Computer Graphics

12 – More Lighting, Texture

Yoonsang Lee
Spring 2021

기말고사 재공지

- 일시: 6월 **7일 (월)** 오전 9시~11시
- 전면 온라인 시험 (LMS의 시험 기능 이용)
- **오픈북 시험 아님!**

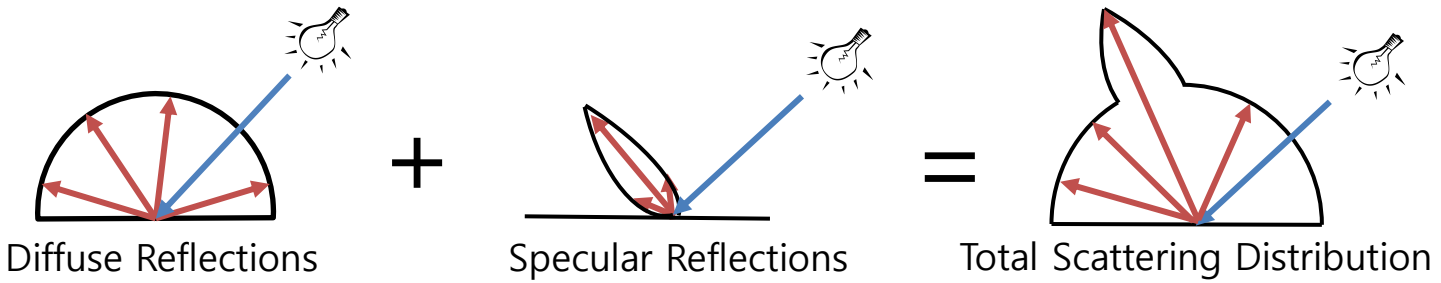
Topics Covered

- More Lighting
  - BRDF
  - Local vs. Global Illumination
- Texture Mapping
  - Texture Coordinates & UV Mapping
  - Various Uses of Texture Maps

More Lighting

Recall: Reflection of General Materials

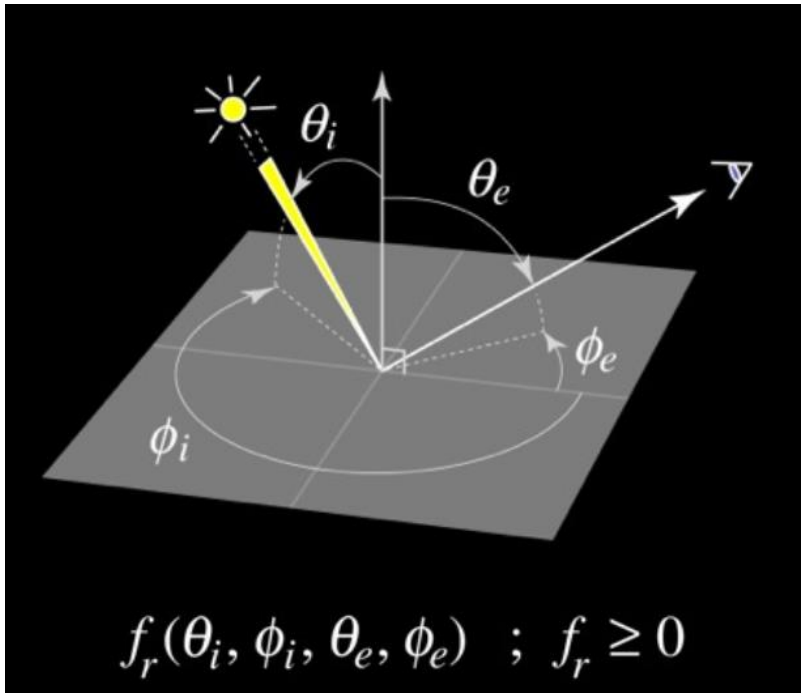
- Many materials' surface have both diffuse reflection and specular reflection.



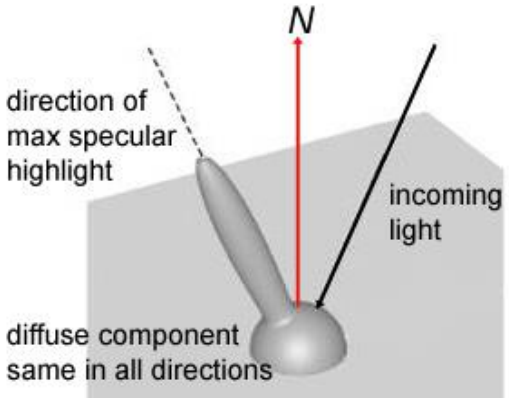
- We can represent reflectance properties of a surface as a **distribution function.**
- → **BRDF**

Bidirectional Reflectance Distribution Function (BRDF)

- Defines how light is reflected at an opaque surface.
  - θ_i, φ_i : incoming light direction
  - θ_e, φ_e : outgoing light direction
  - f_r returns the ratio of reflected radiance exiting along outgoing direction

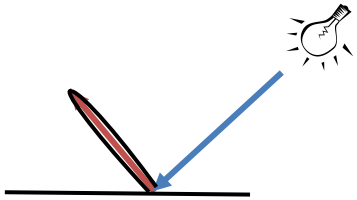


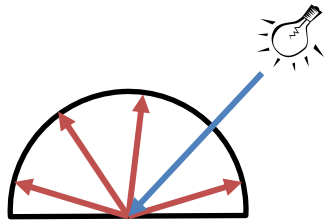
An example BRDF

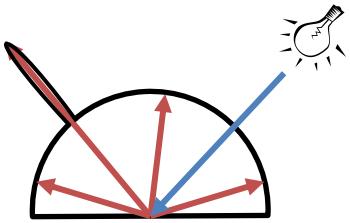


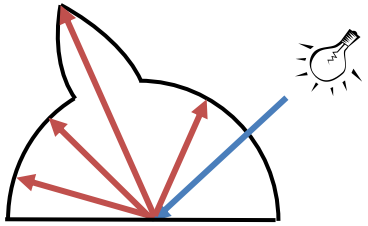
Examples of BRDF

(theoretical approximation, not from measurement)









Measuring BRDF

- BRDFs of specific materials can be measured using devices like this:
  - Basic idea: rotating light source & rotating sensor

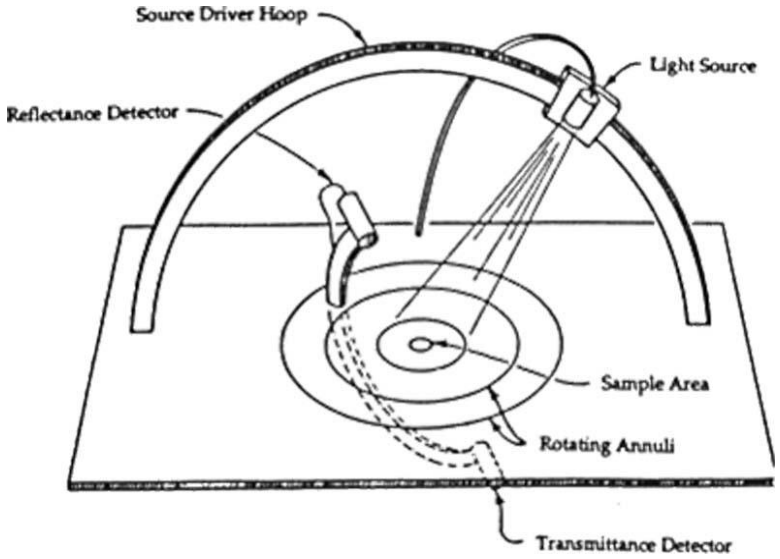


Image credit: Chuck Moidel

How to use BRDF Data?

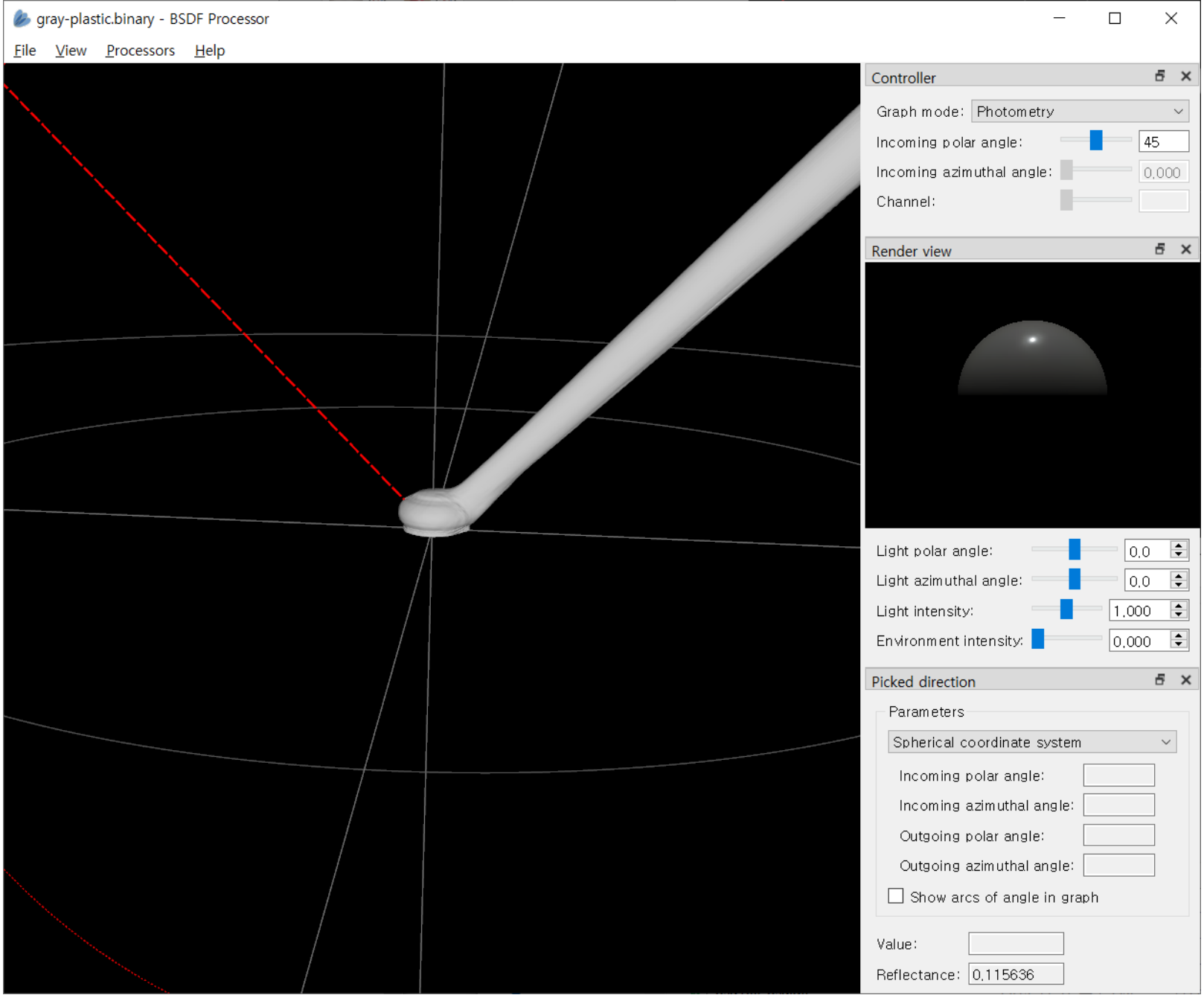
One can make direct use of acquired BRDFs in a renderer

KAIST

[Practice] MERL BRDF Database

- MERL BRDF Database – A database containing measured BRDFs of 100 different materials
  - "A Data-Driven Reflectance Model", Wojciech Matusik, Hanspeter Pfister, Matt Brand and Leonard McMillan, SIGGRAPH 2003
  - https://www.merl.com/brdf/, but the BRDF download link is broken!
  - Instead, download *12 - MERL BRDF samples.zip* from the course homepage.

