A Wave-based Anisotropic Quadrangulation Method

Muyang Zhang Jin Huang* Xinguo Liu Hujun Bao

State Key Lab of CAD&CG, Zhejiang University

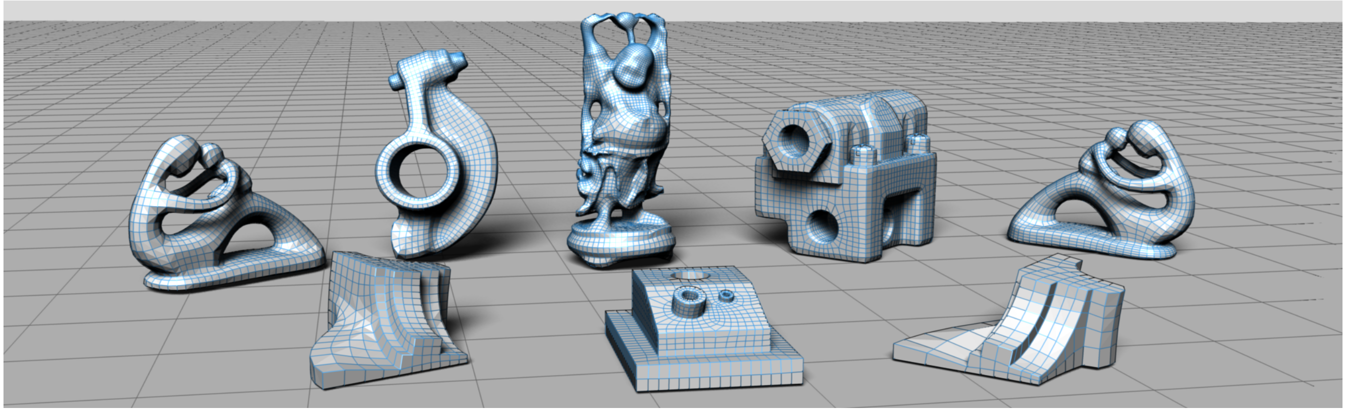


Figure 1: *A collection of quadrangulation results generated by the proposed method.*

Abstract

This paper proposes a new method for remeshing a surface into anisotropically sized quads. The basic idea is to construct a special standing wave on the surface to generate the global quadrilateral structure. This wave based quadrangulation method is capable of controlling the quad size in two directions and precisely aligning the quads with feature lines. Similar to the previous methods, we augment the input surface with a vector field to guide the quad orientation. The anisotropic size control is achieved by using two size fields on the surface. In order to reduce singularity points, the size fields are optimized by a new curl minimization method. The experimental results show that the proposed method can successfully handle various quadrangulation requirements and complex shapes, which is difficult for the existing state-of-the-art methods.

Keywords: anisotropic quadrangulation, standing wave, feature alignment, size control

1 Introduction

Surface quadrangulation is a useful modeling tool for many applications, such as finite element analysis and B-spline fitting. However, it is very challenging to convert a surface into a pure quadrangular mesh. The first challenge is how to create a global quadrilateral partition of the input mesh, which determines the overall quality of the final quadrangulation result. Many earlier methods without such global partition can only obtain quad dominant remeshing results. The second challenge is how to meet various local quadrangulation constraints desired by the users/applications. For example, the edges of the quads should be aligned with the nearby feature lines of the input mesh, particularly when remeshing CAD models. And it is desired that the size of the quad can adapt to the local geometry details, or can be adjusted as freely as possible for particular application purposes. At last, the shape complexity, such as small handles and large number of genus, brings some extra issues for remeshing methods. Most of the previous methods [Alliez et al. 2003; Boier-Martin et al. 2004; Dong et al. 2005; Ray et al. 2006; Bommes et al. 2009] can handle soft orientation control by using direction field to guide the quads. However, they can only generate nearly uniform sized quads. Dong et al. [2006] partition the surface into quadrangular charts using the Morse-Smale complex (MSC) of a scalar function. Huang et al. [2008] extended this method to control the quad size. However their size control method is limited to square quads and not intuitive.

In this paper, we propose a new anisotropic quadrangulation method based on standing wave construction and quasi-dual MSC extraction [Dong et al. 2006]. Our goal is to develop a method that can precisely align the quads with the geometric feature lines or the user specified directions, and provide a flexible anisotropic size control on the quads. Since the final quadrangulation result is essentially determined by the quasi-dual MSC, we formulate these requirements as a single optimization problem when constructing the wave function. Similar to the previous methods, we define a vector field to control the quad orientation. In addition, we define a pair of size fields to control the size of the quads. However, some arbitrarily specified size fields may lead to unnecessary singularity points in the remeshing results. To address this issue, we describe a curl minimization method to optimize the size fields.

The technical contributions of our work include a wave construction method and a curl correction method for the size fields. Our wave based quadrangulation method is featured by: 1) It can remesh a surface into anisotropically sized quads. 2) It can control the quad orientation and size, and align the quad with feature lines in a uniform and flexible way. 3) It can remesh complex shapes with various features, such as small handle/ring, which has been very difficult for previous methods.

*Corresponding author. E-Mails: {hj, xgliu, bao}@cad.zju.edu.cn

2 Related Work

A comprehensive survey on quadrangulation methods can be found in [Alliez et al. 2005; Hormann et al. 2008]. We will review some of the most related ones by how a global quadrilateral structure is constructed, since it is a very important issue for quadrangulation.

Structure from Heuristic Procedure Marinov and Kobbelt [2006] developed a variational shape approximation method to partition the input mesh into nearly developable regions. The regions are then individually remeshed into quads with constraints on the region boundaries to ensure a consistent remeshing. This method handles CAD models well, but doesn't optimize the singularity positions. [Lai et al. 2008] used local topology operation to incrementally convert triangles into quadrangles. The quad orientation aligned to the principal curvature directions, and the quad size is adaptively adjusted according to the local curvature. A major drawback of these heuristic based methods is that they can only obtain quad-dominant remeshing results. In addition, there is no control for the singularity points.

Structure from Curve Tracing By augmenting the surface with direction field, several previous works [Alliez et al. 2003; Boier-Martin et al. 2004; Dong et al. 2005; Tong et al. 2006] generate the quadrilateral structure by directly tracing curves along the direction field. The direction field is defined as the principal curvature directions [Alliez et al. 2003; Boier-Martin et al. 2004] or the gradient of harmonic functions [Dong et al. 2005; Tong et al. 2006]. Though they can naturally control the quad orientation, they usually produce too many singularities. Quad remeshing can also be achieved by computing a global continuous parameterization for the input mesh [Ray et al. 2006; Kälberer et al. 2007]. In these method, a good direction field is required for a high quality quadrangulation, since the singularities are mostly prescribed by the direction field. Fortunately, many approaches have been proposed to design and smooth direction fields for a variety of applications [Zhang et al. 2006; Palacios and Zhang 2007; Ray et al. 2008].

