Part2: Rendering

6. Texture

Outline

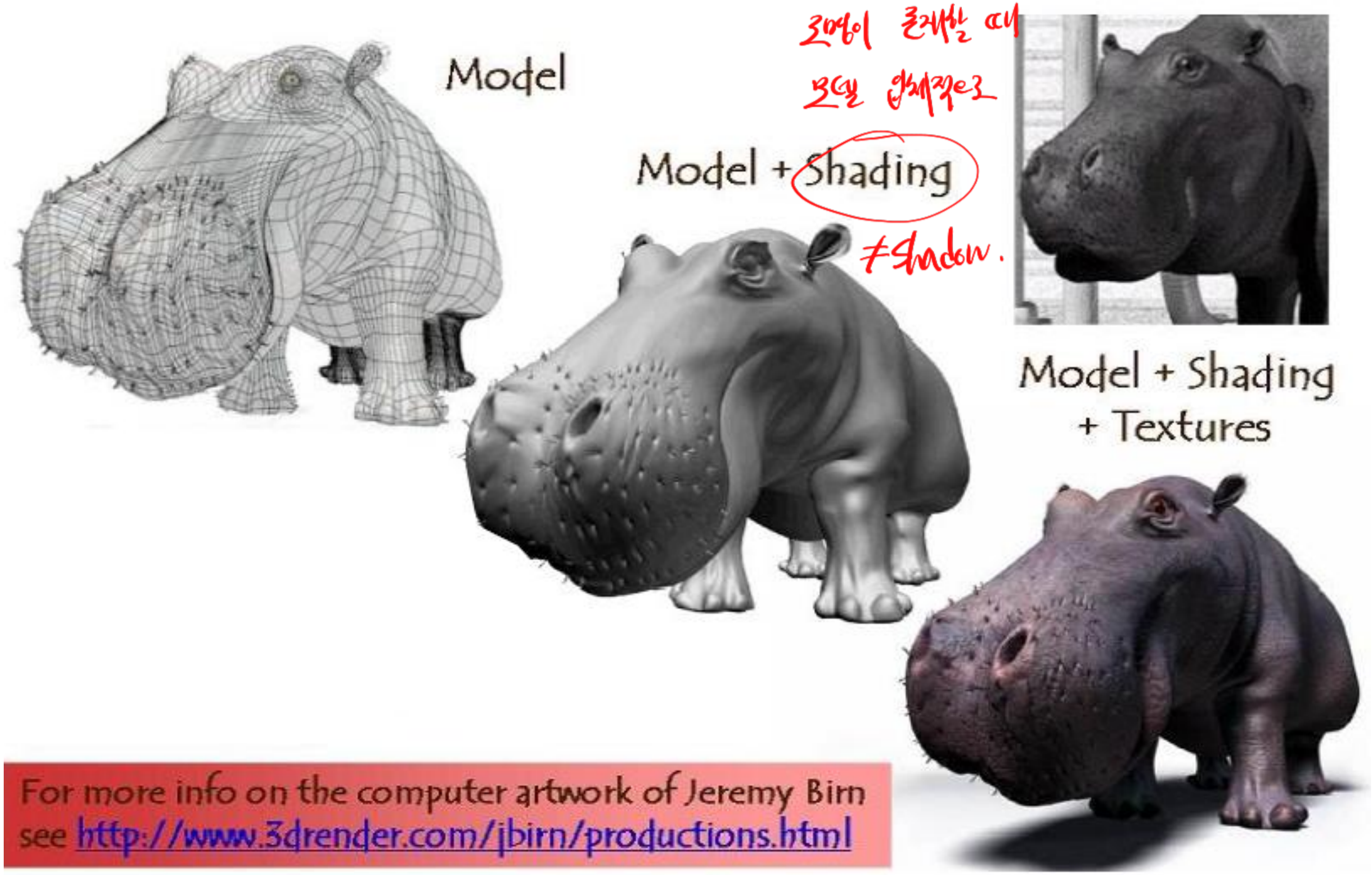
I. Texture Mapping

II. UV Texture Mapping

III. Parametric Texturing

1. Texture Mapping

- Effect of Textures



조명이 존재할 때
모델 입체적으로

≠ Shadow.

For more info on the computer artwork of Jeremy Birn see http://www.3drender.com/jbirn/productions.html

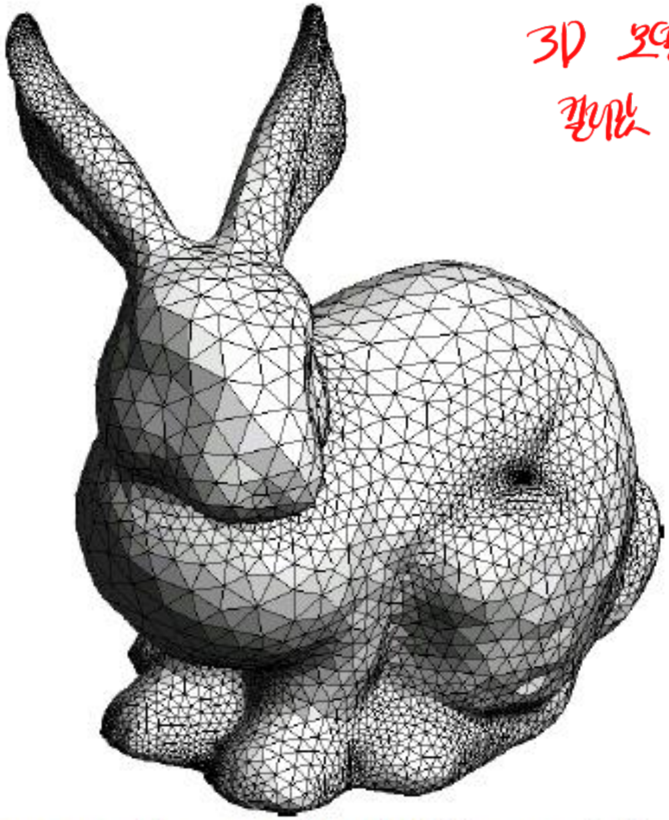
Courtesy of Jeremy Birn

Wojciech Matusik, and Frédo Durand. *6.837 Computer Graphics.* Fall 2012. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC-SA.

