Superquadric과 CSG에 기반한 3차원 모델링

김대현*, 이선호*, 김태은**, 최종수*
*중앙대학교 첨단영상전문대학원 영상공학과 영상정보연구실
**남서울대학교 멀티미디어학과

3D object Modeling based on Superquadrics and Constructive Solid Geometry

Dae-Hyun Kim*, Sun-Ho Lee*, Tae-Eun Kim**, Jong-Soo Choi**
Dept. of Image Eng, Chung-Ang Univ.
Dept. of Multimedia., Nam-Seoul Univ.

요 약

3차원 물체 형상 모델링은 인식에 있어서 중요한 역할을 차지하고 있다. 기존의 픽셀 (pixel)기반 영상 표현은 물체 고유의 유기적 구조를 반영 할 수 없고, 에지 (edge)나 면 기반 물체 표현법은 물체의 자세한 표현이 가능하지만 물체인식을 위해서는 많은 양의 속성들을 만들어내게 된다. 따라서 물체인식을 위해서는 물체의 형상특징을 직선적으로 기술할 수 있는 체적소 기반 물체 표현 방법이 필요하다. 본 논문에서는 몇 개의 파라미터를 이용하여 3차원 정보를 효과적으로 얻을 수 있는 superquadric과 이를 기본 단위로 한 CSG (Constructive Solid Geometry) tree를 이용하여 3차원 물체 형상 모델링에 대해서 기술한다.

I. 서 론

3차원 시각기술은 2차원 시각 기술이 물체의 밝기에 기초한 시각 정보를 사용하는데 반해 주로 센서로부터 물체까지의 거리에 기초한 거리 정보를 사용하여 물체와 장면을 분석함으로써 밝기 정보가 가지는 많은 문제점을 해결할 수 있다. 또한 거리정보를 이용하여 보다 구체적인 물체 표면의 기하학적인 성질 등을 계산할 수 있어, 3차원 물체의 정확한 표현 및 인식이 가능하다[1].

본 논문에서는 실험실 환경에서 다양한 형태의 3차원 물체 형상을 효과적으로 모델링할 수 있는 방법에 대해서 기술한다.

Superquadric은 다양한 형태의 복잡한 3차원 물체를 표현하는데 적합한 모델이다. 뿐만 아니라 몇 개의 파라미터만으로도 3차원 정보를 효과적으로 얻을 수 있어 3차원 물체 형상 모델링에 적합하다[1][2][5].

본 논문에서는 3차원 물체를 superquadric과 변형된 superquadric을 기본 단위로 한 CSG tree에 의해 표현하였다. CSG tree 구성에 필요한 물체간의 집합 연산은 superquadric과 변형된 superquadric의 implicit함수의 역변형 함수를 사용하여 수학적으로 정의된다[2][3][4].

본 논문의 구성은 II장에서는 superquadric의 정의와 물체 표현을 다룬다. III장에서는 superquadric간의 집합 연산을 정의하고, IV장에서는 3차원 물체 형상 모델링을 위한 표면 shading을 설명한다. V장에서는 모의 실험 결과를 보였으며, 마지막으로 VI장에서는 결론을 맺는다.

II. Superquadric을 이용한 3차원 물체의 표현

Superquadric은 다양한 모델을 표현할 수 있고, 입체의 표현 수식이 비교적 간단하여 입체간의 집합연산에 필요한 함수가 수학적으로 잘 정의된다. 따라서 부품(part)을 기초로 한 3차원 물체의 표현에 유용하게 사용될 수 있다[1]. 또한 변형된 superquadric을 함께 이용하면 보다 실제적인 3차원 물체 형상 모델링이 가능하다.

2.1 Superquadric의 정의

Superquadric 표면은 다음의 3차원 벡터로 정의된다.

$$X(\eta,\omega) = \begin{bmatrix} a_1\cos^{\varepsilon_1}(\eta)\ \cos^{\varepsilon_2}(\omega) \\ a_2\cos^{\varepsilon_1}(\eta)\ \sin^{\varepsilon_2}(\omega) \\ a_3\sin^{\varepsilon_1}(\eta) \end{bmatrix} \quad (1)$$

$$-\frac{\pi}{2}\le\eta\le\frac{\pi}{2}\ ,\ -\pi\le\omega\le\pi$$

a_1, a_2, a_3 : x, y, z 축 방향의 크기
ε_1 : 남북 방향의 형태
ε_2 : 동서 방향의 형태
η : 위도 , ω : 경도

벡터 $X(\eta,\omega)$는 물체중심 좌표를 가지며, 독립변수인 η, ω의 크기가 주어진 간격으로 변화할 때 공간에 폐면(closed surface)을 형성한다.

