
3. Solve a non-linear least-squares optimization problem to obtain optimal charts, parallel transport and cross field (Sec. 5.5);
4. Reconstruct the final parametrization by inserting additional vertices at singular points (Sec. 3.3);
5. Reposition singular points in the initial mesh (Sec. 5.6).

The final results are a valid seamless parametrization and a vertex-based cross field with consistent cone points.

1.1 Related Work

As our goal is to build a "single step" seamless parametrization algorithm, it is interesting to consider our work as a specific instance of the broader problem of computing distortion minimizing injective maps [Hormann et al. 2008]. Given a target topology (disk [Lévy et al. 2002; Tutte 1963], orbifold [Aigerman and Lipman 2015], sphere [Kazhdan et al. 2012] or cone metric [Sawhney and Crane 2018; Springborn et al. 2008]), standard distortion minimization finds an optimal geometry in parameter space. In our setting, we aim to find both the geometry and the optimal topology minimizing our distortion criterion. Since this problem is particularly challenging, the most common approach is to first solve for the topology independently of the geometry using a practical proxy such as cross fields or cone placement. Only a few work attempt to directly solve for seamless mappings.

Our method falls into the category of direct seamless mapping with a cross field proxy but proceeds in a single optimization step.

**Cross field based parametrizations.** A cross field is an extremely useful tool for seamless parametrization: it has the same symmetry as a square and thus fixes the element orientation. Direction fields generation is generally well understood for surface meshes. A complete review of the literature is out-of-scope of this paper and we refer to the relevant surveys [de Goes et al. 2016; Vaxman et al. 2016]. In this paper, we will represent crosses by the 4 th root of a complex number [Palacios and Zhang 2007; Ray et al. 2008]. Most recent results guarantee that the smoothest cross field can be computed using a diffusion/renormalization scheme [Viertel and Osting 2019]. However, as shown by our experiments, the smoothest cross field topology does not necessarily minimize the parametrization distortion.

More precisely, a cross field is the rotation part of the QR decomposition of the Jacobian of the parametrization [Panozzo et al. 2014], so computing a parametrization implies recovering the missing part of the Jacobian. The simplest method is to find the functions whose gradient is closest to the field directions [Kälberer et al. 2007]. This can also be done using global periodic functions [Fang et al. 2018; Ray et al. 2006] with the advantage of constraining singularities to integer coordinates. A more stable approach traces the motorcycle graph of the field and then solves for the graph edge lengths [Myles et al. 2014].

Fu et al. [2016] take an ingenious approach: they divide the input mesh into a set of independent triangles and constrain adjacent simplices to be equal up to the cross field rotation. This simple constraint allows them to compute seamless global parametrization for triangle-based frame field. In this paper, we use a similar formulation but adapted to cross field defined at vertices so that singularities are not bound to appear at vertices.

While convenient, this proxy has one major drawback: not all frame fields can generate a valid parametrization. In particular, frame fields can exhibit limit cycles creating degenerated triangles in parameter space. Many solutions have been proposed to circumvent this issue. Limit cycles can be detected and fixed locally by adding a singularity dipole [Myles et al. 2014]. However, this solution stays local and does not take into account the global problem of minimizing the distortion. Diamanti et al. [2015] promote the frame field integrability by constraining it to be the Jacobian of a parametrization, but the distortion minimization has a weak impact on the solution. Therefore, it seems natural to compute the frame field and its parametrization at the same time, so that the frame field constrains the map to be seamless and the map forces the field to be integrable.

**Cone metric deformation.** Overall, a cross field is a convenient way of placing quantized singularities necessary for quad-meshing, however, it is not the only way to solve this problem. Most alternatives
Seamless Global Parametrization in a Single Optimization

1.2 Overview

Our main contributions:

- We compute a parametrization of a triangular mesh from any edge-based parallel transport rotation, allowing singularities to appear anywhere inside triangles;
- We derive a set of necessary and sufficient conditions for seamless parametrization;
- We propose an algorithm optimizing, at the same time, a cross field and its parametrization so that singularities can move in order to reduce the overall parametrization distortion.

Organization. The paper is organized as follows. We first give a formal definitions and properties of seamless maps (Sec. 2.1). We then continue, in Sec. 3, by exposing the key construction of this paper: a constraint for a parametrization to be compatible with a parallel transport. In Sec. 4, we further constrain the parametrization to be seamless and adapted to the boundary, leading to a global optimization problem solved in Sec. 5. The results are evaluated and compared with previous methods in Sec. 6.

2 GLOBAL PARAMETRIZATIONS

In this section, we give a formal definition to three key properties of parametrization mappings used throughout the paper: seamless parametrization, feature adaptation and integer mapping. These properties are illustrated in Fig. 2.

Let $M = (V, E, T)$ be a triangle mesh, consisting of vertices, edges and triangles. A piecewise linear parametrization is a map $f : M \subset \mathbb{R}^3 \rightarrow \mathbb{C}$ assigning, to every point $x \in \mathbb{R}^3$ inside a triangle $t \in T$, a coordinate in the (complex) plane $f_t(x) = (u,v) \in \mathbb{C}$. Such a parametrization can exhibit cuts along some edges, namely adjacent triangles in the initial mesh are no longer adjacent in the parametrization. The linear functions $g$ relating an edge on one side of the cut to its duplicated version on the other side of the cut are called transition functions (see Fig. 2a). Given a closed loop of triangles $(t_0, t_1, \ldots, t_p, t_0)$, all incident to the same vertex $i \in V$, the vertex is said singular if the accumulated transition function $g_i = g_{t_{i+1}} \circ \ldots \circ g_{t_{i+1}} \circ g_{t_i}$ is not identity.
where vertices have two integer coordinates [Lyon et al. 2016], as shown vectors are equal up to a mal and the dual mesh of \( M \) edge. This system of constraints is applied to a collection of vertex In this section, we prove a necessary and sufficient system of equa- of a cross field. Instead we propose to cuts are not given, moving a singularity requires to change the transition functions of multiple triangles. Instead we propose to by other methods like [Bommes et al. 2013; Campen et al. 2015].quantization step is seen as an independent problem possibly, solved in Fig. 2c. In this paper, we do not aim for integer maps and limit discrete case, the holonomy of a loop is just the “rotation defect” of a vector transported around the loop. When the loop enclose a single triangle, the holonomy angle is the discrete Gaussian curvature.

