



## Curvatures estimation on triangular mesh\*

DONG Chen-shi (董辰世), WANG Guo-zhao (汪国昭)<sup>‡</sup>

(Department of Mathematics, Zhejiang University, Hangzhou 310027, China)

E-mail: iluvm@163.com; wgz@math.zju.edu.cn

Received Oct. 27, 2004; revision accepted Jan. 24, 2005

**Abstract:** Curvatures are important geometric attributes of surfaces. There are many applications that require as a first step the accurate estimation of curvatures at arbitrary vertices on a triangulated surface. Chen and Schmitt (1992) and Taubin (1995) presented two simple methods to estimate principal curvatures. They used circular arcs to approximate the normal curvature. We find this may cause large error in some cases. In this paper, we describe a more accurate method to estimate the normal curvature, and present a novel algorithm to estimate principal curvatures by simplifying the Chen and Schmitt's method. Some comparison results are also shown in this paper.

**Key words:** Triangular mesh, Curvatures estimation, Principal curvatures

**doi:**10.1631/jzus.2005.AS0128

**Document code:** A

**CLC number:** TP391.7

### INTRODUCTION

Surface curvatures on differentiable manifolds are important invariants in differential geometry, and play a key role in tasks such as registration, smoothing, simplification, etc. For large meshes, we will find curvatures in reverse engineering (Alrashdan *et al.*, 2000; Yang and Lee, 1999), medial visualization (Monga *et al.*, 1992; Sander and Zucker, 1990), and other applications. For a given triangular mesh, which approximates some unknown smoothing surface, there were many methods proposed for estimating curvatures of the underlying surface (Chen and Schmitt, 1992; Flynn and Jain, 1989; Goldfeather and Interrante, 2004; Meyer *et al.*, 2003; Taubin, 1995). In this article, we will introduce a novel normal curvature estimation method that we believe performs significantly better under certain conditions than many other proposed methods, and simplify the algorithm in Ref (Chen and Schmitt, 1992). In Section 2, we review the basic mathematics behind calculating curvatures. In Section 3, we briefly describe the previous approaches

to estimate curvatures on triangular meshes. In Section 4, we describe our method for approximating the normal curvature. In Section 5, a simple algorithm for computing principal curvatures is described in detail. In Section 6, we compare the result of our method with Taubin's method (Taubin, 1995) and Meyer's method (Meyer *et al.*, 2003). Finally we give the conclusion of this paper.

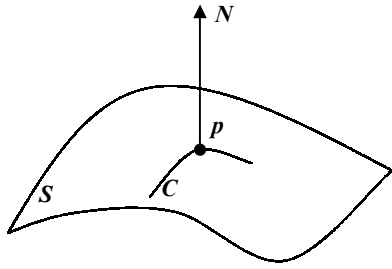
### BACKGROUND

Surface curvature is a concept rooted in differential geometry (do Carmo, 1976). The curvature of a surface intrinsically describes the local shape of that surface. In this section, we present a brief overview of the relevant definitions (the normal curvature, principal curvatures and principal directions, Gaussian and mean curvatures). Consider Fig.1. Let  $\mathbf{r}(u, v)$  be a regular surface  $\mathcal{S}$ . The point  $\mathbf{p}$  lies on the surface  $\mathcal{S}$ . And we specify the orientation of  $\mathcal{S}$  at  $\mathbf{p}$  with the unit length normal  $N$ , where  $N(u, v) = \frac{\mathbf{r}_u(u, v) \times \mathbf{r}_v(u, v)}{\|\mathbf{r}_u(u, v) \times \mathbf{r}_v(u, v)\|}$ ,

$$\mathbf{r}_u(u, v) = \frac{\partial \mathbf{r}(u, v)}{\partial u}, \mathbf{r}_v(u, v) = \frac{\partial \mathbf{r}(u, v)}{\partial v}.$$

<sup>‡</sup>Corresponding author

\*Project supported by the National Natural Science Foundation of China (No. 10371110) and the National Basic Research Program (973) of China (No. 2004CB318000)



**Fig.1** A point  $p$  is at a surface  $S$  with unit length normal  $N$ , and a curve  $C$  on the surface  $S$  passes through the point

To consider a regular curve  $C$  on  $S$  parameterized by  $\alpha(s)=r(u(s),v(s))$ , where  $s$  is the arc length of  $C$ , and with  $\alpha(0)=p$ . The number  $k_n=\langle \ddot{\alpha}(0), N \rangle$  is then called the normal curvature of  $C \subset S$  at  $p$ . All curves lying on a surface  $S$  and having at a given point  $p \in S$ , the same tangent line has at this point the same normal curvatures. This result is from the famous Meusnier theorem (do Carmo, 1976). That allows us to speak of the normal curvature along a given direction at  $p$ . We can see that we actually have an infinite set of these normal curvatures around  $p$  in every tangent direction. The maximum normal curvature  $k_1$  and the minimum normal curvature  $k_2$  are called the principal curvatures at  $p$ ; the corresponding directions  $e_1, e_2$  are called principal directions at  $p$ . The knowledge of the principal curvatures at  $p$  allows us to compute easily the normal curvatures along a given tangent line. In fact, let  $t$  be the unit length tangent vector, since  $e_1$  and  $e_2$  form an orthogonal basis of the tangent plane at  $p$ . We have

$$t=e_1\cos(\theta)+e_2\sin(\theta).$$

The normal curvature  $k_n$  along  $t$  is given by

$$k_n=k_1\cos^2(\theta)+k_2\sin^2(\theta)$$

The last expression is known classically as the Euler formula. In terms of the principal curvatures, we can write  $K_G=k_1k_2$ , which is called Gaussian curvature. This definition highlights negative surface curvatures that occur at hyperbolic patches where only one principal curvature is negative. We can also write  $H=(k_1+ k_2)/2$ , which is called mean curvature. Mean curvature gives insight to the degree of flatness of the surface.

