

Conformal self-organizing map on curved seamless surface

Cheng-Yuan Liou*, Yen-Ting Kuo, Jau-Chi Huang

Department of Computer Science and Information Engineering, National Taiwan University, Taiwan, ROC

ARTICLE INFO

Available online 24 June 2008

Keywords:

Self-organizing map
Conformal mapping
Manifold learning
Texture mapping
Multidimensional scaling
Morphing

ABSTRACT

This paper presents a new mapping to construct the self-organizing map on the curved seamless surface. This mapping is developed for the planar triangle surface derived from the conformal self-organizing map [C.-Y. Liou, Y.-T. Kuo, Conformal self-organizing map for a genus-zero manifold, *Visual Comput.* 21(5) (2005) 340–353]. It shows how to construct a seamless surface for the genus-zero manifold. The constructed surface is both seamless and continuous. The mapping between the model surface and the sphere surface is one-to-one and onto. This kind of surface can facilitate many applications of the self-organizing map. We show experiments in surface reconstructions and texture mappings.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

The goal of surface reconstruction is to obtain a continuous surface that can represent a cloud of pattern points [16]. These cloud patterns are usually obtained from 3D laser scanners and medical scanners. These patterns may also be collected by various vision techniques, such as correlated viewpoints, voxel carving, stereo range images. The conformal self-organizing map (CSM) [15,8,9] has been developed for modeling a continuous surface for the cloud patterns. It is a kind of self-organizing map with conformal contents [10,11]. The CSM is formed by a collection of connected simplices including points and edges. Since these triangles are flat, the surface constructed by these flat triangles is not a smoothly curved one. The flat triangle surface in 3D [8] may have ambiguous mapping for those sphere points which project on the triangle edges. To resolve the problem of ambiguous mapping and derive a smooth surface, we present a method to construct a curved seamless surface (CSS) with unique mapping correspondence.

The CSS is ideal for texture mapping because it contains the property of conformal mapping that preserves the local shapes of textures. The major advantage of the CSS over the NURBS¹ is that the CSS has the property of conformal mapping. The global conformal mapping (GCM) also derives conformal structures for general closed meshes [4] and isometric mapping [3]. It is fundamentally different from the CSS in two ways: first, the GCM is not designed for continuous surface reconstruction and

second, the GCM is not designed for smooth curved surfaces. We show experiments on the CSS and GCM in Section 4.

The rest of this paper is organized as follows. In the next section, we briefly introduce the CSM. In Section 3, we present the CSS that is further derived from a learned CSM surface. In Section 4, we show experiments on 3D point clouds. We discuss many issues on the CSS in the last section.

2. Review CSM

The CSM [10,8] attempts to accomplish conformal transformation between forms. This mapping is angle preserving. Different from other methods, the CSM is based on the self-organizing map (SOM) and has the potential to learn higher dimensional data. We will focus on the 3D objects of genus-zero surfaces in this paper. Since a sphere has the same genus-zero manifold, we use a sphere as the network space of neurons. A sphere can be approximated by icosahedrons at different resolutions [7]; hence the number of neurons varies. Each face of an icosahedron is an equilateral triangle. The basic type of icosahedron has 12 vertices, 30 edges, and 20 equivalent equilateral triangular faces. It is varied by combining more icosahedrons into a single body. We use the term f (frequency) to denote its multiplicity, R to denote the number of faces, and N to denote the number of vertices. The formula of the icosahedron is

$$\begin{aligned} R &= 20f^2, \\ N &= \frac{R}{2} + 2. \end{aligned} \quad (1)$$

In the CSM, the sampled 3D patterns are the training patterns and the mesh is configured by neurons. These neurons are the vertices of the mesh. Each neuron has two vectors, one is the

* Corresponding author.

E-mail address: cyliau@csie.ntu.edu.tw (C.-Y. Liou).

¹ NURBS stands for non-uniform rational B-spline and it is a widely used mathematical model for generating and representing curves and surfaces.

weight vector, w_i , in the pattern space and the other is the position vector, n_i , in the network space. The weight vectors contain the locations of the mesh vertices in the pattern space. The position vectors contain the locations of the neurons on the sphere surface. Fig. 1 shows the relationship of these two vectors. In the CSM, w_i is evolved to match its corresponding pattern and n_i is fixed to preserve the sphere topology. Let X denotes the set that contains all point patterns of a 3D object, $X = \{(x_l, y_l, z_l), l = 1, \dots, P\}$. Let the collection of all equilateral triangles of the icosahedron inside a unit sphere be $S^d = \{\Delta_n^r, r = 1, \dots, R\}$, illustrated in Fig. 1(b). Let the collection of all vertices, denoting neurons, be $\mathbb{N} = \{n_i, i = 1, \dots, N\}$, where n_i is a 3D column vector in the network space and denotes the position of the i th mesh vertex on the unit sphere surface. Each triangle is a mesh hole that can be represented by its three vertices, that is, $\Delta_n^r \equiv [n_i^r, n_j^r, n_k^r]$. The parameterization domain is the mesh S^d which is suitable for the genus-zero manifold.

The CSM constructs a mapping from S^d to M^d [8], where M^d is formed by the weight vectors, w_i , for approximating the sample points X of a 3D object. M^d is composed of a collection of flat triangles, $M^d = \{\Delta_w^r, r = 1, \dots, R\}$. Each equilateral triangle Δ_n^r on S^d in the network space is mapped to its corresponding triangle Δ_w^r on M^d in the pattern space. Each Δ_w^r is an irregular triangle that can be represented by its three vertices, that is, $\Delta_w^r = [w_i^r, w_j^r, w_k^r]$, where w_i^r is a 3D column weight vector in the pattern space and denotes the position of the i th mesh vertex in M^d , see Fig. 1(a). The vertex w_i^r is mapped to the vertex n_i^r . Both Δ_w^r and Δ_n^r are flat triangles.

The major difference between the CSM and SOM is in the mapping of winning neurons. The SOM maps each point x in the pattern space to a particular winning neuron. On the other hand, CSM maps x to a corresponding point r in Δ_n^r under a conformal mapping \mathcal{M} from Δ_w^r to Δ_n^r . This conformal mapping \mathcal{M} is illustrated in Figs. 2 and 3. The learning algorithm is briefly described below.