We say that a parametrization is seamless if duplicated edge vectors are equal up to a \( k\pi/2 \) rotation for any two adjacent triangles (Fig. 2a). Namely, the transition functions are square preserving.

**Definition 2.1. (Seamless)** We call a parametrization seamless if for two adjacent triangles \( t_1, t_2 \in T \) sharing an edge \( ij \in E \), their edge vectors are related by the transition function:

\[
 f_i(p_1) - f_i(p_2) = e^{i\pi/2} (f_{i_0}(p_1) - f_{i_0}(p_2)) , \quad k \in \{0, 1, 2, 3\}.
\]

Another important property is that the surface boundaries and feature edges are isolines of the parametrization (Fig. 2b).

**Definition 2.2. (Adapted)** A parametrization is adapted to its boundary if all feature edges and boundary edges \( ij \in E \) of a boundary triangle \( t \in T \) have one zero coordinate in parameter space:

\[
 \langle f_i(p_1) - f_i(p_2), e_k \rangle = 0 , \quad k \in \{1, 2\},
\]

where \( e_1 = 1 + \theta, e_2 = 0 + i \) are the plane axes.

In order to be able to extract quads from a parametrization, we need boundary vertices to have one integer coordinates and singular vertices have two integer coordinates [Lyon et al. 2016], as shown in Fig. 2c. In this paper, we do not aim for integer maps and limit ourselves to seamless and boundary adapted parametrizations. The quantization step is seen as an independent problem possibly, solved by other methods like [Bommes et al. 2013; Campen et al. 2015].

Def. 2.1 is challenging to use in an optimization process: if the cuts are not given, moving a singularity requires to change the transition functions of multiple triangles. Instead we propose to compute seamless map by “spreading” the transition rotations over the entire mesh, namely integrating the parallel transport of a cross field.

### 3 PARAMETRIZATION FROM A PARALLEL TRANSPORT

In this section, we prove a necessary and sufficient system of equations (Eqs. (E)) for a parametrization to have the same holonomy as a given parallel transport – an assignment of angle per oriented edge. This system of constraints is applied to a collection of vertex charts \( M^C \) (see Fig. 5b), emanating from the intersection of the primal and the dual mesh of \( M \). Our output parametrization may have a different connectivity than the input mesh, since singularities may appear inside triangles. Therefore, we do not parametrize \( M \), but the triangle mesh \( M_p \) where singular triangles are subdivided by inserting a vertex as represented in Fig. 5c.

This parametrization process is summarized in Fig. 4.

#### 3.1 Discrete Parallel Transport

We consider tangent vectors defined at vertices. As in [Knöppel et al. 2013], tangent planes are mapped to the complex plane, so a tangent vector is a complex number in a local coordinate system. To compare vectors at neighboring vertices \( ij \), they must be rotated by an angle \( \omega_{ij} \) to take into account the change of local coordinates. Transporting a vector from tangent space to tangent space along an edge path is called the discrete parallel transport.

A parallel transport is nothing more than the assignment of an angle \( \omega \) per oriented edge. Therefore, a vector \( v_i \in \mathbb{C} \) at vertex \( i \) is parallel transported to vertex \( j \) by the rotation: 

\[
 e^{i\omega_{ij}/v_i} e_i = \exp(\omega_{ij}) v_i .
\]

Note that the transport from \( j \) to \( i \) is given by the inverse rotation thus: \( \omega_{ij} = -\omega_{ji} \).

The integrability constraints built in this section requires only an assignment of a rotation angle per edge. Thus, we do not need to know how the local bases are constructed and this discussion is delayed to Sec. 4.1.

**Holonomy.** The parallel transport is deeply related to the notion of curvature. Indeed, when transporting a vector around a closed loop, it will in general not return to its original orientation. The rotation between the original vector and the transported one is the called the holonomy [Crane et al. 2010] and is illustrated in Fig. 3. Given a vertex closed loop \( y = (i_0, i_1, \ldots, i_N, i_0) \), the holonomy \( \Phi_\omega \) around \( y \) is simply the composition of all rotations along the path:

\[
 \Phi_\omega(y) := \exp(i\omega_{i_0 i_1}) \cdots \exp(i\omega_{i_N i_0}) .
\]

When the loop is reduced to a triangle \( ijk \in T \), the discrete Gaussian curvature \( K \) is defined as the angle of the holonomy rotation [2013]:

\[
 \exp(iK_{ijk}) = \exp(i\omega_{ik}) \exp(i\omega_{jk}) \exp(i\omega_{ij}) . \tag{1}
\]

Of course, a valid parallel transport must satisfy the Gauss-Bonnet theorem:

\[
 \sum_{t \in F} K_t = 2\pi \chi,
\]

where \( \chi \) is the Euler characteristic of \( M \). If this is not the case, the topology of the parallel transport does not match the one of the triangle mesh, so a parametrization cannot be defined and our system of equations (E) does not have a solution. In Sec. 4 we will show how to build a parallel transport compatible with a cross field following the steps of Knoppe et al. [2013].

Our goal is to find a parametrization embedding whose transition functions match a given parallel transport holonomy. Namely, a triangle with zero Gaussian curvature will be parametrized by a flat
We explicitly build the chart vectors in two different bases and, therefore, must be equal up to a vector in \( \mathbb{C} \). Overall, the charts must satisfy angle \( \omega \) consistent with the parallel transport, the transfer of a tangent prescribes relative orientation of tangent planes. For a chart to be singularity. Moreover, underscripts are reserved to make explicit of an edge and sing of the input mesh in parameter space, the letter are highlighted in red in Fig. 5b.

3.2 Parametrization Constraints

Vertex charts and chart collection. In order to parametrize an edge based parallel transport, we must define charts based at vertices. A chart \( C_i \) is a small quad-mesh around vertex \( i \) living in parameter space, in our case the complex plane. Charts are disconnected from each other. We define the chart collection \( M^C = \cup_{i \in V} C_i \) as the set of all charts emanating from the subdivision of the input mesh.

As singularities appear inside triangles, the charts must include a point inside each triangle where a singularity may appear. For this reason, we build our charts as the intersection of mesh \( M \) with its dual. Namely, each triangle is split into three quads, as in Fig. 5b, by adding three edges linking the triangle center to mid-edge points. We explicitly build the chart \( C_i \) at vertex \( i \) as the union of quads containing \( i \). In this construction, adjacent charts are disjoint but share two types of duplicated edges: the primal edges from the input mesh and the dual edges linking edge center to triangle center. They are highlighted in red in Fig. 5b.