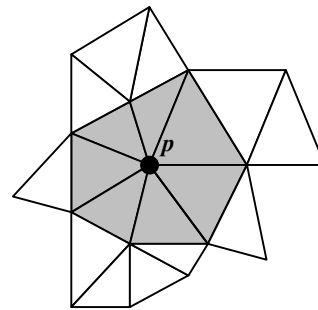
## PREVIOUS WORKS

Approximations to the surface normal and the curvatures of the smooth surface are often required when the surface is defined by a set of discrete points (or a triangular mesh) rather than a formula. Estimating the local surface geometry at a vertex  $p$  of a piecewise linear surface amounts to computing what Sander and Zucker called the augmented Darboux frame (Sander and Zucker, 1990) at  $p$ , i.e.

$$\Delta_p=(p, e_1, e_2, N, k_1, k_2).$$

Estimating the Darboux frames of a subjacent, unknown, piecewise smooth surface from a polyhedral approximation is difficult because of the inherently discrete nature of the data (Petitjean, 2002). In this section, we will introduce some of the most powerful methods for locally estimating the differential properties of a triangular mesh. Our exposition largely follows that of (Chen and Wu, 2004; Petitjean, 2002).

First, we introduce some notations. We specify a triangular mesh as the pair  $M=(V,K)$ , where  $K$  defines the topology and  $V$  defines the geometry. We can think of  $V$  as a point cloud and  $K$  as the interconnection of  $V$  to form the edges and faces of the triangles in  $M$ . And  $M$  is assumed to be an oriented and consistent triangulated surface. In other words, neighboring triangles have their normals pointing to the same side of the surface. We call the one-ring neighborhood of  $p$  the set of triangles incident to  $p$  (Fig.2). We denote by  $Neighbor(p)$  the set of one-ring neighbor vertices  $p_i$  of  $p$  and  $m$  the number of points in  $Neighbor(p)$ . The radius of a neighborhood can be recursively enlarged by defining a  $k$ -neighborhood  $Neighbor^k(p)$  as



**Fig.2** A point  $p$  and its one-ring neighborhood, which are the triangles filled with dark color

$$Neighbor^1(\mathbf{p}) = Neighbor(\mathbf{p})$$

$$Neighbor^k(\mathbf{p}) = \bigcup_{\mathbf{p}_i \in Neighbor^{k-1}(\mathbf{p})} Neighbor^{k-1}(\mathbf{p}_i) \quad (k \geq 2).$$

**The least squares solution**

Chen and Schmitt (1992) described an algorithm to estimate the principal curvatures by Euler formula. Their main idea is to choose a suitable coordinate system  $\{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2\}$  on the local tangent plane. Choose the points  $\mathbf{p}_i$  in  $Neighbor(\mathbf{p})$ , we have a group of unit length tangent vectors

$$\mathbf{t}_i = \frac{(\mathbf{p}_i - \mathbf{p}) - \langle \mathbf{p}_i - \mathbf{p}, \mathbf{N} \rangle \mathbf{N}}{\|(\mathbf{p}_i - \mathbf{p}) - \langle \mathbf{p}_i - \mathbf{p}, \mathbf{N} \rangle \mathbf{N}\|} \quad (i=1, \dots, m)$$

Let  $\theta_0$  be the angle between the vector  $\hat{\mathbf{e}}_1$  and the first principal direction  $\mathbf{e}_1$ . Let  $\theta_i$  be the angle between the tangent vector  $\mathbf{t}_i$  and  $\hat{\mathbf{e}}_1$  (Fig.3). The Euler formula become

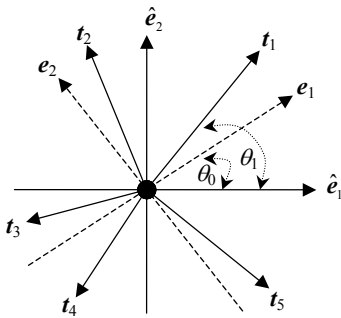
$$k_n(\mathbf{t}_i) = k_1 \cos^2(\theta_i - \theta_0) + k_2 \sin^2(\theta_i - \theta_0), \quad i=1, \dots, m.$$

Then this formula can be rewritten as

$$k_n(\mathbf{t}_i) = a \cos^2(\theta_i) + b \cos(\theta_i) \sin(\theta_i) + c \sin^2(\theta_i), \quad i=1, \dots, m.$$

The normal curvature  $k_n(\mathbf{t}_i)$  can be estimated by a circle fitting algorithm. The constants  $a, b, c$  are calculated by least square method. The principal curvatures and the angle  $\theta_0$  can be solved by their relation with  $a, b, c$

$$\begin{cases} k_1 \cos^2(\theta_0) + k_2 \sin^2(\theta_0) = a \\ 2(k_2 - k_1) \cos(\theta_0) \sin(\theta_0) = b \\ k_1 \sin^2(\theta_0) + k_2 \cos^2(\theta_0) = c \end{cases} \quad (1)$$



**Fig.3** The coordinate system on the tangent plane, the tangent directions and the principal directions

A closely related method is Martin (1998)'s algorithm. We introduce the notations,

$$\mathbf{q}_1 = \begin{pmatrix} \cos(\theta_0) \\ \sin(\theta_0) \end{pmatrix}, \quad \mathbf{q}_2 = \begin{pmatrix} -\sin(\theta_0) \\ \cos(\theta_0) \end{pmatrix},$$

$$\mathbf{W} = \begin{pmatrix} a & b/2 \\ b/2 & c \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} \mathbf{q}_1^T \\ \mathbf{q}_2^T \end{pmatrix}.$$

