

3차원 곡률을 이용한 3차원물체의 정점추출

윤 형 태†

요 약

일반적으로 3차원 물체의 인식이나 모델링을 하기 위해서는 물체의 모양을 표현하는 방법이 필요하다. 실루엣 이미지와 같은 2차원인 경우 물체의 모양을 나타내는 경계선상의 정점 추출은 2차원 곡률함수를 이용하지만, 3차원인 경우는 물체표면의 곡률을 계산할 수 있는 3차원 곡률함수가 없기 때문에 어려운 점이 있다. 따라서 본 논문에서는 2차원 곡률원리와 최소자승법을 이용하여 근사화된 3차원 물체의 표면 곡률값과 정점을 효과적으로 구할 수 있는 새로운 방법을 제시하였다.

The Extraction of Vertex on 3-D Object using 3-D Curvature

Hyung-Tae Yoon †

ABSTRACT

In general, in order to recognize and modelling the 3-D object, it is necessary to have the method to express the shape of 3-D object. In case of 2-D like silhouette image, the extraction of vertex on the boundary of the object can be obtained by using the 2-D curvature function. But, in case of 3-D, because 3-D curvature function that can calculate the surface curvature values of 3-D object doesn't exist, it is difficult to express the shape of 3-D object. Therefore, in this paper, a new method is presented. With this presented method, the approximated surface curvature values and vertex of 3-D object can be obtained effectively using the principle of 2-D curvature and the least square method.

1. 서 론

컴퓨터 비전에서 3차원 물체의 인식분야는 중요한 분야라 할 수 있다. 물체의 인식을 하기 위해서는 물체의 크기나 각도에 관계없이 동등한 특성을 갖는 특징점 추출이 대단히 중요하며 이러한 특징점을 추출하는 일반적인 방법은 물체의 모양을 이용하는 것이다.

2차원의 경우 일반적인 방법은 원화상에서 실루엣 이미지를 추출한 후 경계선의 특징을 다항식이나 도

함수를 이용하여 근사화하거나 특징점을 추출하며, 물체의 크기나 방향에 불변의 특성을 갖는 곡률함수(curvature function)를 많이 이용하고 있다.^[4] 3차원인 경우는 일반적인 3차원 곡률함수가 없기 때문에 물체의 표면을 작은 조각들(patch)로 세그멘테이션(segmentation)한 후 조각을 2차원으로 변환하여 근사화시키는 방법^[1,2]이나 표면에 대한 법선벡터를 이용하는 방법등이 있다. 그러나 이 방법들은 표면의 세그멘테이션이나 사상(mapping)에 따라 정점추출이 달라질 수 있으므로 정확한 표면 곡률값을 얻기에는 다소 문제가 있다.

따라서 본 논문에서는 임의의 3차원 물체에서 특징

※본 논문은 인덕전문대학 학술연구비지원에 의하여 수행되었음.

† 정 회 원: 인덕전문대학 사무자동화과 조교수

논문접수: 1996년 1월 30일, 심사완료: 1996년 7월 31일

점을 얻기위하여 2차원의 곡률원리를 이용하여 3차원 표면곡률값을 구하는 새로운 방법을 제시하려한다.

2. 곡률함수

2.1 2차원 곡률함수

(그림 1)에서 곡선 $y=f(x)$ 위의 한 점 $P(x, y)$ 에서의 접선이 x 축과 이루는 각을 θ 라고 하고 점 P 의 근방의 한 점 $Q(x + \Delta x, y + \Delta y)$ 에서의 접선이 x 축과 이루는 각을 $\theta + \Delta\theta$ 라 하자. 이때 \overline{PQ} 를 Δs 라 하면 $\Delta\theta/\Delta s$ 는 \overline{PQ} 가 단위길이만큼 변할때 접선의 방향이 평균 얼마나 변하는가를 나타내므로, 곡선의 길이에 대한 곡선방향의 평균변화율이 된다. 따라서 $\Delta s \rightarrow 0$ 일때 $d\theta/ds$ 의 극한값은 점 P 에 있어서의 곡률이 된다.