To ease the notation, we use specific letters for each point type. The letter \( p \) denotes the vertex coordinates (with a complex number) of the input mesh in parameter space, the letter \( e \) is the midpoint of an edge and \( s \) the center of the triangle and thus a potential singularity. Moreover, underscripts are reserved to make explicit the membership of a vector to a chart. For example, \( s_{ij}^{e_i} - e_i \) is the vector connecting the edge midpoint to a triangle center in the chart \( C_i \) whereas \( s_{jk}^{e_i} - e_i \) is the duplicated edge in the chart \( C_j \).

Parallel Transport Constraints. By definition, a parallel transport preserves relative orientation of tangent planes. For a chart to be consistent with the parallel transport, the transfer of a tangent vector in \( C_i \) to an adjacent chart \( C_j \) should undergo a rotation of angle \( \omega_{ij} \). Any edges shared by two adjacent charts are tangent vectors in two different bases and, therefore, must be equal up to a rotation by the parallel transport. Overall, the charts must satisfy the integrability constraints:

\[
\begin{align*}
\forall ij \in E: \quad e_j^* - p_j &= \exp(\omega_{ij})\left(e_i^* - p_i\right), \\
\forall ijk \in F: \quad s_j^{e_i} - s_j^{e_i} &= \exp(\omega_{ijk})\left(s_k^{e_i} - s_j^{e_i}\right)
\end{align*}
\]  

Fig. 5. The initial mesh (a) is subdivided in a chart collection (b) used for imposing integrability constraints. Edge midpoint are denoted with letter \( e \) and triangle center point with the letter \( s \). We do not parametrize the initial mesh but a subdivision of it where singularities \( s \) are inserted inside singular triangles (c).

3.3 Parametrization Reconstruction

In this section, we assume that the charts satisfy the system of equations in Eqs. (E). We will show how to recover a parametrization from the chart collection \( M^C \).

Triangle Parametrization. The charts have coordinates in parameter space, so to recover the parametrization of a triangle \( ijk \) we simply have to reassemble the three pieces separated into charts \( C_i, C_j \) and \( C_k \). This is always possible because Eqs. (E) ensure that
duplicated edges have same length. Thus, the triangle piece in $C_j$ can be stitched to the piece in $C_i$ by applying the rigid transformation $x \mapsto \exp(\omega_{ji})x - e_j^i + e_i^j$. Synchronizing $C_j$ and $C_k$ with $C_i$ leads to a triangle parametrization.

We distinguish two cases: (1) the synchronization creates a valid triangle and the final parametrization is obtained by removing dual edges (Fig. 6a), (2) a cut appears so we triangulate the synchronization by adding edges linking the singular point $s$ and triangle vertices (Fig. 6b).

Mesh Parametrization. Given the parametrization of each triangle, it is easy to recover the parametrization of the entire mesh $M_p$ by gluing adjacent triangles along a dual spanning tree. The overall algorithm is summarized in Alg. 1 and by Fig. 4.

Crucially, we can prove that if a chart collection satisfies constraints Eqs. (E), then the parametrization transition functions are determined by the parallel transport holonomy. In particular, a regular triangle (i.e. $K_{ijk} = 0$) is mapped to a regular triangle in the parametrization and singular triangle admits a singularity at point $s_{ijk}$ with an angle defect equal to $K_{ijk}$. Interestingly, only the parallel transport holonomy matters: changing $\omega$ while keeping the holonomy constant does not change the set of solutions of Eqs. (E).

**Theorem 3.1.** Given a parallel transport $\omega$, if the vertex charts satisfy Eqs. (E) then the transition functions of the parametrization constructed with Alg. 1 are equal to the parallel transport holonomy.

Conversely, given any parametrization whose singular vertices are of valence three, there exists a parallel transport $\omega$ and vertex charts satisfying Eqs. (E).

In particular, the parametrization is seamless if and only if the holonomy of $\omega$ around any loops is a rotation by an integer multiple of $\pi/2$.

---

**Algorithm 1: ParametrizationReconstruction**

**Input:** A parallel transport $\omega$ and a chart collection $M^c$ satisfying Eqs. (E).

**Output:** A mesh $M_p$ and its parametrization $M_p \leftarrow M$.

**forall** $ijk \in T$ **do**

Stitch $C_j$ and $C_k$ with $C_i$.

if $K_{ijk} = 0$ then

The parametrization of $ijk$ is a triangle: remove dual edges.

else

Triangulate the charts.

Insert a vertex inside triangle $ijk$ in $M_p$.

end

end

Add an arbitrary initial triangle to a queue.

**while** Queue not empty **do**

Find the translation/rotation matching the edge shared with previous triangle.

Add adjacent triangles to queue.

end

An immediate corollary to Thm. 3.1 is that a parametrization is seamless if and only if the parallel transport is compatible with an integrable cross field.

The parallel transport curvature is not explicitly present in Eqs. (F), so that the singularity locations are needed only during the reconstruction. This is the great benefit of our approach, allowing us to change the curvature while optimizing for the parametrization.

4 SEAMLESSNESS AND BOUNDARY ALIGNMENT

So far, we have found conditions to define a parametrization from a parallel transport $\omega$. However, we do not have guarantees that this parametrization is seamless, adapted to the boundary or locally injective. In this section, we present three additional necessary and sufficient set of constraints to satisfy these properties.

4.1 Seamlessness Constraints

According to Thm. 3.1, a parametrization is seamless if it is compatible with a parallel transport whose holonomy is square preserving. This property is deeply related to cross fields.

**Discrete Cross Fields.** A cross field assigns at each vertex a set of two orthogonal directions:

$$\left\{e^{i k \pi/2} v_i, \quad k = 0, 1, 2, 3 \right\}.$$

A key insight is that, when raised to the 4th power, this set reduces to a single complex value:

$$z_i := v_i^4$$

Thus, a cross can be uniquely represented by a unit complex number whose fourth roots gives the four vectors of the cross [Palacios and Zhang 2007; Ray et al. 2008].

