Periodic Global Parameterization

Nicolas Ray, Wan Chiu Li, Bruno Lévy, Alla Sheffer, Pierre Alliez

Periodic Global Parameterization

Nicolas Ray
Wan Chiu Li
Bruno Lévy
INRIA - Alice
Alla Sheffer
University of British Columbia
Pierre Alliez
INRIA - Geometrica

We present a new globally smooth parameterization method for triangulated surfaces of arbitrary topology. Given two orthogonal piecewise linear vector fields defined over the input mesh (typically the estimated principal curvature directions), our method computes two piecewise linear periodic functions, aligned with the input vector fields, by minimizing an objective function. The bivariate function they define is a smooth parameterization almost everywhere on the surface, except in the vicinity of singular vertices, edges and triangles, where the derivatives of the parameterization vanish. We extract a quadrilateral chart layout from the parameterization function and propose an automatic procedure to detect the singularities, and fix them by splitting and re-parameterizing the containing charts. Our method can construct both quasi-conformal (angle preserving) and quasi-isometric (angle and area preserving) parameterizations. The more restrictive class of quasi-isometric parameterizations is constructed at the expense of introducing more singularities. The constructed parameterizations can be used for a variety of geometry processing applications. Since we can align the parameterization with the principal curvature directions, our result is particularly suitable for surface fitting and remeshing.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Color, shading, shadowing, and texture*; I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling; G.1.6 [**Numerical Analysis**]: Optimization; J.6 [**Computer Aided Engineering**]:

General Terms: Algorithms

Additional Key Words and Phrases: mesh processing, parameterization, conformality.

Author's address: N. Ray, W.C. Li, B. Lévy, Project ALICE, INRIA, 545000 Vandoeuvre, France,
e-mail: {ray | wcli | levy }@loria.fr
Author's address: A. Sheffer, University of British Columbia, Vancouver, BC, V6T 1Z4, Canada,
e-mail: sheffa@cs.ubc.ca
Author's address: P. Alliez, Project GEOMETRICA, INRIA, 06902 Sophia-Antipolis, France,
e-mail: pierre.alliez@sophia.inria.fr

1. INTRODUCTION

A parameterization defines a correspondence between a surface mesh embedded in 3D and a simple 2D domain, referred to as the *parameter space*. In the general case, a parameterization is expected to be bijective, i.e., one-to-one.

Recent advances in geometry processing algorithms and computer graphics hardware have made possible the use of parameterized surface meshes as valid representations for resampling and remeshing, as well as for mapping complex signals such as texture, normal or light maps for efficient rendering purposes.

Common mesh parameterization methods are restricted to domains with simple topology, such as disks [Floater 1997], spheres [Gotsman et al. 2003] or torii [Gortler et al. 2004]. Constructing parameterizations for surfaces of arbitrary topology remains a challenging problem. One common solution is to introduce cuts into the initial surface mesh so as to convert it into one [Gu et al. 2002] or several topological disks [Lévy et al. 2002]. However, these cuts introduce discontinuities which may be visible as mapping artifacts during rendering, especially when mip-mapping is activated. These cuts also introduce artifacts for remeshing, such as artificial alignments of edges along the boundary, and unwanted variations of element sizes. Last but not least, finding appropriate cuts so as to minimize the artifacts listed above is notoriously difficult.

Globally smooth parameterization techniques aim at reducing such discontinuities [Khodakovsky et al. 2003]. As reviewed in the next section, these techniques are limited in their ability to control the parametric distortion and the number and placement of singularities introduced into the parameterization. Many of these also require an a priori segmentation of the mesh into charts, which remains an open problem.

For many mesh processing applications it is advantageous to have a parameterization aligned with the principal curvature directions on the surface. In particular, as explained in [d'Azevedo 2000], for both surface fitting and remeshing, alignment with curvature improves convergence. To the best of our knowledge, no existing parameterization method supports such alignment.

In this paper, we propose a new globally smooth parameterization method which accepts as input a pair of orthogonal vector fields and computes a parameterization aligned with these fields. Given the principal curvature directions as input, our method provides parameterizations particularly suitable for approximation applications such as remeshing and surface fitting. The proposed method does not require prior partition into charts nor any cutting. Instead, the chart layout (i.e., the topology of the base complex) and the parameterization emerge simultaneously from a global numerical optimization process. In contrast to previous approaches, the charts generated by our method are mostly well shaped quadrilaterals with regular, valence-four, connectivity. The rest of this section reviews the previous work and gives an overview of our approach.

1.1 Previous Work

We focus our review on globally smooth parameterization methods, a review of the many other available mesh parameterization techniques being beyond the scope of this paper. The reader is referred to [Floater and Hormann 2004] for a complete survey.

To construct a globally smooth parameterization, existing methods use two different strategies. One class of methods first partitions the object into a set of charts, parameterizes each chart independently, and applies a post-relaxation procedure to blur the discontinuities

along chart boundaries. The other class of methods directly takes the topology of the surface into account and uses a global formulation to obtain the parameterization.

Inter-chart relaxation:. In [Khodakovsky et al. 2003], transition functions are introduced to define a relaxation procedure that optimizes inter-chart continuity. This relaxation is applied simultaneously to all charts.

A similar approach is used in *polycube maps* [Tarini et al. 2004]. First, a quadrilateral chart layout is manually constructed by the user. Then, the charts are parameterized using a globally smooth version of the MIPS method [Hormann and Greiner 2000]. The parameterization constructed by our method shares some similarities with a polycube map, with the major difference that in our case, the quadrilateral chart layout is constructed *automatically*.

In [Kraevoy and Sheffer 2004; Schreiner et al. 2004] a parameterization between pairs of input models is computed. In both papers a triangular chart layout and a corresponding base-mesh are constructed automatically, and the parameterization is smoothed either onto the base-mesh [Kraevoy and Sheffer 2004] or onto the models themselves [Schreiner et al. 2004]. Note that a triangular chart layout is not suitable for quadrilateral remeshing and is far less suitable for spline fitting.

Global approaches:. In the context of texture synthesis, the *lapped textures* approach [Praun et al. 2000] covers an object with overlapping tiles. Wei and Levoy [2001] use two orthogonal vector fields to control a texture synthesis process. The *jump maps* technique [Zelinka and Garland 2003] also uses local parameterizations aligned to a vector field. In the last two methods, texture coordinates are generated by a greedy procedure.

For parameterization purposes, and in contrast to texture synthesis, we wish to avoid any overlap or gap in-between the charts. Our approach ensures continuity of the mappings by minimizing a global energy functional.

Gu and Yau [2003] propose to construct the so-called *conformal structure* of a surface S. A set of mutually compatible local parameterizations is extracted from this structure. The continuity is achieved everywhere except at a number of *singular points*. Intuitively, and given a sphere with its common parameterization, those singular points correspond to its two poles. Gu and Yau [2004] find the unique locations of the singular points that satisfy conformality. They also solve for the optimal conformal transformation that minimizes global stretch. Since the number of singular points remains constant, the resulting parameterizations often still exhibit significant stretch.

The theoretical background for seamless parameterizations is studied in [Gortler et al. 2004], based on the elegant formalism of *one-forms*. From the properties of these one-forms, the authors develop a discrete counterpart of the Hopf-Poincaré index theorem. This theorem generalizes the notion of singular points mentioned above and allows their splitting and merging. It states that for a surface of genus g, the indices of all singular points sum to $2-2g$, where the index of a singular point corresponds to its multiplicity.

Our method constructs a class of globally continuous overlapping local parameterizations along with continuity conditions. The main difference is that, motivated by geometry processing applications, we incorporate more geometric information into the problem setting. We optimize an energy functional that both minimizes the mapping distortion and optimizes the alignment of the iso-parametric curves with two orthogonal vector fields.

1.2 Algorithm Overview

The input to our algorithm is a triangle surface mesh, together with two orthogonal vector fields $\vec{K}$ and $\vec{K^{\perp}}$. The vector fields are typically the estimated principal directions of curvature, but any pair of user-defined vector fields are also valid as long as they match the orthogonality constraint.

Our goal is to construct a globally smooth parameterization aligned with the input vector fields. More formally, the gradients (iso-parametric curves) of the parameterization optimized by our algorithm will be as tangential as possible to the input vector fields. If the vector fields are the directions of principal curvature, the iso-parametric curves will then be automatically aligned to the significant geometric features of the shape (fillets, axes of symmetry, etc.). The principal curvature directions can be estimated by a variety of techniques, such as [Cohen-Steiner and Morvan 2003].

One of the main achievements of our algorithm is its ability to automatically extract a quadrilateral chart layout for the global parameterization. The size (side length) of the charts is determined by a user prescribed parameter ω. As demonstrated in Figure 1, the extracted charts are mostly well shaped and have uniform sizing. They also exhibit a highly regular connectivity, with mostly valence four vertices.

Our method can construct a curvature-adapted globally smooth conformal parameterization or a quasi-isometric parameterization. In the latter case, the near zero distortion is obtained at the expense of introducing more singular points. This trade off is achieved by an additional processing step which controls the curl of the vector field.

The algorithm for computing our global parameterizations consists of four stages:

—**Vector field smoothing** (optional): This pre-processing step smoothes the input vector fields and extrapolates them into mesh areas where the associated directions are ill-defined. When using principal curvature directions as input those areas correspond to isotropic areas which exhibit an infinite number of principal directions. (Figure 1-B and Section 4.2). Since this procedure is stand-alone it can be used as-is for other applications that utilize curvature vector fields on meshes, such as curvature aligned quad-dominant remeshing ([Alliez et al. 2003] and [Marinov and Kobbelt 2004]).

—**Curl-correction** (optional): A global isometric parameterization is usually not possible without a large number of singular points. As explained in Section 4.1, in our setting most of these points correspond to areas where the input vector fields are not curl-free. Hence, to reduce the number of singular points we introduce an optional procedure which rescales the vector fields so as to reduce their curl. The rescaled vector fields are used as input to the subsequent parameterization step. Notice that if curl-correction is applied, the resulting parameterizations will remain conformal and exhibit much fewer singular points, but will usually exhibit larger stretch (Figure 7).
Since both vector field smoothing and curl correction are optional pre-processing stages, we prefer to describe them in the second part of the paper.

—**Parameterization using alternative variables**: This is the main step of our algorithm. To explicitly account for translational and rotational degrees of freedom in the parameterization formulation, we develop an energy functional using alternative variables which are trigonometric functions of the actual parameterization. The derivation of the functional is described in detail in Section 2. The derived energy functional (Equation 15) is minimized using the numerical procedure described in Section 2.5.

