Communications in Information and Systems
Vol. 8, No. 1, pp. 1-16, 2008

COMPUTATION OF CURVATURES USING CONFORMAL PARAMETERIZATION

LOK MING LUI*, JEFF KWAN†, YALIN WANG‡, AND SHING-TUNG YAU§

Abstract. Curvatures on the surface are important geometric invariants and are widely used in different area of research. Examples include feature recognition, segmentation, or shape analysis. Therefore, it is of interest to develop an effective algorithm to approximate the curvatures accurately. The classical methods to compute these quantities involve the estimation of the normal and some involve the computation of the second derivatives of the 3 coordinate functions under the parameterization. Error is inevitably introduced because of the inaccurate approximation of the second derivatives and the normal. In this paper, we propose several novel methods to compute curvatures on the surface using the conformal parameterization. With the conformal parameterization, the conformal factor function λ can be defined on the surface. Mean curvature (H) and Gaussian curvatures (K) can then be computed with the conformal Factor (λ). It involves computing only the derivatives of the function λ, instead of the 3 coordinate functions and the normal. We also introduce a technique to compute H from K and vice versa, using the parallel surface.

Key words: Mean curvature, Gaussian curvature, normal, conformal parameterization, conformal factor.

1. Introduction.

Curvatures are important geometric quantities on the surface for different areas of research. They have found applications in many aspects such as smoothing/fairing [1], remeshing [2], non-photo-realistic rendering [3] as well as feature detection [4][5][6]. For example, many segmentation or mesh smoothing algorithms use curvatures to act as the feature to determine region boundaries. In medical research, curvatures are used to define the shape term for surface registration. Besides, curvature are used to detect the anatomic features, such as sulcal landmarks in the brain mapping research [6]. Since curvatures are the crucial components for various applications, it is of interest to develop an effective way to compute these geometric quantities accurately. In this work, we try to develop some effective ways to compute the curvatures and geodesic on the surface accurately.

Computing curvatures on the surface has been studied widely by different research groups. The most common techniques to compute these quantities all involve the estimation of the normal while some involve the computation of the second derivatives of the 3 coordinates functions under the parameterization. Because of the inaccurate approximation of the normal as well as the second derivatives, error is inevitably introduced. In this paper, we describe several methods to compute curvatures on

*Lok Ming Lui is with the Mathematics Department, University of California, Los Angeles.
†Jeff Kwan is with the Northfield Mount Hermon School, One Lamplighter Way, Mount Hermon, MA 01354.
‡Yalin Wang is with the Mathematics Department, University of California, Los Angeles.
§Shing-Tung Yau is with the Mathematics Department, Harvard University.

the surface using the conformal parameterization. With the conformal parameterization, we define the conformal factor function λ on the surface which encodes a lot of geometric information of the surface. Mean curvature H and Gaussian curvature K can then be computed with the conformal factor. It involves computing only the derivatives of the function λ, instead of the 3 coordinates functions and the normal. We also introduce a novel technique to compute H from K and vice versa, using the parallel surface.

The organization of the paper is as follow: in section II, we briefly describe some previous related works by different research groups. In section III, we explain some important mathematical background. In section IV, we describe briefly how conformal parameterization of the surface and its conformal factor can be computed. In section V, we explain in details our algorithms to compute curvatures on the surface. Experimental results are discussed in section VI. Finally, we summarize our work and describe possible future works in section VII.

2. Previous Work. The computation of curvatures has been studied extensively by various research groups. They have developed different methods and formulae to compute curvatures in order to improve the computational speed and accuracy. Surfaces are usually represented by triangle mesh. Generally speaking, curvature estimation techniques for triangle meshes can be divided into two categories, namely, discrete method and continuous method. In discrete methods, curvatures are approximated by formulating the closed forms for discrete differential operators that work directly on the triangle mesh. In continuous methods, curvatures are usually computed by interrogating the fitted smooth surface by polynomial fitting. This involves fitting a smooth surface (usually 2nd or 3rd order) locally to the vertex and some neighborhood around it. The most commonly used curvature estimation techniques can be listed as follow:

- **Discrete curvatures estimation**: Meyer et al. [7] proposed a set of discrete differential geometry operators to approximate important geometric attributes, such as normal and curvatures, on arbitrary triangle meshes. This is done by using averaging Voronoi cells and mixed finite-element/finite-volume method. Another commonly cited method is to approximate the discrete Gaussian curvature at each vertex, by computing the angles subtended at the vertex by each triangle in the 1-ring neighborhood and its area [8].
- **Eigenvalues/Eigenvectors method**: Taubin [9] proposed a method to compute the principal curvatures and principal directions at each vertex of a triangle mesh, by computing the eigenvalues and eigenvector of a 3×3 symmetric matrix. Surazhsky [10] later on proposed algorithms to improve Taubin's method. For the computation of the normal at each vertex, he suggested to replace the weighted average of the incident triangles by the

weighted incident angles. He also suggested to replace the the directional curvature by average of curvatures at the two vertices of an edge, which significantly smoothes the large variations in directional curvatures.

- **High order surface fitting method**: Hamann [11] proposed to fit a quadric approximated locally to a neighborhood of vertices. Curvatures are calculated by interrogating the quadric. Yokoya et al. [12] proposed to use a local quadratic function to fit the surface for segmentating the point cloud (range data) image, using a $(2m+1) \times (2m+1)$ window around the pixel of interest. The solution is obtained from standard least square fit. Goldfeather et al. [13] proposed a cubic order approximation method by adding normal vectors at adjacent vertices to create third degree term in the least-squares solution which results in a better fitting surface.
- **Normal based estimation**: Theisel et al. [14] proposed a normal based technique to estimate the curvature tensor on the triangle mesh. The algorithm estimates the curvature tensor for a single triangle equipped with an estimated or exact surface normals. The result is a continuous function for the curvature tensor inside each triangle.
- **Tensor averaging method**: Rusinkiewicz et al. [15] proposed to estimate the curvature per face by computing the directional derivative of the surface normal, which is obtained by differences between normal vectors on the face. The vertex curvature is obtained by taking a weighted average of the adjacent tensor faces' normal vectors.

