

# Superquadric과 Z-버퍼 CSG 렌더링 기반의 3차원 형상 모델링

김태은\*

## 요약

본 논문에서는 superquadric과 Z-버퍼 (Z-buffer) CSG (Constructive Solid Geometry) 렌더링 알고리즘을 이용한 3차원 물체 형상 모델링의 방법을 제안하였다. Superquadric은 몇 개의 계수만으로 다양한 형태의 3차원 모델을 얻을 수 있다. 그리고 CSG 트리 (tree)를 이루는 각각의 기본소 (primitive)는 superquadric과 변형된 superquadric을 가지고 표현하였다. CSG 트리를 구성하는 기본소들간의 집합 연산은 Z-버퍼 알고리즘과 스텐실 버퍼 (stencil buffer)를 사용하여 효과적으로 정의하였다. 본 논문에서 제안한 알고리즘은 단순히 물체의 깊이정보를 비교하여 표현하기 때문에 기존의 implicit 함수를 이용한 물체 표현법에서 각각의 물체의 좌표계를 고려해야 하는 문제점을 해결할 수 있다.

## 3D Shape Reconstruction based on Superquadrics and Single Z-buffer CSG Rendering

Tae-Eun Kim\*

## Abstract

In this paper, we have proposed 3D shape reconstruction using superquadrics and single z-buffer Constructive Solid Geometry (CSG) rendering algorithm. Superquadrics can obtain various 3D model using 11 parameters and both superquadrics and deformed-superquadrics play a role of primitives which are consisted of CSG tree. In addition, we defined some effective equations using z-buffer algorithm and stencil buffer for synthesizing 3D model. Using this proposed algorithm, we need not to consider the coordinate of each 3D model because we simply compare the depth value of 3D model.

Keywords : Superquadric, CSG, z-buffer, 3D model, Depth image

## 1. 서론

3차원 시각기술은 2차원 시각 기술이 물체의 밝기에 기초한 시각 정보를 사용하는데 반해 센서로부터 물체까지의 거리에 기초한 거리 정보를 사용하여 물체와 장면을 분석함으로써 밝기 정보가 갖는 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].

기존의 3차원 형상을 기술하는 많은 방법들은 면 방향이나 면 곡률 등의 국소적인 특징들을 추출하여 물체를 표현하였지만, 복잡하고 실제적인 3차원 물체를 표현하기에는 역부족이었다 [3][4]. 이에 반해 superquadric은 크기 인자, 위치 인자, 변형 인자 등 몇 개의 계수들만으로 실세계에 존재하는 다양하고 복잡한 물체들을 효과적으로 기술하는데 유용하다[1][2][5][6].

기존의 superquadric과 CSG 트리 기법을 이용한 3차원 물체 표현에서는 implicit 함수로 정의된 집합 연산을 사용하였다[1][2][6]. 그러나 표현하려는 물체가 복잡해질수록 implicit함수의 표현 수식이 복잡해지는 단점이 있다. 뿐만 아니라 superquadric이 물체 중심 좌표계를 갖기 때문에 여러 개의 물체를 합성하는 경우 기준축에 대한 고려가 필요하다. 따라서 본 논문에서는 su

※제일저자(First Author): 김태은  
접수일자:2008년05월14일, 심사완료:2008년05월24일  
\* 남서울대학교 교수,  
tekim@nsu.ac.kr

per-quadric과 변형된 superquadric을 기본소(primitive)로 하는 CSG 트리(Constructive Solid Geometry tree) 기법을 이용하여 물체를 표현하였다. CSG 트리 구성에 필요한 집합 연산은 Z-버퍼 알고리즘과 스텐실 버퍼(stencil buffer)를 이용하여 정의한다[7][8][9]. Z-버퍼 알고리즘과 스텐실 버퍼를 이용한 CSG 트리의 경우 물체의 깊이 정보만을 비교하여 물체를 합성하기 때문에 implicit 함수를 이용하는 경우보다 물체의 표현이 더욱 효율적이다. 또한 superquadric이 갖는 대칭성에 대한 문제점도 여러 개의 물체를 변형, 합성하여 표현함으로써 비대칭적인 물체의 표현도 가능하다.

본 논문의 구성은 2장에서는 superquadric의 정의와 Z-버퍼와 CSG 트리를 이용한 물체 표현을 다룬다. 3장에서는 3차원 물체 형상 모델링을 위한 표면 shading을 설명한다. 4장에서는 실험 결과를 보였으며, 마지막으로 5장에서는 결론을 맺는다.

