Curvature Estimation of 3D Point Cloud Surfaces Through the Fitting of Normal Section Curvatures

Xiaopeng Zhang*[1]/LIAMA-NLPR, Institute of Automation, CAS; Hongjun Li[2]/LIAMA-NLPR, Institute of Automation, CAS; Zhanglin Cheng[3]/LIAMA-NLPR, Institute of Automation, CAS
*[1]*xpzhang@nlpr.ia.ac.cn*, [2]*hjli@nlpr.ia.ac.cn*, [3]*zhanglin.cheng@gmail.com*

Abstract: As the technical development of laser scanning and image based modeling, more and more point cloud data are obtained to represent 3D geometric shapes of natural objects. Calculation of differential properties of 3D discrete geometry becomes one fundamental work. Through the relation of discrete normal curvatures and principal curvatures, a new algorithm is presented on estimating the principal curvatures and principal directions 3D point cloud surface. Based on the local fitting of each normal section circle properties with the position and the normal at a neighbor point, principal curvatures and principal directions are estimated from the contribution of these neighbor points. Optimization of this estimation is converted as a linear system by least squares fitting to all discrete normal curvatures corresponding to its neighbor points. A local feature curve, called as normal curvature index lines, is constructed to show the efficiency of this work. This curve is intuitive and equivalent to Dupin index line. Experiments are designed on Gaussian curvature, mean curvature and principal directions for an analytical surface and discrete surfaces of point cloud data. Experimental results show that this work is more advantageous than similar approaches, ad have applications to shape analysis and measurements.

Keywords: Normal section curvature, principal curvatures, principal directions, least squares fitting.

1. Introduction

In recent years, the increasing availability and power of range scanners has enabled us to scan larger and more complex objects, to obtain rich detail features of objects and to get larger quantity of point cloud data, which is helpful to the reconstruction of geometric objects, shape modeling and shape feature analysis. Some basic differential geometric properties should still be better estimated for object reconstruction, shape modeling or analysis. Two of these most important properties are the main curvatures together with principal directions on an estimated surface. If points are from a known analytic surface, the curvature at every point can be precisely calculated by classic differential geometric methods. However, if points were sampled from an unknown surface, with a laser-canner for example, estimating main curvatures and principal directions of every point would be an interesting and challengeable topic.

Estimating curvatures of 3D B-Rep (Boundary Representation) models has been cared since 1980s. Conventional methods often include point cloud denoising, mesh generation and curvature estimation. Some preprocessing may be performed at first, such as denoising and alignment. According to the mathematic tools adopted for geometric models, these methods can be divided into three categories. The first category is surface fitting with a polynomial surface at a local area. The surface can be a quadric surface [Sander1990; Stokely1992; Hamann1993; Krsek1998; Krsek1997], a cubic surface [Goldfeather2004], or a general polynomial surface. The second category is to compute the curvatures at each vertex of a mesh model by measuring the angle of each polygon passing through this vertex [Dyn2001; Kim2001]. The third category is to calculate the normal curvature of a direction from neighbor points, and all the normal curvatures are weightily averaged with central angle or with of tangent vectors [Taubin1995]. These methods work well for mesh models, and they can be applied to point cloud model after an extension.

We will present a new fast method for point cloud models, where both mesh reconstruction and surface fitting are all avoided. Our method just uses neighbor points and normal vectors to estimate the normal curvature, where a normal section curve is thought of as spanned by one neighbor point and corresponding normal vector there. Principal curvatures and principal directions are estimated through the least square fitting of all normal curvatures related to all neighbor points. It can be shown that this new approach is better to some degree than other similar methods, and it can be sued for shape analysis and shape measurements.

To show the effect of curvature estimation and to compare different approaches, we will construct a new kind of graph, called as normal curvature index lines, abbreviated as NCIL. This graph is simple and intuitive. With NCIL, it will be illustrated that our method is more precise and faster than the method by Taubin [Taubin1995], and it is better than the approach of Goldfeather [Goldfeather2004] also.

The organization of this paper is as flows. State-of-art work of this topic is described in Session 2. The main algorithm is reasoned and described in Session 3. Experiments and analysis on analytic data are presented in Session 4, and on discrete data are in Session 5. Conclusions and further work are described at last in Session 6.

2. Related Work

Various algorithms to estimate curvature have been proposed in literatures in recent years. They can be simply classified as two categories: direct curvature fitting and indirect curvature fitting though surface fitting. Two approaches are selected as representatives of two types approaches, Taubin's approach [Taobin1995] and Jack Goldfeather's approach [Goldfeather2004].

The curvature tensor of a surface at each point of a polyhedral approximation is estimated in Taubin's approach. Principal curvatures are obtained by computing the eigenvalues and eigenvectors of certain 3×3 symmetric matrices defined by integral formulas. This algorithm is a function of the number of points and it is linear in time and in space. This approach does not employ normal vector, so large errors may be produced. It will be improved in this paper.

Goldfeather presented three methods in his work [Goldfeather2004]: normal curvature approximation, quadratic surface approximation and adjacent-normal cubic approximation. Since the third one is the best among these three, we will choose it for an analysis and result comparison. The objective of the adjacent-normal cubic approximation method is to fit the neighbor point with a cubic polynomial surface (1) by neighbor points and corresponding normal vectors.

$$z(x,y)=\frac{a}{2}x^2+bxy+\frac{c}{2}y^2+dx^3+ex^2y+fxy^2+gy^3 \quad (1)$$

For each given point ***p*** with normal ***N***, its *m* neighbor points are ***q****ᵢ* with normal vector ***M****ᵢ*, *i*=1,2,…,*m*. In the coordinate system with ***p*** as the origin and ***N*** as z-axis, the coordinate of ***q****ᵢ* is (x_i, y_i, z_i), and $\boldsymbol{M}_i$ is (A_i, B_i, C_i). The estimation need subject to

$$\begin{cases} ax_i+by_i+3dx_i^2+2ex_iy_i+fy_i^2=-A_i/C_i \\ bx_i+cy_i+ex_i^2+2fx_iy_i+3gy_i^2=-B_i/C_i \end{cases} \quad (2)$$