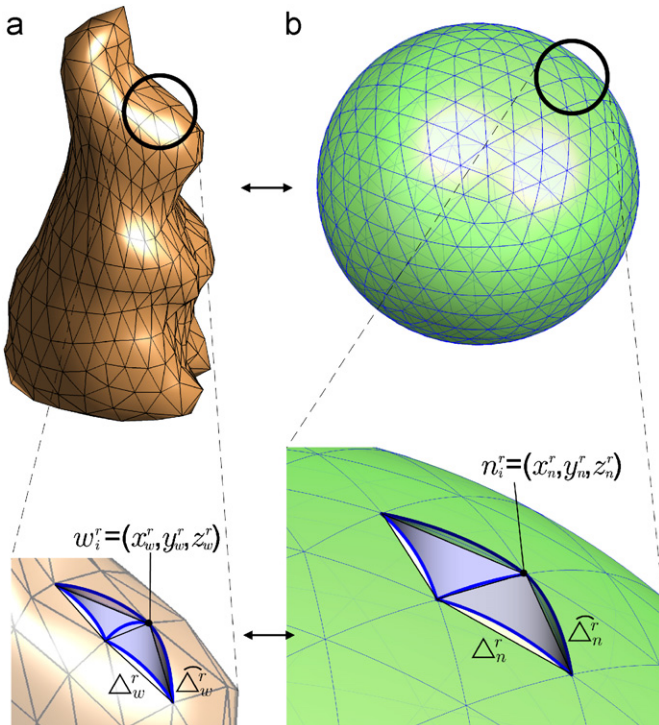


Fig. 1. Illustration of the mesh for a 3D object and the network space in the CSM. (a) The mesh of a 3D object. (b) The neurons' position vectors. The curved domes are shown with blue arcs.

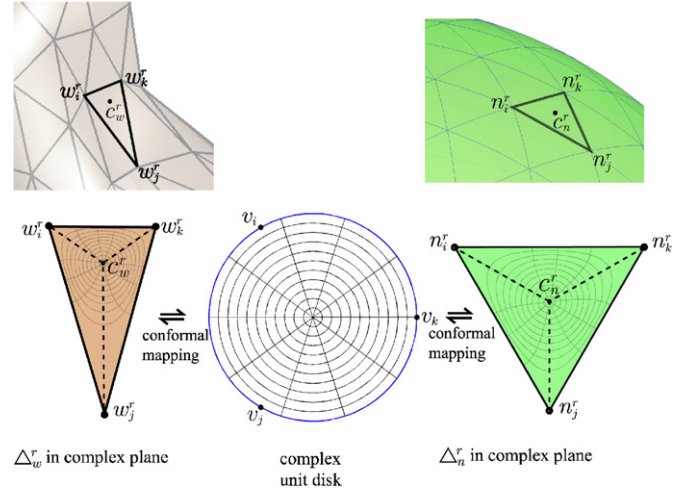


Fig. 2. Conformal mapping between the two triangles Δ_n^r and Δ_w^r . Each triangle in 3D is translated to the complex plane. Then their conformal mappings to a unit disk can be computed by the Schwarz–Christoffel method [12] to build the point correspondence.

CSM Learning Algorithm. *Input:* Input pattern X , sphere mesh S^d and its vertices \mathbb{N} .

Output: Model mesh M^d .

1. *Initialization.* Initialize the CSM network. The neurons' weight vectors are initialized as the same uniform vertices \mathbb{N} on a sphere, $w_i(t=0) = n_i$. Set learning constants. Let t denotes the learning time. We start the algorithm from $t = 0$.
2. *Sampling.* Randomly select an input point $x \in X$ with equal probability.
3. *Similarity matching.* Determine the winning or best-matching neuron by using

$$\|w_c - x\| = \min_i \|w_i(t) - x\|, \quad w_i(t) \in W(t), \quad (2)$$

where w_c is the weight vector of the winning neuron for the corresponding input x in time t . $W(t)$ is the set of all weight vectors. Then the projected simplex $s = \Delta_w^r$ is determined by checking all simplices connecting w_c . Finally, the reference vector $r = \mathcal{M}(x)$ is calculated by projection and conformal mapping. The function \mathcal{M} first projects pattern x onto the simplex, s , and then maps it to the network space using conformal mapping (Eqs. (4), (5)). Fig. 3 illustrates the transformation of input x into the reference vector r .

4. *Updating.* Update all weight vectors according to the following equation:

$$\begin{aligned} \delta w_i &= \alpha_t h(d(\mathcal{M}(x), n_i))(x - w_i(t)) \\ &= \alpha_t h(d(r, n_i))(x - w_i(t)), \\ w_i(t+1) &= w_i(t) + \delta w_i, \end{aligned} \quad (3)$$

where $\alpha_t \in [0, 1)$ is the learning rate at time t , and h is the neighborhood function which decreases monotonously with the distance metric d in the network space. This step is to improve the similarity of the weight vectors toward the pattern x .

5. *Continuation.* Continue with Step 2 until a satisfactory result is obtained.

In Step 4 of the learning algorithm, the function \mathcal{M} requires the use of conformal mapping to map simplex s in the input space to s'