만약 두 점 P, Q 를 지나면서 중심이 O 이고 반지름이 r 인 원이 존재한다면 $\Delta\theta$ 는 P, Q 에서의 두 접선의 교각과 같고 $\angle POQ$ 와 같게 된다. 따라서 원의 곡률 ρ 는 식(1)과 같이 반지름의 역수와 같게 된다.

$$\frac{\Delta\theta}{\Delta s} = \frac{\angle POQ}{PQ} = \frac{1}{r} = \rho \tag{1}$$

$\tan\theta$ 는 dy/dx 이므로 θ 는 식(2)와 같이 쓸 수 있으며 식(2)를 x 로 미분하면 식(3)과 같이 된다.

$$\theta = \tan^{-1}\left(\frac{dy}{dx}\right) \tag{2}$$

$$\frac{d\theta}{dx} = \frac{d^2y/dx^2}{1 + (dy/dx)^2} \tag{3}$$

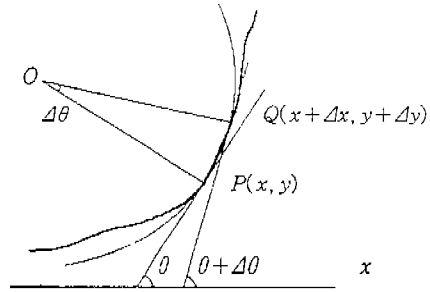
그리고 $\Delta\theta \rightarrow 0$ 인 경우 $\overline{PQ} \rightarrow \overline{PQ}$ 로 근접하므로, $ds^2 = dx^2 + dy^2$ 라고 할 수 있으며, 일반 곡선에 대해서도 이것이 성립함을 인정한다면 $(ds/dx)^2 = 1 + (dy/dx)^2$ 가 된다. 따라서 곡률은 식(4)와 같이 쓸 수 있다.

$$\frac{d\theta}{ds} = \frac{d\theta/dx}{ds/dx} = \frac{d^2y/dx^2}{\left(\sqrt{1 + (dy/dx)^2}\right)^3} \tag{4}$$

2.2 2차원 Gaussian 곡률함수

일반적으로 (그림 2)와 같은 2진화상에서 경계선함수를 식(5)와 같이 표현하면

$$s(t) = (x(t), y(t)) \tag{5}$$



(그림 1) 원상의 곡률
(Fig. 1) The curvature on circle.

경계선의 기울기는 식(6)과 같이 1차 도함수로 쓸 수 있다.

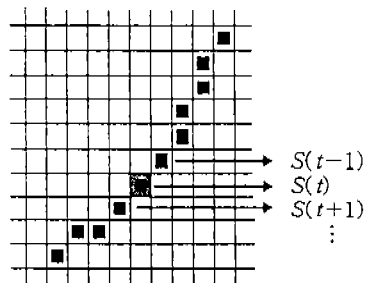
$$\dot{s}(t) = \frac{dy}{dx} \tag{6}$$

곡률함수는 경계선위의 한 점에서 곡선기울기의 순간변화율이므로 곡률 $k(t)$ 는 식(7)과 같이 정의할 수 있다^[3].

$$\dot{k}(t) = \frac{ds}{dt} = \frac{\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t)}{(\dot{x}(t)^2 + \dot{y}(t)^2)^{3/2}} \tag{7}$$

$$\begin{cases} \dot{x}(t) = x(t) * \dot{g}(t, \sigma) & \dot{y}(t) = y(t) * \dot{g}(t, \sigma) \\ \ddot{x}(t) = x(t) * \ddot{g}(t, \sigma) & \ddot{y}(t) = y(t) * \ddot{g}(t, \sigma) \\ g(t, \sigma) = \exp(-t^2/2\sigma^2) \end{cases}$$

$x(t)$ 와 $y(t)$ 의 1차도함수와 2차 도함수는 Gaussian 도함수를 $x(t)$ 와 $y(t)$ 에 각각 컨볼루션(convolution)함



(그림 2) 2차원 물체의 경계선 일부
(Fig. 2) The boundary part of 2-D object

으로써 얻을 수 있으며 식(7)에 의하여 곡률값을 구할 수 있다. 이러한 Gaussian 곡률함수는 이산신호로 발생되는 잡음을 줄일 수 있으며, σ 값의 변화로 경계선 위에 발생하는 정점의 수를 조절할 수 있다.