Structure from MSC Dong et al. [2006] proposed an excellent method to generate the global quadrilateral structure by extracting the quasi-dual MSC [Edelsbrunner et al. 2001; Bremer et al. 2004] from a spectral function on the input mesh. The MSC based method can automatically optimize the distribution of the singularity points. Huang et al. [2008] formulate a constrained generalized eigen-problem to define the scalar function. The scalar function can be regarded as a periodic function, and the period is prescribed by choosing an appropriate eigenvalue. However, there is no guarantee that the period determined in this way is compatible with the orientation and alignment control. Thus, the spectral quadrangulation method may fail to generate high quality quads when complex alignment controls are imposed.

3 Standing Wave and Quadrangulation

In this section, we introduce a novel formulation for the scalar function based on standing waves. Let S be the input surface, represented by a triangular mesh $\mathcal{M} = (P, K)$, where $P = \{p_i\}_{i=1}^n$ is the vertex set of the mesh, and K represents the connectivity of the mesh.

Let scalar function w on S be a standing wave and have a separated representation: $w(x, y, t) = f(x, y)\psi(t)$. According to the wave equation, we have

$$\frac{\nabla^2 f}{f} = \frac{\ddot{\psi}}{c^2 \psi} \equiv -\lambda, \tag{1}$$

where $\lambda > 0$ is a constant value.

In the spectral based method [Dong et al. 2006], the scalar function used for extracting the MSC is the eigenvector of the Laplacian matrix, i.e. the solution of $\nabla^2 f = -\lambda f$. In this sense, the spectral based method actually adopted a special type of standing wave that has the same time frequency all over the mesh. Huang et al. [2008] generalize the spectral method with non-uniform λ. However, both of these two methods can not perform anisotropic quadrangulation, since there is no way to control the anisotropic quad size.

To enable anisotropic size control, we assume that the standing wave can be further separated into $f(x, y) = g(x)h(y)$. Then the standing wave equation in Equation 1 can be rewritten as follows:

$$\frac{\partial^2 g}{\partial x^2} = -\omega_x^2 g \quad \text{and} \quad \frac{\partial^2 h}{\partial y^2} = -\omega_y^2 h,$$

where ω_x and ω_y are two constant values satisfying $\omega_x^2 + \omega_y^2 = \lambda$. As we know, the general solutions for $g(x)$ and $h(y)$ are:

$$g(x) = A_x \cos(\omega_x x + c_x)$$
$$h(y) = A_y \cos(\omega_y y + c_y)$$

where A_x, A_y, c_x and c_y are constant values. We define $\theta = \omega_x x + c_x$, $\phi = \omega_y y + c_y$, and assume $A_x = A_y = 1$ without loss of generality. Then the standing wave $f(x, y)$ can represented as:

$$f(x, y) = \cos(\theta)\cos(\phi). \tag{2}$$

In [Huang et al. 2008], the same formulation is used to estimate the orientation of quasi-dual MSC for orientation alignment. In this paper, we will explore more properties of the wave function for flexible quad remeshing. First, the θ, ϕ parameters of the wave's extreme points are integer times of π. Second, the θ, ϕ parameters of the wave's saddle points are $\pi/2$ plus integer times of π. Recall that the primal MSC is constructed by connecting each saddle point to its neighboring extreme points, and the quasi-dual MSC is constructed by connecting the minimum-maximum diagonal within each Morse-Smale region of the primal MSC. Therefore, the quad orientation of the quasi-dual MSC actually coincides with the direction of iso-parameter lines, and the size of the quasi-dual MSC is half of the wave period, i.e., $(\pi/\omega_x, \pi/\omega_y)$.

In the following sections, we will present our quadrangulation method, which consists of 5 main steps:

1. Prepare the vector/size fields for quad orientation and size.
2. Construct a standing wave function on the mesh.
3. Extract the quasi-dual MSC of the wave.
4. Optimize the quasi-dual MSC and parameterize the mesh.
5. Subdivide the mesh into quads by the parameters.

The Step 3 and Step 5 are almost the same as the previous algorithms in [Dong et al. 2006]. We will detail the Step 1, 2, and 4 in turn in the following sections.

4 Vector/Size Field Preparation

We use a pair of orthogonal unit vector fields $(\vec{X}, \vec{Y})$ to guide the quad orientation. We define the vector fields by computing and smoothing the principal curvature directions [Cohen-Steiner and Morvan 2003; Ray et al. 2006]. We can also draw some desired vectors to replace the principal directions in the region of interest. In this case, the user specified vectors are assigned with large weights so that it can be maintained during the smoothing step. We also define a pair of size fields, (μ, η), to guide the size of the quads. Different from the previous works, we generate non-square quads, therefore each quad needs two length values to determine its size.

As shown in Figure 2, we record the vector fields and the size fields on the mesh edges [Bommes et al. 2009], since it is convenience for approximating the curl in Equation 4 and the angles in Equation 15. For any edge $e \in K$, we denote the corresponding vectors by $\vec{X}[e]$ and $\vec{Y}[e]$, and the corresponding size values by $\mu[e]$ and $\eta[e]$.

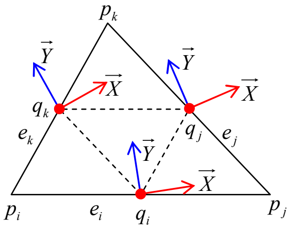


Figure 2: *The vector/size fields in a triangle. The vector fields are defined on the edges, and the size fields are defined at the middle points of the edges.*

In our wave construction method described in Section 5, $(\vec{X}, \vec{Y})$ prescribes the local coordinate frame, and (μ, η) prescribes the wave length. When there is a curl in the scaled vector field $(\mu\vec{X}, \eta\vec{Y})$, the state of the wave will change after traveling along a loop. Therefore, it is desired to reduce the curl. Ray et al. [2006] proposed an optimization method to reduce the curl for isotropic size fields. In the following, we will present a new method to deal with anisotropic size fields.

4.1 Size Field Optimization

Let $T = (p_i, p_j, p_k)$ be a triangle of the input mesh. As shown in Figure 2, the edges of T are denoted by $\{e_i, e_j, e_k\}$, and the middle points of the edges are denoted by $\{q_i, q_j, q_k\}$.