This is an overdetermined equation system, and it can be written in the following matrix form

$$M\mu=R \quad (3)$$

where

$$\mu=(a \quad b \quad c \quad d \quad e \quad f \quad g)^T$$

$$M_{3m\times 7}=\begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{2}x_i^2 & x_iy_i & \frac{1}{2}y_i^2 & x_i^3 & x_i^2y_i & x_iy_i^2 & y_i^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_i & y_i & 0 & 3x_i^2 & 2x_iy_i & y_i^2 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & x_i & y_i & 0 & x_i^2 & 2x_iy_i & 3y_i^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

$$R_{3m\times 1}=(\cdots \quad z_i \quad \cdots \quad -A_i/C_i \quad \cdots \quad -B_i/C_i \quad \cdots)^T$$

Least square fitting will be applied to find the best solution of (3),, where only a, b and c are useful to the Weingarten matrix (4)

$$W=\begin{bmatrix} a & b \\ b & c \end{bmatrix} \quad (4)$$

With the eigenvalues and the eigenvectors of (4), the principal curvatures and principal direction vectors all can be calculated directly. From Goldfeather's the adjacent-normal cubic approximation method, it can be safely deduce that the normal vectors of neighbor points are important factors, but the complexity of the system (3) is as three times more complex that of the system of the input point coordinates, which will increase computer time and space for computation. Our method will improve in this complexity while maintaining the optimization.

According to a comparison of all curvature estimation methods in [Magid2007] without considering normal vectors, Taubin's approach is neither the best and nor the worst for points from an ellipsoid. But if points are from a plane or a sphere, the Taubin's approach is the best among all other methods indexed in [Magid2007]. Therefore, we just choose the Taubin approach as representative. Jack Goldfeather's method [Goldfeather2004] employs normal vectors and yields much better results, and it is probably the best in recent years.

The moving least square used for curvature estimation [Yang2007] is too complex in determining Gaussian parameter. The robust statistical estimation [Kalogerakis2007] needs more time for computation. The anisotropic filtering approach on normal field and curvature tensor field [Liu2007] focused more on pre-processing. Curvature-domain shape processing [Eigensatz2008] emphasized on post processing, especially geometry processing operations in the curvature domain.

Most papers provide comparison between different methods. In these comparisons, error analysis is very important. Traditionally, data-error or noise-error figure and table are employed, even in special paper of comparison methods [Magid2007; Mee2000; Mokhtarian2002]. We will adopt these figures or tables too. Meanwhile, we will give a new comparison figure, which is very intuitive for the effect of normal curvature computation.

3. Curvature Estimation

To estimate the curvatures at a point on a discrete surface, the distribution of neighbor points are used. Main steps of this method included using the estimation of normal section lines for normal curvature and the optimization of all these normal curvatures.

Normal direction at each sample point is regarded as the input data of the method presented here. If no normal vector information is given, we can select one method, for example the Max method [Max1999] to fit the normal vectors at first, which is not discussed in this paper.

3.1. Principle

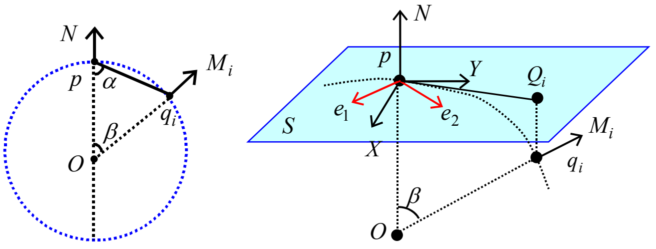
All neighbor points of a specific point on a point cloud surface determine the local shape. Big errors may be generated if curvatures are estimated through surface fitting. The contribution of normal vectors should be considered.

To estimate the curvature on a point, we will consider the contribution of one neighbor point only. This contribution is converted as the construction of a normal section curve. We will construct a normal section circle and estimate the normal curvature from the positions and normal vectors of two points, the object point and one of its neighbors.

3.2. Local Fitting for Normal Curvatures

For each point ***p*** in the point cloud, let ***N*** be the unit normal vector obtained elsewhere. We will use point coordinates and normal vectors to estimate normal curvatures at point ***p***.

Suppose there are m points in the neighbor of the point ***p*** and let ***q****ᵢ* be the *i*-th ($i=1,2,...,m$) neighbor point. The normal vector corresponding to $\boldsymbol{q}_i$ is M_i. Let {$\boldsymbol{p}$, $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{N}$} be an orthogonal coordinates system, called local coordinates L at point $\boldsymbol{p}$. $\boldsymbol{N}$ denotes the normal vector at $\boldsymbol{p}$. $\boldsymbol{X}$ and $\boldsymbol{Y}$ are orthogonal unit vectors and they are not needed to be specified given here. In L, the coordinates of $\boldsymbol{p}$, $\boldsymbol{q}_i$, and $\boldsymbol{M}_i$ can be $\boldsymbol{p}$ is (0,0,0), $\boldsymbol{q}_i$ is (x_i, y_i, z_i), $\boldsymbol{M}_i$ is ($n_{x,i,}$, $n_{y,i,}$, $n_{z,i,}$). Then we can estimate normal curvature k_n^i of point $\boldsymbol{p}$ with an osculating circle passing through point $\boldsymbol{p}$ and $\boldsymbol{q}_i$ with normal $\boldsymbol{N}$ and $\boldsymbol{M}_i$. Figure 1 shows the geometric relation of these variables.