- Viewer
  - BSDF Processor: https://github.com/KimuraRyo/BSDFProcessor



BRDF of gray plastic

Quiz #1

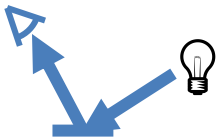
- Go to https://www.slido.com/
- Join #**cg-ys**
- Click “Polls”

- Submit your answer in the following format:
  - **Student ID: Your answer**
  - **e.g. 2017123456: 4)**

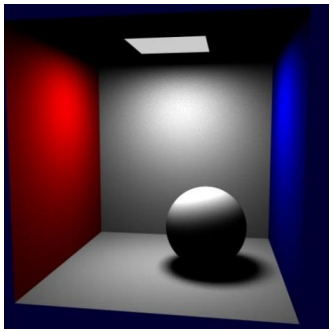
- Note that you must submit all quiz answers in the above format to be checked for “attendance”.

Local vs. Global Illumination

- **Local (or direct, or non-global) illumination**
  - Models the direct illumination of object surfaces **by light sources**
  - Can be rendered fast, but less realistic (unrealistic)
  - e.g. Phong illumination model

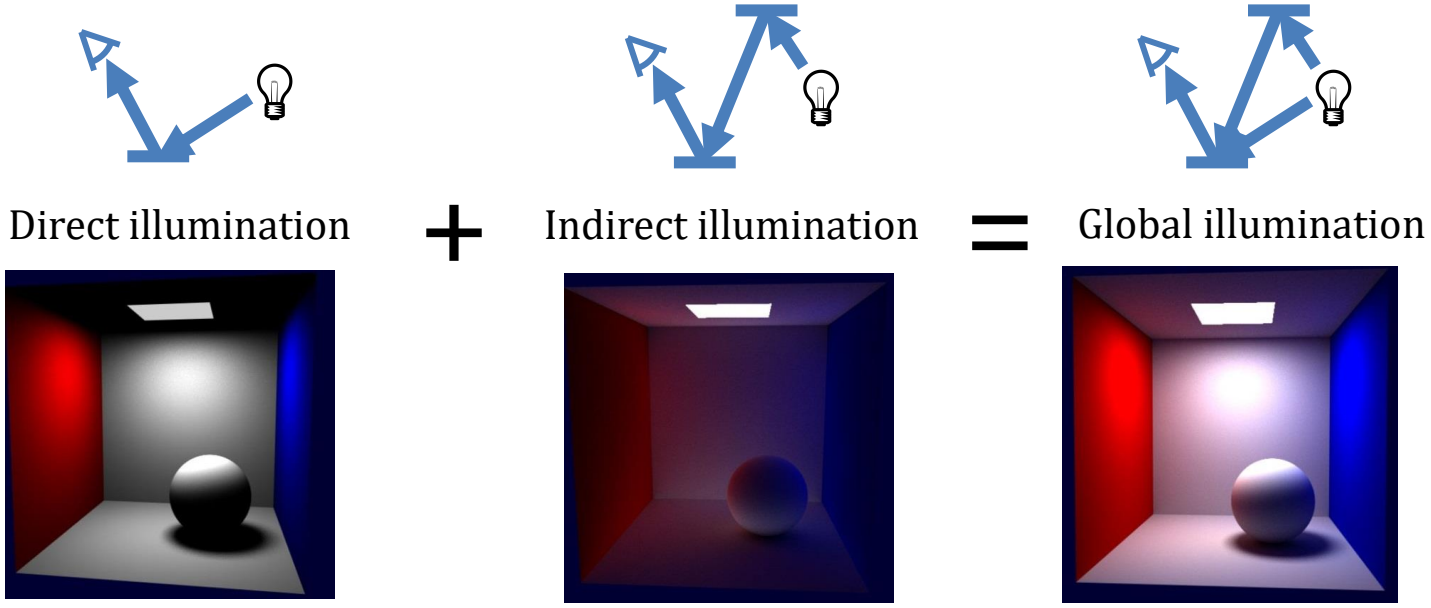


Direct illumination



Local vs. Global Illumination

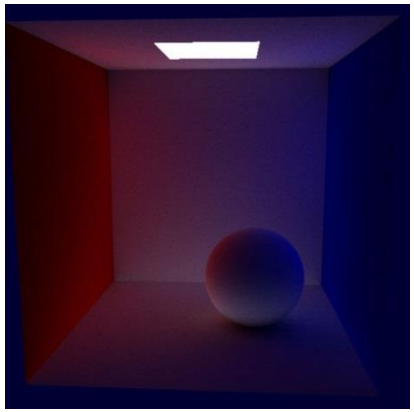
- **Global illumination**
  - Direct illumination + Indirect illumination (All inter-object reflections)
  - Slow, but much more realistic
  - e.g. Ray tracing, Path tracing, Radiosity



Phong Illumination Model & Local, Global Illumination

- Phong illumination model is basically a local illumination model.

- Indirect illumination is severely approximated by the ambient component.

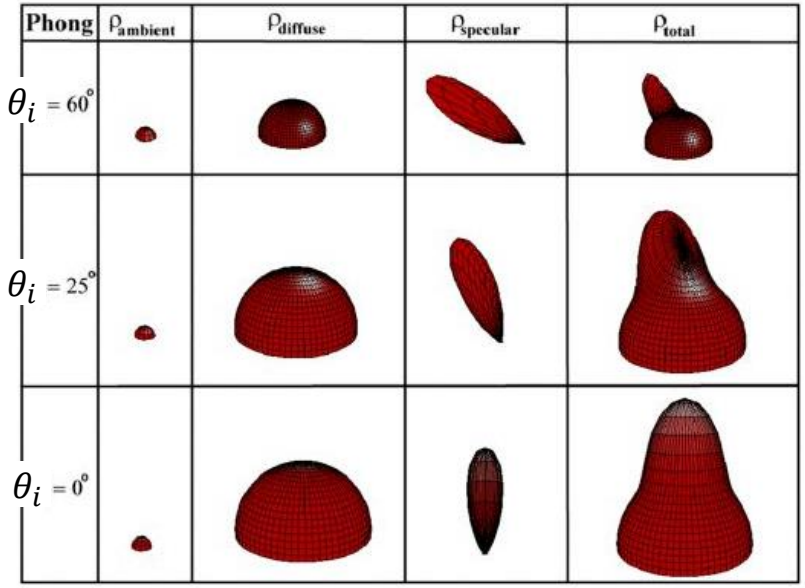


Ambient

Phong Illumination Model & BRDF

- Local illumination model
  - Models the direct illumination of object surfaces by light sources.
  - No other than modeling BRDFs!

- Phong's model models BRDFs with
  - a hemisphere (by diffuse component)
  - and a lobe (by specular component using $\cos^n(\alpha)$)



Quiz #2

- Go to https://www.slido.com/
- Join #**cg-ys**
- Click “Polls”

- Submit your answer in the following format:
  - **Student ID: Your answer**
  - **e.g. 2017123456: 4)**

- Note that you must submit all quiz answers in the above format to be checked for “attendance”.

Texture Mapping

How to add surface details?

- Option 1:
- Model the surface with more polygons
  - Slows down rendering speed
  - Hard to model fine features

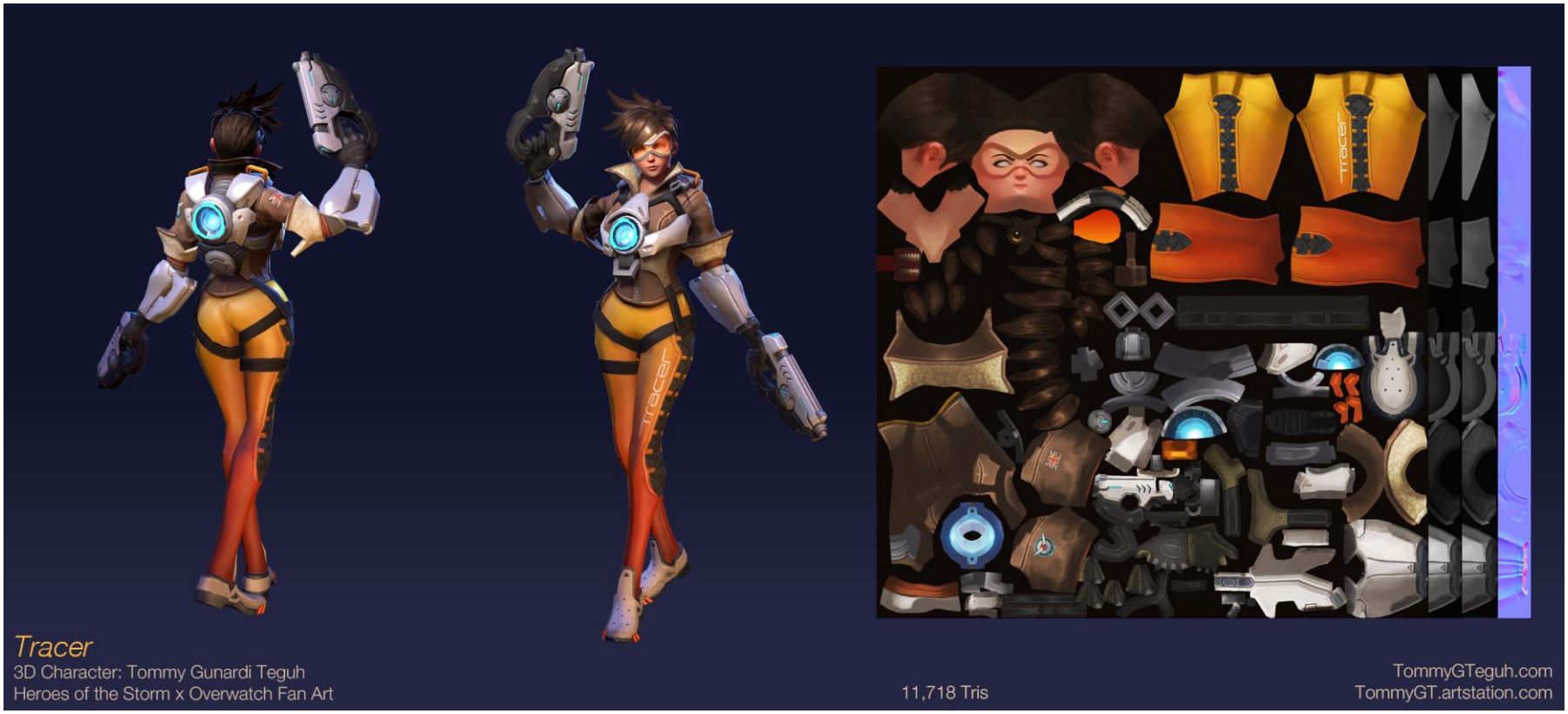
(3.3 million polygons)

How to add surface details?

