# New Computation of Normal Vector and Curvature 

Hyoung-Seok Kim<br>Department of Multimedia Engineering, Dong-Eui University<br>995 Eomgwangro, Busanjingu, Busan, 614-714, KOREA<br>hskim@deu.ac.kr<br>Ho-Sook Kim<br>Division of Advanced Computer Information, Dong-Eui Institute of Technology<br>152 Yangji 5 ro, Busanjingu, Busan, 614-715, KOREA<br>hskim@dit.ac.kr


#### Abstract

The local geometric properties such as curvatures and normal vectors play important roles in analyzing the local shape of objects. The result of the geometric operations such as mesh simplification and mesh smoothing is dependent on how to compute the normal vectors and the curvatures of vertices, because there are no exact definitions of the normal vector and the discrete curvature in meshes. Therefore, the discrete curvature and normal vector estimation play the fundamental roles in the fields of computer graphics and computer vision. In this paper, we propose new methods for computing normal vector and curvature well, which are more intuitive than the previous methods. Our normal vector computation algorithm is able to compute the normal vectors more accurately and is available to meshes of arbitrary topology. It is due to the properties of local conformal mapping and the mean value coordinates. Secondly, we point out the fatal error of the previous discrete curvature estimations, and then propose a new discrete sectional-curvature estimation to be able to overcome the error. The method is based on the parabola interpolation and the geometric properties of Bezier curve. It is confirmed by experiment that the normal vector and the curvature generated by our algorithm are more accurate than that of the previous methods.


Key-Words: - Normal vector, curvature, local geometric property, mesh segmentation

## 1 Introduction

The complicated objects dealt in the field of computer graphics may be obtained by acquiring 3D position information by using 3D scanners, or generated by operating 3D modeling tools such as 3D Studio MAX or Maya. In order to efficiently deal with the objects, they may be represented by a form of mesh which includes the position information of vertices and the connectivity of them. According to the demand of users, the geometric operations such as mesh simplification and smoothing are applied to the objects[1,2,3]. In order to accomplish such geometric operations efficiently, we have to find out rightly the local properties of the mesh. In general, normal vector and curvature play the important roles as the local properties in the field of differential geometry such as curve and surface $[19,20]$. So the users want to use the same properties in the discrete space such as polygon and mesh as they were. However, there is not the exact formula of such properties because mesh is a discrete type. Therefore, we need a more exact measure of the local properties. Especially, the normal vector of vertices plays an important role in the rendering process which makes much account of reality.

## 2 Previous Works

### 2.1 Normal Vector

There are two approaches in the computation of normal vectors [4,5]. The first approach is to approximate the surface interpolating a vertex and compute the normal vector of the surface at the vertex. The result of this approach is tightly dependent on the configuration of the 1-ring neighborhood vertices and the degree of approximated surface. The second approach is to combine the normal vectors of faces adjacent to a vertex. Gouroud proposed the first algorithm of this approach that uses the average ( $N_{G}$ ) of the normal vectors of adjacent faces to the vertex in a triangular mesh [6]. This method is simple so that it is fast. However, it never uses the geometric information of the neighborhood of the vertex so that we can not apply a variety of geometric operations directly. Moreover, this method may bring the different normal vectors according to the change of topology as shown in Figure 1.


Fig. 1 The change of normal vector according to the topology

Thurmer et al. proposed a more improved method that is based on the angle-weight ( $N_{T}$ ) in order to resolve the topology-dependent problem of Gouroud’ method [7]. This method uses only the interior angle without the information such as the area of adjacent faces and the length of incident edges. It is not considered that this method prefectly reflect the geometric properties.

$$
\begin{aligned}
& N_{G}=\frac{1}{n} \sum_{1}^{n} N f_{i} \\
& N_{T}=\frac{\sum_{1}^{n} N f_{i}}{\left\|\sum_{1}^{n} N f_{i}\right\|} \\
& N_{M}=\frac{\sum_{1}^{n}\left|f_{i}\right| N f_{i}}{\left\|\sum_{1}^{n}\left|f_{i}\right| N f_{i}\right\|}
\end{aligned}
$$

Taubin [8] and Max [9] proposed area-weight normal vector ( $N_{M}$ ) algorithms in 1995 and 1999. The normal vector of a vertex by these methods is generated when we approximate a surface interpolating the vertex and represent the surface with Taylor series. The area-weight is induced when the normal vector of the surface is converted to a mesh. It is not considered that they derived a more exact normal vector because the parameter used in their Taylor's series is not geodesic so that the error of approximation is large. In order to improve the area-weighted normal vector computation, Chen [10]
proposed a new area-weighted normal vector ( $N_{C}$ ) computation the weight of which is the value of the area of an adjacent face divided by the length of the line segment between the given vertex and the center of the face. The normal vertex of a vertex $v$ is defined by

$$
N_{C}=\frac{\sum_{1}^{n} \omega_{i} N f_{i}}{\left\|\sum_{1}^{n} \omega_{i} N f_{i}\right\|}, \quad \omega_{i}=\frac{\left|f_{i}\right|}{\left|g_{i}-v\right|}
$$

where $\left|f_{i}\right|$ and $g_{i}$ are the area of face $f_{i}$ and the center of mass of $f_{i}$, respectively. This normal vector is affected by the non-adjacent vertices as well as the two adjacent vertices so that this method may cause inaccurate normal vectors. Figure 2 shows an example that two faces have the same local geometric property but the different weights.