2.3 최소자승법을 이용한 2차원 곡률함수

Gaussian 곡률함수는 경계선의 정점을 추출하는데 가장 보편적으로 사용되지만 식(7)은 3차원으로 확장하기에는 어려운 점이 있다. 따라서 3차원 물체의 곡률값을 구하기 위해서는 3차원으로 적용하기 쉬운 다른 방법이 필요하다.

2.1절의 식(1)에서 원위의 한 점에서 곡률은 반지름의 역수와 같으므로 일반 곡선의 경우 현재점을 기준으로 주변경계점을 지나는 원의 반지름의 역수를 그 점에서의 곡률값이라 할 수 있다.

그러나 임의의 곡선인 경우 경계점들이 정확히 원을 지나간다는 보장이 없으므로 가장 근사하게 지나가는 원의 반지름을 구해야 한다. 식(8)은 일반적인 원의 방정식이다.

$$x^2 + y^2 + Ax + By + C = 0 \tag{8}$$

$$\begin{cases} A = -2a, & B = -2b \\ C = a^2 + b^2 - r^2 \end{cases}$$

식(8)을 다시쓰면 식(9)와 같이 쓸 수 있다.

$$xA + yB + C = -(x^2 + y^2) \tag{9}$$

곡선위의 점 $(x(t), y(t))$ 에 현재점과 이웃하는 주변 경계점을 $\{(x(t), y(t)) | t=1, 2, \dots, n\}$ 이라고 한다면 이 경계점을 지나는 원의 반지름은 식(9)에 의하여 식(10)과 같은 일차연립방정식으로 쓸 수 있으며 이 방정식을 풀면 원의 반지름을 구할 수 있다.

$$\begin{bmatrix} x(1) & y(1) & 1 \\ x(2) & y(2) & 1 \\ x(3) & y(3) & 1 \\ \vdots & \vdots & \vdots \\ x(n) & y(n) & 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} -\{x(1)^2 + y(1)^2\} \\ -\{x(2)^2 + y(2)^2\} \\ -\{x(3)^2 + y(3)^2\} \\ \vdots \\ -\{x(n)^2 + y(n)^2\} \end{bmatrix} \tag{10}$$

그러나 식(10)의 경우 미지수보다 좌표의 수 n 이 더 크므로 직접 해를 구할 수는 없다. 따라서 본 논문에서는 최소자승법(Least Square Method)^[5]을 이용하여 미지수와 방정식의 수를 같게 만들어 해를 구하는 방법을 사용하였다. 식(10)에서 구해진 A, B, C를 식(11)에 대입하면 경계선을 지나는 원의 반지름을 구할 수 있으며 곡률값은 얻어진 반지름의 역수가 된다. 만약 식(10)의 좌변행렬값이 0이면 이때의 곡선은 직선에 해당하므로 곡률값은 0이 된다.

$$\begin{cases} a = -A/2, & b = -B/2 \\ r = \sqrt{a^2 + b^2 - C}, & r \geq 0 \end{cases} \tag{11}$$

2.4 2차원 최소자승곡률(LSC) 알고리즘

2차원 화상에서 최소자승곡률방법(Least Square Curvature method)을 이용하여 정점을 추출하는 경우의 전체 알고리즘은 다음과 같다.

- [단계 1] 이진화상 추출: 경계선을 추출하기 위해서는 먼저 거리데이터 화상 $I(x, y)$ 에서 배경과 물체를 분리하여 2진화상 $B(x, y)$ 로 변환한다.
- [단계 2] 경계선 추출: 경계선 $s(t)$ 를 얻기 위해서 8방향 체인코드를 이용하여 경계선을 추적한다.
- [단계 3] 곡률값 계산: 모든 경계선 좌표에 식(12)와 식(13)을 이용하여 곡률값을 계산한다.