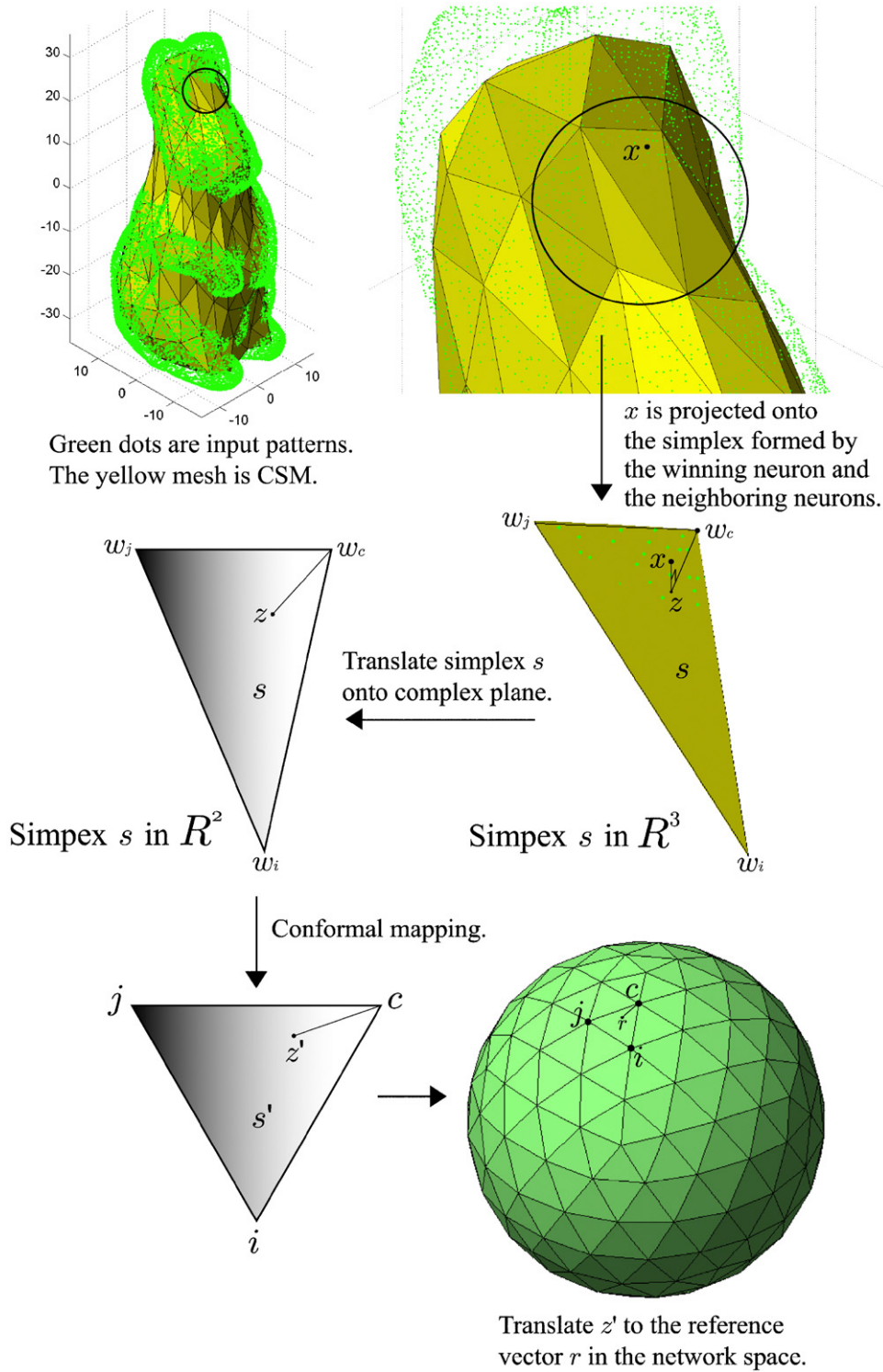


Fig. 3. The procedure for mapping input pattern x to the reference vector r in the network space.

in the network space, see Figs. 2 and 3. The conformal mapping from simplex s to equilateral simplex s' can be approximated by means of the Schwarz–Christoffel mapping [12], see Fig. 4.

In Fig. 4, the mapping function from the v -plane to the z -plane is given by

$$z = f_1(v) = a_1 + B_1 \int_0^v \frac{1}{\zeta^2} \prod_{i=1}^3 \left(1 - \frac{v_i}{\zeta}\right)^{-\beta_i} d\zeta. \quad (4)$$

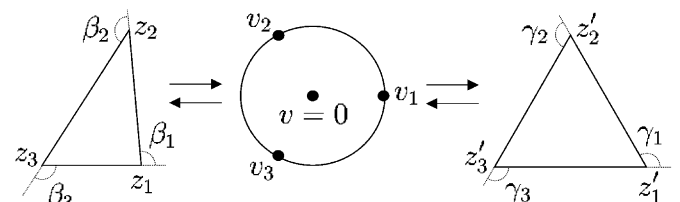


Fig. 4. The conformal mapping from an arbitrary triangle to a unit disk and then to an equilateral triangle and vice versa.

The mapping function from the v -plane to the z' -plane is given by

$$z' = f_2(v) = a_2 + B_2 \int_0^v \frac{1}{\zeta^2} \prod_{i=1}^3 \left(1 - \frac{v_i}{\zeta}\right)^{-\gamma_i} d\zeta. \quad (5)$$

Since β_i and γ_i are known, a, B and v_i have to be solved in the above equations. Therefore, the mapping from simplex s to simplex s' is $z' = f_2(f_1^{-1}(z))$, where z is any point on s , and z' is its corresponding point on s' . Then, the reference vector r is computed using $r = n_c + \text{proj}(z')$. Function $\text{proj}(z')$ projects z' in the complex plane onto the network space. Note that r may be normalized with the same magnitude as that of n_i , $|r| = |n_i| = 1$ in certain cases.

3. CSS parameterization

By using the CSM [8], we can derive the M^d and obtain the mapping between each triangle Δ_n^r and its corresponding triangle Δ_w^r . Let the portion of the sphere surface right above the flat triangle, Δ_n^r , be $\hat{\Delta}_n^r$, see Fig. 1(b). The triangular dome $\hat{\Delta}_n^r$ can be obtained by cutting the three arc curves, $\{n_i^r n_j^r, n_j^r n_k^r, n_i^r n_k^r\}$, on the sphere surface right above the three edges, $\{\bar{n}_i^r \bar{n}_j^r, \bar{n}_j^r \bar{n}_k^r, \bar{n}_i^r \bar{n}_k^r\}$, of the triangle Δ_n^r . Each arc point is the intersection of the sphere surface and the line that passes through the sphere center, $c = (0, 0, 0)^T$, and an edge point. The arc $\bar{n}_i^r \bar{n}_j^r$, edge $\bar{n}_i^r \bar{n}_j^r$, and center c are in the same plane. $\hat{\Delta}_n^r$ is the geodesic dome of Δ_n^r and has a triangular tent shape. The sphere surface, S , is the collection of every $\hat{\Delta}_n^r$, $S = \{\hat{\Delta}_n^r, r = 1, \dots, R\}$. Since the dome $\hat{\Delta}_n^r$ is beautiful, we want to borrow and fit (deform) the dome to construct a model surface for the cloud X .

The basic idea to map a sphere surface S to a seamless surface M is to deform each curved triangle $\hat{\Delta}_n^r$ on the sphere to a curved $\hat{\Delta}_w^r$ triangle on the model surface. Imagine that the surface of the sphere is like an elastic balloon. We deform, stretches or squeezes, every little pieces of the balloon so that this balloon has a new shape as that of a 3D object. In practice, we define this deformation mapping in a pointwise manner. This means that every point on $\hat{\Delta}_n^r$ has a corresponding point on $\hat{\Delta}_w^r$. Suppose there is a point p on the curved triangle $\hat{\Delta}_n^r$ of the sphere. The procedure of this mapping is stated as follows:

1. Map p from $\hat{\Delta}_n^r$ to p' on Δ_n^r . In other words, point p on the sphere is mapped to p' on the flat triangle derived by the CSM in the network space.
2. Conformally map p' from Δ_n^r to q' on Δ_w^r . In other words, the point p' on the spherical mesh, S^d , is mapped to point q' on the model mesh, M^d .
3. Finally map q' to point q on $\hat{\Delta}_w^r$. The point q is the corresponding point on the CSS of point p .

Because this mapping is pointwise, a collection of vertices \mathbb{N}^{new} of a sphere is prepared for each point p . Then, this procedure iterates on all $p \in \mathbb{N}^{\text{new}}$ until each corresponding point q is determined. The details of this mapping are presented in the following algorithm.

CSS Algorithm. *Input:* Vertices of a new dense mesh \mathbb{N}^{new} , sphere mesh S^d , model mesh M^d .

Output: Curved seamless surface M , $M = \{\hat{\Delta}_w^r, r = 1, \dots, R\}$.

1. For each triangle Δ_n^r , $\Delta_n^r \in S^d$, find the center c_n^r of the triangle Δ_n^r and its conformal mapping point, c_w^r , in Δ_w^r . c_n^r can be determined by this equation:

$$c_n^r = \frac{n_i^r + n_j^r + n_k^r}{3}, \quad (6)$$

and the conformal mapped point c_w^r is computed by Eqs. (4) and (5).