Since $v_i \in \mathbb{C}$ is a tangent vector, it can be transported to an edge adjacent vertex by a parallel transport. Conversely, a vector field defines a parallel transport as the rotation relating any two adjacent vectors. The same can be said of an arbitrary cross field. When
represented by the fourth root of a unit vector, it defines a parallel transport $\omega$ as:

$$ z_i = \exp(4\omega_{ij})z_j. \quad (2) $$

This parallel transport, associated to a cross field, has the distinctive property of preserving the cross symmetry. In particular, it means that its holonomy around any loop are rotations by an angle of $k\pi/2$.

As the cross field lives in the complex tangent space, Eq. 2 is purely combinatorial in the sense that the geometry of $M$ is not necessary to build it. However, for visualization purposes and for the stability of our optimization scheme, it is important to build local bases and give the cross field a geometrical meaning.

Parallel transport from Levi-Civita connection. The construction of a vertex-based frame field follows the one by Knöppel et al. [Knöppel et al. 2013]. A tangent vector $v_i \in \mathbb{C}$ is expressed in the local basis of the vertex $i$. This local basis defines a tangent space at this vertex. Vertices on a triangulated surface are generally not flat as the inner angles do not sum to $2\pi$. Therefore any parallel transport can be decomposed into the sum of the cross field rotation $\alpha$ and the Levi-Civita parallel transport $\rho$:

$$ z_i = \exp(4\alpha_{ij})r_{ij}^4z_j. \quad (F) $$

In our optimization scheme we use the cross field rotation as a variable as it can be easily bounded and initialized.

$\rho : E \to \mathbb{R}$ as the rotation angle aligning the basis at $j$ to the one at $i$:

$$ \rho_{ij} : = \phi_{ij} = \phi_{ij} + \pi, $$

by comparing the angles of the shared edge $ij$. The change of basis is then $r_{ij} = \exp(\rho_{ij})$. By construction, this parallel transport is associated to the Levi-Civita connection. As shown by Knöppel et al. [Knöppel et al. 2013], its Gaussian curvature, as defined in Eq. (1) satisfies the Gauss-Bonnet theorem (for meshes without boundaries).

Therefore, any parallel transport can be decomposed into the sum of the cross field rotation $\alpha$ and the Levi-Civita parallel transport $\rho$:

$$ z_i = \exp(4\alpha_{ij})r_{ij}^4z_j. \quad (F) $$

4.2 Boundary and feature edge constraints

Another requirement we discussed in Section 2 is that the parametrization should be adapted to the mesh boundary and features edges, so additional constraints are required on the cross field and on the vertex charts. We process feature edges extracted from a boundary or user specified in the same manner.

Single vertex corner. We define a vertex corner as a set of adjacent triangles incident to the same vertex $i$ and delimited by two feature edges $ij_0$ and $ij_n$. In this paragraph, we consider a single vertex corner. We assume that target corner angles $\Omega_i^{b_{j0}a}$ integer multiple of $\pi/2$ are provided as input. At a corner $ij_0i_a$, inner angles are normalized to match the prescribed angle:

$$ \hat{\theta}_{ij}^{p_{j0}p_{i1}} = \Theta_i^{b_{j0}a} \hat{\theta}_{ij}^{p_{j0}p_{i1}} / \Theta_i^{b_{j0}a}, $$

where $\Theta_i^{b_{j0}a}$ is the total inner angle between edges $ij_0$ and $ij_n$. The edge angles in the tangent plane are obtained by accumulating the modified inner angles as in Eq. (3). The representation vector $z_i$ is constrained to be equal to $e^{4\phi_{ij, p_{j0}}}$, by construction the vectors of this cross agrees with the feature edge directions. An illustration of this procedure is given in Fig. 7.

The corresponding vertex corner in chart $C_i$ must also be constrained to the corner angle. To do so, the primal edge vector corresponding to a feature edge must remain orthogonal to the feature edge normal. Therefore, at a vertex corner, we enforce three constraints linear with respect to the chart coordinates and the cross field:

$$ z_i = \exp(4\alpha_{ij})r_{ij}^4z_j, $$

$$ (\varphi_{ij} - \alpha_{ij}) + \exp(\varphi_{ij}) = 0, $$

$$ (\varphi_{ij} - \alpha_{ij}) - \exp(\varphi_{ij}) = 0. $$

Note that since the cross field is constrained on feature vertices, the rotation $\alpha$ is also known along feature edges.

Singular vertices. Let us consider the case where an interior vertex $i$ is incident to multiple feature edges $(ij_a, ij_a, \ldots, ij_n)$ forming $n + 1$ corners. If the sum of all corner angles $\Omega_i = \sum_{p=0}^{n-1} \Omega_i^{f_{j0}f_{p+1}} + \Omega_i^{f_{na}f_{j0}}$ is a multiple of $2\pi$, then the flattening of each corner creates a valid vertex chart. However, when $\Omega_i \neq 2k\pi$, vertex $i$ is singular and an additional seam is necessary to lay out edges in the tangent plane. Thus, we cut open the vertex neighborhood at edge $ij_0$ introducing a duplicated edge $ij_0'$. For the parametrization of chart $C_i$ to remain seamless, edges on each side of the cut must have equal
length yielding an additional linear constraint:
\[
\langle p_i - e_i^\alpha, \exp(p_{ij}^\alpha) \rangle = \langle p_i - e_i^\alpha, \exp(p_{ij}^\alpha) \rangle. \tag{B}
\]

Fig. 8 illustrates the computation of the tangent plane at a cube corner.

4.3 Local injectivity

We need to ee that the final parametrization is locally injective, in other words triangle areas must remain positive in parameter space. This boils down to two different constraints. First, the orientation of primal edges in vertex charts should be preserved:
\[
\det(p_i - e_i^\alpha, p_i - e_i^\alpha) > 0. \tag{D_1}
\]

Second, the singularity point \( s_{ijk} \) should stay inside triangle \( ijk \):
\[
\begin{align*}
\det(p_i - e_i^\alpha, p_i - s_{ijk}^\alpha) & > 0, \\
\det(p_i - s_{ijk}^\alpha, p_i - e_i^\alpha) & > 0. \tag{D_2}
\end{align*}
\]

4.4 Theoretical guarantees

The constraints introduces in Sections 3, 4.1, 4.2 and 4.3 define a system of equations whose solution is a boundary adapted seamless parametrization. Most importantly all of these constraints are independent of the cross field symmetries and do not rely on integer variables. Therefore they are readily usable in any standard continuous solver.

