Computer Graphics

12 - More Lighting, Texture

Yoonsang Lee
Hanyang University

Spring 2023

Final Exam Announcement

- Date & time: **June 12 (Mon), 7:30 - 8:30 PM**
- Place: IT.BT **507, 508**
  - Student list for each room will be announced later.

- Scope: Lecture & Lab **8 ~ 13**

- **You cannot leave until 30 minutes after the start of the exam** even if you finish the exam earlier.

- That means, **you cannot enter the room after 30 minutes from the start of the exam** (do not be late, never too late!).

- Please bring your **student ID card** to the exam.

Outline

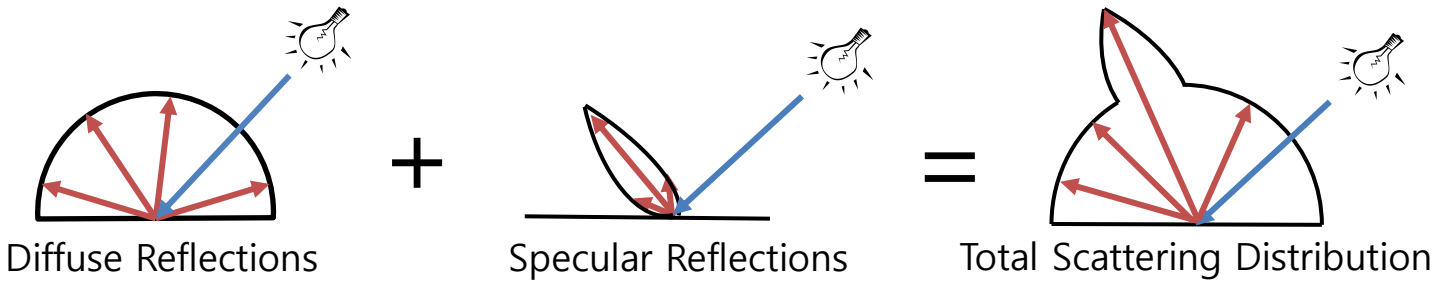
- More Lighting
  - BRDF
  - Local & Global Illumination

- Texture Mapping
  - Concept
  - UV Mapping
  - Texture Mapping Process
  - Defining Texture Coordinate Function
  - Rendering Texture-Mapped Objects
  - Diffuse, Specular, Normal Maps
  - Various Uses of Texture Maps

More Lighting

Recall: Reflection of General Materials

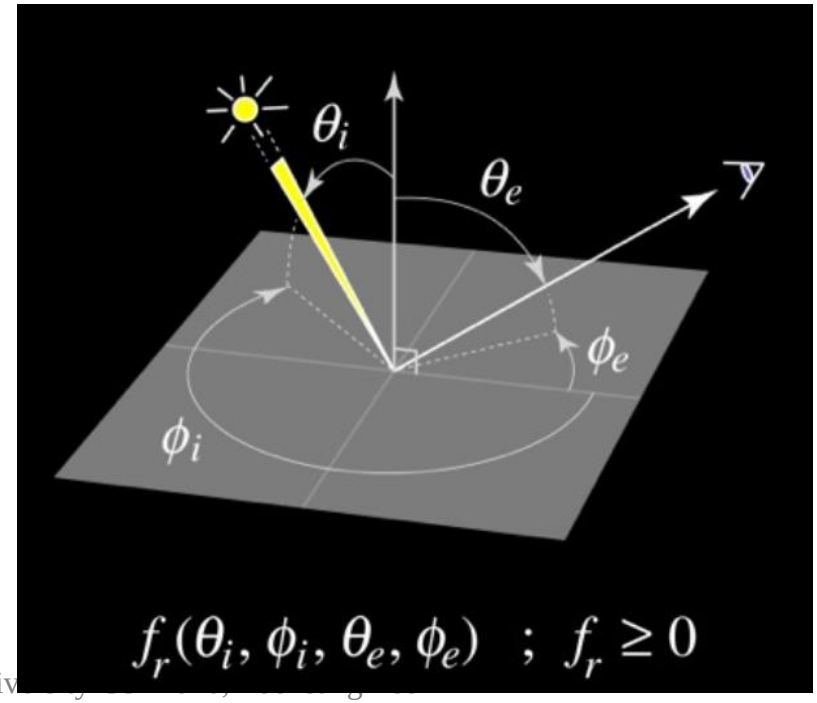
- Many materials' surface have both diffuse reflection and specular reflection.



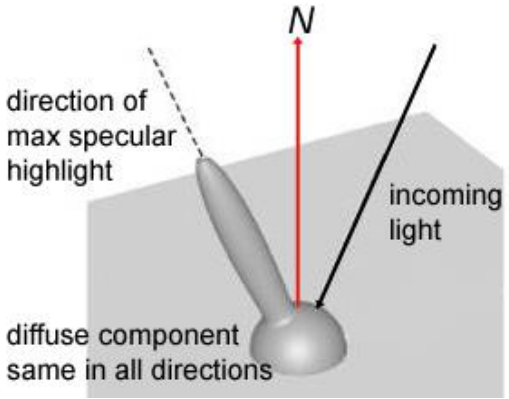
- We can represent reflectance properties of a surface as a **distribution function.**
- → **BRDF**

Bidirectional Reflectance Distribution Function (BRDF)

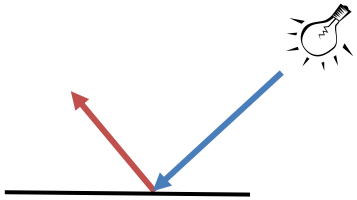
- Defines how light is reflected at an opaque surface.
  - θ_i, φ_i : incoming light direction
  - θ_e, φ_e : outgoing light direction
  - f_r returns the ratio of reflected radiance exiting along outgoing direction