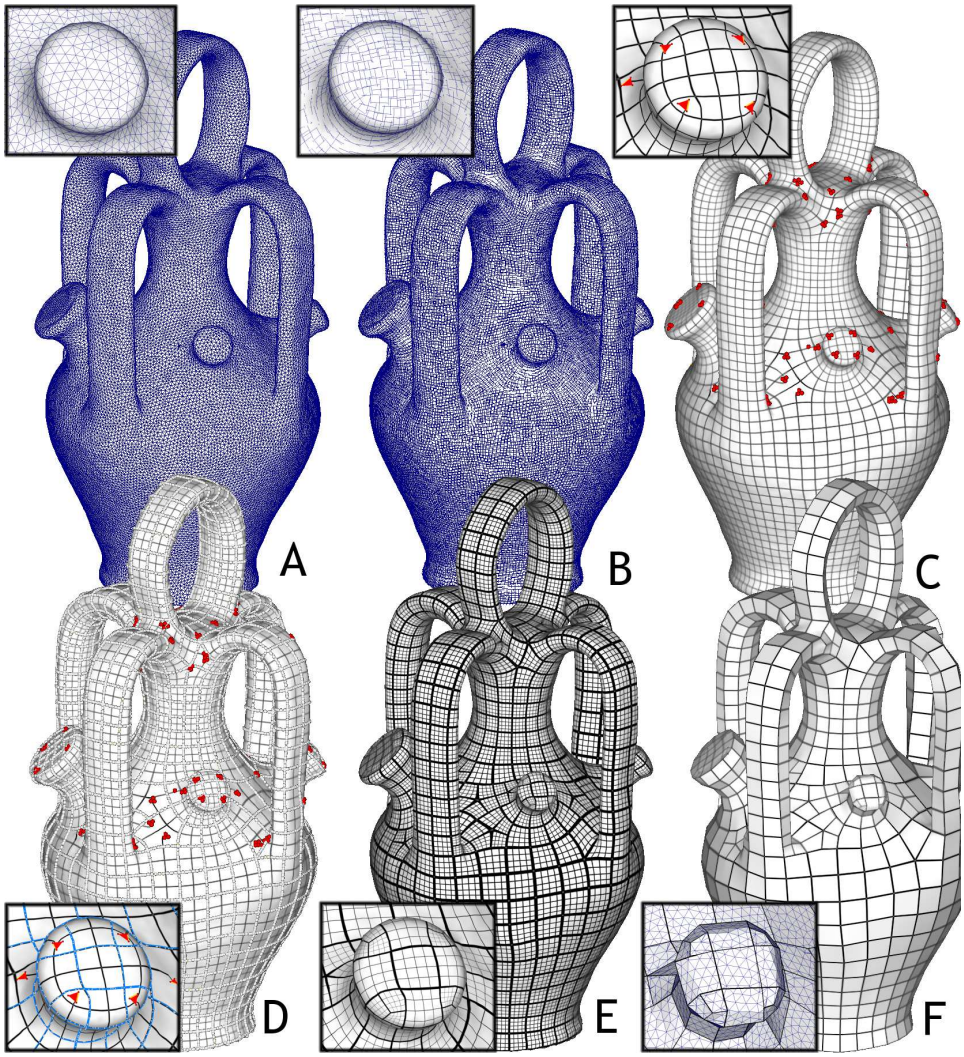


Fig. 1. Algorithm overview. A: input mesh model; B: smoothed curvature directions; C: iso-$k\pi$ θ and ϕ curves. The singular vertices, edges and triangles are highlighted; D: chart layout (extracted from the iso-$2k\pi$ curves); E: final result, obtained after fixing the charts with singularities; F: the resulting base complex.

—**Extraction of chart layout and chart parameterization**: The final stage of the algorithm computes the actual surface parameterization given the solution in terms of alternative variables (Section 3):

—First, the algorithm extracts the parameterizations for each individual mesh triangle. Figure 1-C depicts the iso-$k\pi$ curves of these parameterizations.

—Second, the method detects the singularities present in the computed parameterizations. These are vertices, edges and triangles that do not satisfy the requirements of a valid 2D planar triangulation (highlighted in Figure 1-C,D).

—Third, the per-triangle parameterizations are used to define a global chart layout.

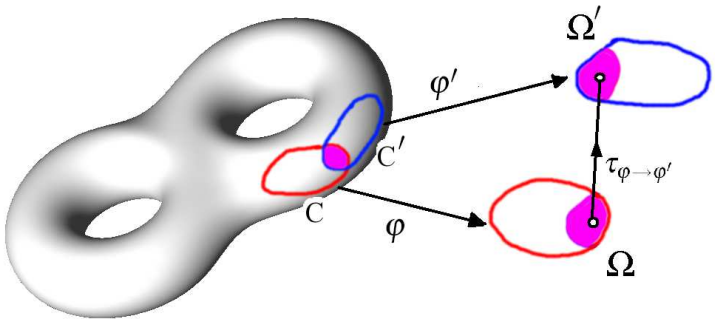


Fig. 2. A global parameterization (or manifold) is a set of overlapping parameterizations $(\varphi, \varphi', \ldots)$ connected by transition functions $(\tau_{\varphi\rightarrow\varphi'} \ldots)$.

—Finally the algorithm computes the per-chart parameterizations. If a chart does not contain any singularity, it reconstructs its parameterization by assembling the individual triangles in parameter space. Otherwise, the chart is split and re-parameterized moving the singularities to chart boundaries. The final parameterization together with the corresponding base complex are shown in Figure 1-C and D respectively.

The result of the procedure is a global conformal parameterization, continuous almost everywhere. The parameterization is aligned with the input vector fields. If no curl-correction is performed, the parameterization is even shown to be quasi-isometric.

Our parameterization technique can be used for a variety of mesh processing applications. In Section 5, we demonstrate its use for curvature-aligned quad-dominant remeshing. Given our globally smooth curvature-aligned parameterization, the mesh generation procedure is elegant and straightforward. Other applications of our method are smooth surface reconstruction and texture mapping.

2. PERIODIC GLOBAL PARAMETERIZATION

2.1 Definition

We first give the definition of a *manifold* (also called a global parameterization in our context). This notion allows us to define a globally smooth parameterization of a surface with arbitrary genus, by combining multiple parameterizations of charts extracted from the surface and linked by transition functions. To our knowledge, the notion of manifold was first used for geometric modeling by Grimm and Hugues [Grimm and Hugues 1995]. More recently, a C^∞ class of surfaces based on manifolds was proposed in [Ying and Zorin 2004].

Given a surface $\mathbf{S}$, we consider a set of (possibly overlapping) topological disks $\{\mathbf{C}\}$ called *charts*, and a set of functions $\{\varphi\}$ mapping each chart $\mathbf{C}$ to a 2D domain Ω (see Figure 2). The coordinates in 2D space will be denoted by θ, ϕ in what follows[1]. The set of functions $\{\varphi\}$ is called a global parameterization (or a manifold) if it satisfies the following condition:

[1]We use θ and ϕ rather than the accustomed u, v or s, t to indicate the periodic nature of these coordinates.

Given two charts $\mathbf{C}$ *and* $\mathbf{C}'$*, if their intersection* $\mathbf{C} \cap \mathbf{C}'$ *is a topological disk, then the images of the intersection* $\mathbf{C} \cap \mathbf{C}'$ *in parameter space through* φ *and* φ' *are linked by a simple geometric transform* $\tau_{\varphi \to \varphi'}$*:*

$$\forall \mathbf{p} \in \mathbf{C} \cap \mathbf{C}', \quad \varphi'(\mathbf{p}) = \tau_{\varphi \to \varphi'}(\varphi(\mathbf{p}))$$

The $\tau_{\varphi \to \varphi'}$ functions are called *transition functions*. (see [Khodakovsky et al. 2003]). Manifolds are called *affine* if all the transition functions are translations. *Complex* manifolds admit a more general class of holomorphic transition functions, including similarities (i.e. rotation + translation + scaling), see [Weisstein] for a definition of these classes of objects. Whereas previous work focus on affine manifolds [Gu and Yau 2003; Gortler et al. 2004], our method constructs a sub-class of complex manifolds, allowing for both translational and rotational degrees of freedom in the transition functions. The extra degree of freedom allows for greater flexibility when aligning the parameterization with the input vector fields, as shown below.

Our goal is to construct a global parameterization such that the gradients $\nabla\theta$, $\nabla\phi$ of the parameter-space coordinates θ, ϕ are aligned with two prescribed vector fields (for instance, the principal directions of curvature). We first start with the simplest possible charts, i.e. the triangles. In our initial setting, the global parameterization is defined by the coordinates θ_i^T, ϕ_i^T at the corners of the triangles, where the global index i denotes a vertex, and T denotes a triangle. Using the so-defined manifold structure, it is possible to derive a parameterization of more general charts, by assembling the triangles in parameter space, as explained later in Section 3.3.

We first consider the case of an affine manifold (i.e. the transition functions $\tau_{\varphi \to \varphi'}$ are translations). We will then show how to introduce the rotational degree of freedom. Given two triangles $T = (i, j, k)$ and $T' = (k, j, l)$ sharing the edge (j, k), their parameter-space coordinates (θ, ϕ) define an affine manifold if:

$$\begin{pmatrix} \theta_j^T \\ \phi_j^T \end{pmatrix} - \begin{pmatrix} \theta_j^{T'} \\ \phi_j^{T'} \end{pmatrix} = \begin{pmatrix} \theta_k^T \\ \phi_k^T \end{pmatrix} - \begin{pmatrix} \theta_k^{T'} \\ \phi_k^{T'} \end{pmatrix} \tag{1}$$

We now need to derive an energy functional F, depending on all the (θ_i^T, ϕ_i^T) coordinates and characterizing the alignment of the gradients $(\nabla\theta, \nabla\phi)$ to the principal directions of curvatures. In our formulation of the energy functional F, instead of expressing Equation 1 as a constraint, we replace the (θ_i^T, ϕ_i^T) variables with alternative variables, associated to the vertices (rather than to the corners of the triangles), and naturally satisfying the constraints. We will then show how to retrieve the (θ_i^T, ϕ_i^T)'s from those alternative variables.

We introduce an additional restriction on the transition functions $\tau_{\varphi \to \varphi'}$: the coordinates of the translation vectors connecting two charts should be integer multiples of 2π. With this additional constraint, we have:

$$\begin{pmatrix} \cos\theta_j^T \\ \sin\theta_j^T \end{pmatrix} = \begin{pmatrix} \cos(\theta_j^{T'} + 2s\pi) \\ \sin(\theta_j^{T'} + 2s\pi) \end{pmatrix} = \begin{pmatrix} \cos\theta_j^{T'} \\ \sin\theta_j^{T'} \end{pmatrix} \quad ; \quad \begin{pmatrix} \cos\phi_j^T \\ \sin\phi_j^T \end{pmatrix} = \begin{pmatrix} \cos\phi_j^{T'} \\ \sin\phi_j^{T'} \end{pmatrix} \tag{2}$$

(this condition is also satisfied at vertex k). As a consequence, and given a vertex i, for all the triangles T incident to i, the values of $\cos\theta_i^T$ and $\sin\theta_i^T$ (resp. ϕ) coincide. We

introduce the variables $U_i = (\cos\theta_i^T, \sin\theta_i^T)$ and $V_i = (\cos\phi_i^T, \sin\phi_i^T)$, which no-longer depend on T and are attached to the vertices instead.

We now consider the more general case of a sub-class of complex manifolds where transition functions $\tau_{\varphi\to\varphi'}$ can be combinations of translation *and* rotation. The coordinates of the translations are constrained to be multiples of 2π as before, and the rotation angles are constrained to be multiples of $\pi/2$. We will refer to this configuration as a periodic global parameterization. In this setting, the compatibility condition connecting two triangles (Equation 2) is replaced with:

$$\begin{array}{l}\exists r \in \{0,1,2,3\} \\ \begin{pmatrix}\cos\theta_j^T \\ \sin\theta_j^T\end{pmatrix} = \begin{pmatrix}0 & -1 \\ 1 & 0\end{pmatrix}^r \begin{pmatrix}\cos\theta_j^{T'} \\ \sin\theta_j^{T'}\end{pmatrix} \quad ; \quad \begin{pmatrix}\cos\phi_j^T \\ \sin\phi_j^T\end{pmatrix} = \begin{pmatrix}0 & -1 \\ 1 & 0\end{pmatrix}^r \begin{pmatrix}\cos\phi_j^{T'} \\ \sin\phi_j^{T'}\end{pmatrix}\end{array} \tag{3}$$

As a consequence, given a vertex i, for all the triangles T incident to i, the values of $\cos\theta_i^T$ and $\sin\theta_i^T$ (resp. ϕ) coincide up to a change of sign and a swapping of the sine and cosine.

We now show how to express the alignment with the input vector fields in terms of the variables (U_i, V_i) (Sections 2.2,2.3,2.4) and explain how to retrieve the parameter-space coordinates (θ_i^T, ϕ_i^T) from these variables (Section 3.1). We will then proceed to extract the chart layout (Section 3.3), and show how to retrieve a parameterization of the charts from the per-triangle coordinates (θ_i^T, ϕ_i^T) (Section 3.4).