We first consider vector field $\mu\vec{X}$. In order to compute the curl of $\mu\vec{X}$, let $u - v$ be a local coordinate frame of triangle T, and $\vec{X} = (X_u, X_v)$ in the local frame. By the curl definition, we have

$$\mathrm{curl}(\mu\vec{X}) = \frac{\partial(\mu X_v)}{\partial u} - \frac{\partial(\mu X_u)}{\partial v} = \mathrm{curl}(\vec{X})\mu - \vec{X}^{\perp} \cdot \nabla\mu.$$

Thus, the curl free requirement, $\mathrm{curl}(\mu\vec{X}) = 0$, is equivalent to:

$$\mathrm{curl}(\vec{X}) = \frac{1}{\mu}\vec{X}^{\perp} \cdot \nabla\mu = \vec{X}^{\perp} \cdot \left(\frac{1}{\mu}\nabla\mu\right) = \vec{X}^{\perp} \cdot \nabla\tilde{\mu}, \tag{3}$$

where $\tilde{\mu} = \log(\mu)$. In the following two paragraphs, we will show how to approximate the curl and the gradient in above.

Curl Approximation By the relationship between the curl and the circulation density [Polthier and Preuß 2003], the curl in triangle T can be approximated as:

$$\mathrm{curl}(\vec{X})\,|_T \approx \frac{1}{|T|}\int_{\partial T} \vec{X} \cdot ds \approx \frac{1}{|T|}\sum_{e=e_i,e_j,e_k} \vec{X}[e] \cdot \vec{e}, \tag{4}$$

where $|T|$ denotes the area of triangle T, and $\vec{e}$ denotes the vector of edge e. Specifically, let v_1 and v_2 be the end points of edge e in counter-clockwise order in T, then $\vec{e}$ is defined as $\vec{e} = v_2 - v_1$.

Gradient Approximation Recall that the size values are defined at the middle points of the edges. Consider the size field on a triangle $\hat{T} = (q_i, q_j, q_k)$ formed by the middle points of the edges, as shown in Figure 2. For any point $p \in \hat{T}$, we interpolate the logged size field using the barycenteric coordinates: $\tilde{\mu}(p) = \sum_{\nu=i,j,k} \tilde{\mu}[e_\nu] B_{\hat{T},\nu}(p)$. Since the barycenteric coordinates $B_{\hat{T},\nu}(p)$ are linear functions of p in $\hat{T}$, their gradients are constant vectors. Therefore, the gradient of $\tilde{\mu}(p)$ is also constant in $\hat{T}$ and can be expressed as:

$$\nabla\tilde{\mu} = \sum_{\nu=i,j,k} \tilde{\mu}[e_\nu]\nabla B_{\hat{T},\nu}. \tag{5}$$

Combine Equations 3, 4 and 5, then we have the following equation for the curl free requirement in triangle T:

$$\sum_{\nu=i,j,k} b_{1T}^{\nu}\, \tilde{\mu}[e_\nu] - c_{1T} = 0, \tag{6}$$

where $b_{1T}^{\nu} = \vec{X}^{\perp} \cdot \nabla B_{\hat{T},\nu}$ and $c_{1T} = \mathrm{curl}(\vec{X})|_T$.

Vector Alignment Note that, the above curl and gradient approximation methods are valid only when the vectors of the triangle edges are properly aligned in terms of continuity. Let the vector $\vec{X}[e_i]$ of edge e_i be the reference. Whether the vector of edge e_j is consistently aligned can be determined by examining the dot product of $\vec{X}[e_i]$ and $\mathbf{R}_{90}^{r}\vec{X}[e_j]$, $r = 0, 1, 2, 3$, where $\mathbf{R}_{90}^{r}$ represents r times of 90° rotation around the surface normal [Ray et al. 2006; Ray et al. 2008]. Let

$$\kappa = \arg \max_{r \in \{0,1,2,3\}} \left\{ \vec{X}[e_i] \cdot \left(\mathbf{R}_{90}^{r}\vec{X}[e_j]\right) \right\}. \tag{7}$$

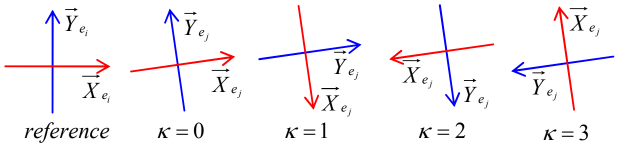


Figure 3: *Cases in vector field alignment.*

As shown in Figure 3, when $\kappa \neq 0$, the vector field is not well aligned, and we need to change the vector field at edge e_j by the following rules:

$$(\mu[e_j], \vec{X}[e_j]) = \begin{cases} (\eta_{old}[e_j], \ \ \vec{Y}_{old}[e_j]), & \text{if } \ \kappa = 1. \\ (\mu_{old}[e_j], -\vec{X}_{old}[e_j]), & \text{if } \ \kappa = 2. \\ (\eta_{old}[e_j], -\vec{Y}_{old}[e_j]), & \text{if } \ \kappa = 3. \end{cases} \tag{8}$$

In other words, we only need to substitute the variables of the size/vector field in Equations (4, 5 and 6) with the corresponding ones according to the above permutation rules when $\kappa \neq 0$. For clarity, we will omit such permutation in the following description.

Applying the above approximation approach to the orthogonal vector field $\eta\vec{Y}$ in triangle T, we have:

$$\sum_{\nu=i,j,k} b_{2T}^{\nu}\, \tilde{\eta}[e_\nu] - c_{2T} = 0, \tag{9}$$

where $b_{2T}^{\nu} = \vec{Y}^{\perp} \cdot \nabla B_{\hat{T},\nu}$ and $c_{2T} = \mathrm{curl}(\vec{Y})|_T$.

Curl Energy Now, we are ready to define a energy term of summed squares to measure the curl of the vector field:

$$\begin{aligned} E_{curl}(\tilde{\mu}, \tilde{\eta}) = \sum_{T \in K} \left(\sum_{\nu=i,j,k} b_{1T}^{\nu}\, \tilde{\eta}[e_\nu] - c_{1T} \right)^2 + \\ \sum_{T \in K} \left(\sum_{\nu=i,j,k} b_{2T}^{\nu}\, \tilde{\mu}[e_\nu] - c_{2T} \right)^2. \end{aligned} \tag{10}$$

However, the size field μ that satisfies $\text{curl}(\mu\vec{X}) = 0$ is not unique. First, it can be uniformly scaled by any global factor α, since $\text{curl}((\alpha\mu)\vec{X}) = \alpha\ \text{curl}(\mu\vec{X}) = 0$. Second, it can be multiplied by any size field ξ that satisfies $\vec{X}^{\perp} \cdot \nabla\xi = 0$, since $\text{curl}((\xi\mu)\vec{X}) = \text{curl}(\xi(\mu\vec{X})) = \text{curl}(\mu\vec{X})\xi - \mu\vec{X}^{\perp} \cdot \nabla\xi = 0$. So, the minimizer of the above curl energy is not unique, and the vibration of the minimizer may be arbitrarily large (since ξ can be multiplied by any times). Therefore, we can add extra constraints to obtain the optimal size field.

