

Part zero: Introduction

1. Intro

Outline

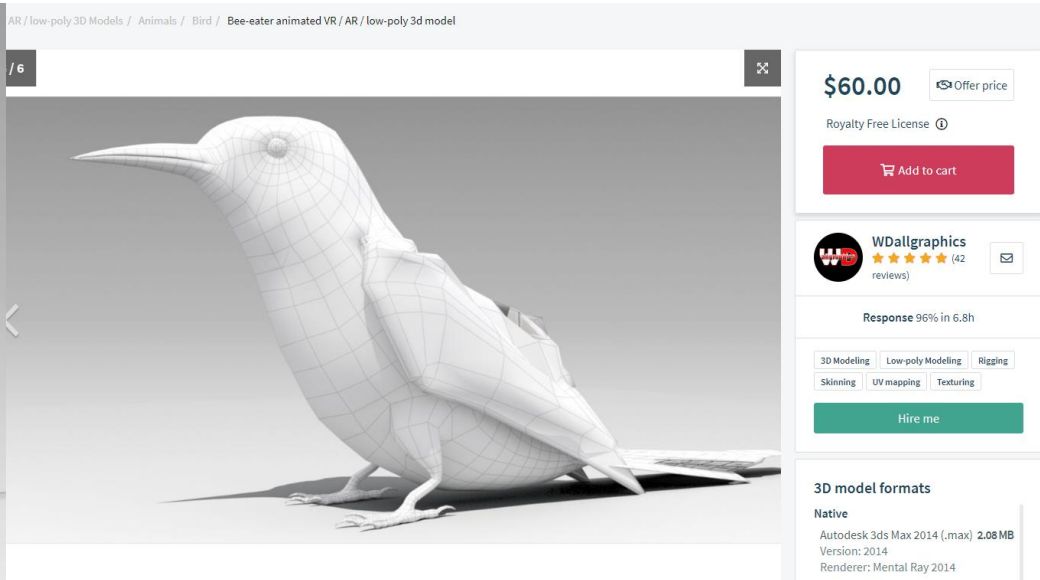
- I. Graphics
- II. Modeling
- III. Rendering

o. Real? or Fake?

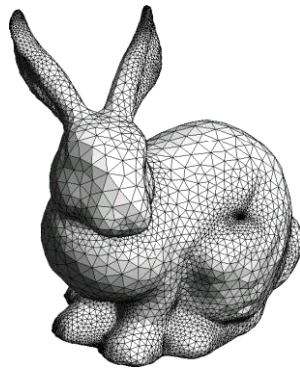


사진? Graphic?

o. Real? or Fake?



1. Graphics



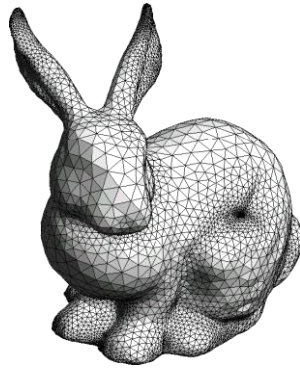
Modeling



Rendering

재질, 색깔 바꾸기.

1. Graphics



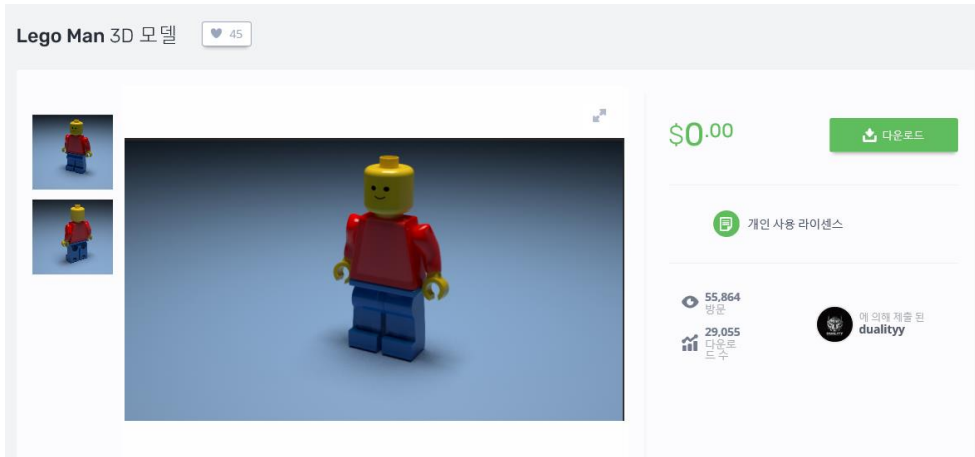
Modeling



Rendering



1. Graphics



<https://free3d.com/3d-model/lego-man-8986.html>

오브젝트 파일 → 텍스트 파일 →
(~.obj)



vertex.
 3개씩 한 쌍으로
한 포인트.
(바탕색, 빨라, 노랑색)

바탕색을 이루는 vertex의 개수.

→ 삼각형 총 개수 계산.