Fig. 2: Two faces with the same local geometric property and the different weights

In this paper, we propose a new normal vector computation method which considers the interior angle and the length of adjacent edges simultaneously. Our method applies conformal mapping to the 1-ring neighborhood vertices of a given vertex, and then computes the mean value coordinates of the vertex with respect to the middle points between the mapped adjacent vertices and the Origin. Finally, the normal vector of the vertex is represented by linear combination of the edge normal vectors with the mean value coordinates. The edge normal vector is the average of the normal vectors of two adjacent faces. It is well-known that the conformal mapping reflects the local geometric properties well and the mean value coordinate is continuous over the entire domain. Our experimental results show that our method can compute the more exact normal vectors than the previous algorithms. Moreover, we may resolve the problem of phong shading that the intensity at a point is dependent on how to choose a coordinate system of world
space because the mean value coordinate is unrelated to the world space's coordinates system.

### 2.2 Curvature

The estimation of curvatures in triangular meshes plays important role in many applications such as surface segmentation, anisotropic remeshing or rendering. A lot of efforts have been devoted to this problem, but there is no consensus on the appropriate way[12-18]. Popular methods include quadratic fitting, where the estimated curvature is the one of the quadratic that best fits a certain neighborhood of a vertex locally. Most recently, Goldfeather proposed the use of a cubic approximation scheme that takes into account vertex normals in the 1 -ring[14]. The accuracy of these curvature estimations is dependent of that of fitting. If the one-ring neighborhood has many vertices or has a oscillated shape, then the approximated surface does not resemble the local shape and these estimations may yield a high error. Other methods typically consider some definition of curvature that can be extended to the polyhedral setting. These methods compute Gaussian curvature and Mean curvature based on the Gauss-Bonnet theory and Euler theory. Taubin presented a method to estimate the tensor of curvature of a surface at vertices of a mesh [17]. Watanabe proposed a simple method of estimating the principal curvatures of a discrete surface [18]. Meyer et. al proposed a discrete analog of the Laplace-Beltrami operator to estimate the discrete curvature[13].


Figure 3. Several Polygons with the same discrete curvature

Most of these methods compute directly the sectional curvatures for each adjacent edge of a vertex. They assume the normal curve interpolate both the given vertex
and an adjacent vertex and the curve is represented by Taylor series. However, they make the same mistake that they adopt the distance between the given vertex and its adjacent neighbor vertex as the parameter of the series. Figure 3 shows their drawback. There are several polygons of different interior angles, all of which are circumscribed by circles of the same radius. The discrete curvatures estimated by those methods are the same as that of the circle. It is quite alien to universal concepts.

In this paper, we present symmetric parabola-based discrete curvature estimation in order to solve such a problem in computing the sectional curvatures by the previous discrete curvature estimations. Our method is based on the parabola interpolation. We show that our method has a good geometric property so that we may derive a more simple formula and resembles the local shape better than the previous methods. Moreover, we detect a fatal error of the circle-based discrete curvature estimations.

## 3 Normal Vector Computation

In this section, we explain the mean value coordinates that is a generalization of the bary-centric coordinate and introduce our normal vector computation base on the mean value coordinates.

### 3.1 Mean Value Coordinates

We have been often asked a question "Can we represent a point inside a polygon as a linear combination of the vertices of the polygon uniquely?". If possible, we may represent all of information of a point inside a polygon with the information of the vertices. There is a convex combination approach as a general solution of the problem. Specially, the representative solution of the approach is the mean value coordinate proposed by Floater [11].


Fig. 4: Configuration of mean value coordinates
Equation (1) shows a general form of the convex combination approach that is a linear combination of the vertices $U_{i}, i=1, \cdots, m$ of the polygon for an arbitrary interior point $P$ of a polygon as shown in Figure 3.

$$
\begin{equation*}
P=\sum_{i=1}^{m} \lambda_{i} U_{i}, \sum_{i=1}^{m} \lambda_{i}=1, \lambda_{i} \geq 0 \tag{1}
\end{equation*}
$$

Floater defined the coefficients of the linear combination satisfying the above equation by the following manner:

$$
\lambda_{i}=\frac{\omega_{j}}{\sum_{k=1}^{m} \omega_{k}}, \omega_{j}=\frac{\tan \left(\frac{\alpha_{j-1}}{2}\right)+\tan \left(\frac{\alpha_{j}}{2}\right)}{\left\|U_{j}-P\right\|}
$$

Figure 5 shows an example of excellence of the mean value coordinate. The same color means the same mean value coordinate with respect to the vertex of a regular 7 -gon, which is on the y-axis. The left and the right figures are the results for a convex polygon and for a concave polygon, respectively. The upper and the lower figures are generated by the mean value coordinate and the barycentric coordinate, respectively. The mean value coordinate system generates the continuous coefficient whether or not a point is inside or outside, and whether or not the polygon is convex or concave. On the other hand, the result generated by the barycentric coordinate system is very confused and the value of coefficients is discontinuous. The continuity of the coefficient plays an important role in a variety of application such as surface interpolation.