3. Mathematical Theory. In this section, we are going to give a brief review of some important mathematical background related to our algorithm.

We start by briefly describing the concept of curvatures on a surface. Given a surface S, the normal curvature κ_n in some direction is the reciprocal of the radius of the circle that best approximate a normal slice of the surface in that direction. For smooth surface, it can be computed from a 2×2 symmetric matrix $\mathbf{W}$, called the Weingarten matrix, by:

$$\kappa_n = (u,v)\mathbf{W}\begin{pmatrix} u \\ v \end{pmatrix} = (u,v)\begin{pmatrix} e & f \\ f & g \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} \tag{1}$$

for any unit length vector (u,v) in the tangent plane of the surface.

The principal curvatures, principal directions, mean curvatures and gaussian curvatures can be defined by the eigenvalues and eigenvectors of $\mathbf{W}$. Let k_1, k_2 be the eigenvalues of $\mathbf{W}$ and $\vec{d_1}$, $\vec{d_2}$ be the eigenvectors of $\mathbf{W}$. Then, k_1 and k_2 are called the principal curvatures; $\vec{d_1}$ and $\vec{d_2}$ are called the principal directions. The mean curvature H is defined as the average of the principal curvatures: $H = \frac{k_1+k_2}{2}$. The Gaussian curvature K is defined as the product of the principal curvatures: $H = k_1 k_2$

Lastly, we are going to describe the concept of conformal parameteriation of a Riemann surface. All Riemann surfaces are locally Euclidean. Given two Riemann surfaces M and N. We can represent them locally as $\phi_M(x_1,x_2): R^2 \to M \subseteq R^3$ and $\phi_N(x_1,x_2): R^2 \to N \subseteq R^3$ respectively, where (x_1,x_2) are their coordinates. The inner product of the tangent vectors at each point of the surface can be represented by its first fundamental form. The first fundamental form on M can be written as $ds_M^2 = \sum_{i,j} g_{ij} dx^i dx^j$, where $g_{ij} = \frac{\partial \phi_M}{\partial x^i} \cdot \frac{\partial \phi_M}{\partial x^j}$ and $i,j = 1,2$. Similarly, the first fundamental form on N can be written as $ds_N^2 = \sum_{i,j} \widetilde{g}_{ij} dx^i dx^j$ where $\tilde{g}_{ij} = \frac{\partial \phi_N}{\partial x^i} \cdot \frac{\partial \phi_N}{\partial x^j}$ and $i,j = 1,2$. Given a map $f: M \to N$ between the M and N. With the local parameterization, f can be represented locally by its coordinates as $f: R^2 \to R^2$, $f(x_1,x_2) = (f_1(x_1,x_2), f_2(x_1,x_2))$. Every tangent vectors $\vec{v}$ on M can be mapped ($push\ forward$) by f to a tangent vectors $f_*(\vec{v})$ on N. The inner product of the vectors $f_*(\vec{v_1})$ and $f_*(\vec{v_2})$), where $\vec{v_1}$ and $\vec{v_2}$ are tangent vectors on M, is:

$$\begin{aligned} f^*(ds_N^2)(v_1,v_2) &:= < f_*(v_1), f_*(v_2) > \\ &= \sum_{i,j} \widetilde{g}_{ij} f_*(v_i) \cdot f_*(v_j) \\ &= \sum_{i,j} (\sum_{m,n} \widetilde{g}_{mn} \frac{\partial f_i}{\partial x^m} \frac{\partial f_j}{\partial x^n}) v_i v_j) \end{aligned} \tag{2}$$

Therefore, a new Riemannian metric $f^*(ds_N^2)$ on M is induced by f and ds_N^2, called the *pull back metric.* We say that the map f is **conformal** if

$$f^*(ds_N^2) = \lambda(x_1,x_2) ds_M^2$$

A parameterization $\varphi: R^2 \to M$ is a **conformal parameterization** if φ is a conformal map.

Intuitively, a map is conformal if it preserves the inner product of the tangent vectors up to a scaling factor, called the conformal factor λ. An immediate consequence is that every conformal map preserves angles.

4. The conformal parameterization of the surface and its conformal factor. One crucial component in our algorithm is to get a conformal parameterization of the surface. In this paper, we apply the algorithm proposed by Gu et al. [16], [17],[18] to parameterize the surface to 2D rectangles by computing its holomorphic one form. It is done by computing the surface's homology basis, De-Rham cohomology basis, harmonic one form and its hodge-star conjugates. We are going to describe briefly the basic idea of this algorithm in this section.

To parameterize a compact surface onto 2D rectangles, one intuitive technique is to cut it open along some suitable cutting boundaries. If the cut is suitably chosen, the parameterization could be conformal. In the algorithm that we use to parameterize the surface, we search for the suitable cutting boundaries on the surface in order to

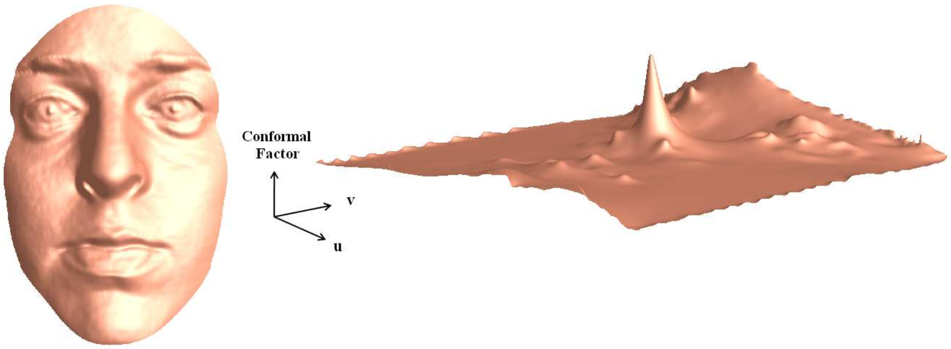

FIG. 1. *The plot of the conformal factor λ of a human face verses u and v of the parameter domain. The conformal factor is a smooth function which describe the stretching effect under the conformal parameterization. Observe that the approximation of the conformal factor function is reasonably smooth.*

get a conformal map. This is done by computing the holomorphic one form on the surface and obtain the conformal map by integrating the holomorphic one form.