## 2. 3차원 물체의 표현

### 2.1. Superquadric의 정의

Superquadric 표면은 식(1)과 같이 3차원 벡터로 정의된다. 벡터  $X(, )$ 는 물체중심 좌표를 가지며, 독립변수인  $\eta, \omega$ 의 크기가 주어진 간격으로 변화할 때 공간에 폐면 (closed surface)을 형성한다. 식(1)에서  $a_1, a_2, a_3$ 는 x, y, z 축 방향의 크기,  $\epsilon_1, \epsilon_2$ 는 각각 수직, 수평 방향의 형태,  $\eta$ 는 수직 방향의 각도로써  $[-\pi/2, \pi/2]$ 의 범위를 갖고,  $\omega$ 는 수직 방향의 각도로써  $[-\pi, \pi]$ 의 범위를 갖는다.

$$X(\eta, \omega) = \begin{bmatrix} a_1 \cos^{\epsilon_1}(\eta) \cos^{\epsilon_2}(\omega) \\ a_2 \cos^{\epsilon_1}(\eta) \sin^{\epsilon_2}(\omega) \\ a_3 \sin^{\epsilon_1}(\eta) \end{bmatrix} \quad (1)$$

### 2.2. Implicit Function

식(1)은 superquadric 표면의 파라미터 방정식이다.  $\cos^2(\eta) + \sin^2(\eta) = 1$ 을 이용하여  $\eta$ 를 제거하면 식(2)와 같은 implicit 함수를 구할 수 있다. (그림 1)(a)는 식(2)를 이용해서 z축의 스

케일이 큰 실린더 모델을 생성한 예를 보여 주고 있다.

$$F(x, y, z) = \left[ \left[ \left( \frac{x}{a_1} \right)^{2/\epsilon_2} + \left( \frac{y}{a_2} \right)^{2/\epsilon_2} \right]^{\epsilon_1} + \left( \frac{z}{a_3} \right)^{2/\epsilon_1} \right]^{\epsilon_1} \quad (2)$$

### 2.3. Superquadric의 변형

#### 2.3.1 점감 (Tapering)

점감은 식(3)의 (a)와 같이 회전 중심축(z축) 함수에서 점감 정도(기울기)를 계산하고, 이 계산 결과를 이용해 중심축과 직각인 두 방향(x, y축)의 좌표를 식(3)의 (b), (c)와 같이 증가 또는 감소시킨다. (그림 1)(a)의 실린더 모델에 점감을 적용한 결과는 (그림 1)(b)와 같다.

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

#### 2.3.2 꼬임 (Twisting)

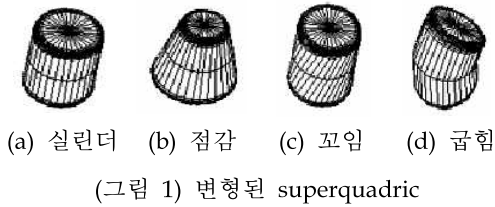
꼬임은 식(4)의 (a)와 같이 회전 중심축과 꼬임 인자  $t_w$ 의 함수로부터 계산된 꼬임 각도를 이용해, 중심축과 직각인 두 방향(x, y축)의 좌표를 회전시키는 것이다. (그림 1)(a)의 실린더 모델에 꼬임을 적용한 결과는 (그림 1)(c)와 같다.

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

#### 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. (그림 1)(a)의 실린더 모델에 굽힘을 적용한 결과는 (그림 1)(d)와 같다.

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



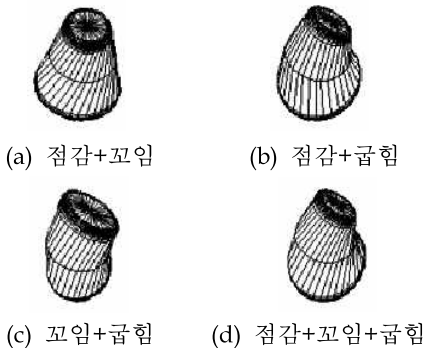
(그림 1) 변형된 superquadric

2.3.4 복합 변형

복합 변형은 수식(6)과 같이 점감 (TA), 꼬임 (TW), 굽힘 (BE), 이동 (TR)의 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. 복합 변형된 superquadric 표면상의 점은 식(7)과 같이 원래 superquadric의 표면상의 점으로 복구될 수 있다. (그림 2)는 복합 변형을 적용한 다양한 결과를 보여 주고 있다.