Fig.5: Mean value coordinates (upper) vs. Barycentric coordinates (lower) for a convex polygon (a,c) and a concave polygon (b,d)

### 3.2 Normal Vector Computation

In this subsection, we introduce our algorithm for computing the normal vector of a vertex based on the mean value coordinates. We assume that a vertex $V$ has m adjacent vertices $V_{1}, V_{2}, \cdots, V_{m}$. Then the vertex $V$ is surrounded with faces $f_{i}=\Delta V_{i} V V_{i+1}, i=1, \cdots, m$. We denote the unit normal vector of a face $f_{i}=\Delta V_{i} V V_{i+1}$ by $N f_{i}$ and the unit normal vector of the common edge $e_{i}$ of faces $f_{i}$ and $f_{i+1}$ by $N e_{i}$. In order to get the relationship of $V$ with respect to the adjacent vertices $V_{i}$, we apply the conformal mapping to the 1-ring neighborhood of the vertex $V$ as shown in the Figure 6. The left figure is the configuration of the 1-ring neighborhood, and the right is the result of the conformal mapping of the neighborhood. The vertices $V$ and $V_{i}$ are mapped to the vertices $P$ and $U_{i}$, respectively, satisfying the following conditions:

$$
\left\|P U_{i}\right\|=\left\|V V_{i}\right\|, \quad \alpha_{i}=\frac{2 \pi}{\sum_{j=1}^{m} \theta_{j}}
$$



Fig. 6: Conformal mapping: (a) 3D Mesh (b) 2D polygon
We define a new normal vector based on the mean value coordinates and a local conformal mapping by the following manner:

$$
N_{C M}=\frac{\sum_{1}^{m} \lambda_{i} N e_{i}}{\left\|\sum_{1}^{m} \lambda_{i} N e_{i}\right\|}, \quad \lambda_{i}=\frac{\omega_{i}}{\sum_{k=1}^{m} \omega_{k}}
$$

$$
\omega_{k}=\frac{\tan \left(\frac{\alpha_{k-1}}{2}\right)+\tan \left(\frac{\alpha_{k}}{2}\right)}{\left\|U_{k}-P\right\|} .
$$

The normal vector ( $N_{C M}$ ) of our method may be regarded as that it reflects the local geometric property better than the previous methods because it is generated by applying the conformal mapping and the mean value coordinates. In general, the geometric property of a vertex in a 3D mesh has a tendency to resemble the property of the surface interpolating the 1-ring neighborhood and the vertex. The edges of the 1-ring neighborhood are approximated to the surface better than the faces because of the interpolating condition. So our method is very intuitive.

## 4 Curvature Computation

First of all, we review the circle-based discrete curvature estimation adopted in the previous methods of estimating the discrete curvature of meshes. This method utilizes the Taylor series of a curve and adopts the distance between two vertices as parameter of the curve. It makes the trajectory of points with the same curvature be a circle. So, in this paper we call this method as C-discrete curvature.

### 4.1 C-Discrete Curvature Formula

Most of the previous discrete curvature estimation directly compute the sectional discrete curvature whether they use the tensor of curvature or Laplacian Operator [13,14,17]. All of them compute the curvature by using the Taylor series of a curve interpolating two vertices. Let p and $\mathrm{p}_{\mathrm{i}}$ be a given vertex and its adjacent vertex, respectively and let $g(s)$ be a continuous curve passing through the two vertices: $g(0)=p, g(s)=p_{i}$. Then the curve may be represented by its Taylor series:

$$
\begin{aligned}
& g(s)=g(0)+\frac{g^{\prime}(0)}{1!} s+\frac{g^{\prime \prime}(0)}{2!} s^{2}+O\left(s^{3}\right) \\
& g(s)=g(0)+T \cdot s+\frac{N}{2} \cdot \kappa(p) \cdot s^{2}+O\left(s^{3}\right)
\end{aligned}
$$

where T and N are the unit tangent vector of the unit normal vector, respectively, and $\kappa(p)$ is the sectional curvature of $g(s)$ at point $p$ to the direction $p_{i}$. By applying the inner product to both sides with N , we may get the following equation:

$$
N \cdot(g(s)-p) \simeq s(N \cdot T)+s^{2} \frac{k(p)}{2}(N \cdot N)
$$

Because $\mathrm{N} \cdot \mathrm{T}=0$ and $\mathrm{N} \cdot \mathrm{N}=1$, the above equation may be changed to the following simple equation:

$$
\kappa(\mathrm{p}) \simeq \frac{2[\mathrm{~N} \cdot(\mathrm{~g}(\mathrm{~s})-\mathrm{p})]}{\mathrm{s}^{2}}
$$

The previous methods define the C-discrete curvature $\kappa_{\mathrm{C}}$ (p) by assuming that the parameter of the series is the distance between two vertices:

$$
\kappa_{C}(p) \simeq \frac{2[N \cdot(g(s)-p)]}{|g(s)-p|^{2}}
$$

Now, we point out the fatal error of the c-Discrete method. First of all, we find out the set of points with the same discrete curvature $\alpha$. For the convenience of computation, we assume that $\mathrm{p}=(0,0), \mathrm{p}_{\mathrm{i}}=(\mathrm{x}, \mathrm{y})$ and $\mathrm{N}=$ $(0,-1)$. Then the formula of the C-discrete curvature of $p$ is as follows:

$$
\kappa_{C}(p) \simeq \frac{2[N \cdot(g(s)-p)]}{|g(s)-p|^{2}}=\frac{-2 y}{x^{2}+y^{2}}=\alpha
$$