(a) (b)

Figure 1 (a) Triangle defined by Osculating circle, neighbor point and normal vectors. (b) Local coordinates system L

The normal curvature can be estimated with the radius at point $\boldsymbol{q}_i$.

$$k_n^i = -\frac{\sin\beta}{|pq_i|\sin\alpha} \tag{5}$$

Where, α is the included angle between vectors -$\boldsymbol{N}$ and $\boldsymbol{pq}_i$, and β between vectors $\boldsymbol{N}$ and $\boldsymbol{M}_i$. An approximation of equation (5) is given by:

$$k_n^i = -\frac{n_{xy}}{\sqrt{n_{xy}^2 + n_z^2} \cdot \sqrt{x_i^2 + y_i^2}} \tag{6}$$

where

$$n_{xy} = \frac{x_i \cdot n_{x,i} + y_i \cdot n_{y,i}}{\sqrt{x_i^2 + y_i^2}}, \quad n_z = n_{z,i} \, .$$

This method employs chord, neighbor normal vector and osculating circle, so we call this method as Chord And Normal vectors (CAN) method. The advantage of this approach is that the normal vectors of neighboring points are used to estimate the main curvatures of a point.

3.3. Least Square Fitting with Euler Equation

The relation of all normal curvatures with main curvatures is analyzed here. In order to estimate principle curvatures at a point with equation (6), all coordinates of neighbor points are transformed into the coordinates of local coordinates system. Given the normal vector $\boldsymbol{N}$ at $\boldsymbol{p}$

$$\boldsymbol{N} = (n_{x,p}, n_{y,p}, n_{z,p})$$

and

$$\boldsymbol{X} = (-\sin\varphi, \cos\varphi, 0) \tag{7}$$

$$\boldsymbol{Y} = (\cos\psi\cos\varphi, \cos\psi\sin\varphi, -\sin\psi) \tag{8}$$

where, $\psi = \arccos(n_{z,p})$, $\varphi = \arctan(n_{y,p}/n_{x,p})$. The local coordinates system L at point $\boldsymbol{p}$ becomes {$\boldsymbol{p}$, $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{N}$} (Figure 1 (b)).

Let S be the plane through point p with normal vector $\boldsymbol{N}$. Let $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are principal directions at point $\boldsymbol{p}$, corresponding principal curvatures k_1 and k_2, and both are unknown. Let unknown parameter θ be the included angle between vectors $\boldsymbol{e}_1$ and $\boldsymbol{X}$, θ_i be the included angle between vectors $\boldsymbol{X}$ and the vector $\boldsymbol{pQ}_i$ obtained by projecting the vector $\boldsymbol{pq}_i$ onto the plane S. θ_i can be calculated with local coordinate of point $\boldsymbol{q}_i$.

We employ Euler formula

$$k_n^i = k_1\cos^2(\theta_i + \theta\ ) + k_2\sin^2(\theta_i + \theta\ ) \quad, i = 1,2\ldots,m \tag{9}$$

The task can be written as a optimize question:

$$\min_{k_1,k_2,\theta} \sum_{i=1}^{m}\left[k_1\cos^2(\theta_i+\theta) + k_2\sin^2(\theta_i+\theta) - k_n^i\right]^2 \tag{10}$$

Since

$$\begin{aligned} & k_1\cos^2(\theta_i + \theta\ ) + k_2\sin^2(\theta_i + \theta\ ) \\ &= \cos^2\theta_i(k_1\cos^2\theta + k_2\sin^2\theta) \\ &\quad + 2\cos\theta_i\sin\theta_i(k_2\cos\theta\sin\theta - k_1\cos\theta\sin\theta) \\ &\quad + \sin^2\theta_i(k_1\sin^2\theta + k_2\cos^2\theta) \end{aligned}$$

Expression (10) can be written as the matrix form

$$\min_{\mu} \ \|M\mu - R\|^2 \tag{11}$$

where

$$M_{m\times 3} = \begin{bmatrix} \cos^2\theta_1 & 2\cos\theta_1\sin\theta_1 & \sin^2\theta_1 \\ \vdots & \vdots & \vdots \\ \cos^2\theta_i & 2\cos\theta_i\sin\theta_i & \sin^2\theta_i \\ \vdots & \vdots & \vdots \\ \cos^2\theta_m & 2\cos\theta_m\sin\theta_m & \sin^2\theta_m \end{bmatrix}, \ R_{m\times 1} = \begin{bmatrix} k_n^1 \\ \vdots \\ k_n^i \\ \vdots \\ k_n^m \end{bmatrix}$$

$$\mu = (A, B, C)^T$$

and

$$A = k_1\cos^2\theta + k_2\sin^2\theta$$

$$B = (k_2 - k_1)\cos\theta\sin\theta$$

$$C = k_1\sin^2\theta + k_2\cos^2\theta$$

After the least square fitting of (11), the estimation value of μ can be obtained accordingly. It can be deduced that

$$\begin{aligned} \begin{bmatrix} A & B \\ B & C \end{bmatrix} &= \\ \begin{bmatrix} k_1\cos^2\theta + k_2\sin^2\theta & (k_2 - k_1)\cos\theta\sin\theta \\ (k_2 - k_1)\cos\theta\sin\theta & k_1\sin^2\theta + k_2\cos^2\theta \end{bmatrix} \\ = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \end{aligned}$$

So k_1 and k_2 are Eigen values of the matrix

$$W = \begin{bmatrix} A & B \\ B & C \end{bmatrix}$$

If we transform the unit eigenvectors of W into global coordinate system with local coordinate system L, we obtain the principal direction vectors.

Experiments will be designed in the next two sections on analytic surfaces and on discrete models to show the efficiency of the new method. Differential invariant properties such as Gaussian and mean curvatures are one of the most essential features in practice. The results about Gaussian and mean curvatures will be compared with different methods.

4. Experiments on Analytic Data

The precision of this approach is shown on analytic surface in this section, where the value of principal curvatures and principal directions are known before the estimation.