↓

정 세개
위에 면 생각.

```

v 29.2550 63.0786 4.4300
v 31.8154 63.0786 4.4300
v 31.8104 63.0786 4.2768
v 31.7952 63.0786 4.1243
v 31.7700 63.0786 3.9729
v 31.7348 63.0786 3.8234
v 31.6899 63.0786 3.6763
v 31.6353 63.0786 3.5322
v 31.5712 63.0786 3.3917
v 31.4981 63.0786 3.2553
v 31.4160 63.0786 3.1236
v 31.3254 63.0786 2.9971
v 31.2266 63.0786 2.8764
v 31.1200 63.0786 2.7618
v 31.0061 63.0786 2.6540
v 30.8853 63.0786 2.5532
v 30.7580 63.0786 2.4598
v 30.6249 63.0786 2.3744
v 30.4864 63.0786 2.2970
v 30.3431 63.0786 2.2282
v 30.1956 63.0786 2.1680
v 30.0445 63.0786 2.1167
v 31.7348 64.3786 5.0366
v 31.7700 64.3786 4.8871
v 31.7952 64.3786 4.7357
v 31.8104 64.3786 4.5832
# 6302 vertices

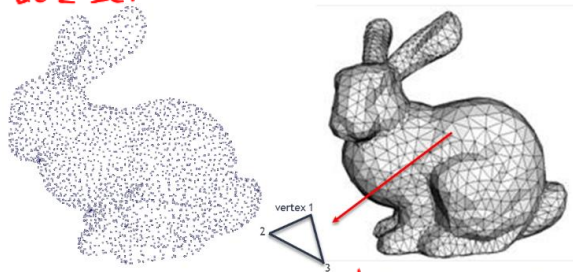
vn 0.0000 1.0000 -0.0000
vn -0.0072 1.0000 0.0005
vn -0.0472 0.9989 0.0031
vn -0.0474 0.9989 -0.0000
vn -0.0072 1.0000 -0.0000
vn -0.0071 1.0000 0.0009
vn -0.0469 0.9989 0.0061
vn -0.9864 0.0000 -0.1643
vn -0.9951 0.0000 -0.0989
vn -0.9995 0.0000 -0.0330
# 6205 vertex normals

vt 10.4836 7.5263 0.0000
vt 10.5039 7.5250 0.0000
vt 10.5241 7.5238 0.0000
# 3407 texture coords

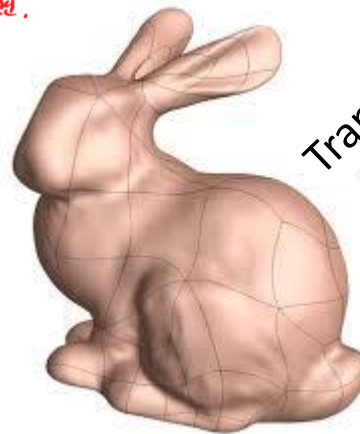
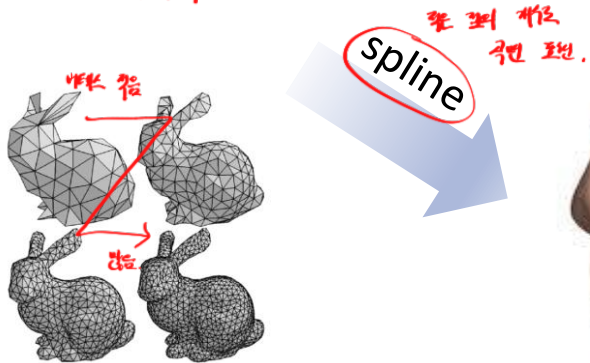
f 6289/88/6192 6290/89/6192 89/89/6192 88/88/6192
f 6290/89/6193 6291/90/6193 90/90/6193 89/89/6193
f 6291/90/6194 6292/91/6194 91/91/6194 90/90/6194
f 6292/91/6195 6293/92/6195 92/92/6195 91/91/6195
f 6293/92/6196 6294/93/6196 93/93/6196 92/92/6196
f 6294/93/6197 6295/94/6197 94/94/6197 93/93/6197
f 6295/94/6198 6296/95/6198 95/95/6198 94/94/6198
f 6296/95/6199 6297/96/6199 96/96/6199 95/95/6199
f 6297/96/6200 6298/97/6200 97/97/6200 96/96/6200
f 6298/97/6201 6299/98/6201 98/98/6201 97/97/6201
f 6299/98/6202 6300/99/6202 99/99/6202 98/98/6202
f 6300/99/6203 6301/100/6203 100/100/6203 99/99/6203
f 6301/100/6204 6302/101/6204 101/101/6204 100/100/6204
f 6302/101/6205 6204/2/6205 2/2/6205 101/101/6205
# 6200 polygons - 200 triangles
    
```

1. Modeling

원점으로 표현.

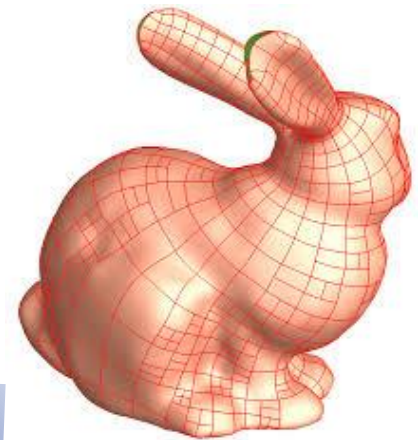


Point Cloud 3개의 point → 하나의 면.
Mesh (shaded)

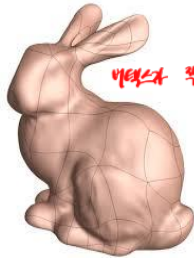
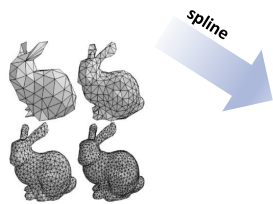
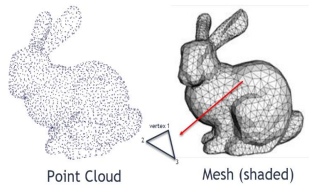


Transform

위치 변환..



1. 1 Splines



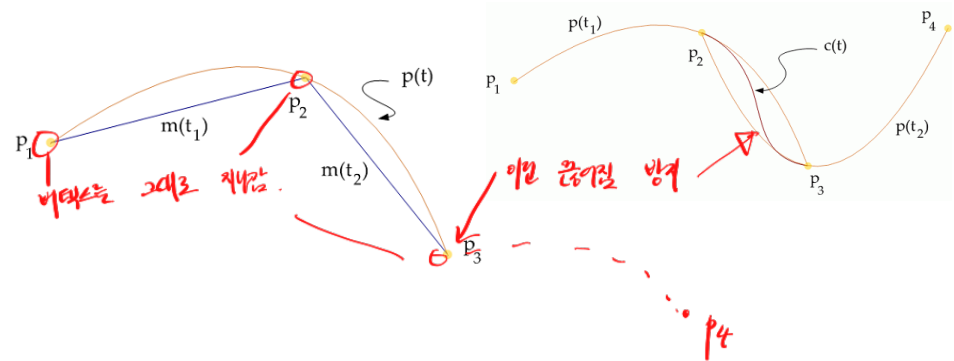
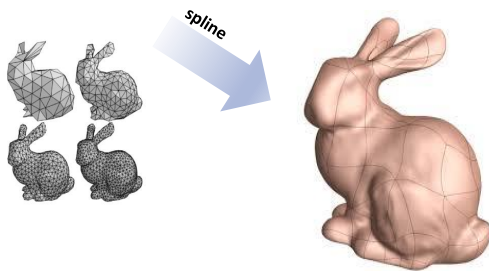
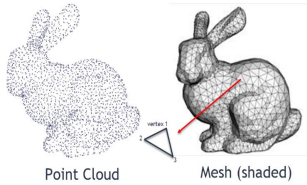
비슷하고 꼭아도 강하게 굽게
모델 로딩.