Figure 7. The set of points with curvature $\alpha$
Then, $\mathrm{x}^{2}+(\mathrm{y}+1 / \alpha)^{2}=1 / \alpha^{2}$ The trajectory is the circle of center $(0,-1 / \alpha)$ and radius $1 / \alpha$ as shown in Figure 7. That is, whenever the adjacent vertex $\mathrm{p}_{\mathrm{i}}$ is on the radius of the circle though it makes the different interior angle with the given vertex $p$, the sectional C-discrete curvature of $p$ to the direction $\mathrm{pp}_{\mathrm{i}}$ is the same as that of the circle. In general, the interior angle of a triangle is sharper than those of square and octagon. So, we may consider that the curvature of a vertex in triangles is greater than that of other regular polygons (see Figure 3). The result of the C-discrete curvature estimation breaks the general concepts. Moreover, there is another drawback of this method that the range of the value is restricted. If $\|p p i\|>1$,
then $\mathrm{y} /(\mathrm{x} 2+\mathrm{y} 2)<1$. Hence, $\boldsymbol{\kappa}(\mathbf{p})=\frac{\mathbf{2 y}}{\mathbf{x}^{2}+\mathbf{y}^{2}}<\mathbf{2}$. Therefore, we have known that this method cannot reflect the local shape although the adjacent vertices have a sharp interior angle because of the upper bound of curvature values(see Figure 8).


Figure 8. An counter-example of C-discrete curvature

Figure 9 shows an example that indicates the drawbacks of the C-discrete curvature estimation. The vertices of a polygon lie on the union of two circles of the same radius. By C-type estimation, there are two types of the curvature value: one is the curvature of a point adjacent to points on different circles, the other is that of a point adjacent to points on the same circle. The former is shown by green square, the other is shown by red disc. Although the points of the second type have the same curvature, the interior angles are not same. For example, the left vertex has a smaller interior angle than the other vertices, so that we may guess that its curvature is larger than the others. Therefore, it is a contrst to the result of the C-type curvature estimation.


Figure 9. The geometric meaning of
C-discrete curvature

### 4.2. Parabola-Based Discrete Curvature

In this subsection, we introduce a new discrete curvature estimation based on the parabola interpolation. The formula has a good geometric property, so the curvature estimated by our method well resembles the local shape of polygons. In general, the local shape of a polygon at a vertex is determined by the geometric relationship between the vertex and its adjacent vertices. We have
known that the concept of curvature is derived from a curve and it is a quantity to measure the local bending of curves. Therefore, the best method to resemble the local shape, following the original definition of a curvature is to use the quadratic curve interpolating the three consecutive vertices.

We adopt a quadratic Bezier curve as an interpolating curve because it has a good geometric property. Let $A, B, C$ be three consecutive vertices. The general form of a quadratic Bezier curve is as follows:

$$
\mathrm{P}(\mathrm{t})=\mathrm{P}_{0} \mathrm{~B}_{0}^{2}(\mathrm{t})+\mathrm{P}_{1} \mathrm{~B}_{1}^{2}(\mathrm{t})+\mathrm{P}_{2} \mathrm{~B}_{2}^{2}(\mathrm{t})
$$

where $P_{i}$ are the control points of the Bezier curve and

$$
\mathrm{B}_{\mathrm{i}}^{\mathrm{n}}(\mathrm{t})=\binom{\mathrm{n}}{\mathrm{i}}(1-\mathrm{t})^{\mathrm{n}-\mathrm{i}} \mathrm{t}^{\mathrm{i}}
$$

are the Bernstein polynomials of degree n . In general, there are several methods to find the Bezier curve that interpolates the given vertices. That is, the conditions for two end vertices are already determined as follows: $\mathrm{P}(0)$ $=\mathrm{A}, \mathrm{P}(1)=\mathrm{C}$, so the method is determined according to when the curve passes through the intermediate vertex $B$. This problem is a parameterization of curves. We can consider the following two methods:

```
- Standard: \(\mathrm{P}(1 / 2)=\mathrm{B}\)
- Length-based : \(\mathrm{P}(\|\mathrm{AB}\| /(||\mathrm{AB}\|+| | \mathrm{BC}\|))=\mathrm{B}\)
```

The standard parameterization is simple to derive the good geometric relations, where the length-based parameterization well resembles the local shape but yields more complex formula. In this paper, we adopt the standard parameterization so that we may find out a good relationship between the curvature of a curve and the discrete curvature of a polygon.