원화상	2진화상	경계선택화상	정 점
0000000000	0000000000	0000000000
0000000000	0000000000	0000000000
0000000000	0000000000	0000000000
0005555000	0001111000	0001111000	...0·0·
0005555000	0001111000	0001001000
0005555000	0001111000	0001001000
0005555000	0001111000	0001111000	...0·0·
0000000000	0000000000	0000000000
0000000000	0000000000	0000000000
0000000000	0000000000	0000000000

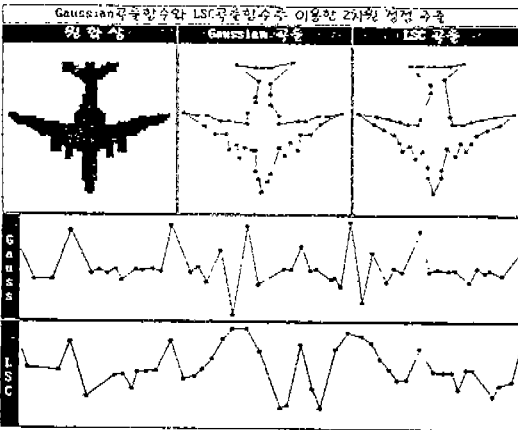
LSC 곡률값						
.0 .0 .0	.0	.0	.0	.0	.0	.0 .0 .0
.0 .0 .0	.0	.0	.0	.0	.0	.0 .0 .0
.0 .0 .0	.0	.0	.0	.0	.0	.0 .0 .0
.0 .0 .0	.7614	.6666	.6666	.7614	.0	.0 .0 .0
.0 .0 .0	.6666	.0	.6666	.6666	.0	.0 .0 .0
.0 .0 .0	.6666	.0	.6666	.6666	.0	.0 .0 .0
.0 .0 .0	.7615	.6666	.6666	.7614	.0	.0 .0 .0
.0 .0 .0	.0	.0	.0	.0	.0	.0 .0 .0
.0 .0 .0	.0	.0	.0	.0	.0	.0 .0 .0
.0 .0 .0	.0	.0	.0	.0	.0	.0 .0 .0

(그림. 3) 2차원 화상의 LSC방법 결과
(Fig. 3) The result of LSC method at 2-D image

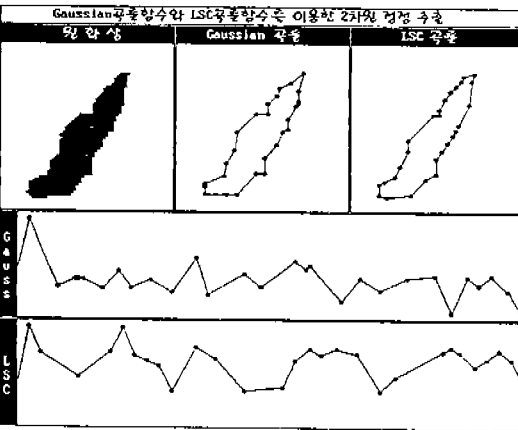
[단계 4] 정점 추출: 정점은 곡률값이 최대, 최소가 되는 점에서 발생하므로 최대, 최소 극점을 찾는다.

(그림 3)은 정사각형 화상을 LSC 방법을 이용하여 각 단계별로 처리한 결과를 보여주고 있다.

(그림 4)는 식(7)의 Gaussian 곡률방법과 식(13)의 LSC방법을 적용한 결과를 비교한 그림이다. (그림 4)는 각 방법에 대하여 추출한 정점의 위치를 경계선에 표시하였으며, 그림의 하단에서는 경계선에 대한 곡률곡선을 보여주고 있다. 두개의 곡률곡선은 거의



(a)비행기화상



(b)지도화상

(그림 4) Gaussian 곡률함수와 LSC 방법을 적용한 결과
(Fig. 4) The result of the Gaussian Curvature function and LSC method

비슷한 위치에서 정점이 발생된 것을 알 수 있다. 그림(a)는 Gaussian의 경우 $\sigma=0.2, n=5$ 이며, LSC의 경우 $n=5$ 다. 그림(b)는 Gaussian의 경우 $\sigma=0.2, n=5$ 이며, LSC의 경우 $n=3$ 이다.

3. 3차원 곡률 함수

3.1 3차원 곡률함수

경계선의 정점을 추출하는데 LSC방법을 사용하는 이유는 3차원 정점을 추출하는데 Gaussian 방법은 일반적인 3차원 곡률함수를 유도하기 어려우며, 또한 2차원 곡률함수를 3차원에 적용하기 어렵기 때문이다. LSC방법은 적용이 쉽고 간단한 알고리즘 구조를 가지고 있어 효과적인 3차원 곡률값을 구할 수 있으며 2차원 LSC방법을 그대로 3차원으로 확장하는 개념을 이용하고 있다.