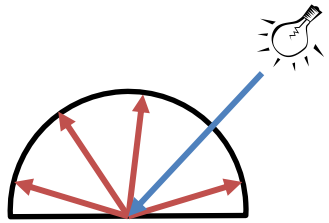


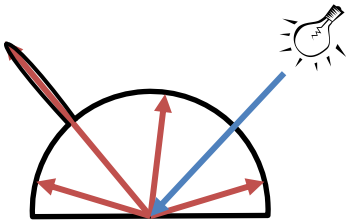
An example BRDF

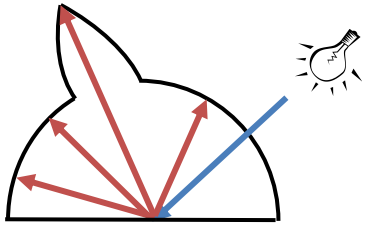


Examples of BRDF
(theoretical approximation, not from measurement)



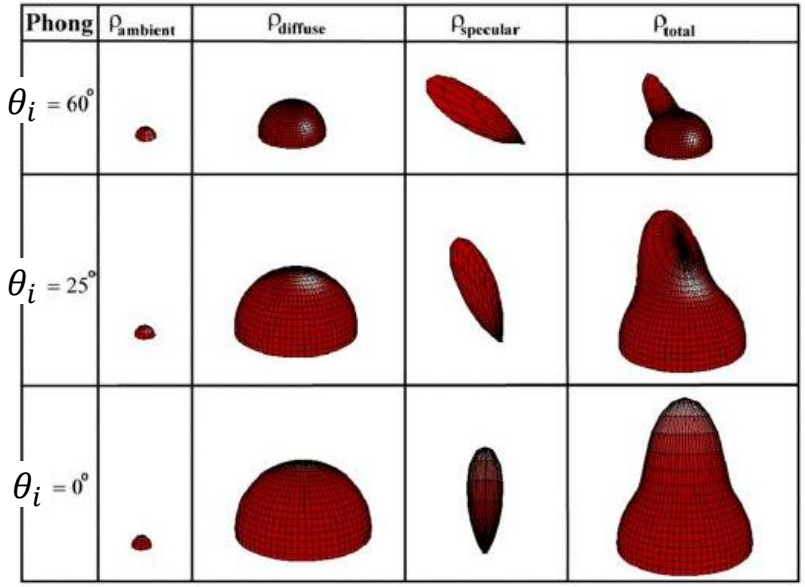






Phong Illumination Model & BRDF

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



Measuring BRDF

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

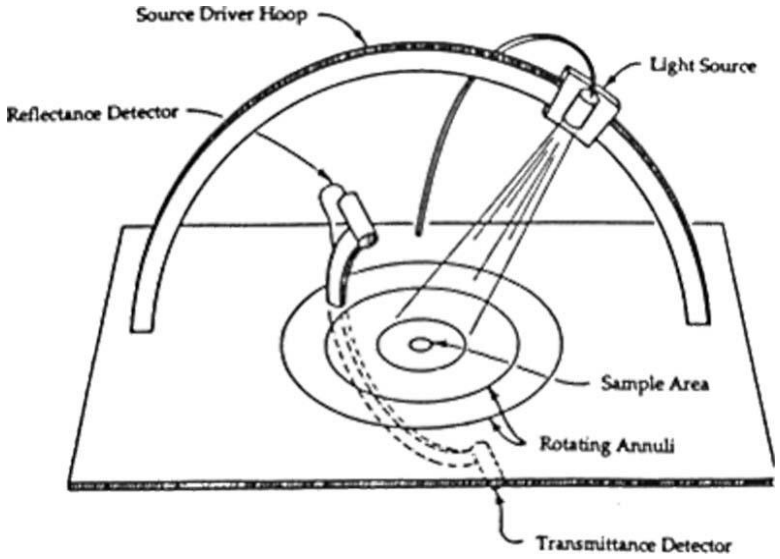
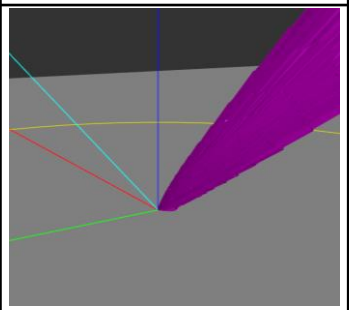
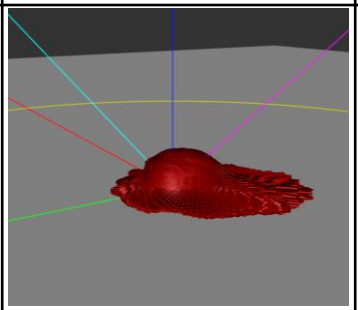
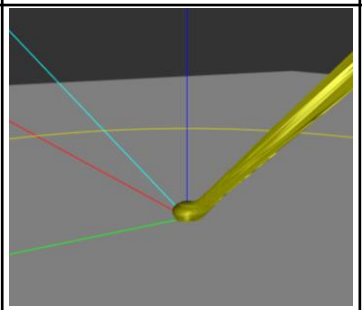
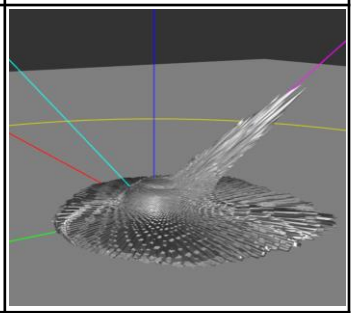
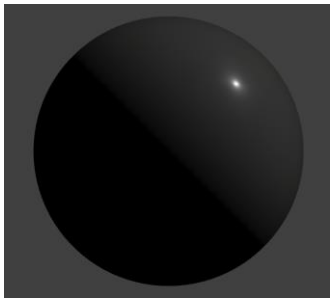
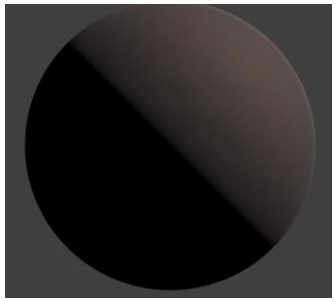


Image credit: Chuck Moidel

Using Measured BRDF for Rendering

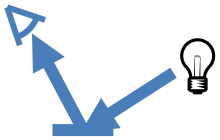
- Measured BRDFs can be used for rendering.