2.2 Superquadric Inside-Outside Function(Implicit Function)

식(1)은 superquadric 표면의 파라미터 방정식이다.

$\cos^2(\alpha)+\sin^2(\alpha)=1$을 이용하여 η, ω를 제거하면 다음의 implicit 함수를 구할 수 있다.

$$F(x,y,z)=\left[\left[\left(\frac{x}{a_1}\right)^{2/\varepsilon_2}+\left(\frac{y}{a_2}\right)^{2/\varepsilon_2}\right]^{\frac{\varepsilon_2}{\varepsilon_1}}+\left(\frac{z}{a_3}\right)^{2/\varepsilon_1}\right]^{\varepsilon_1} \quad (2)$$

2.3 Superquadric의 변형

2.3.1 점감(Tapering)

점감은 식(3)-(a)의 회전 중심축(z축) 함수에서 점감 정도(기울기)를 계산하고, 이 계산된 결과를 이용해 중심축과 직각인 두 방향(x, y축)의 좌표를 식(3)-(b), (c)와 같이 증가 또는 감소시킨다. 그 결과는 **그림1** (b)와 같다.

$$\begin{array}{ll}(a) & r=f(z,t)=1.0+(a_3+z)t/(2a_3)\\ (b) & X=rx\\ (c) & Y=ry\end{array} \quad (3)$$

2.3.2 꼬임(Twisting)

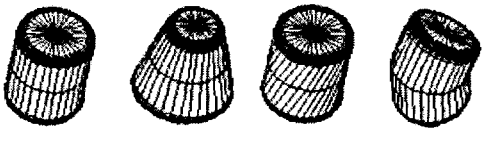
꼬임은 식(4)-(a)의 회전 중심축과 꼬임 인자 t_w의 함수로부터 계산된 꼬임각도를 이용해, 중심축과 직각인 두 방향(x, y축)의 좌표을 회전시키는 것이다. 그 결과는 **그림1** (c)와 같다.

$$\begin{array}{ll}(a) & \theta=f(z,t_w)=t_w(z+a_3)\\ (b) & X=x\cos\theta-y\sin\theta\\ (c) & Y=x\sin\theta+y\cos\theta\\ (d) & Z=z\end{array} \quad (4)$$

2.3.3 굽힘(Bending)

k를 굽힘 곡률이라 할 때, 굽힘각은 굽힘 곡률과 중심축의 함수 $\theta=z\times k$에 의해 계산되며, 계산된 결과는 식(5)를 이용해서 각 점을 회전시킨다. 그 결과는 **그림1** (d)와 같다.

$$\begin{array}{l}r=\cos(\alpha-\beta)\sqrt{x^2+y^2}\\ R=k^{-1}-\cos(\theta)(k^{-1}+r)\\ X=x+\cos(\alpha)(R-r)\\ Y=y+\sin(\alpha)(R-r)\\ Z=\sin(\theta)(k^{-1}-r)\end{array} \quad (5)$$



(a) 실린더형의 물체 (b) 점감된 물체 (c) 꼬인물체 (d) 굽힌물체

그림 1. 변형된 superquadric

2.3.4 복합변형

변형은 점감, 꼬임, 굽힘의 순서에 의해서 반복적으로 적용할 수 있다. 점감된 물체는 점감된 물체 표면상의 점에 꼬임의 수식을 적용하여 꼬임의 변형을 가할 수 있다. 같은 방법으로 변형 후의 물체 표면상의 점에 굽힘의 수식을 적용하여 반복적인 변형이 가능하다.

$$X'=TR\cdot BE\cdot TW\cdot TA\cdot X=D(x)\cdot X$$

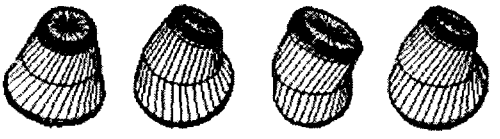
$$=\begin{bmatrix}1&0&0&\Delta X\\0&1&0&\Delta Y\\0&0&1&\Delta Z\\0&0&0&1\end{bmatrix}\begin{bmatrix}1&0&0&\cos(\alpha)\cdot(R-r)\\0&1&0&\sin(\alpha)\cdot(R-r)\\0&0&\sin(\gamma)\cdot(1/k-r)&0\\0&0&0&1\end{bmatrix} \quad (6)$$

$$\begin{bmatrix}\cos(rw)&-\sin(rw)&0&0\\\sin(rw)&\cos(rw)&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}rx&0&0&0\\0&ry&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