Now, let us rewrite Eq.(1) as

$$\mathbf{Q} \begin{pmatrix} k_1 & 0 \\ 0 & k_2 \end{pmatrix} \mathbf{Q}^T = \mathbf{W}.$$

It follows that  $k_1$  and  $k_2$  are the eigenvalues, and  $\mathbf{q}_1, \mathbf{q}_2$  are the associated unit eigenvectors of the matrix  $\mathbf{W}$ . In fact  $\mathbf{W}$  is the Weingarten curvature matrix (do Carmo, 1976) in the basis  $\{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2\}$  on the tangent plane. So the principal curvature estimation in (Meyer et al., 2003) should produce similar result.

**The integral method**

Using the notions in Section 2, let  $\mathbf{v}$  be a tangent vector with unit length, and  $\theta$  be angle between  $\mathbf{v}$  and the first principal direction  $\mathbf{e}_1$ . Considering the following symmetric 3x3 matrix

$$\mathbf{B} = \frac{1}{2\pi} \int_{-\pi}^{\pi} k_n(\mathbf{v}) \mathbf{v} \mathbf{v}^T d\theta, \quad (2)$$

where  $\mathbf{v} = \mathbf{e}_1 \cos(\theta) + \mathbf{e}_2 \sin(\theta)$ .

Taubin (1995) proved that  $\mathbf{B}$  can be factored as follows:

$$\mathbf{B} = \boldsymbol{\varepsilon}^T \begin{pmatrix} m_1 & 0 \\ 0 & m_2 \end{pmatrix} \boldsymbol{\varepsilon},$$

where  $\boldsymbol{\varepsilon} = [\mathbf{e}_1, \mathbf{e}_2]$  is the 3x2 matrix constructed by concatenating the column vectors  $\mathbf{e}_1, \mathbf{e}_2$  and  $k_1 = 3m_1 - m_2, k_2 = 3m_2 - m_1$ . That is, the eigenvalues of  $\mathbf{B}$  are 0,  $m_1, m_2$  and the corresponding eigenvectors are  $\mathbf{N}, \mathbf{e}_1, \mathbf{e}_2$ . Taubin approximated Eq.(2) as

$$\tilde{\mathbf{B}}_p = \sum_{i=1}^m w_i k_n(\mathbf{t}_i) \mathbf{t}_i \mathbf{t}_i^T, \quad (3)$$

where  $\mathbf{t}_i$  is defined in Section 3.1.

The weight  $w_i$  is chosen to be proportional to the sum of the areas of the triangles incident to both  $\mathbf{p}$  and  $\mathbf{p}_i$ , and such that  $\sum_{i=1}^m w_i = 1$ . The normal curvature  $k_n(\mathbf{t}_i)$  is estimated as

$$k_n(\mathbf{t}_i) = \frac{2\langle \mathbf{N}, \mathbf{p}_i - \mathbf{p} \rangle}{\|\mathbf{p}_i - \mathbf{p}\|} \quad (i=1, \dots, m).$$

Taubin decomposed the matrix in left side of Eq.(3) with a Householder transformation and a given rotation. Taubin's method is improved by Pege *et al.* (2002) with normal vector voting. The improved algorithm can be more robust in the presence of noise, but also become more complicated. In (Chen and Wu, 2004), the authors presented a new set of weights for Eq.(3). They obtain more accurate result.

**Other methods**

Meyer *et al.*(2003) applied the Gauss-Bonnet theorem to estimate the Gaussian curvatures, and introduced the Laplace-Beltrami operator to approximate the mean curvature. Flynn and Jain (1989) used a sphere passing through four vertices to estimate curvature. Meek and Walton (2000) asymptotically analyzed several methods and compared the result with the result by discretization and interpolation method. Goldfeather and Interrante (2004) used the normal information at the adjacent points to fit a cubic surface. Coefficients from the fitting are used in the Weingarten curvature matrix. Gatzke and Grimm (2003) systematically examined the result of calculating the curvature of surfaces represented by triangular meshes and recommended the surface fitting method (Goldfeather and Interrante, 2004) and 2-ring polynomial fitting method (Gatzke and Grimm, 2003) in the presence of noise. However the surface fitting methods are somewhat more expensive computationally. The interested reader is referred to Petitjean (2002) for the surface fitting methods. Because the surface fitting method disregards the point-to-point connectivity and only uses the 3D vertex set, we only introduce the previous approaches using the mesh topological attribute. Anyway we prefer the methods of Chen and Schmitt (1992), Meyer *et al.*(2003), Taubin (1995), which are simple.

Chen and Schmitt's method uses the circle passing through three points to estimate the normal

curvature. It is simple to image in that estimating a curve approximated by three points with a circle, can cause big errors, especially if the angle is very sharp. Wollmann (2000) used the quadratic Bèzier curve to replace the circular arc, but the normal information is not used. Taubin (1995) employed a truncated Laurent series to approximate the normal curvatures. He used two vertices and one normal vector, the adjacent normal information is not used. Karbacher and Häusler (1998) used the angle between the unit length normal vectors at two neighboring points to estimate the normal curvature, but the sign of the curvature is difficult to assign. Sacchi *et al.*(1999) also used the neighboring surface normal to approximate the curvatures, and it is also difficult to assign the sign of the curvature. In this paper, we will describe a simple formula to estimate the normal curvature with the adjacent surface normal information. At the same time we will simplify Chen and Schmitt's method.