Size Constraints We define the following energy term to softly constrain the size field at some desired or randomly picked edges:

$$E_z(\tilde{\mu},\tilde{\eta}) = \sum_{i=1}^{n_c}(\tilde{\mu}[e_{c_i}]-\log(s_i))^2+\sum_{j=1}^{n_d}(\tilde{\eta}[e_{d_j}]-\log(t_j))^2, \quad (11)$$

where s_i is the constrained value of μ at edge e_{c_i}, and t_j is the constrained value of η at edge e_{d_j}.

Smoothness Constraints In order to suppress rapid vibration in the size field for generating smoothly sized quads, we define the following energy tern to minimize the Laplacian of the size field:

$$E_s(\tilde{\mu},\tilde{\eta}) = \| \mathbf{LP}\tilde{\mu} \|^2 + \| \mathbf{LP}\tilde{\eta} \|^2, \quad (12)$$

where $\mathbf{L}$ is the Laplacian matrix of the input mesh defined by the non-area weighted cotangent form Laplacian [Meyer et al. 2002], and $\mathbf{P}$ is a matrix defined to estimate the size value of each mesh vertex as the average of its incident edges.

Anisotropic Constraints We can also control the anisotropic degree of the quadrangulation results by constraining the ratio of the size field. For example, if the quad in the vicinity of edge e is expected to have a size ratio of γ_e, then we can constrain the size field at edge e by $\mu[e]/\eta[e] = \gamma_e$. Therefore, we define the following energy for anisotropic constraints:

$$E_a(\tilde{\mu},\tilde{\eta}) = \sum_{e\in\Gamma}(\tilde{\mu}[e] - \tilde{\eta}[e] - \log(\gamma_e))^2, \quad (13)$$

where Γ denotes the set of edges with anisotropic constraints.

It is interesting that the effect of the anisotropic constraint actually constrains the quad to be isotropic when all $\gamma_e = 1$. A typical application of the anisotropic constraint is to guide the quad size by ratio of the principal curvatures, as shown in Figure 9.

Total Energy Taking a weighted sum of the above energy term in Equations (10, 11, 12, 13), we define the following total energy functional for optimizing the size field:

$$\begin{aligned} E_{total}(\tilde{\mu},\tilde{\eta}) = \frac{1}{B^2}E_{curl}(\tilde{\mu},\tilde{\eta}) + w_z E_z(\tilde{\mu},\tilde{\eta}) \\ + w_s E_s(\tilde{\mu},\tilde{\eta}) + w_a E_a(\tilde{\mu},\tilde{\eta}), \end{aligned} \quad (14)$$

where B is the length of the model's bounding box, and $w_{z/s/a}$ are the weighting coefficients. We empirically set $w_z = w_s = 1.0$ and $w_a = 0.3$ as the default weighting values. Note that E_{curl} is weighted by $\frac{1}{B^2}$ to eliminate its dependence on model size. Equation 14 is linear least squares, which can be solved very efficiently. Finally, μ and η are obtained by taking the exponential of $\tilde{\mu}$ and $\tilde{\eta}$.

5 Wave Construction

In this section, we use the vector field as the local coordinate frame to construct a wave function on the input mesh. As shown in Section 3, the sizes of the quasi-dual MSC of the wave function are equal to $(\pi/\omega_x, \pi/\omega_x)$. We convert the size field to the angular speed by $\omega_x = \pi/\mu$ and $\omega_y = \pi/\eta$.

Our goal is to compute the wave values on the mesh vertices. We first explore the relationship between the wave values of the neighboring mesh vertices. Consider an edge $e_{ij} =< p_i, p_j >$ on the mesh. The local coordinate frame around edge e_{ij} is $(\vec{X}[e_{ij}], \vec{Y}[e_{ij}])$. Let the angle parameters of p_i and p_j be (θ_i, ϕ_i) and (θ_j, ϕ_j), respectively. Then, we have the following first order approximations by the relationship between the angle parameters and surface parameters:

$$\begin{cases} \theta_j - \theta_i \approx \omega_x \vec{X}[e_{ij}] \cdot (p_j - p_i) \\ \phi_j - \phi_i \approx \omega_y \vec{Y}[e_{ij}] \cdot (p_j - p_i) \end{cases} \quad (15)$$

Please note that the above approximation holds only when the vector field of edge e_{ij} is aligned with the local coordinate frames of p_i and p_j. Similar to the vector field alignment discussed in Section 4.1, we can permutate the angles θ and ϕ according to the alignment configurations, so that Equation 15 will hold. For clarity, we will assume that the vector field is well aligned with the local coordinate frames in the following description without loss of generality.

Denote $\alpha_{ij} = \theta_j - \theta_i$ and $\beta_{ij} = \phi_j - \phi_i$. Then the wave value at p_j can be approximated by

$$f(p_j) \approx \cos(\theta_i + \alpha_{ij})\cos(\phi_i + \beta_{ij}) = \mathbf{c}_{ij} \cdot \mathbf{f}_i, \quad (16)$$

where

$$\mathbf{c}_{ij} = \begin{pmatrix} \cos(\alpha_{ij})\cos(\beta_{ij}) \\ -\cos(\alpha_{ij})\sin(\beta_{ij}) \\ -\sin(\alpha_{ij})\cos(\beta_{ij}) \\ \sin(\alpha_{ij})\sin(\beta_{ij}) \end{pmatrix}, \ \mathbf{f}_i = \begin{pmatrix} \cos(\theta_i)\cos(\phi_i) \\ \cos(\theta_i)\sin(\phi_i) \\ \sin(\theta_i)\cos(\phi_i) \\ \sin(\theta_i)\sin(\phi_i) \end{pmatrix}.$$

Note that $\mathbf{c}_{ij}$ consists of known values that can be directly computed using the input mesh data and the vector field, while $\mathbf{f}_i$ consists of unknown variables including the wave value $\cos(\theta_i)\cos(\phi_i)$. Our wave construction method will solve for $\mathbf{f}_1, \ldots, \mathbf{f}_n$.

For convenience, we denote $cc_i = \cos(\theta_i)\cos(\phi_i)$, $cs_i = \cos(\theta_i)\sin(\phi_i)$, $sc_i = \sin(\theta_i)\cos(\phi_i)$, and $ss_i = \sin(\theta_i)\sin(\phi_i)$. It is obvious that the four components in $\mathbf{f}_i$ are dependent variables satisfying:

$$cc_i^2 + cs_i^2 + sc_i^2 + ss_i^2 = 1 \quad (17)$$

$$cc_i ss_i - cs_i sc_i = 0 \quad (18)$$

Let $\{\mathbf{e}_i\}_{i=1}^4$ be unit 4D column vectors whose i-th component is 1, and $\mathbf{H} = \mathbf{e}_1\mathbf{e}_4^t - \mathbf{e}_2\mathbf{e}_3^t$ be a 4×4 matrix. Then we can rewrite Equations (16, 17 and 18) in vector-matrix forms: $\mathbf{e}_1 \cdot \mathbf{f}_j - \mathbf{c}_{ij} \cdot \mathbf{f}_i = 0$, $\mathbf{f}_i \cdot \mathbf{f}_i - 1 = 0$, and $\mathbf{f}_i^t\mathbf{H}\mathbf{f}_i = 0$. Similarly, we have $\mathbf{e}_1 \cdot \mathbf{f}_i - \mathbf{c}_{ji} \cdot \mathbf{f}_j = 0$ by swapping p_i and p_j in above approximations.