2.2 Parameterization Alignment

As described in Section 1.2 the input to our algorithm consists of two orthogonal *control* vector fields $\vec{K}$ and $\vec{K}^\perp$ defined on a surface S, and a chart size parameter ω. The control vector fields are defined at the vertices of the surface mesh and are linear across the triangles. The meaning of this parameter ω and the way to choose it are explained below. Our method aims at constructing a complex manifold $\{\varphi^T\} = \{(\theta^T, \phi^T)\}$ such that each function φ^T associated to a triangle T satisfies:

$$\nabla\theta^T = \omega\vec{K} \quad ; \quad \nabla\phi^T = \omega\vec{K}^\perp \tag{4}$$

In addition, the complex manifold should be a periodic global parameterization, i.e. the transition functions $\tau_{T\to T'}$ should be solely composed of translations multiples of 2π and rotations multiples of $\pi/2$.

The gradient of the parameterization should not depend on the magnitude of the input vector fields. When our goal is to construct a parameterization *as isometric as possible*, we normalize the control vector fields $\|\vec{K}\| = \|\vec{K}^\perp\| = 1$. If we want to reduce the curl of the vector fields and hence minimize the number of singularities in the parameterization, the vector fields are first normalized and then scaled as described in Section 4.1. This leads to a parameterization which is no longer isometric, but which remains conformal.

Due to this normalization, the parameter ω controls the period of the θ and ϕ functions. As described in Section 3.3, we will use the 2π periods of the parameterization to define the chart layout. Hence, ω will determine the size of the charts. In all our examples, we set ω to ten times the average edge length in the input mesh. Note that if ω is too large, charts that are not homeomorphic to disks may be generated. This can be easily detected by computing the Euler-Poincaré characteristic of the charts, and ω can be automatically decreased if such a configuration is detected.

Since $curl(\nabla\rho) = 0$ for any scalar field ρ, a solution to Equation 4 exists only if $curl(\vec{K}) = curl(\vec{K}^{\perp}) = 0$ [Needham 1994]. In general, the input vector fields might have non-zero curl. Hence, we have to restate our goal in weaker terms by minimizing the following energy functional:

$$F = \int_S \left(\|\nabla\theta^T - \omega\vec{K}\|^2 + \|\nabla\phi^T - \omega\vec{K}^{\perp}\|^2 \right) dS \quad (5)$$

Given this problem setting, the main difficulty is to express the alignment of the parameterization gradients with the control vector fields independently from the translational and rotational degrees of freedom. We first introduce translation-invariance into the formulation in Section 2.3, and then refine the formulation to introduce the rotational degree of freedom (Section 2.4).

2.3 Translation-invariant Energy Functional

The main challenge in the formulation given by Equation 5 is to find a way to solve for a *periodic* function. As explained in Section 2.1, to support translational invariance in parameter space, we propose to use the 2π periodicity of the sine and cosine functions. As shown below, it is possible to restate the alignment with the control vector fields in terms of the sines and cosines $U = (\cos\theta, \sin\theta)$ and $V = (\cos\phi, \sin\phi)$ of the parameters θ and ϕ. The U and V's will be the unknowns of our problem. Thus, we will obtain a periodic definition of the minimizer of the energy functional F.

Following the proof in Appendix A we can minimize the following function F^* instead of F, as it admits the same minimizer:

$$F^* = \sum_T \int_T \left(\|\nabla\theta^T - \omega\vec{K}_T\|^2 + \|\nabla\phi^T - \omega\vec{K}_T^{\perp}\|^2 \right) ds \quad (6)$$

where $\vec{K}_T$ and $\vec{K}_T^{\perp}$ denote the average value of $\vec{K}$ (resp. $\vec{K}^{\perp}$) across the triangle. Since $\nabla\theta^T$ and $\nabla\phi^T$ are constant across each triangle, we have

$$F^* = \sum_T \left(\|\nabla\theta^T - \omega\vec{K}_T\|^2 + \|\nabla\phi^T - \omega\vec{K}_T^{\perp}\|^2 \right) A_T \quad (7)$$

where A_T is the area of triangle T. We now consider a single entry in this sum:

$$F_T = \left(\|\nabla\theta^T - \omega\vec{K}_T\|^2 + \|\nabla\phi^T - \omega\vec{K}_T^{\perp}\|^2 \right) A_T \quad (8)$$

Since it is difficult to introduce translational invariance directly into F_T, we will first study $F_{T,i}^{\theta}$, the energy along the edges $\vec{e}_i$ of T (Figure 3) with respect to θ. The energy $F_{T,i}^{\phi}$ with respect to ϕ is derived in a similar manner. We will then express F_T as a linear combination of the $F_{T,i}^{\theta}$'s and $F_{T,i}^{\phi}$'s.

Intuitively, when considering the difference along the edge $\vec{e}_i$, we need to consider the difference between the projection of $\vec{K}$ to the edge and the gradient $\nabla\theta$ along the edge. The projection is $\vec{K} \cdot \vec{e}_i / \|\vec{e}_i\|$, and the gradient along the edge is $(\theta_{i\oplus 2} - \theta_{i\oplus 1})/\|\vec{e}_i\|$ where $i \in \{1,2,3\}$ denotes a local index in T (see Figure 3) and $\oplus$ denotes addition modulo 3.

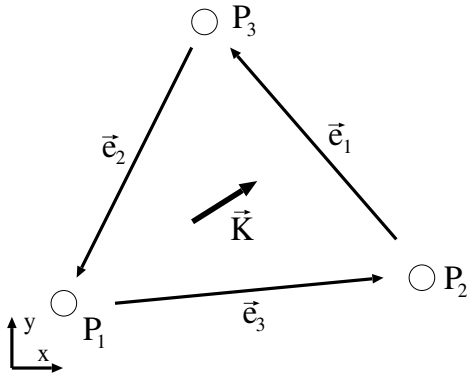


Fig. 3. Triangle notations.

We define the energy along the edge as

$$F^{\theta}_{e_i} = \int_{\vec{e}_i} \left(\theta_{i\oplus 2} - \theta_{i\oplus 1} - \vec{K}\cdot\vec{e}_i\right)^2 / \|\vec{e}_i\|^2 ds \tag{9}$$

With a derivation similar to the one given in Appendix A, since $\vec{K}$ and $\vec{K^{\perp}}$ are linear along the edges, we can replace $\vec{K}$ with $\vec{K}_i = (\vec{K}_{i\oplus 2} + \vec{K}_{i\oplus 1})/2$ and scale the minimized function by $\|\vec{e}_i\|$ without changing the minimizer. The new energy functional that we will minimize per edge is

$$F^{\theta}_{T,i} = \left((\theta_{i\oplus 2} - \theta_{i\oplus 1}) - \omega\vec{K}_i\cdot\vec{e}_i\right)^2 \tag{10}$$

Using this energy formulation, it is now easy to introduce the translational invariance, replacing the difference by a difference modulo translation by 2π:

$$F^{\theta}_{T,i} = \min_s \left\{((2s\pi + \theta_{i\oplus 2} - \theta_{i\oplus 1}) - \omega\vec{K}_i\cdot\vec{e}_i)^2\right\} \tag{11}$$

By approximating this difference by the norm of the difference of the sine and cosine vectors, corresponding to order 1 Taylor expansion, we obtain:

$$F^{\theta}_{T,i} \simeq \left\| U_{i\oplus 2} - \begin{pmatrix} \cos(\omega\vec{K}_i\cdot\vec{e}_i) & -\sin(\omega\vec{K}_i\cdot\vec{e}_i) \\ \sin(\omega\vec{K}_i\cdot\vec{e}_i) & \cos(\omega\vec{K}_i\cdot\vec{e}_i) \end{pmatrix} U_{i\oplus 1} \right\|^2 \tag{12}$$

where:

$$U_i = (\cos\theta_i, \sin\theta_i)$$

Note that using this formulation, we no longer depend on the translational coefficient s (Equation 11).

Note that theoretically at this point, we can simply minimize $\sum_{T,i\in 1,2,3}(F^{\theta}_{T,i} + F^{\phi}_{T,i})$, which corresponds to a discrete, edge-based version of the energy. The resulting method works well for regularly sampled surfaces but is sensitive to anisotropic samplings. For this reason, we propose in Appendix B a variant of our objective functional which integrates the energy over the triangles. In our experiments this greatly improved the results without adding too much computation overheads.

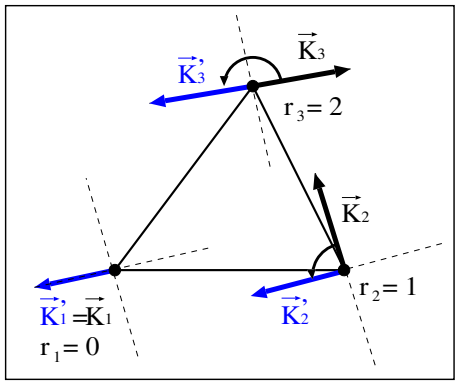
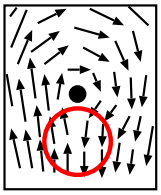


Fig. 4. Locally re-orienting the control vector field.

Our current formulation for F^* supports translational invariance. We now proceed to introducing rotational invariance into the formulation.

2.4 Rotation-invariant Energy Functional



In general, it is not possible to *globally* orient a vector field in a consistent way (see the circled region). To alleviate this issue, we modify the formulation of the triangle energy (Equation 8) by *locally* reorienting the vector field in the new formulation (Figure 4). The orientations of the vectors $\vec{K}_1, \vec{K}_2$ and $\vec{K}_3$ at the respective vertices of the triangle (Figure 4) are now allowed to vary by multiples of $\pi/2$. Thus, $\vec{K}_2$ (resp. $\vec{K}_3$) is aligned with $\vec{K}_1$ by applying r_2 rotations of $\pi/2$ (resp. r_3).

The rotation is applied simultaneously to the control vector fields $(\vec{K}_i, \vec{K}_i^{\perp})$ and to the unknowns $(\theta_i \quad \phi_i)$. Note that an odd difference of r_i along an edge means swapping the unknowns (i.e., connecting θ's with ϕ's).

To define the objective function F_T on the triangles, we use the same approach as in previous Section. We first express the deviation $F_{T,i}$ along an edge, then express F_T as a linear combination of the $F_{T,i}$'s as defined by Equation 24. Since the θ's and the ϕ's may be coupled, we can no longer separate them.