The holomorphic one form ω is a complex differential form. To compute the holomorphic 1-form, we start by computing the harmonic 1-form ω on the surface. Similar to complex analysis, we can compute a harmonic conjugate $*\omega$ of ω, such that $W := \omega + i * \omega$ is a holomorphic (analytic) 1-form. The harmonic one form can be computed from the homology basis of the surface. Given a homology basis $\{e_1, ..., e_{2g}\}$ on the surface, we can compute a set (basis) of the harmonic 1-forms $\{\omega_1, ..., \omega_{2g}\}$ (cohomology) by solving the following system:

$$\begin{cases} d\omega = \sum_{i=1}^{3}\omega([u_{j-1},u_j]) = 0, \forall [u_0,u_1,u_2]\in M, u_0=u_3 & \text{(closedness);}\\ \triangle\omega = \sum_{[u,v]\in M}\omega([u,v]) = 0\forall[u,v]\in M & \text{(harmonicity);}\\ \int_{e_i}\omega = \sum_{i=1}^{n_i}\omega([u_{j-1}^i,u_j^i]) = \delta_{ij}\forall e_i = \sum_{j=1}^{n_i}[u_{j-1}^i,u_j^i], u_0^i = u_{n_i}^i & \text{(conjugacy).}\end{cases}$$

where $[u_0, u_1, u_2]$ represents a face on M; $[u, v]$ represents an edge on M; $k_{uv} = \frac{1}{2}(cot\alpha + cot\beta)$ in which α, β are the angles against the edge $[u, v]$.

After we get the holomorphic 1-form, we can compute the conformal parameterization ϕ by integrating the one form: $\phi(p) = \int_\gamma \omega = \int_\gamma f(z_\alpha)dz_\alpha$, where γ is any path joining p to a fixed point c on the surface and $\omega = f(z_\alpha)dz_\alpha$.

Double covering techniques are applied to surfaces with boundaries to convert them to closed symmetric surfaces.

Further details about the algorithm can be found in [16], [17].

Given the conformal parameterization of the surface, we can obtain a conformal factor function λ. By definition, the conformal parameterization has a simple Riemannian metric, namely,

$$g_{ij} = \begin{cases} \lambda & \text{if } i = j;\\ 0 & \text{if } i \neq j.\end{cases}$$

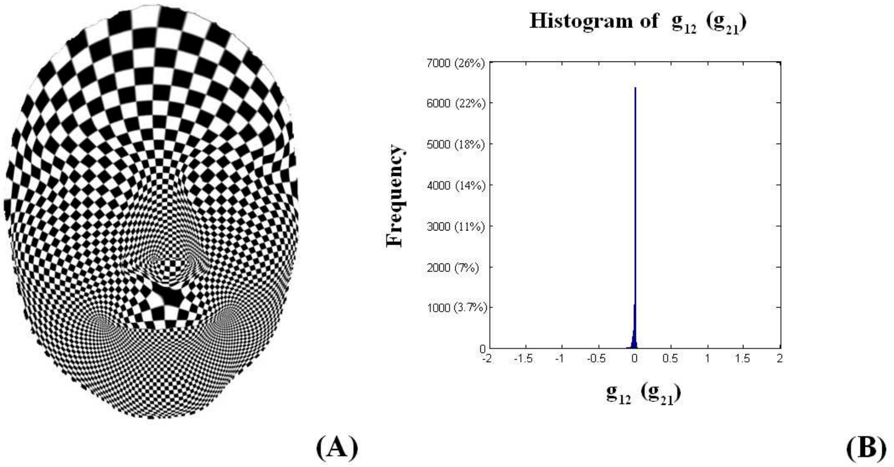

Fig. 2. *(A) shows the conformal coordinates grid on the human face introduced using the conformal parameterization. (B) shows the histogram of $g_{12} = g_{21}$ of a Riemann surface under the conformal parameterization. Observe that $g_{12} = g_{21}$ are very close to zero at most vertex. It means the Riemannian metric is a diagonal matrix, which results in simple expression for the curvatures computation.*

In other words, the four metric coefficients are reduced to one coefficient metric λ, called the conformal factor. With this property, surface differential operators can be expressed within the conformal coordinates with simple formulae. The expressions are similar to the usual Euclidean differential operators, except for a scalar multiplication of the conformal factor. The conformal factor at a point p on the surface S can be determined by computing the scaling factor of a small area around p under the parameterization $\phi : R^2 \to S$. Mathematically, $\lambda(p) = \frac{\mathbf{Area}(B_\epsilon(p))}{\mathbf{Area}(\phi^{-1}(\mathbf{B}_\epsilon(\mathbf{p})))}$, where $B_\epsilon(p)$ is an open ball around p of radius p. Figure 1 shows the plot of conformal factor λ verses u and v of the parameter domain. The conformal factor is a smooth function which describe the stretching effect under the conformal parameterization. Observe that the approximation of the conformal factor function is reasonably smooth. Figure 2 (A) shows the conformal coordinates grid on the human face introduced using the conformal parameterization. Figure 2(B) shows the histogram of g_{12} of the Riemann surface under the conformal parameterization. Note that by definition, $g_{12} = g_{21} = \phi_u \cdot \phi_v$, where $\phi(u, v)$ is the conformal parameterization of the surface. Observe that $g_{12}(= g_{21})$ are very close to zero at most vertex. It means the Riemannian metric is diagonal under the conformal parameterization.

The conformal factor λ encodes a lot of important geometric information about the surface and can be used to compute curvatures and geodesic.

