Estimating normal vectors and curvatures by centroid weights

Sheng-Gwo Chen *, Jyh-Yang Wu

Department of Mathematics, National Chung Cheng University, Chia-Yi 621, Taiwan, ROC

Received 20 February 2004; received in revised form 20 February 2004; accepted 20 February 2004
Available online 10 March 2004

Abstract

The tensors of curvature play an important role in differential geometry. In surface theory (1990), it is determined by the derivative of unit normal vectors of tangent spaces of the underlying surface. However, every geometric object in computation is a discrete model. We can only approximate them. In estimating the curvature on polyhedral surfaces, how to approximate normal vectors is a crucial step. Chen and Schmitt (1992) and Taubin (1995) described two simple methods to estimate the principal curvatures. The weights they choose is related to the triangle areas. But this choice not the best. Max (1999) presented a new kind of weight to estimate the normal vector.

In this paper, we will present a new set of weights from duality and gravity. We choose the centroid weights to approximate normal vectors and estimate the principle curvatures. This will lead to a better estimation than the area-weights. Our results are also comparable with Max’s weights.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Centroid weight; Normal vector; Principal curvature

1. Introduction

The tensors of curvature on differentiable manifolds are important invariants in differential geometry. When the dimension is two, the tensors of curvature are determined by Gaussian curvatures and Mean curvatures. In fact, Gaussian curvatures and Mean curvatures come from the determinant and trace of the derivative of the normal vectors, respectively. From the theory of linear algebra, we know these values depend on the eigenvalues of the derivative of the normal vectors. Since the 1990s, people presented many methods and different ideas to estimate the Gaussian curvatures and Mean curvatures on various kinds of surfaces (1989), but only a few people investigated the tensors of curvature from the viewpoint of principal curvatures on polyhedral surfaces.

* Corresponding author.
E-mail addresses: csg.chen@msa.hinet.net (S.-G. Chen), jywu@math.ccu.edu.tw (J.-Y. Wu).
In (1992) Chen and Schmitt described an algorithm to estimate principal curvatures and principal
directions from polyhedral surfaces. Their main idea lies in a clever choice of suitable coordinates. This
allow them to solve the Euler formula on this coordinate system by least square method. Their result is
quite good. However, the form of principal curvature is complicated.

In (1995) Taubin improved Chen and Schmitt’s algorithm. He used integration to avoid the
trigonometric functions in Euler formula, and used summation instead of the integration in Euler formula.
This new idea improves the form of principal curvatures. The error in Taubin’s algorithm comes from
the following three parts: (1) the evaluation of the normal vector, (2) the Taylor expansion of normal
curvatures and (3) the estimate for the integration of Euler formula. In (2001) Page, Koschan, Paik, and
Abidi described a new reformulation of Taubin’s algorithm. They used the method of normal voting to
evaluate the curvatures on triangle mesh. Even though their method slightly improves Taubin’s algorithm,
their form and analysis are still complicated. In (1999) Max’s weights improve this estimation method
effectively.

2. Estimation of curvatures on surfaces

Before we discuss the approximation of the normal curvature, let us recall the Euler formula.

Theorem (Do Carmo, 1976). The tensors of curvature on regular surfaces in \( \mathbb{R}^3 \) always satisfies the
Euler formula for normal curvatures. Given two principal directions, \( e_1, e_2 \), every normal curvature \( k_{T_0} \)
along the tangent vector \( T_0 \) satisfies

\[
k_{T_0} = k_1 \cos^2 \theta + k_2 \sin^2 \theta,
\]

where \( k_1, k_2 \) are principal curvatures and \( \theta \) is the angle between the tangent vector \( T_0 \) and the coordinate
vector \( e_1 \).

An important task in CAGD is to define and estimate differential objects like the tangent vector,
normal vector and curvatures in discrete terms. Here, we present some natural methods to approximate
these values.

First, we introduce some notation. Consider a triangulated surface \( S \) represented as a pair of lists
\( \{V, F\} \), where \( V = \{v_i \mid 1 \leq i \leq n_V\} \) is the list of vertices and \( F = \{f_k \mid 1 \leq k \leq n_F\} \) is the list
of triangles. We assume \( S \) is oriented and consistent. Choose a vertex \( v_i \in S \). We say \( v_j \) is a neighbor of \( v_i \)
if there exists a triangle \( f_k \) such that \( v_i \in f_k \) and \( v_j \in f_k \). Denote the area of the triangle \( f_k \) by \( |f_k| \). The
set of triangles that contain the vertex \( v_i \) will be denoted by \( F_i \). If the triangle \( f_k \) is in \( F_i \), we say \( f_k \) is
incident to \( v_i \), and the number of element of \( F_i \) will denoted by \( |F_i| \).