2. Split Δ_w^r , $\Delta_w^r \in M^d$, into three subtriangles, $\{[c_w^r, w_i^r, w_j^r], [c_w^r, w_j^r, w_k^r], [c_w^r, w_i^r, w_k^r]\}$, by using the three line sections, $\{c_w^r w_i^r, c_w^r w_j^r, c_w^r w_k^r\}$, and the three edges, $\{w_i^r w_j^r, w_j^r w_k^r, w_i^r w_k^r\}$.
3. Calculate the unit normal vector of the triangle plane Δ_w^r (Fig. 5):

$$\hat{n}_w^r = \frac{(w_j^r - w_i^r) \times (w_k^r - w_i^r)}{|(w_j^r - w_i^r) \times (w_k^r - w_i^r)|}. \quad (7)$$

4. Let the triangle next to Δ_w^r in M^d that shares the edge $w_i^r w_j^r$ is Δ_w^{r1} , that shares the edge $w_j^r w_k^r$ is Δ_w^{r2} , that shares the edge $w_i^r w_k^r$ is Δ_w^{r3} . Calculate the unit normal vectors of the three adjacent triangles, $\Delta_w^{r1}, \Delta_w^{r2}, \Delta_w^{r3}$, by using the same equation in the above step. Let the obtained normal vectors be $\hat{n}_w^{r1}, \hat{n}_w^{r2}$, and \hat{n}_w^{r3} , respectively (Fig. 5).

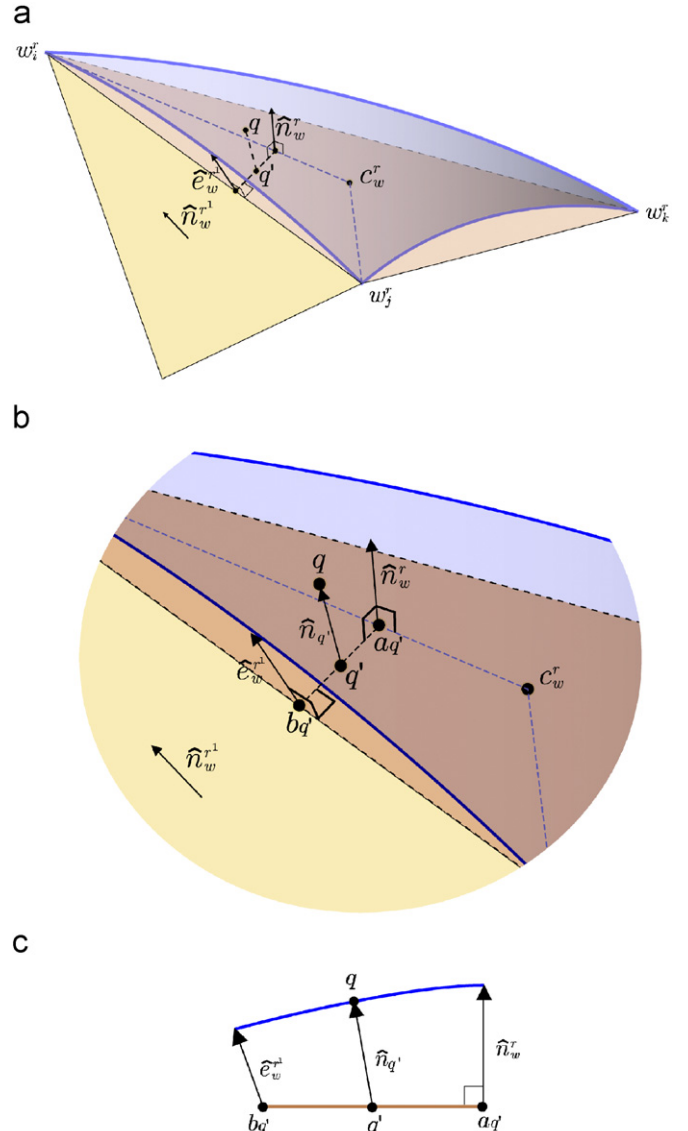


Fig. 5. Illustration of the relations between the points q' and q . (a) Mapped point q on $\hat{\Delta}_w^r$, (b) magnified plot and (c) profile.

5. Calculate the unit normal vectors of the edges, $\overline{w_i^r w_j^r}$, $\overline{w_j^r w_k^r}$, and $\overline{w_i^r w_k^r}$:

$$\hat{e}_w^{r1} = \frac{\hat{n}_w^r + \hat{n}_w^{r1}}{|\hat{n}_w^r + \hat{n}_w^{r1}|}, \quad \hat{e}_w^{r2} = \frac{\hat{n}_w^r + \hat{n}_w^{r2}}{|\hat{n}_w^r + \hat{n}_w^{r2}|},$$

and $\hat{e}_w^{r3} = \frac{\hat{n}_w^r + \hat{n}_w^{r3}}{|\hat{n}_w^r + \hat{n}_w^{r3}|}.$ (8)

Note that $\hat{e}_w^{r1} \perp \overline{w_i^r w_j^r}$, $\hat{e}_w^{r2} \perp \overline{w_j^r w_k^r}$, and $\hat{e}_w^{r3} \perp \overline{w_i^r w_k^r}$ (Fig. 5).

6. Select a point p on Δ_n^r , $p \in \mathbb{N}^{\text{new}}$ where \mathbb{N}^{new} is given as the input of this algorithm. Find its projection, p' , on the flat triangle Δ_n^r :

$$p' \equiv \overline{pc} \cap \Delta_n^r. \quad (9)$$

Here $c = (0, 0, 0)^T$ is the center of the unit sphere and p' is the intersection point of the line \overline{pc} and Δ_n^r , as shown in Fig. 6(a). Let $p' = (x', y', z')$ and $p = (x_p, y_p, z_p)$. Eq. (9) is solved by the following four simultaneous equations:

$$0 = \begin{vmatrix} x' & y' & z' & 1 \\ x_i^A & y_i^A & z_i^A & 1 \\ x_j^A & y_j^A & z_j^A & 1 \\ x_k^A & y_k^A & z_k^A & 1 \end{vmatrix},$$

$$x' = x_c + (x_p - x_c)t,$$

$$y' = y_c + (y_p - y_c)t,$$

$$z' = z_c + (z_p - z_c)t.$$

t can be determined by

$$t = \frac{\begin{vmatrix} 1 & 1 & 1 & 1 \\ x_i^A & x_j^A & x_k^A & x_p \\ y_i^A & y_j^A & y_k^A & y_p \\ z_i^A & z_j^A & z_k^A & z_p \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 1 & 0 \\ x_i^A & x_j^A & x_k^A & x_p - x_c \\ y_i^A & y_j^A & y_k^A & y_p - y_c \\ z_i^A & z_j^A & z_k^A & z_p - z_c \end{vmatrix}}.$$

Note that $c = (x_c, y_c, z_c)^T = (0, 0, 0)^T$, $\Delta_n^r \equiv [n_i^r, n_j^r, n_k^r]$, $(x_i^A, y_i^A, z_i^A)^T = n_i^r$, $(x_j^A, y_j^A, z_j^A)^T = n_j^r$, and $(x_k^A, y_k^A, z_k^A)^T = n_k^r$, see Fig. 6(a).