5. Algorithm. In this section, we explain in details our algorithms to compute curvatures on the surface.

5.1. Computation of the gaussian curvature K from the conformal factor. Let M be a Riemann surface and let $\phi(u,v) : R^2 \to M$ be the parameterization of M. Denote the first fundamental form by $ds^2_M = Edu^2 + Fdudv + Gdv^2$. When ϕ is orthogonal, $F = 0$. When ϕ is conformal, $E = F = \lambda$, where λ is the conformal factor. We propose to compute the gaussian curvature K with the functions E and G when ϕ is orthogonal, and compute K with λ when ϕ is conformal. Under the conformal parameterization, gaussian curvature can be computed easily [19]. These can be described by the following lemma and theorems.

Lemma 1.

$$K = -[(\Gamma^2_{12})_u - (\Gamma^2_{11})_v + \Gamma^1_{12}\Gamma^2_{11} + \Gamma^2_{12}\Gamma^2_{12} - \Gamma^2_{11}\Gamma^2_{22} - \Gamma^1_{11}\Gamma^2_{12}]/E \tag{3}$$

Theorem 1. *Suppose ϕ is orthogonal ($F = 0$). The gaussian curvature K can be computed by:*

$$K = -\frac{1}{2\sqrt{EG}}[(\frac{E_v}{\sqrt{EG}})_v + (\frac{G_u}{\sqrt{EG}})_u] \tag{4}$$

Proof. With $F = 0$, $\Gamma^2_{12} = \frac{G_u}{2G}$; $\Gamma^2_{11} = -\frac{E_v}{2G}$; $\Gamma^1_{12} = \frac{E_v}{2E}$; $\Gamma^1_{11} = \frac{E_u}{2E}$; $\Gamma^2_{11} = -\frac{E_u}{2E}$; $\Gamma^2_{22} = \frac{E_u}{2E}$. Putting them into equation 3, we have:

$$\begin{aligned} K &= -[(\frac{G_u}{2G})_u + (\frac{E_v}{2G})_v - \frac{E^2_v}{4EG} + \frac{G^2_u}{4G^2} + \frac{E_vG_v}{4G^2} - \frac{E_uG_u}{4EG}]/E \\ &= -[(\frac{G_{uu}}{2G} - \frac{G^2_u}{2G^2}) + (\frac{E_{vv}}{2G} - \frac{E_vG_v}{2G^2}) - \frac{E^2_v}{4EG} + \frac{G^2_u}{4G^2} + \frac{E_vG_v}{4G^2} - \frac{E_uG_u}{4EG}]/E \\ &= -[\frac{G_{uu}}{2EG} + \frac{E_{vv}}{2EG} - \frac{E^2_{vv}}{4E^2G} - \frac{G^2_u}{4EG^2}) - \frac{E_vG_v}{4EG^2} - \frac{E_uG_u}{4E^2G}] \\ &= -\frac{1}{2\sqrt{EG}}[(\frac{E_v}{\sqrt{EG}})_v + (\frac{G_u}{\sqrt{EG}})_u] \end{aligned} \tag{5}$$

Theorem 2. *Suppose ϕ is conformal with $E = G = \lambda$ and $F = 0$, where λ is the conformal factor. The gaussian curvature K can be computed by:*

$$K = -\frac{1}{2\lambda}\Delta log\ \lambda \tag{6}$$

Proof. Suppose ϕ is conformal and $\lambda = \lambda(u,v)$ is the conformal factor with respect to ϕ. Put $E = G = \lambda$ into equation 4, we have:

$$\begin{aligned} K &= -\frac{1}{2\sqrt{\lambda^2}}[(\frac{\lambda_v}{\sqrt{\lambda^2}})_v + (\frac{\lambda_u}{\sqrt{\lambda^2}})_u] \\ &= -\frac{1}{2\lambda}[(\frac{\lambda_v}{\lambda})_v + (\frac{\lambda_u}{\lambda})_u] \\ &= -\frac{1}{2\lambda}\Delta log\ \lambda \end{aligned} \tag{7}$$

When ϕ is a conformal parameterization of the surface S, we see that we can compute K on S easily by equation 6. This significantly reduces the error in the approximation compared to other classical methods. In the classical methods, the computation of K involves the approximation of the second derivatives of the coordinates functions as well as the normal. In our proposed method, we compute K using one function λ defined on S. The computation can then be greatly simplified and the numerical error can be significantly reduced.

Due to the computational error , the conformal parameterization we approximate may sometimes only be orthogonal but not conformal. That is, $F = 0$ but $E \neq G$. In this case, instead of using equation 6, we use equation 4 to compute the Gaussian curvature K so as to get a more accurate result.

5.2. Computation of the mean curvature H from the conformal factor.

With the conformal parameterization $\phi = \phi(u, v)$ of the surface, the computation of the mean curvature H will be easier [19].

Given an arbitrary parameterization (not necessarily conformal) of the surface, we can generally compute the mean curvature H using the following formula:

$$H = \frac{1}{2}\frac{Eg - 2fF + Ge}{EG - F^2} \tag{8}$$

where: $E =< \phi_u, \phi_u >$; $F =< \phi_u, \phi_v >$; $G =< \phi_v, \phi_v >$; $e =< \phi_{uu}, \vec{N} >$; $e =< \phi_{uv}, \vec{N} >$; $g =< \phi_{vv}, \vec{N} >$ and $\vec{N}$ is the surface normal. This formula is complicated for arbitrary parameterization and involves several partial derivatives in the formula. This inevitably causes numerical error. However, with the conformal parameterization, the formula can be significantly simplified.

Theorem 3. *Suppose ϕ is conformal with $E = G = \lambda$ and $F = 0$, where λ is the conformal factor. The mean curvature H can be computed by:*

$$H = \frac{1}{2\lambda}\mathbf{sign}(\phi)|\Delta\phi| = \pm\frac{1}{2\lambda}|\phi_{uu} + \phi_{vv}| \tag{9}$$

where $\vec{N}$ is the (unit) surface normal, $\mathbf{sign}(\phi) = \frac{<\Delta\phi, \vec{N}>}{|\Delta\phi|} = \pm 1$.