자로 곡선을 긋는다. → 곡면 띄기.

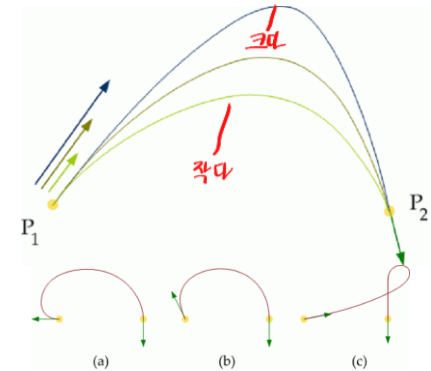
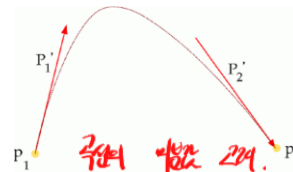


1. 1 Splines

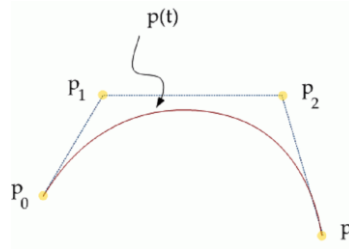
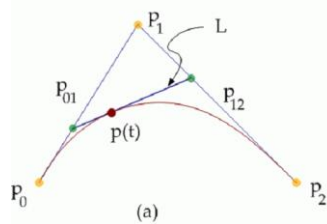
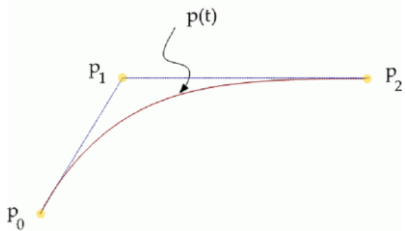
① Cardinal Splines



② Hermite Splines

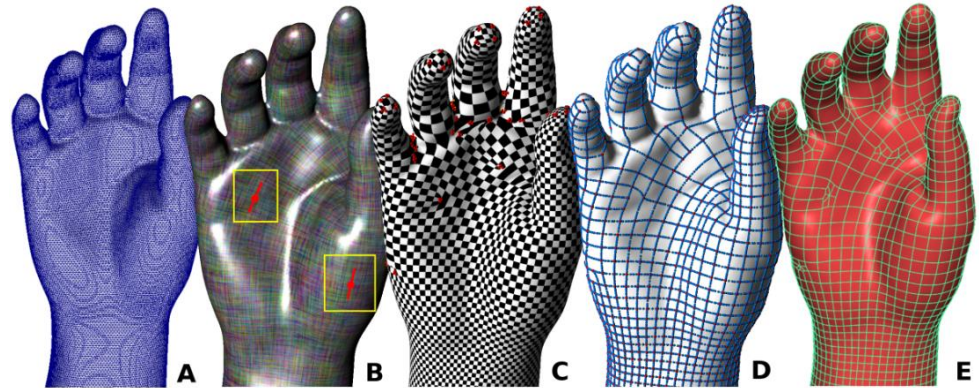
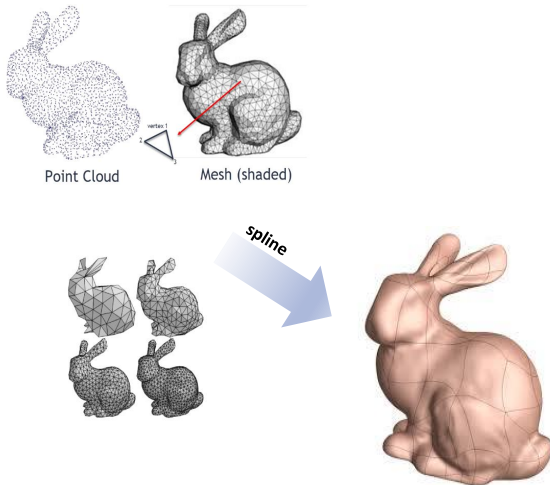


③ Bezier Splines

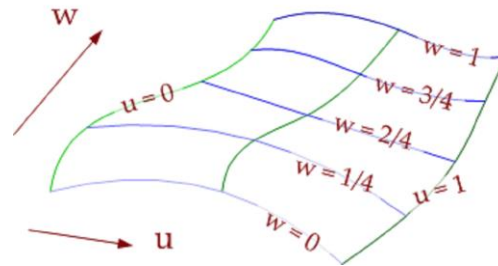


1. 2 Surfaces

Spline 곡면

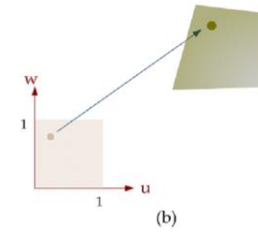
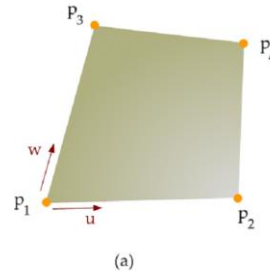
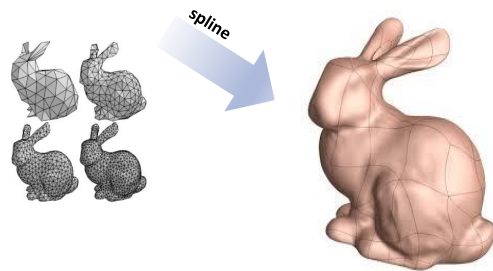
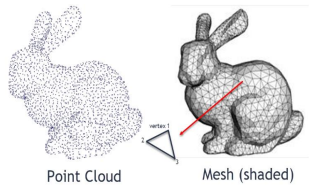


실제 물체들은 꼭짓보다 곡선의 개수를
- 그러나 곡선을 많이 사용하면 시간이 오래 걸린다.
그래픽스의 생애는 죽도!