Taking all edges into account, we finally formulate the following energy functional for constructing the wave:

$$\begin{aligned} E_{wave}(\mathbf{f}_1, \ldots, \mathbf{f}_n) = \sum_{e_{ij}\in K} \begin{pmatrix} (\mathbf{e}_1 \cdot \mathbf{f}_j - \mathbf{c}_{ij} \cdot \mathbf{f}_i)^2 \\ +(\mathbf{e}_1 \cdot \mathbf{f}_i - \mathbf{c}_{ji} \cdot \mathbf{f}_j)^2 \end{pmatrix} \\ + w_1 \sum_{p_i \in P}(\mathbf{f}_i \cdot \mathbf{f}_i - 1)^2 + w_2 \sum_{p_i \in P}(\mathbf{f}_i^t\mathbf{H}\mathbf{f}_i)^2, \end{aligned} \quad (19)$$

where w_1 and w_2 are two weighting coefficients. We empirically set $w_1 = w_2 = 0.15$ in our experiments.

The above minimization problem is nonlinear with $4n$ variables (n is the number of mesh vertices). We use Gaussian-Newton iteration method to solve it. We set the initial value to be all 0 except for a randomly picked vertex whose $\mathbf{f}$ vector's initial value is set to be $(1\ 0\ 0\ 0)$. The result of the above minimization is a 4D vector $\mathbf{f}$ for each vertex, but only the first component is the wave function that we want to obtain.

Feature Line Constraint To quadrangulate a model with sharp features, such as the CAD model in Figure 1, the quad edges are often required to be aligned with the feature lines on the model. Since the quad edges are mostly determined by the edges of the quasi-dual MSC, we need to impose some constraints to the wave function to align the quasi-dual MSC's edges with the feature lines.

As shown in Section 3, the edges of the quasi-dual MSC are the iso-parameter lines of $\theta = k\pi$ or $\phi = k\pi$. So, a vertex p_i lies at an edge of the quasi-dual MSC if and only if $\sin(\theta_i) = 0$ or $\sin(\phi_i) = 0$, i.e.,

$$ss_i = 0. \tag{20}$$

Therefore, we add the above hard constraints for vertices on the feature lines.

If a vertex p_i is the intersection of two feature lines, then the wave function must take extreme value at p_i. Such vertex is usually the corner of the model, which should also be the corner of the MSC chart. Therefore, we add the following three hard constraints:

$$cs_i = sc_i = ss_i = 0. \tag{21}$$

6 Parameterization and Quadrangulation

Once the wave is constructed, we adopt the same method in [Dong et al. 2006] to extract the quasi-dual MSC, which generates a set of quadrangular charts, $\{U_i\}_{i=1}^{m}$. These charts are then optimized iteratively by a globally smooth parameterization method and a node relocation method. Dong et al. [2006] mapped each chart to a unit square. To measure the parameterization smoothness between any two neighboring charts U_α and U_β, they defined a transition function $\phi_{\beta\alpha}$ to transform the coordinates from patch U_β to U_α by a rotation and a translation:

$$\phi_{\beta\alpha} = \mathbf{R}_{90}^{r_{\beta\alpha}} + T_{\beta\alpha}, \tag{22}$$

where the rotation time $r_{\beta\alpha}$ and the translation vector $T_{\beta\alpha}$ can be fully determined from the relative layout of the chart U_α and U_β.

There is no problem in the above isometric transition function when the charts are all expected to have the same isotropic size. But it will cause undesired distortion when we apply it to parameterize charts with different anisotropic sizes in our method. To address this problem, we extend the transition function to include anisotropic scalings. Specifically, we map each chart U_i to a quad domain: $U_i \mapsto [w_i, h_i]$, and define a new transition function as follows:

$$\phi_{\beta\alpha}^{new} = \mathbf{S}(w_\alpha, h_\alpha)\phi_{\beta\alpha}\mathbf{S}^{-1}(w_\beta, h_\beta), \tag{23}$$

where $\mathbf{S}(\cdot,\cdot)$ represents the 2D scaling transformation. The domain sizes of chart U_i, w_i and h_i, are determined by computing the average of the size field in U_i.

We then adopt the iterative relaxation strategy in [Dong et al. 2006] to optimize the charts, except that the transition function is replaced by the new one defined in Equation 23. To accurately keep the feature lines during optimization, for any feature line L on the chart's MSC boundary, the parameter coordinates of the vertices on L are forced to be on the corresponding parametric boundary.

Finally we generate the quads by uniformly subdivide all of the charts using a user specified subdivision parameter.

7 Results

We have implemented the proposed algorithms and tested them on a number of models. Figure 1 shows a set of quadrangulation results, and Table 1 lists the computation time to generate them. The timing is carried out on a PC with a dual-core CPU of 2.8 GHz and 2 GB RAM.

Mesh	N_v	N_c	t_{wave}	t_{msc}	t_{para}
Rockram (Figure 1)	37K	780	42.1	0.9	23.5
Fandisk (Figure 7)	25K	319	28.4	0.3	12.1
Fertility (Figure 9)	55K	1156	59.6	1.4	34.2
Buddha (Figure 10)	61K	4394	68.9	4.5	85.1
CAD (Figure 11)	136K	5915	121.6	6.8	459.8

Table 1: *List of computation time in second. N_v is the vertex number, N_c is the MSC chart number, and $t_{wave}/t_{msc}/t_{para}$ are the computation time of wave construction/MSC extraction/ parameterization.*

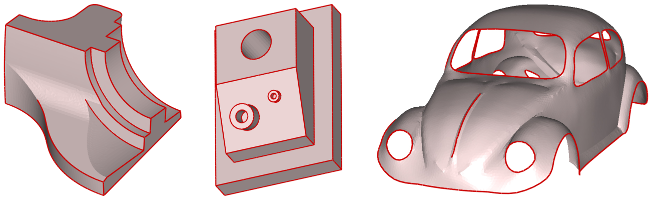


Figure 4: *Some models with their feature lines overdrawn in red.*

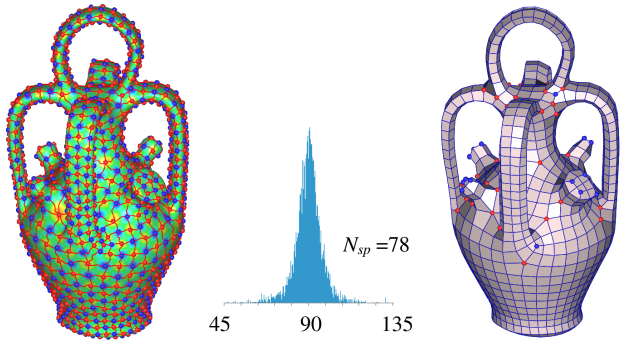


Figure 5: *Quadrangulation result of a model with many handles. The histogram in the middle shows the distribution of the quad angles. N_{sp} denotes the number of singularity points.*

In the following we will presents more results shown in Figures 4∼11. In each quadrangulation result, we visualize the wave function and the corresponding quasi-dual MSC on the input mesh. The red spots represent the maximum of the wave function, and the blue spots represent the minimum. The red curves are the edges of the quasi-dual MSC. We can see that the quasi-dual MSCs provide a good quadrilateral partition of the surface. The angle distribution of the quads and the number of the singularity points (N_{sp}) demonstrate the quality of the quadrangulation results.