Proof. Suppose ϕ is conformal and $\lambda = \lambda(u, v)$ is the conformal factor with respect to ϕ.

We have $< \phi_u, \phi_u >=< \phi_v, \phi_v >= \lambda$ and $< \phi_u, \phi_v >= 0$. By differentiation, we have:

$$< \phi_{uu}, \phi_u >=< \phi_{vu}, \phi_v >= - < \phi_u, \phi_{vv} >$$

We get: $< \phi_{uu} + \phi_{vv}, \phi_u >= 0$. And similarly we get: $< \phi_{uu} + \phi_{vv}, \phi_v >= 0$. Therefore, $\Delta\phi$ is parallel to $\vec{N}$ and $\mathbf{sign}(\phi) = \frac{<\Delta\phi, \vec{N}>}{|\Delta\phi|} = \pm 1$.

Now,

$$H = \frac{1}{2}\frac{Eg - 2fF + Ge}{EG - F^2} = \frac{1}{2}\frac{g + e}{\lambda} = \frac{1}{2}\frac{< \phi_{uu} + \phi_{vv}, \vec{N} >}{\lambda}$$

So,

$$H = \frac{1}{2}\frac{< \Delta\phi, \vec{N} >}{\lambda} = \frac{1}{2}\frac{< \Delta\phi, \vec{N} >}{|\Delta\phi|\lambda}|\Delta\phi| = \frac{1}{2\lambda}\mathbf{sign}(\phi)|\Delta\phi|$$

The formula for computing mean curvature is significantly simplified with conformal parameterization. As a result, the computational cost and numerical error can be reduced. Suppose the surface normal can be accurately approximated, we can further reduce the second derivative in the formula to first derivative:

$$H = \frac{1}{2\lambda}\mathbf{sign}(\phi)|\Delta\phi| = \mp\frac{1}{2\lambda} < \phi_u, \vec{N}_u > + < \phi_v, \vec{N}_v > \tag{10}$$

However, in practice, the surface normal is usually inaccurately approximated. This is one of the main problem why many curvature estimation method do not give an accurate approximation of the mean curvature. This problem can be significantly improved by the above formula $H = \frac{1}{2\lambda}\mathbf{sign}(\phi)|\Delta\phi|$. As we can see, the only part we need to use the surface normal $\vec{N}$ is when we compute $\mathbf{sign}(\phi) = \frac{<\Delta\phi, \vec{N}>}{|\Delta\phi|} = \pm 1$. Since we know the value of it is either 1 or -1, the surface normal $\vec{N}$ does not need to be accurately estimated. As long as it can tell us the convexity of the surface at a particular point on the surface, we can determine whether $\mathbf{sign}(\phi)$ is equal to 1 or -1. Using the formula that we described above, the curvature can be more accurately computed, even if the surface normal $\vec{N}$ cannot be estimated accurately.

5.3. Computation of H from K and K from H using parallel surface. After we have the accurate approximation of the Gaussian curvature K, we can get the mean curvature H from K using the idea of parallel surface. Similarly, we can get K from H using the parallel surface.

Given a parameterization $\phi = \phi(u, v)$ of S, a parallel surface S_a of S is the parameterized surface with parameterization: $\phi^a(u, v) = \phi(u, v) + a\vec{N}(u, v)$ where a is a constant; $\vec{N}$ is the surface normal of S. Figure 3 shows the parallel surface of a human face. The human face is firstly parameterized by a conformal parameterization ϕ. The surface normal is computed by $\vec{N} = \frac{\phi_u \times \phi_v}{|\phi_u \times \phi_v|}$. Here, we set a to be equal to 0.01.

With the concept of the parallel surface, we can compute H from K and vice versa using the following theorem.

Theorem 4. *Suppose ϕ is the conformal parameterization of S and S_a is its parallel surface. Let K and K_s be the Gaussian curvatures on S and S_a respectively. Let H and H_s be the mean curvatures on S and S_a respectively.*
The mean curvature H can be computed by:

$$H = \frac{(K_s - K) + (KK_s)a^2}{2aK_s} \tag{11}$$

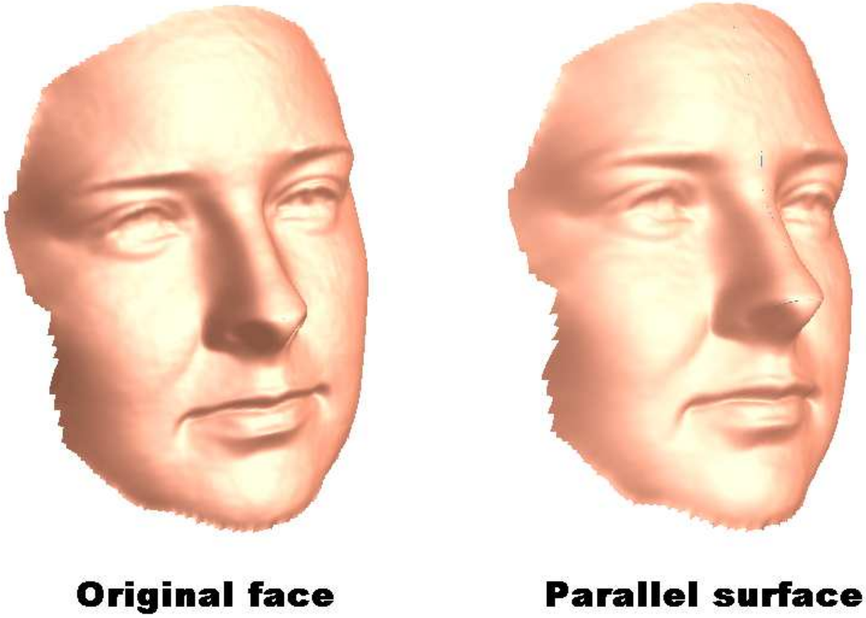

FIG. 3. *The figure shows the parallel surface of a human face. The human face is firstly parameterized by a conformal parameterization ϕ. The surface normal is computed by $\vec{N} = \frac{\phi_u \times \phi_v}{|\phi_u \times \phi_v|}$. The parallel surface is shown on the right which can be computed by the formula: $\phi^a(u,v) = \phi(u,v) + a\vec{N}(u,v)$ where a is a constant; $\vec{N}$ is the surface normal of S. Here, a is equal to 0.01.*