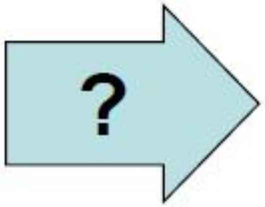
1. Texture Mapping

3D model

Texture mapped model



3D 모델에
컬러값 입히기.



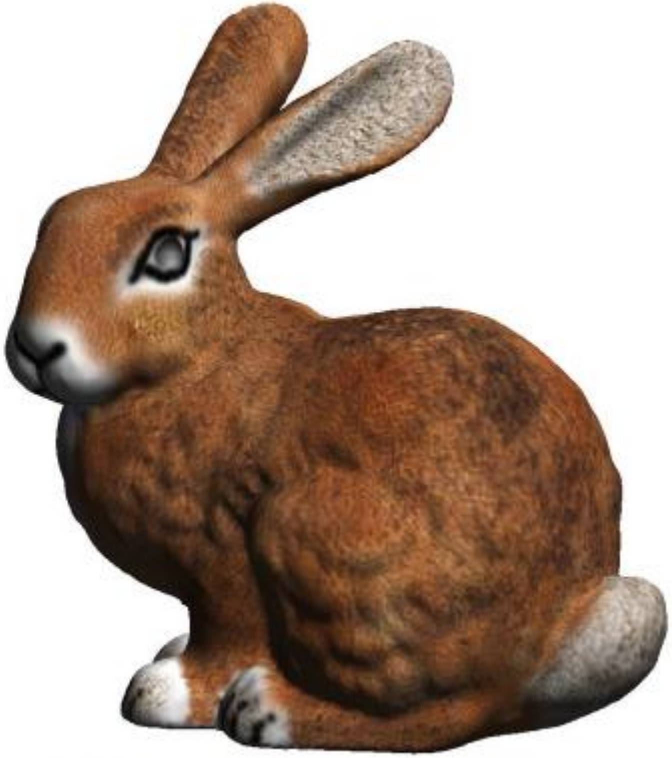


Image: Praun et al.

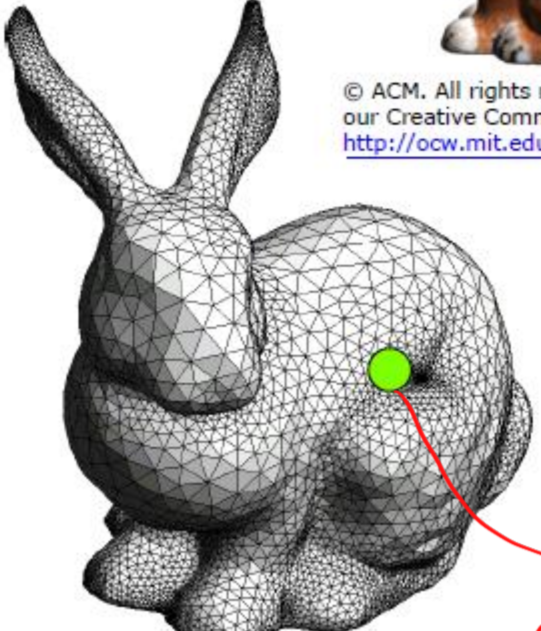
Wojciech Matusik, and Frédo Durand. *6.837 Computer Graphics.* Fall 2012. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC-SA.

2 UV Texture Mapping

Texture mapped model

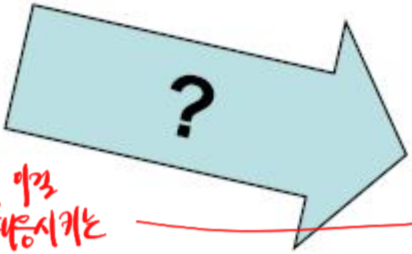
© ACM. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.

We need a function that associates each surface point with a 2D coordinate in the texture map



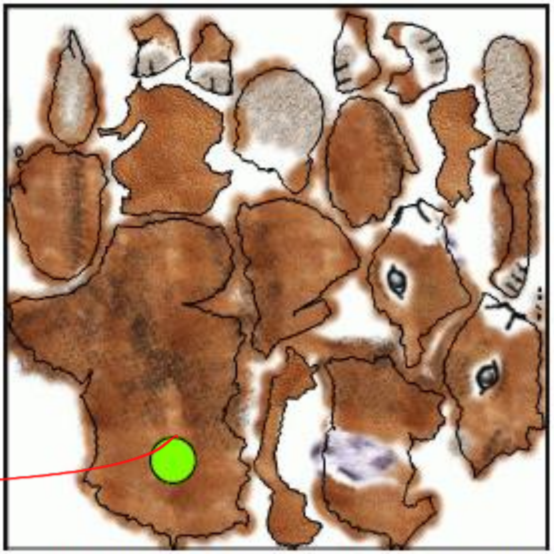
© Oscar Meruvia-Pastor, Daniel Rypl. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.

2D 이미지에
RGB 컬러 저장.



이걸
대응시키는
(vertex point – texture map)

Texture map (2D image)



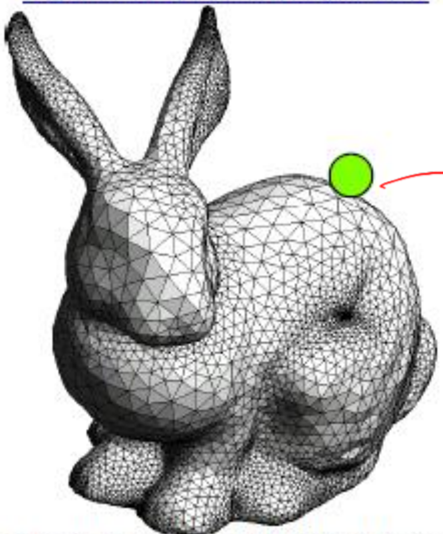
© ACM. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.