7. For the point p' , calculate its conformal mapping point q' in Δ_w^r ,

$$q' = \mathcal{M}_{\Delta_n^r}^{\Delta_w^r}(p'), \quad (10)$$

where $\mathcal{M}_{\Delta_n^r}^{\Delta_w^r}$ is the conformal mapping [8] from the flat triangle Δ_n^r to the flat triangle Δ_w^r , see Fig. 6(b). q' may fall in any one of the three subtriangles. Suppose q' is in the subtriangle $[c_w^r, w_i^r, w_j^r]$.

8. Calculate the projection point $b_{q'}$ of q' on the line section $\overline{w_i^r w_j^r}$. This means that $\overline{w_i^r w_j^r} \perp \overline{b_{q'} q'}$. Calculate the intersection point $a_{q'}$ of the line $\overline{b_{q'} q'}$ and one of the other two edges of the subtriangle $[c_w^r, w_i^r, w_j^r]$. The locations of $a_{q'}$ and $b_{q'}$ are shown in Fig. 5.

9. Calculate the unit direction $\hat{n}_{q'}$ at the point q' :

$$\hat{n}_{q'} = \frac{\vec{n}_{q'}}{|\vec{n}_{q'}|}, \quad \vec{n}_{q'} = \frac{|q' - a_{q'}|}{|b_{q'} - a_{q'}|}(\hat{e}_w^{r1} - \hat{n}_w^r) + \hat{n}_w^r. \quad (11)$$

Note that q' , $a_{q'}$, $b_{q'}$, and $\hat{n}_{q'}$ are in the same plane. The two vectors \hat{e}_w^{r1} and \hat{n}_w^r in Fig. 5 which pass the points $b_{q'}$ and $a_{q'}$ separately are also in this plane.

10. In this step we plan to determine a point $q \in M$, for the dome Δ_w^r that corresponds to p . q is obtained from the dome height $|p - p'|$ and the unit direction $\hat{n}_{q'}$. q can be obtained by

$$q = q' + |p - p'| \hat{n}_{q'}. \quad (12)$$

The whole construction is shown in Figs. 5 and 6.

11. Repeat Step 6 and select another point, $p \in \mathbb{N}^{\text{new}}$, until all points in \mathbb{N}^{new} are processed.

We operate this algorithm on all R triangles. We may map any number of surface points for each triangle dome. This mapping is bijection which maps from the triangular sphere surface Δ_n^r to the triangular geodesic model surface Δ_w^r . The surface border of Δ_w^r along the three boundary arc curves, $\{w_i^r w_j^r, w_j^r w_k^r, w_i^r w_k^r\}$, is perfectly seamless and continuous but may not be smooth. To our knowledge, this is the only seamless arc connecting two triangular domes.² In all experiments, the surface point p on Δ_n^r is a vertex of a new denser mesh \mathbb{N}^{new} , where $N^{\text{new}} > N$. \mathbb{N}^{new} may be an icosahedron with more vertices. \mathbb{N}^{new} may have dense vertices in an area with fine texture.

² Note that the widely used smooth surface such as B-spline and NURBS work with quadrilateral patches rather than triangular patches.

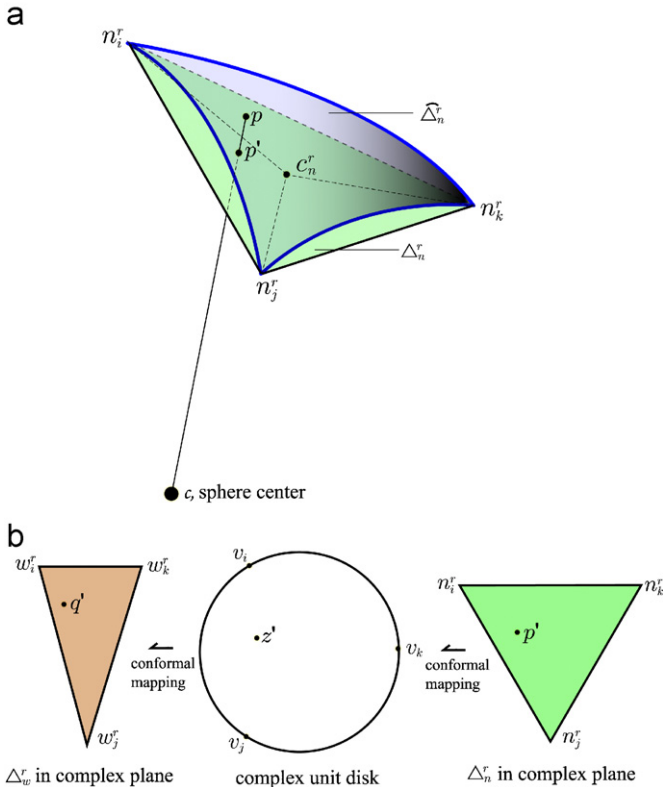


Fig. 6. Illustration of the relations among the points p , p' , q' and the sphere center c . (a) Sphere surface and (b) conformal mapping.

4. Implementation and results

The goal of the CSS construction is to accomplish a surface of a real object than a triangular mesh. To verify it, we prepare two sets of sampled points of a same object. One has fewer points, denoted as X_{few} , and the other has more points, denoted as X_{more} . Let CSM compute the conformal mapping using the fewer one. We then apply the CSS to improve it and obtain the model surface M . The performance is evaluated by the mean square error. The error is named as mismatch error, E_{mis} . It is

$$E_{\text{mis}} = \frac{1}{\#(X_{\text{more}})} \sum_{x \in X_{\text{more}}} \text{dist}(x, M)^2. \quad (13)$$

The distance from point x to surface M , $\text{dist}(x, M)$, is the projection distance from x to M .

In certain case, such as the bunny model has long extrusive parts, the ears, it may be difficult for the CSM to learn those concave ear shapes. An edge swap with multi-resolution learning [16] and a growing neural mesh [6] may be used to overcome this difficulty. Since they change the regular mesh connection, we will not use them in the CSM. We solve this difficult learning by giving the priority to those extrusive parts during the CSM learning. This priority can resolve those parts and keep the regular connection.

There are two models used in experiments including the Stanford bunny [14] and the Igea head [2]. The mesh size of these models is listed in Table 1.

In the CSM learning phase, we set the number of training epoch be $\text{epoch} = 80$. In each epoch, 8000 random sample patterns are used in the learning. The parameters for the neighborhood variance are set as $\sigma_0 = 0.4$ and $\tau_1 = 20$. The parameters for the learning rate are set as $\alpha_0 = 0.01$ and $\tau_2 = 60$. The training parameters, $\{\text{epoch}, \sigma_0, \tau_1, \alpha_0, \tau_2\}$, are defined in [8].

4.1. Comparison with the global conformal mesh

The method in [4] shows a way to accomplish the global conformal parameterization (GCM) for the genus-zero surface. The proposed CSS can be applied to the GCM meshes. The GCM devises an iterative procedure to optimize a conformal energy function to evolve the mesh. The GCM operates two phases in its procedure. The first phase is to apply the barri-centric mapping algorithm and the second phase is to apply the conformal mapping algorithm. The procedure starts with a given mesh with all acute angles in all of its mesh holes. With any obtuse angle, the string constant in the energy function is negative and the constructed harmonic map is not bijective. Therefore, all mesh holes of this given mesh must have acute angles. The GCM cannot start with the unorganized cloud patterns X . The GCM requires an explicit given mesh to start its procedure. Another difficulty is that during the optimization procedure, certain triangles may be deformed to a line shape. This introduces a zero denominator in the formula of

the string constant. This constant approaches infinity and the procedure will not work. To avoid such situation, an extra effort is taken to detect the zero denominator and to prevent it from happening.