	Nickel	Pink fabric	Gray plastic	Nylon
Measured BRDF				
Rendered object				
Real object example				

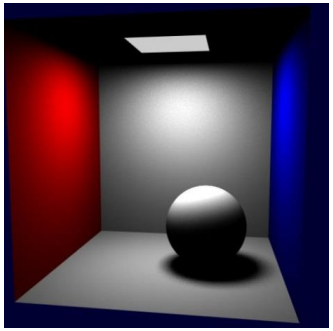
* BRDF data and some images are from https://cdfg.csail.mit.edu/wojciech/brdfdatabase

Local vs. Global Illumination

- **Local (or direct, or non-global) illumination**
  - Models light that **comes directly from a light source**
  - Can be rendered fast, but less realistic (unrealistic)
  - e.g. Phong illumination model

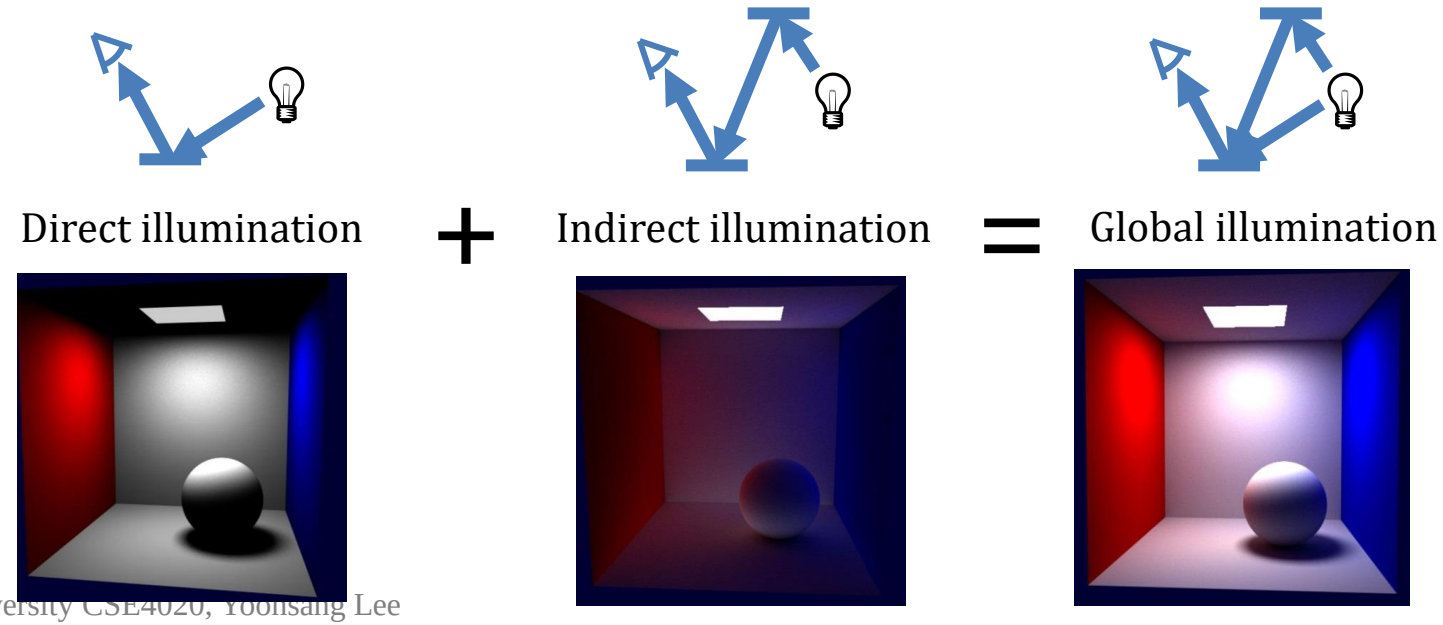


Direct illumination



Local vs. Global Illumination

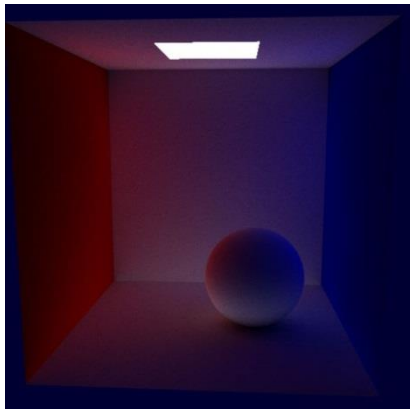
- **Global illumination**
  - Direct illumination +
  - Indirect illumination that models light **reflected by other surfaces** (inter-object reflections)
  - Slow, but much more realistic
  - e.g. Ray tracing, Path tracing, Radiosity



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

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. 2021123456: 4.0**

- Note that your quiz answer must be submitted **in the above format** to receive a quiz score!