ESTIMATING THE NORMAL CURVATURE

Now, we can describe the new method to approximate the normal curvature. With the notions in Section 2, to consider a regular curve  $C \subset S$  parameterized by  $\alpha(s)$ , where  $s$  is the arc length of  $C$ , and with  $\alpha(0)=\mathbf{p}$ . If we denote by  $\mathbf{N}(s)$  the restriction of the normal vector  $\mathbf{N}$  to the curve  $\alpha(s)$ , we have  $\langle \dot{\alpha}(s), \mathbf{N}(s) \rangle = 0$ . Hence

$$\langle \ddot{\alpha}(s), \mathbf{N}(s) \rangle = -\langle \dot{\alpha}(s), \dot{\mathbf{N}}(s) \rangle.$$

Therefore

$$k_n = \langle \ddot{\alpha}(0), \mathbf{N}(0) \rangle = -\langle \dot{\alpha}(0), \dot{\mathbf{N}}(0) \rangle = -\left\langle \frac{d\alpha}{ds} \Big|_{s=0}, \frac{d\mathbf{N}}{ds} \Big|_{s=0} \right\rangle = -\frac{\langle d\alpha, d\mathbf{N} \rangle}{ds^2} \Big|_{s=0} \quad (4)$$

Considering the definitions of the curve and the surface, we have

$$d\alpha = \frac{\partial \mathbf{r}}{\partial u} du + \frac{\partial \mathbf{r}}{\partial v} dv = d\mathbf{r} . \quad (5)$$

The parameter  $s$  is the arc length of the curve, we see that

$$ds = \|d\alpha\| = \|d\mathbf{r}\| . \quad (6)$$

From Eqs.(4)~(6) we obtain

$$k_n = -\frac{\langle d\mathbf{r}, d\mathbf{N} \rangle}{\langle d\mathbf{r}, d\mathbf{r} \rangle} = -\lim_{\|\Delta\mathbf{r}\| \rightarrow 0} \frac{\langle \Delta\mathbf{r}, \Delta\mathbf{N} \rangle}{\|\Delta\mathbf{r}\|^2}.$$

If  $\mathbf{p}_1$  with unit length surface normal  $\mathbf{N}_1$  is another different point on the surface very close to  $\mathbf{p}$ , and  $\mathbf{t}$  is the normalized projection of the vector  $\mathbf{p}_1 - \mathbf{p}$  onto the tangent plane at  $\mathbf{p}$ , the normal curvature along the tangent direction  $\mathbf{t}$  can be approximated as

$$k_n(\mathbf{t}) = -\frac{\langle \mathbf{p}_1 - \mathbf{p}, \mathbf{N}_1 - \mathbf{N} \rangle}{\|\mathbf{p}_1 - \mathbf{p}\|^2}, \quad (7)$$

where  $\mathbf{t} = \frac{(\mathbf{p}_1 - \mathbf{p}) - \langle \mathbf{p}_1 - \mathbf{p}, \mathbf{N} \rangle \mathbf{N}}{\|(\mathbf{p}_1 - \mathbf{p}) - \langle \mathbf{p}_1 - \mathbf{p}, \mathbf{N} \rangle \mathbf{N}\|}.$

Justification for this approximation of  $k_n$  is best explained with the help of Fig.4. Let  $\mathbf{p}_1, \mathbf{p}$  be the points from a sphere with radius  $R > 1$ , the osculating circle with radius  $R$  pass through the two points, a concentric circle with unit length pass through  $\mathbf{N}_1, \mathbf{N}$ . Then we have that the vectors  $(\mathbf{p}_1 - \mathbf{p}), (\mathbf{N}_1 - \mathbf{N})$  are parallel and in same direction, i.e.

$$\langle \mathbf{N}_1 - \mathbf{N}, \mathbf{p}_1 - \mathbf{p} \rangle = \|\mathbf{N}_1 - \mathbf{N}\| \|\mathbf{p}_1 - \mathbf{p}\|.$$

In the triangle  $\Delta \mathbf{p}_1 \mathbf{O} \mathbf{p}$ , it is easy to find that

$$\frac{1}{R} = \frac{\|\mathbf{N}_1 - \mathbf{N}\|}{\|\mathbf{p}_1 - \mathbf{p}\|} = \frac{\langle \mathbf{N}_1 - \mathbf{N}, \mathbf{p}_1 - \mathbf{p} \rangle}{\langle \mathbf{p}_1 - \mathbf{p}, \mathbf{p}_1 - \mathbf{p} \rangle}.$$

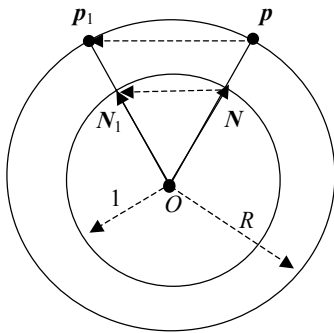


Fig.4 The concentric circles passing  $\mathbf{N}_1, \mathbf{N}$  and  $\mathbf{p}_1, \mathbf{p}$

ALGORITHM

We now consider a triangular mesh  $\mathbf{M}=(\mathbf{V},\mathbf{K})$  (Section 3) that we will look on as an approximation of an unknown smooth surface. Our goal is to estimate the principal curvatures and principal directions at the vertices of  $\mathbf{M}$ . The algorithm that we shall present is based on Chen and Schmitt's method, but we estimate normal vectors, and normal curvatures in a different way.