$$X' = TR \cdot BE \cdot TW \cdot TA \cdot X = D(x) \cdot X \quad (6)$$

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



(그림 2) 복합 변형된 superquadric

2.4. Z-버퍼 알고리즘

3차원 물체를 현실감 있게 포착하기 위해서는 특정 방향에서 관찰자에게 보이지 않는 모서리와 면들을 제거해야 한다. 어떠한 모서리나 면들을 제거할 것인가를 결정하는 작업을 은선/은면 제거라 한다[10][11].

Z-버퍼 알고리즘은 가장 간단한 은면 제거 알

고리즘의 하나로 가장 작은 Z값으로 Z-버퍼를 초기화하고 모든 화소의 위치에서 버퍼에 저장되어 있는 값과 화상의 Z값을 비교한다. 만약 새로운 z값이 기존의 값보다 크다면, 이는 조사하고 있는 다각형이 관찰자로부터 더 가까운 것이므로, Z-버퍼는 이 새로운 값으로 대체된다. 이렇게 연산된 결과로 화상 공간 (screen space)에서 모든 다각형이 조사되고 나면, Z-버퍼는 은면 제거 문제에 대한 해를 갖게 된다. Z-버퍼 알고리즘은 물체 표면의 각 점에 대한 깊이 정보를 계산할 수 있는 모든 형태에 적용이 가능하다. 뿐만 아니라 개별적으로 표현된 3차원 물체들의 Z-버퍼 정보와 CSG를 이용한 모델링이 가능하다.

2.5. Superquadric간의 집합 연산

기존의 superquadric을 기초로 한 3차원 물체 표현 방법에서는 superquadric과 변형된 superquadric의 집합에 의해서 물체를 표현하였다. 이러한 방법은 단순히 3차원 공간상의 이동에 의해 정의되는 집합에 의해서만 가능하여 물체 표현이 비효율적인 단점을 갖는다. 그러나 superquadric과 변형된 superquadric간의 집합 연산을 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 정의할 수 있다. 여러 개의 입체들을 합성 할 때, 식(2)에서 주어진 implicit함수는 변형되기 전 영상의 점들에 대해서만 만족하기 때문에 영상간의 CSG 집합 연산을 하기 전에 변형된 영상을 식(7)을 이용하여 원래 영상의 점으로 변환한 뒤 implicit함수를 이용한다.

2.5.1 합집합 (Union)

입체간의 합집합에서는 식(8)과 같이 물체와 물체가 만나서 생기는 교집합에 해당하는 표면의 점들이 제거한다. 따라서 단순히 물체들간의 깊이 정보만을 비교하기 때문에 Z-버퍼 알고리즘만 이용하여 표현이 가능하다. (그림 3)(a)는

CSG를 이용한 합집합 결과를 보여주고 있다.

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

### 2.5.2 교집합 (Intersection)

교집합은 합집합과 반대로 식(9)와 같이 물체와 물체가 만나서 생기는 표면의 점들만 남긴다. 따라서 물체 A와 B로 둘러싸인 체적을 표현한다. (그림 3)(b)는 CSG를 이용한 교집합 결과를 보여주고 있다.

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

### 2.5.3 차집합 (Difference)

차집합은 식(10)과 같이 교집합 연산의 반대과정으로 생각할 수 있다. 즉, 물체 A와 물체 B의 여집합과의 교집합과 같다. (그림 3)(c)는 CSG를 이용한 차집합 결과를 보여주고 있다.

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



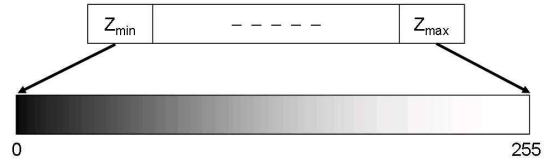
(a) 합집합 (b) 교집합 (c) 차집합  
(그림 3) 집합 연산 결과

## 3. 3차원 정보의 표현

### 3.1 Z값의 gray-level 변환

Superquadric을 이용하여 얻은 3차원 깊이값(Z)은 물체를 중심으로 대칭인 관계에 있다. 따라서 깊이 정보를 물체의 표면에 나타내려면 (그림 4)와 같이 깊이값의 범위를 256 gray-level로 변환해야 한다. 식(11)은 변환식을 보인다.