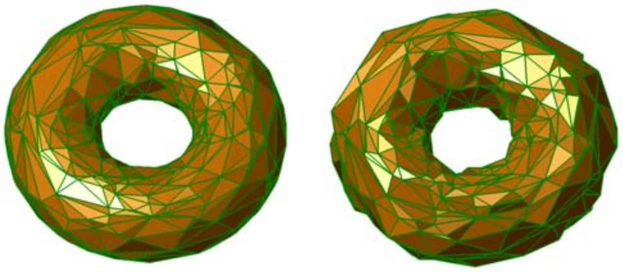
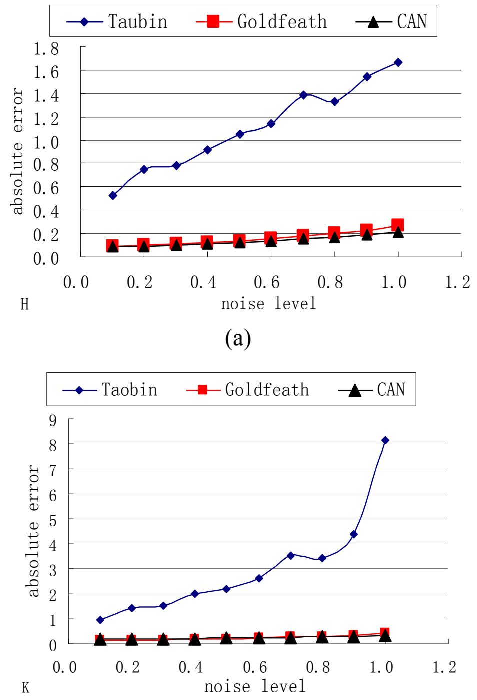

(a) (b)

Figure 2: Point cloud *torus* model with uniform noise. (a). Noise level h=0.0. (b). Noise level h=1.0.

The procedure of the experiment with CAN method is performed in two steps: calculation of normal curvature related to each neighbor point in (6) and least square fitting (in section 3.3) for principal curvatures and principal directions. Gaussian curvature and mean curvature are obtained from principal curvatures.

Experiments are performed with C language programming in a PC with the configuration of Intel(R) Core(TM) 2 CPU, 4400 @ 2G, 1.99GHz, and 2G memory.



(a)

(b)

Figure 3: Average of the absolute error produced by different methods (15 neighbor points). (a). Gaussian curvature. (b). Mean curvature.

4.1 Analytic Surface

We use a torus as the representative of analytic surfaces, since rich geometric curvature phenomena are included, like convexity and concavity. The equation of the torus is in (12).

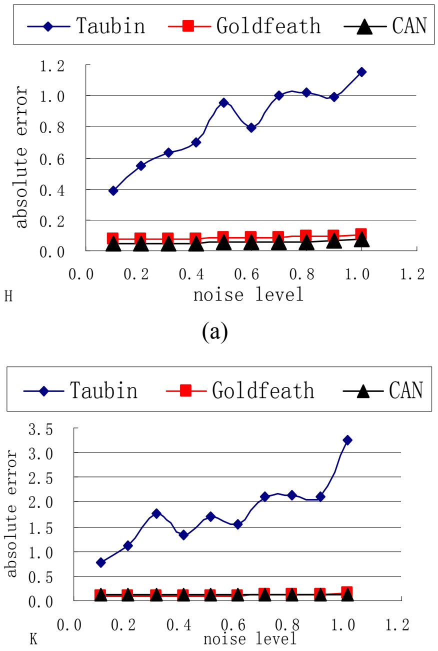
$$r(u,v) = ((R + r\cos u)\cos v, (R + r\cos u)\sin v, r\sin u)$$
$$(0 \le u \le 2\pi, \quad 0 \le v \le 2\pi) \qquad (12)$$

where parameters are specified as R=2, r=1.

5000 points are sampled from (12) according to a uniform distribution. With analytic method, normal vector and principal curvature of each point are calculated analytically. Along normal direction, noise is added to each point which subjects to uniform distribution U (-mh, mh), where parameter m is the median of distances of all pair of points. Noise level was under the control of positive real number h. The values of h are 0.1, 0.2,..., 1.0 respectively. To display clearly the shape variation of the point data surface, a mesh model is constructed and rendered filled. Topological relations of vertexes in the mesh model is never used here for curvature estimation, the. Figure 2 shows a mesh model of 1000 sample points selected from the 5000 ones with two noise levels, level h=0.0 and level h=1.0.

4.2 Comparison on Absolute Errors

In order to show a comparison of errors statistically of each approach, experiments are repeated 30 times at every noise level for each point cloud data set in section 4.1. The average error will be adopted for comparison in each case. The experiment results of Taubin method (legend as the Taubin), the Adjacent-normal cubic approximation method (legend as the Goldfeath) and our method (legend as CAN) are explained in Figure 3 and Figure 4.



(a)

(b)

Figure 4: Average of the absolute error produced by different methods (30 neighbor points). (a). Gaussian curvature. (b). Mean curvature.

Figure 3 and Figure 4 show that the absolute error of estimating Gaussian and mean curvature produced by the Taubin method is far bigger than that by the Goldfeath method and our method CAN. Both the Goldfeath and the CAN are robust to noise. Table 1 shows that the CAN tends to produce better results than the Goldfeath, especially when the noise becomes larger.

Table 1: A part of average absolute ~~average~~ error produced by the Goldfeath and CAN method

Noise	Gaussian curvature error		mean curvature error	
level	Goldfeath	CAN	Goldfeath	CAN
0.2	0.097138	0.088388	0.141680	0.192398
0.4	0.122485	0.108530	0.179423	0.206980
0.6	0.159127	0.139515	0.246997	0.238619
0.8	0.197023	0.167308	0.303347	0.272985
1.0	0.264466	0.215675	0.452247	0.343714

Absolute error levels of each point data are shown in Figure 5 and Figure 6. All points are samples from the torus (12). The redder, the bigger the error is. The blacker, the more probable that the error is zero. It is clearly that Figure 5(a) has lots of high error points. We just compare Figure 5(b) with Figure 5(c). Two blue circles in Figure 5 and 6 mean that larger absolute errors are produced by the Goldfeath at some points, but smaller error produced by the CAN.