TA: tapering, **TW**: twisting, **BE**: bending,
TR: translation, **X**: given point(vector)

복합 변형된 superquadric 표면상의 점은 식(7)과 같이 원래 superquadric의 표면상의 점으로 복구될 수 있다.

$$X=(D(x))^{-1}\cdot X'=(TR\cdot BE\cdot TW\cdot TA)^{-1}\cdot X \quad (7)$$



(a) 점감후 꼬인 물체 (b) 점감후 굽힌 물체 (c) 꼬인후 굽힌 물체 (d) 점감, 꼬인 후 굽힌 물체

그림 2. 복합 변형된 superquadric

III. Superquadric간의 집합 연산

기존의 superquadric을 기초로 한 3차원 물체 표현 방법에서는 superquadric과 변형된 superquadric의 접합에 의해서 물체를 표현하였다. 이러한 방법은 단순히 3차원 공간상의 이동에 의해 정의되는 접합에 의해서만 가능하여 물체 표현이 비효율적인 단점을 갖는다. 그러나 superquadric과 변형된 superquadric간의 집합연산을 정의한 CSG tree 방식을 이용하면 3차원 물체를 보다 효율적으로 표현할 수 있다.

Superquadric은 implicit함수에 의해 3차원 공간을 입체의 표면, 외부, 내부로 나누므로 3차원 물체를 입체의 표면에 의해 표현할 수 있다. 따라서 두 superquadric SQ1과 SQ2의 implicit함수를 F1, F2라 하면, 두 입체간의 연산인 합집합, 차집합, 교집합은 다음과 같이 정의된다.

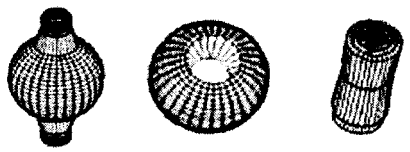
$$\begin{array}{l}SQ1\cup SQ2=\{(x,y,z)|(F1(x,y,z)=1 \text{ and } F2(x,y,z)>1)\\ \quad or\ (F1(x,y,z)>1 \text{ and } F2(x,y,z)=1)\}\end{array} \quad (8)$$

입체간의 합집합에서는 물체와 물체가 만나서 생기는 교집합에 해당하는 표면의 점들이 제거된다.

$$SQ1-SQ2=\{(x,y,z)|(F1(x,y,z)=1 \text{ and } F2(x,y,z)>1) \text{ or } (F1(x,y,z)\le 1 \text{ and } F2(x,y,z)=1)\} \quad (9)$$

입체간의 차집합에서는 SQ1의 안에 있는 SQ2의 표면의 점이 추가된다.

$$SQ1\cap SQ2=\{(x,y,z)|(F1(x,y,z)=1 \text{ and } F2(x,y,z)\le 1) \text{ or } (F1(x,y,z)\le 1 \text{ and } F2(x,y,z)=1)\} \quad (10)$$



(a) 두 물체의 합집합 (b) 두 물체의 차집합 (c) 두 물체의 교집합

그림 3. 변형되지 않은 두 superquadric의 집합 연산 결과

식(2)에 주어진 implicit함수는 변형되기 전 영상의 점들에 대해서 만족한다. 따라서 영상간의 CSG 집합 연산을 하기 전에 변형된 영상을 식(7)을 이용하여 원래 영상의 점으로 변환한 뒤 implicit함수를 이용한다.

Ⅳ. 3차원 정보를 이용한 표면 shading

4.1 z값의 gray-level 변환

Superquadric을 이용하여 얻은 3차원 깊이값(z)은 물체를 중심으로 대칭인 관계에 있다. 따라서 깊이 정보를 물체의 표면에 나타내려면 이 값을 256 gray-level로 변환해야 한다. 아래에 변환식을 보인다.