- Option 2:
- Map a **texture** to the surface
  - More details with less polygons
  - Much faster rendering speed! (because image complexity does not affect complexity of processing)

Examples of Texture Mapping

Examples of Texture Mapping



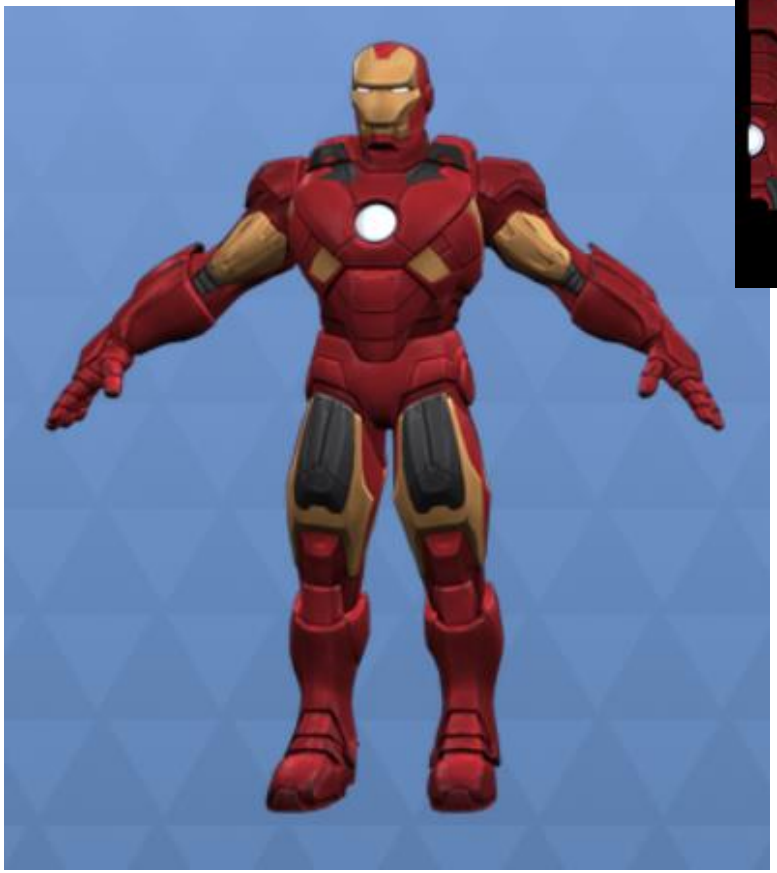
Texture Mapping

- A technique of defining surface properties as a function of position on the surface
  - usually with texture images (or texture maps)

- What can you put in a texture map?
  - **Diffuse color**, **specular color**
  - Specular exponents, transparency or reflectivity coefficients
  - **Surface normal**
  - Projected reflections or shadows

  (bold text indicates most commonly used properties)

Examples of Diffuse, Specular, Normal Map



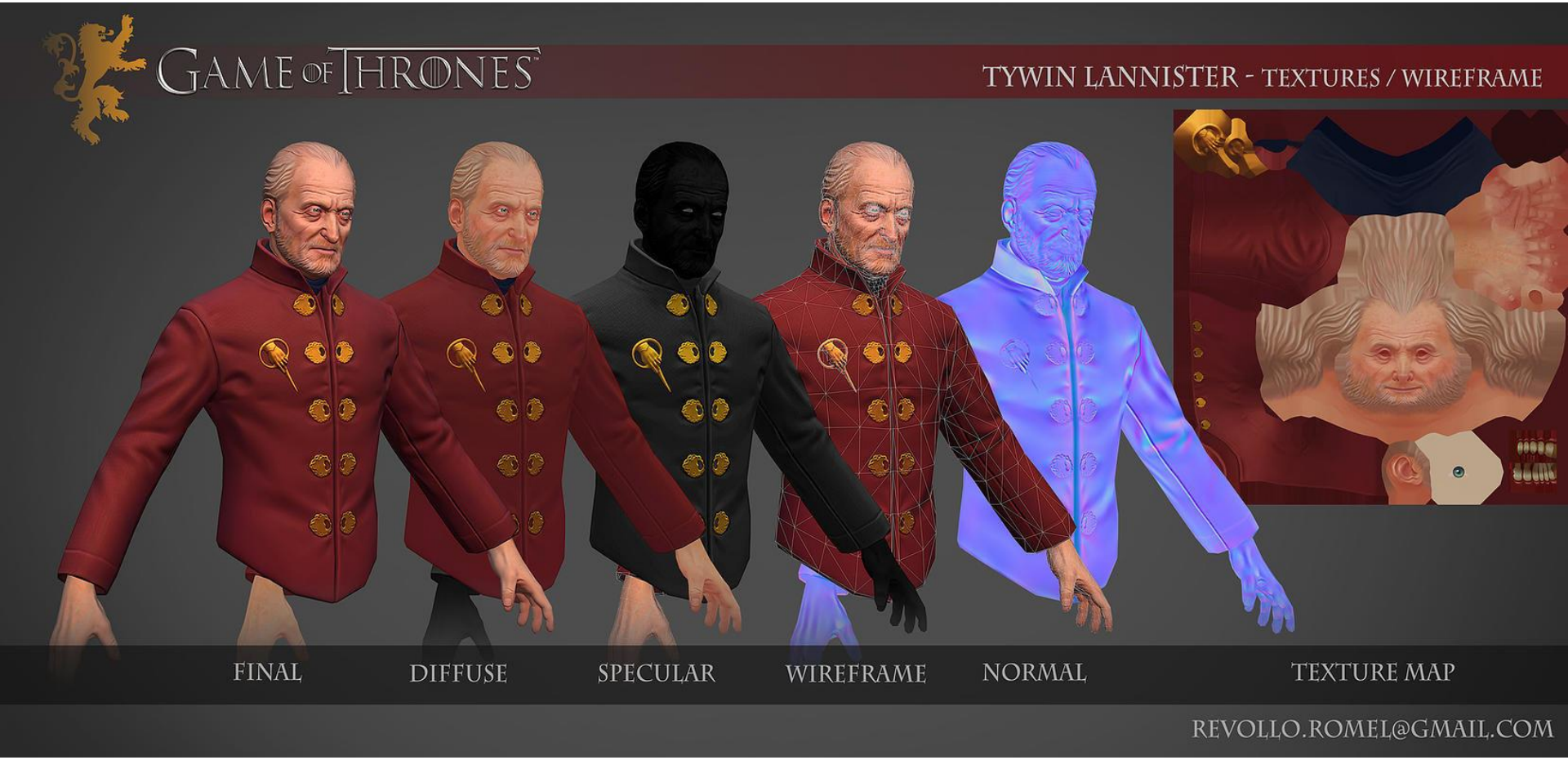
diffuse map

specular map

normal map

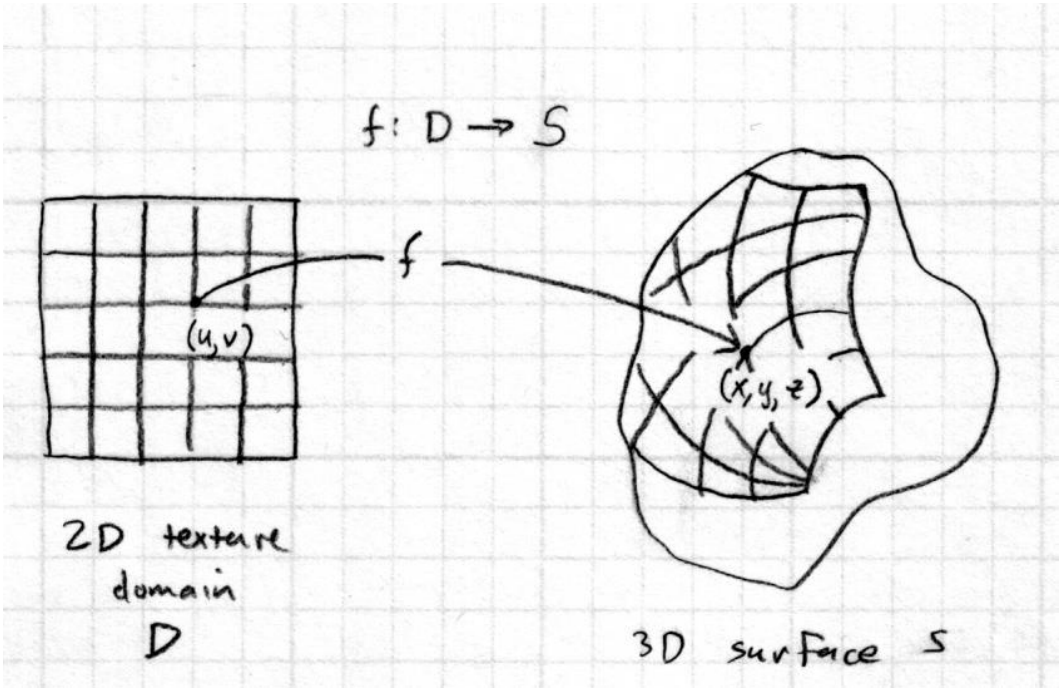
https://www.models-resource.com/mobile/marvelscontestofchampions/model/15341/

Examples of Diffuse, Specular, Normal Map



Mapping textures to surfaces

- "Putting the image on the surface"
  - this means we need a function f that tells where each point on the image goes



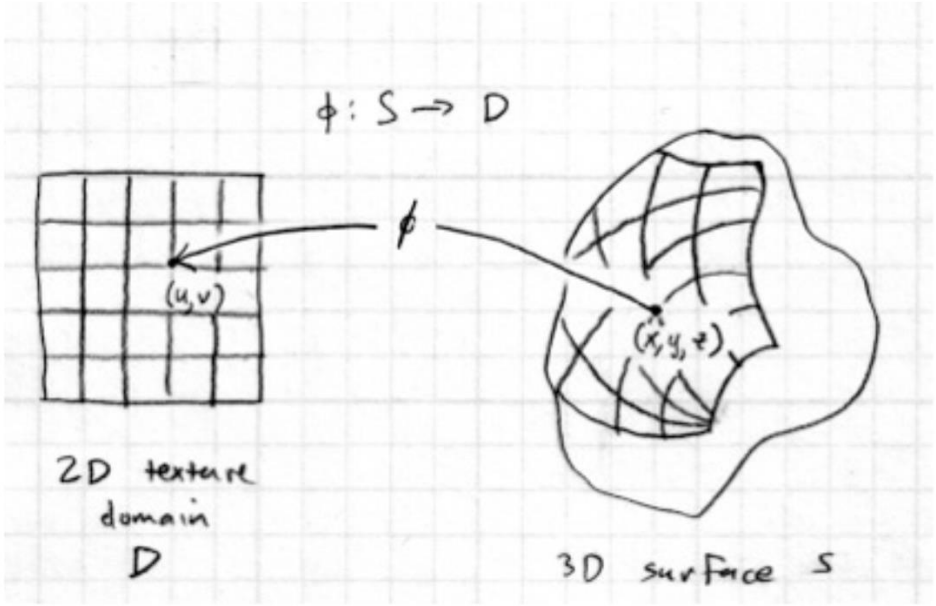
© 2008 Steve Marschner

Texture Coordinate Function

- Usually be thought as an inverse function: “Putting the surface on the image”

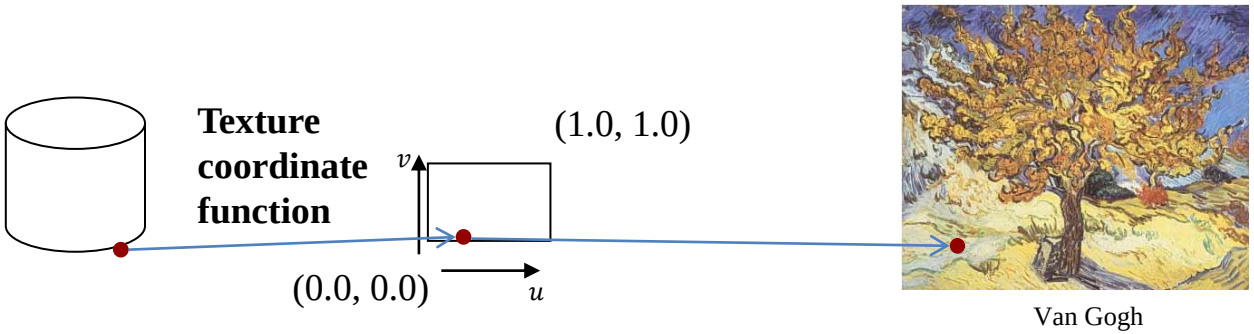
: **Texture coordinate function**

or **UV mapping**



Texture Mapping Technique (1/8)