**Theorem 4.1.** If a set of charts, a cross field and a parallel transport satisfy the system of Eqs. (E), (F), (B), (D_1) and (D_2), then we can recover a locally injective, boundary adapted seamless parametrization. Moreover, the singular vertices appear only in singular faces with the angle defect prescribed by the parallel transport curvature.

5 DISTORTION MINIMIZING MAP

So far, we have defined a system of necessary and sufficient equations for computing valid global seamless parametrizations. In this section, we describe our optimization process in order to find vertex charts, cross field rotations \( \alpha \) and representation vector \( z \) solutions to this system.

5.1 Simplifying constraints using the Cayley transform

One great challenge when numerically solving Eqs. (E) and (F) is the presence of a periodic complex exponential. Luckily, there exists a clever way to parametrize the space of rotations by non-periodic polynomial functions, thus considerably reducing the complexity of the problem. The **Cayley map** [Kobilarov et al. 2009; Zhang et al. 2021], noted \( \text{cay} \), is a complex fraction equal to the rotation by an angle \( \tan(\alpha/2) \):
\[
\text{cay}(\alpha) := \frac{1 - \text{exp}(\tan\frac{\alpha}{2})}{1 + \text{exp}(\tan\frac{\alpha}{2})}
\]

Instead of using the cross field rotation angle \( \alpha_{ij} \), we will consider its tangent half angle. So, using the change of variables \( \alpha_{ij} = \tan(\alpha_{ij}/2) \), Eqs. (E) and (F) can be rewritten as polynomial equations:
\[
\begin{align*}
(1 + i\alpha_{ij})(e_i^j - p_i) &= (1 - i\alpha_{ij})(e_i^j - p_i) \\
(1 + i\alpha_{ij})(e_i^j - e_i^i) &= (1 - i\alpha_{ij})(e_i^j - e_i^i) \tag{E}
\end{align*}
\]

\[
\begin{align*}
(1 + i\alpha_{ij})s_{ijk} &= (1 - i\alpha_{ij})s_{ijk}^i s_{ijk}^j \\
&= (1 - i\alpha_{ij})s_{ijk}^i s_{ijk}^j \tag{F}
\end{align*}
\]

5.2 Log-barrier

The inequality constraints of Eqs. \( D_1 \) and \( D_2 \) are enforced using the continuously differentiable barrier function \( B_\eta : \mathbb{R}_{>0} \to \mathbb{R} \):
\[
B_\eta(x) = \begin{cases}
+\infty & x \leq 0, \\
\log(\frac{x}{\eta}) & 0 < x \leq \eta, \\
0 & \eta \leq x.
\end{cases}
\]

Given a determinant \( d \) whose initial value is \( d_0 \), we set the threshold to \( \eta = d_0/2 \).

5.3 Optimization

We would like to find the chart vertex coordinates \( x, y \), the frame rotation \( \bar{\alpha} \) and the cross field \( z \) solution of the constrained optimization problem:
\[
\min_{\bar{\alpha}, z, x, y} E(x, y) \\
\text{s.t.} \quad (\bar{\alpha}, (\bar{F}), (B), (D_1), (D_2)). \tag{4}
\]

where \( E : \mathcal{M}^c \to \mathbb{R} \) is any given distortion energy.

To make the problem tractable, we enforce non-linear constraints as a sum of squared energy:
\[
R(\bar{\alpha}, z, x, y) = \lambda_E ||(\bar{\alpha})||^2 + \lambda_F ||(\bar{F})||^2 + \lambda_{det} ||B_\eta(D_1)||^2 + \lambda_{det} ||B_\eta(D_2)||^2.
\]
leading to a non-linear least-squares optimization with linear constraints:

$$\min_{\alpha, z, x, y} eE(x, y) + R(\alpha, z, x, y) \quad \text{s.t.} \quad (\mathcal{B}).$$  \hspace{1cm} (5)$$

The Levenberg-Marquardt algorithm is tailored for minimizing non-linear least squares with zero being the global minimum. We follow the implementation of [Marumo et al. 2020] whose evolution of the damping parameter ensures global convergence toward a local minimum and a second order rate of convergence near a zero of the objective function. Each step involves solving a quadratic optimization problem with linear constraints. For this, we use the open source solver OSQP [Stellato et al. 2020].

The problem in Eq. (5) is solved several time for decreasing values of $\epsilon$. For our distortion minimizing experiments, we start at $\epsilon = 10$ and divide by a factor of 10 until $\epsilon = 10^{-4}$. Unless specified otherwise, we set all weights $\lambda$ to 1.

Note that for surfaces with boundary or feature edges, we do not need to force $z$ to be unit norm. It suffices that the cross field is constrained somewhere so that whenever Eqs. (\mathcal{F}) are satisfied, the cross is unitary everywhere. For surfaces without boundary or feature edges, we constrain $z$ at a random vertex to avoid the trivial zero solution.

### 5.4 Initialization

As for any non-convex optimization, the initialization should be chosen carefully.

The charts are initialized using the tangent plane flattening introduced in Secs. 4.1 and 4.2. Given the normalized triangle inner angles, the chart primal edges are laid out in the plane with their initial edge lengths. The chart is completed by placing singular points $s$ at the triangle barycenters. These initial charts keep the triangle edge lengths unchanged so they can be seen as an isometric parametrization of the input triangle mesh.

The frame field is set to zero everywhere except at boundary and feature edges where it is fully constrained (see Sec. 4.2).

For surfaces without boundary or feature edges, the frame rotation $\alpha$ is simply set to zero. However, in some cases this initialization can lead to highly distorted solutions. Typically, for CAD models where feature edges meet at very acute angles, a singularity appears directly inside the sharp triangle. Inspired by [Desobry et al. 2021], we compute an initial frame rotation $\alpha$ by forcing the cross field to be regular on all triangles belonging to 3-ring $T(i)$ of a sharp corner $i$. In practice, we solve the quadratic problem:

$$\min_{\alpha \in \mathbb{R}^E} ||\alpha||^2$$

s.t. $\alpha_{ij} + \alpha_{jk} + \alpha_{ki} = -K_{ijk}$, $\forall ijk \in T(i)$

$$\alpha_{ij} = a_{ij}^0, \quad \forall ij \in \partial E$$

where $a_{ij}^0$ are the frame rotations along feature edges. Fig. 9 illustrates how this initialization drastically changes the parametrization and reduces the distortion.