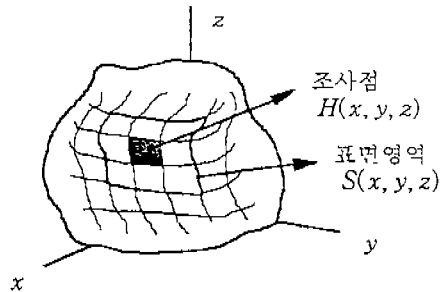
식(12)는 3차원 좌표계에서 일반적인 구의 방정식이며

$$x^2 + y^2 + z^2 + Ax + By + Cz + D = 0 \quad (12)$$

$$\begin{cases} A = -2a, & B = -2b, & C = -2c \\ D = a^2 + b^2 + c^2 - r^2 \end{cases}$$

다시 쓰면 식(13)과 같이 쓸 수 있다.

$$xA + yB + Cz + D = -(x^2 + y^2 + z^2) \quad (13)$$



(그림 5) 3차원물체의 표면점
(Fig. 5) The surface point of 3-D object

(그림 5)에서 물체표면위의 한 점 $H(x(t), y(t), z(t))$ 를 중심으로 인접하는 물체의 표면점 $S(x(t), y(t), z(t))$ 를 다음과 같이 정의하면

$$\{S(x(t), y(t), z(t)) \mid t=1, 2, 3, \dots, m\}$$

이 표면점들을 지나는 구의 반지름은 식(13)에 의하여 식(14)와 같이 1차 연립방정식으로 표현할 수 있으며 이 방정식을 풀면 A, B, C, D를 구할 수 있다.

$$\begin{bmatrix} x(1) & y(1) & z(1) & 1 \\ x(2) & y(2) & z(2) & 1 \\ x(3) & y(3) & z(3) & 1 \\ \vdots & \vdots & \vdots & 1 \\ x(m) & y(m) & z(m) & 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} -\{x(1)^2 + y(1)^2 + z(1)^2\} \\ -\{x(2)^2 + y(2)^2 + z(2)^2\} \\ -\{x(3)^2 + y(3)^2 + z(3)^2\} \\ \vdots \\ -\{x(m)^2 + y(m)^2 + z(m)^2\} \end{bmatrix} \quad (14)$$

그러나 식(14)의 경우 미지수보다 표면점의 수 m 이 더 크므로, 2차원 LSC방법과 마찬가지로 최소자승법을 이용하여 미지수와 방정식의 수를 같게 만들어 해를 구해야한다. 식(14)에서 구해진 A, B, C, D값을 식(15)에 대입하면 물체표면을 지나는 구의 반지름을 구할 수 있다. 따라서 반지름의 역수가 표면에 대한 곡률값이 된다. 만약 식(14)의 좌변행렬값이 0이면 이때의 표면은 평면에 해당하므로 곡률값은 0이 된다.

$$\begin{cases} a = -A/2, & b = -B/2, & c = -C/2 \\ r = \sqrt{a^2 + b^2 + c^2 - D}, & r \geq 0 \end{cases} \quad (15)$$

3.2 3차원 LSC 곡률함수 알고리즘

3차원 화상에서 LSC방법을 이용하여 정점을 추출하기 위해서는 먼저 물체의 모든 표면에 대한 위치정보가 필요하다. 따라서 2차원인 경우에는 달리 배경과 물체를 분리할때 배경의 밝기값은 0이고 물체에 대한 부분은 거리데이터값을 그대로 갖고 있어야 한다.