- Done in two steps:
  - Map a point on object to a point on unit square (a proxy for the actual texture map)
  - Map unit square point to point on texture

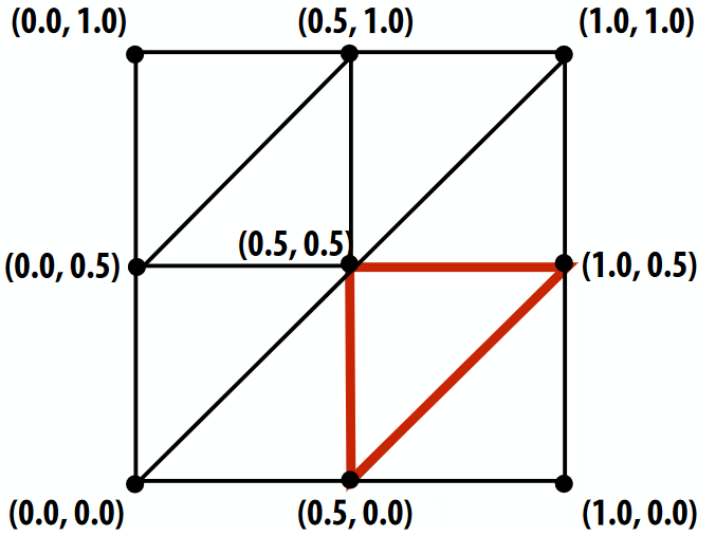


Van Gogh

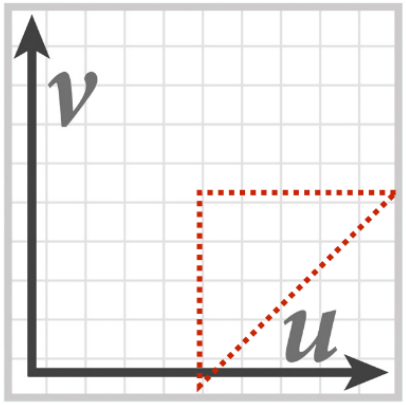
- The second mapping is easy; it's just scaling and automatically performed by system
- **What we have to focus on is the first step; texture coordinate function.**
- **Texture coordinate function determines which texture coordinates (u,v) each vertex has.**

Texture coordinates

"Texture coordinates" define a mapping from surface coordinates (points on triangle) to points in texture domain.



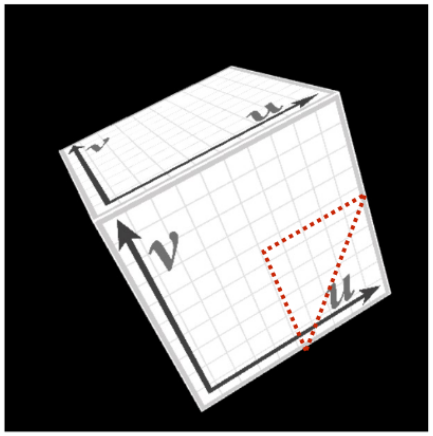
Eight triangles (one face of cube) with surface parameterization provided as per-vertex texture coordinates.



`myTex(u,v)` is a function defined on the $[0,1]^2$ domain:

myTex : $[0,1]^2 \rightarrow$ float3
(represented by 2048x2048 image)

Location of highlighted triangle in texture space shown in red.



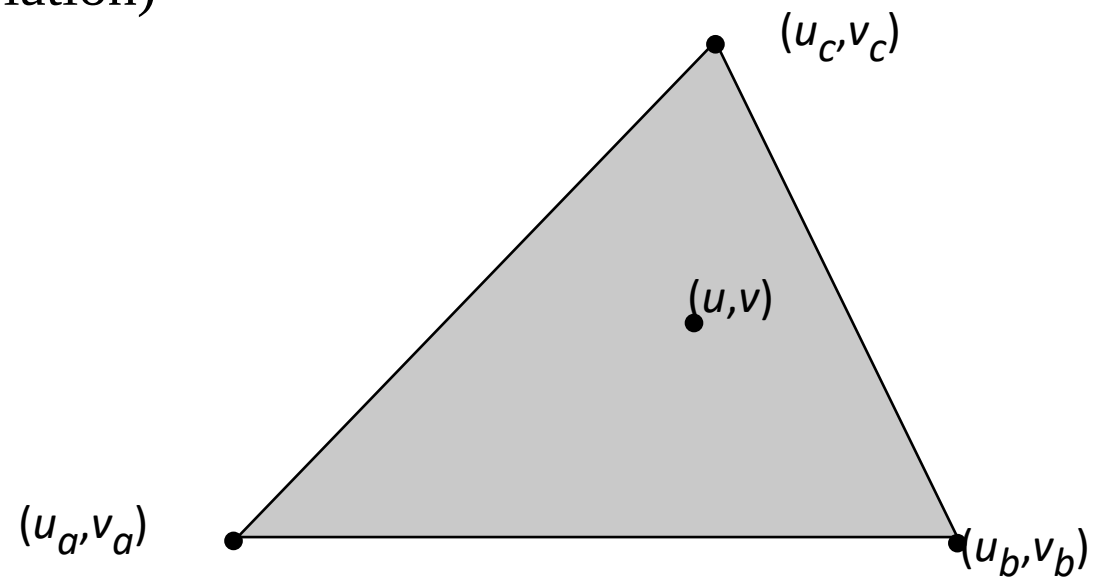
Final rendered result (entire cube shown).

Location of triangle after projection onto screen shown in red.

Texture Coordinates for a Triangle

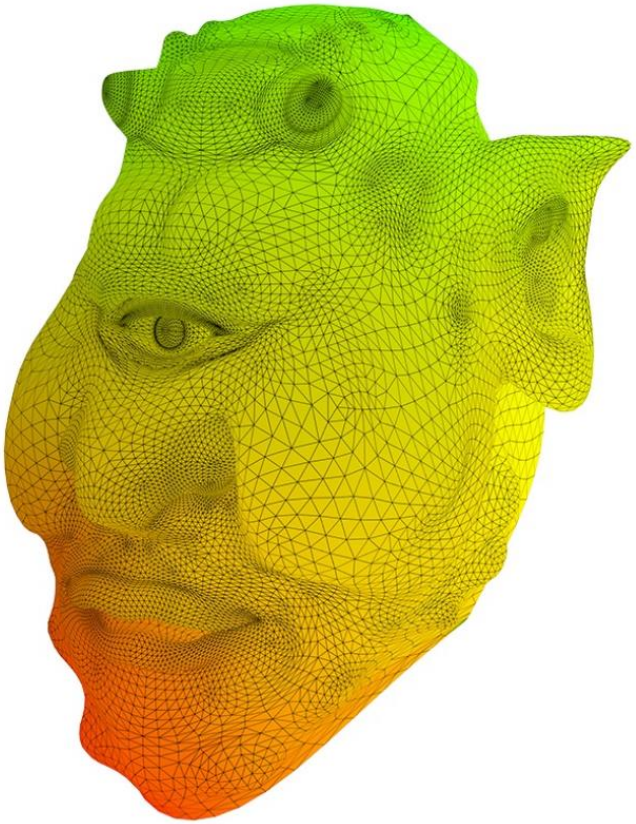
- You have to specify (u,v) for each vertex
  - Recall that **texture coordinates** are one of **vertex attributes** in OpenGL
  - Texture coordinate : specified by glTexCoord*()

- (u,v) for interior points are interpolated (by barycentric interpolation)

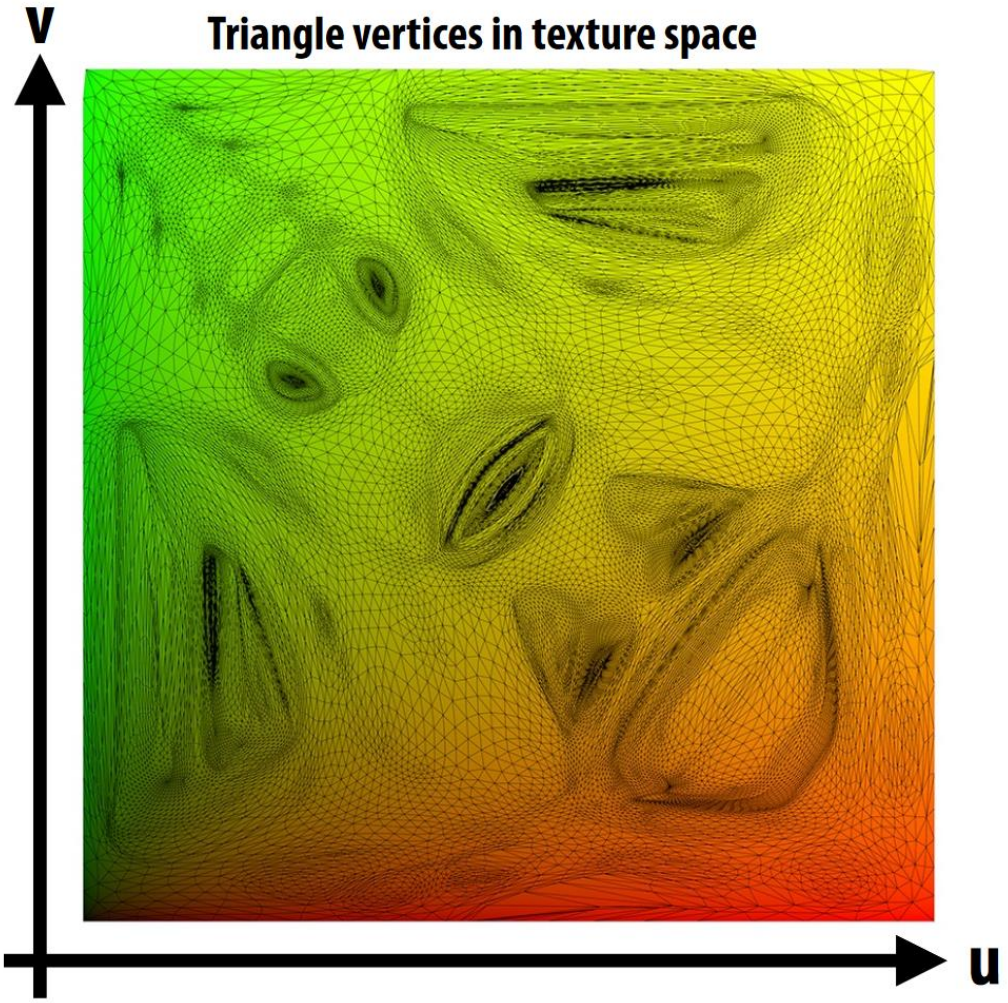


More complex mapping

Visualization of texture coordinates



Triangle vertices in texture space



Each vertex has a coordinate (u,v) in texture space.

(Actually coming up with these coordinates is another story!)