The stretching factors [5] in the GCM are highly non-uniform near the extrusive parts, such as the ears of the bunny, e.g. Figs. 8(e,f) and 11(b). When the bunny is conformally mapped to a sphere, the ears parts are shrunk to tiny regions. The GCM introduces a maneuver by punching small holes on the surface. This is because the GCM parametrization is highly dependent on the topology, the topology can be changed easily without affecting the geometry too much by punching holes on the surface. Generally, several faces are manually removed in advance from the extrusive parts of the surface, for example, the ear tips and the center of the bottom of the bunny. Since the CSM does not remove faces from the mesh, we keep the topology of the GCM unchanged in our experiments.

4.2. Results

Table 2 shows the CSS mismatch errors for the two meshes obtained by the CSM and GCM methods. For Igea model, both surfaces can reduce the errors, near 80% reduction. Fig. 7 shows the surface M of the Igea data derived from the CSM and GCM meshes. Both surfaces show the significant improvement. For bunny model, the surface from the GCM has large mismatch error. This is because the ears, see Figs. 8(e) and (f), are deformed. The reason is that the ears are shrinking in the GCM parameterization, an enlarged diagram shown in Fig. 8(f). The dense mesh \mathbb{N}^{new} , is incapable of making up the ears.

One of the reason for building the CSS is to facilitate the texture mapping. One can trace the texture details equally on the CSS without any ambiguous correspondence. The texture mapping of a genus-zero manifold is achieved in the following way. Suppose there are a texture image I , model surface M , and sphere surface S . First we select a desired view point of the model M and the north pole of the sphere S based on that view point. Then apply stereographic projection from image I to sphere S . Then map the image I from the sphere S to the model surface M . We use this projection is because it is a conformal mapping and is very regular near the point of tangency. Fig. 9(a) shows the result of a wood texture mapped on the bunny model. In Fig. 9(b), the checkerboard texture is mapped on the surface. A facial makeup of Chinese operas in Fig. 9(c) is mapped to the CSS in Fig. 9(d).

Fig. 10 shows two texture mapping cases. Figs. 10(a) and (b) show the checkerboard texture on the CSS Igea surface in Fig. 7(c). A facial makeup in Chinese opera in Fig. 10(c) is mapped on the curved Igea surface, see Fig. 10(d).

Fig. 11 shows the texture mapping results derived from the obtuse triangulation (a) and the acute triangulation (b) by the GCM. As aforementioned, the GCM could not work correctly for an obtuse mesh while the CSM can learn from scattered points (Fig. 11(c)).

Table 1
Model parameters

	Igea model		Bunny model	
	Vertices	Faces	Vertices	Faces
X_{few}	12,963	25,922	12,963	25,921
X_{more}	134,345	268,686	35,947	69,451
CSM	12,962	25,920	12,962	25,920
\mathbb{N}^{new}	64,002	128,000	64,002	128,000

Table 2
Mismatch error

	Igea model, E_{mis}	Bunny model, E_{mis}
CSM	1.4169×10^{-5}	5.2631×10^{-5}
GCM	1.4860×10^{-5}	0.0128
X_{few}	8.0370×10^{-5}	1.089×10^{-4}

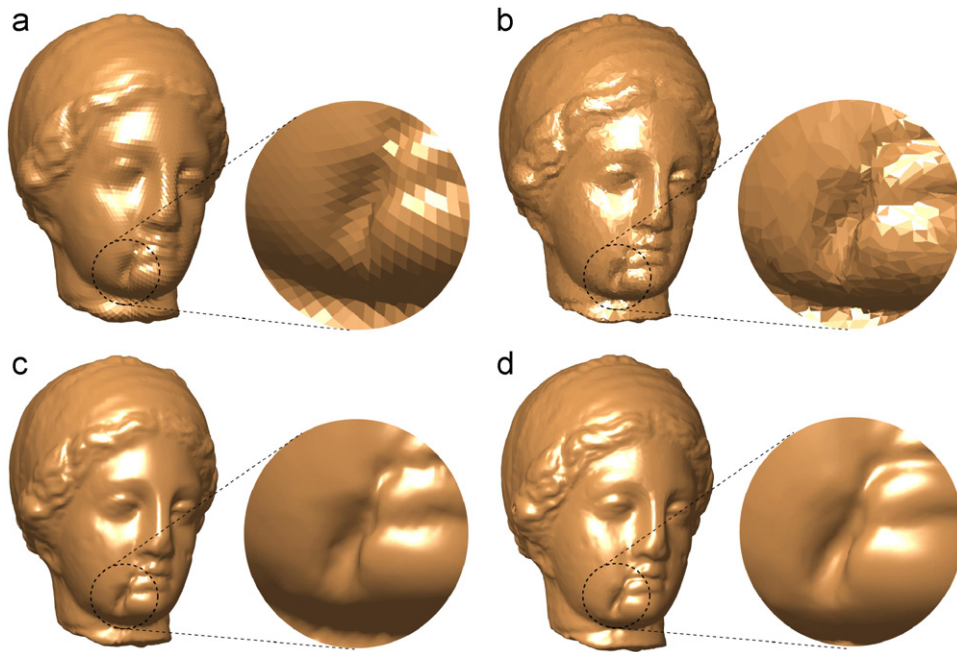


Fig. 7. The smooth surfaces obtained by the CSS using the meshes derived from the CSM and GCM methods for the Igea model. (a) The CSM mesh skeleton without the CSS. (b) The GCM mesh skeleton without the CSS. (c) The CSS computed from the CSM mesh. (d) The CSS computed from the GCM mesh.