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



(그림 4) 256 gray-level 분포도

## 3.2 축 보간

Superquadric의 모양을 결정짓는  $\epsilon_1, \epsilon_2$ 의 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.

### 3.2.1 z축 보간

$\epsilon_1 = 1.0$ 인 경우 z축을 따라 발생하는 점들간의 간격은 일정하지만,  $\epsilon_1 \ll 1.0$ 인 경우에는 z축 양끝으로 모여드는 양극화 현상을 보인다. 따라서 두 점 사이의 간격이 문턱치 이상이면 두 점 사이를 일정한 간격으로 보간 한다.

### 3.2.2 x, y축 보간

z축 보간과 마찬가지로  $\epsilon_2 = 1.0$ 인 경우 x, y축을 따라 발생하는 점들의 간격은 일정하지만,  $\epsilon_2 \ll 1.0$ 인 경우 발생하는 점들은 서로 다른 분포 특성을 보인다. 따라서 두 점 사이의 간격이 문턱치 이상이면 두 점의 중점에 강제로 하나의 점을 추가시켜서 보간 한다.

## 4. 실험 및 고찰

### 4.1 축 보간

Superquadric을 이용하여 생성한 3차원 모델은 기본소에 따라서 생성되는 3차원 점의 위치가 다르다. (그림 5)(a)는 superquadric을 이용하여 실린더 모델을 생성한 결과이다. 그림에서 알 수 있듯이, x, y 축 방향으로는 3차원 점의 분포가 많지만, z축 방향으로는 3차원 점의 분포가 낮다. 그 결과, 이 모델에 대해 식(11)을 이용해서 깊이값을 정의하면 (그림 5)(b)와 같이 자연스럽지 못한 거리 영상이 만들어진다. 따라서 실

린더 모델의 경우에는 (그림 5)(c)와 같이 z축에 대한 보간이 필요하다. 그 결과, 보다 자연스러운 거리 영상을 얻을 수 있다.

(그림 6)(a)는 superquadric을 이용하여 육면체 모델을 생성한 결과이다. 육면체 모델의 경우에는 실린더 모델과 반대로 x,y축에 대한 3차원 점의 분포가 낮다. 그 결과, 이 모델에 대해 식 (11)을 이용해서 깊이값을 정의하면 (그림 6)(b)와 같이 수평 평면 영역에서 자연스럽게 못한 거리 영상이 만들어진다. 따라서 육면체 모델의 경우에는 (그림 6)(c)와 같이 x-y축에 대한 보간이 필요하고, z축 보간 결과 (그림 6)(d)와 같이 보다 자연스러운 거리 영상을 얻을 수 있다.



(a) 보간 전 wireframe (b) 보간 전 거리영상



(c) 보간 후 wireframe (d) 보간 후 거리영상

(그림 5) Z축 보간 전후의 wireframe과 거리영상



(a) 보간 전 wireframe (b) 보간 전 거리영상



(c) 보간 후 wireframe (d) 보간 후 거리영상

(그림 6) x, y축 보간 전후의 wireframe과 거리영상

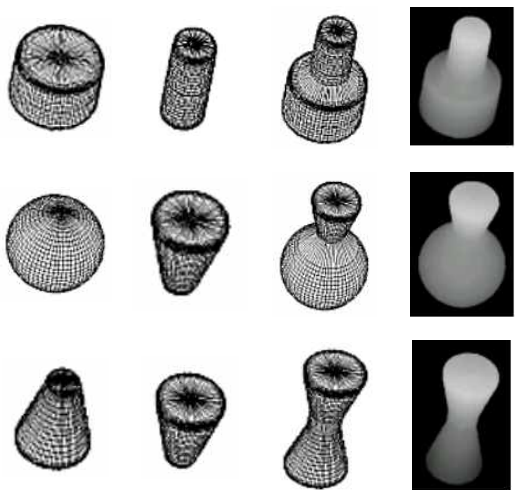
#### 4.2 3차원 모델링

여러 개의 기본소를 이용해서 3차원 물체를 모델링하기 위해서는 우선 합성할 기본소의 개수를 정하고, 각각의 기본소를 구성하는 superquadric의 계수를 입력받는다. 그리고 각각의 기본소들간의 집합 연산을 결정하여 4장에서 정의한 알고리즘을 이용해 개개의 물체를 합성한다.