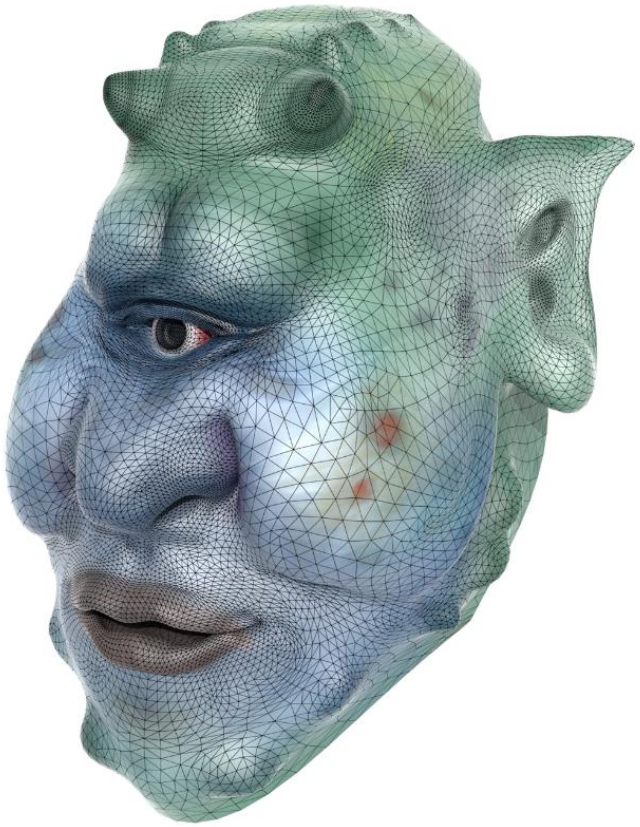
Simple texture mapping operation

a pixel of an object on the screen

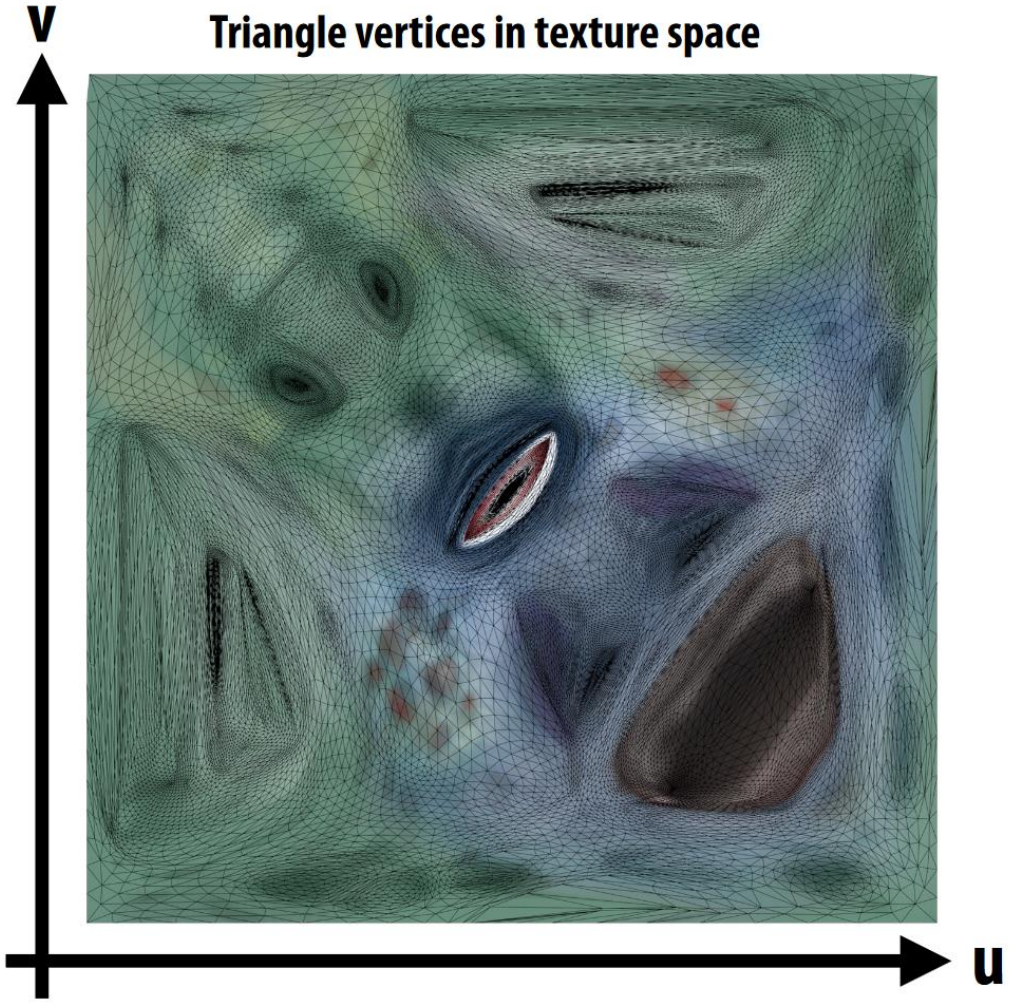
```
for each covered screen sample (x,y):
    (u,v) = evaluate texcoord value at (x,y)
    float3 texcolor = texture.sample(u,v);
    set sample's color to texcolor;
```

Texture mapping adds detail

Rendered result



Triangle vertices in texture space

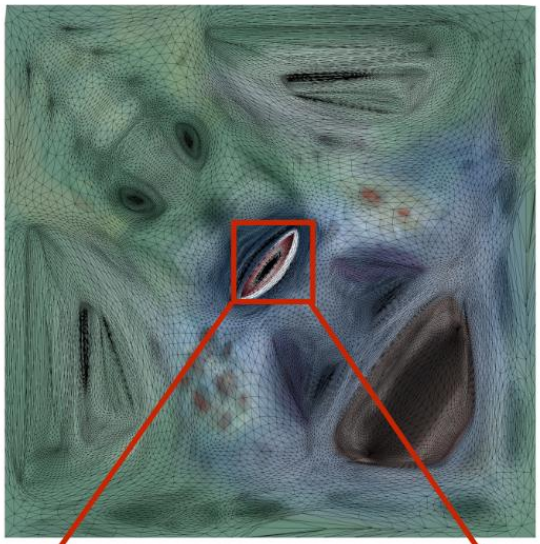


Texture mapping adds detail

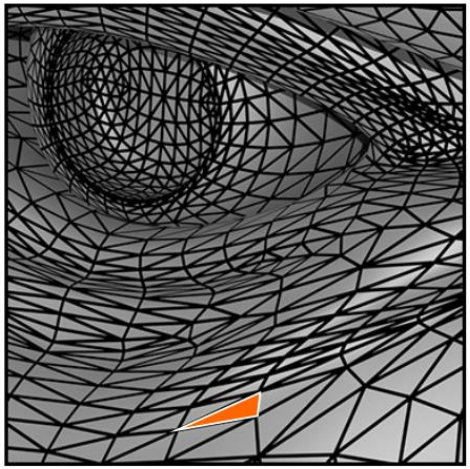
rendering without texture

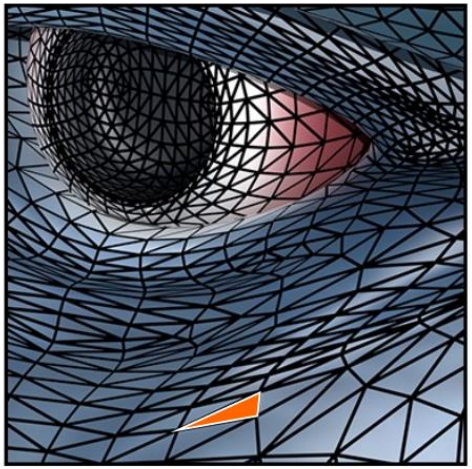
rendering with texture

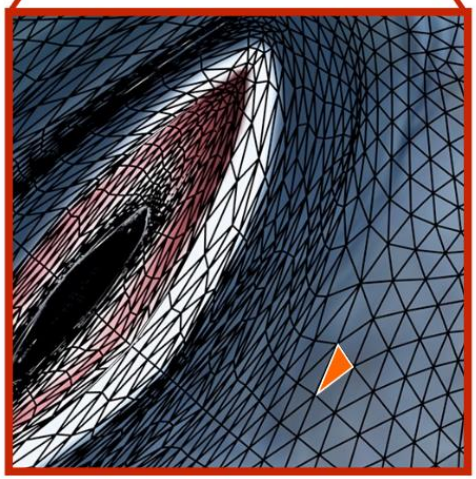
texture image



zoom



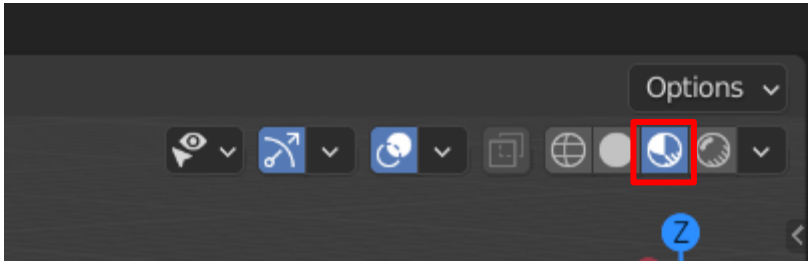




Each triangle "copies" a piece of the image back to the surface.

[Practice] UV Mapping in Blender

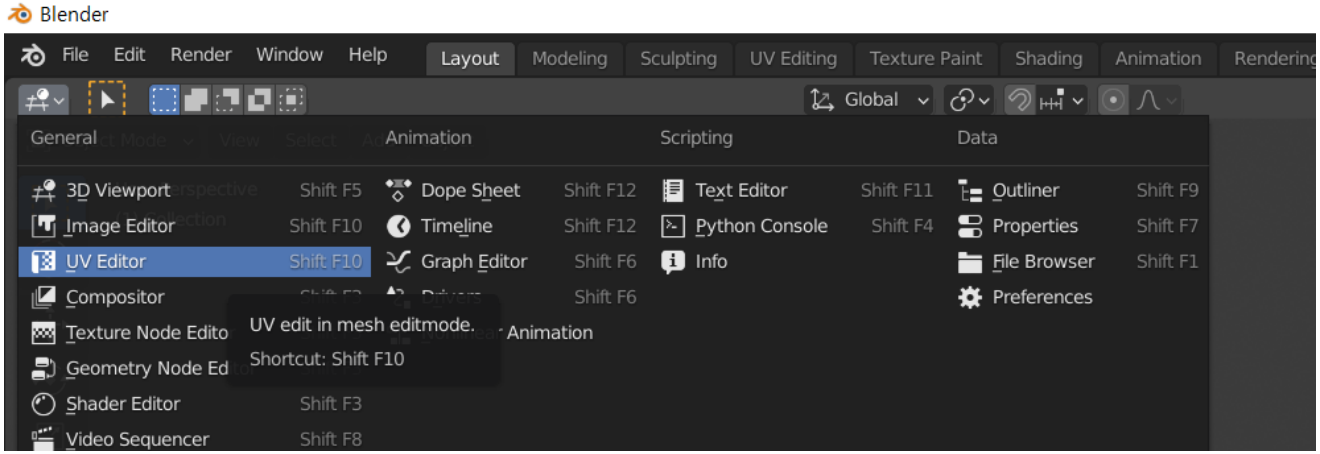
- Launch Blender
  - Download it from https://www.blender.org/download/
- Delete the default cube using ‘del’ key
- File – Import – obj – open *car.obj*
  - Download *12-car.zip* from our lecture home and extract it
- Change viewport shading type to 'Material Preview' mode.



[Practice] UV Mapping in Blender

- Drag the upper right corner of the view to left to split it

- Change the left view type to UV Editor



[Practice] UV Mapping in Blender

- Click the car model on the right view and press tab to change to the Edit Mode

- Now you'll be able to see the UV map of the car model on the left view
  - If you cannot see vertices in UV Editor (left view), click 3D View (right view) and press 'a' to select all vertices in Edit mode.

[Practice] UV Mapping in Blender

- Try to move vertex positions in UV Editor.
  - In the UV Editor, click a vertex - shift + space - move the vertex.