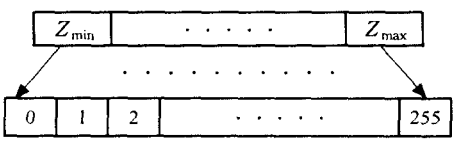


그림 4. 256 gray-level 분포도

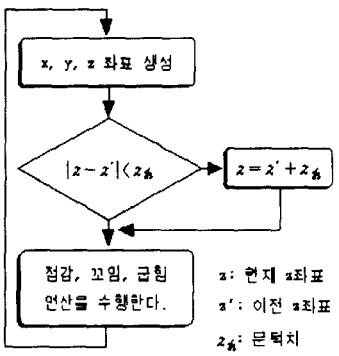
$$g=\frac{(Z_{value}-Z_{min})\times 256}{Z_{max}-Z_{min}} \quad (11)$$

4.2 축 보간

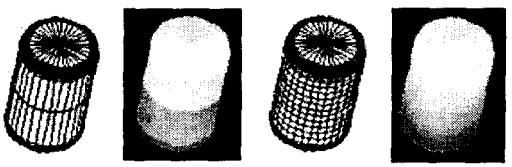
Superquadric의 모양을 결정짓는 ε_1, ε_2의 크기에 따라 만들어지는 점들간의 간격이 달라진다. 이것은 CSG를 이용한 3차원 물체 형상 모델링에서 집합 연산에 쓰일 점들의 서로 다른 분포 특성으로 인해 많은 문제점을 나타내므로 x, y, z 각 축에 대해 보간 할 필요성이 있다.

4.2.1 z축 보간

$\varepsilon_1=1.0$인 경우 z축을 따라 발생하는 점들간의 간격은 일정하지만, $\varepsilon_1\ll 1.0$인 경우 발생하는 점들은 z축 양끝으로 모여드는 양극화 현상을 보인다. 따라서 두 점 사이의 간격이 문턱치 이상이면 두 점 사이를 일정한 간격으로 보간한다.



(a) z축 보간 흐름도

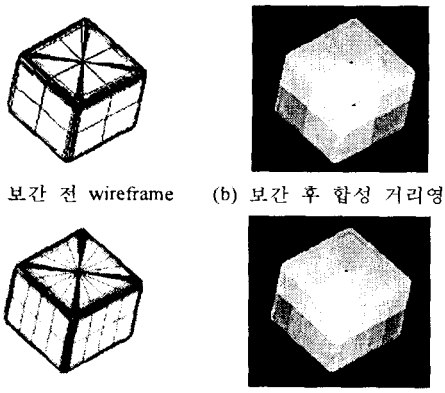


(b) 보간 전 wireframe (c) 보간 전 합성 거리영상 (d) 보간 후 wireframe (e) 보간 후 합성 거리영상

그림 5. z축 보간 흐름도(a)와 보간 전·후의 wireframe(b), (d)과 거리 영상(c), (e)

4.2.2 x, y축 보간

z축 보간과 마찬가지로 $\varepsilon_2=1.0$인 경우 x, y 축을 따라 발생하는 점들의 간격은 일정하지만, $\varepsilon_2\ll 1.0$인 경우 발생하는 점들은 서로 다른 분포 특성을 보인다. 따라서 두 점 사이의 간격이 문턱치 이상이면 두 점의 중점에 강제로 하나의 점을 추가시켜서 보간한다.