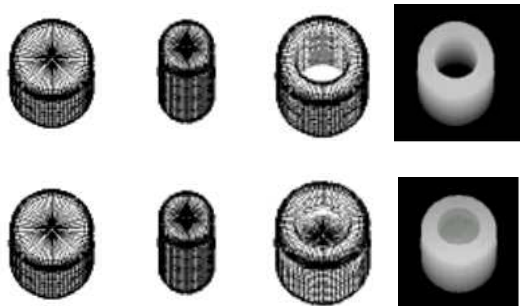
(그림 7)은 superquadric으로 생성한 다양한 기본소 모델에 대해서 합집합 연산을 통해 만든 3차원 모델을 보여주고 있다. 그림에서 알 수 있듯이, 기본소 모델에 대해서 합집합 연산을 통해 다양한 모델 합성이 가능할 뿐만 아니라 만들어진 모델의 집합 부분이 자연스럽게 이어져 있음을 알 수 있다. 그리고 합성된 모델에 깊이값에 대해서 거리 영상을 합성한 결과는 (그림 5), (그림 6)에 비해서 보다 자연스럽다. 이는 각 점들이 이루고 있는 면에 대해서 지정된 깊이값을 주변 면과의 선형 보간을 통해서 보다 부드럽게 표현한 결과이다.

(그림 8)은 superquadric으로 생성한 다양한 기본소 모델에 대해서 차집합 연산을 통해 만든 3차원 모델을 보여주고 있다. 두 결과 모두가 서로 다른 실린더 모델을 대해서 차집합을 수행한 결과이다. 그러나 첫 번째 결과는 작은 실린더가 큰 실린더를 관통한 모델이고, 두 번째 결과는 관통하지 않은 결과이다. 이처럼 사용자가 원하는 다양한 형태의 모델을 생성 가능함을 알 수 있다.

(그림 9)는 3개로 구성된 기본소들에 대해서 컵을 모델링하기 위한 집합연산 과정을 보여주고 있다. 우선 컵을 만들기 위해 크기가 서로 다른 실린더 모델을 superquadric 모델을 생성한 다음 두 실린더에 대해서 차집합 연산을 수행한다. 그리고 컵의 손잡이를 만들기 위해 실린더 모델에 굽힘을 수행하여 손잡이 모델을 만든다. 그리고 최종적으로 컵과 손잡이를 합치기 위해 두 모델을 합집합 연산을 수행한다. 그 결과, (그림 9)의 맨 위에 보이는 것처럼 컵 모델을 얻을 수 있다. (그림 10)은 이러한 과정을 통해 얻을 수 있는 다양한 모델과 합성 거리 영상들을 보여 주고 있다.



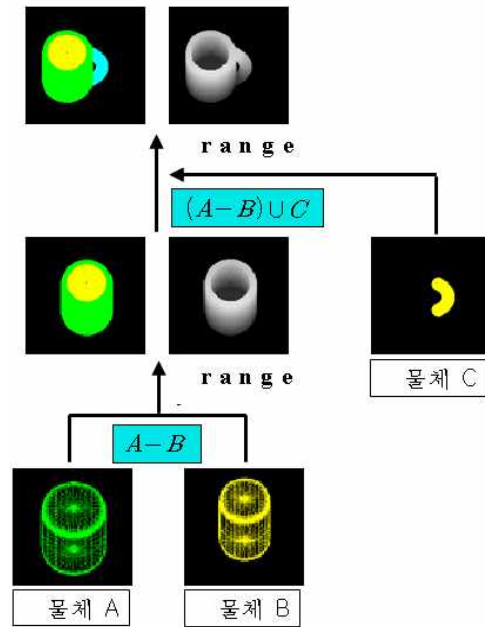
(a) 물체 1 (b) 물체 2 (c) 합집합 (d) 거리영상  
(그림 7) 변형된 superquadric간의 합집합과 거리