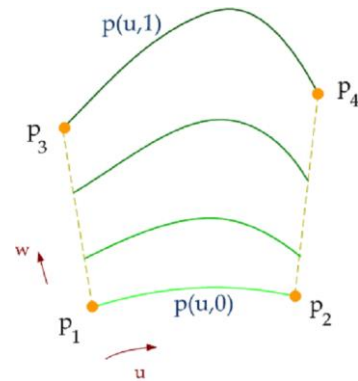


1. 2 Surfaces

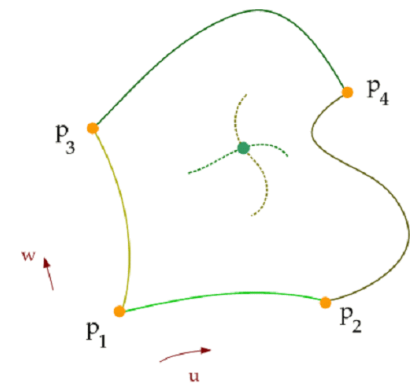
공간상 4점



2개의 공간 곡선

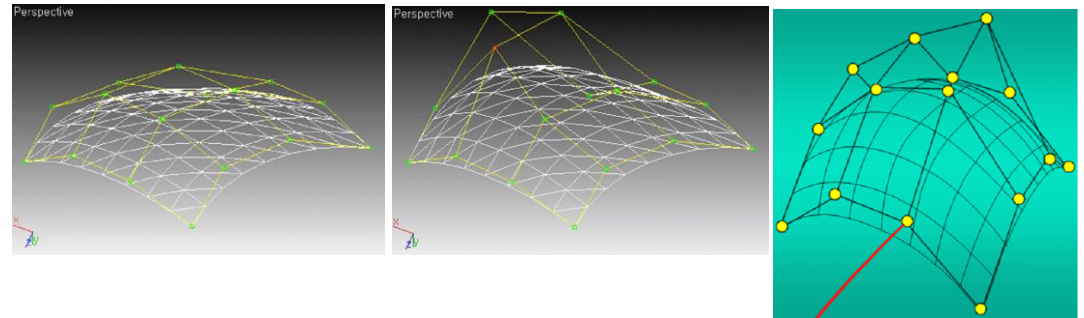
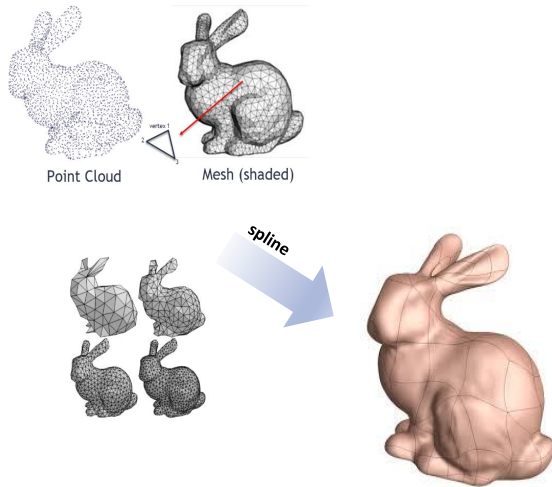


4개의 경계 곡선



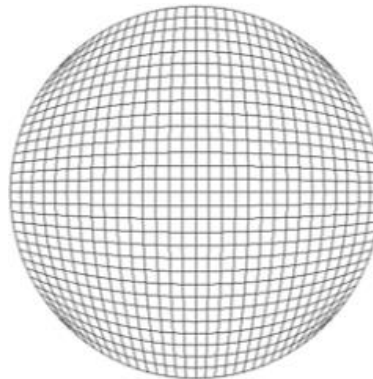
1. 2 Surfaces

Bezier spline과 비슷.
Bezier Surfaces



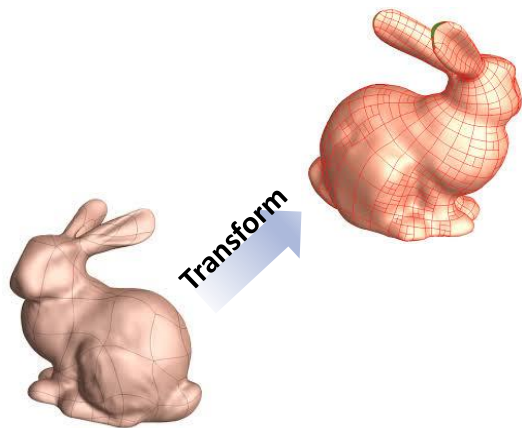
노란 점: vertex.
해당 점들을 지니지 않으면서
가장 유연한 곡면 만들기

Quadric Surfaces

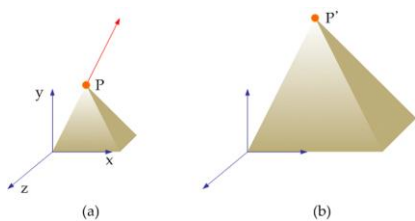


1. 3 Transform

→ 행렬을 통해 변환 가능. vertex 각 좌표를
연산해서 좌이전다.

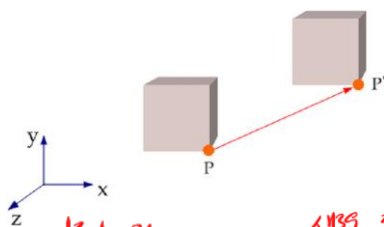


③ 크기변환 Scaling



$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

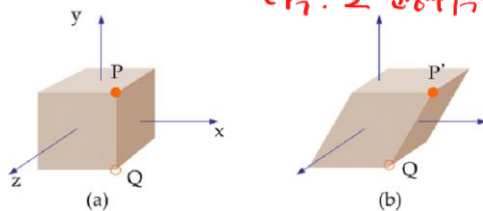
① 위치 바꾸기 Translation



기원점. 새로운 점.

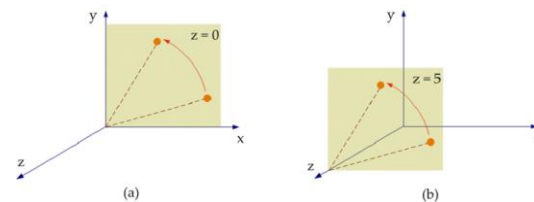
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

④ 당기기 Shearing



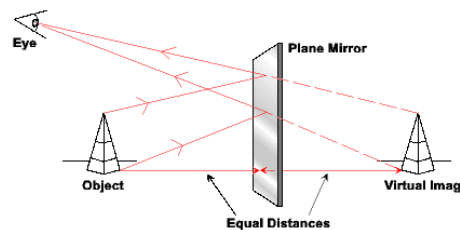
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & Sh_y & 0 & 0 \\ Sh_x & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