First of all, we compute the three control points $\mathrm{P}_{0}, \mathrm{P}_{1}, \mathrm{P}_{2}$ of the quadratic Bezier curve, satisfying the standard parameterization condition:

$$
\mathrm{P}_{0}=\mathrm{A}, \quad \mathrm{P}_{1}=(4 \mathrm{~B}-\mathrm{A}-\mathrm{C}) / 2, \quad \mathrm{P}_{2}=\mathrm{C} .
$$

Therefore, the interpolating Bezier curve is as follows:

$$
\mathrm{P}(\mathrm{t})=\mathrm{A} \mathrm{~B}_{0}^{2}(\mathrm{t})+(4 \mathrm{~B}-\mathrm{A}-\mathrm{C}) / 2 \mathrm{~B}_{1}^{2}(\mathrm{t})+\mathrm{C} \mathrm{~B}_{2}^{2}(\mathrm{t})
$$

The curvature of $\mathrm{P}(\mathrm{t})$ at $\mathrm{t}=0.5$ is

$$
\kappa_{\mathrm{P}}(0.5)=\left\|\mathrm{P}^{\prime \prime}(0.5) \times \mathrm{P}^{\prime}(0.5)\right\| /\left\|\mathrm{P}^{\prime}(0.5)\right\|^{3}
$$

$$
=\|4(\mathrm{~A}-2 \mathrm{~B}+\mathrm{C}) \times(\mathrm{C}-\mathrm{A})\| /\|\mathrm{C}-\mathrm{A}\|^{3} .
$$

Hence, we can define a new Parabola-based discrete curvature of the given vertex $B$ as follows:

$$
\kappa_{P}(\mathrm{p}) \simeq \frac{\|4(\mathrm{~A}-2 \mathrm{~B}+\mathrm{C}) \times(\mathrm{C}-\mathrm{A})\|}{\|\mathrm{C}-\mathrm{A}\|^{2}}
$$

First of all, we find out the geometric properties of the P-discrete curvature formula. Let $\mathrm{V}=(\mathrm{C}-\mathrm{A}) / 2$ and $\mathrm{G}=$ $\mathrm{A}-2 \mathrm{~B}+\mathrm{C}$. The P -discrete curvature formula is

$$
\kappa_{P}(\mathrm{p}) \simeq \frac{\|4 \mathrm{G} \times 2 \mathrm{~V}\|}{\|2 \mathrm{~V}\|^{3}}=\frac{\|\mathrm{G}\| \sin \theta}{\|\mathrm{V}\|^{2}}
$$

where $\theta$ is the in-between angle of the vectors $G$ and $V$. The numerator of the above equation is the area of the parallelogram BDEF and is four times as much as the area of the triangle BCF as shown in Figure 10. Therefore, the P-discrete curvature formula is $\kappa_{P}(B)=(2 h) / v^{2}$, where, $h$ and $v$ are the height and the width of the triangle BCF, respectively.


Figure 10. The geometric meaning of P -curvature formula

In order to verify the propriety of the P-discrete curvature, we regularly sample $n$ points on a circle of radius $r$ and compute their P-discrete curvature. Let $\mathrm{p}_{\mathrm{i}}=(\mathrm{r} \cos ((2 \pi \mathrm{i})$ $/ n), r \sin ((2 \pi i) / n)), i=0, \ldots, n-1$, be the vertices of a n -gon on the circle. By trigonometry, we can compute the values of $v$ and $h$ as follows:

$$
\begin{aligned}
& v=r \sin ((2 \pi i) / n), \\
& h=r(1-\cos ((2 \pi i) / n)) .
\end{aligned}
$$

Therefore, as the number of sampling points increases to the infinity, the value of curvature at a vertex of the n-gon becomes that of the circle.

$$
\lim _{n \rightarrow \infty} \kappa_{p}\left(p_{n}\right)=\lim _{n \rightarrow \infty} \frac{2 r\left(1-\cos \left(\frac{2 \pi}{n} i\right)\right)}{r^{2} \sin ^{2}\left(\frac{2 \pi}{n} i\right)}=\frac{1}{r}
$$



Figure 11. Polygons with the different p-curvature values

Table 1. The P-discrete curvature and its radius of curvature of the sampled points on the circle of radius 1

| n | 3 | 4 | 5 | 6 | 100 |
| :--- | :--- | :--- | :--- | :--- | :--- |
| Curvature | 4.00 | 2.00 | 1.53 | 1.33 | 1.00 |
| radius | 0.25 | 0.50 | 0.65 | 0.75 | 1.00 |

Figure 11 and Table 1 show the several polygons on a circle of radius 1 and their P-discrete curvature values. This result is an excellent contrast to that of C-discrete curvature estimation (Figure 3). The C-type estimation wishes that the curvature of the sampled vertices may become that of a circle because the vertices are sampled on the circle. The method puts emphasis on the point of view that the vertices are on a circle. However, it goes against the concept of curvatures. It loses the information on the local shape. On the other hand, our method recognizes the vertices of polygons on the circle to have a sharper angle, not to be on a circle. That is, the P-discrete curvature of vertices of a triangle is 4 and that of rectangle is 2.0 (see Figure 11). More the number of vertices increases to the infinity, less the curvature value decreases to 1 . Therefore, our method resembles the local shape of vertices more than the C-discrete curvature estimation.

### 4.3. Symmetric P- Discrete Curvature



Figure 12. Polygons with the same p-curvature values

There is an example that the P-discrete curvature estimation is not yet the perfect solution of resembling the local shape of polygons. Figure 12 contains three polygons each of which has the same width of 2 v and the same height of $h$. So, they have the same discrete curvature. However, we can expect that the third polygon has the largest curvature among them because the vertex
of the last polygon has a sharper angle than the others. In order to solve this imperfection, we propose the symmetric parabola-based discrete curvature estimation using a symmetric parabola.