Texture Mapping

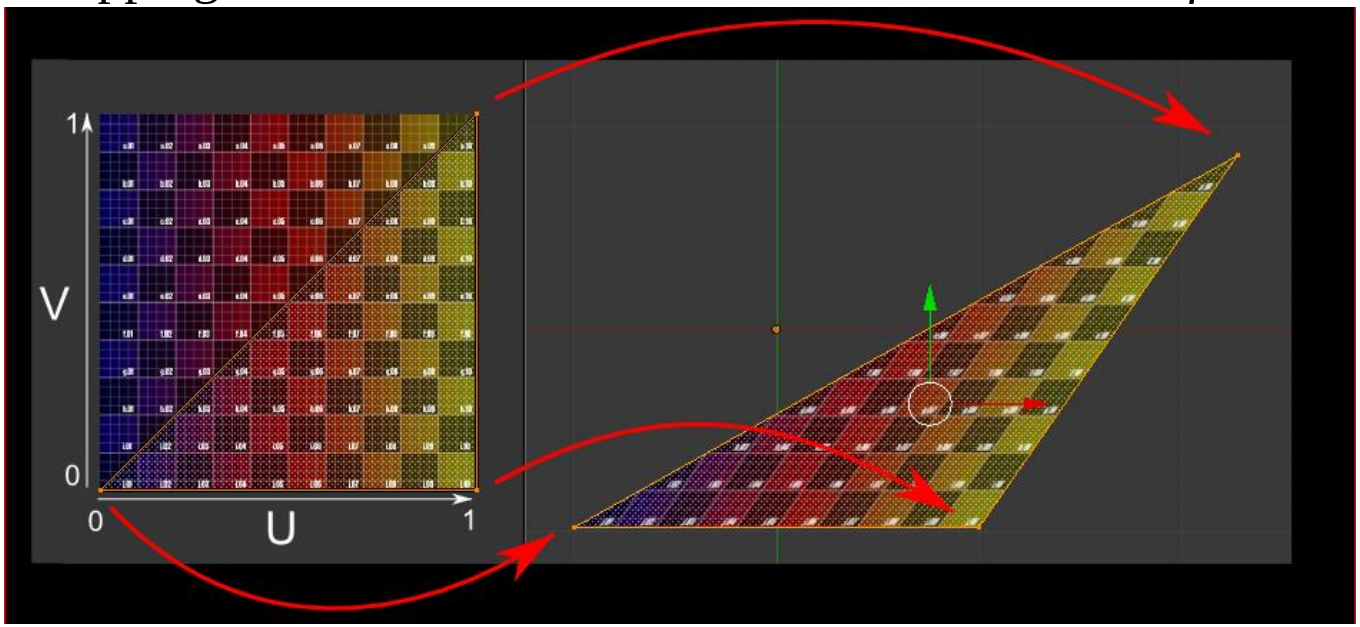
Texture Mapping

- A technique of defining surface properties as a function of position on the surface
  - Usually, 2D texture images (a.k.a. texture maps) are applied onto a 3D object' surface.

- Main objective: Adding surface detail
  - Compared to high-poly models ,
  - Higher level of detail with fewer polygons
  - Significantly faster rendering speed
    - Image complexity does not impact processing complexity.

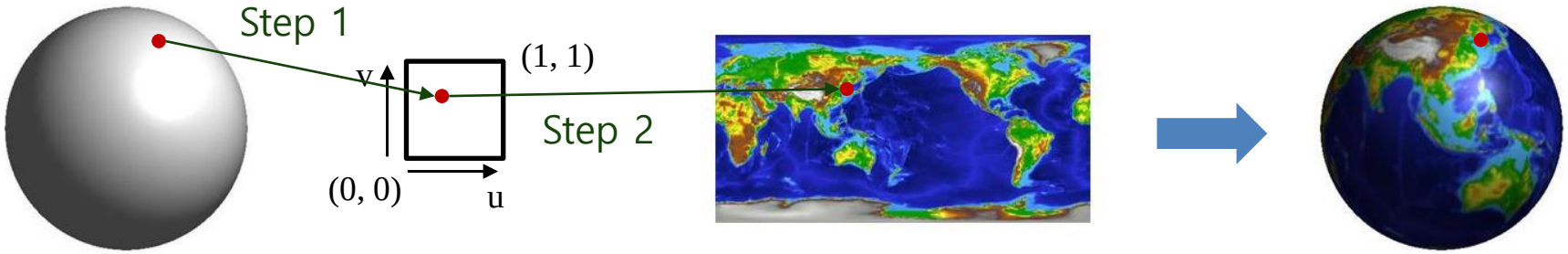
UV Mapping

- Texture mapping is commonly defined using *UV mapping*.
- UV mapping assigns 2D texture coordinates (*u*, *v*) to each vertex of a 3D model.
  - Describes the specific location on a 2D image where each vertex corresponds to.
- UV mapping function is also known as *texture coordinate function*.



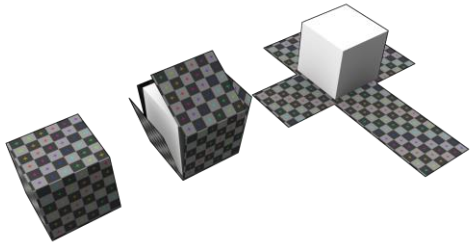
 * This image is from http://www.opengl-tutorial.org/beginners-tutorials/tutorial-5-a-textured-cubeD

Texture Mapping Process



- 1. Map a surface point to a point in UV space (UV mapping).
- 2. Map the UV coordinates to a point on texture.
- Step 2 is simply a scaling, automatically done by system.
- Step 1 is the *texture coordinate function* we need to define.