### 5.5 Distortion Energy

Among all possible valid seamless parametrizations, we would like to find the one minimizing a distortion criterion. In theory, we could use any distortion energy on the charts. In practice, we mostly focus on quad remeshing application, so we would like to avoid as much as possible shearing i.e. quads with non-orthogonal edges. The simplest way to promote these square-like quads is to penalize parametrizations whose parameter gradients are non-orthogonal. Thus, as a distortion measure, we use the LSCM [Lévy et al. 2002] energy sum over all charts:

$$E(x, y) = \sum_{i \in V} \sum_{t \in T_i} ||\nabla y_t - J \nabla x_t||_2^2,$$

where $(x, y)$ are the chart vertex coordinates and $J$ is the $90^\circ$ counter-clockwise rotation around the normal. The gradient operator is computed with respect to the initial isometric chart parametrization described in Sec. 5.4. According to Prop. 5.1, this energy is zero if and only if the vertex positions $(x, y)$ are equal to the initial chart coordinates up to a rotation and a global scaling. Thus, this energy penalizes shearing and stretching deformations of each vertex rings while staying quadratic.

**Proposition 5.1.** The LSCM energy of a flat triangle mesh vanishes only for rotations and global scaling of the vertex positions.

Unless specified otherwise our results are obtained with the LSCM energy.

### 5.6 Singularity Repositionning

Once the parametrization is reconstructed, each singular triangle is split into three (Fig. 6b). However, the position of the singular point on the surface mesh $M_p$ is not yet optimally determined. We would like to find the vertex position $s_{ijk}$ on the surface mesh minimizing the distortion of the three incident triangles $(t_1, t_2, t_3)$. To do so, we optimize the as-rigid-as-possible (ARAP) energy:

$$s_{ijk} = \arg\min_{x \in \mathbb{R}^E} \sum_{i = 1}^3 A_i \|J_i(x) - R_t\|^2$$

where $A_i$ is the triangle area in parameter space and $J_i$ is the Jacobian matrix from uv-coordinates to the surface mesh $M_p$. We solve this problem alternating minimization in $x$ and in the rotation $R$, as described by Sorkine et al. [Sorkine and Alexa 2007].
6 EVALUATION AND RESULTS

We evaluate our method on a variety of triangular meshes, both with and without feature edges. Models were taken from databases gathered by [Myles et al. 2014], [Levi 2021a] and the Mambo CAD dataset. All our results are included in supplemental material.

Quantitative results are gathered in Table 1 for model without features and in Table 2 for CAD models. We choose two evaluation criterion: the number of cone singularities and the average distortion.

In the following section, we provide more in detail comparison with previous works by considering the scale distortion, defined as the determinant of Jacobian, this metric is scaled with respect to the total area so that the optimal value is 1. Moreover, we plot distortion histograms in log scale of the triangle count. When quad meshes are provided, they are obtained by computing an integer grid map using [Bommes et al. 2013] and by then extracting quads with [Ebke et al. 2013]. Most importantly, they have the same singularities (positions and indices) as the input parametrizations. More results and comparisons can be found in the supplemental material.

All our experiments were conducted on a Ubuntu workstation with a height-core, 2.6-GHz Intel Core i5. Our python implementation takes a few minutes on a thousand triangle mesh and up to an hour for 30k triangles.

Convergence and local minima. As shown by Thm. 4.1, we have mathematical guarantees that if our constraints are satisfied, then we can extract a global parametrization. However, our optimization problem in Eq. (5) is non-convex and could get stuck in a local minimum. In practice, we never encounter this situation. To our understanding, the objective function can always be locally decreased by either reducing the norm of $z$ or by making the cross field better satisfy Eq. (5). Thus, assuming $\epsilon = 0$, our optimization either converges toward a zero of the objective or tries to reach a global minimum at infinity.

The latter case is typically created by a frame field with a limit cycle whose parametrization is degenerated. Fig. 17a shows an example of such a configuration. Our optimization with standard parameters is not able to escape this limit cycle.

However, we can lower the weight on the cross field constraint to $\lambda_F = 0.1$, so that the edge constraint dominates the energy, and switch to the isometric distortion energy:

$$E(x, y) = \sum_{t \in T} \sum_{I \in C_t} \|J^T I J - I\|_2^2. \tag{6}$$

where $J_t$ is the Jacobian of the triangle $t$. This way our algorithm converges to a global minimum and introduces two new singularities as in Fig. 17b. Thus, our method does deliver its promises: the parametrization and the frame field influence each other in order to create a valid parametrization.

Influence of distortion energies. Fig. 1 illustrates the influence of distortion energies on singularity positions. The LSCM energy (Fig. 1d) is compared with the isometric energy in Eq. (6) (Fig. 1c) and with the shear energy:

$$E(x, y) = \sum_{t \in T} \sum_{I \in C_t} \| \nabla y^T J \nabla x \|^2_2.$$

penalizing orthogonal but possibly stretch quads (Fig. 1b). As expected the singularity positions depends on the distortion being optimized. For this model, the LSCM energy is best to minimize both stretching and shearing of the parametrization.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>62 1.197</td>
<td>60 1.339</td>
<td>- -</td>
<td>44 1.202</td>
<td>40 1.152</td>
</tr>
<tr>
<td>Bunny</td>
<td>50 1.132</td>
<td>56 1.171</td>
<td>- -</td>
<td>34 1.135</td>
<td>32 1.124</td>
</tr>
<tr>
<td>Dilo</td>
<td>82 1.331</td>
<td>140 1.397</td>
<td>204 1.314</td>
<td>72 1.347</td>
<td>70 1.241</td>
</tr>
<tr>
<td>Dancer</td>
<td>66 1.308</td>
<td>146 1.348</td>
<td>105 1.228</td>
<td>66 1.243</td>
<td>70 1.200</td>
</tr>
<tr>
<td>Shark</td>
<td>64 1.199</td>
<td>187 1.522</td>
<td>105 1.202</td>
<td>66 1.280</td>
<td>64 1.283</td>
</tr>
<tr>
<td>Rhino</td>
<td>84 1.138</td>
<td>- -</td>
<td>60 1.189</td>
<td>62 1.113</td>
<td></td>
</tr>
<tr>
<td>Spot</td>
<td>35 1.160</td>
<td>- -</td>
<td>40 1.181</td>
<td>42 1.131</td>
<td></td>
</tr>
<tr>
<td>Head</td>
<td>93 1.184</td>
<td>- -</td>
<td>20 1.276</td>
<td>8 1.085</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Comparison of the number of singularity cones and the mean stretch distortion for eight models. Dash indicate missing values.