[단계 1] 물체의 표면정보구성

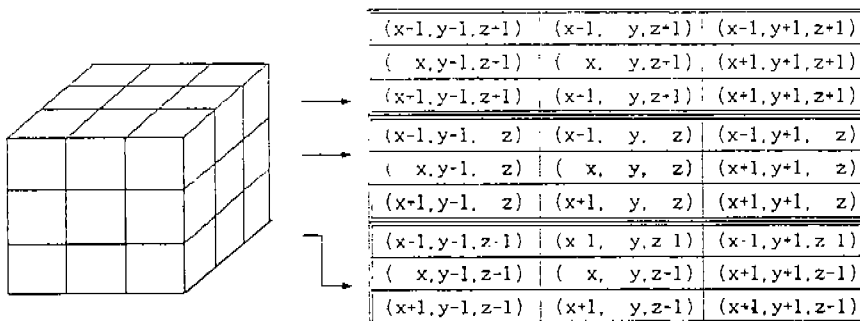
가로, 세로 크기가 N이며 밝기단계가 G인 거리데이터인 경우 표면정보를 표현할 3차원 공간은 X축을 화상의 가로, Y축을 세로, Z축을 밝기단계만큼 필요하다. 물체의 표면정보는 XY평면에서 각 좌표에 해당하는 밝기만큼 Z축을 1로 채운 후, 물체의 내부점들을 제거함으로써 얻을 수 있다.

내부점을 제거하는 방법은 (그림 6)과 같이 현재점 (x, y, z)가 1인 경우 현재점을 기준으로 X, Y, Z축으로 이웃하는 26개의 점이 모두 1인 경우는 제거후보점으로 선정한다. 이 과정을 모든 공간좌표에 대하여 처리한 후 후보점으로 선정된 점을 제거하면 표면정보에 대한 공간정보를 얻을 수 있다.

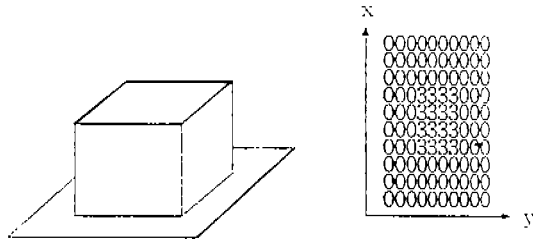
(그림 7)의 (b)는 (a)와 같은 물체의 거리데이터값이다. (c)는 (b)의 값을 이용하여 구성된 3차원 정보를 Z축을 따라 자른 XY 단면들을 나타내고 있다. (d)는 (c)에서 내부점을 제거한 것으로서 표면의 위치정보를 갖게된다.

[단계 2] 표면곡률값 계산

일단 표면에 대한 공간이미지가 구성되면 식(14)와



(그림 6) 현재점과 이웃하는 주변점
(Fig. 6) The current point and neighborhood points



(a) 3차원 이미지

(b) 거리데이터

(X, Y, 0)	(X, Y, 1)	(X, Y, 2)	(X, Y, 3)
000000000	000000000	000000000	000000000
000000000	000000000	000000000	000000000
000000000	000000000	000000000	000000000
000333300	000111100	000111100	000111100
000333300	000111100	000111100	000111100
000333300	000111100	000111100	000111100
000333300	000111100	000111100	000111100
000000000	000111100	000111100	000111100
000000000	000000000	000000000	000000000
000000000	000000000	000000000	000000000
000000000	000000000	000000000	000000000

(c) 구성된 3차원 정보

(X, Y, 0)	(X, Y, 1)	(X, Y, 2)	(X, Y, 3)
000000000	000000000	000000000	000000000
000000000	000000000	000000000	000000000
000000000	000000000	000000000	000000000
000111100	000100100	000111100	000111100
000111100	000100100	000100100	000111100
000111100	000100100	000100100	000111100
000111100	000111100	000111100	000111100
000000000	000000000	000000000	000000000
000000000	000000000	000000000	000000000
000000000	000000000	000000000	000000000

(d) 구성된 표면정보

(그림 7) 3차원 물체의 표면정보

(Fig. 7) The Surface information of 3-D object

.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	1.154	.904	.904	1.154	.0	.0	.0	.0
.0	.0	.904	.756	.756	.904	.0	.0	.0	.0
.0	.0	.904	.756	.756	.904	.0	.0	.0	.0
.0	.0	1.154	.904	.904	1.154	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0

(a) (X,Y,0) 단면

.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.904	.756	.756	.904	.0	.0	.0
.0	.0	.0	.756	.0	.0	.756	.0	.0	.0
.0	.0	.0	.756	.0	.0	.756	.0	.0	.0
.0	.0	.0	.904	.756	.756	.904	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0