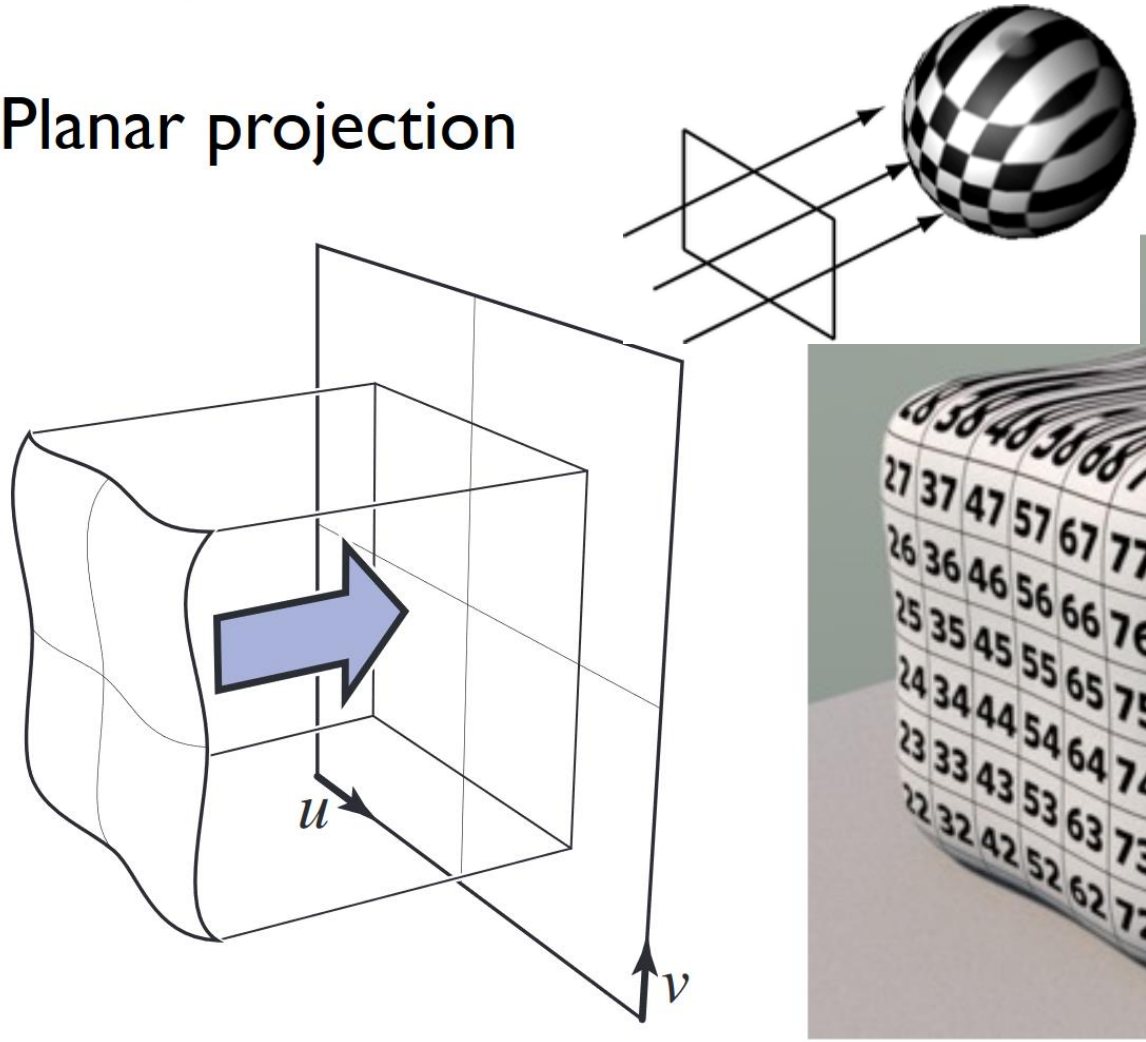
- Try to “paint” the texture.
  - Change the left view type to Image Editor
  - In the Image Editor, switch the mode to 'Paint'.
  - Select 'Draw' tool and draw as you want.

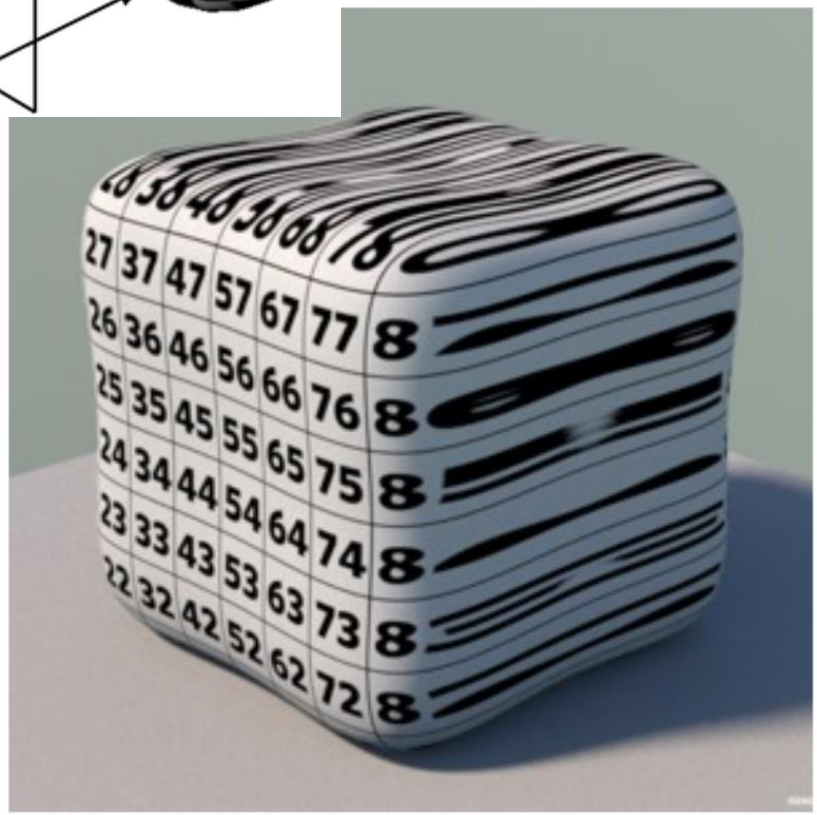
How to Make a UV Map for an Object?

- Projections to parametric surfaces
  - For objects similar to parametric surfaces such as a rectangle, sphere, cylinder
- Automatic “unwrapping” algorithms
  - For objects with complex shapes
- Generated texture coordinates are often further adjusted by artists