<table>
<thead>
<tr>
<th>Model</th>
<th>[Levi 2021a] Singus Stretch</th>
<th>Fixed Frame Field Singus Stretch</th>
<th>Ours Singus Stretch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sculpt</td>
<td>217 1.422</td>
<td>16 1.288</td>
<td>16 1.206</td>
</tr>
<tr>
<td>Beeble</td>
<td>50 1.505</td>
<td>42 1.358</td>
<td>74 1.215</td>
</tr>
<tr>
<td>Metatron</td>
<td>188 1.833</td>
<td>112 1.431</td>
<td>34 1.338</td>
</tr>
<tr>
<td>Fandisk</td>
<td>68 1.284</td>
<td>36 1.369</td>
<td>116 1.512</td>
</tr>
<tr>
<td>Casting</td>
<td>126 1.512</td>
<td>120 1.423</td>
<td>116 1.512</td>
</tr>
<tr>
<td>Hilbert</td>
<td>- -</td>
<td>248 1.000</td>
<td>248 1.000</td>
</tr>
<tr>
<td>M6</td>
<td>- -</td>
<td>60 1.183</td>
<td>52 1.186</td>
</tr>
<tr>
<td>S40</td>
<td>- -</td>
<td>48 1.215</td>
<td>48 1.184</td>
</tr>
</tbody>
</table>

Table 2. Comparison of the number of singularity cones and the mean stretch distortion for eight models with feature edges. Dash indicate missing values.

https://gitlab.com/franck.ledoux/mambo
Comparison with frame field integration. In this experiment, we investigate the benefits of optimizing for the rotation \( \alpha \) along with the parametrization. To this end, we run our algorithm with a fixed frame field generated by Viertel et al. [Viertel and Osting 2019]. Although the two parametrizations are optimized for the same distortion energy, we remark that our free cone placement always outperforms the fixed frame field method in term of average distortion (Tables 1 and 2). Thus, we consistently obtain a better singularity distribution for stretching minimization. For the Metatron, Fig. 13 shows that reducing the number of cones by 30% leads to an equivalent stretch distribution over triangles at the price of more area distortion.
Comparison with [Levi 2021a]. Levi [2021a] pursues the same goal as us: directly computing a seamless parametrization from a triangle mesh. His method is greedy and is biased by its initialization. For the Dilo model in Fig. 11, the singularities appear on a single side of the mesh seem unrelated to the geometry. The parametrizations provided by the author minimize the maximum stretch. This can be observe in Fig. 12 where the stretch bounded. In comparison, our method yields a higher maximal value but a smaller stretch for a
majority of triangles (see histograms), as well as a reduced number of cone singularities – 16 against 217. Our optimization does not attempt to reduce triangle scaling thus our area distortion is not as competitive. Moreover, a great number of triangles tend to be stuck around 0.5 because of the barrier function threshold.

We also compare quad-meshes computed from our respective parametrizations in Figs. 15 and 14. In both cases, our quads have smaller stretch and are more orthogonal.

Comparison with [Diamanti et al. 2015]. The integrable Polyvector field method [Diamanti et al. 2015] takes an initial frame field and optimizes it to guarantee its integrability. However, this process only weakly minimizes distortion and is prone to introduce new singularities. In Fig. 11, we again exhibit a lower distortion for a majority of triangles as well as a smaller number of cones – 140 against 70 in this example.

Comparison with Myles and Zorin [2012]. Myles and Zorin [2012] propose a greedy algorithm for cone placement along with a post-processing procedure to obtain k/4 singularity indices. Cones tend to be placed in high curvature regions which can be suboptimal. In general, we achieve lower stretch for fewer singularities (Table 1) as typically represented for the Dilo model in Figs. 11.

Comparison with Fang et al. [2018]. We also compare our results to a method producing quad-meshes without the quantization step. Fang et al. [2018] use periodic functions to integrate a frame field directly into an integer parametrization. This type of methods generally output a high quality mesh on most of the input surface. However, some localized regions need to be repaired which often introduce new singularities.

Figs. 16 and 14 show the histograms of our metrics for the Dilo model. Our quads are more orthogonal and suffer less from stretching.

7 CONCLUSION AND FUTURE WORKS

By constraining a parametrization to a parallel transport instead of a cross field, we constructed an algorithm for global seamless parametrization without prescribing cones in advance. This direct link between singularities and the parametrization enables us to find cone points whose position minimize a given distortion energy. Unlike previous methods our solver is not greedy and does not rely on integer variables.

In the perspective of quad meshing, the main limitation of our work is that it does not handle integer coordinates of cone points. We still rely on a post-processing quantization step which impacts the overall distortion distribution. So the cone placement is optimal for the seamless parametrization but suboptimal for the final quadmesh.

A second limitation is that our current implementation is slower than previous methods because of the a higher number of variables – $2|F| + 4|E|$ for vertex charts coordinates, $|E|$ for the parallel transport and $2|V|$ for the cross field. In the future, we would like to remove the cross field in the parallel transport constraints, to reduce the number of unknowns.

We believe that this contribution will have a great impact on future works because the parallel transport integrability constraints (Sec. 3) can be rewritten for higher dimension meshes. This opens the door to computing valid seamless parametrization for tetrahedral meshes which is a currently a major challenge for hexmeshing.

REFERENCES

A PROOF OF THEOREM 3.1

Proof. The parametrization resulting of Alg. 1 can be equivalently obtained by (1) synchronizing charts along a spanning tree on vertices, (2) triangulating quads by adding an edge between triangle centers and triangle vertices and (3) isometrically moving pieces of triangles across the cut so that only singular triangles are split and the cut follows primal edges. Therefore, the quad parametrization is equivalent to the parametrization of Alg. 1 in the sense that they share the same singularity index around loops. For this reason, we will carry out the proof on the quad parametrization which is simpler to relate to transport rotations.