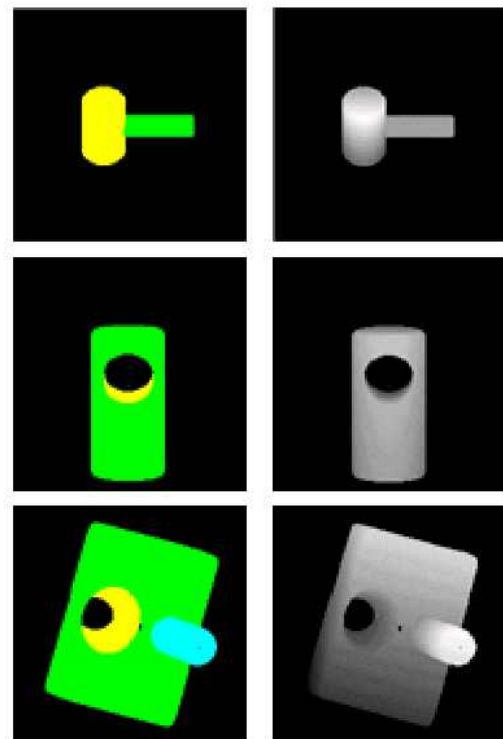
(a) 물체 1 (b) 물체 2 (c) 차집합 (d) 거리영상  
(그림 8) 변형된 superquadric간의 차집합과 거리

### 5. 결론

본 논문에서는 superquadric으로 얻어진 3차원 정보와 하나의 z-버퍼 렌더링 알고리즘을 이용한 3차원 물체 표현 기법을 제안하였다. 기존의 implicit함수를 이용한 CSG 트리 기법에서는 물체의 이동이나 변형에 대해서 implicit 함수의 수식이 복잡하며 기준 좌표계에 대한 고려가 필요하다. 따라서 물체의 자유로운 표현이 불가능하였다. 그러나 Z-버퍼 알고리즘과 CSG 렌더링 알고리즘을 이용하면 물체의 회전이나 이동, 변형에 따른 변화에 상관없이 물체의 깊이 정보만



(그림 9) 3개 물체에 대한 CSG 집합연산



(그림 10) 다양한 기본소를 이용한 3차원 모델과 거리영상

을 비교하여 합성하고 표현하기 때문에 기존의 방법보다 더욱 효율적으로 3차원 모델을 표현할 수 있다. 특히 Z-버퍼와 스텐실 버퍼를 이용한 CSG 트리 기법은 보다 복잡한 실세계 물체 표현을 가능하게 했으며, 대칭성의 문제를 갖는 superquadric의 단점을 보완, 비대칭인 물체의 표현을 가능하게 했다. 본 논문에서 제안한 알고리즘을 이용해서 다양한 3차원 모델에 대한 모델링이 가능할 뿐만 아니라 합성 거리 영상을 표현할 수 있음을 실험 결과를 통해서 보였다. 향후에는 실사 영상으로부터 3차원 모델을 획득하는 부분에 대한 연구가 진행되어야 할 것으로 판단된다.

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] 송호근, 최종수, "미분 기하학 특징 정보를 이용한 3차원 거리 영상 분할", 대한 전자공학회 논문지, 제31권, B편, 제2호, pp.123-134, 1995

[4] 박중훈, 최종수, "면 법선 영상 기반형 3차원 물체 인식에서의 새로운 매칭 기법", 대한 전자공학회 논문지, 제30권, B편, 제7호, pp.695-705, 1993

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

[6] 김대현, 현대환, 이선호, 최종수, "Superquadric과 CSG에 기반한 3차원 물체 형상 모델링", 1999년도 하계통신학회 학술대회 논문, Vol.19, No.1 pp.521-524

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

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

[9] Tom McReynolds, SGI, "Programing with OpenGL: Advanced Rendering", SIGGRAPH '96 Course

[10] Vera B. Anand, "Computer Graphics and Geometric

Modeling for Engineers", WILEY

[11] A. Watt, F. Policarpo, "The Computer Image", Addison-Wesley

[12] I. Biederman, "Human image understanding: recent research and theory", Computer Vision, Graphics, Image Processing, Vol.32, No.32, pp.29-73, 1985



김 대 은

1989년: 중앙대학교 전기공학과 공학사  
 1992년: 중앙대학교 전자공학과 공학석사  
 1997년: 중앙대학교 전자공학과 공학박사

1993~1996년 : 한국재단참여연구원  
 1995년 : 삼성전자 휴먼테크논문 대상은상수상  
 1997~현 재 : 남서울대학교 멀티미디어학과 교수  
 관심분야 : 멀티미디어시스템, 영상인식, 증강현실, 웹3D처리기술