The first task is to estimate the normal vectors at the vertices of the triangular mesh. We denote the triangular face in  $\mathbf{M}$  as  $f$ . Since the triangular faces of the mesh are planar, each of that faces  $f_i$  has a well defined unit length normal vector  $\mathbf{N}_{f_i}$ , and the triangular mesh is oriented in such a way all these normal vectors point to the same side of the surface. It is commonly suggested to compute the normal at vertex  $\mathbf{p}$  of  $\mathbf{M}$  as a weighted average normal to the triangular faces adjacent to  $\mathbf{p}$ , i.e.

$$\mathbf{N} = \frac{\sum_{i=1}^m w_i \mathbf{N}_{f_i}}{\left\| \sum_{i=1}^m w_i \mathbf{N}_{f_i} \right\|},$$

where  $\mathbf{N}_{f_i}$  are the unit length normal to the triangles in the one-ring neighborhood of  $\mathbf{p}$ . We use the weight in (Chen and Wu, 2004), who suggested choosing the weights as

$$w_i = \frac{1}{\|\mathbf{g}_i - \mathbf{p}\|} \quad (i=1, \dots, m)$$

where  $\mathbf{g}_i$  is the center of the triangle face  $f_i$  determined as

$$\mathbf{g}_i = \sum_{\mathbf{p}_j \in f_i} \mathbf{p}_j / 3 \quad (i=1, \dots, m).$$

In a similar way, we estimate the unit length normal vector  $\mathbf{N}_i$  at the point  $\mathbf{p}_i$  in  $Neighbor(\mathbf{p})$ .

For each neighbor  $\mathbf{p}_i$  of  $\mathbf{p}$ , we define  $\mathbf{t}_i$  as the unit length projection of the vector  $\mathbf{p}_i - \mathbf{p}$  onto the tangent plane,

$$\mathbf{t}_i = \frac{(\mathbf{p}_i - \mathbf{p}) - \langle \mathbf{p}_i - \mathbf{p}, \mathbf{N} \rangle \mathbf{N}}{\|(\mathbf{p}_i - \mathbf{p}) - \langle \mathbf{p}_i - \mathbf{p}, \mathbf{N} \rangle \mathbf{N}\|} \quad (i=1, \dots, m).$$

We approximate the normal curvature  $k_n(\mathbf{t}_i)$  using Eq.(7)

$$k_n(\mathbf{t}_i) = -\frac{\langle \mathbf{p}_i - \mathbf{p}, \mathbf{N}_i - \mathbf{N} \rangle}{\langle \mathbf{p}_i - \mathbf{p}, \mathbf{p}_i - \mathbf{p} \rangle} \quad (i=1, \dots, m).$$

We select the maximum normal curvature in  $k_n(\mathbf{t}_i)$  ( $i=1, \dots, m$ ). Supposing  $k_n(\mathbf{t}_{id})$  is the maximum, we can choose a special coordinate system  $\{\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2\}$  on the tangent plane at  $\mathbf{p}$ :

$$\hat{\mathbf{e}}_1 = \mathbf{t}_{id}, \quad \hat{\mathbf{e}}_2 = \frac{\hat{\mathbf{e}}_1 \times \mathbf{N}}{\|\hat{\mathbf{e}}_1 \times \mathbf{N}\|}.$$

According to the Chen and Schmitt' method, we obtain the following equations

$$k_n(\mathbf{t}_i) = a \cos^2(\theta_i) + b \cos(\theta_i) \sin(\theta_i) + c \sin^2(\theta_i), \quad i=1, \dots, m.$$

where  $\theta_i$  is the angle between  $\mathbf{t}_i$  and  $\hat{\mathbf{e}}_1$ . We let  $a = k_n(\mathbf{t}_{id})$ , and then we only have to resolve the remaining constants  $b$  and  $c$ , which are easy to compute by least square method. To calculate the following equations is just as easy

$$a_{11} = \sum_{i=1}^m \cos^2(\theta_i) \sin^2(\theta_i), \quad a_{12} = \sum_{i=1}^m \cos(\theta_i) \sin^3(\theta_i)$$

$$a_{21} = a_{12}, \quad a_{22} = \sum_{i=1}^m \sin^4(\theta_i)$$

$$a_{13} = \sum_{i=1}^m (k_n(\mathbf{t}_i) - a \cos^2(\theta_i)) \cos(\theta_i) \sin(\theta_i)$$

$$a_{23} = \sum_{i=1}^m (k_n(\mathbf{t}_i) - a \cos^2(\theta_i)) \sin^2(\theta_i).$$

It follows that

$$a = k_n(\mathbf{t}_{id}), \quad b = \frac{a_{13}a_{22} - a_{23}a_{12}}{a_{11}a_{22} - (a_{12})^2}, \quad c = \frac{a_{11}a_{23} - a_{12}a_{13}}{a_{11}a_{22} - (a_{12})^2}.$$

From the above equations, we immediately obtain

$$K_G = ac - b^2/4, \quad H = (a+c)/2, \quad k_{1,2} = H \pm \sqrt{H^2 - K_G}.$$

If  $k_1$  and  $k_2$  are different, we first calculate the angle

$$\theta_0 = 0.5 \arcsin[b/(k_2 - k_1)],$$

and then we can estimate the principal directions

$$\mathbf{e}_1 = \cos(\theta_0)\hat{\mathbf{e}}_1 + \sin(\theta_0)\hat{\mathbf{e}}_2,$$

$$\mathbf{e}_2 = \cos(\theta_0)\hat{\mathbf{e}}_2 - \sin(\theta_0)\hat{\mathbf{e}}_1.$$

The above algorithm is easy to extend to  $k$ -ring neighborhood (defined in Section 3), by just replacing the set  $Neighbor(\mathbf{p})$  with the set  $Neighbor^k(\mathbf{p})$ . In the following paper we use the algorithm with one-ring neighborhood if there is no special reason for not using it.