② 회전 Rotation



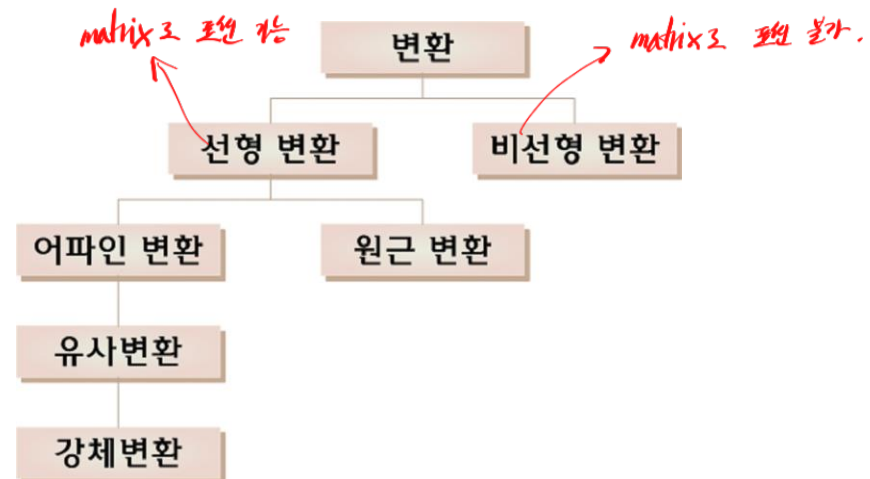
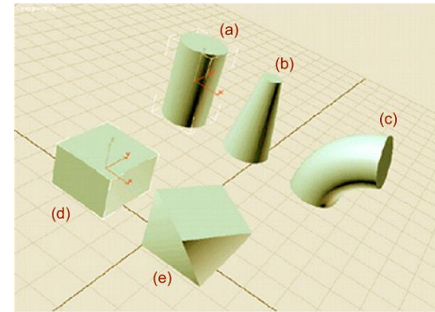
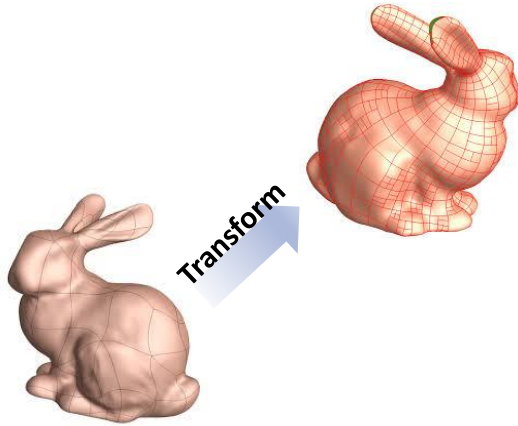
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

⑤ 반사. Reflection

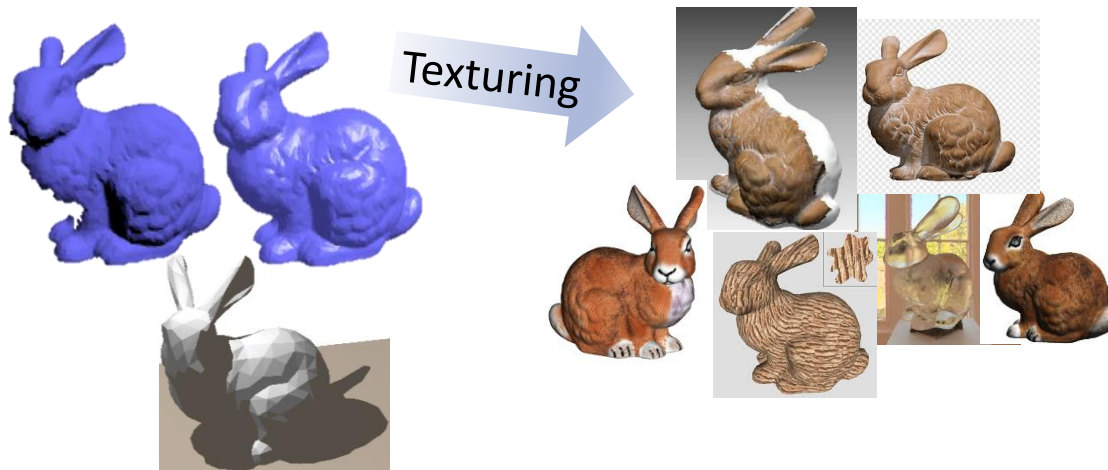
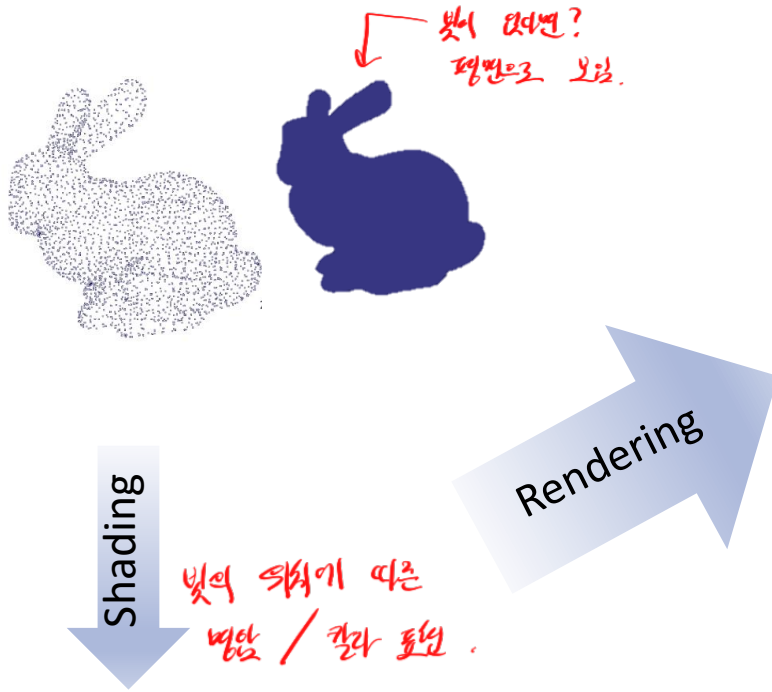