Feature Alignment Figures 6, 7 and 8 show some quadrangulation results with feature line constraints. The feature lines and the models are shown in Figure 4.

For the CAD models in Figure 4 (left) and (middle), we automatically detect the feature lines by comparing the normal angle of

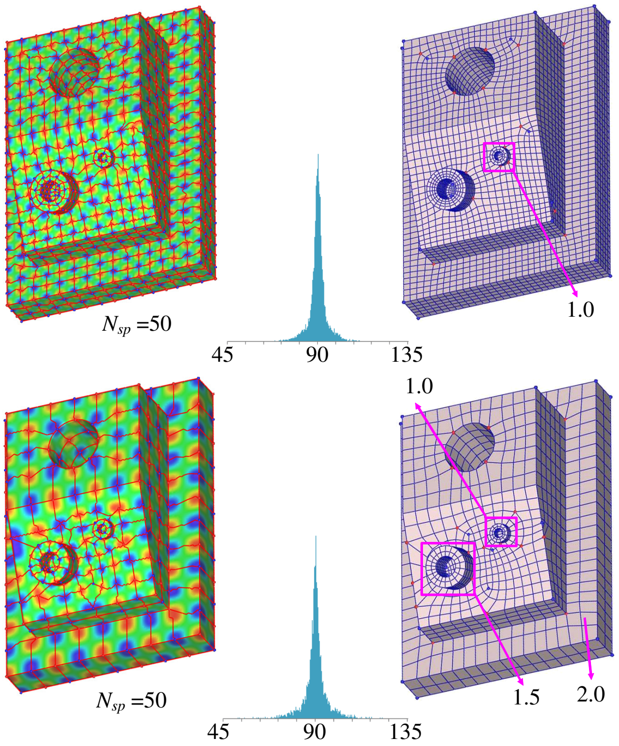


Figure 6: *Quadrangulation results with feature line constraints and size controls. The feature lines are shown in Figure 4. the size values specified by the user are labeled in the right column.*

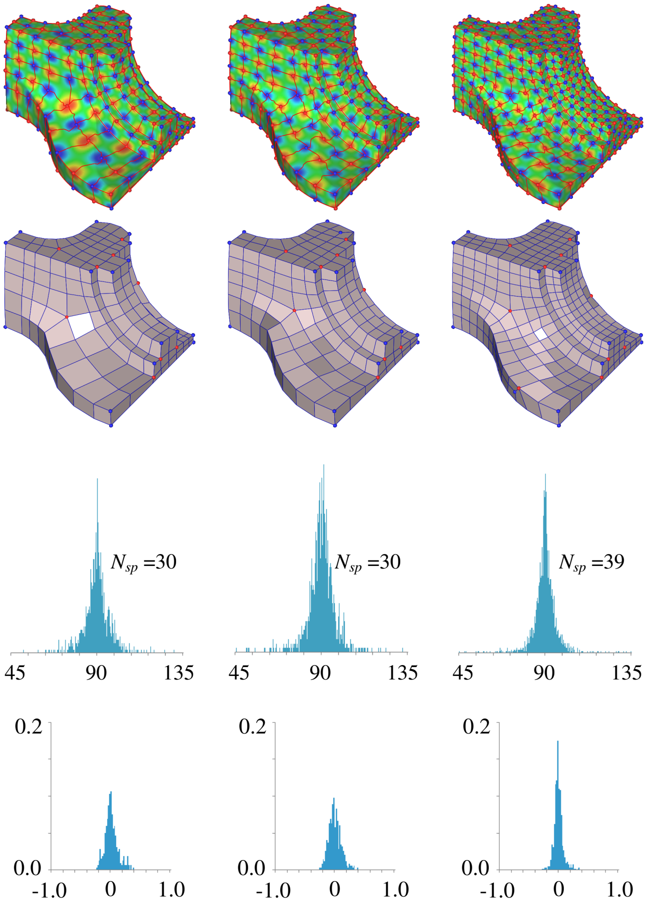
neighboring triangles with a prescribed threshold, say 45°. The corresponding quadrangulation results in Figures 6 and 7 show that the feature line constraint works successfully.

By treating the boundaries as feature lines, our method can easily deal with opened meshes, such as the car panel in Figure 4 (right). Since feature line constraint is hard constraint, the quad edges are well aligned with the boundaries, as shown in Figure 8.

Size Control Figure 6 demonstrates the application of size constraints. In the top row, we specified only one small size value to take care of the smallest feature of the CAD part, yielding a nearly uniform quadrangulation. In the bottom row, we manually specified three size values according to the local feature size, as labeled in the figure. As we expected, the region with small size field is finely remeshed, and vice versa. This example also shows a satisfactory transition between the three parts.

Anisotropic Control Figures 7 and 9 demonstrate the application of isotropic and anisotropic constraints. In Figure 7 we set the ratio to be 1 and change the weighting coefficient of the isotropic constraint between the columns. It is easy to see that, as the weight increases, the quads become more and more isotropic. And in this procedure, the singularity points are automatically re-distributed to achieve the optimal results.

In order to measure how the final quad edge lengths fit the target size ratio prescribed by the size field, we compute $(A_q - B_q)/B_q$ for each quad as the fitting error, where A_q is the edge length ratio of the quad, and B_q is the target size ratio (averaged over the corresponding quad). The bottom row of Figure 7 gives the distributions of the fitting errors, where the vertical axis represents the quad number divided by the total.



(a) $w_a = 0.4$ (b) $w_a = 0.8$ (c) $w_a = 1.5$

Figure 7: *Quadrangulation results with feature line constraints and isotropic constraints. The number w_a is the isotropic weight for results in the corresponding column.*

In Figure 9 (top row), we compute the principal curvature of the mesh, and use its inverse as the ratio of the anisotropic constraint (γ in Equation 13). In Figure 9 (middle row), we set the ratio to be $\gamma = 1$, achieving the effect of isotropic constraint. The fitting errors of the anisotropic constraint and the isotropic constraint are shown in the bottom row from left to right.