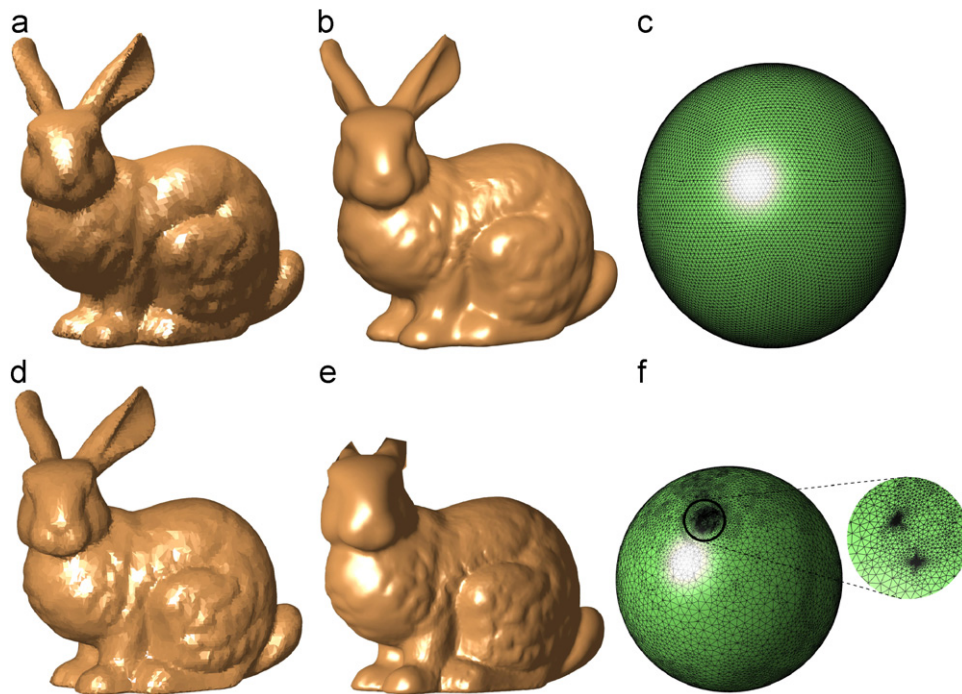


Fig. 8. The smooth surfaces (b,e) obtained by the CSS using the meshes derived from the CSM (a) and GCM (d) methods. (a) The CSM mesh skeleton without the CSS. (b) The CSS surface using the mesh derived by the CSM and its spherical parameterization (c). (d) The GCM mesh without the CSS. (e) The CSS surface using the GCM mesh and its spherical parameterization (f). The enlarged diagram in (f) shows the ears' part of the bunny model.

5. Discussion

The CSS algorithm accomplishes the curved surface instead of the flat triangle surface obtained from the CSM. The CSM [10] was designed, originally, to trace the system state which changes continuously and to resolve various severe competitions among finite neurons in the SOM. The SOM with finite neurons cannot be

used for monitoring the continuous state. The folded mapping in the SOM can be indicated and resolved by the negative values of the Jacobian of the mapping function [11]. The CSM can save and accommodate fine textures in the plane map. The CSM with flat triangle surface in 3D [8] may have ambiguous resolution along the triangle edges. So, the 3D surfaces S^d and M^d are not very suitable for tracing the continuous state. The surfaces S and M will



Fig. 9. (a) The wood texture mapping on M . (b) The checkerboard texture mapping on M . (c) The 2D facial makeup in Chinese opera. (d) The facial texture mapping on M .



Fig. 10. Texture mapping on the Igea's head. (a,b) The checkerboard texture mapping. (c) The 2D facial makeup in Chinese opera. (d) The facial makeup mapping on the Igea's head.

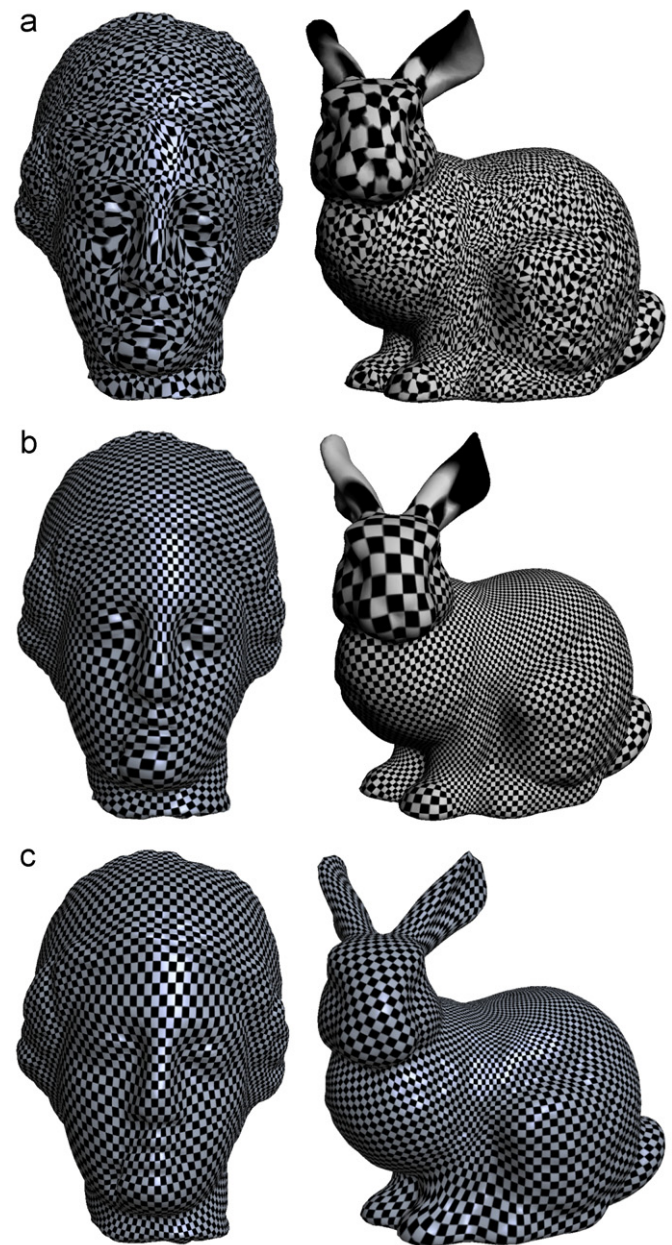


Fig. 11. Checkerboard texture mapping. (a) The mapping on the GCM mesh using an obtuse mesh as the initial input. (b) The mapping on the GCM mesh using an acute mesh as its initial input. (c) The mapping on the CSS obtained from the CSM mesh.

do. Note that the CSM [10] in 2D flat plane does not have such ambiguous problem.

In the CSS, instead of a surface in the form of flat triangles, a curved seamless parameterization for unorganized data patterns is accomplished for the model surface. This CSS serves as a kind of interpolation for the CSM mesh. Since the sphere surface possesses so many well behaved properties, we expect that the CSS algorithm can carry the full extent of these beautiful properties to the model surface, such as differentiation and integration. Texture mapping is a direct application of the CSS. The CSS can be additively applied to many triangular meshes obtained by existing methods. Other potential applications are the brain-to-brain registration, consistent parameterization [1], facial expression synthesis, deformable object simulation and computing geodesic path on a model surface. The parameterization for higher genus is also under our study.

As for mapping distortion, the CSS mapping from a sphere is quasi-distortion-free. This is because that the mapping from $\hat{\Delta}_n^r$ to $\hat{\Delta}_w^r$ is bijective (Appendix) and the CSM mapping from Δ_n^r to Δ_w^r is conformal. All triangles Δ_n^r are nearly equilateral triangles [8]. The stereographic mapping from the texture image to a sphere has a radial distortion but this distortion is very small near the point of tangency.

In comparison to other smooth surface representations such as the B-spline and NURBS [13], the CSS is superior to them. This is because the CSS possesses a unique conformal mapping to a sphere while the uv coordinate of the NURBS is not conformal to a sphere. One interesting question is that whether the NURBS could be applied to a learned CSM mesh. The answer is no because the CSM is formed by triangular patches but the NURBS is parameterized on quadrilateral patches [15].