Examples of coordinate functions

- Planar projection

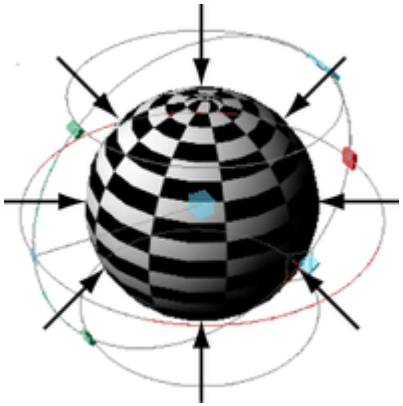


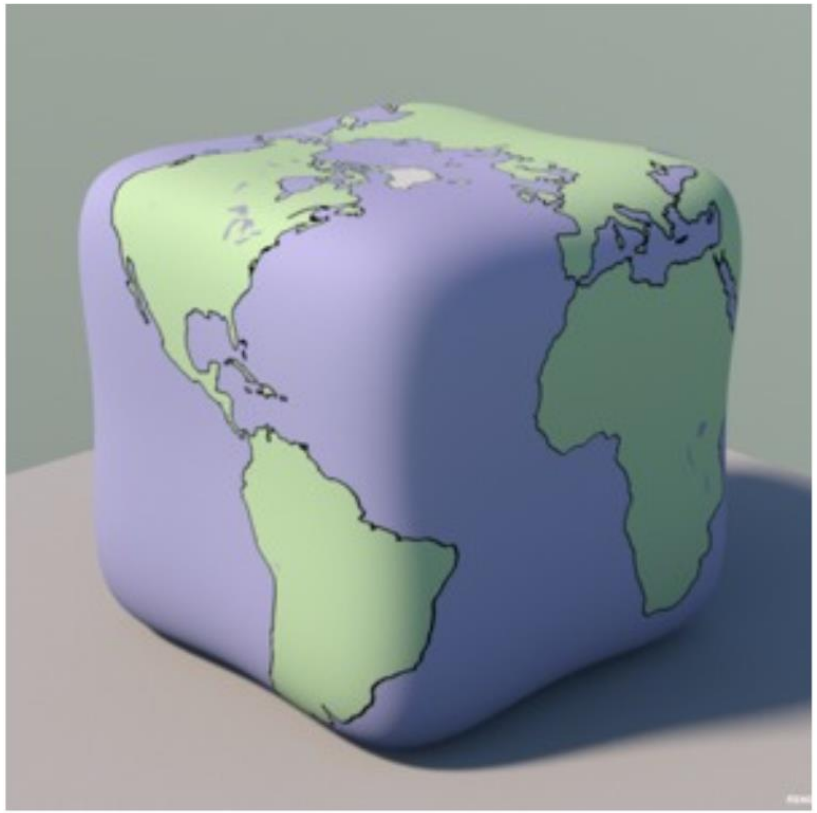


© 2014 Steve Marschner

Examples of coordinate functions

- Spherical projection

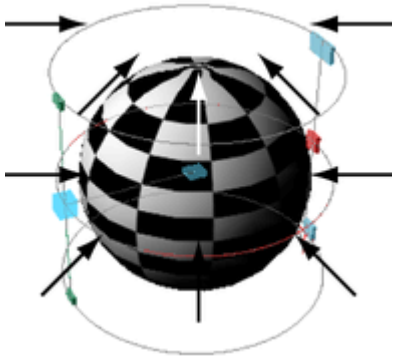


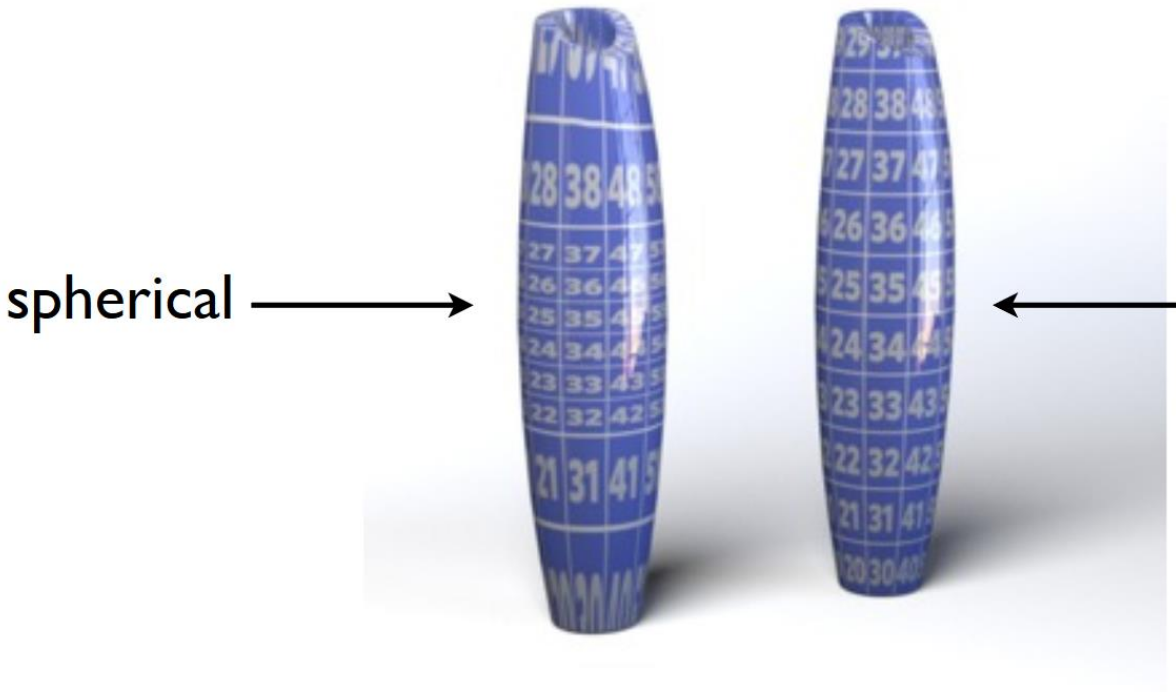


© 2014 Steve Marschner

Examples of coordinate functions

- Cylindrical projection

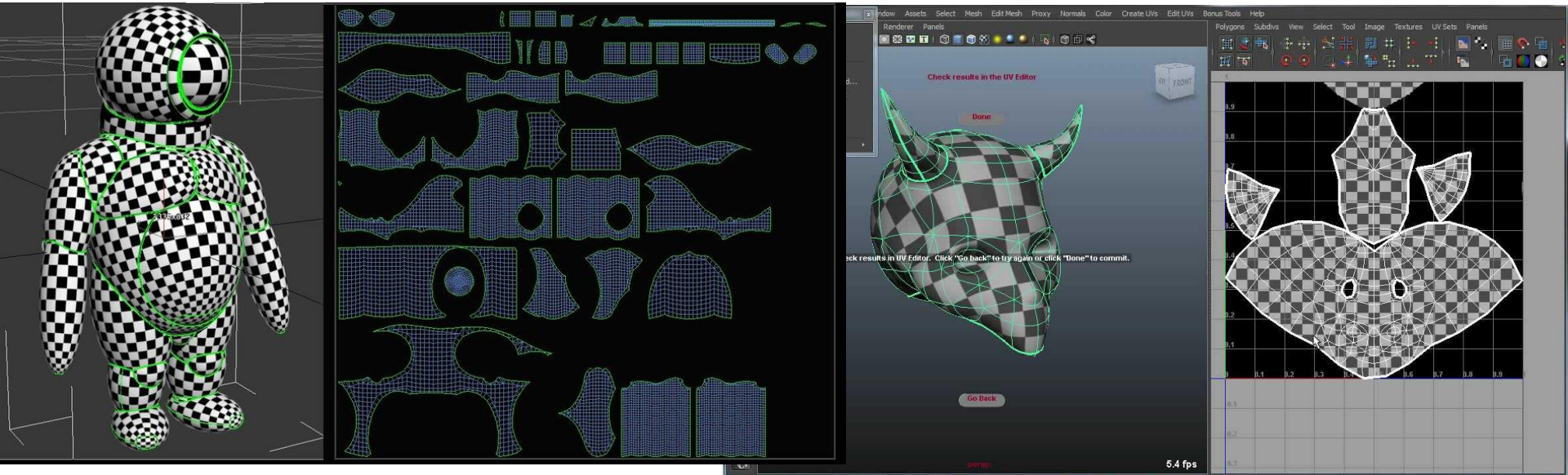




© 2014 Steve Marschner

Examples of coordinate functions

- Automatic “unwrapping” algorithms
  - Some kind of "optimization" algorithms can be used to “unwrap” the objects, which tries to choose vertex (u,v)s to result in a smooth, low distortion map
  - Different algorithms are provided by different modeling software (such as Blender, Maya, ...)



Quiz #3

- Go to https://www.slido.com/
- Join #**cg-ys**
- Click “Polls”

- Submit your answer in the following format:
  - **Student ID: Your answer**
  - **e.g. 2017123456: 4)**

- Note that you must submit all quiz answers in the above format to be checked for “attendance”.

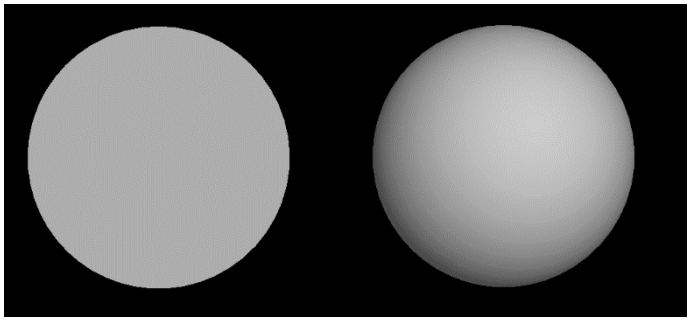
Various Uses of Texture Maps

- Texture maps are used for various purposes:

- To add surface details
  - Diffuse & specular property
  - Surface normal / height differences
  - ...

- To model lighting / environment
  - Light
  - Shadow
  - Environment
  - ...

Overview: Surface Detail

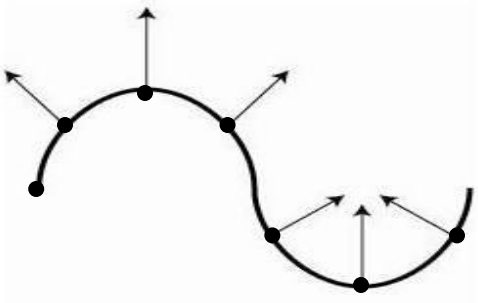


- Observation
  - What if we replaced the 3D sphere on the right with a 2D circle?
  - The circle would have fewer triangles (thus renders faster)
  - If we kept the sphere's **normals**, the circle would still look like a sphere!
  - Works because human visual system infers shape from patterns of light and dark regions ("shape from shading"). Brightness at any point is determined **by normal vector, not by actual geometry of model**

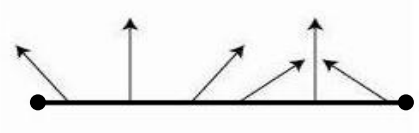
Image credit: Dave Kilian, '13

Idea: Surface Detail

- Start with a high-poly (high polygon count) model
- Make a low-poly model (remove triangles)
- Encode high-poly normal information into texture
- Map texture (i.e., normals) onto low-poly mesh

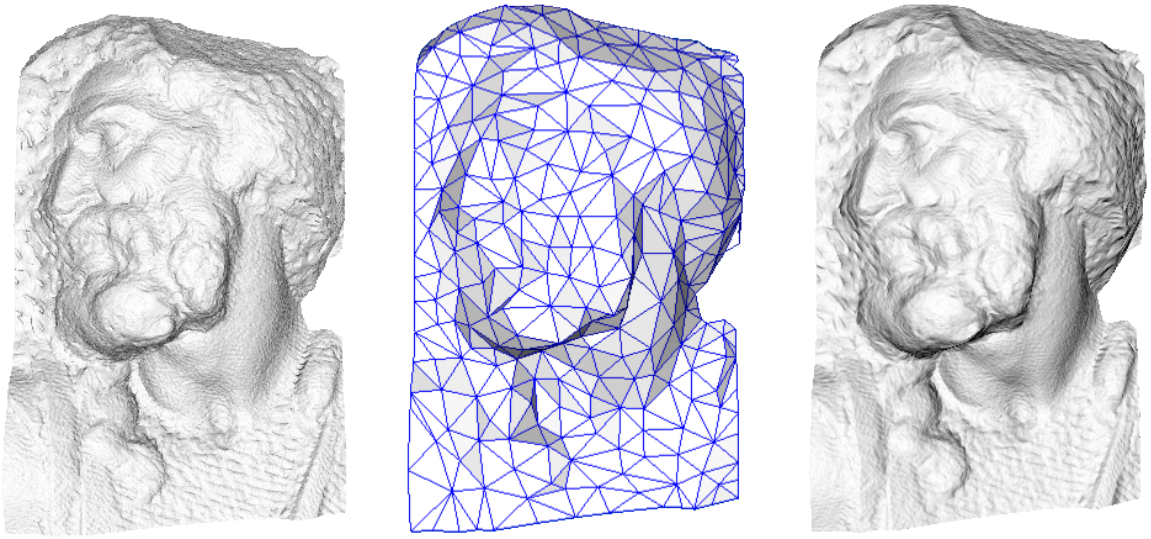


Original high-poly model



Low-poly model with high-poly model's normals preserved

Normal Mapping Example



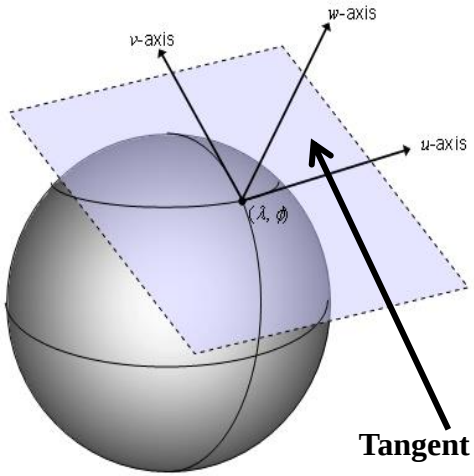
- Normal mapping can completely alter the perceived geometry of a model

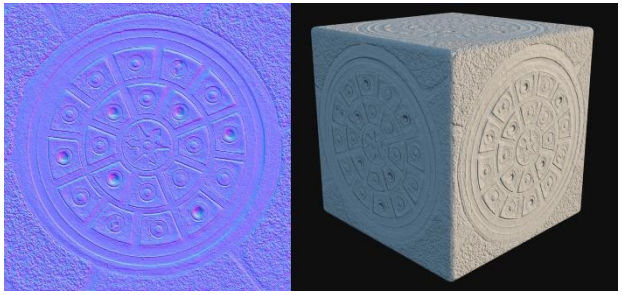
Image courtesy of www.anticz.com

Normal Mapping

- Idea: Encode normal vectors into RGB values of each pixel in a texture map

- *RGB values of each pixel* indicates the *u, v, w component of the normal vector* at the pixel's location
  - e.g. $R = N_U$, $G = N_V$, $B = N_W$
  - U: -1 to +1 → Red: 0 to 255
  - V: -1 to +1 → Green: 0 to 255
  - W: 0 to +1 → Blue: 128 to 255



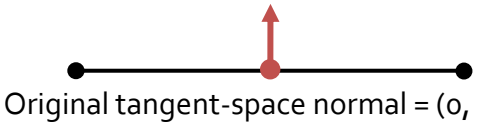


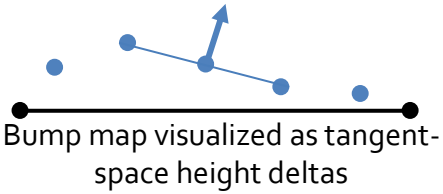
Tangent space normal map

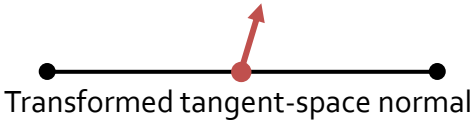
Blue is predominant as the object normals typically face the viewer ~(0, 0, 1) and get mapped to ~(128, 128, 255) in tangent space.