Adding rotational invariance, Equation 11 becomes

$$F_{T,i} = \min_{s,t} \left\| \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}^{r_{i\oplus 2}} \begin{pmatrix} \theta_{i\oplus 2} \\ \phi_{i\oplus 2} \end{pmatrix} - \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}^{r_{i\oplus 1}} \begin{pmatrix} \theta_{i\oplus 1} + 2s\pi \\ \phi_{i\oplus 1} + 2t\pi \end{pmatrix} - \begin{pmatrix} \delta_i \\ \delta_i^{\perp} \end{pmatrix} \right\|^2$$

where:

$$r_i = \underset{r \in \{0,1,2,3\}}{\operatorname{argmax}} \left(\vec{K}_1 . \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}^{r} \vec{K}_i \right) ; \vec{K}_i{}' = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}^{r_i} \vec{K}_i \tag{13}$$

$$\delta_i = \omega/2 \left(\vec{K}'_{i\oplus 1} + \vec{K}'_{i\oplus 2} \right) \cdot \vec{e}_i \quad ; \quad \delta_i^{\perp} = \omega/2 \left(\vec{K}'^{\perp}_{i\oplus 1} + \vec{K}'^{\perp}_{i\oplus 2} \right) \cdot \vec{e}_i$$

As in previous Section, to take the periodicity of the (θ, ϕ) parameters into account, we

solve for the sines and the cosines of these parameters. $F_{T,i}$ as a function of the sines and cosines (using the same order 1 approximation as in Equation 12) is then given by:

$$F_{T,i} \simeq \left\| M^{r_{i\oplus 2}} X_{i\oplus 2} - \begin{pmatrix} \cos\delta_i & -\sin\delta_i & 0 & 0 \\ \sin\delta_i & \cos\delta_i & 0 & 0 \\ 0 & 0 & \cos\delta_i^{\perp} & -\sin\delta_i^{\perp} \\ 0 & 0 & \sin\delta_i^{\perp} & \cos\delta_i^{\perp} \end{pmatrix} M^{r_{i\oplus 1}} X_{i\oplus 1} \right\|^2 \tag{14}$$

$$\text{where:} \quad M = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad ; \quad X_i = \begin{pmatrix} U_i \\ \hline V_i \end{pmatrix} = \begin{pmatrix} \cos\theta_i \\ \sin\theta_i \\ \cos\phi_i \\ \sin\phi_i \end{pmatrix}$$

As in the previous section, we can simply minimize $F^* = \sum_{T,i\in 1,2,3}(F^{\theta}_{T,i} + F^{\phi}_{T,i})$, which corresponds to a discrete, edge-based version of the energy, or we can plug this expression into the triangle energy formulation (Appendix B, Equation 24). We now have

$$F^* = \sum_T F_T = \sum_T \sum_{i=1}^{3} \lambda_i^T (F^{\theta}_{T,i} + F^{\phi}_{T,i}) \tag{15}$$

where λ_i^T are given by Equation 24 and $F^{\theta}_{T,i}, F^{\phi}_{T,i}$ are given by Equation 14. The minimizer of F^* is computed as described next.

2.5 Numerical Solution Mechanism

To obtain the minimizer of F^*, we lock one of the vertices $U_1 = (1,0), V_1 = (1,0)$ and minimize F^* with respect to all the other variables. Since F^* is a quadratic form, this means solving a sparse symmetric system. We use the conjugate gradient algorithm with Jacobi's preconditioner. For models with more than 50K vertices, the norms of the U_i, V_i's quickly decrease when we move far away from the locked vertex, resulting in both weighting biases and numerical instabilities. To stabilize the system, we add a (non-linear) penalty term, preventing the norms of the U, V's from decreasing:

$$F^{**} = F^* + \varepsilon \sum_i \left((\|U_i\|^2 - 1)^2 + (\|V_i\|^2 - 1)^2 \right)$$

This augmented energy functional is minimized using Newton's algorithm. In our tests, $\varepsilon = 10^{-3}$ gives good results. We initialize the Newton iteration with the solution of the linear formulation, and convergence, to $\nabla F^{**} < 10^{-6}$, is reached after no more than 5 outer-loop iterations for all the models shown in this paper.

As shown in Figure 5, the penalty term has an interesting property. The singular points correspond to vectors with zero norm. Since two singular points located in the same region drastically increase the penalty function in that region, the augmented energy functional attempts to avoid those configurations and evenly distributes the singular points over the surface.

3. PARAMETERIZATION EXTRACTION

The output of the solution mechanism described in previous Section is a set of U_i, V_i variables. These variables correspond to the sines and cosines of the unknown θ_i, ϕ_i coordinates

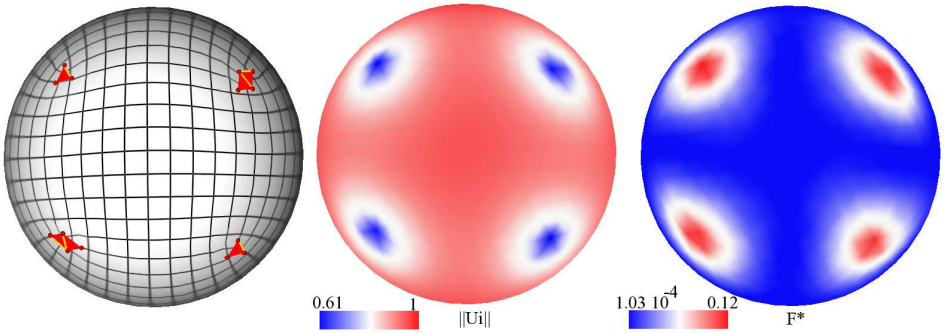


Fig. 5. Plots of the norm $\|U_i\|$ and the energy functional F^*. As can be seen, singularities are characterized by both a small norm of the U_i's (resp. V_i's) and a high energy F^*.

that define the global parameterization. To construct a global parameterization from those U_i, V_i variables, we proceed as follows:

(1) reconstruct a (θ, ϕ) parameterization over the simplest possible charts, i.e. within each individual triangle (Section 3.1),

(2) detect the singular vertices, edges and triangles (Section 3.2),

(3) define the chart layout based on the 2π periods of the per-triangle parameterizations and compute the per-chart parameterizations for charts with no interior singularities (Section 3.3),

(4) split and re-parameterize the charts that contain singularities (Section 3.4).

3.1 Per-triangle Parameterization

Given the U,V variables at the vertices of a triangle $T = (i, j, k)$, finding the θ_i, ϕ_i (resp. j,k) coordinates means determining the integer translational (s_i, t_i) and rotational r_i degrees of freedom. We explicitly determine the values of r,s,t that minimize the edge-energy term (Equation 13) within each triangle as follows.

To define the global position and orientation of the triangle in parameter space, we set the degrees of freedom r_i, s_i, t_i of the first vertex i to $(0,0,0)$. Thus, the θ_i^T, ϕ_i^T coordinates at vertex i are given by $\theta_i^T = \text{angle}(U_i)$ and $\phi_i^T = \text{angle}(V_i)$ where $\text{angle}(U_i) = \text{sign}(U_{i,y})\arccos(U_{i,x}/\|U_i\|)$.

We now assign the coordinates at the two other vertices j and k by determining the differences s_e^T, t_e^T, r_e^T of the translational and rotational degrees of freedom along the edge $e = (i, j)$ (resp. $(i,k),(j,k)$), given by $s_e^T = (s_j^T - s_i^T), t_e^T = (t_j^T - t_i^T)$ and $r_e^T = (r_j^T - r_i^T)$. We first consider the edge (i, j). Given the coordinates (θ_i^T, ϕ_i^T) at vertex i and the control vector field values $(K_i, K_i^\perp)$ and $(K_j, K_j^\perp)$ at the vertices i, j, Algorithm 1 explicitly computes the differences s_e^T, t_e^T, r_e^T, then the values of θ_j^T and ϕ_j^T. The computation for the two other edges is obtained by a circular permutation of indices (i, j, k).

The algorithm first determines whether or not the control vector fields undergo a rotation along the edge. In this case, we sometimes change the correspondence between θ, ϕ and U,V. For instance a rotation of $\pi/2$ corresponds to switching U and V. In this case, θ becomes a function of V and ϕ a function of $-U$. We then determines the difference s_e^T, t_e^T of the translational degree of freedom by explicitly minimizing the edge energy.

Algorithm 1 reconstruction of θ, ϕ along an edge

propagate from i **to** j **along** $e = (i, j)$:
// determine and apply rotation r_e

$$r_e^T \leftarrow \underset{r \in \{0,1,2,3\}}{\operatorname{argmax}_r} \left(\vec{K}_i . \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}^r \vec{K}_j \right)$$

$$\vec{K}_j \leftarrow \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}^{r_e^T} \vec{K}_j \quad ; \quad \vec{K}_j^{\perp} \leftarrow \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}^{r_e^T} \vec{K}_j^{\perp}$$

$$\begin{matrix} \theta_j^T \leftarrow \text{angle}(U_j^T) \\ \phi_j^T \leftarrow \text{angle}(V_j^T) \end{matrix} \quad ; \quad \left(\begin{matrix} \theta_j^T \\ \hline \phi_j^T \end{matrix} \right) \leftarrow \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}^{r_e^T} \left(\begin{matrix} \theta_j^T \\ \hline \phi_j^T \end{matrix} \right)$$

// determine and apply translations s, t
$\vec{n} \leftarrow \vec{e} / \|\vec{e}\|$
$s_e^T \leftarrow \text{argmin}_s \left| \theta_i^T - (\pi/\omega)\vec{n} \cdot (\vec{K}_i + \vec{K}_j) - \theta_j^T + 2s\pi \right|$
$t_e^T \leftarrow \text{argmin}_t \left| \phi_i^T - (\pi/\omega)\vec{n} \cdot (\vec{K}_i^{\perp} + \vec{K}_j^{\perp}) - \phi_j^T + 2t\pi \right|$
$\theta_j^T \leftarrow \theta_j^T + 2s_e^T \pi \quad ; \quad \phi_j^T \leftarrow \phi_j^T + 2t_e^T \pi$

3.2 Characterization of Singular Vertices, Edges and Triangles

Once we have reconstructed the parameterization separately in each triangle, we need to check if these triangles can be assembled in parameter-space in such a way that they form a valid planar triangulation. We already know that the solution of the continuous version of the equation presents singularities where the derivatives of the solution vanish. In our discrete setting, these singularities appear as vertices, edges and triangles that violate the conditions of a valid planar triangulation. These singular vertices, edges and triangles can be characterized as follows (see [Sheffer and de Sturler 2001]):

—**Singular vertices:** a vertex v is singular if the angles at the corners of the triangles incident to v do not sum to 2π. In practice, a singular vertex v can also be characterized by the fact that applying Algorithm 1 to the one-ring neighborhood of v results in an open path.

—**Singular edges:** an edge $e = (i, j)$ is singular if its length in parameter-space mismatches with the one of $e' = (j, i)$.

—**Singular triangles:** a triangle T is singular if applying Algorithm 1 to its three edges results in an open path or if T has a negative area.

We explicitly test for those conditions. Example of all three types of singularities can be seen in Figures 1 and 8.

3.3 Chart Layout

Once we have computed the local parameterization in each triangle T, we construct the chart layout. In our setting, the chart boundaries are defined to be the iso-$2k\pi$ lines of θ

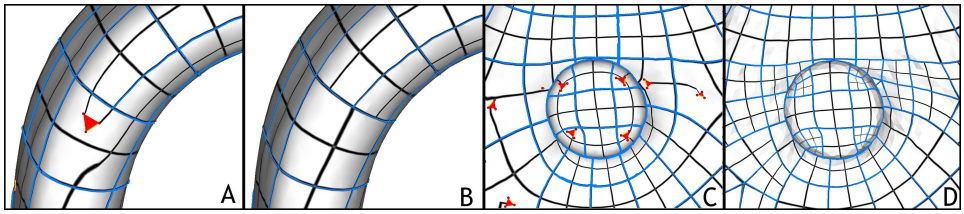


Fig. 6. Re-parameterizing charts with singularities. A,B: charts with four corners are re-parameterized using mean-value coordinates. C,D: N-sided charts are split into quadrilateral charts and those are re-parameterized.

and ϕ. This defines a set of segments in each triangle. We show below that the set of all the iso-$2k\pi$ lines of θ and ϕ is invariant under our transition functions. As a consequence, the end-points of these independent segments match along the non-singular edges of the triangulation, and the segments form continuous polygonal lines.

—**invariance of the set of iso-lines under valid translations :** if a triangle T is traversed by an iso-$2k\pi$ line of θ (resp. ϕ), this triangle translated by $2s\pi$ will be traversed at the same location by the iso-$2(k+s)\pi$ line of θ (resp. ϕ).

—**invariance of the set of iso-lines under valid rotations :** if a triangle T is traversed by an iso-$2k\pi$ line of θ (resp. ϕ), this triangle rotated by $\pi/2$ will be traversed at the same location by the iso-$2(-k)\pi$ line of ϕ (resp. iso-$2k\pi$ line of θ). The same argument applies to rotation by any multiple of $\pi/2$.