(b) (X,Y,1) 단면

.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.904	.756	.756	.904	.0	.0	.0
.0	.0	.0	.756	.0	.0	.756	.0	.0	.0
.0	.0	.0	.756	.0	.0	.756	.0	.0	.0
.0	.0	.0	.904	.756	.756	.904	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0

(c) (X,Y,2) 단면

.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	1.154	.904	.904	1.154	.0	.0	.0
.0	.0	.0	.904	.756	.756	.904	.0	.0	.0
.0	.0	.0	.904	.756	.756	.904	.0	.0	.0
.0	.0	.0	1.154	.904	.904	1.154	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
.0	.0	.0	.0	.0	.0	.0	.0	.0	.0

(d) (X,Y,3) 단면

(그림 8) (그림 7)의 곡률값

(Fig. 8) The Curvature value of (Fig. 7)

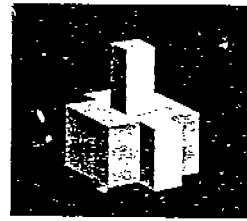
식(15)를 이용하여 모든 표면점에 대한 곡률값을 구할 수 있다. 정점의 수는 곡률값을 계산할 현재점의 주변영역크기에 따라 조절할 수 있으며 영역이 클수록 적은 정점을 추출하게 된다. (그림 8)은 (그림 7)의 곡률값을 나타내고 있다.

[단계 3] 3차원 정점 추출

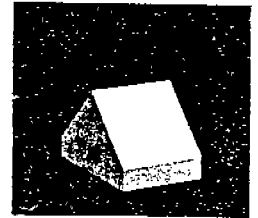
정점추출은 2차원때와 유사한 방법을 이용하지만 선분이 아닌 면이기때문에 다른 방법을 이용해야한다. (그림 8)에서 조사할 현재점이 (x, y, z)라고 한다면 X, Y, Z축으로 이웃하는 26개의 점중 물체의 표면에 해당하는 좌표만을 조사해야한다. 현재점이 정점일 조건은 조사할 표면점들의 곡률값들이 현재점의 곡률값보다 모두 크거나 작은 극점이 된다. 그러나 컴퓨터 계산수치 표현상 양자화 에러로 인하여 잘못된 정점을 찾는 경우가 발생하므로 극점을 찾을때 에러허용값을 고려해야 한다.

4. 시뮬레이션 및 결과

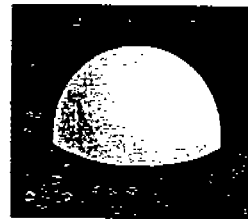
본 논문에서는 단순한 물체의 거리데이터를 이용하여 시뮬레이션하였다. 실험화상은 크기가 40×40, 밝기는 32 단계이며 m은 3으로 실험하였다. (그림 9)는 실험화상을 3차원으로 랜더링한 이미지이며 제한된 방법을 이용하여 추출된 정점을 (그림 10)에 나타내었다. (그림 10)에서 다각형 물체의 경우 추출된 정점은 물체의 표면에서 가장 급격한 변화를 갖는 꼭지점 부분에서 발생되었으며, (b)와 같이 수직으로 오목한 내부 표면에서도 내부 정점들을 추출할 수 있었다. 또한 구나 원기둥과 같이 균일한 특성의 곡면이나 곡선을 갖는 경우는 정점추출이 어려우나 원화상 자체가 양자화되어 있으므로 곡률변화가 심한 곳에서 정



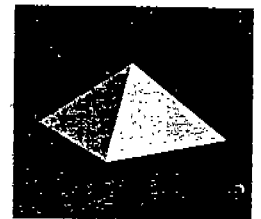
(c) 복잡한 기둥



(d) 오각 기둥



(e) 반구

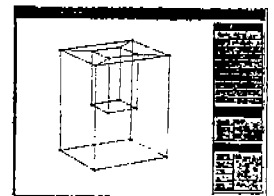


(f) 사각뿔

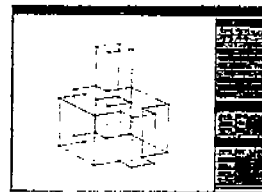
(그림 9) 랜더링한 화상
(Fig. 9) The rendering images