### COMPARISON OF RESULTS

We have implemented the proposed algorithm in C++ code. Since the curvature calculation is a local operation on a mesh, we can use simple surfaces to test the algorithms. First we use two basic shapes for our test cases: a sphere and a torus. The test cases built from these surfaces are split into two cases: those that have points on the surface with a global regular triangulation of the parameter space (Fig.5a and Fig.7a), and those that have points on the surfaces but with global random triangulation by jittering the locations of the interior points within the parameter space (Fig.5b and Fig.7b). Using the adjacent normal information in our method makes the result more accurate, especially in the regular cases, the result improved very dramatically. Here we present comparison results in Figs.6~8, we test Taubin's method and our method to estimate the normal curvature and Gaussian curvature on the unit sphere and the torus.

The  $a$ -axis in Figs.6~8 represents the index on the triangular mesh. In the tests, we estimate the error by the formula:

$$\text{Error of normal curvatures} = |\hat{k}_n - k_n|/k_n,$$

where  $k_n$  is the true normal curvature, and  $\hat{k}_n$  is the estimated normal curvature.

The error of Gaussian curvatures is  $|\hat{K}_G - K_G|$ , where  $K_G$  is the true Gaussian curvature on the surface, and  $\hat{K}_G$  is the estimated Gaussian curvature. We use

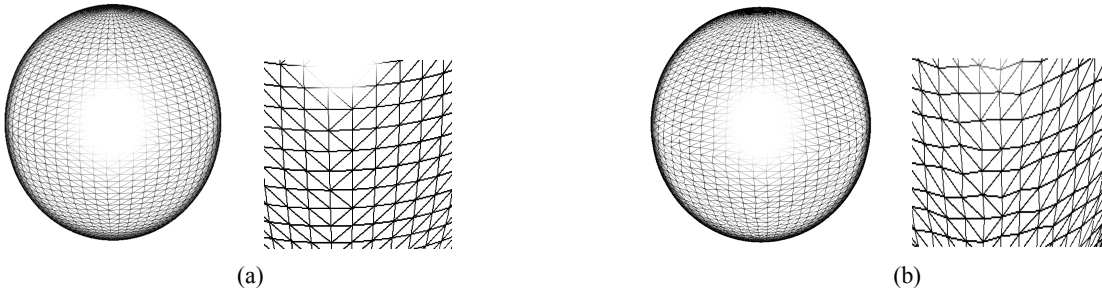


Fig.5 The sphere with regular triangulation (a) and with random triangulation (b), on the left is the global mesh and on the right is a part of the mesh zoom in

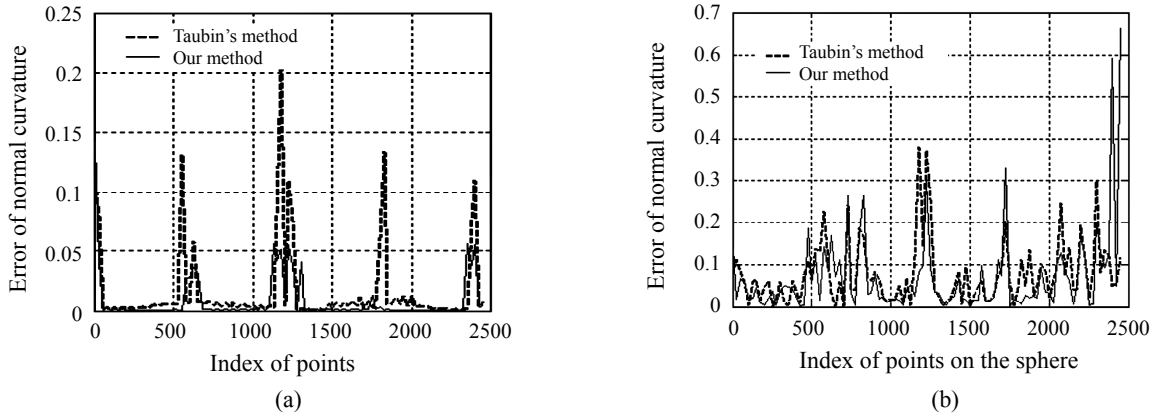


Fig.6 The error of the normal curvature on the sphere with regular triangulation (a) and with random triangulation (b)

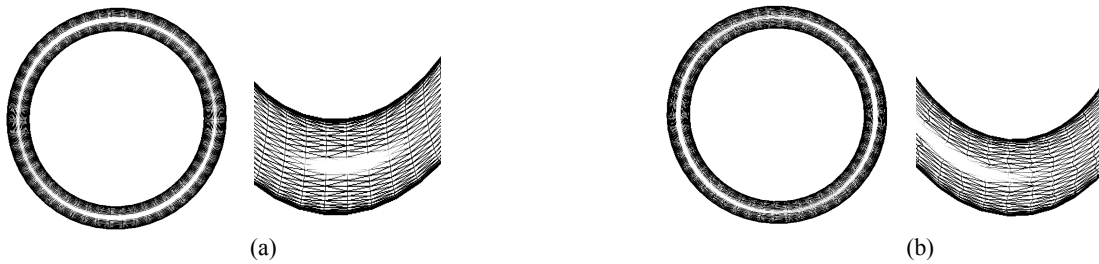


Fig.7 The torus with regular triangulation (a) and with random triangulation (b) on the left is the global mesh and on the right is a part of the mesh zoom in