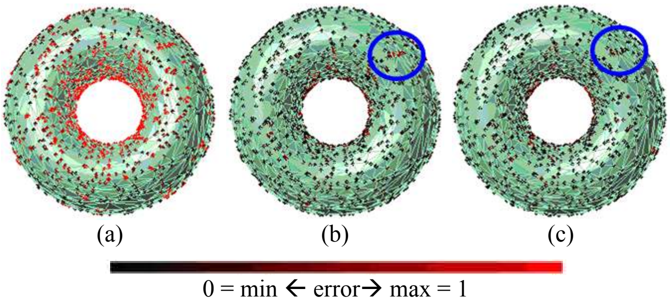


Figure 5: Absolute error of Gaussian curvature estimation in different color. (a). The Taubin. (b). The Goldfeath. (c). The CAN.

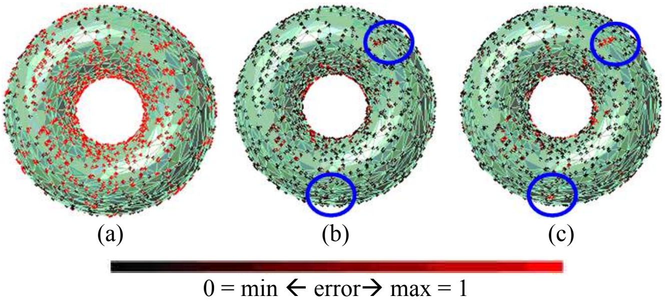


Figure 6: Absolute error of mean curvature estimation in different color. (a). The Taubin. (b). The Goldfeath. (c). The CAN.

In addition, compare with the Goldfeath method, the CAN algorithm is good in both time and memory complexity. The memory complexity can be explained from the system (43) in ~~see~~ section 2 with its coefficient matrix of 3*m* rows and 7 columns, but the system (11) (see section 3.3) with *m* rows and 3 columns. In an experiment to calculate curvatures of 10000 points while adopting 30 neighbor points as neighbors, the Goldfeath spent 5328 ms, but the CAN 2234 ms.

4.3 Normal curvature index lines

In fact, different method uses different normal curvature estimation, which means that at each point, principal curvatures (k_1, k_2) obtained by different approaches must have error. In order to have an intuitionistic view about the result, an index line is defined and constructed. Inspired by Dupin index line, normal curvature index lines (NCIL) are designed here. In Dupin index line, its polar coordinates equation is $\rho = 1/\sqrt{|k_n|}$. A polar coordinate equation of NCIL is designed as $\rho = k_n$ in Figure 7.

Along any direction in tangent plane of point ***p*** on a surface, the normal curvature can be approximately calculated accordingly. The way to draw an NCIL is to take the relation of the included angle and the normal curvature corresponding to a neighbor point. The curve of NCIL comes from sample points obtained and connected.

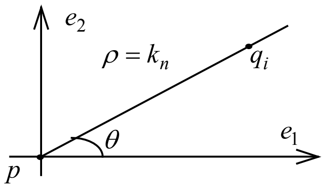


Figure 7: Normal curvature index lines.

According to the above experimental data, a point ***p*** accompanied by its 30 neighbor points is randomly selected, and 30 pairs of (θ, k_n) at point ***p*** are obtained then. Principal curvatures and principal directions are estimated respectively with three methods described above. So we will get three curves when corresponding pairs (θ, k_n) are connected sequentially as NCILs for the three methods. A NCIL will be drawn with real values of (θ, k_n) for each of three approaches compared in polar system in Figure 8 and rectangular system in Figure 9.

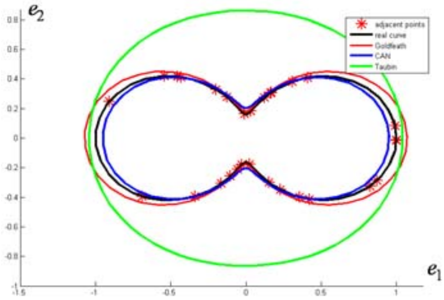


Figure 8: Normal curvature index lines of three approaches in a polar system

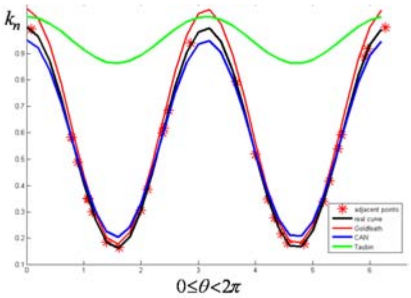


Figure 9: Normal curvature index lines of three approaches in an orthogonal system

Figure 8 shows that the NCIL (blue) obtained by the CAN is more close to the real curve (black). But the situation of the green one for the Taubin is very serous, which may cause large errors. The red curve for the Goldfeath is the moderate.

It can be seen in Figure 9 that the Taubin method generates large errors also. One reason of these large errors is that the mean of k_n is large and the amplitude is small.

Normal curvature index lines can be used to at least two things. One is an analysis of the effect of curvature fitting, which helps to evaluate curvature estimation method. The other is to compare error of several normal curvature approximations, which helps to select the best one among existing algorithms.

5. Experiments on discrete models

In order to show the accuracy and the robust of the CAN algorithm, we apply it to a discrete model *hand*, where the height of the model *hand* is 202 units. This hand model in Figure 10 (a) has 6258 points with normal vectors. This model is a mesh model and we only use the vertexes to perform experiments. The polygon information here is only adopted to visualize the principal directions estimated.