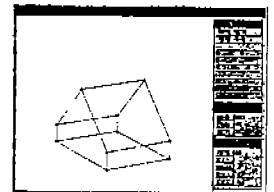
(a) 원기둥



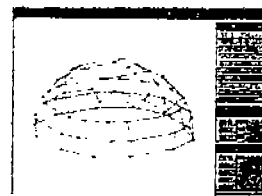
(b) 오목한 사각기둥



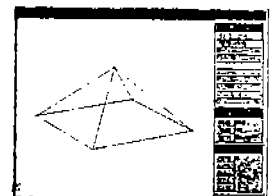
(c) 복잡한 기둥



(d) 오각 기둥

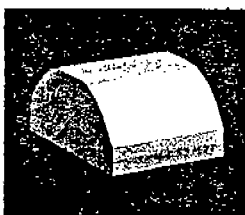


(e) 반구

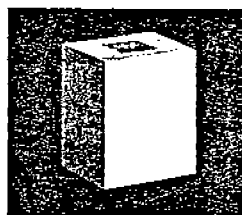


(f) 사각뿔

(그림 10) 추출된 정점
(Fig. 10) The extracted vertex



(a) 원기둥



(b) 오목한 사각기둥

점이 추출되었다. 이러한 정점들은 물체의 모양을 정의하는 특징점으로서 이 정점들로부터 물체의 모서리, 면 그리고 구조의 정보를 추출할 수 있다.

표 1은 IBM-PC 486DX 59MHz 컴퓨터에서 (그림 10)의 각 화상을 처리한 처리시간이며 처리시간은 물체의 표면 면적과 곡률계산시 조사영역크기에 따라 비례한다.

〈표 1〉 (그림 10)의 처리시간 (단위 : 초)
 (Table 1) The Processing time of Fig 10. (Second)

물체	시간	물체	시간
(a) 원기둥	21.81	(b) 오목한 사각기둥	24.06
(c) 복잡한 기둥	22.91	(d) 오각기둥	22.90
(e) 반구	20.05	(f) 사각뿔	22.79

5. 결 론

본 논문에서는 3차원의 물체의 모양을 정의할 수 있는 정점을 찾을 수 있는 새로운 알고리즘을 제시하였다. 이 방법은 기존의 2차원 곡률원리와 최소자승법을 이용하여 곡률값을 구하는 것이며 3차원으로 확장하는 경우 3차원 물체의 표면에 대한 곡률값을 구할 수 있다. 따라서 3차원 물체의 정점을 쉽게 추출할 수 있으며, 추출된 정점을 이용하여 3차원 물체인식에 효과적으로 적용할 수 있으리라 사료된다.

참 고 문 헌

[1] E.Artzy, G.Frieder, and G.T.Herman, "The theory, design, implementation, and evaluation of a three dimensional surface detection algorithm," Comp.Graphics, vol. 14, pp.2-9, 1980.
 [2] E.M.Stokely and Shang You Wu, "Surface Parameterization Measurement of Arbitrary 3-D Objects: Five Practical Methods," IEEE Trans. Patt. Anal. Machine Intell, vol.PAMI-14, pp.833-840, aug. 1992.
 [3] Erwin Kreyszig, Advanced Engineering Mathematics, John Wily & Sons, Inc., pp.373-386, 1979.
 [4] D.M.Wuescher and K.L.Boyer, "Robust Contour

Decomposition using a constant Curvature Criterion," IEEE Trans. Pattern Anal. Machine Intell., vol.PAMI-13, pp.41-51, jan. 1991.

[5] Sidney Yakowitz and Ferenc Szidarovszky, An Introduction to Numerical Computation, Maxwell Macmillan Publishing inc. pp.298-322, 1990.



윤형태

1984년 성균관대학교 전자공학과 졸업(학사)
 1986년 성균관대학교 대학원 전자공학과 졸업(공학석사)
 1993년 성균관대학교 대학원 전자공학과 졸업(공학박사)

1991년~현재 인덕전문대학 사무자동화과 조교수
 관심분야: 2차원신호처리, 화상처리, 컴퓨터그래픽