The gaussian curvature K can be computed by:

$$K = \frac{(H - H_s) + 2HH_s a}{a + H_s a^2} \tag{12}$$

Proof. Let $\phi = \phi(u,v)$ be the parameterization of S and $\phi^a(u,v) = \phi(u,v) + a\vec{N}(u,v)$ be the parameterization of the parallel surface S_a, where $\vec{N}$ is the surface normal of S.
Firstly, we will prove that: $\phi^a_u \times \phi^a_v = (1 - 2Ha + Ka^2)\phi_u \times \phi_v$.
Note that:

$$\begin{aligned}
\phi^a_u \times \phi^a_v &= (\phi_u + a\vec{N}_u) \times (\phi_v + a\vec{N}_v) \\
&= \phi_u \times \phi_v + a\phi_u \times \vec{N}_v + a\vec{N}_u \times \phi_v + a^2(\vec{N}_u \times \vec{N}_v) \\
&= \phi_u \times \phi_v - 2Ha\phi_u \times \phi_v + a^2 K(\phi_u \times \phi_v) \\
&= (1 - 2Ha + Ka^2)\phi_u \times \phi_v
\end{aligned}$$

Therefore, the surface normal $\vec{N}^a$ of S_a is the same as the surface normal $\vec{N}$ of S.
Now,

$$\begin{aligned}\vec{N}_u^a \times \vec{N}_v^a &= K_s \phi_u^a \times \phi_v^a = K_s(1 - 2Ha + Ka^2)\phi_u \times \phi_v \\ &= \vec{N}_u \times \vec{N}_v = K\phi_u \times \phi_v\end{aligned}$$

So,

$$K_s = \frac{K}{1 - 2Ha + Ka^2}$$

and,

$$H = \frac{(K_s - K) + (KK_s)a^2}{2aK_s}$$

Since $\phi = \phi^a - a\vec{N}$, we have:

$$K = \frac{K_s}{1 + 2H_s a + K_s a^2}$$

and so,

$$\begin{aligned}H_s = \frac{K_s - K - KK_s a^2}{2Ka} &= \frac{\frac{K}{1-sHa+Ka^2} - K - \frac{K^2a^2}{1-sHa+Ka^2}}{2Ka} \\ = \frac{H - aK}{1 - 2Ha + Ka^2}\end{aligned}$$

we get,

$$K = \frac{(H - H_s) + 2HH_s a}{a + H_s a^2}$$

6. Experimental Result. We tested our algorithm on the synthetic data and the real face data. The experimental result shows that our algorithm can effectively compute the curvatures on the surface more accurately.

To test whether our algorithm can accurately approximate the curvatures, we test computing the curvatures using our method on synthetic data on which the exact curvatures are know. In our experiment, we tested our curvature estimation method on a catenoid on which the exact curvatures are known. Figure 4 left shows the catenoid surface. Figure 4 middle shows the plot of mean curvature of the catenoid verses the u and v of the parameter domain (resolution=2500). The exact mean curvature of the catenoid is zero since it is a minimal surface. Note that the approximated values are very close to zero. Figure 4 right shows the plot of the gaussian curvature verses the u and v of the parameter domain. It closely resembles the exact mean curvature of a catenoid. Note that both the approximation of the mean and gaussian curvature are very smooth.

We also stuided how the error of the curvature estimation differs with different resolution (number of vertices on the surface). Figure 5 shows the plot of the maximum error of the mean curvature estimation under different resolution using different

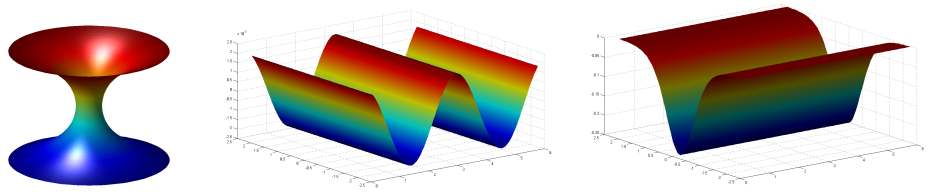

Fig. 4. *We test our curvature estimation method on a catenoid on which the exact curvatures are known. The left shows the catenoid surface. The middle shows the plot of mean curvature of the catenoid verses the u and v of the parameter domain (resolution=2500). The exact mean curvature of the catenoid is zero since it is a minimal surface. Note that the approximated values are very close to zero. The right shows the plot of the gaussian curvature verses the u and v of the parameter domain. It closely resembles the exact mean curvature of a catenoid. Note that both the approximation of the mean and gaussian curvature are very smooth.*

method. The vertical axis represents the maximum error amongst all vertices. The horizontal axis represents the resolution. The red curve shows the error using the normal based curvature estimation method. The green curve shows the error using the Weingarten matrix estimation method. The blue curve shows the error using our proposed method, which compute the mean curvature from the conformal factor λ. Note that the error using our proposed is the least when comparing with the other two methods. It shows that our method can effectively compute the mean curvature accurately.

We also tested our methods on the real human face data. We applied our algorithm to compute the mean curvature of the human face using the conformal factor. Figure 6 shows the plot of the mean curvature of a human face verses u and v of the parameter domain. The surface is firstly parameterized with a conformal parameterization ϕ. The mean curvature is then computed with the formula: $H = \frac{1}{2\lambda}\mathbf{sign}(\phi)|\Delta\phi|$.

We applied our method to compute the gaussian curvature of the human face with the conformal factor. Figure 7 shows the approximation of the gaussian curvature on a human face from the conformal factor. The surface is firstly parameterized with a conformal parameterization. Conformal factor is then computed to compute the gaussian curvature. The figure shows the plot of the gaussian curvature verses u and v of the parameter domain.