We do the experiment with different noise levels, free noise in Figure 10, noise level 0.5 units (h=0.5) in Figure 11, and noise level 2.0 units (h=2.0) in Figure 12. The effect of noise level can be seen in Figure 12(a). Noises are added according to uniform distribution $U(-h, h)$ along the normal direction of each point. The principle curvatures and directions are estimated by the CAN approach in Figure 10(b), Figure 11(b) and Figure 12(b), and by the Goldfeath algorithm Figure 10(c), Figure 11(c) and Figure 12(c) respectively. Since the Taubin algorithm produces large curvature error, it is not used here.

Figure 10 shows that the estimated principal directions are good with the Goldfeath and the CAN thanks to the point cloud hand model free of noise.

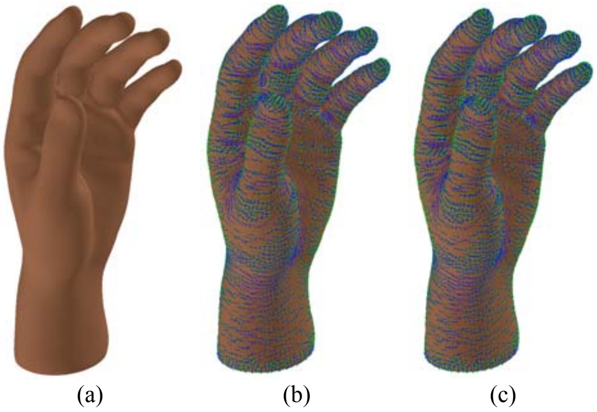


Figure 10: Point cloud *hand* model free of noise and principal directions estimated by two approaches. (a). Point cloud (display as mesh only for a view). (s). CAN. (c). Goldfeath

Figure 11 shows that with low noise level, both CAN and Goldfeath produce good results. The principal directions are organized in order in both cases. It can be seen that the principal directions on the surface of the middle part of thumb back in Figure 11 (c) are different from the ones in Figure 11 (b) and Figure 10 (c). Therefore, the Goldfeath does not work well for this level of noise.

Figure 12 shows that with large noise level, the CAN is better than the Goldfeath. It can be seen that the principal directions on the surface of the middle part of thumb back are in order in Figure 12 (b), which is close to the directions in Figure 10 (b) and Figure 10(c). Chaos of principal directions can be seen in Figure 12(c), which is quite different from Figure 10(b) and Figure 10(c). This means that the errors of principal directions estimated by the CAN is far smaller than that estimated by the Goldfeath. Therefore, the CAN is more robust than the Goldfeath.

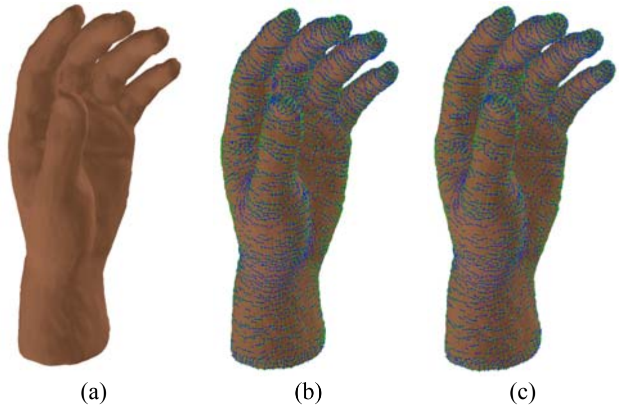


Figure 11: Point cloud *hand* model with noise level 0.5 and principal directions estimated by two approaches. (a). Point cloud (display as mesh only for watch). (b). CAN. (c).

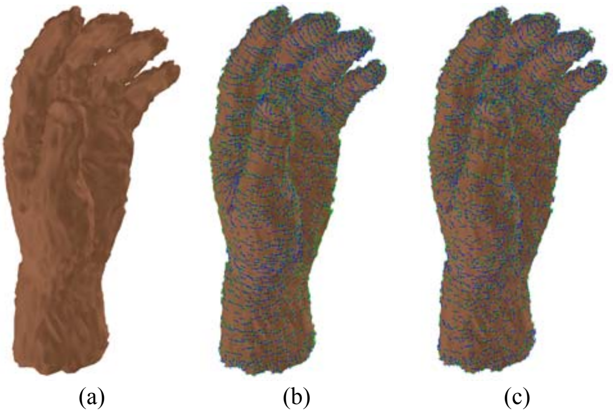


Figure 12: Point cloud *hand* model with noise level 2.0 and principal directions estimated by two approaches. (a). Point cloud (display as mesh only for watch). (b). CAN. (c). Goldfeath.

Another discrete model *elephant* was used also, where the height of this model is 75 units. This model consisted of 6859 points with normal vectors. The experimental results of different noise levels are presented in Figure 13, Figure ~~13~~ 14 and Figure 15. Mesh models in Figure 13(a), Figure 14(a), and Figure 15(a) are displayed only for data visualization. From Figure 15, it can be seen that the

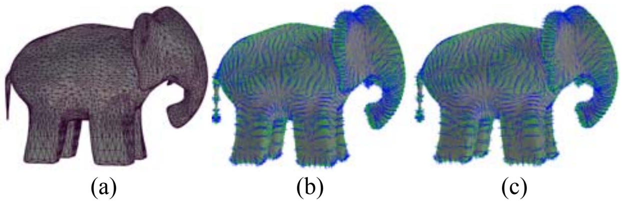


Figure 13: Point cloud model *elephant* free of noise and principal directions estimated by two approaches. (a). Point cloud surface (display as mesh only for data visualization). (b). CAN. (c). Goldfeath.