Next, let we estimate the unit normal vector \( N_{v_i} \) at the vertex \( v_i \) in \( S = \{V, F\} \) by

\[
N_{v_i} = \frac{\sum_{f_k \in F_i} \omega_k N_{f_k}}{\|\sum_{f_k \in F_i} \omega_k N_{f_k}\|},
\]

where the weights \( \omega_k \) is positive, and \( N_{f_k} \) is the unit normal vector of the triangle \( f_k \).

The unit tangent vector \( T_{ij} \) is obtained from the projection of the vector \( v_j - v_i \) onto the tangent plane
\( \langle N_{v_i} \rangle^\perp \).

\[
T_{ij} = \frac{\langle N_{v_i}, v_j - v_i \rangle N_{v_i} - (v_j - v_i)}{\|\langle N_{v_i}, v_j - v_i \rangle N_{v_i} - (v_j - v_i)\|}.
\]
Finally, we need to evaluate the normal curvature \( k_{ij} \) along the tangent vector \( T_{ij} \) at the vertex \( v_i \). For this, let us consider a curve \( x(s) \) to be a normal section on surface \( S \) in \( \mathbb{R}^3 \) by arc-length parameter. From the Taylor expansion of \( x(s) \), we have

\[
x(s) = x(0) + x'(0)s + \frac{x''(0)s^2}{2!} + O(s^3).
\]

Assume \( p = x(0) \) and \( T = x'(0) \). Because \( x(s) \) is a normal section with arc-length parameter, \( x''(0) = k_p(T)N \) where \( N \) is the unit normal vector on \( S \) at \( x(0) \). Hence,

\[
x(s) = p + Ts + \frac{1}{2}k_p(T)Ns^2 + O(s^3).
\]

This yields

\[
2\langle N, x(s) - p \rangle = k_p(T)s^2 + O(s^3).
\]

The Taylor expansion gives again,

\[
\langle x(s) - p, x(s) - p \rangle = \langle Ts, Ts \rangle + O(s^3).
\]

Therefore, we have

\[
\|x(s) - p\|^2 = s^2 + O(s^3).
\]

From Eqs. (1) and (2), we obtain

\[
\frac{2\langle N, (x(s) - p) \rangle}{\|x(s) - p\|^2} = k_p(T) + O(s).
\]

Hence,

\[
k_p(T) = \lim_{s \to 0} \frac{2\langle N, (x(s) - p) \rangle}{\|x(s) - p\|^2}.
\]

Set \( q = x(s) \). When \( s \) approaches to 0, we have the normal curvature \( k_p(T) \)

\[
k_p(T) \approx \frac{2\langle N, (q - p) \rangle}{\|q - p\|^2}.
\]

Hence we can approximate the normal curvature \( k_{ij} \) along the tangent vector \( T_{ij} \) at \( v_i \) by Eq. (3). Hence,

\[
k_{ij} = \frac{2\langle N_{v_i}, v_j - v_i \rangle}{\|v_j - v_i\|^2}.
\]

3. Survey of old algorithm

3.1. Chen and Schmitt’s algorithm (1992)

In (1992) Chen and Schmitt described an algorithm to estimate the principal curvatures by Euler formula. Their main idea is to choose a suitable coordinate system \( \{r_1, r_2\} \) such that the formula become \( k_{T_0} = k_1 \cos^2(\theta + \theta_0) + k_2 \sin^2(\theta + \theta_0) \). This formula can be rewritten as

\[
k_{T_0} = C_1 \cos^2 \theta + C_2 \cos \theta \sin \theta + C_3 \sin^2 \theta
\]
for some constants $C_1, C_2, C_3$ and $\theta$ is the angle between the tangent vector $T_\theta$ and the first coordinate vector $r_1$. Then they choose $r_1 = Ti_1$ and estimate the constants $C_1, C_2, C_3$ by least square method:

$$\min \sum_j \left| (C_1 \cos^2 \theta_{ij} + C_2 \sin \theta_{ij} \cos \theta_{ij} + C_3 \sin^2 \theta_{ij}) - k_{ij} \right|^2$$

where $\theta_{ij}$ is the angle between $r_1$ and $T_{ij}$.