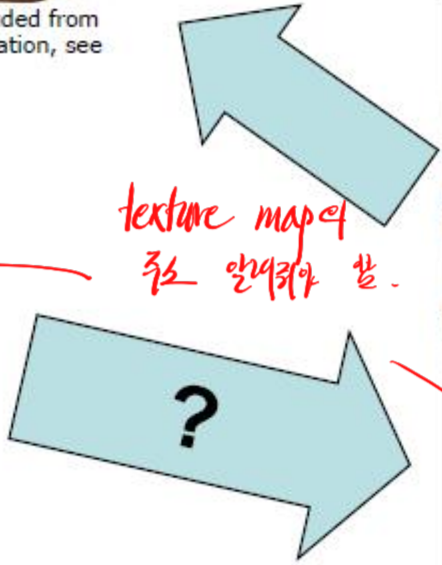
Wojciech Matusik, and Frédo Durand. *6.837 Computer Graphics.* Fall 2012. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC-SA.

2.1 UV Texture Mapping

Texture
mapped model

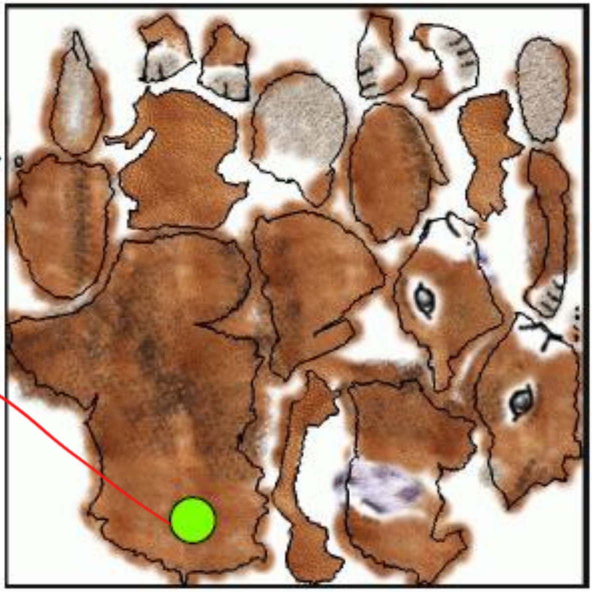
For each point rendered, look up color in texture map





texture map의
주소 알려줘야 함.

Texture map (2D image)



Wojciech Matusik, and Frédo Durand. *6.837 Computer Graphics.* Fall 2012. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu. License: Creative Commons BY-NC-SA.

좌표계의 값.

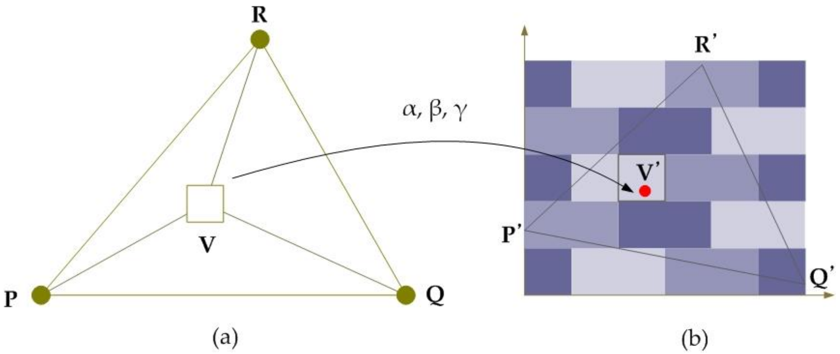
2.2 UV Coordinates

- 모든 Vertex P는 2D (u,v) texture coordinates 를 가짐
- UV는 texture map의 2D 위치임
- Barycentrics를 통해서 interpolation 값을 획득

$$V = \alpha P + \beta Q + \gamma R$$

$$V' = \alpha P' + \beta Q' + \gamma R'$$

텍스쳐 맵.



(a)

(b)

모든 vertex마다 uv값 할당.

2.3 Barycentric Coordinates

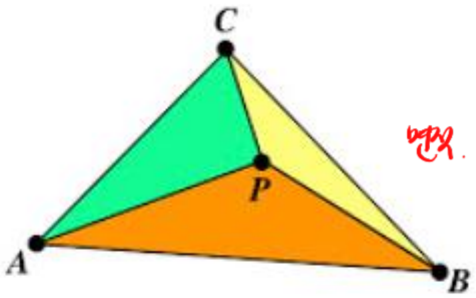
- 모든 Vertex P는 2D (u,v) texture coordinates 를 가짐

- UV는 texture map의 2D 위치임

- Barycentrics를 통해서 interpolation 값을 획득

면적에 비례해서 점 A~C가 영향을 줌.

$$P = w_A \times A + w_B \times B + w_C \times C$$



면적.

$$w_A = \frac{\Delta PBC}{\Delta ABC} =$$

$$w_B = \frac{\Delta PCA}{\Delta ABC} =$$

$$w_C = \frac{\Delta PAB}{\Delta ABC} =$$

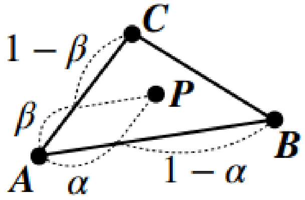
inside condition

$0 \le w_A, w_B, w_C \le 1 \qquad w_A + w_B + w_C = 1$

http://wanochoi.com/lecture/BarycentricCoordinates.pdf

녹강에선 안씀.. 증명같은데

2.3 Barycentric Coordinates



$$AB = B - A$$
$$AC = C - A$$
$$Q = P - A$$

$$\alpha AB + \beta AC + A = P$$

$$\Leftrightarrow \alpha AB + \beta AC = P - A$$

$$\Leftrightarrow \alpha AB + \beta AC = Q$$

$$\Leftrightarrow \begin{matrix} \alpha AB.x + \beta AC.x = Q.x \\ \alpha AB.y + \beta AC.y = Q.y \end{matrix}$$