### 4.1 SP-Discrete Curvature Formula

Let N be the unit vector bisecting the interior angle of B : $\mathrm{N}=(\mathrm{BA} /\|\mathrm{BA}\|+\mathrm{BC} /\|\mathrm{BC}\|) /\|\mathrm{BA} /\| \mathrm{BA}\|+\mathrm{BC} /\| \mathrm{BC}\| \|$. Then, we can consider two right-angled triangles AA'B and $C^{\prime} C^{\prime} \mathrm{B}$ as shown in Figure 13. One has the width of length $\mathrm{v}_{\mathrm{A}}$ and the height of $\mathrm{h}_{\mathrm{A}}$, the other has the width of length $\mathrm{v}_{\mathrm{C}}$ and the height of $\mathrm{h}_{\mathrm{C}}$ :


Figure 13. Polygon and its Parabola
$\mathrm{h}_{\mathrm{A}}=\mathrm{N} \cdot \mathrm{BA}, \mathrm{v}_{\mathrm{A}}=\|\mathrm{BA}-(\mathrm{N} \cdot \mathrm{BA}) \mathrm{N}\|, \quad \mathrm{h}_{\mathrm{C}}=\mathrm{N} \cdot \mathrm{BC}, \mathrm{v}_{\mathrm{C}}=$ $\|B C-(N \cdot B C) N\|$.

Then, we can consider the right-angled triangle MM'B with the width $\mathrm{v}_{\mathrm{m}}=\left(\mathrm{v}_{\mathrm{A}}+\mathrm{v}_{\mathrm{C}}\right) / 2$ and the height $\mathrm{h}_{\mathrm{m}}=\left(\mathrm{h}_{\mathrm{A}}+\right.$ $\left.\mathrm{h}_{\mathrm{C}}\right) / 2$. Therefore, we define the symmetric P-type discrete curvature of the vertex $B$ as $\kappa_{\mathrm{SP}}(B) \equiv\left(2 \mathrm{~h}_{\mathrm{m}}\right) / \mathrm{v}_{\mathrm{m}}{ }^{2}$.

## 5 Experimental Results

### 5.1 Normal Vector

Figure 14 shows a variety of polyhedrons which are approximations of a unit sphere. The upper polyhedrons are generated by recursively subdividing a regular tetrahedron, and the lower ones are generated by using the parameters of longitude and latitude. The upper cases are more irregular than the lower cases because the triangles of a different size may be appear, so that the error of the normal vector of a vertex in recursively generated polyhedrons may be larger than the others.

Table 2 and 3 show the mean error of normal vectors of a vertex in the approximated polyhedral of the unit sphere. In order to measure the exactness of the normal vectors, we used the following error energy

$$
E\left(v_{i}\right)=\left|1-<N v_{i}, N>\right|,
$$

where $N v_{i}$ is the unit normal vector of a vertex $V_{i}$ of the approximated sphere and $N$ is the normal vector of the unit sphere.


Fig. 14: Approximated Spehres
Table 2. Mean error for mesh generated by recursive method

| No. Of <br> vertices | 10 | 34 | 130 | 514 | 2050 |
| :--- | :--- | :--- | :--- | :--- | :--- |
| $N_{G}[2]$ | $0.00 \mathrm{E}-08$ | $7.25 \mathrm{E}-03$ | $1.48 \mathrm{E}-03$ | $3.41 \mathrm{E}-04$ | $7.00 \mathrm{E}-05$ |
| $N_{T}[3]$ | $0.00 \mathrm{E}-08$ | $3.57 \mathrm{E}-03$ | $6.98 \mathrm{E}-04$ | $1.35 \mathrm{E}-04$ | $2.50 \mathrm{E}-05$ |
| $N_{M}[4]$ | $0.00 \mathrm{E}-08$ | $1.42 \mathrm{E}-02$ | $2.89 \mathrm{E}-03$ | $6.47 \mathrm{E}-04$ | $1.31 \mathrm{E}-04$ |
| $N_{C}[6]$ | $0.00 \mathrm{E}-08$ | $7.42 \mathrm{E}-04$ | $3.57 \mathrm{E}-04$ | $9.30 \mathrm{E}-05$ | $2.00 \mathrm{E}-05$ |
| $N_{C M}$ | $0.00 \mathrm{E}-08$ | $5.22 \mathrm{E}-04$ | $1.41 \mathrm{E}-04$ | $3.10 \mathrm{E}-05$ | $6.00 \mathrm{E}-06$ |

Table 3. Mean error for mesh generated by parametric method

| No. of <br> Vertices | 23 | 50 | 122 | 262 | 578 |
| :--- | :--- | :--- | :--- | :--- | :--- |
| $N_{G}[2]$ | $4.03 \mathrm{E}-03$ | $7.86 \mathrm{E}-04$ | $2.22 \mathrm{E}-04$ | $5.20 \mathrm{E}-05$ | $2.40 \mathrm{E}-05$ |
| $N_{T}[3]$ | $1.41 \mathrm{E}-03$ | $5.54 \mathrm{E}-04$ | $1.11 \mathrm{E}-04$ | $6.00 \mathrm{E}-05$ | $6.00 \mathrm{E}-06$ |
| $N_{M}[4]$ | $8.87 \mathrm{E}-03$ | $1.96 \mathrm{E}-03$ | $5.88 \mathrm{E}-04$ | $6.30 \mathrm{E}-04$ | $6.30 \mathrm{E}-05$ |
| $N_{C}[6]$ | $1.77 \mathrm{E}-03$ | $1.11 \mathrm{E}-03$ | $1.11 \mathrm{E}-04$ | $5.00 \mathrm{E}-05$ | $5.00 \mathrm{E}-06$ |
| $N_{C M}$ | $7.89 \mathrm{E}-04$ | $4.63 \mathrm{E}-04$ | $5.70 \mathrm{E}-05$ | $1.60 \mathrm{E}-05$ | $2.00 \mathrm{E}-06$ |

According to the result of experiment, the normal vector computation proposed by Chen [10] is better than the other previous methods. However, the exactness of our method is two times than that of Chen. So we argue that our method is the best of all.