We also tested our algorithm for computing the mean curvature from the gaussian curvature using the parallel surface. In Figure 8, we illustrate how we can compute the mean curvature of the human face from the gaussian curvature using the parallel surface. The gaussian curvature on the human face is firstly computed from the conformal factor. The mean curvature can then be computed using the parallel surface. Note that the approximation of the mean curvature is reasonably smooth.

7. Conclusion. In this paper, we have described several novel methods to compute curvatures on the surface using the conformal parameterization. With the confor-

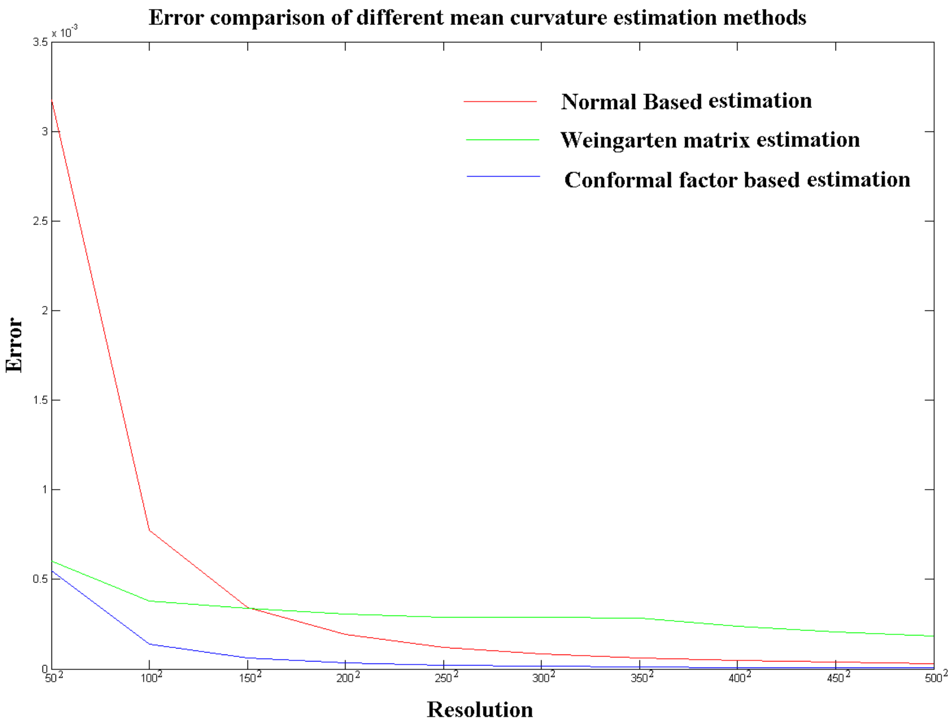

Fig. 5. *The plot of the maximum error of the mean curvature estimation under different resolution using different method. The vertical axis represents the maximum error amongst all vertices. The horizontal axis represents the resolution. The red curve shows the error using the normal based curvature estimation method. The green curve shows the error using the Weingarten matrix estimation method. The blue curve shows the error using our proposed method, which compute the mean curvature from the conformal factor λ. Note that the error using our proposed method is the least when comparing with the other two methods. It shows that our method can effectively compute the mean curvature accurately.*

mal parameterization, the conformal factor function λ can be defined on the surface. Mean H and Gaussian K curvatures can then be computed with the conformal factor. It involves computing only the derivatives of the function λ, instead of the 3 coordinates functions and the normal. We also introduce a technique to compute H from K and vice versa, using the parallel surface. As far as we know, we are the first group using the conformalilty to compute the curvatures on the surface. Experimental results show that our methods can effectively and accurately compute the curvatures on the surface. In the future, we will test our methods on more surfaces and study numerically how the error in the approximation will be affected by the error in the approximation of the conformal factor.

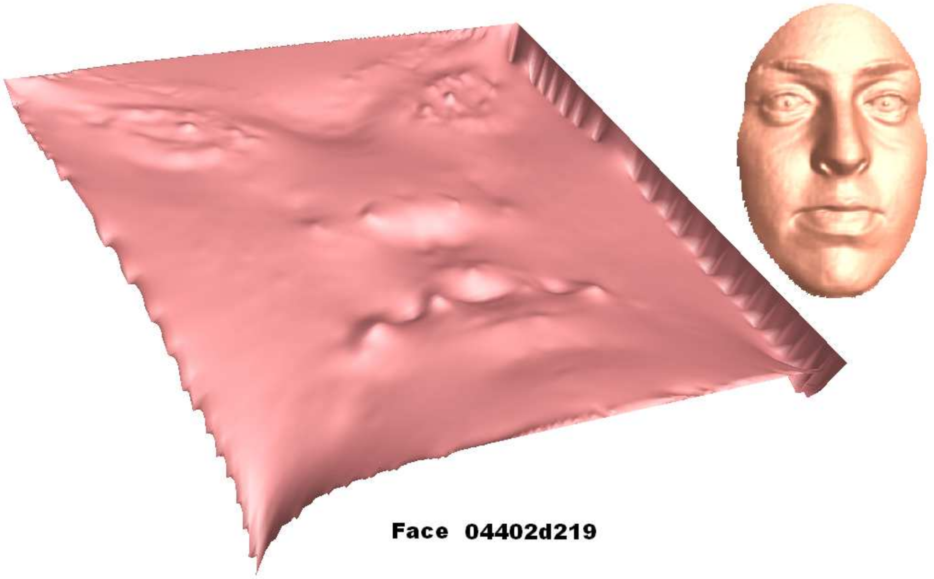

Fig. 6. *The plot of the mean curvature of a human face verses u and v of the parameter domain. The surface is firstly parameterized with a conformal parameterization ϕ. The mean curvature is then computed with the formula:* $H = \frac{1}{2\lambda}\mathbf{sign}(\phi)|\Delta\phi|$