Note that given two adjacent mesh triangles T_1 and T_2 sharing a non-singular edge $e=(i,j)$ the transition function $\tau_{T_1\to T_2}$ between the per-triangle parameterizations can be computed as follows:

$$\begin{aligned}\tau_s &= s_i^{T_1} - s_i^{T_2}\\ \tau_t &= t_i^{T_1} - t_i^{T_2}\\ \tau_r &= r_e^{T_1} - r_e^{T_2}\\ \tau(p) &= R^{\tau_r} p + (\tau_s, \tau_t)\end{aligned} \tag{16}$$

where R is rotation by $\pi/2$. Hence the transition functions satisfy the invariance criteria above and therefore the end points of the iso-lines match.

Algorithm 2 computes the chart boundaries. Each triangle stores a list of segments, and each edge stores a list of segment end-points. The algorithm computes the individual segments defined by the intersections of the triangles with the $(\theta = 2k\pi)$ and $(\phi = 2k\pi)$ lines. Both 2D and 3D locations at the end-points of the segments are computed. The algorithm merges the segment end-points along the edges and intersects the segments inside the triangles adding the intersections as new end-points. All the dangling segments are removed (each segment end-point of valence 1 is "nibbled" until an end-point of valence higher than 2 is encountered).

The parameterization of the charts that do not contain any singularity can be retrieved by assembling the triangles in 2D space by a classic greedy algorithm (see e.g., [Sheffer and de Sturler 2001]). For the remaining charts with singularities, we split them and re-parameterize them as shown next. If needed, the chart boundaries can be inserted in the triangulation (e.g., by using our embedded cellular complex data structure [Li et al. 2005]).

Algorithm 2 chart boundaries construction

compute chart boundaries:
for each triangle T
  if T is non-singular
    for $k \in \mathbb{N}$ such that $2k\pi \in [\min_T(\theta), \mathrm{Max}_T(\theta)]$
      Line $l \leftarrow$ line of equation$(\theta = 2k\pi)$
      Segment $S \leftarrow l \cap T$ //*in parameter space*
      store S in T
      store the end-points of S in the corresponding edges of T
    end//*for*
    for $k \in \mathbb{N}$ such that $2k\pi \in [\min_T(\phi), \mathrm{Max}_T(\phi)]$
      Line $l \leftarrow$ line of equation$(\phi = 2k\pi)$
      Segment $S \leftarrow l \cap T$ //*in parameter space*
      store S in T
      store the end-points of S in the corresponding edges of T
    end//*for*
  end//*if*
end//*for*
for each edge e
  merge the segment end-points stored in e that have the same geometric location in 3D
end//*for*
for each triangle T
  compute the intersections between the edges stored in T
end//*for*
recursively remove all dangling segments

3.4 Re-parameterization of Charts with Singularities

In the charts that contain singularities, an unfolding mechanism will not be able to construct a valid planar parameterization, from the per-triangle parameterizations. Note that in most cases it is possible to split the charts along the separatrices of the singularities and locally fix the parameterization in the singular triangles. However, the corresponding algorithm is quite delicate to implement, and might fail under certain configurations of singularities. We opted for a simpler approach:

—Four-sided charts are re-parameterized, using the mean value coordinates method [Floater 2003] (Figure 6 A and B). The parameterization of the boundary vertices is adjusted to preserve C^0 continuity along the chart boundaries.

—N-sided charts are split into quads, by inserting one vertex at the center of the chart and one vertex at the middle of each side of the chart (Figure 6 C and D). The new vertices are connected by geodesics, and the mesh is cut along these geodesics (our implementation uses [Li et al. 2005]). The resulting quadrilateral charts are parameterized as described above. The parameter domains on the created charts are adjusted to preserve C^0 continuity along chart boundaries.

—If desired, the cross-boundary continuity can be improved by applying local relaxation as described in [Khodakovsky et al. 2003; Schreiner et al. 2004].

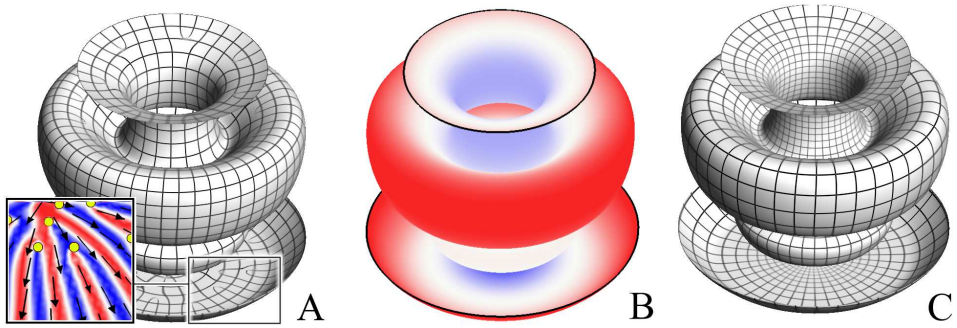


Fig. 7. Curl-correction applied to an object of revolution. A: the quasi-isometric parameterization contains singularities (shown as dots) which correspond to sources of the diverging vector field; B: scaling factor - the magnitude of the curl-corrected vector fields; C: solution obtained with the curl-corrected vector fields.

In our experiments, only a small fraction (between 2 and 5 %) of the charts contain singularities and require this additional processing.

By combining the Periodic Global Parameterization (Section 2) with the reconstruction algorithm presented in this Section, it is possible to compute a global parameterization of a surface. The next section presents the two optional pre-processing stages applicable to the input vector fields.

4. VECTOR FIELD PROCESSING

This section presents two optional algorithms for processing the input vector fields before computing the periodic global parameterization. Section 4.1 shows how to rescale the control vector field to minimize the number of singularities. Section 4.2 presents a method for consistent vector field smoothing which is particularly useful when the input vector fields are the principal curvature directions. In this case the method extrapolates the vector fields in a consistent manner from the anisotropic zones of the mesh where the principal curvatures are well defined into the isotropic regions, where the principal curvature directions are undefined.

4.1 Curl Correction

In the formulation presented in Section 2, the emphasis was on constructing a parameterization as isometric as possible. We therefore normalized the input vector fields to account for uniform spacing. This formulation is suitable for many applications, such as 3D paint systems, where parametric distortion is the dominant consideration.

For other applications isometry is less important, while the number of singularities in the parameterization is a serious concern. For instance, in quad-dominant remeshing, each branching point results in an undesirable T-vertex in the mesh.

We therefore introduce an optional pre-processing technique, that scales the vector fields prior to parameterization in order to minimize the number of T-vertices. As a result of the scaling the resulting parameterizations remain conformal (since the fields remain orthogonal and have the same norm) but are no longer isometric.

Since no-zero curl in the vector field leads to singularities in the parameterization, the goal of the proposed rescaling is to minimize the curl of the control vector fields $\vec{K}$ and

$\vec{K}^{\perp}$. Since in our initial setting (Section 2.2) the control vector fields are normalized, curl can arise only from non-parallel vectors (directional curl). As a consequence, eliminating the curl means rescaling the vector fields in such a way that the modular curl, arising from variations of the norm $\|\vec{K}\|$, cancels the directional curl. Note that our problem is different from computing a Hodge decomposition (see e.g., [Tong et al. 2003]), since we want to preserve the directions of the vector fields.

Given a unit vector field $\vec{K}$ defined over a surface S, we want to find a scalar field v such that $curl(v\vec{K}) = \vec{0}$. The vectors will become shorter in converging regions $(v < 1)$ and longer in diverging regions $(v > 1)$. Note that since $\vec{K}$ and $\vec{K}^{\perp}$ are coupled by relationship $curl(\vec{K}) = div(\vec{K}^{\perp})\vec{N}$ (where $\vec{N}$ denotes the normal to S), the same scaling v needs to be applied to both $\vec{K}$ and $\vec{K}^{\perp}$. In terms of complex analysis, this coupling can be explained also by the fact that the θ and ϕ functions determine a *complex potential* of $\vec{K}$, which is necessarily a *conformal* function (see [Needham 1994]).

To develop a simple linear formulation for computing v we use a different setting for defining $\vec{K}$ (and $\vec{K}^{\perp}$) than the one used in the main parameterization procedure (Section 2). Note that since the two procedures are stand-alone, this has no baring on the final result. For curl-correction purposes we assume that the *direction* of $\vec{K}$ varies linearly over T. In other words, given a local orthonormal frame (x,y) of T, we can parameterize the vector $\vec{K}$ by the angle γ between $\vec{K}$ and the x axis: $\vec{K} = (\cos(\gamma), \sin(\gamma))$, with $\gamma = ax + by + c$ (γ varies linearly over T). Replacing $\vec{K}$ with this expression we can express the zero-curl requirement per triangle as:

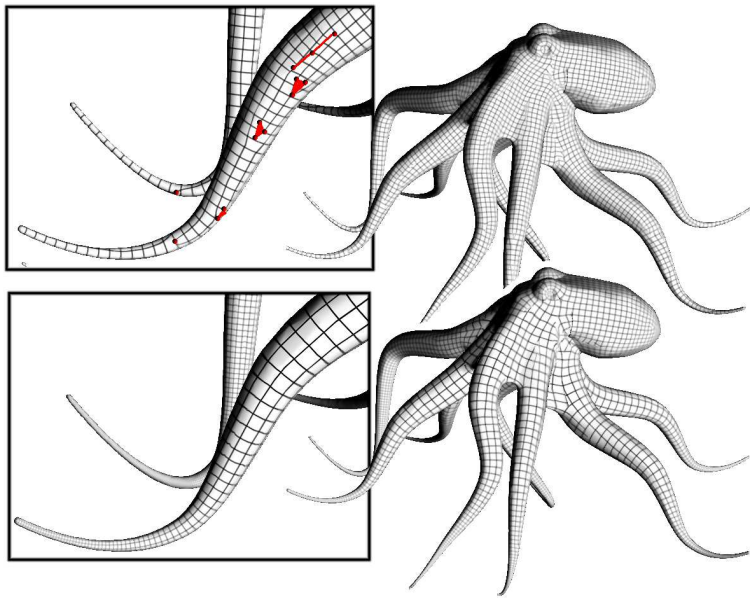


Fig. 8. Curl-correction applied to a model with sharp features. Note how all the singularities (triangles, edges, and vertices) on the tentacles (top) are removed by the curl correction (bottom).

$$curl(v\vec{K})\cdot\vec{N} = \left(-\frac{\partial v}{\partial y}+va\right)\cos(\gamma)+\left(\frac{\partial v}{\partial x}+vb\right)\sin(\gamma)=0 \tag{17}$$

where $\gamma = ax+by+c$

We search for solutions of Equation 17 which are independent of rotations applied to the vector field $\vec{K}$, i.e., independent of the constant c. The solutions of the following system of PDEs meet this requirement:

$$\begin{cases} -\partial v/\partial y + va &= 0 \\ \partial v/\partial x + vb &= 0 \end{cases} \tag{18}$$

The solutions of Equation 18 have the form $v = Ce^{ay-bx}$, where C denotes a constant. To solve for the values v_i of v at the vertices *globally*, we express the condition satisfied by the variations of v:

$$\begin{array}{ll} \log(v) & = log(C)+ay-bx \\ \nabla\log(v) & = \nabla(ay-bx) = (-b \;\; a) \end{array} \tag{19}$$

Since, a solution with zero-curl over the entire surface might not exit, we solve for the $\tilde{v}_i = \log(v_i)$'s using a least squares formulation:

$$G(\tilde{v}) = \sum_T A_T \left\| J_T \begin{pmatrix} \tilde{v}_1 \\ \tilde{v}_2 \\ \tilde{v}_3 \end{pmatrix} - \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} J_T \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{pmatrix} \right\|^2 \tag{20}$$

where:

$$J_T = 1/2A_T \begin{pmatrix} y_2-y_3 & y_3-y_1 & y_1-y_2 \\ x_3-x_2 & x_1-x_3 & x_2-x_1 \end{pmatrix}$$

where (x_i, y_i) are the coordinates of the vertices of T in the local frame.