The principal curvatures can be solved from the constants $C_1, C_2, C_3$ via the relations

$$\begin{cases}
    k_1 \cos 2\theta_0 + k_2 \sin 2\theta_0 = C_1, \\
    2(k_2 - k_1) \cos \theta_0 \sin \theta_0 = C_2, \\
    k_1 \sin 2\theta_0 + k_2 \cos 2\theta_0 = C_3.
\end{cases}$$

Therefore, the principal curvatures $k_1, k_2$ can be obtained and principal directions also can be computed by

$$\begin{align*}
    T_1 &= \cos(-\theta_0)r_1 + \sin(-\theta_0)r_2, \\
    T_2 &= \sin(-\theta_0)r_1 + \cos(-\theta_0)r_2,
\end{align*}$$

where $(r_1, r_2)$ is the chosen coordinate system.

### 3.2. Taubin’s algorithm (1995)

Although Chen and Schmitt’s idea is very simple, the formula for principal curvatures are complicated. In (1995) Taubin present a simpler algorithm to estimate these principal curvatures and principal directions on polyhedral surfaces. He integrated the Euler formula from $-\pi$ to $\pi$, and obtained matrix $M_{vi} = \frac{1}{2\pi} \int_{-\pi}^{\pi} k_{vi}(T_\theta)T_\theta T_\theta^T d\theta$. Moreover,

$$M_{vi} = T^T \begin{pmatrix} m_{11}^{vi} & m_{12}^{vi} \\ m_{21}^{vi} & m_{22}^{vi} \end{pmatrix} T$$

where $T = [e_1, e_2]$ is the $2 \times 2$ matrix given by principal directions.

From this formula, we get $m_{12}^{vi} = m_{21}^{vi} = 0$, $m_{11}^{vi} = \frac{1}{8} k_1^{vi} + \frac{4}{8} k_2^{vi}$, and $m_{22}^{vi} = \frac{1}{8} k_1^{vi} + \frac{3}{8} k_2^{vi}$. Thus we have the principal curvatures

$$\begin{align*}
    k_1^{vi} &= 3m_{11}^{vi} - m_{22}^{vi}, \\
    k_2^{vi} &= 3m_{22}^{vi} - m_{11}^{vi}.
\end{align*}$$

Taubin approximated normal vectors by choosing the weight proportional to the areas of triangle surfaces:

$$N_{vi} = \frac{\sum_{f_k \in F_i} |f_k| N_{f_k}}{\| \sum_{f_k \in F_i} |f_k| N_{f_k} \|}.$$

Once those values are obtained, the only problem in Taubin’s algorithm is to estimate the integrate $M_{vi}$. Taubin’s idea is to use

$$\tilde{M}_{vi} = \sum_{v_j \in V_i} w_{ij} k_{ij} T_{ij} T_{ij}^T$$
to approximate the matrix $M_{vi}$, where the weight $w_{ij}$ is proportional to the sum of the triangle areas of all triangles that are incident to both $v_j$ and $v_i$.

In Taubin’s paper, the matrix $\tilde{M}_{vi}$ can be transferred to the form

$$\begin{pmatrix}
0 & 0 & 0 \\
0 & \tilde{m}_{11} & \tilde{m}_{12} \\
0 & \tilde{m}_{21} & \tilde{m}_{22}
\end{pmatrix}.$$ 

The matrix $(\tilde{m}_{11} \quad \tilde{m}_{12} \quad \tilde{m}_{21} \quad \tilde{m}_{22})$ can be diagonalized by rotation angle $\theta$ and obtain the eigenvalue $\tilde{T}_1$ and $\tilde{T}_2$ of matrix $\tilde{M}_{vi}$. Then the corresponding principal directions are

$$\begin{align*}
T_1 &= \cos \theta \cdot \tilde{T}_1 + \sin \theta \cdot \tilde{T}_2, \\
T_2 &= \sin \theta \cdot \tilde{T}_1 + \cos \theta \cdot \tilde{T}_2,
\end{align*}$$

where $\tilde{T}_1$, $\tilde{T}_2$ are eigenvectors of $\tilde{M}_{vi}$.

4. The centroid-weight algorithm

The algorithm that we shall present is based on Taubin’s algorithm. However, we shall estimate normal vectors, tangent vectors, and normal curvatures in a different way by using centroid weights due to the following observation.