$$\Leftrightarrow \begin{bmatrix} AB.x & AC.x \\ AB.y & AC.y \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} Q.x \\ Q.y \end{bmatrix}$$

$$\Leftrightarrow \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} AB.x & AC.x \\ AB.y & AC.y \end{bmatrix}^{-1} \begin{bmatrix} Q.x \\ Q.y \end{bmatrix}$$

$$\Leftrightarrow \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \frac{1}{det} \begin{bmatrix} AC.y & -AC.x \\ -AB.y & AB.x \end{bmatrix} \begin{bmatrix} Q.x \\ Q.y \end{bmatrix}$$

$$det = AB.x \times AC.y - AC.x \times AB.y$$

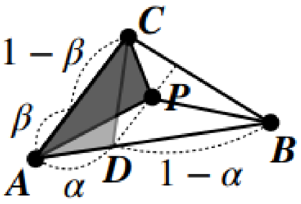
$$\Leftrightarrow \begin{matrix} \alpha = (AC.y \times Q.x - AC.x \times Q.y)/det \\ \beta = (AB.x \times Q.y - AB.y \times Q.x)/det \end{matrix}$$

It is inside in the triangle iff:

$$\beta \geq 0 \ \ \& \ \ \beta \leq 1 \ \ \& \alpha \geq 0 \ \ \& \ \ \alpha + \beta \leq 1$$

Its barycentric coordinates:

$$w_A = 1 - \alpha - \beta$$
$$w_B = \alpha$$
$$w_C = \beta$$

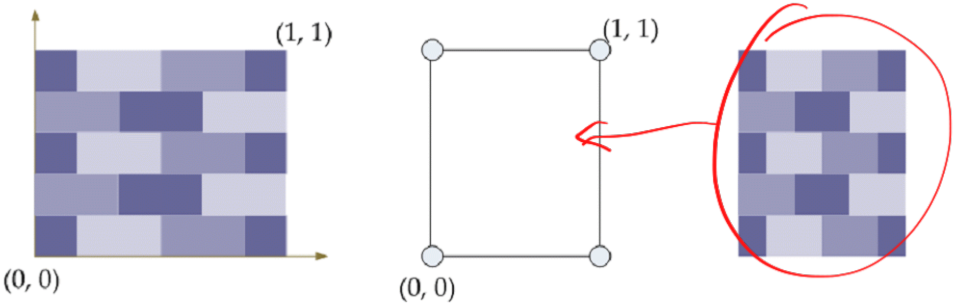


$$w_B = \frac{\Delta APC}{\Delta ABC} = \frac{\Delta ADC}{\Delta ABC} = \alpha$$

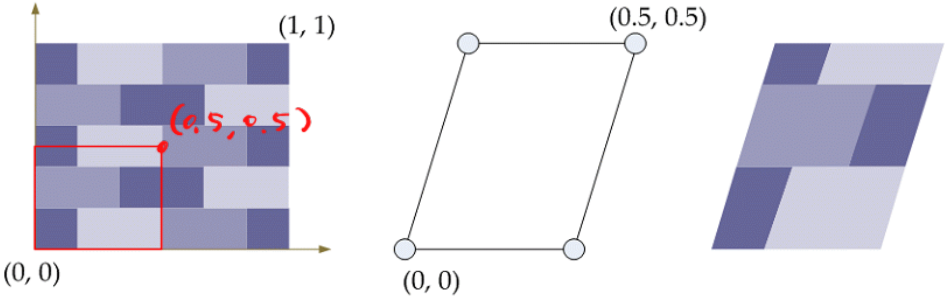
http://wanochoi.com/lecture/BarycentricCoordinates.pdf

2.3 평면 다각형

- 수작업 좌표 명시



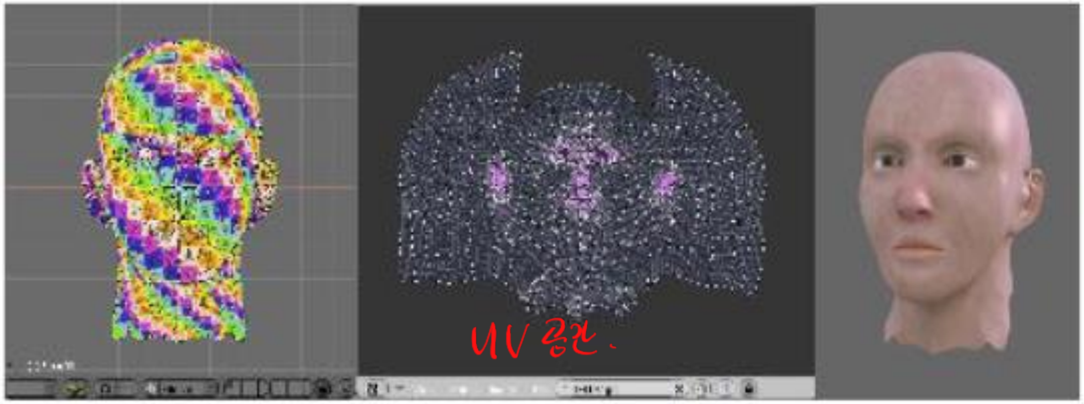
(a) (b) (c)