Since the solution is independent of a global scaling applied to all the v_i's, we set $\tilde{v}_1 = 0$ and solve for all the other $\tilde{v}_i$'s. Then, we compute the scaling coefficients $v_i = \exp(\tilde{v}_i)$, and normalize them by dividing them by $\max(v_i)$.

The computed scaling coefficients v_i are introduced into Equation 5

$$F \quad = \quad \int_S \left(\|\nabla\theta - \omega v\vec{K}\|^2 + \|\nabla\phi - \omega v\vec{K}^{\perp}\|^2 \right) dS$$

and the parameterization algorithm proceeds as described in Section 2.

4.2 Vector Field Smoothing

In this section we describe a procedure that smoothes the input vector fields and extrapolates them into mesh areas where the associated directions are ill-defined. In our description we focus on the case where the input vector fields are the principal directions of curvature. In this case the anisotropy value $(|kmax/kmin|-1)$ indicates (vanishes) when the vector field directions are ill-defined. The procedure can be used for smoothing other types of fields if a replacement indicator of vector field fidelity is provided.

In all the examples in this paper we use an approximation of the principal directions of curvature as defined in [Cohen-Steiner and Morvan 2003] to define a geometrically meaningful control vector field. This choice provides parameterizations particularly suitable for approximation applications such as remeshing and surface fitting. The use of principal curvatures enables us to align the parameterization with the main features of the surface S. However, in isotropic regions, the principal directions of curvature are undefined. As a consequence, the estimation is meaningless in regions where the anisotropy $(|kmax/kmin|-1)$ vanishes.

To obtain vector fields which are well defined everywhere, we introduce a method for extending the directions from the anisotropic regions of the surface onto the isotropic ones. This method can be applied as a pre-processing step of our parameterization algorithm or can be used by other applications which make use of curvature fields. The curvature smoothing method presented in [Hertzmann and Zorin 2000] shares some similarities with ours. An advantage of our formulation is that it allows the use of efficient numerical solvers and exhibits faster convergence. Moreover, our method allows greater control of the singularities of the curvature field, as it can handle direction equality modulo $\pi/2$, π or 2π. As shown in Figure 9, this creates 8 quarter poles, 4 half-poles or two poles respectively for a sphere model.

To smooth the principal directions, we apply a regularized fitting procedure to a set of variables α_i, corresponding to the direction of $\vec{K}_i$. α_i is defined as the angle between the vector $\vec{K}_i$ and a reference direction $\vec{H}_i$ in the tangent plane of the vertex i. To obtain a reference direction we select an edge $\vec{e}$ emanating from i and project it to the plane:

$$\vec{H}_i = \text{normalize}\left(\overrightarrow{e} - (\overrightarrow{e}\cdot\vec{N}_i)\vec{N}_i\right)$$

where $\vec{N}_i$ is the normal at i. To smoothly fit the curvature directions, we minimize an energy functional providing a balance between fitting and smoothness. The fitting term aims at keeping the new α_i angles close to the original angles α_i^0, computed form the initial values of the vector field $\vec{K}$ at the vertices. As before, angle differences are approximated by the norm of the difference of the sine/cosine vectors, yielding the following formulation:

$$R = (1-\rho)\underbrace{\sum_i |kmax_i/kmin_i| \left\|(\cos\alpha_i,\sin\alpha_i) - (\cos\alpha_i^0,\sin\alpha_i^0)\right\|^2}_{\text{fitting term}} + \underbrace{\rho\sum_T R_T}_{\text{smoothing term}} \qquad (21)$$

where $kmax_i$ (resp. $kmin_i$) is the maximum (resp. minimum) curvature at vertex i. The user-defined coefficient ρ corresponds to the desired smoothing intensity (in all our examples, $\rho = 0.8$). The smoothing term R_T on a triangle T minimizes the variations of α over T and is given by:

$$R_T = \sum \lambda_i^T R_{T,i}$$

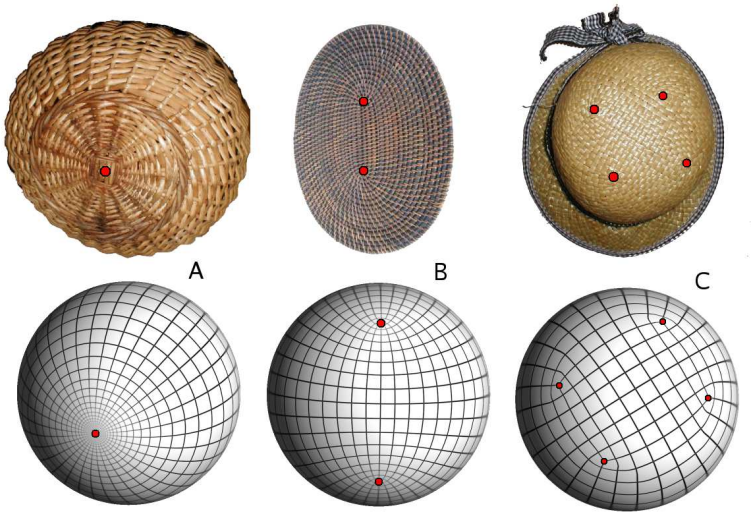


Fig. 9. Rattan objects and globally smooth parameterizations modulo 2π (A), π (B) and $\pi/2$ (C). A globally smooth parameterization modulo $2\pi/k$ has poles of index $1/k$ (red dots).

where the variation $R_{T,i}$ of α along the edge e_i is given by:

$$R_{T,i} = \left\| \begin{pmatrix} \cos\alpha_{i\oplus 2} \\ \sin\alpha_{i\oplus 2} \end{pmatrix} - \begin{pmatrix} \cos\beta_i & \sin\beta_i \\ -\sin\beta_i & \cos\beta_i \end{pmatrix} \begin{pmatrix} \cos\alpha_{i\oplus 1} \\ \sin\alpha_{i\oplus 1} \end{pmatrix} \right\|^2 \tag{22}$$

$$\text{with:} \quad \cos\beta_i = \vec{H}_{i\oplus 1} \cdot \vec{H}_{i\oplus 2} \quad ; \quad \sin\beta_i = (\vec{H}_{i\oplus 1} \times \vec{H}_{i\oplus 2}).N_T$$

In this equation, β_i denotes the angle between $\vec{H}_{i\oplus 1}$ and $\vec{H}_{i\oplus 2}$, and the λ_i^T's used to define triangle integrals are computed as described in Appendix B, Equation 24.

We optimize Equation 21 for the unknowns $(\cos\alpha_i, \sin\alpha_i)$, using the same solution mechanism as in Section 2.5. We introduce a penalty term that prevents the norm of the unknowns from vanishing.

Similar to the penalty function in Section 2.5, the penalty term evenly distributes the singularities over the surface, as shown in Figure 9.

Given $(\cos\alpha_i, \sin\alpha_i)$ we recompute the vector field $\vec{K}_i = \cos\alpha_i \vec{H}_i + \sin\alpha_i \vec{H}_i \times \vec{N}_i$ and $\vec{K}_i^{\perp} = \vec{N}_i \times \vec{K}_i$.

Using the formulation given in Equations 21 and 22, equality between the α_i's is considered modulo 2π. As a consequence, the created singular points are *poles*. Namely, they are points around which the vector field winds once, which corresponds to a 2π rotation.

To allow rotations of π (or $\pi/2$) (half-poles and quarter-poles), it is possible to enforce equality modulo π (or $\pi/2$) by solving for intermediary variables $\tilde{\alpha}_i = 2\alpha_i$ (or $\tilde{\alpha}_i = 4\alpha_i$). In practice, this simply means dividing all the α_i^0's and the β_i's by 2 (resp. 4), minimizing Equation 21, and multiplying the α_i's by 2 (resp. 4). The result on a sphere is shown in Figure 9. Either two poles, four half-poles, or eight quarter-poles are obtained (the total number of poles weighted by their multiplicity sums to $2-2g$ as predicted by the Hopf-Poincaré theorem). As can be seen in Figure 9-C, this property is used by hat makers,

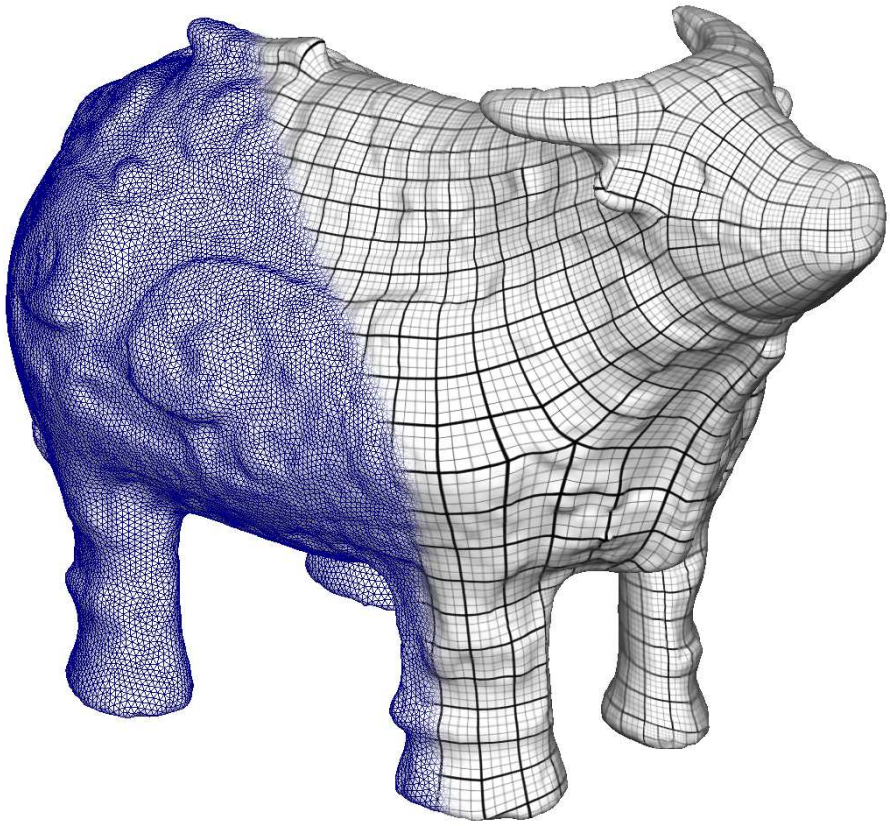


Fig. 10. Our method applied to the "buffle" data set.

who prefer using four quarter-poles rather than introducing "branching" singular points. In the case of modulo $\pi/2$ and modulo π equality, our method has some similarity with the vector field preprocessing described in [Wei and Levoy 2001]. The main difference is that our formulation with periodic variables allows the use of efficient numerical solvers. In addition, and thanks to our *global* formulation, the numerical solver evenly distributes the singular points over the surface. This is not observed with common *local* relaxation procedures.

5. RESULTS AND APPLICATION TO REMESHING

The Periodic Global Parameterization method (PGP) was implemented as part of the Graphite [Graphite 2003] mesh processing package. Throughout the paper we demonstrate the parameterizations computed with PGP on different complex models.