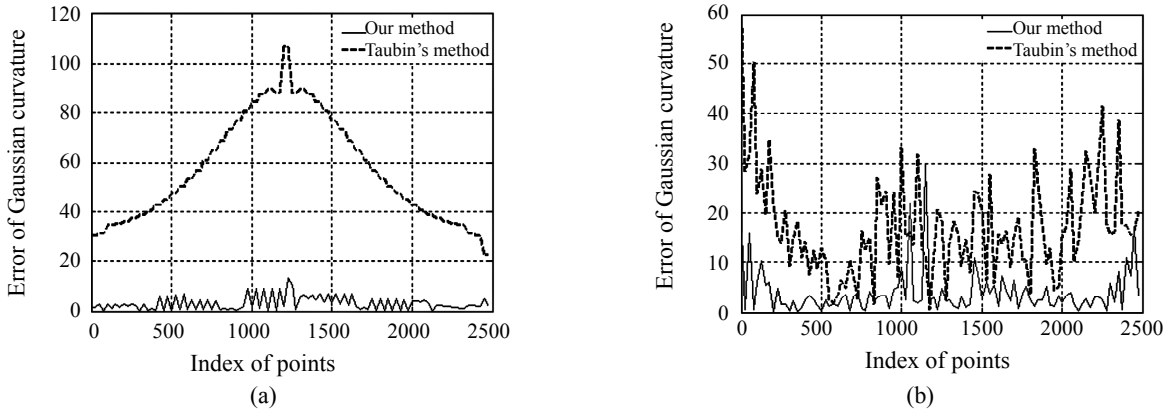
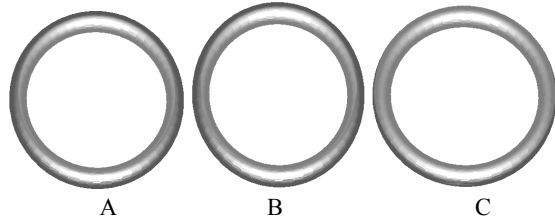
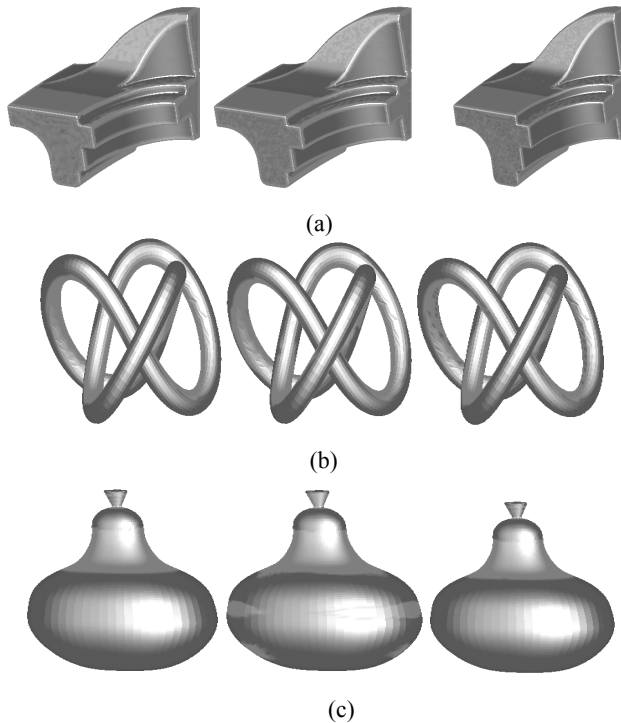


Fig.8 The error of the Gaussian curvature on the torus with regular triangulation (a) and with random triangulation (b)

curvatures to set the RGB color, and then draw colors on the mesh (Figs.9~11), Red color  $r = \exp(-wK_G)$ , Green color  $g=0$ , Blue color  $b=1-r$ . To demonstrate the algorithm capabilities, we also experimented with a variety of data from scanned data: a Fandisk



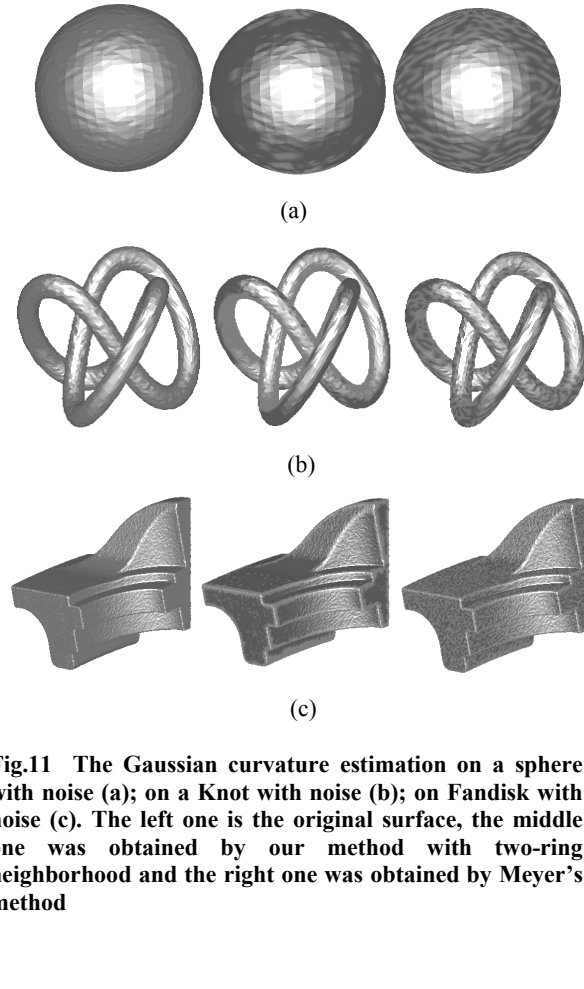
**Fig.9** The Gaussian curvature on the torus. Case A is the true Gaussian curvatures on the mesh, Case B is our estimation of the Gaussian curvature on the random mesh, Case C is Taubin's method estimation of the Gaussian curvature on the random mesh



**Fig.10** The mean curvature estimation on the Fandisk (a); on Knot (b); on Pear (c). The left Fandisk was obtained by our method, the right one was obtained by Meyer's method and the middle one was obtained by Taubin's method. The Fandisk model is an example of free-form CAD model with sharp edges (The original triangular mesh in (a) is available at <http://research.microsoft.com/research/graphics/hoppe/>; the original discrete points in (c) are available at <http://www.cs.princeton.edu/gfx/proj/sugcon/models/>)

(Fig.10a), a Knot (Fig.10b) and a Pear (Fig.10c).