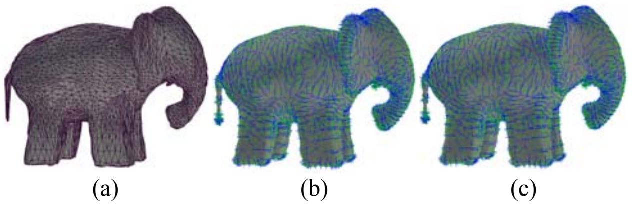

(a) (b) (c)

Figure 14: Point model *elephant* with noise level 0.50 and principal directions estimated by two approaches. (a). Point cloud surface. (b). CAN. (c). Goldfeath.

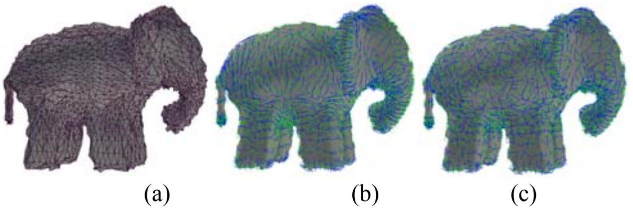

(a) (b) (c)

Figure 15: Point model *elephant* with noise level 2.0 and principal directions estimated by two approaches. (a). Point cloud surface. (b). CAN. (c). Goldfeath.

principal directions on the surface of the right side of the elephant back are in order in Figure 15 (b), which is close to the directions in Figure 13 (b) and Figure 13(c). Chaos of principal directions can be seen in Figure 15 (c), which is quite different from Figure 13 (b) and Figure 13(c). This means that in this comparison, the performance of the CAN approach are significantly better than that of the Goldfeath.

Above experiments illustrate that if the noise level is low, both CAN and Goldfeath methods produce accurately principal directions; if the noise level is high, the CAN has significant advantages in estimating principal curvature and direction.

It is interesting and challengeable to use the CAN algorithm to scanned data of complex objects. The point data of a tree is obtained with a laser scanner of a trucked tree, and the result is range image shown in Figure 16 (a). This range image points are clustered according the difference of the distance of neighboring points, so four different branch blocks are obtained then. This height of this model is 12 meter. 6950 points are included.

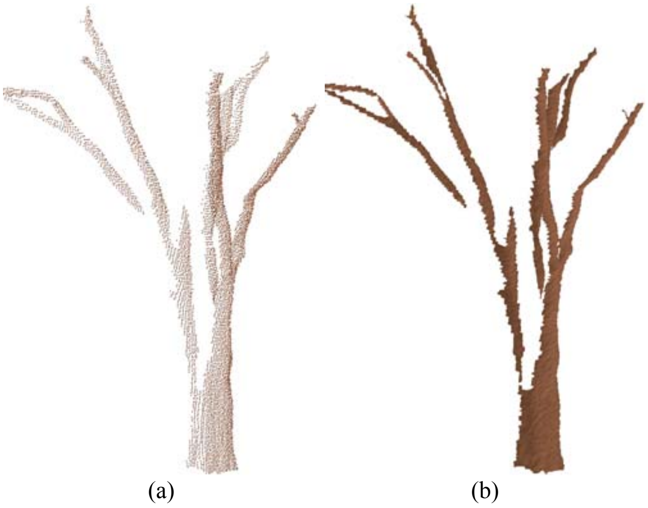
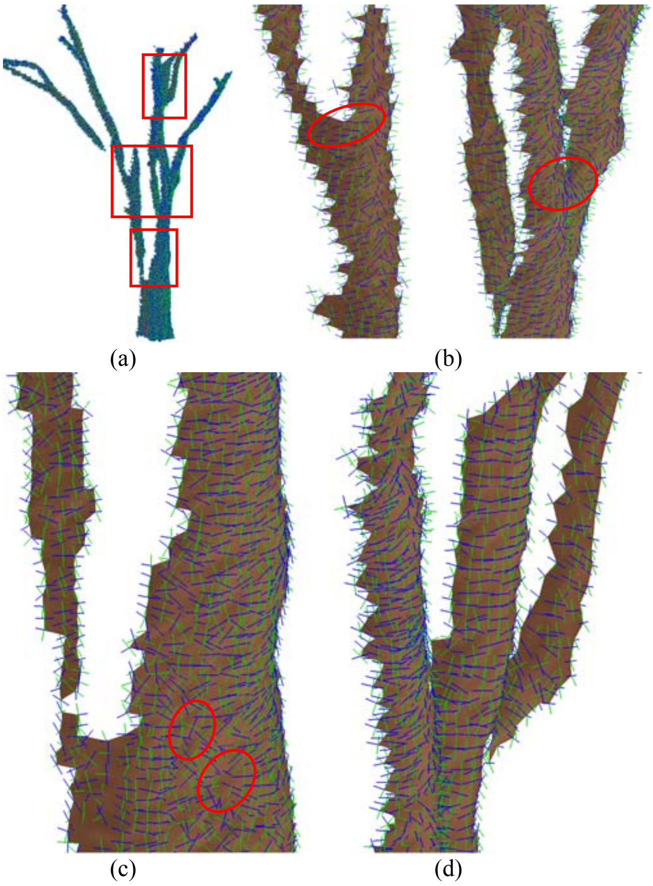

(a) (b)

Figure 16: Point cloud tree branch model. (a). Points. (b). Rendered model when points are connected to a mesh model according point neighborhood in branch positions.

Figure 17 shows the estimated principal directions with the CAN on the tree branch surface. Figure 17(a) is a global view. Figure 17(b) shows that the gradual changes of estimated principal directions at branch ramification node are reasonable. If the surface of tree branch is concave, the max principal directions will sink together, which can be seen in two areas in Figure 17(c), where the two local elliptic points are on the bark surface. From Figure 17(d), it can be seen that the blue lines as the latitude are nearly the cross-sectional directions, and the green ones as the longitude, are nearly the growth directions of branches. This example demonstrates that the CAN can be used for the shape analysis of tree branch surfaces and tree skeleton extraction [Cheng2007], which should be a bridge to tree reconstruction and measurements.