Complex Shapes We tested our method on some large and complex shapes. Figure 10 shows a result on the Buddha model. This model contains many handles of various sizes, including a very small one on the left part of the waist. And there are many folds and curling on the cloth. The shape complexity brings challenges for the previous methods. The result in Figure 11 shows that our method can deal with a lot of feature line constraints.

Comparison In our wave construction method, it is possible to optimize $\cos(\theta)$, $\sin(\theta)$, $\cos(\phi)$ and $\sin(\phi)$ as Ray et al. did [2006], then multiply $\cos(\theta)$ and $\cos(\phi)$ to obtain the final wave function. This indirect method has almost the same computation cost as ours. However, the wave function obtained by it usually gives greater residual energy in Equation 19 than ours, which means that such wave function is sub-optimal.

Figure 12 shows a comparison between the indirect method and our method. One can see that the quadrangulation result by the indi-

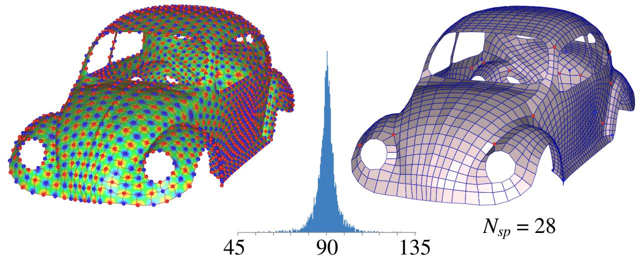


Figure 8: *Quadrangulation result of an opened mesh with many boundaries Figure 4 (right).*

rect method is quite good in most places, except in the bottom part where the orientation of some quad are distorted. This artifacts may be explained as follows. Recall that $|\cos(\theta)|$=1 or $|\cos(\phi)| = 1$ at the chart boundaries. By directly optimizing the wave function $\cos(\theta)\cos(\phi)$, our approach implicitly places relatively more weighting at the MSC chart boundaries, and less weighting elsewhere. This is very desirable, because the chart boundaries are very important for our framework to extract the quasi-dual MSC. On the contrary, the indirect method implicitly places uniform weighting value, i.e. 1, everywhere, thus reduces the relative importance at the chart boundaries.

8 Conclusion and Discussion

We have presented an anisotropic quadrangulation method, and successfully applied it to many testing mesh models. The results showed that the proposed method is very flexible, and provides a lot of useful and intuitive constraints to meet the user requirements in feature line alignment and anisotropic/isotropic size control, while it dose not incur much user interactions. Thanks to the curl optimization procedure and the MSC extraction method, the number of the singularity points is minimized and the positions are automatically well distributed. It is also robust in handling complex models.

In our current implementation, the input mesh must be tessellated finely enough, such that the length of each mesh edge is much smaller than the corresponding size fields. Otherwise, the wave function will be under-sampled, and miss some extreme points. This problem can be addressed by locally subdividing the input mesh according to the size field.

To properly deal with small handles, the edge length of corresponding quad must be smaller than the handle size. Otherwise, the extreme points of the wave function may lie out of region of the handle, and the corresponding quasi-dual MSC will fail to correctly capture the topology of the handle. In the future, we would like to automatically detected such degenerated case by computing the Euler characteristic of the quasi-dual MSC, and locally reduce the edge length of the corresponding quads by adding appropriate size constraints.

Acknowledgments

Many thanks to Xiaofei He for proofreading, and the anonymous reviewers for their valuable comments. This work was partially supported by China 973 Program (No. 2009CB320801) and NSFC (No. 60970074 and 60703039).

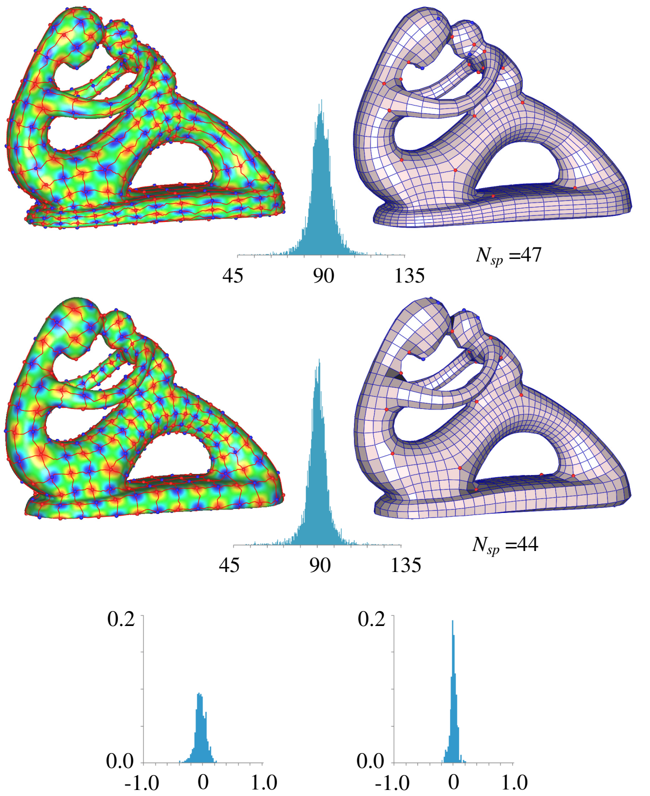


Figure 9: *Quadrangulation results using anisotropic (top) and isotropic (middle) size constraints.*

References

ALLIEZ, P., COHEN-STEINER, D., DEVILLERS, O., LÉVY, B., AND DESBRUN, M. 2003. Anisotropic polygonal remeshing. *ACM Transactions on Graphics 22*, 3, 485–493.

ALLIEZ, P., UCELLI, G., GOTSMAN, C., AND ATTENE, M. 2005. Recent advances in remeshing of surfaces. Research report, AIM@SHAPE Network of Excellence.

BEN-CHEN, M., GOTSMAN, C., AND BUNIN, G. 2008. Conformal flattening by curvature prescription and metric scaling. *Computer Graphics Forum 27*, 2, 449–458.

BLACKSTOCK, D. T. 2000. *Fundamentals Of Physical Acoustics*. Wiley-interscience.

BOIER-MARTIN, I., RUSHMEIER, H., AND JIN, J. 2004. Parameterization of triangle meshes over quadrilateral domains. In *Eurographics Symposium on Geometry Processing*, ACM, New York, NY, USA, 193–203.

BOMMES, D., ZIMMER, H., AND KOBBELT, L. 2009. Mixed-integer quadrangulation. *ACM Transactions on Graphics 28*, 3, 1–10.

BREMER, P.-T., EDELSBRUNNER, H., HAMANN, B., AND PASCUCCI, V. 2004. A topological hierarchy for functions on triangulated surfaces. *IEEE Transactions on Visualization and Computer Graphics 10*, 385–396.