Bump Mapping

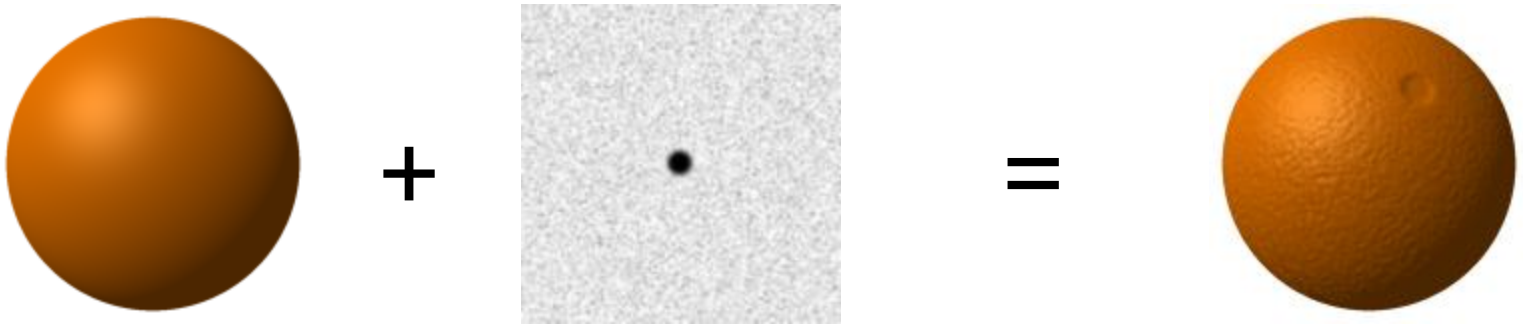
- Similar to normal mapping
- Idea: instead of encoding normals themselves in the map, encode **relative heights**
  - Black: minimum height delta
  - White: maximum height delta
- Normals are compute from a height map, and then applied







Bump Mapping: Example



Original object (plain sphere)

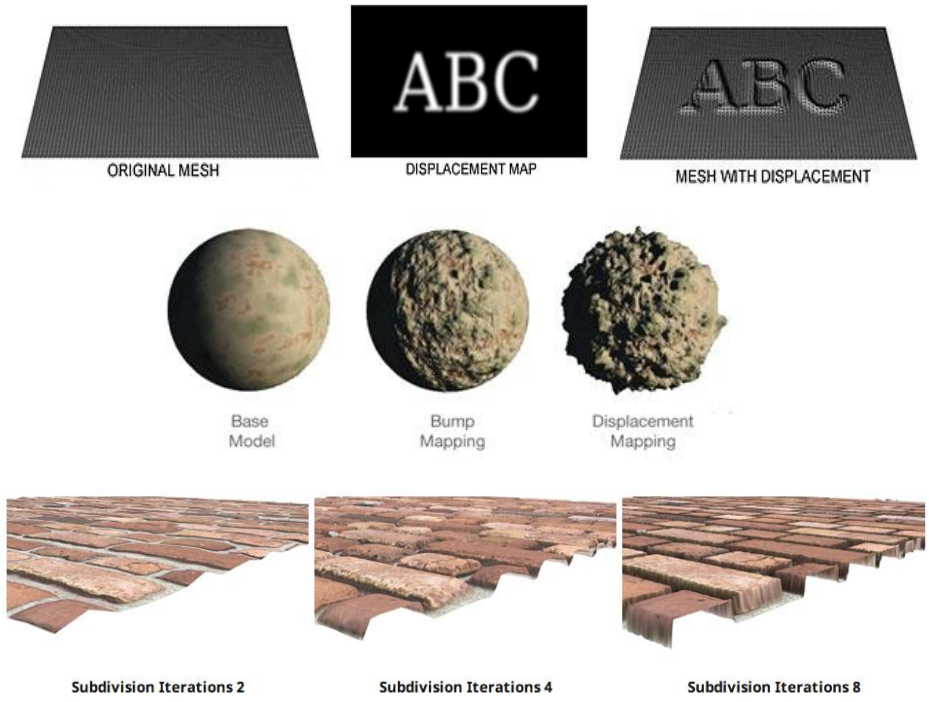
Bump map (height map to perturb normals)

Sphere with bump-mapped normals

http://cse.csusb.edu/tongyu/courses/cs520/notes/texture.php

Displacement Mapping

- *Actually move the vertices* by looking up height deltas in a height map
  - Unlike bump/normal mapping, this will produce correct silhouettes and self-shadowing
- By default, does not provide detail between vertices like normal/bump mapping
  - To increase detail we have to add more vertices, thus can become very costly



Displacement map on a plane at different levels of subdivision

http://en.wikipedia.org/wiki/Displacement_mapping
http://www.nvidia.com/object/tessellation.html
https://support.solidangle.com/display/AFMUG/Displacement

[Practice] Online Demos

- Normal mapping / Displacement mapping
  - https://threejs.org/examples/#webgl_materials_displacementmap
- Bump mapping
  - https://threejs.org/examples/#webgl_materials_bumpmap

Light maps in Quake

- **Light maps are used to store pre-computed illumination**

	Texture Maps	Light Maps
Data	RGB	Intensity
Resolution	High	Low

Light map image by Nick Chirkov

Textures Only

Textures & Light Maps

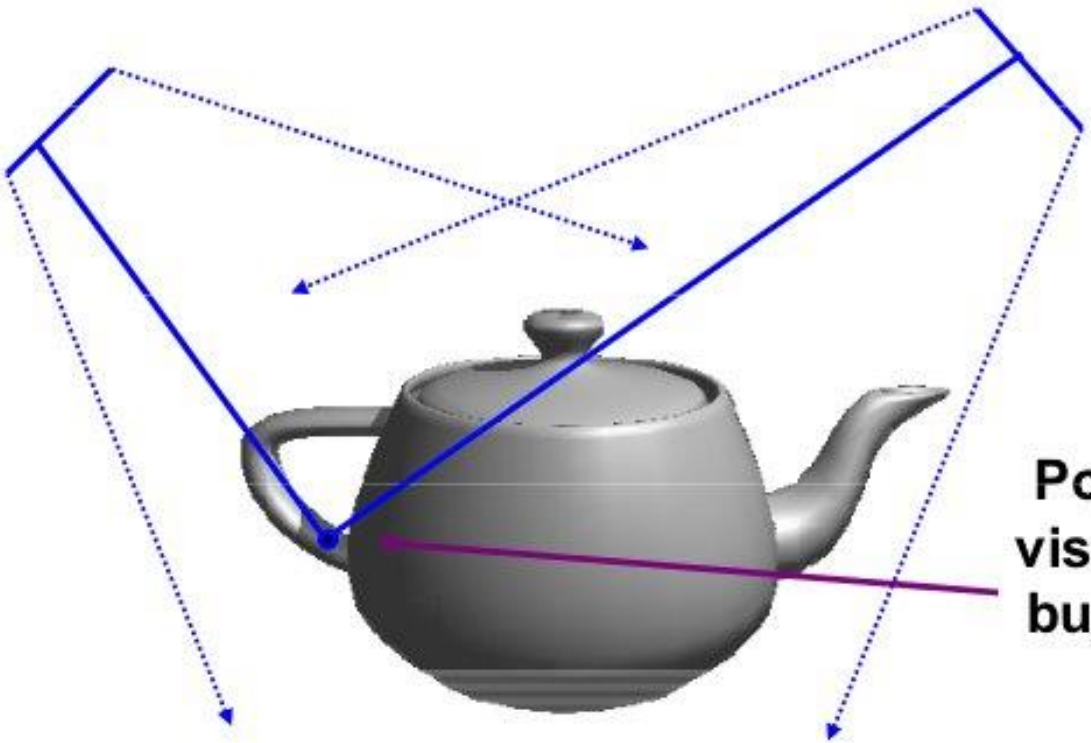
KAIST

Shadow maps

Eye

Use the depth map in the light view to determine if sample point is visible

Light

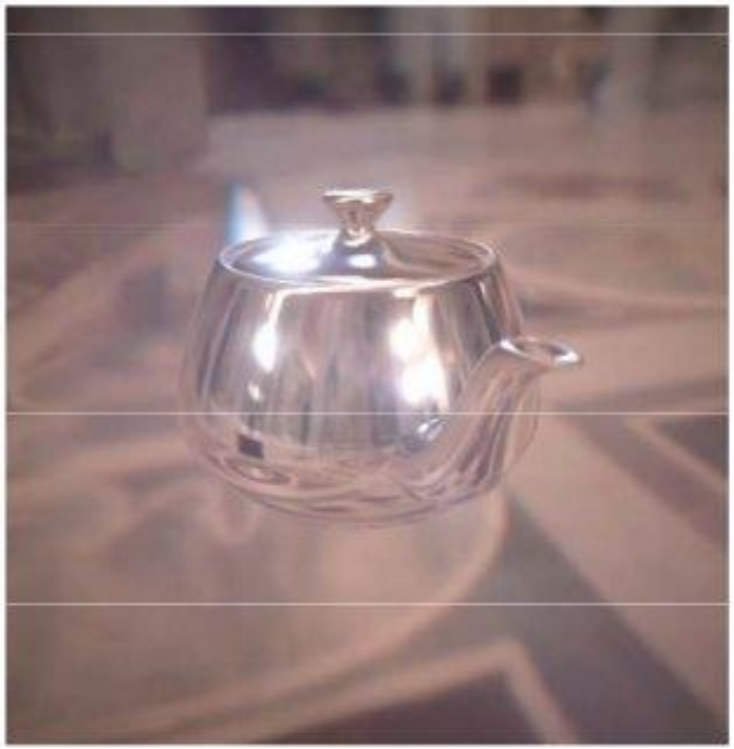


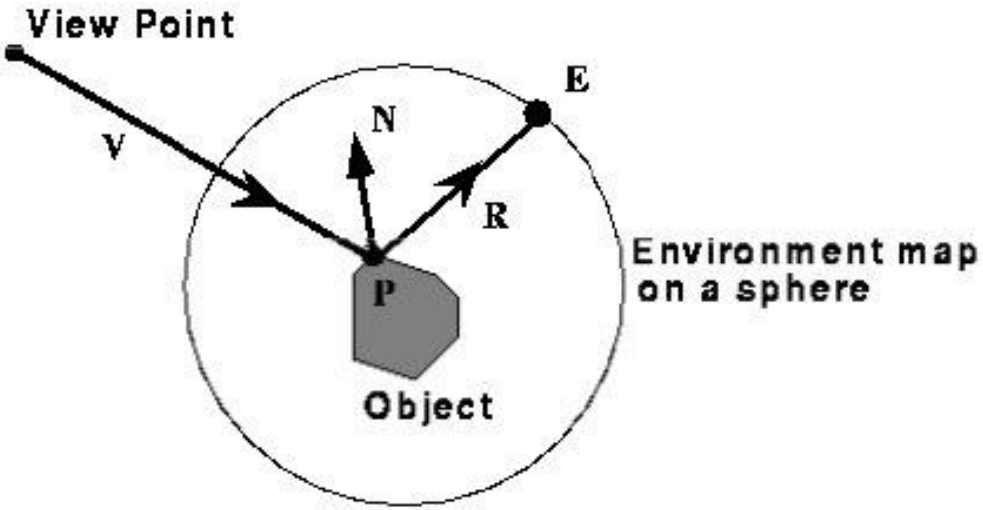
Point in shadow visible to the eye, but not visible to the light

KAIST

Environment maps

- **Simulate complex mirror-like objects**
  - **Use textures to capture environment of objects**
  - **Use surface normal to compute texture coordinates**





[Practice] Online Demos

- Light mapping
  - https://threejs.org/examples/?q=light#webgl_materials_lightmap
- Shadow mapping
  - https://threejs.org/examples/#webgl_shadowmap
- Environment mapping
  - https://threejs.org/examples/#webgl_materials_cubemap_dynamic
  - https://threejs.org/examples/#webgl_materials_cubemap_balls_refraction

Many details remain

- Cannot cover every details of texture mapping
  - Texture mapping in OpenGL
  - Magnification / minification, filtering, mipmap
  - Tiling / stretch
  - Details in various uses of texture mapping
  - ...

- We’ve covered key ideas and important applications of texture mapping.

Next Time

- Lab in this week:
  - 기말고사 원격시험 환경 설정 및 테스트 시험 1

- Next, the last lecture:
  - 13 - Rasterization & Visibility

- Acknowledgement: Some materials come from the lecture slides of
  - Prof. Andy van Dam, Brown Univ., http://cs.brown.edu/courses/csci1230/lectures.shtml
  - Prof. Sung-eui Yoon, KAIST, https://sglab.kaist.ac.kr/~sungeui/CG/
  - Prof. Kayvon Fatahalian and Prof. Keenan Crane, CMU, http://15462.courses.cs.cmu.edu/fall2015/