(a) (b)

(c) (d)

Figure 17: Curvature directions estimated by the CAN. (a). A global view; (b) Zoom in to the middle rectangle area of (a); (c) Zoom in to the lower rectangle area of (a); (d) Zoom in to the upper rectangle area of (a).

Briefly speaking, the CAN method can be used to estimate principal curvatures and directions with a high precision from point cloud coordinates accompanied by normal vectors. This approach is robust for point cloud data with noise.

6. Conclusions and Further Work

A new algorithm is presented for estimating the principal curvatures and principal directions on a discrete point cloud data. This algorithm, called as CAN, can be used to scattered sampled point data, and mesh vertexes data as well.

Experiments show that this new algorithm can be used to extract local differential properties with high reliability, and the effects are better than other similar approaches. The complexity of this new algorithm is lower than that of the neighbor-normal cubic approximation method [Goldfeather2004], both in time cost and in computer memory cost. The CAN algorithm is robust to data with strong noise also.

We present an index tool for error analysis, i.e. Normal curvature index lines (NCIL). It is interesting and intuitive. The errors generated by curvature estimation can be shown directly with NCIL.

In the future, we would like to apply CAN extract higher features from point cloud data, like high derivatives, and further be used to construct curvature lines, create lines and the parameterization of point cloud data. In the application to specific problems, the CAN method could be used to reconstruct real tree geometry of the data scanned with noises, which is valuable to the measurement of structural and functional information of trees.

Acknowledgement

This work is supported in part by National Natural Science Foundation of China with projects No. 60672148; in part by the National High Technology Development 863 Program of China under Grant No. 2006AA01Z301, 2008AA01Z301, and by Beijing Municipal Natural Science Foundation under Grant No. 4062033.

References

Cheng Z., Zhang X., Chen B. 2007. Simple reconstruction of tree branches from a single range image. *Journal of Computer Science and Technology*, Vol. 22, No. 6, pp. 846-858.

Dyn, N., Hormann, K., Kim, S., and Levin, D. 2001. Optimizing 3D triangulations using discrete curvature analysis. In *Mathematical Methods For Curves and Surfaces: Oslo 2000*, T. Lyche and L. L. Schumaker, Eds. 1 ed. Vanderbilt Univ. Press *Innovations In Applied Mathematics Series*. Vanderbilt University, Nashville, TN, 135-146.

Eigensatz, M. Sumner, R. and Pauly, M. 2008. Curvature-Domain Shape Processing, *EUROGRAPHICS 2008*, Volume 27 (2008), Number 2

Goldfeather, J. and Interrante, V. 2004. A novel cubic-order algorithm for approximating principal direction vectors. *ACM Trans. Graph.* 23, 1 (Jan. 2004), 45-63.

Hamann, B. 1993. Curvature approximation for triangulated surfaces. In *Geometric Modelling*, G. Farin, H. Hagen, H. Noltemeier, and W. Knödel, Eds. *Springer Computing Supplementum*, vol. 8. Springer-Verlag, London, 139-153.

Kalogerakis, E., Simari, P., Nowrouzezahrai, D., and Singh, K. 2007. Robust statistical estimation of curvature on discretized surfaces. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (Barcelona, Spain, July 04 - 06, 2007).

Kim, S.J. Kim, C.-H. and Levin, D. 2001. Surface simplification using discrete curvature norm, in: *The Third Israel-Korea Binational Conference on Geometric Modeling and Computer Graphics*, Seoul, Korea, October 2001.

Krsek, P, Luka′cs, C. and Martin, R. R. 1998. Algorithms for computing curvatures from range data. in: *The Mathematics of Surfaces VIII, Information Geometers*, in: A. Ball et al. (Eds.), 1998, 1–16

Krsek, P. Pajdla, T. and Hlava′c V. 1997. Estimation of differential parameters on triangulated surface, in: *The 21st Workshop of the Austrian Association for Pattern Recognition,* May 1997

Liu, M., Liu, Y., and Ramani, K. 2007. Anisotropic filtering on normal field and curvature tensor field using optimal estimation theory. In *Proceedings of the IEEE international Conference on Shape Modeling and Applications 2007* (June 13 - 15, 2007).

Magid, E., Soldea, O., and Rivlin, E. 2007. A comparison of Gaussian and mean curvature estimation methods on triangular meshes of range image data. *Comput. Vis. Image Underst.* 107, 3 (Sep. 2007), 139-159.

Max N. 1999. Weights for computing vertex normals from facet normals. *Journal of Graphics Tools* 4,2,1-6

Meek, D. S. and Walton, D. J. 2000. On surface normal and Gaussian curvature approximations given data sampled from a smooth surface. *Comput. Aided Geom. Des.* 17, 6 (Jul. 2000), 521-543.

Mokhtarian, F., Khalili, N., and Yuen, P. 2002. Estimation of Error in Curvature Computation on Multi-Scale Free-Form Surfaces. *Int. J. Comput. Vision* 48, 2 (Jul. 2002), 131-149.

Sander1990, P. T. and Zucker, S. W. 1990. Inferring Surface Trace and Differential Structure from 3-D Images. *IEEE Trans. Pattern Anal. Mach. Intell.* 12, 9 (Sep. 1990), 833-854.

Stokely, E. M. and Wu, S. Y. 1992. Surface Parametrization and Curvature Measurement of Arbitrary 3-D Objects: Five Practical Methods. *IEEE Trans. Pattern Anal. Mach. Intell.* Vol. 14, No. 8.

Taubin, G. 1995. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Proceedings of the Fifth international Conference on Computer Vision* (June 20 - 23, 1995). ICCV.

Watanabe, K. and Belyaev A.G.2001. Detection of salient curvature features on polygonal surfaces, in: *Eurographics 2001,* vol. 20, No. 3, 2001.

Yang, P. and Qian, X. 2007. Direct Computing of Surface Curvatures for Point-Set Surfaces, E*urographics Symposium on Point-Based Graphics (2007)* submission.