Figures 1, 10 show the final parameterization and the extracted chart structure. Figure 12 shows the iso $k\pi$ lines obtained with the "David" dataset. Those Figures demonstrate that even on very complex models the number of singularities generated by our method remains very small ($2-3\%$ of the triangles). The number is particularly low when curl-correction is applied (zero in the example in Figure 7 and 27 for the octopus in Figure

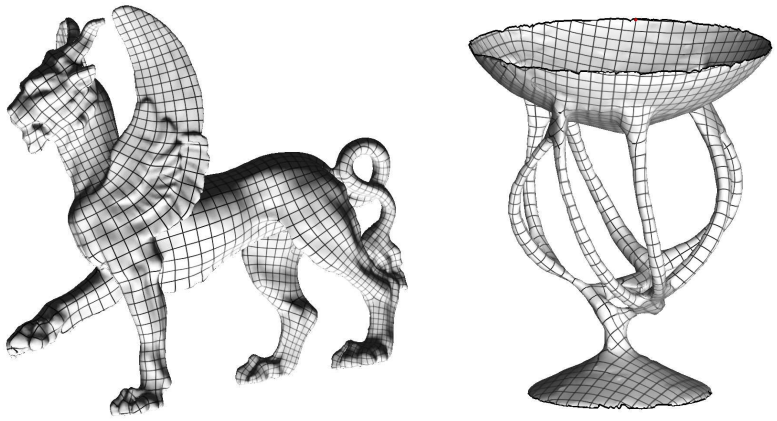


Fig. 11. Periodic global parameterizations with various genii (iso $k\pi$ lines).

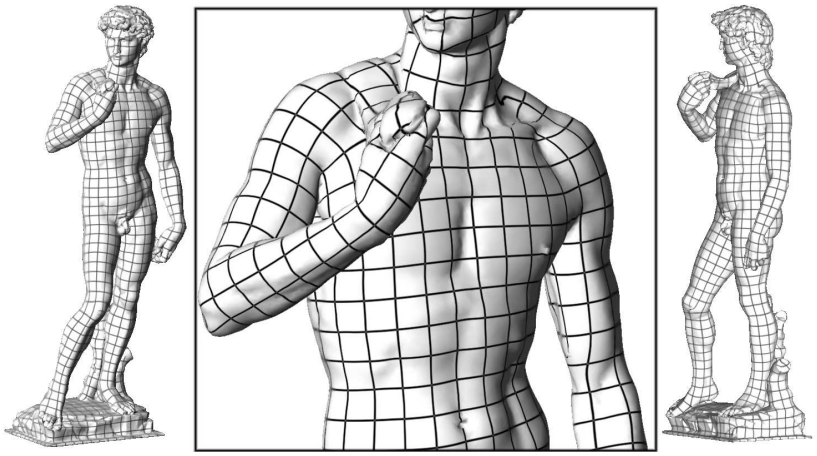


Fig. 12. Quasi-isometric global parameterization of the 200K facets "David" data set (iso $k\pi$ lines).

8). When no curl-correction is applied, we obtain quasi-isometric parameterization and in this case the singularities are located where we intuitively expect them - in regions where the local-feature-size changes. Figures 11,12 demonstrate that our method can work on models of any genus, as well as models with boundaries.

Table I provides distortion statistics and timings for our approach. The times were measured on a 1.7 GHz machine. For two models we compare the distortion caused by our method, with that caused by global conformal parameterization [Gu and Yau 2003]. Both with and without curl-correction the stretch introduced by our method is drastically lower compared to the other method, while the shear is slightly larger. We also compared our results to those generated using stretch-minimizing parameterization [Sander et al. 2002] after cutting the model using [Sheffer and Hart 2002]. The cuts were used to generate disk topology and reduce the stretch. The number of singularities for the models parameterized using this techniques is the number of boundary vertices, since these are the points of

Model	♯Δ	Algorithm	Stretch	Shear	time
Horse	20K	Gu et al.	6.777	0.07	NA
		PGP	1.07	0.20	45 s.
		ccPGP	1.176	0.12	53 s.
Bunny	25K	Gu et al.	2.65	0.042	NA
		PGP	1.029	0.167	58 s.
		ccPGP	1.14	0.14	1 min. 12 s.
Bull	34.5K	Sander et al.	1.030	0.1558	1 min. 11 s.
		PGP	1.064	0.1774	1 min. 26 s.
		ccPGP	1.209	0.0885	1 min. 35 s.
Camel	78K	Sander et al.	1.053	0.227	3 min. 51 s.
		PGP	1.048	0.1596	5 min. 46 s.
		ccPGP	1.654	0.0711	6 min. 51 s.
David	200K	PGP	1.121	0.2398	17 min. 35 s.
		ccPGP	1.270	0.1310	20 min. 43 s.
Lion	400K	PGP	1.123	0.1728	33 min. 42 s.
		ccPGP	1.425	0.0826	45 min. 18 s.

Table I. Statistics and timings of our method without and with curl-correction (PGP and ccPGP respectively). The numbers are compared, when data is available, to Gu et al.'s global parameterization and to Sander et al.'s method. The number of singularities for Sander et al. is the number of vertices on the cut.

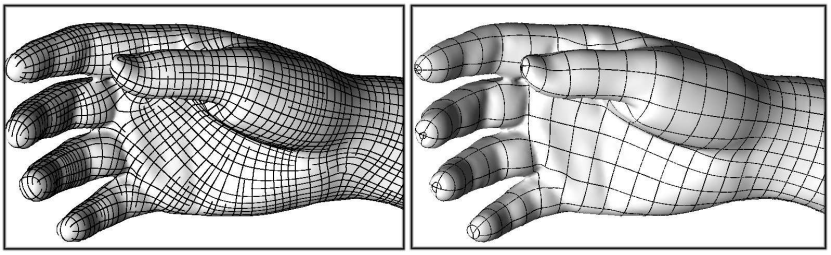


Fig. 13. Left: explicit remeshing using placement of lines of curvature generates an uneven sampling density and gaps. Right: implicit remeshing using our method generates a more regular sampling.

discontinuity in this context. Even though our parameterization technique is much more constrained due to the requirement of vector field alignment the distortion introduced by the two methods is comparable. For the Camel model, PGP introduces less distortion than stretch-minimizing parameterization, both in terms of stretch and in terms of shear. For the Bull model, our method introduces slightly more distortion.

Curvature aligned parameterizations are particularly suitable for surface approximation applications such as remeshing and surface fitting. In the next section we explain how to use the parameterizations generated by PGP for curvature aligned quad-dominant remeshing.

Implicit Remeshing

As explained in [d'Azevedo 2000], approximation theory predicts that aligning the elements with the axes of curvature improves the convergence of remeshing methods. This idea was exploited in [Alliez et al. 2003], that places a set of streamlines which are everywhere tangential to the estimated principal curvature directions. Those streamlines are

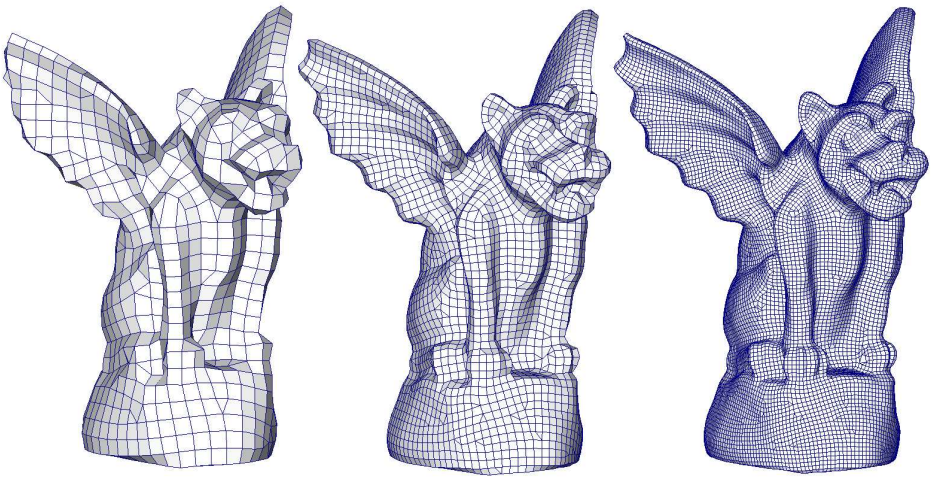


Fig. 14. Quad-dominant remeshings of the gargoyle data set with three different resolutions.

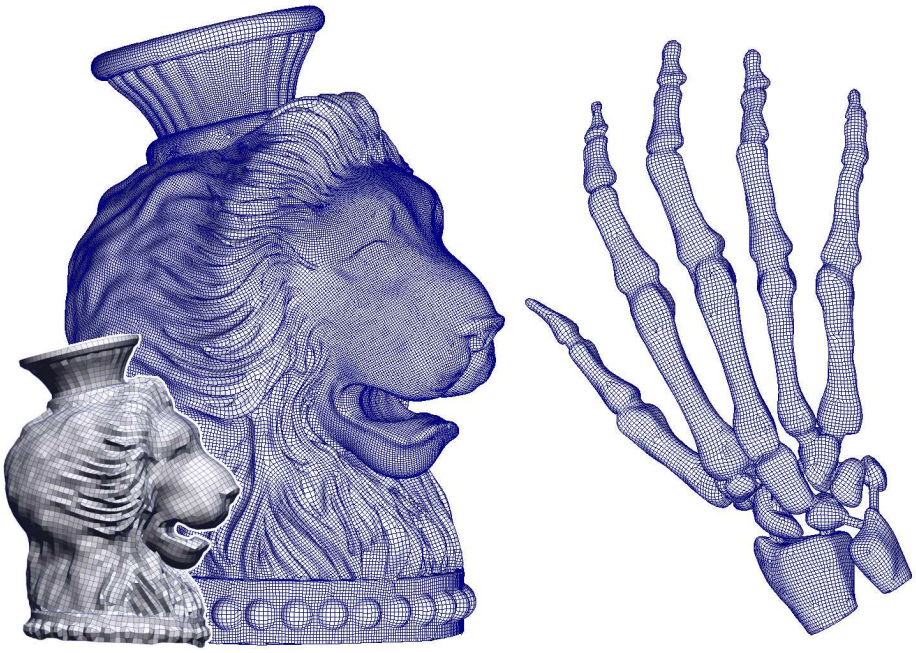


Fig. 15. Quad-dominant remeshing. Left: remeshing the "lion" dataset with two different resolutions; Right: this dataset presents thin features, likely to be missed by explicit (Runge-Kutta) remeshing algorithms.

characterized by an ordinary differential equation which solution can be approximated by solving an explicit integration scheme such as Runge-Kutta. This integration is computed either in parameter space [Alliez et al. 2003], or directly onto the input mesh [Marinov and Kobbelt 2004]. The main difficulty with this approach is to evenly distribute the streamlines over the surface, so as to minimize the number of streamline terminations while matching a local ideal line density. In previous work, this was done by using variants of the local streamline seeding strategy described in [Jobard and Lefer 1997]. However, the greedy nature of this approach results in an uneven placement of the streamlines (see Figure 13 (Left)).

The method described in [Dong et al. 2004] is an attempt at designing an *implicit* scheme, and shares some common points with our method. The method constructs a harmonic scalar function, and extracts a set of streamlines from a subset of iso-curves of this function. However, this method requires the user to manually define the singular points, does not align the streamlines to the features of the surface, and still requires an explicit integration for placing a set of streamlines along the orthogonal direction.

Using a parameterization computed by our method the remeshing procedure is straightforward. We find the sines and cosines of the parameter functions and compute the per-triangle parameterization (Sections 2,3.1). For remeshing purposes we can skip the subsequent chart-layout and singularities removal stages. We then directly extract the iso-θ and iso-ϕ lines of the parameterization as an initial polygonal mesh. The process is similar to the chart-boundary extraction (Section 3.3). The only difference is that instead of iso-$2k\pi$ curves we extract curves at a different density, specified by the user. Since the parameterization is quasi-conformal we obtain well shaped quadrilaterals everywhere, except in the immediate vicinity of the parameterization singularities. As done in [Alliez et al. 2003], the polygons which are not quadrilateral are split into a union of quads and triangles. T-vertices are handles by introducing additional triangles into the mesh.