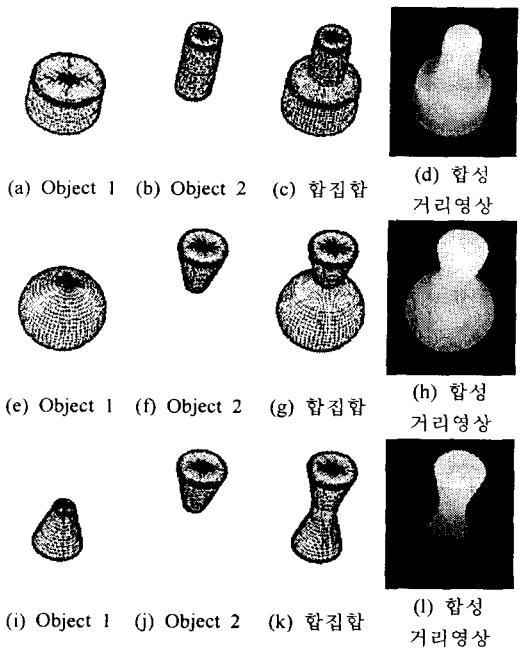


(a) 보간 전 wireframe (b) 보간 후 합성 거리영상

(c) 보간 후 wireframe (d) 보간 후 합성 거리영상

그림 6. x,y축 보간 전·후의 wireframe과 거리영상

V. 실험 및 고찰



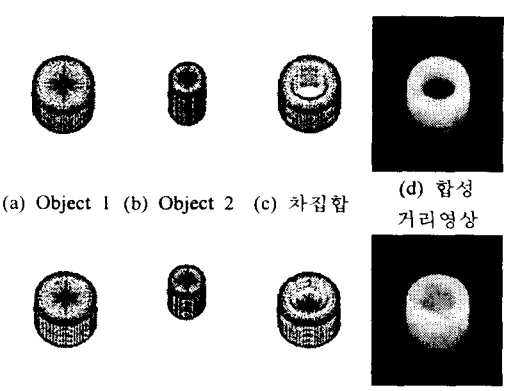
(a) Object 1 (b) Object 2 (c) 합집합 (d) 합성 거리영상

(e) Object 1 (f) Object 2 (g) 합집합 (h) 합성 거리영상

(i) Object 1 (j) Object 2 (k) 합집합 (l) 합성 거리영상

그림 7. 변형된 superquadric간의 합집합과 거리 영상

그림7 (a), (b)는 서로 다른 scale을 갖고 이동된 두 물체간의 합집합을 나타내고, (d)와(e)는 이동된 구와 점감된 두 물체간의 합집합을 보여준다. (h)와(i)는 점감된 두 물체를 이동시킨 뒤 합집합 연산을 이용하여 표현하였다. (d), (g), (k)에서 보여주는 합성 거리영상은 x, y, z의 각 축에 대해서 보간 하였으며, 마지막 표현단계에서는 3D Bilinear Interpolation을 이용하였다.



(a) Object 1 (b) Object 2 (c) 차집합 (d) 합성 거리영상

(e) Object 1 (f) Object 2 (g) 차집합 (h) 합성 거리영상

그림 8. 변형된 superquadric간의 차집합과 거리 영상

마찬가지의 방법으로 차집합에 대한 결과를 그림8 에서 보여준다. 그림8 (d)는 양쪽방향으로 구멍이 뚫린 실린더 형의 물체를 나타내고, (h)는 한쪽방향으로만 구멍이 뚫린 실린더 형의 물체를 나타낸다.

VI. 결 론

본 논문에서는 superquadric으로 얻어진 3차원 정보와 CSG tree 방법에 기반한 집합연산을 이용하여 3차원 물체 형상 모델링에 대해 기술하였다. 모의 실험결과에서 보여주듯이, 다양한 형태의 3차원 물체를 superquadric을 이용해서 쉽게 구현할 수 있었으며, 보다 복잡한 물체의 표현에서는 변형된 superquadric들간의 집합연산을 이용하여 쉽게 표현할 수 있었다.

본 논문에서 제안하고 있는 3차원 물체 형상 모델링은 superquadric과 집합연산에서 얻은 3차원 정보를 물체의 표면에 shading 함으로써 합성 거리영상 (Synthetic range image)을 취득할 수 있다. 이것을 이용하면 실험실 환경에서 고가의 하드웨어 없이 원하는 형태의 3차원 거리 데이터를 쉽게 얻을 수 있을 것이다.

참 고 문 헌

[1] 황삼청, 양현승, "Superquadric을 이용한 3차원 물체의 표현과 인식", 한국정보과학회 논문지 '90, Vol.17, No.5 September.

[2] S. C. Hwang, H. S. Yang, "3D Object Representation Using the CSG tree and Superquadrics", KITE Journal of Electronics Engineering, Vol.2, no.1 June, 1991.

[3] Nigel Stewart, Geoff Leach, "An Improved Z-Buffer CSG Rendering Algorithm", 1998 Eurographics/Siggraph Workshop on Graphics Hardware, pp. 25-30

[4] N. Stewart, G. Leach, S. John, "A Single Z-Buffer CSG Rendering Algorithm for Convex Objects", Siggraph '99 Submission.

[5] Frank Solina, Ruzena Bajcsy, "Recovery of Parametric Models from Range Images: The Case for Superquadrics with Global Deformations", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.12, No.2, Feb, 1990.

[6] C. K. Tang, G. Medioni, "Inference of Integrated Surface, Curve, and Junction Descriptions From Sparse 3D Data", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.20, No.11, November, 1998.