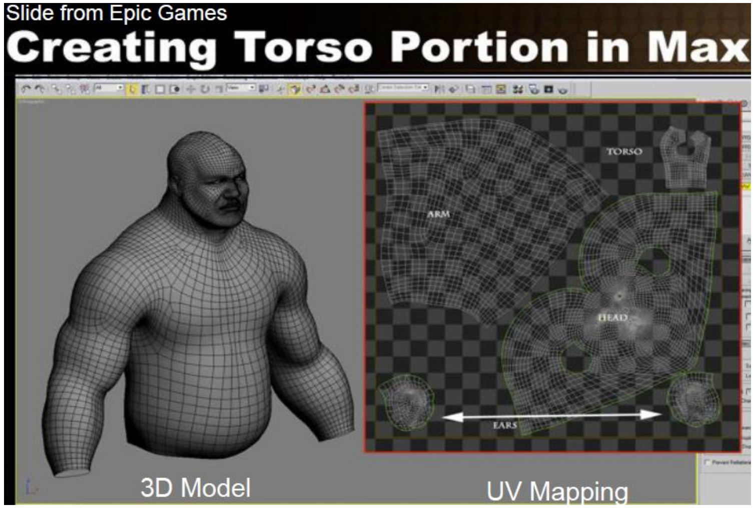


(a) (b) (c)

2.4 곡면 UV Texture Mapping

- Goal : “flatten” 3D object onto 2D UV coordinates
- For each vertex, find coordinates U,V such that distortion is minimized




Slide from Epic Games

Creating Torso Portion in Max

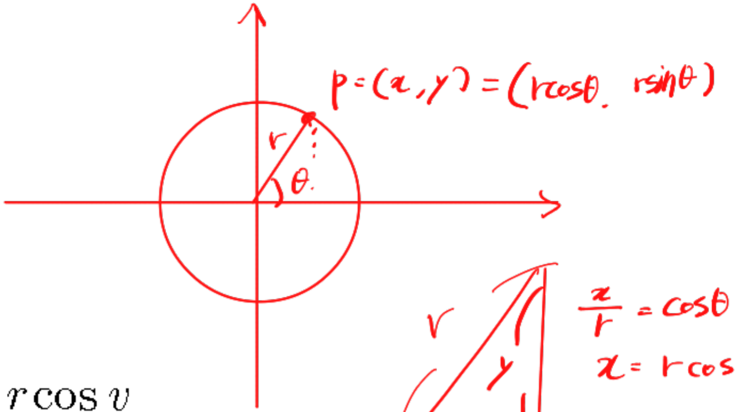
3. Parametric Texturing (파라미터 곡면)

- 구와 같은 non-triangular geometry 에 적용 vertex가 없기 때문에 UV를 정의 할 수 없음

수식으로 표현.