Figure 15 shows the results of rendering generated by Phong shading which are based on the $N_{G}$ method, $N_{C}$ method, and $N_{C M}$ method. The positions of camera and light are the same so that the highlighted region should symmetrically appear near the vertex positioned in the center of the rendering. Figure 14(a) is the wire-frame of a cube. It has irregular topology. The vertex positioned in
the center is adjacent to 4 triangles on the upper rectangle, where it is adjacent to a triangle on the left and the right rectangles, respectively. Therefore, the highlighted area should be shifted up and down. Figure 15 (b) and (c) are the results of $N_{G}$ method and $N_{C}$ method, respectively.
Figure 15 (d) generated by our method shows more realistic rendering than the other methods. The highlighted area of our method looks more realistic than the others.


Fig. 15 results of rendering

### 5.2 Comparison of SP-DC and C-DC

We compared the symmetric parabola-based discrete curvature with the circle-based discrete curvature by using the Taylor series of curves interpolating the given two points. The C-curvature adopts the distance BM between two points as the parameter of the curve, whereas the SP-curvature adopts the horizontal distance MM' between them (see Figure 13). First of all, we derive the analytic formula of the symmetric parabora-based discrete curvature by using the Taylor series. For the convenience of deriving, we assume that the bisecting unit vector is $\mathrm{N}=(0,-1)$, a given vertex is $\mathrm{p}=\left(0, \mathrm{~h}_{\mathrm{m}}\right)$, and its adjacent vertex is $p_{i}\left(v_{m}, 0\right)$. Then, the parabola $g(t)=$ $(\mathrm{t}, \mathrm{f}(\mathrm{t})$ ) interpolating two vertices is symmetric so that the unit normal vector at p is the same as N and $\mathrm{f}^{\prime}(0)=0$. Hence, the first and second derivatives of the parabola at the vertex p are $\$ \mathrm{~g}^{\prime}(0)=(1,0), \mathrm{g}^{\prime \prime}(0)=\left(0, \mathrm{f}^{\prime \prime}(0)\right)$. Therefore, we can compute the curvature of the curve at the vertex p as follows:

$$
\kappa=\frac{\left\|\mathrm{g}^{\prime \prime}(0) \times \mathrm{g}^{\prime}(0)\right\|}{\left\|\mathrm{g}^{\prime}(0)\right\|^{3}}=\left\|\mathrm{f}^{\prime \prime}(0)\right\|
$$

In order to compute the second derivative of $f(t)$, we use the Taylor series of $f(t)$. Because $f(t)$ is a quadratic function, the form is as follows: $f(t)=f(0)+f^{\prime}(0) t+f$
${ }^{\prime \prime}(0) / 2 \mathrm{t}^{2}, \mathrm{f}^{\prime \prime}(0)=(2(f(t)-f(0))) / t^{2}=\left(2 \mathrm{~h}_{\mathrm{m}}\right) / \mathrm{v}_{\mathrm{m}}{ }^{2}$. Therefore, the symmetric parabora-based discrete curvature is the same as the curvature of the curve $\mathrm{g}(\mathrm{t})$ : $\kappa_{\mathrm{g}}(0)=\kappa_{\mathrm{SP}}(\mathrm{p})$.

Table 4 The comparison of SP-curvature with C-curvature

|  | C-Discrete <br> Curvature | SP-Discrete <br> Curvature |
| :---: | :--- | :--- |
| Formula | $\kappa_{\mathrm{C}}(\mathrm{B})=(2$ <br> $\mathrm{N} \cdot \mathrm{BA}) /\\|\mathrm{AB}\\|^{2}$ | $\mathrm{K}_{\mathrm{SP}}(\mathrm{B})=(2 \mathrm{~N} \cdot \mathrm{BA})$ <br> $/ \\| \mathrm{BA}-(\mathrm{N}$ <br> $\cdot \mathrm{BA}) \mathrm{N} \\|^{2}$ |
| Parameter | Distance | Horizontal <br> Distance |
| Range | $\kappa_{\mathrm{C}}(\mathrm{B}) \leq 2$ if <br> $\\|\mathrm{AB}\\| \geq 1$ | $\kappa_{\mathrm{SP}}(\mathrm{B})<\infty$ |
| Trajectory | circle | parabola |
| Magnitude | $\kappa_{\mathrm{C}}(\mathrm{B}) \leq \kappa_{\mathrm{SP}}(\mathrm{B})$ |  |

The differences of C-discrete curvature and SP-discrete curvature are shown in Table 4. It includes the formulae of curvatures, the parameters of the curves in the Taylor series, the trajectory of points with the constant curvature, the range of curvature values, and the comparison of their magnitudes.