Defining Texture Coordinate Function - Creating "UV map"

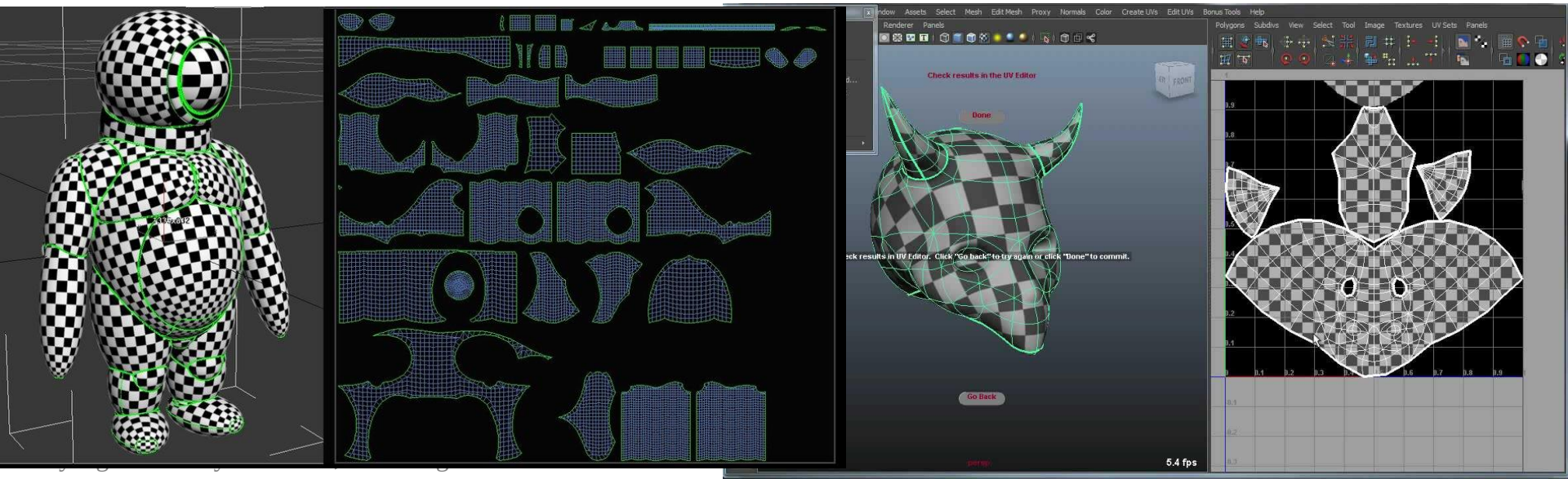


- “Unwrap” an object
  - by using 3D modeling software such as Blender

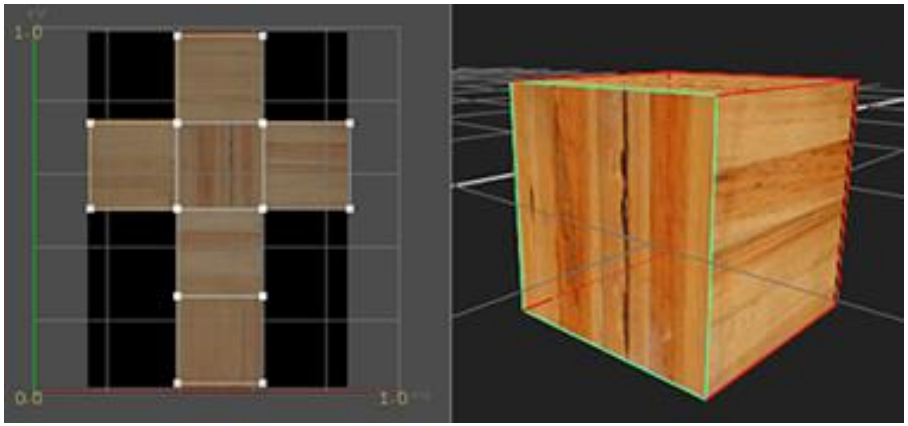
- (If necessary) Artists refine the generated texture coordinates (UV map) through manual adjustments.

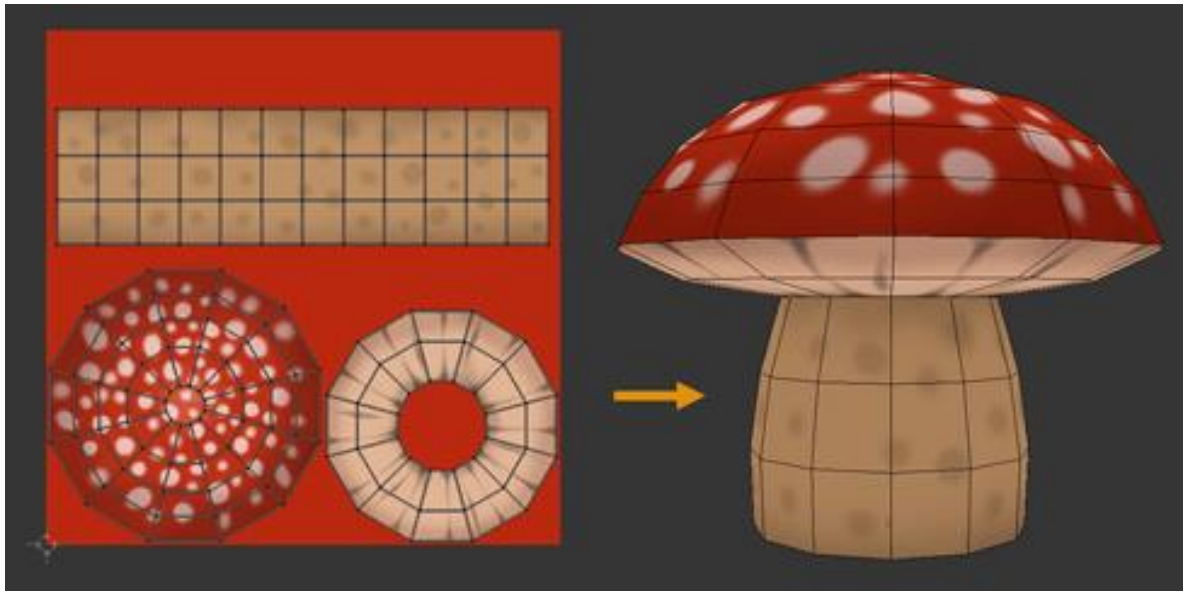
"Unwrapping"