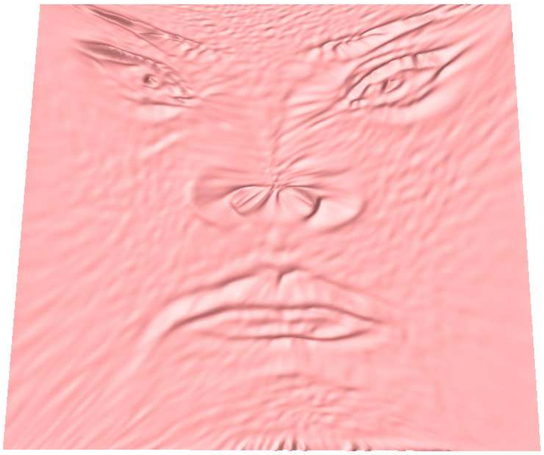

Fig. 7. *Approximation of the gaussian curvature on a human face from the conformal factor. The surface is firstly parameterized with a conformal parameterization. Conformal factor is then computed to compute the gaussian curvature. The figure shows the plot of the gaussian curvature verses u and v of the parameter domain.*

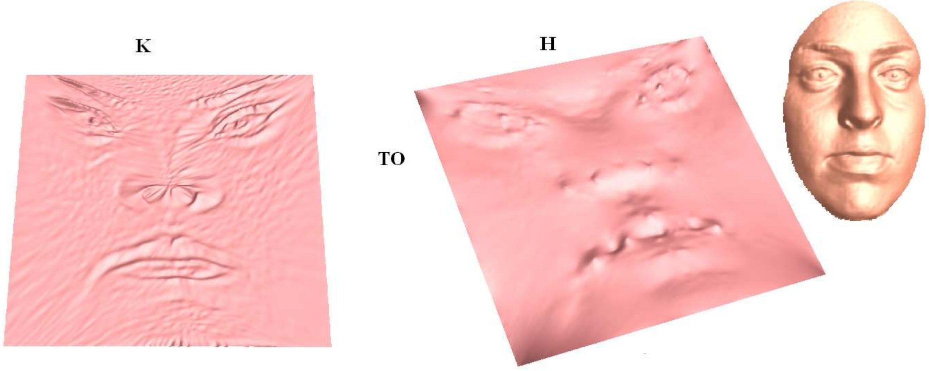

Fig. 8. *Illustration of computing the mean curvature of the human face from the gaussian curvature using the parallel surface. The gaussian curvature on the human face is firstly computed from the conformal factor. The mean curvature can then be computed using the parallel surface. Note that the approximation of the mean curvature is reasonably smooth.*

REFERENCES

[1] M. Desbrun, M. Meyer, P. Schroder, and A. H. Barr, *Implicit fairing of irregular meshes using diffusion and curvature flow*, Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pp. 317–324, 1999.

[2] P. Alliez, E. C. de Verdiere, O. Devillers, and M. Isenburg, *Anisotropic polygonal remeshing*, ACM Transactions on Graphics. Special issue for SIGGRAPH conference, pp. 485–493, 2003.

[3] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella, *Suggestive contours for conveying shape*, ACM Transactions on Graphics (TOG), 22:3(2003), pp. 848–855.

[4] Y. Ohtake, A. Belyaev, and H. Seidel, *Ridge-valley lines on meshes via implicit surface fitting*, ACM Transaction of Graphic, 23:3, pp. 609–612.

[5] K. Hildebrandt, K. Polthier, and M. Wardetzky, *Smooth feature lines on surface meshes*, Proceedings of Symposium on Geometry Processing, pp. 85–90, 2005.

[6] L. M. Lui, Y. Wang, T. F. Chan, and P. Thompson, *Automatic landmark tracking and its application to the optimization of brain conformal mapping*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), New York, vol. 2, pp. 1784–1792, 2006.

[7] M. Meyer, M. Desbrun, P. Schroder, and A. Barr, *Discrete differential-geometry operators for triangulated 2-manifolds*, Visualization and mathematics, 3(2002), pp. 34–57.

[8] A. Gray, *Modern differential geometry of curves and surfaces with mathematica*, CRC Press, 1998.

[9] G. Taubin, *Estimating the tensor of curvature of a surface from polyhedra approximation*, Proceedings of IEEE International Conference on Computer Vision, vol. 3, pp. 902–907, 1995.

[10] T. Surazhsky, E. Magid, O. Soldea, G. Elber, and E. Rivlin, *A comparison of gassian and mean curvatures methods on triangular meshes*, Proceedings of IEEE international conference on robotics and automation, pp. 1021–1026, September 2003.

[11] B. Hamann, *Curvature approximation for triangulated surfaces*, Springer Computing Supplementum, pp. 139–153, 1993.

[12] N. Yokoya and M. Levine, *Range image segmentation based on differential geometry, a hybrid*

approach, IEEE Transaction of Pattern Analysis and Machine Intelligence, 2:6(1989), pp. 634–639.

[13] J. Goldfeather and V. Interrante, *A novel cubic-order algorithm for approximating principal direction vectors*, ACM Transaction of Graphics, 23:1(2004), pp. 139–153.

[14] H. Theisel, C. Rossl, R. Zayer, and H.-P. Seidel, *Normal based estimation of the curvature tensor for triangular meshes*, Proceedings of 12th Pacific conference on computer graphics and applications, pp. 288–297, 2004.

[15] S. Rusinkiewicz, *Estimation of curvatures and their derivatives on triangle meshes*, Proceedings of Second international symposium on 3D data processing, pp. 486–493, 2004.

[16] X. Gu and S.-T. Yau, *Global conformal surface parameterization*, ACM Symposium on Geometry Processing 2003, 2003.

[17] Y. Wang, X. Gu, K. Hayashi, T. F. Chan, P. Thompson, and S. Yau, *Brain surface parameterization using Riemann surface structure*, Medical Image Computing and Computer-Assisted Intervention - MICCAI 2005, pp. 657–665, Oct 2005.

[18] S.-T. Y. Xianfeng Gu, *Computational Conformal Geometry.* Higher education Press, 2007.

[19] X. Gu, Y. Wang, and S.-T. Yau, *Geometric compression using Riemann surface structure*, Communication in Information System, 3:3(2003), pp. 171–182.