We also compared our method with Meyer's method (Meyer *et al.*, 2003) in Figs.10~11. When surface noise is very small (Fig.10), our result is similar to the result by Meyer's method. If it is known a priori that there is no noise in the mesh data, Meyer's method is a logical choice. However Meyer's method is not easy to extend to two-ring neighborhood. When the noise in the surface data (Fig.11) is large, our method with two-ring neighborhood is better.



**Fig.11** The Gaussian curvature estimation on a sphere with noise (a); on a Knot with noise (b); on Fandisk with noise (c). The left one is the original surface, the middle one was obtained by our method with two-ring neighborhood and the right one was obtained by Meyer's method

## CONCLUSION

We have introduced a new method to estimate normal curvature and a novel algorithm to compute the principal curvatures on a triangular mesh. During the normal curvature estimation, we take more advantage of the known information on the surface, i.e. the adjacent normal, we obtain more accurate result.

In the estimation of principal curvatures, we choose the direction associated with the maximum normal curvature in the one-ring neighborhood as one coordinate axis. We only need to resolve only two unknown constants in our method. The impact of noise and mesh irregularity are significant for one-ring neighborhood in the curvatures calculation. If there is noise in the data and mesh triangulation is ill-behaved, we suggest the readers using two-ring neighborhood or higher-ring neighborhood to estimate the curvatures. Our algorithm is easy to extend to the two-ring neighborhood. There are many important directions for future work in this area, for example surface smoothing, segmentation, features identification, etc.

## References

- Alrashdan, A., Motavali, S., Fallahi, B., 2000. Automatic segmentation of digitized data for reverse engineering applications. *IIE Trans.*, **32**:59-69.
- Chen, X., Schmitt, F., 1992. Intrinsic Surface Properties from Surface Triangulation. Proceedings of the European Conference on Computer Vision, p.739-743.
- Chen, G., Wu, Y., 2004. Estimating normal vectors and curvatures by centroid weights. *Computer Aided Geometric Design*, **21**:447-458.
- do Carmo, M.P., 1976. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, NJ.
- Flynn, P.J., Jain, A.K., 1989. On Reliable Curvature Estimation. Proceeding of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, p.110-116.
- Gatzke, T., Grimm, C., 2003. Improved Curvature Estimation on Triangular Meshes. Eurographics Symposium on Geometry Processing, p.57-67.
- Goldfeather, J., Interrante, V., 2004. A novel cubic-order algorithm for approximating principal direction vectors. *ACM Trans. on Graphics*, **23**(1):45-63.
- Karbacher, S., Häusler, G., 1998. A New Approach for Modeling and Smoothing of Scattered 3D Data. SPIE Proceedings for Three-Dimensional Image Capture and Application, **3313**:168-177.
- Martin, R., 1998. Estimation of principal curvatures from range data. *Int. J. of Shape Modeling*, **3**(4):99-109.
- Meek, D.S., Walton, D.J., 2000. On surface normal and Gaussian curvature approximations given data sampled from a smooth surface. *Computer Aided Geometric Design*, **17**:521-543.
- Meyer, M., Desbrun, M., Schröder, P., Barr, A.H., 2003. Discrete Differential Geometry Operators for Triangulated 2-Manifolds. In: Hege, H.C., Polthier, K.(Eds), *Visualization and Mathematics III*. Springer Verlag, Heidelberg, Germany, p.35-57.
- Monga, O., Benayoun, S., Faugeras, O., 1992. From Partial Derivatives of 3D Density Images to Ridge Lines. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, p.354-359.
- Page, D.L., Sun, Y., Koschan, A.F., Paik, J., Abidi, M.A., 2002. Normal vector voting: crease detection and curvature estimation on large, noisy meshes. *Graphical Models*, **64**:199-229.
- Petitjean, S., 2002. A survey of methods for recovering quadratics in triangular meshes. *ACM Computing Surveys*, **2**(34):1-61.
- Sacchi, R., Poliakoff, J., Thomas, P., Häfele, K.H., 1999. Curvature Estimation for Segmentation of Triangulated Surfaces. Proc. of 2nd International Conference on 3D Imaging and Modeling. Ottawa, Canada, p.536-543.
- Sander, P.T., Zucker, S.W., 1990. Inferring surface trace and differential structure from 3D images. *IEEE Trans. Pattern Anal. Mach. Intell.*, **12**(9):833-854.
- Taubin, G., 1995. Estimating the Tensor of Curvature of a Surface from a Polyhedral Approximation. Proceedings of the Fifth International Conference on Computer Vision, p.902-907.
- Wollmann, C., 2000. Estimation of the principle curvatures of approximated surfaces. *Computer Aided Geometric Design*, **17**:621-630.
- Yang, M., Lee, E., 1999. Segmentation of measured point data using a parametric quadric surface approximation. *Computer-Aided Design*, **31**:449-457.