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



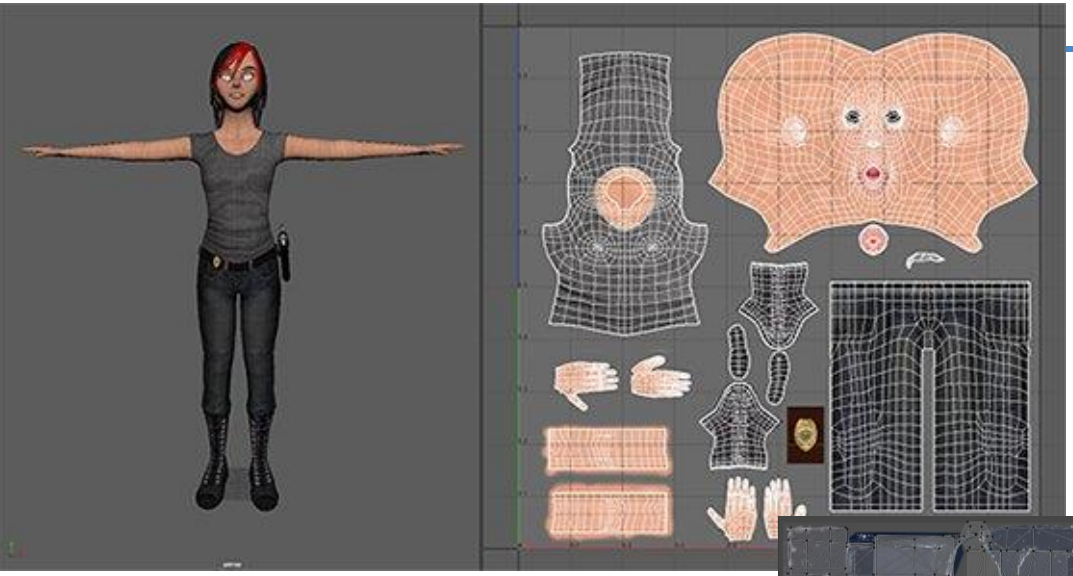
UV Map Examples

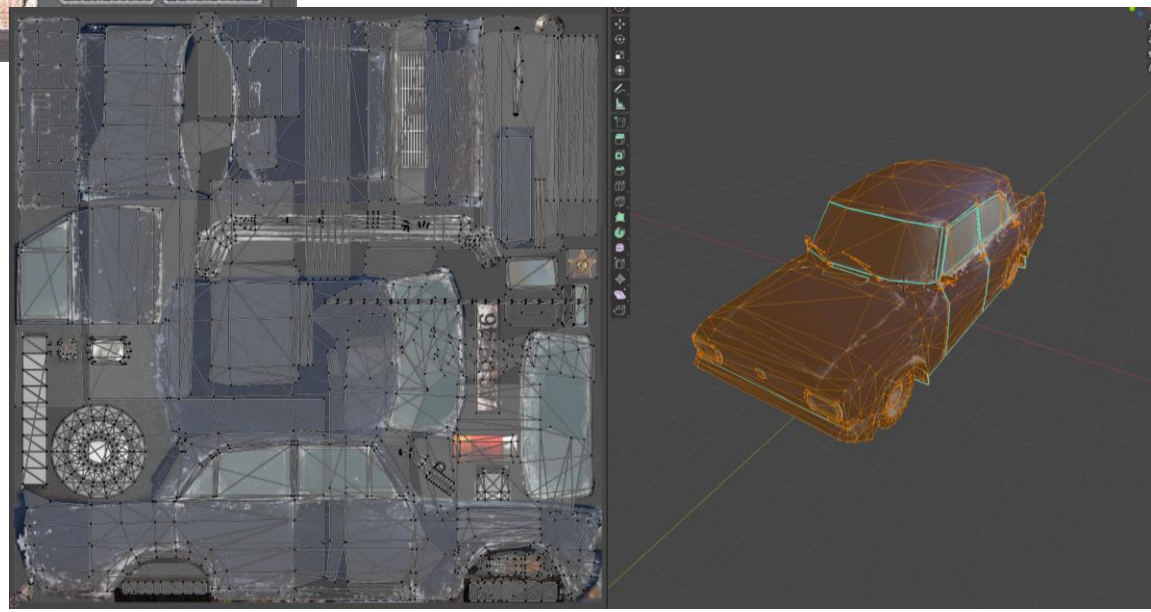




* The images are from
https://learn.foundry.com/nuke/content/comp_environment/modelbuilder/uv_unwrapping.html,
https://www.blendernation.com/2017/04/22/tutorial-modeling-uv-unwrapping-texturing-mushroom/

UV Map Examples





* The images are from https://help.autodesk.com/view/MAYAUL/2023/ENU/?guid=GUID-FDCD0C68-2496-4405-A785-3AA93E9A3B25, Blender rendering of a model downloaded from https://free3d.com

