Eurographics Symposium on Geometry Processing 2013
Yaron Lipman and Richard Hao Zhang
(Guest Editors)

*Volume 32 (2013), Number 5*

# Locally Injective Mappings

Christian Schüller[1] Ladislav Kavan[2] Daniele Panozzo[1] Olga Sorkine-Hornung[1]

[1]ETH Zurich, Switzerland
[2]University of Pennsylvania, USA

## 1 Additional Results

**Green's strain parametrization.** With our method, Green's strain energy can also be used to compute high-quality parameterizations (Figure 1). Without barriers, this produces a parameterization that is very close to an isometry, but with many flipped triangles. With our method, we obtain a parameterization that is similar to the one computed with ARAP in Figure 13 of the paper. Green's strain energy produces a parametrization with a slightly higher average conformal distortion of 5.38 compared to 4.23 of ARAP with barriers and 2.80 for ARAP without barriers. The number of iterations increases to 417 and the computation time is 75.2 seconds.

## 2 Deformation Energies

In this section we review the basics of deformation modeling, more details can be found in the literature [NMK*06]. We assume the input shapes and our functions are discretized using a piecewise linear representation, made of triangles for planar maps and tetrahedra for volumetric maps. In both cases we will denote the positions of the input mesh vertices (rest pose) as an $n \cdot d$ column vector $\mathbf{v}_0$, where $n$ is the number of vertices and $d$ the number of dimensions of the Euclidean space where the mesh is embedded (we consider 2 and 3 dimensions); the deformed vertex positions are denoted as $\mathbf{v}$. We do not address the deformation of surfaces in 3D, but rather volumetric shapes, because the notion of triangle inversion does not exist in 3D.

We use the word *element* to denote either triangle (if $d = 2$) or tetrahedron (if $d = 3$). A basic tool to study deformation of the $j$th element is to look at its displacement, encoded by the *deformation gradient*, which is a matrix $\mathbf{F}_j \in \mathbb{R}^{d\times d}$ mapping rest-pose vertices of this element to their deformed pose. Each deformation gradient is a linear function of $\mathbf{v}$, whose coefficients depend solely on $\mathbf{v}_0$. If $\mathbf{v} = \mathbf{v}_0$ (no deformation), each deformation gradient will be the identity matrix $\mathbf{I} \in \mathbb{R}^{d\times d}$. The simplest deformation measure is the Dirichlet energy (of the displacements):

$$E_D = \sum_{j\in\mathcal{E}} \lambda_j(\mathbf{v}_0)\|\mathbf{F}_j - \mathbf{I}\|_F^2 \quad (1)$$

where $\mathcal{E}$ is the set of all elements and $\|\cdot\|_F$ denotes Frobenius matrix norm. The function $\lambda_j : \mathbb{R}^{dn} \to \mathbb{R}^+$ is the Lebesgue measure of the $j$-th element, i.e., area in 2D and volume in 3D. It can be shown [BS08] that

$$E_D = (\mathbf{v}-\mathbf{v}_0)^T\mathbf{L}(\mathbf{v}-\mathbf{v}_0) \quad (2)$$

where $\mathbf{L} \in \mathbb{R}^{dn\times dn}$ is the standard linear finite element stiffness matrix (the cotan matrix for $d = 2$), extended to $dn \times dn$ by the Kronecker product with the $d \times d$ identity matrix. This energy is known as the Dirichlet energy of the displacements.

A somewhat similar but higher-order energy is the biharmonic, or Laplacian energy $E_L$. The biharmonic energy, that is used in the simple Laplacian editing formulation [SCOL*04, BS08], is defined as

$$E_L = (\mathbf{v}-\mathbf{v}_0)^T\mathbf{L}\mathbf{M}^{-1}\mathbf{L}(\mathbf{v}-\mathbf{v}_0) \quad (3)$$

where $\mathbf{M}$ the mass matrix of our mesh, again extended by Kronecker product with the identity.

These energies are invariant to translation of $\mathbf{v}$ but penalize global rotation, which limits their applicability only to small deformations. One way to achieve rotation invariance is by using a nonlinear strain measure, such as Green's strain. The simplest energy obtained using this recipe is

$$E_{GS} = \sum_{j\in\mathcal{E}} \lambda_j(\mathbf{v}_0)\|\mathbf{F}_j^T\mathbf{F}_j - \mathbf{I}\|_F^2. \quad (4)$$

Unlike the $E_D$ and $E_L$, $E_{GS}$ is indeed invariant to rotations: if we take a decomposition $\mathbf{F}_j = \mathbf{R}_j\mathbf{S}_j$, where $\mathbf{R}_j \in SO(d)$, we can see that $\mathbf{F}_j^T\mathbf{F}_j = \mathbf{S}_j^T\mathbf{S}_j$, because $\mathbf{R}_j^T\mathbf{R}_j = \mathbf{I}$. Unfortunately, $E_{GS}$ is also invariant to reflections, because the sign of $\det(\mathbf{S}_j)$ cancels out in the $\mathbf{S}_j^T\mathbf{S}_j$ term. This means that $E_{GS}$ does not penalize inverted elements at all. However, our method makes this limitation moot because it forbids inversion.