## 6 Conclusion

The analysis on the local properties of 3D meshes plays an important role in the applications such as morphing, simplification, smoothing. In this paper, we proposed new computation algorithms of the normal vector and curvature at a vertex of meshes of arbitrary topology. Our normal vector computation applies conformal mappings to the 1 -ring neighborhood of a vertex and computes the mean value coordinates of the vertex with respect to the vertices of the neighborhood. So, the normal vector generated by our method may well reflect the local geometric property due to the conformal mapping, and preserve the continuity of the coefficients over the mapped domain. Therefore, it could be confirmed by our experimental results that our method computes the normal vectors of vertices of a mesh more accurately than others. Moreover, this method may be applied to not only triangular meshes but also the meshes with an arbitrary topology, and has the advantage of getting the more realistic rendering. As a future research, we will develop the main algorithms of Phong shading based on the result of this research.

By exploiting the more exact computation of normal vectors, we are able to compute the right discrete curvature of vertices. The common previous methods compute directly the sectional curvatures for each one-ring neighbor, and then derive the gaussian curvature and the mean curvature using the sectional curvatures.

All of them use the circle-based discrete curvature to compute the sectional curvature. In this paper, we point out the fatal error of C-type discrete curvature computation, and proposed the parabola-based discrete curvature computation to resolve the problem. Our method may be the basis of normal vector estimation and segmentation of meshes.

## References:

[1] Dennis Maier, Jürgen Hesser, Reinhard Männer, Fast and Accurate Closest Point Search on Triangulated Surfaces and its Application to Head Motion Estimation, Wseas Transactions on Computers, Issue 4, Vol. 2, pp 874-878, 2003
[2] Jifang Li and Renfang Wang, Robust Denoising of Point-Sampled Surfaces Wseas Transactions on Computers, Issue 1, Vol. 8, pp 153-162, 2009
[3] Ravi S. Gautam and Sachin B. Patkar, Fitting convex surface to scattered data points with applications to medical imaging, Wseas Transactions on Computers, Issue 1, Vol. 3, pp 50-56, 2004
[4] Hyoungseok Kim and Hosook Kim, Computing Vertex Normals from Aribitrary Meshes, Proceeding of the 9th WSEAS Int. Conference on Signal Processing, Computational Geometry \& Artificial Vision(ISCGAV’09), pp 68-72, 2009
[5] Shuangshuang Jin, Robert R. Lewis, and David West, A Comparison of Algorithms for Vertex Normal Computation, The Visual Computer, Vol. 21, No. 1-2, pp 71-82, 2005
[6] Henri Gouraud, Continuous Shading of Curved Surfaces, IEEE Transactions of Computers, Vol. 20, No. 6, pp 623-629, 1971
[7] Grit Thurmer and Charles A. Wuthrich, Computing vertex normals from polygonal facets, Journal of Graphics Tools, 3(1), pp 43-46, 1998
[8].G. Taubin, Estimating the tensor of curvature of a surface from a polyhedral approximation, Proceeding of the Fifth International Conference on Computer Vision, pp 902-907, 1995
[9] N Max, Weights for computing vertex normals from facet normals, Journal of Graphics Tools, 4(2), pp 1-6, 1999
[10] Sheng-Gwo Chen and Jyh-Yang Wu, Estimating normal vectors and curvatures by centroid weights, Computer Aided Geometric Design 21(5), p 447-458, 2004
[11] Michael S. Floater, Mean value coordinates, Comp. Aided Geom. Design Vol. 20, p 19-27, 2003.
[12] Sylvain Petitjean, A Survey of Methods for Recovering Quadrics in Triangle Meshes, ACM Computing Surveys, Vol. 34, No. 2, pp 211-262 (2002).
[13] Mark Meyer, Mathieu Desbrun, Peter Schroder, and Alan H. Barr, Discrete Differential Geometry Operators for Triangulated 2-Manifolds, Proc of VisMath, (2002)
[14] J. Goldfeather and V. Interrante, A novel cubic-order algorithm for approximating principal direction vectors, ACM Transaction on Graphics, Vol . 23, No. 1, pp 45-63 (2004)
[15] Holger Theisel, Christian Rossl, Rhaleb Zayer, and Hans-Peter Seidel, Normal Based Estimation of the Curvature Tensor for Triangular Meshes, Proc. of Pacific Graphics'04, pp 288-297 (2004)
[16] Tatiana Surazhsky, Evgeny Magid, Octavian Soldea, Gershon Elber, and Ehud Rivlin, A Comparison of Gaussian and Mean Curvatures Estimation Methods on Triangular Meshes, Proc. of 2003 IEEE International Conference on Robotics and Automation, (2003)
[17] Gabriel Taubin, Estimating the Tensor of Curvature of a Surface from Polyhedral Approximation, Proc. of the Fifth International Conference on Computer Vision, (1995)
[18] Kouki Watanabe and Alexander G. Belyaev, Detection of Salient Curvatures Features on Polygonal Surfaces, Computer Graphics Forum, Vol. 20, No. 3, (2001)
[19] David Cohen-Steiner and Jean-Marie Morvan, Restricted Delaunay Triangulations and Normal Cycle, ACM Symposium of Computational Geometry, (2003)
[20] M. Garland and P. S. Heckbert, Surface simplification using quadratic error metrics, SIGGRAPH '97, pp 209—216 (1997)