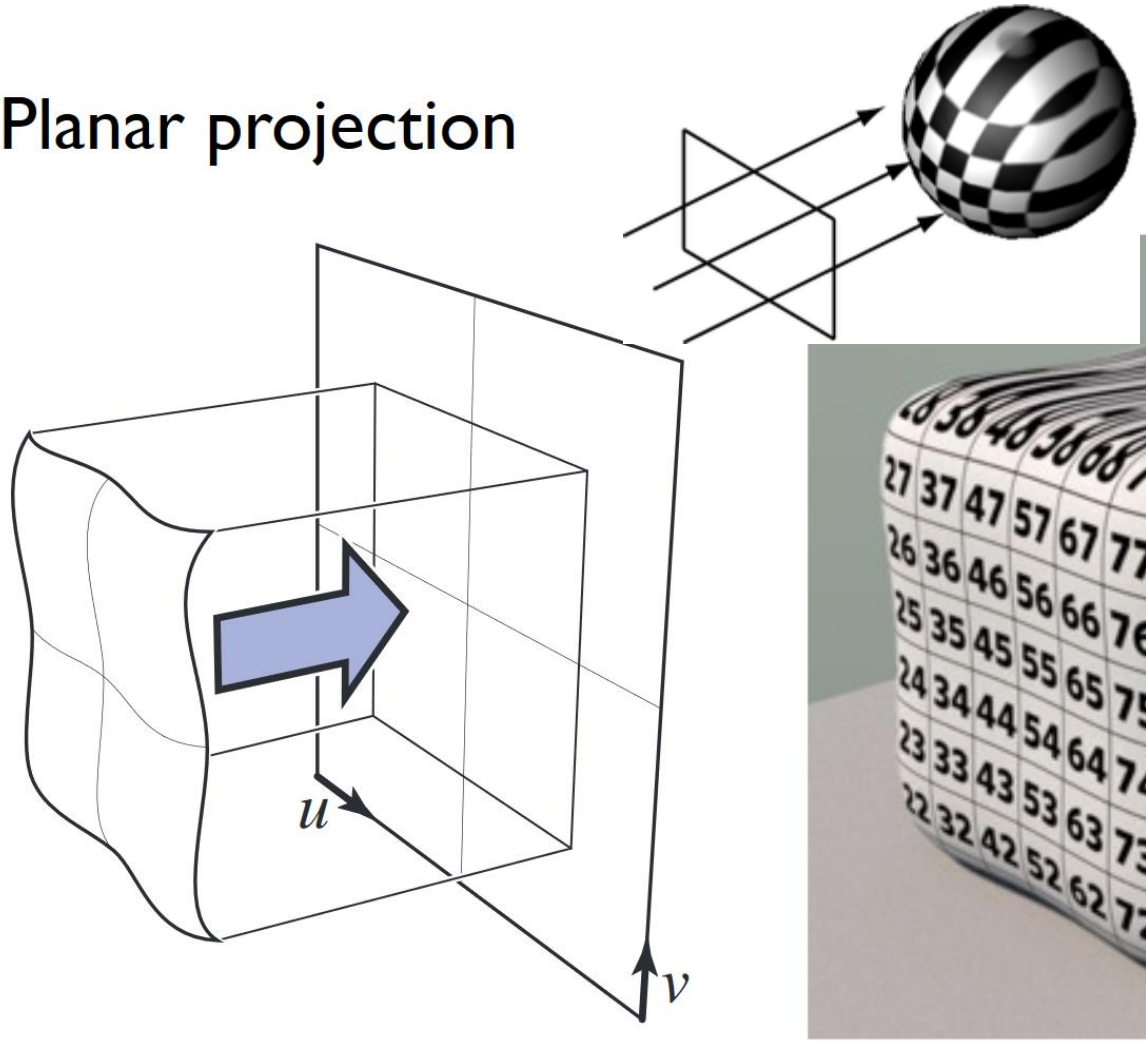
Defining Texture Coordinate Function - Projections to Parametric Surfaces

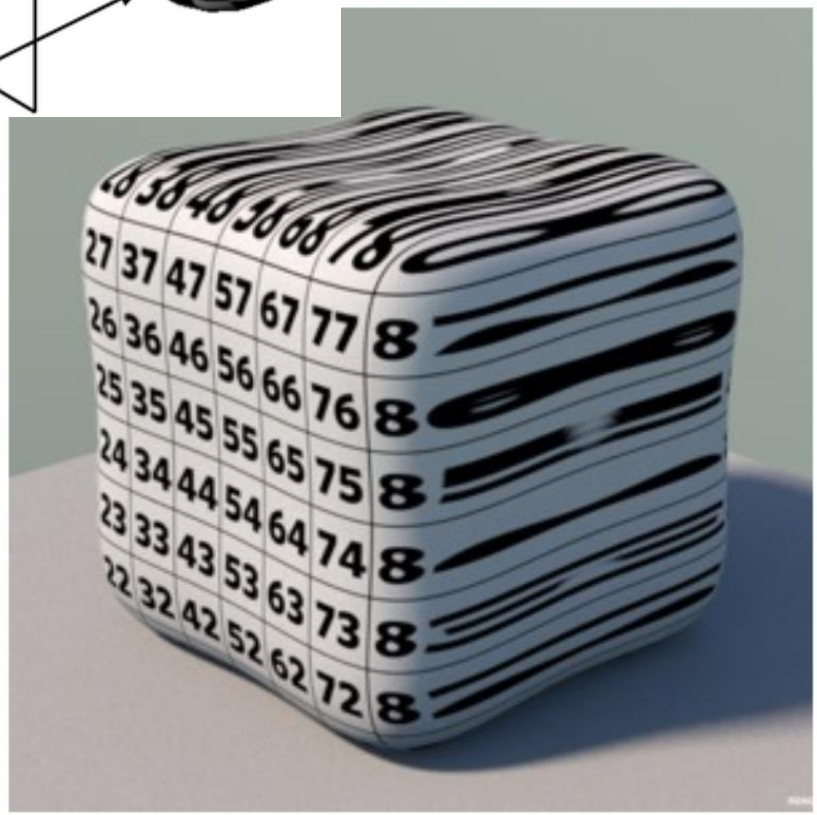
- For objects similar to parametric surfaces such as a rectangle, sphere, cylinder,

- We can project the object vertices onto those parametric surfaces. → Texture coordinate function