$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

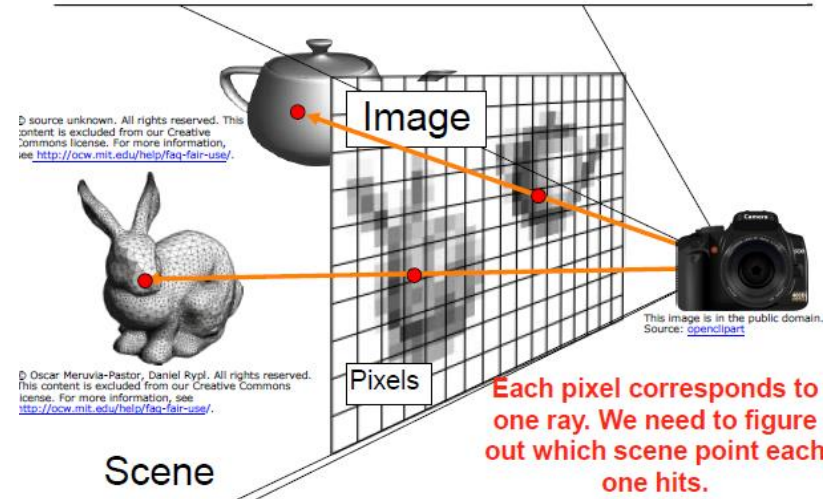
1. 3 Transform



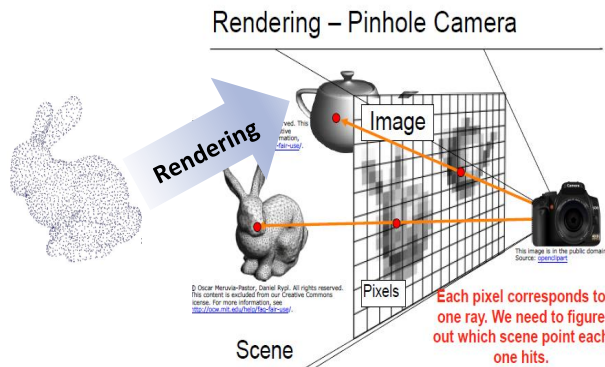
2. Rendering



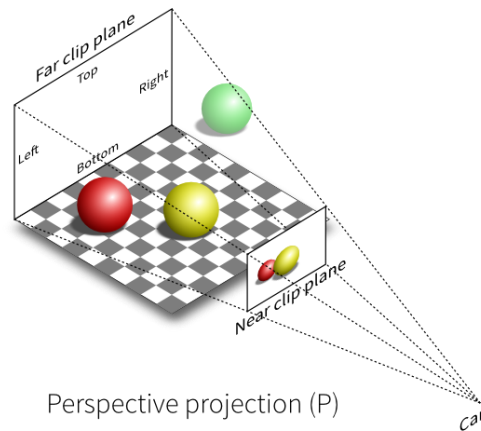
Rendering – Pinhole Camera



2.1 Projection, 투상변환



Perspective Projection



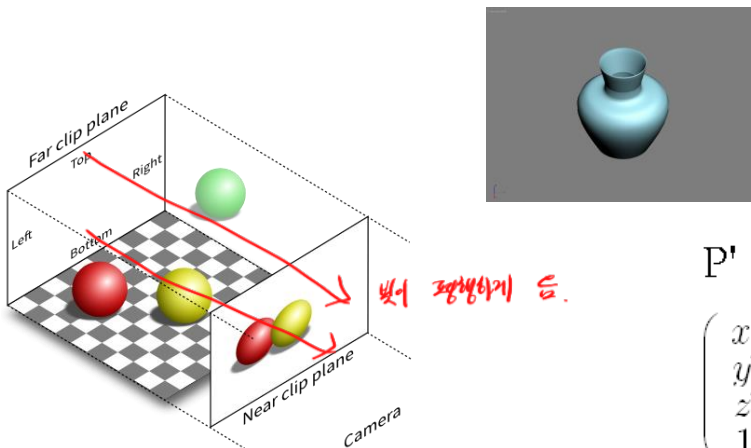
$$P' = \begin{pmatrix} x' \\ y' \\ -d \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ -z/d \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/d & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

matrix 포워드를.

우리 눈이나
카메라 같은 것
앞이 포워드.

Orthographic Projection



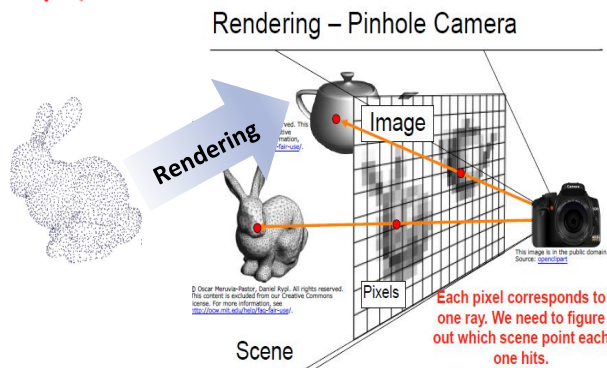
빛이 평행하게 들어.

$$P' = M_{\text{parallel}} \cdot P$$

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -d \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ -d \\ 1 \end{pmatrix}$$

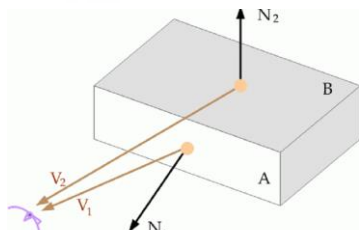
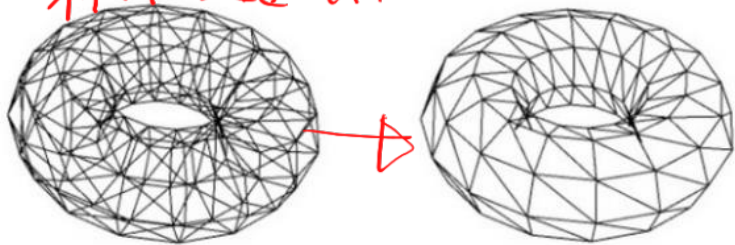
2.2 가시성 변환

가려진 부분 작기.



후면 제거

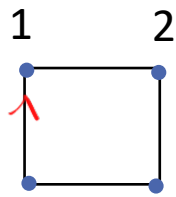
뒤쪽까지 표현된 것.



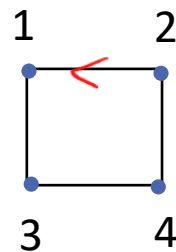
$$\text{Backface} = (N \cdot V < 0) = (|N| |V| \cos \theta < 0)$$

벡터의 비직각을 통해 판단.

반시계 방향.
시계방향
표면과 이면
그림자 같나다.