Several works proposed a rotation-invariant energy that penalizes inverted elements [ITF04, SA07, LZX*08, CPSS10], whose basic form is

$$E_{ARAP} = \sum_{j\in\mathcal{E}} \lambda_j(\mathbf{v}_0)\|\mathbf{F}_j - \mathbf{R}_j\|_F^2 \quad (5)$$

where $\mathbf{F}_j = \mathbf{R}_j\mathbf{S}_j$ as before and $\mathbf{R}_j \in SO(d)$ is the closest rotation to $\mathbf{F}_j$. Due to a property of the Frobenius norm, we

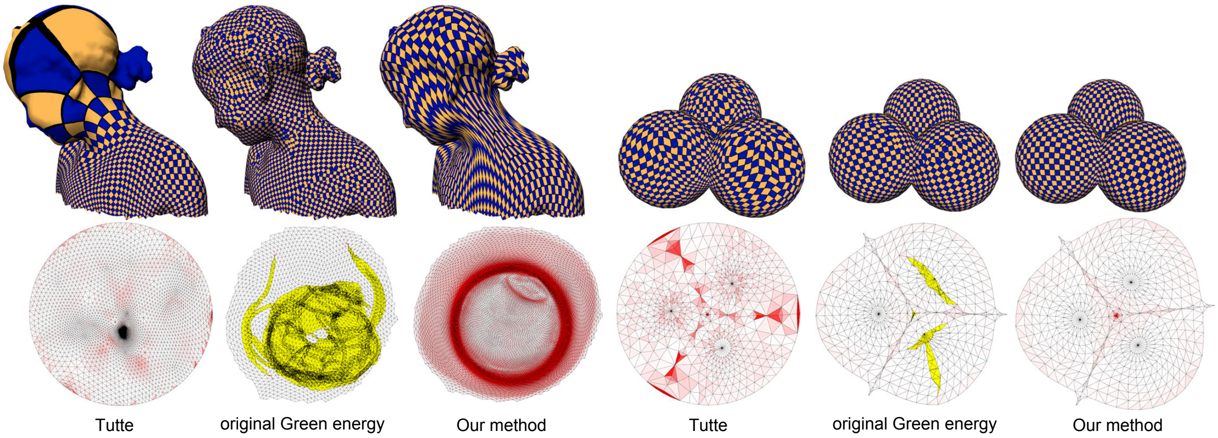


**Figure 1:** *Two examples of mesh parameterization using the Green's strain energy, with and without our barriers method. The initial guess is Tutte's mapping onto a circle. Although it is highly distorting, our method manages to converge to a reasonable solution while avoiding any element inversions. In contrast, the original Green energy without barriers easily leads to flips since it is insensitive to them.*

can rewrite

$$\|\mathbf{F}_j - \mathbf{R}_j\|_F^2 = \|\mathbf{R}_j\mathbf{S}_j - \mathbf{R}_j\|_F^2 = \|\mathbf{S}_j - \mathbf{I}\|_F^2. \qquad (6)$$

To verify the broad applicability of our method, we tested it with all of the above discussed deformation energies.

## 3 The gradient and Hessian of the barriers

Here we provide the gradient and Hessian of our barrier function $\phi_j(c_j(\mathbf{v}))$ for one element $j$. Recall $c_j(\mathbf{v}) = \lambda_j(\mathbf{v}) - \varepsilon$, where $\lambda_j(\mathbf{v})$ is the signed area or volume function of element $j$. Assuming the cubic spline $g_j$ (Equation 5 in the paper) and its derivatives are evaluated at $c_j(\mathbf{v})$, we can write:

$$\nabla\phi_j(\mathbf{v}) = -\frac{g'_j}{g_j^2} \cdot \nabla\lambda_j(\mathbf{v}),$$

where the gradient of the area/volume $\lambda_j(\mathbf{v})$ is the standard perpendicular construction for the vertices of the triangle/tetrahedron, and zero for the vertices not incident on the element. Following the same conventions, for the Hessian we obtain:

$$\nabla^2\phi_j(\mathbf{v}) = \frac{2(g'_j)^2 - g''_j g_j}{g_j^3} \cdot \nabla\lambda_j(\mathbf{v}) \cdot \nabla\lambda_j(\mathbf{v})^T - \frac{g'_j}{g_j^2} \cdot \nabla^2\lambda_j(\mathbf{v}).$$

## References

[BS08] Botsch M., Sorkine O.: On linear variational surface deformation methods. *IEEE Trans. Vis. Comput. Graph. 14*, 1 (2008). 1

[CPSS10] Chao I., Pinkall U., Sanan P., Schröder P.: A simple geometric model for elastic deformations. *ACM Trans. Graph. 29*, 4 (2010). 1

[ITF04] Irving G., Teran J., Fedkiw R.: Invertible finite elements for robust simulation of large deformation. In *Proc. SCA* (2004), pp. 131–140. 1

[LZX*08] Liu L., Zhang L., Xu Y., Gotsman C., Gortler S. J.: A local/global approach to mesh parameterization. *Comput. Graph. Forum 27*, 5 (2008). 1

[NMK*06] Nealen A., Müller M., Keiser R., Boxerman E., Carlson M.: Physically based deformable models in computer graphics. *Comput. Graph. Forum 25*, 4 (2006). 1

[SA07] Sorkine O., Alexa M.: As-rigid-as-possible surface modeling. In *Proc. Symposium on Geometry Processing* (2007). 1

[SCOL*04] Sorkine O., Cohen-Or D., Lipman Y., Alexa M., Rössl C., Seidel H.-P.: Laplacian surface editing. In *Proc. Symposium on Geometry Processing* (2004), pp. 175–184. 1