Finally, we assert, without proof, that the mapping of each $\hat{\Delta}_n^r$ to its corresponding dome, $\hat{\Delta}_w^r$, is bijection, Lemma 1. $f_p : p \rightarrow p'$ is bijective; Lemma 2. $\mathcal{M}_{\Delta_n^r}^{\Delta_w^r} : p' \rightarrow q'$ is bijective; Lemma 3. $f_q : q' \rightarrow q$ is bijective. Appendix provides intuitive sketches for the proof. This mapping is from the triangular sphere surface $\hat{\Delta}_n^r$ to the triangular geodesic model surface $\hat{\Delta}_w^r$. We also assert, without proof, that the mapping of each subtriangular dome, $\hat{\Delta}_n^r$, to its corresponding dome, $\hat{\Delta}_w^r$, is smooth.

Acknowledgment

This work was supported by National Science Council.

Appendix A. Bijective mapping

Instead of a rigorous proof, we provide some intuitive observations. The CSS is composed of three consecutive mappings,

$$f_p : p \rightarrow p', \quad \mathcal{M}_{\Delta_n^r}^{\Delta_w^r} : p' \rightarrow q', \quad f_q : q' \rightarrow q.$$

To show the mapping, $q = f_q \circ \mathcal{M}_{\Delta_n^r}^{\Delta_w^r} \circ f_p(p)$, is bijective, we show all f_p , $\mathcal{M}_{\Delta_n^r}^{\Delta_w^r}$ and f_q are bijective separately.

Lemma 1. $f_p : p \rightarrow p'$ is bijective.

Proof. According to the definition, p is on a spherical triangle and p' is on the corresponding flat triangle Δ_n^r (Fig. 6):

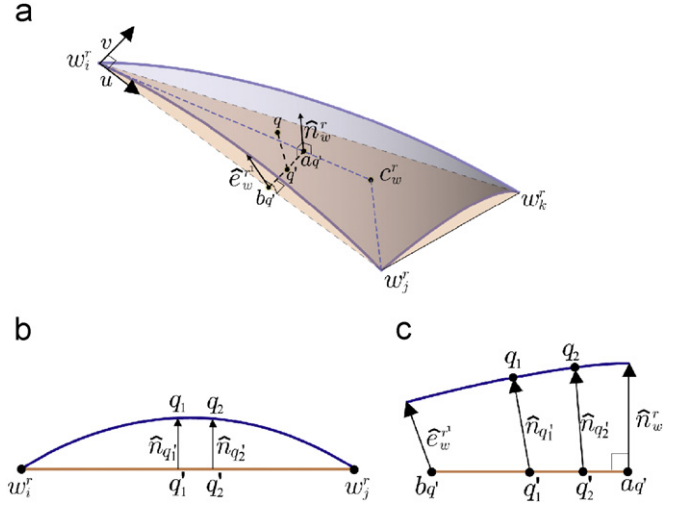
$$f_p : p' \equiv \overline{pc} \cap \Delta_n^r.$$

Since the line \overline{pc} does not parallel to Δ_n^r , there exists only one intersection point. Therefore, f_p is one-to-one (injective). Since all flat triangles Δ_n^r inscribe the unit sphere, for any point p on the sphere, the line \overline{pc} will intersect one Δ_n^r on a single projected point p' . Therefore f_p is onto (surjective). Because f_p is both injective and surjective, it is bijective. \square

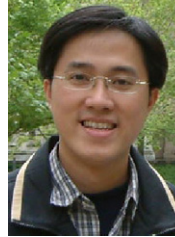
Lemma 2. $\mathcal{M}_{\Delta_n^r}^{\Delta_w^r} : p' \rightarrow q'$ is bijective.

Proof. According to the definition, $\mathcal{M}_{\Delta_n^r}^{\Delta_w^r}$ is a conformal mapping from the triangle Δ_n^r to its corresponding one Δ_w^r via an intermediate complex unit disk D . This mapping is computed by Schwarz–Christoffel transformation. The Schwarz–Christoffel transformation is one-to-one and onto. Therefore $\mathcal{M}_{\Delta_n^r}^{\Delta_w^r}$ is bijective. \square

Lemma 3. $f_q : q' \rightarrow q$ is bijective.



- [5] X. Gu, S.-T. Yau, Global conformal surface parameterization, in: L. Kobbelt, P. Schröder, H. Hoppe (Eds.), in: Symposium on Geometry Processing, ACM International Conference Proceeding Series, vol. 43, 2003.
- [6] I. Ivrişimţzis, W.-K. Jeong, S. Lee, Y. Lee, H.-P. Seidel, Neural meshes: surface reconstruction with a learning algorithm, Technical Report MPI-I-2004-4-005, Max-Planck-Institut Informatik, 2004.
- [7] H. Kenner, Geodesic Math and How to Use It, University of California Press, Berkeley, 1976.
- [8] C.-Y. Liou, Y.-T. Kuo, Conformal self-organizing map for a genus-zero manifold, Visual Comput. 21 (5) (2005) 340–353.
- [9] C.-Y. Liou, Y.-T. Kuo, J.-C. Huang, Smooth seamless surface construction based on conformal self-organizing map, in: The 13th International Conference on Neural Information Processing, Lecture Notes in Computer Science, vol. 4232, Springer, Hong Kong, 2006.
- [10] C.-Y. Liou, W.-P. Tai, Conformal self-organization for continuity on a feature map, Neural Networks 12 (6) (1999) 893–905.
- [11] C.-Y. Liou, W.-P. Tai, Conformality in the self-organization network, Artif. Intell. 116 (1–2) (2000) 265–286.
- [12] Z. Nehari, Conformal Mapping, McGraw-Hill, New York, 1975.
- [13] L. Piegl, W. Tiller, The NURBS Book, Springer, New York, NY, 1997.
- [14] Stanford Computer Graphics Laboratory, Data archives, 2004 (<http://www-graphics.stanford.edu/data/>).
- [15] W.-P. Tai, C.-Y. Liou, Image representation by self-organizing conformal network, Visual Comput. 16 (2) (2000) 91–105.
- [16] Y. Yu, Surface reconstruction from unorganized points using self-organizing neural networks, in: Proceedings of the IEEE Visualization '99, San Francisco, California, 1999.



Yen-Ting Kuo was born in Taichung, Taiwan, in 1977. He received the B.S. degree in Architecture from the National Cheng Kung University in 2001 and the M.S. degree in Computer Science and Information Engineering from the National Taiwan University in 2003. He is a Ph.D. candidate in the Department of Information System, University of Melbourne. His interests include learnable agents, knowledge modeling and neural networks.

Jau-Chi Huang received the B.S. and M.S. degrees from Department of Computer Science and Information Engineering, National Taiwan University in 2000 and 2002. She is a Ph.D. candidate in the same department. Her research interests are neural networks, artificial intelligence, and compositional indexing techniques.



Cheng-Yuan Liou was born in Taiwan in 1951. He received his Ph.D. from Massachusetts Institute of Technology in 1985. He is a Professor in the Department of Computer Science and Information Engineering, National Taiwan University. His research interests include brain theory and neural networks.