Figure 15 shows examples of models remeshed using our method. The quality of the meshes is very high with most elements being near-perfect squares, aligned with the curvature. The "lion" dataset is remeshed with two different resolutions. The same parameterization was used for both remeshes and the only difference was in the density parameter used for extracting iso-lines. The "hand" dataset has some extremely thin features such as the tubular bones on the palm. Our method generates near-perfect square elements in these regions. Previous explicit streamline integration techniques would likely fail to capture these features.

Conclusions and Future Work

In this paper, we proposed a new globally smooth parameterization method for triangle meshes. The main advantage of the method over previous techniques is its ability to align the parameterization with orthogonal input vector fields. The parameterization is obtained by optimization of two periodic scalar functions so that their gradients are as tangential as possible to the input vector fields. When using the principal curvature directions as input vector fields, a geometrically meaningful quad-dominant chart layout is obtained by extracting the iso-curves of the optimized functions.

We demonstrate an application of our parameterization to quad-dominant remeshing, where most elements are well shaped, and evenly spaced. Other possible applications include surface fitting, texture synthesis and geometry compression using geometry images.

These will be described in future papers.

Our method offers many opportunities for future work. In particular, we would like to investigate the possibility to add more interactivity into the design of the input vector fields by removing some degrees of freedom from the energy functional. Improving the efficiency and the interactivity of the method using e.g. hierarchical solvers [Aksoylu et al. 2004] is also of great interest.

Acknowledgements

We thank the AIM@SHAPE European Network of Excellence, INRIA ARC GEOREP, and NSERC for their support, Pedro Sander and David Gu for providing us with some statistics on their methods, and Laurent Alonso for help with the proof in Appendix A. We also thank Stanford University, the ENST and SensAble for their 3D data sets.

REFERENCES

AKSOYLU, B., KHODAKOVSKY, A., AND SCHRODER, P. 2004. Multilevel solvers for unstructured surface meshes. *Submitted to SIAM J. Sci. Comput.*.

ALLIEZ, P., STEINER, D. C., DEVILLERS, O., LEVY, B., AND DESBRUN, M. 2003. Anisotropic Polygonal Remeshing. *ACM TOG (SIGGRAPH)*.

COHEN-STEINER, D. AND MORVAN, J.-M. 2003. Restricted delaunay triangulations and normal cycle.

D'AZEVEDO, E.-F. 2000. Are bilinear quadrilaterals better than linear triangles? *SIAM J. of Scientific Computing 22,* 1, 198–217.

DONG, S., KIRCHER, S., AND GARLAND, M. 2004. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. Tech. rep. August.

FLOATER, M. 1997. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design 14,* 3 (April), 231–250.

FLOATER, M. S. 2003. Mean value coordinates. *CAGD 20*, 19–27.

FLOATER, M. S. AND HORMANN, K. 2004. Surface parameterization: a tutorial and survey. In *Advances on Multiresolution in Geometric Modelling*, M. S. F. N. Dodgson and M. Sabin, Eds. Springer-Verlag, Heidelberg.

GORTLER, S., GOTSMAN, C., AND THURSTON, D. 2004. One-forms on meshes and applications to 3d mesh parameterization. Tech. rep., Harvard University.

GOTSMAN, C., GU, X., AND SHEFFER, A. 2003. Fundamentals of spherical parameterization for 3d meshes. *ACM TOG (SIGGRAPH) 22*, 358–363.

GRAPHITE. 2003. http://www.loria.fr/ levy/Graphite/index.html.

GRIMM, C. AND HUGUES, J. 1995. Modeling surfaces of arbitrary topology using manifolds. In *SIGGRAPH conference proceedings*.

GU, X., GORTLER, S. J., AND HOPPE, H. 2002. Geometry images. In *SIGGRAPH conf. proc.* ACM Press, 355–361.

GU, X. AND YAU, S.-T. 2003. Global conformal surface parameterization. In *Symposium on Geometry Processing*. ACM.

HERTZMANN, A. AND ZORIN, D. 2000. Illustrating smooth surfaces. In *SIGGRAPH Conference Proceedings*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA.

HORMANN, K. AND GREINER, G. 2000. MIPS: An efficient global parametrization method. In *Curve and Surface Design*.

JOBARD, B. AND LEFER, W. 1997. Creating evenly-spaced streamlines of arbitrary density. In *Proc. of the Workshop on Vis. in Sci. Comp.* Eurographics.

KHODAKOVSKY, A., LITKE, N., AND SCHRODER, P. 2003. Globally smooth parameterizations with low distortion. *ACM TOG (SIGGRAPH)*.

KRAEVOY, V. AND SHEFFER, A. 2004. Cross-parameterization and compatible remeshing of 3d models. *ACM TOG (Proc. SIGGRAPH 2004)*.

LÉVY, B., PETITJEAN, S., RAY, N., AND MAILLOT, J. 2002. Least Squares Conformal Maps for Automatic Texture Atlas Generation. *ACM Transactions on Graphics (SIGGRAPH conf. proc.)*, 362–371.

LI, W. C., LEVY, B., AND PAUL, J.-C. 2005. Mesh editing with an embedded network of curves. In *Shape Modeling International conference proceedings*.

MARINOV, M. AND KOBBELT, L. 2004. Direct anisotropic quad-dominant remeshing. In *Proc. Pacific Graphics*. 207–216.

NEEDHAM, T. 1994. *Visual Complex Analysis*. Oxford Press.

PRAUN, E., FINKELSTEIN, A., AND HOPPE, H. 2000. Lapped textures. In *SIGGRAPH 00 Conf. Proc.* ACM Press, 465–470.

SANDER, P., GORTLER, S., SNYDER, J., AND HOPPE, H. 2002. Signal-specialized parametrization. In *Eurographics Workshop on Rendering*.

SCHREINER, J., PRAKASH, A., PRAUN, E., AND HOPPE, H. 2004. Inter-surface mapping. *ACM TOG (Proc. SIGGRAPH 2004)*.

SHEFFER, A. AND DE STURLER, E. 2001. Parameterization of faceted surfaces for meshing using angle based flattening. *Engineering with Computers 17*, 326–337.

SHEFFER, A. AND HART, J. 2002. Seamster: Inconspicuous low-distortion texture seam layout. In *IEEE Visualization*.

TARINI, M., HORMANN, K., CIGNONI, P., AND MONTANI, C. 2004. Polycube-maps. *ACM TOG (SIGGRAPH)*.

TONG, Y., LOMBEYDA, S., HIRANI, A., AND DESBRUN, M. 2003. Discrete multiscale vector field decomposition. *ACM TOG (SIGGRAPH)*.

WEI, L.-Y. AND LEVOY, M. 2001. Texture synthesis over arbitrary manifold surfaces. In *SIGGRAPH 2001, Computer Graphics Proceedings*, E. Fiume, Ed. ACM Press / ACM SIGGRAPH, 355–360.

WEISSTEIN, E. W. Manifold.

YING, L. AND ZORIN, D. 2004. A simple manifold-based construction of surfaces of arbitrary smoothness. *ACM TOG (SIGGRAPH)*.

ZELINKA, S. AND GARLAND, M. 2003. Interactive texture synthesis on surfaces using jump maps. In *Symposium on Rendering*. Eurographics.

A. INTEGRAL OF A PIECEWISE LINEAR VECTOR FIELD

Given a piecewise linear vector field $\vec{K}$ and its average value $\vec{K}_T$ on a triangle T, we prove that the two energy functionals $F_T = \int_T (\nabla\theta - \omega\vec{K})^2 ds$ and $F'_T = \int_T (\nabla\theta - \omega\vec{K}_T)^2 ds$ have the same minimizer:

$$\begin{aligned} F_T &= \int_T (\nabla\theta - \omega\vec{K})^2 ds \\ &= \int_T \left((\nabla\theta - \omega\vec{K}_T) + (\omega\vec{K}_T - \omega\vec{K}) \right)^2 ds \\ &= \int_T (\nabla\theta - \omega\vec{K}_T)^2 + 2\int_T (\omega\vec{K}_T - \omega\vec{K})^t (\nabla\theta - \omega\vec{K}_T) ds + \int_T (\omega\vec{K}_T - \omega\vec{K})^2 ds \\ &= \int_T (\nabla\theta - \omega\vec{K}_T)^2 + 2(\nabla\theta - \omega\vec{K}_T)^t \int_T (\omega\vec{K}_T - \omega\vec{K}) ds + \int_T (\omega\vec{K}_T - \omega\vec{K})^2 ds \end{aligned} \tag{23}$$

The first term of this expression is F'_T, the second term vanishes (by definition of the average value $\vec{K}_T$) and the third term does not depend on θ. As a consequence, we have $F_T = F'_T + \text{constant}$. Therefore, F_T and F'_T have the same minimizer.

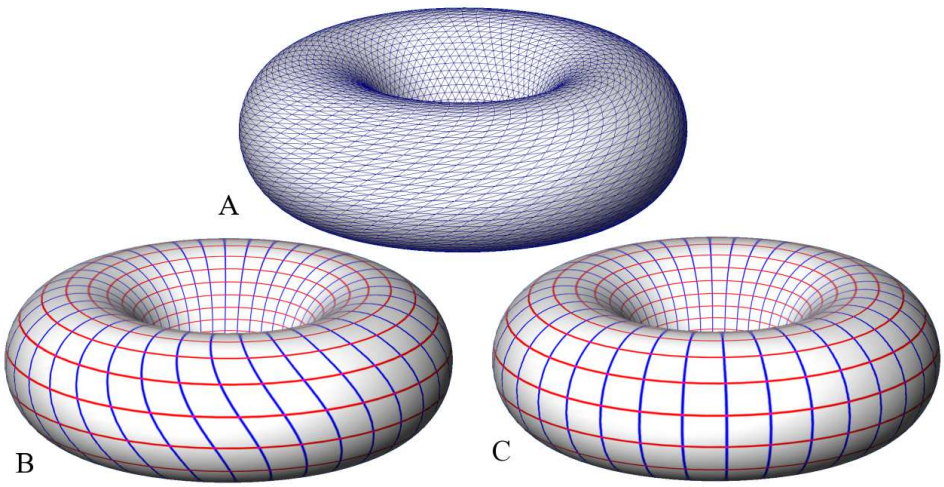


Fig. 16. A: a meshed torus with a strong mesh anisotropy; B: result of the edge-based PGP: the parameterization is influenced by the mesh anisotropy; C: result of the triangle-based PGP: the parameterization solely depends on the geometry.

B. ENERGY INTEGRATED OVER THE TRIANGLES

The energy F_T integrated over the triangle T can be expressed as a linear combination of the $F^{\theta}_{T,i}$ and $F^{\phi}_{T,i}$ edge energies:

$$F_T = (\|\nabla\theta - \omega\vec{K}_T\|^2 + \|\nabla\phi - \omega\vec{K}_T^{\perp}\|^2)A_T = \sum_{i=1}^{3} \lambda_i^T (F^{\theta}_{T,i} + F^{\phi}_{T,i})$$

where $(\lambda_1^T, \lambda_2^T, \lambda_3^T)$ are the solutions of :

$$\begin{pmatrix} (e_{1,x})^2 & (e_{2,x})^2 & (e_{3,x})^2 \\ (e_{1,y})^2 & (e_{2,y})^2 & (e_{3,y})^2 \\ 2e_{1,x}e_{1,y} & 2e_{2,x}e_{2,y} & 2e_{3,x}e_{3,y} \end{pmatrix} \begin{pmatrix} \lambda_1^T \\ \lambda_2^T \\ \lambda_3^T \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \tag{24}$$

The linear system is obtained by expanding and equating both terms of Equation 24. Figure 16 compares the results obtained with the edge-based and the triangle-based energy on a mesh with a strong anisotropy.