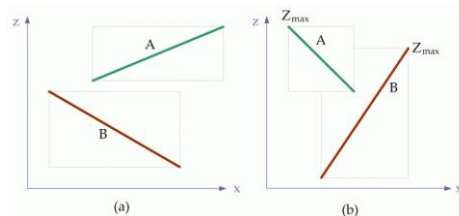
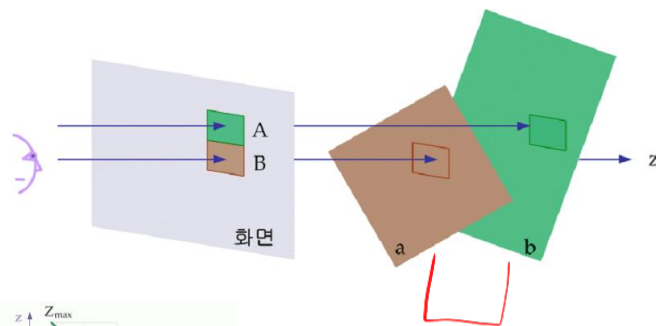


Polygon: 1, 2, 4, 3



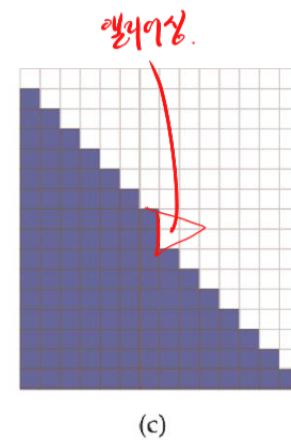
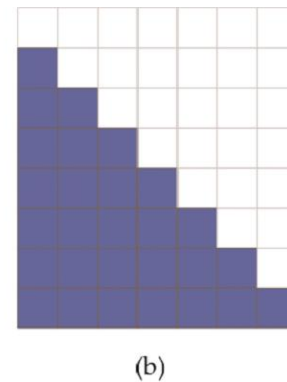
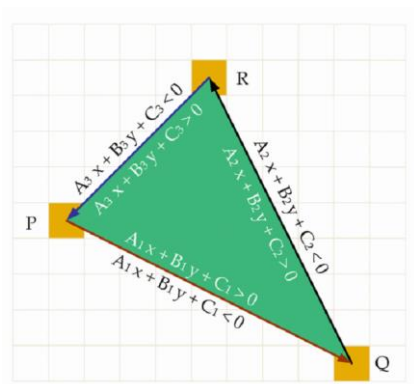
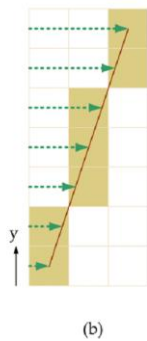
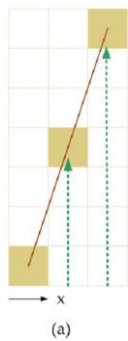
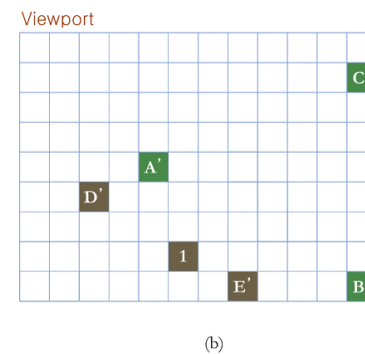
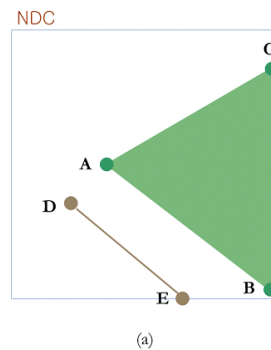
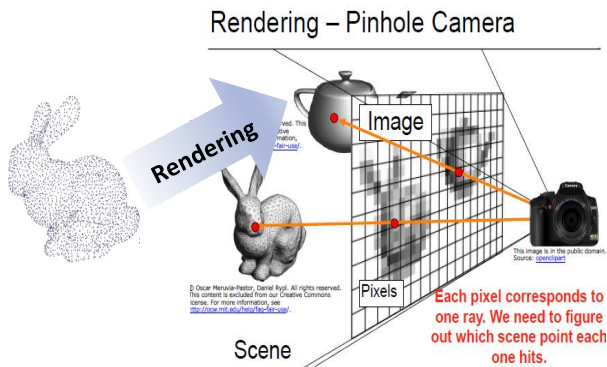
Polygon: 1, 3, 4, 2

은면 제거

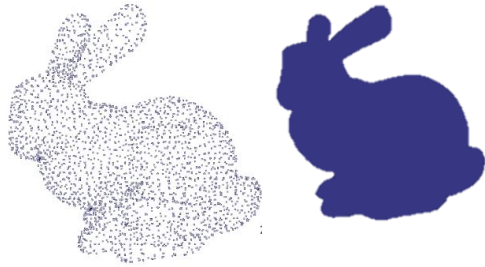


더 멀리있는데 먼저
그림자 통해 비교.

2.3 래스터 변환 → 픽셀값 할당하기.



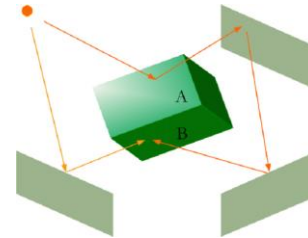
2. 4 Lighting



Shading

① 주광.

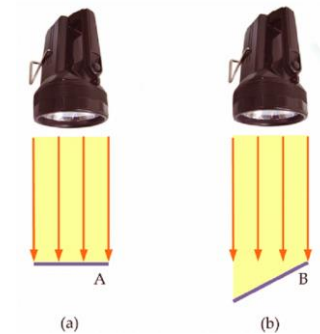
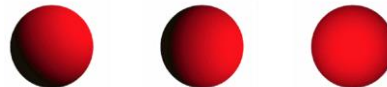
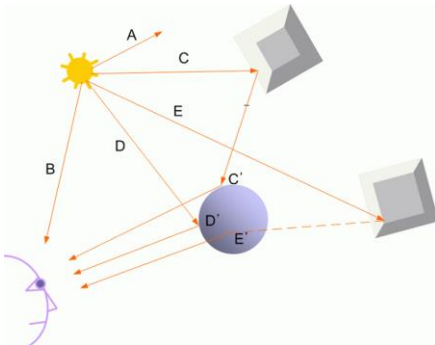
Ambient Reflection



② 확산광.

Diffusive Reflection

빛의 방향을 알 수 없음.