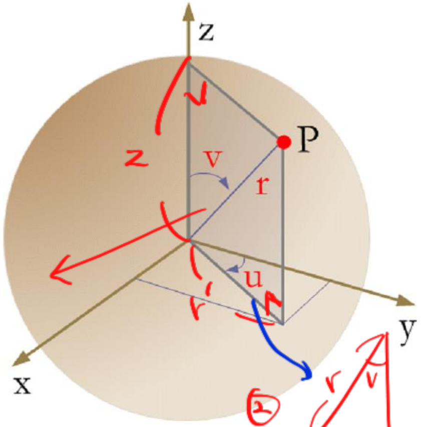
3.1 원구

- 표면상의 점을 경도, 위도로 표현 가능



① $\cos v = \frac{z}{r}$

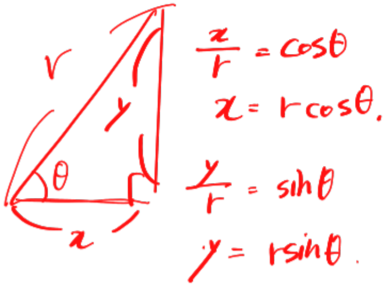
$z = r\cos.$



$\frac{r'}{r} = \sin v.$

$r' = r\sin v.$

$$z = r\cos v$$
$$y = r\sin v\ \cos u$$
$$x = r\sin v\sin u$$

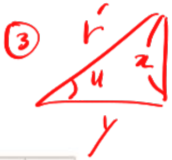


$\frac{x}{r} = \cos\theta$

$x = r\cos\theta,$

$\frac{y}{r} = \sin\theta$

$y = r\sin\theta.$

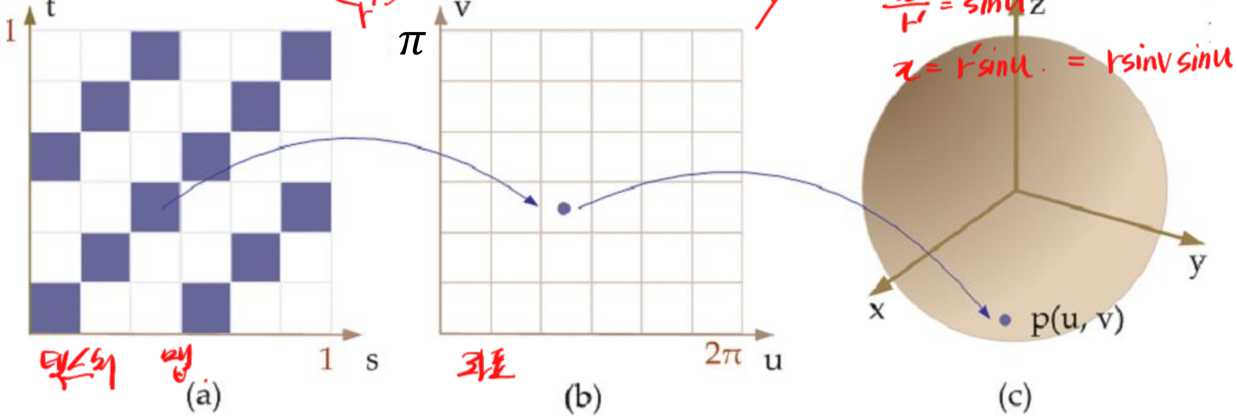


$\frac{y}{r'} = \cos u$

$y = r'\cos u. = r\sin v\cos u$

$\frac{z}{r'} = \sin u$

$z = r'\sin u. = r\sin v\sin u$



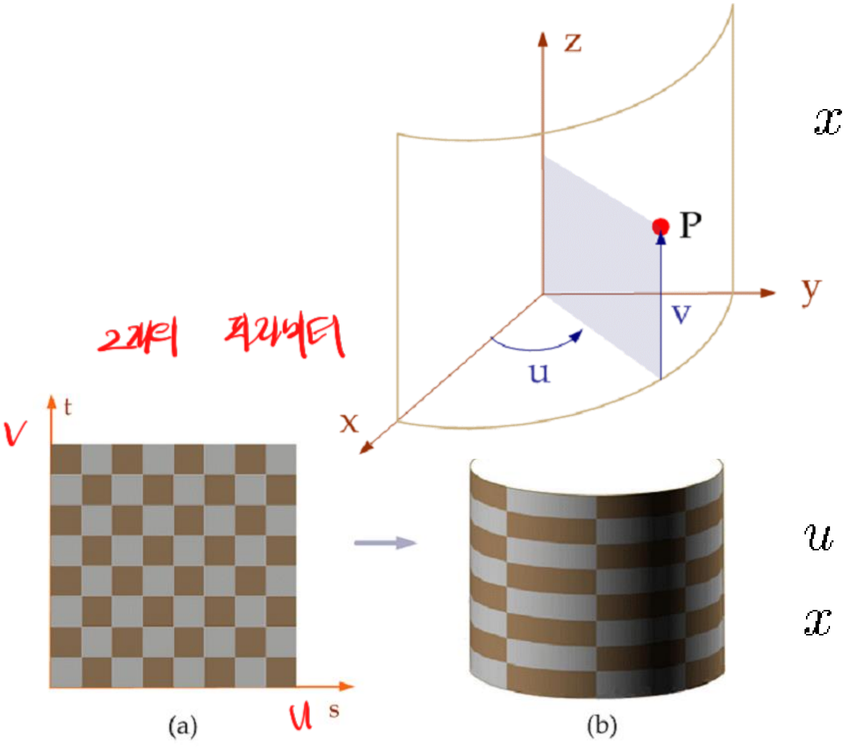
$0 \le u \le 360$

$0 \le v \le 180$ ⟶ $u = 2\pi s, \quad v = \cancel{2}\pi t$

$$z = r\cos\ \pi t$$
$$y = r\sin\ \pi t\ \cos 2\pi s$$
$$x = r\sin\ \pi t\ \sin 2\pi s$$

3.2 다각형 곡면

- 2 단계 매핑(2-Stage Mapping)
- 곡면을 매개변수로 표시할 수 없을 때
- S 매핑(S Mapping)에서는 텍스쳐를 원기둥, 육면체, 원구 등 중개면 (仲介, Intermediate Surface)에 입힘.
- S 매핑의 예: 원기둥 중개면



$$x = r\cos u,\ y = r\sin u,\ z = v$$

$$u = 2\pi s,\quad v = t$$

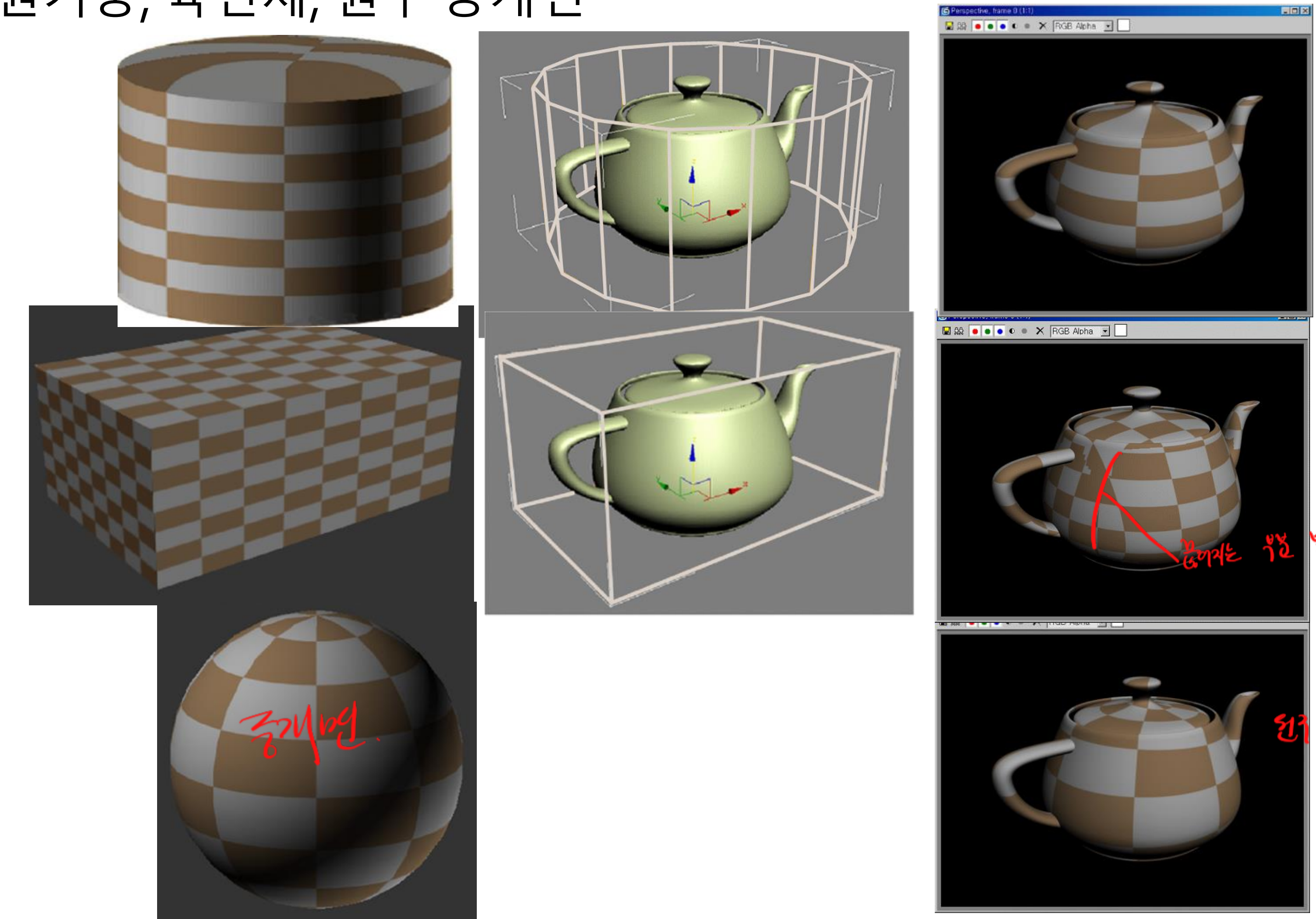
$$x = r\cos 2\pi s,\ y = r\sin 2\pi s,\ z = t$$