Necessary condition. Suppose that there exists charts satisfying Eqs. (E), we will show that Alg. 1 constructs a parametrization compatible with the holonomy of \( \omega \).

Since the quad parametrization is built by recursively aligning charts along a (primal) spanning tree, a loop of vertices is obtained by considering an unvisited edge \( i_0i_N \) and walking back to the tree root from both side of the edge. Let \( y = (i_0, \ldots, i_N, i_0) \) be such a loop. We denote \( r_i \in \mathbb{C} \) the total rotation of chart \( C_i \) during the synchronization. Let \( u_i \in \mathbb{C}, v_i \in \mathbb{C} \) be duplicated edge vectors of charts \( C_{i_0}, C_{i_1} \) and \( u = r_{i_0}u_{i_0}, v = r_{i_1}v_{i_1} \), their coordinates in the quad parametrization. Applying rotations recursively along the loop leads to:

\[
\begin{align*}
    u &= r_{i_0}u_{i_0} \\
    &= r_{i_0}\exp(\omega(i_0,i_1))v_{i_0} \\
    &= r_{i_N}\prod_{k=0}^{N-1}\exp(\omega(i_k,i_{k+1}))v_{i_k} \\
    &= r_{i_N}\prod_{k=0}^{N-1}\exp(\omega(i_k,i_{k+1}))\exp(\omega(i_N,i_0))v_{i_N} \\
    &= \Phi_{i_0}(y)v.
\end{align*}
\]

Therefore, duplicated edges, which include edge at the quad parametrization cut, are equal up to the total rotation along the loop. If \( \Phi_{i_0}(y) \) is a rotation by an angle \( k\pi/2, k \in \mathbb{Z} \) for all loops then the parametrization is seamless.

Sufficient condition. Suppose that we are given a parametrization whose singular vertices are of valency three. Let us subdivide regular triangles into three quad and turn the three triples adjacent to a singularity into three quads. By moving triangle pieces across cuts so that the cut follows dual edges, we recover a quad-parametrization with transition rotations \( r \in \mathbb{C}, |r| = 1 \). The choice of \( \omega \) equal to \( \arg(r) \) on cuts and 0 on regular edges and charts coordinates equal to the parametrization coordinates obviously satisfies Eqs. (E). \( \square \)

B PROOF OF THEOREM 4.1

Proof. Since \( \omega \) is compatible with a frame field (Eqs. (F)), its holonomy on any loops is constrained to be an integer multiple of \( \pi/2 \). Since it also satisfies Eqs. (E), Thm. 3.1 guarantees that the reconstructed parametrization is seamless. Eqs. (D) ensure local injectivity as all triangles \( (P_1, P_2, S_{ijk}) \) composing the parametrization must have positive area.

Eqs. (B) force the feature edge in charts to be aligned with the cross field and corner vertex have \( k\pi/2 \) total inner angle. Since
Fig. 18. Two versions of the same vertex rings: one with an angle defect of 0 and the other with an angle defect of \(2\pi\), both have only positive area triangles.

Eqs. \(E\) are satisfied the feature edges are reconstructed as straight lines with possible corners at vertices. Because \(\omega\) is compatible with a frame field aligned with feature edges, two disconnected sets of feature edges linked by a vertex path \(\gamma\) will be related by a total rotation of \(k\pi/2\). Thus, feature edges are (up to a global rotation) isolines of the parametrization.

**Double coverings.** We are now going to prove that the parametrization singularities are exactly those prescribed by the parallel transport. According to Thm. 3.1 the parametrization transition functions match the parallel transport holonomy. Therefore the parametrization angle defect at vertices is equal to the parallel transport Gaussian curvature modulo \(2\pi\). We will show that this additional parasite angle corresponding to integer singularities cannot appear in our parametrization.

By construction, singular vertices have only three incident edges, thus the angle defect is strictly bounded by \(-\pi\) and \(2\pi\) – otherwise triangles are invalid. So, angle defects of singular triangles are exactly equal to the prescribed Gaussian curvature.

Vertices of the input mesh also have prescribed singularity indices: 0 for inner vertices and user prescribed on feature edges. At these vertices, unwanted negative integer index singularities could appear even if all triangles positive areas (see Fig. 18). However, since \(\omega\) satisfies Eqs. \((E)\), its total Gaussian curvature is constraint by the Gauss-Bonnet theorem. Thus, any additional singularities must have indices summing to 0 to maintain feasibility of the parametrization. As a consequence, any strictly negative index singularities appearing at a vertex must be compensated by positive singularities. As we have shown, these singularities cannot appear inside triangles and vertices only accept negative singularities. Thus, double coverings cannot happen. □

**C PROOF OF PROPOSITION 5.1**

**Proof.** Let us show that if there exists two piece-wise linear functions \(f, g \in \mathbb{R}^{\mid V\mid}\) on a flat triangle mesh such that \(\nabla f = \mathcal{J} \nabla g\) then they must be a rotation and a global scaling of the vertex positions.

By construction of piece-wise linear finite elements, the gradient operator is continuous in the direction tangent to an edge:

\[
\langle \nabla f_{ijk}, e_{ij} \rangle = \langle \nabla f_{ijl}, e_{ij} \rangle.
\]

The function \(g\) satisfies the LSCM equation, so that \(\nabla f\) is also continuous in the direction normal to the edge:

\[
\langle \nabla f_{ijk}, J e_{ij} \rangle = \langle \mathcal{J}^T \nabla f_{ijk}, e_{ij} \rangle \]

\[
= -\langle \nabla g_{ijk}, e_{ij} \rangle
\]

\[
= -\langle \nabla g_{ijl}, e_{ij} \rangle
\]

\[
= \langle \nabla f_{ijl}, J e_{ij} \rangle.
\]

As \(\nabla f\) and \(\nabla g\) are continuous across all edges, they must be constant over the entire mesh. Therefore, there exists \(a, b, c, d \in \mathbb{R}^2\) such that \(f\) and \(g\) are affine functions:

\[
f_i = a^\top \begin{pmatrix} x_i \\ y_i \end{pmatrix} + b,
\]

\[
g_i = c^\top \begin{pmatrix} x_i \\ y_i \end{pmatrix} + d.
\]

Using the LSCM equation again, the vectors \(a\) and \(c\) are such that \(c = Ja\), therefore \(f, g\) are scaled-rotation of the vertex coordinates. □