Specular Reflection

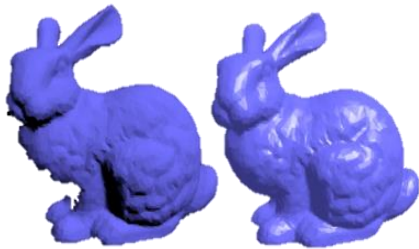
확산광

주광

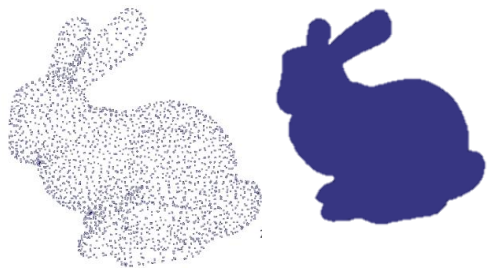


(a)

(b)

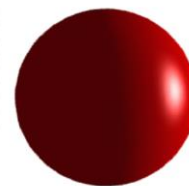
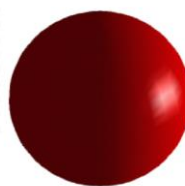


2. 5 Shading (음영), Shade (그림자)

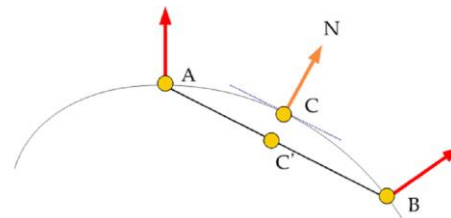
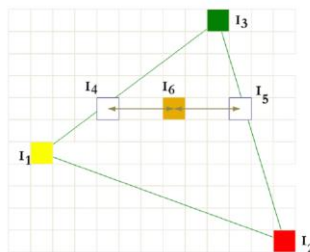
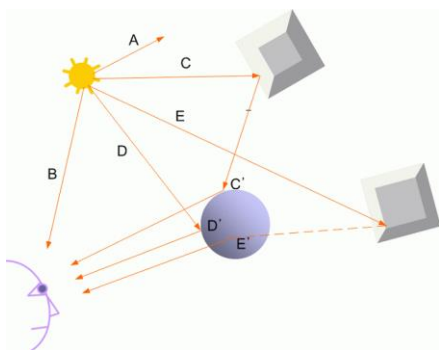


→ 면의 음영을 줌.
플랫, 구로, 폰 세이딩

→ 백터 값의 차이점을 이용.

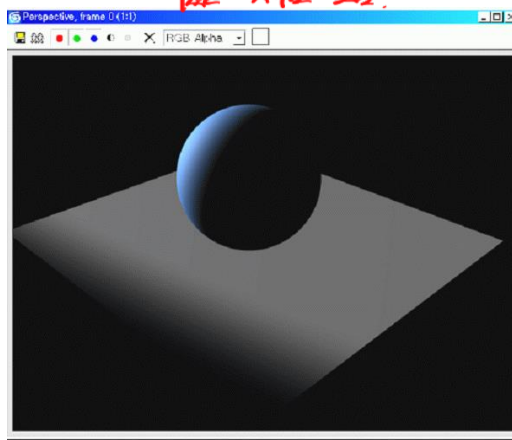
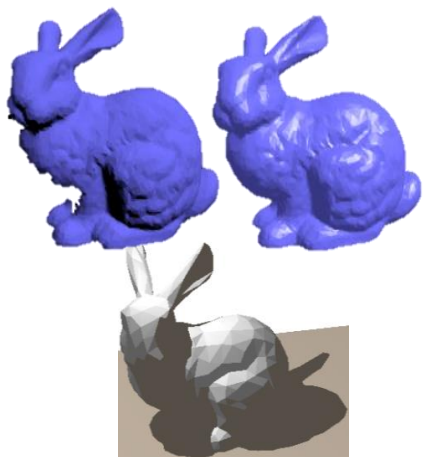


Shading

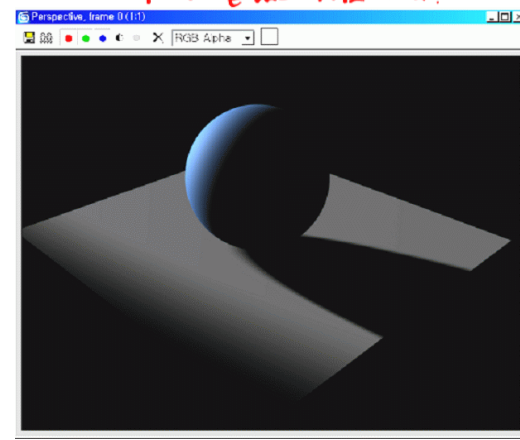


그림자

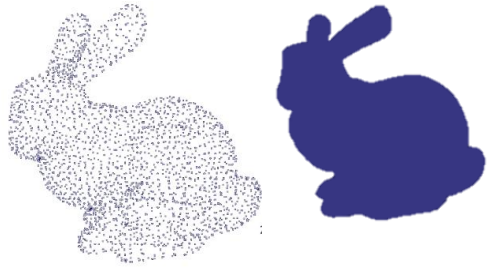
어두운 것처럼 보임.



거리에 비례하는 것들 보임.



2.6 Texture Mapping

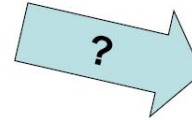
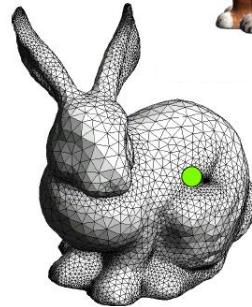


UV Texture

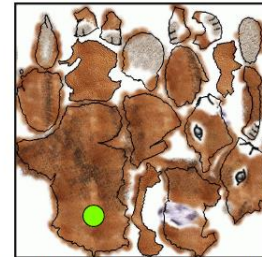
Texture mapped model



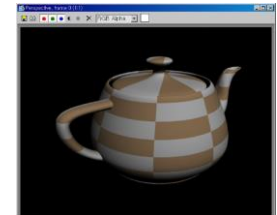
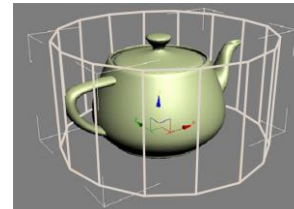
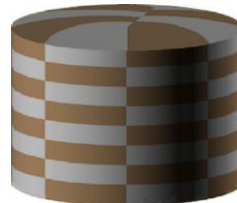
vertex 마다 color 할당.



Texture map (2D image)



다각형 곡면



Shading

Texturing

이 단계.