Observation. In the formula of the normal vector $N_{vi}$ with area-weights, if two triangles $f_{k1}$ and $f_{k2}$ have the same normal vectors and equal areas, then they will have equal contributions for the resulting normal vector $N_{vi}$, see Fig. 1. However, we know from the theory of curvatures of regular surfaces, $f_{k1}$ and $f_{k2}$ should have different effect on normal vectors and also on curvatures estimation, since the farthest point $p_1$ in $f_{k1}$ is closer to $v_i$ an the farthest point $p_2$ in $f_{k2}$.

This seems to indicate that the area-weights need to be improved to reflect this observation. Therefore, we try to search for better weights $\omega$ in order to improve the area weights. We use the viewpoint of duality. Consider the neighbors of a vertex $v$ as in Fig. 2.

Fig. 1. If $|f_{k1}| = |f_{k2}|$, they have the same effects under area-weight.
We first represent the triangle $f_k$ by a point $g_k$ in $f_k$. The natural candidate for $g_k$ is the centroid. Since the product $k d^2$ of the curvature $k$ and the distance $d$ is a scaling variant, this suggests that we can choose the weights:

$$\omega_k = \frac{1}{\|g_k - v_i\|^2}.$$  

Note also that $g_k$ is nothing but the center of mass the triangle $f_k$ and it is determined by

$$g_k = \frac{\sum_{v_j \in f_k} v_j}{3}.$$  

The new weight

$$\omega_k = \frac{1}{\|g_k - v_i\|^2}$$

is called the centroid-weights to reflect its mathematical or physical meaning. Indeed, the centroid weights also work for any polyhedron models, not just for triangular meshes.

The centroid weight we shall use is proportional to the square of the inverse of the distance from vertex $v_i$ to the centroid point $g_k$ of the triangle $f_k$. Namely, we give

$$\omega_k = \frac{1}{\|g_k - v_i\|^2}$$

where $g_k = (\sum_{v_j \in f_k} v_j)/3$.

Then, we have the normal vector

$$N_{v_i} = \frac{\sum_{f_k \in F} \omega_k N_{f_k}}{\|\sum_{f_k \in F} \omega_k N_{f_k}\|}.$$  

the tangent vector

$$T_{ij} = \frac{\langle N_{v_j}, v_j - v_i \rangle N_{v_j} - (v_j - v_i)}{\|\langle N_{v_j}, v_j - v_i \rangle N_{v_j} - (v_j - v_i)\|}$$

and normal curvatures

$$k_{ij} = \frac{2\langle N_{v_j}, v_j - v_i \rangle}{\|v_j - v_i\|^2}.$$  

Finally, we also estimate the matrix $\tilde{M}_{v_i}$ in Taubin’s algorithm by using the centroid weights

$$\omega_{ij} = \frac{\sum_{f_k \in (F^i \cap F^j)} \frac{1}{\|g_k - v_i\|^2}}{\sum_{v_i \in V^i} (\sum_{f_k \in (F^i \cap F^j)} \frac{1}{\|g_k - v_i\|^2})}.$$
and get
\[ \tilde{M}_v = \sum_{v_j \in V_v} w_{ij} k_{ij} T_{ij} T_{ij}^t. \]

After the method of diagonalization, we yield the principal curvatures and their corresponding directions as in the Taubin’s forms by replacing area weights by centroid weights.

5. Computational results

In Chen, Schmitt and Taubin’s algorithms, we can rewrite the formula to estimate the normal vector \( N_v \) and the matrix \( \tilde{M}_v \) more generally.

\[
N_v = \frac{\sum_{f_k \in F_v} w_k N_{f_k}}{\|\sum_{f_k \in F_v} w_k N_{f_k}\|},
\]

\[
\tilde{M}_v = \sum_{v_j \in V_v} w_{ij} T_{ij} T_{ij}^t
\]

with \( \omega_k > 0 \) and \( \omega_{ij} > 0 \)

Hence, in order to obtain good estimations of \( N_v \) and \( \tilde{M}_v \), the key problem is to choosing good weights in these formulas. In their algorithms, they choose these weights proportional to the area of triangle surfaces. However, there exists a problem when two triangles with the same areas have different shapes as pointed out in the observation. In fact, they should have different effects on the estimation of normal vectors and principal curvatures. However, such two triangles will contribute equally in the area-weight methods.