COHEN-STEINER, D., AND MORVAN, J.-M. 2003. Restricted delaunay triangulations and normal cycle. In *Symposium on Computational geometry*, 312–321.

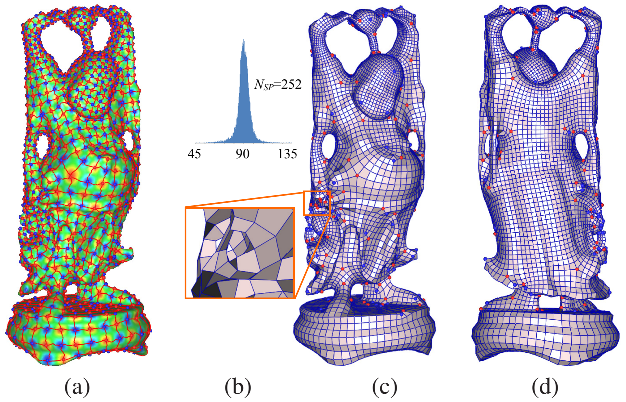


Figure 10: *Quadrangulation result of the Buddha model. The orange frame shows a zoomed view a very small ring shape.*

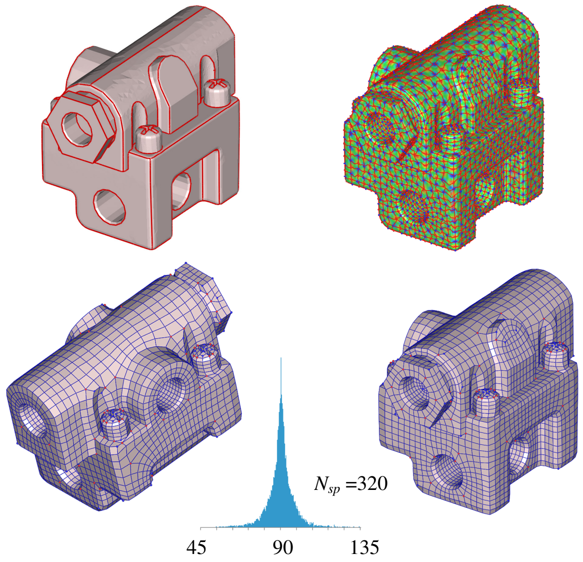


Figure 11: *Quadrangulation result of a complex CAD model with many feature line constraints.*

DONG, S., KIRCHER, S., AND GARLAND, M. 2005. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Comput. Aided Geom. Des. 22*, 5, 392–423.

DONG, S., BREMER, P.-T., GARLAND, M., PASCUCCI, V., AND HART, J. C. 2006. Spectral surface quadrangulation. *ACM Transactions on Graphics 25*, 3, 1057–1066.

EDELSBRUNNER, H., HARER, J., AND ZOMORODIAN, A. 2001. Hierarchical morse complexes for piecewise linear 2-manifolds. In *Symposium on Computational geometry*, ACM, New York, NY, USA, 70–79.

HORMANN, K., POLTHIER, K., AND SHEFFER, A. 2008. Mesh parameterization: theory and practice. In *SIGGRAPH Asia courses*, ACM, New York, NY, USA, 1–87.

HUANG, J., ZHANG, M., MA, J., LIU, X., KOBBELT, L., AND BAO, H. 2008. Spectral quadrangulation with orientation and alignment control. *ACM Transactions on Graphics 27*, 5, 1–9.

KÄLBERER, F., NIESER, M., , AND POLTHIER, K. 2007. Quadcover - surface parameterization using branched coverings. *Computer Graphics Forum*.

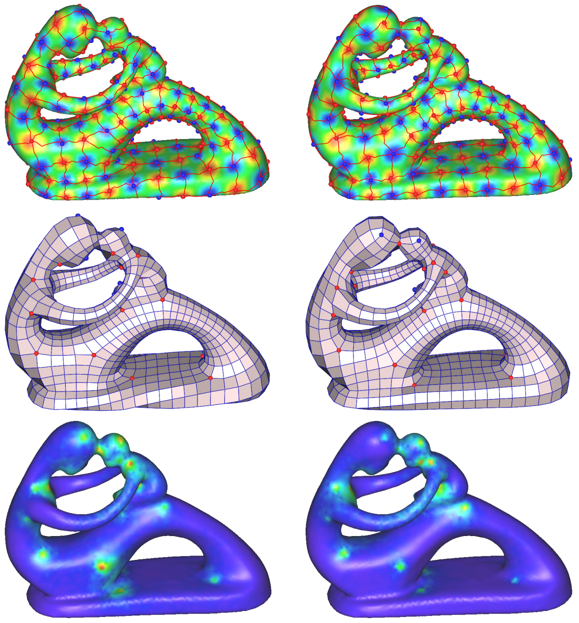


Figure 12: *Comparison between the indirect method using PGP variables (Left) and our method (Right). The bottom row shows the residual energy, where red color indicates large value.*

LAI, Y.-K., KOBBELT, L., AND HU, S.-M. 2008. An incremental approach to feature aligned quad dominant remeshing. In *ACM symposium on Solid and physical modeling*, ACM, New York, NY, USA, 137–145.

MARINOV, M., AND KOBBELT, L. 2006. A robust two-step procedure for quad-dominant remeshing. *Computer Graphics Forum 25*, 3 (Sept.), 537–546.

MEYER, M., DESBRUN, M., SCHRÖDER, P., AND BARR, A. H. 2002. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*, H.-C. Hege and K. Polthier, Eds. Springer-Verlag, 35–57.

PALACIOS, J., AND ZHANG, E. 2007. Rotational symmetry field design on surfaces. *ACM Transactions on Graphics 26*, 3, 55.

POLTHIER, K., AND PREUSS, E. 2003. Identifying vector field singularities using a discrete hodge decomposition. In *Visualization and Mathematics III*, H.-C. Hege and K. Polthier, Eds. Springer Verlag, Heidelberg, Germany, 113–134.

RAY, N., LI, W. C., LÉVY, B., SHEFFER, A., AND ALLIEZ, P. 2006. Periodic global parameterization. *ACM Transactions on Graphics 25*, 4, 1460–1485.

RAY, N., VALLET, B., LI, W. C., AND LÉVY, B. 2008. N-symmetry direction field design. *ACM Transactions on Graphics 27*, 2, 1–13.

TONG, Y., ALLIEZ, P., COHEN-STEINER, D., AND DESBRUN, M. 2006. Designing quadrangulations with discrete harmonic forms. In *Eurographics Symposium on Geometry Processing*, Eurographics Association, Aire-la-Ville, Switzerland, 201–210.

ZHANG, E., MISCHAIKOW, K., AND TURK, G. 2006. Vector field design on surfaces. *ACM Transactions on Graphics 25*, 4, 1294–1326.