3.2 다각형 곡면

- O 매핑

중개면 종류에 따라 텍스쳐 달라짐.

- 물체를 중개면 내부에 넣고 물체면에 텍스쳐를 입힘.
- 원기둥, 육면체, 원구 중개면



중개면.

끊어지는 부분 발생.

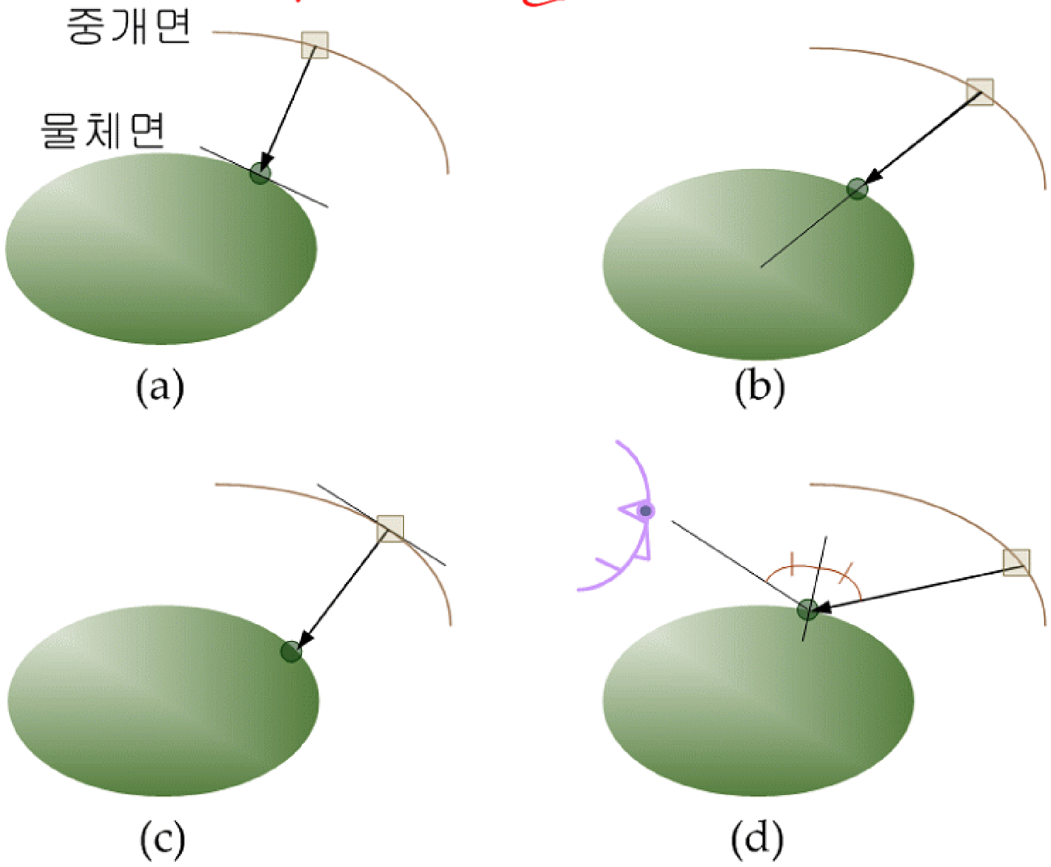
원구 : 상대적으로 자연스러움?

3.2 다각형 곡면

- O 매핑의 종류

- 물체면 법선벡터, 물체 중심, 중개면 법선벡터, 시점 반사벡터

a b c d



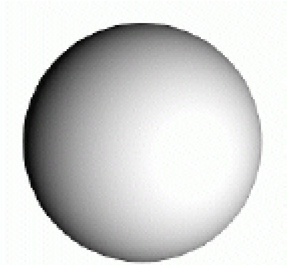
(a) (b)

(c) (d)

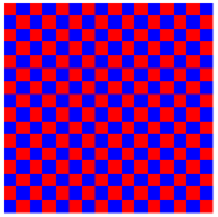
영강 여기까지.

4. Texture mapping & Illumination

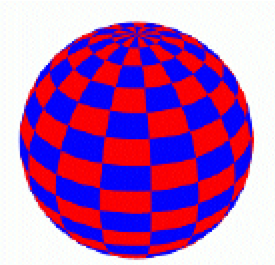
- Texture mapping can be used to alter some or all of the constants in the illumination equation