To improve this, the centroid weights \( w_k \) provide a better choice. Especially, when we evaluate the normal vector by using the centroid weight, the result improves very dramatically. Here we present some computational results. In Figs. 3–6, we test Taubin’s method and the centroid method to estimate the normal vector and Gaussian curvature on the unit sphere and the torus. The \( x \)-axis in Figs. 3 and 4 represents the index of vertex on these polyhedrons which are permuted by the error of the centroid method. Let \( \{S_i\}_{i=1}^n \) be a set of polyhedron which come from the unit sphere (the left of Fig. 3 and Fig. 4) and torus (the right of Fig. 4). The number of vertex on \( S_i \) is increasing when \( i \) increases. In Figs. 5 and 6, we estimate the normal vector and Gaussian curvature on \( \{S_i\}_{i=1}^n \) and observe average error for each \( S_i \).

The \( x \)-axis in Figs. 5 and 6 represents the index \( i \) of \( S_i \). Obviously, the centroid weight method is more stable than Taubin’s method. In these tests, we estimate the error by the formula:

\[
\text{Error of Normal Vector} = 1.0 - \frac{\langle N_v, N \rangle}{\|N_v\| \|N\|}
\]

and

\[
\text{Error Curvature} = \frac{|k - k_v|}{k}
\]

In (1999) Max presented several kinds of weight to estimate the normal vector. Here, let us denote these weights by: “\( \omega_i = \theta_i \)” is the angle weight, “\( \omega_i = \frac{1}{|V_i||V_{i+1}|} \)” is the edge weight, “\( \omega_i = \frac{1}{\sqrt{|V_i||V_{i+1}|}} \)” is
Fig. 3. The error of sphere with 24 vertices.

Fig. 4. The error of sphere with 760 vertices.
the sqrt_edge weight, \( \omega_i = \frac{\sin \theta_i}{|V_i| V_{i+1}} \) is the Max's weight and \( \omega_i = 1 \) is unweight. The details of these weights can be found in Max’s paper (Max, 1999). In Figs. 7–9, we test these estimates of the normal vectors and Gaussian curvature in random polynomial graphs, that is

\[
x(u, v) = \left( u, v, f(u, v) \right)
\]

with

\[
f(u, v) = \sum_{i=2}^{m} \sum_{j=0}^{i} c_{i,j} u^i v^j,
\]

where the coefficient \( c_{i,j} \) is a random number in the interval \([-0.5, 0.5]\). We estimate the normal vector and Gaussian curvature at the point \( p = x(0, 0) \) on 65,535 random surfaces. For each surface, we choose 50,000 random partitions into triangles surrounding \( p \) and compute the average of these errors, then compare those average numbers in Figs. 6 and 7. The set of neighbors of the point \( p \) in our tests is constructed by

\[
\left\{ (r_i \cos \theta_i, r_i \sin \theta_i, f(r_i \cos \theta_i, r_i \sin \theta_i)) \mid i \in \{1, 2, \ldots, n\} \right\}
\]
where \( \{\theta_0, \theta_1, \ldots, \theta_n\} \) is a partition of \([0, 2\pi]\) such that \( \theta_{j+1} - \theta_j \leq \pi \) for each \( j \in \{0, 1, \ldots, n-1\} \) and \( r_i, \theta_j, n, m \) are random positive numbers with

\[
\begin{align*}
    r_i &\in (0, 0.01], \quad \theta_j \in [0, 2\pi], \\
    n &\in \{3, 4, \ldots, 9\}, \quad m \in \{2, 3, \ldots, 6\}.
\end{align*}
\]

Fig. 6 describes the histogram of the RMS errors. Obviously, the centroid weights and Max’s weight (Max, 1999) are better than other kinds of weights.

The centroid weights, Max’s weights and 1/edges weights have the same statistical style, see Fig. 7. In Fig. 9, the \( x \)-axis represent the value \( r \) which controls the ratio of the partition of \([0, 2\pi]\) and the partition of \([0, 2\pi]\) is determined by:
Fig. 8. The graph of the error of Gaussian curvature of random polynomial surfaces.
First, we choose a random partition \{p_i\}_{i=0}^n of the close interval [0, 1] with \(p_i < p_j\) as \(i < j\). Then the partition is \{\theta_i\}_{i=0}^n with \(\theta_i = \frac{2\pi(0.01r + p_i)}{0.01r + 1}\) for each \(i = 1, 2, \ldots, n\) and \(\theta_0 = 0\).

When \(r\) approaches \(\infty\), the partition is uniform. The centroid weights better than the others weights when the partition of \([0, 2\pi]\) becomes more and more uniform, see Fig. 9.

References