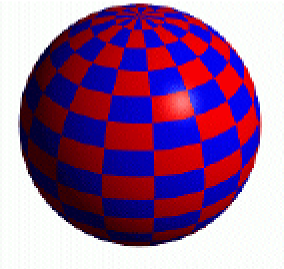
Constant Diffuse Color



Diffuse Texture Color



Texture used as Label

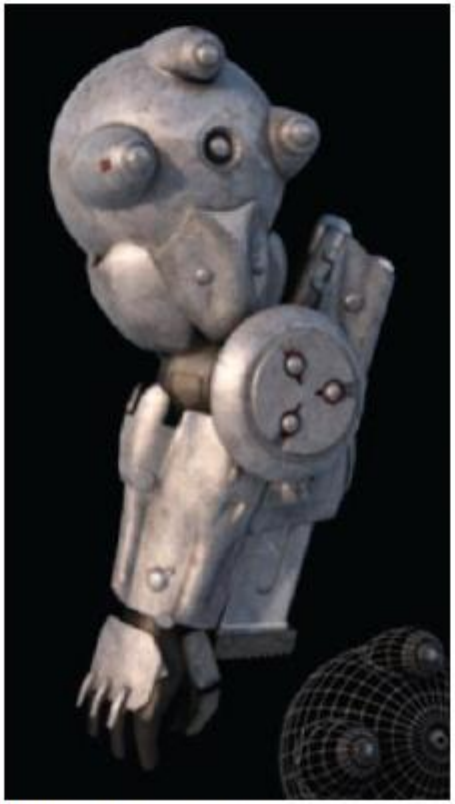


Texture used as Diffuse Color

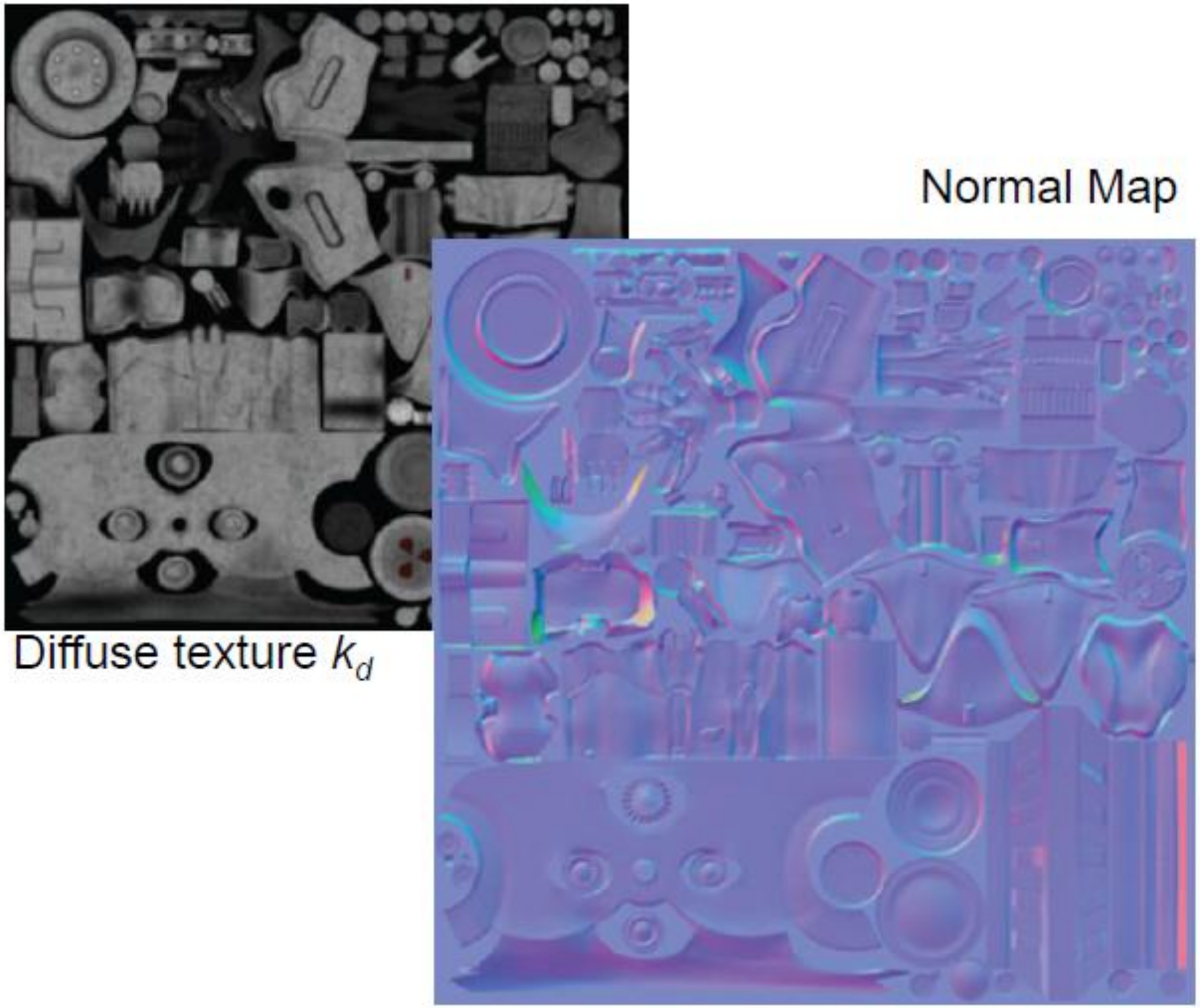
5. Normal Mapping

- The normal vector is really important in conveying the small-scale surface detail
- For each shaded point, normal is given by a 2D image normalMap that stores the 3D normal

5. Normal Mapping



Final render



Diffuse texture k_d

Normal Map

5. Normal Mapping

- Model a detailed mesh
- •Generate a UV parameterization for the mesh
- –A UV mapping such that each 3D point has unique image coordinates in the 2D texture map
- –This is a difficult problem, but tools are available
- •E.g., the DirectX SDK has functionality to do this
- •Simplify the mesh (again, see DirectX SDK)
- •Overlay simplified and original model
- •For each point P on the simplified mesh, find closest point P’ on original model (ray casting)
- •Store the normal at P’ in the normal map. Done!

5. Normal Mapping

- You can store an object-space normal
- –Convenient if you have a unique parameterization
- •....but if you want to use a tiling
- normal map, this will not work
- –Must account for the curvature of the object!
- –Think of mapping this diffuse+normal map combination on a cylindrical tower
- •Solution: Tangent space normal map
- –Encode a “difference” from the geometric normal in a local coord. system