Examples of coordinate functions

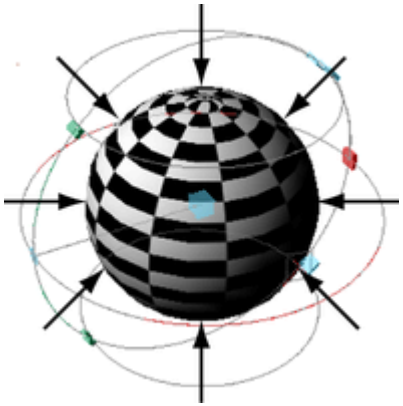
- Planar projection

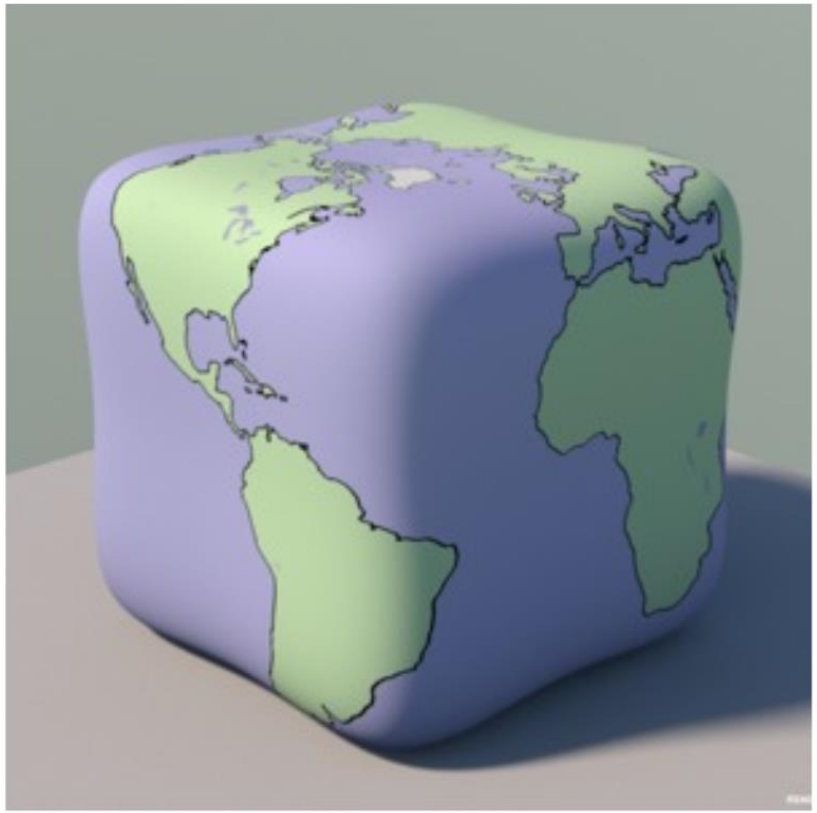




Examples of coordinate functions

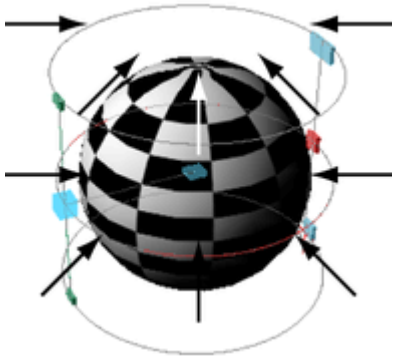
- Spherical projection

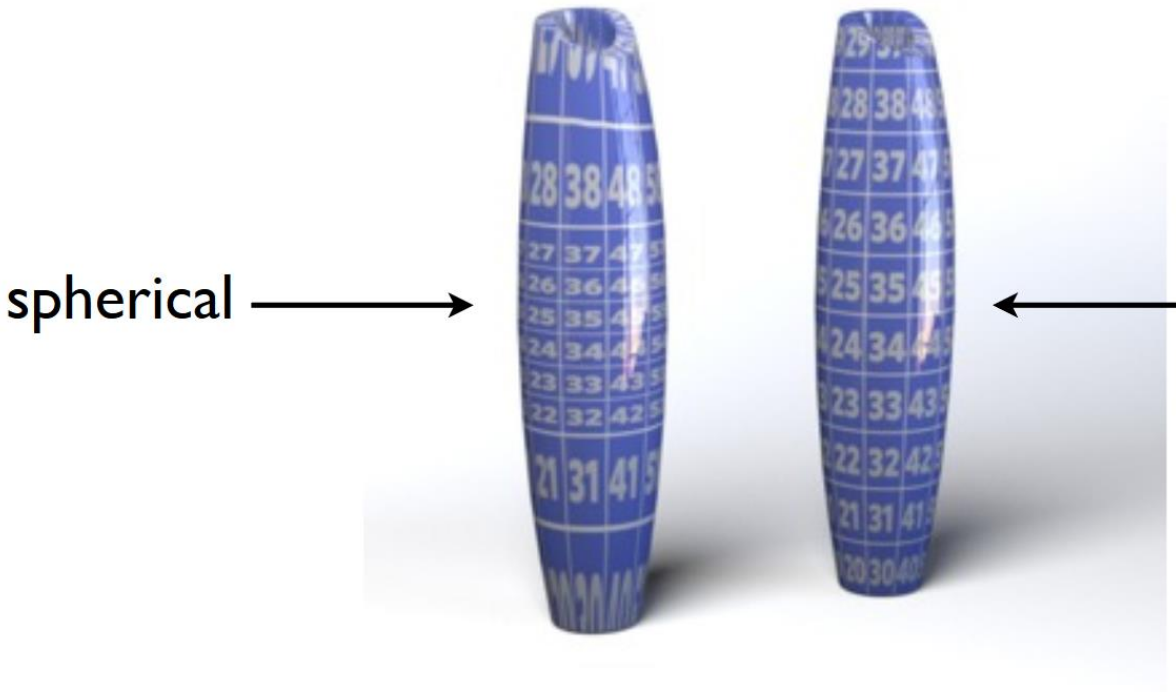




Examples of coordinate functions

- Cylindrical projection





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. 2021123456: 4.0**

- Note that your quiz answer must be submitted **in the above format** to receive a quiz score!

Rendering Texture-Mapped Objects

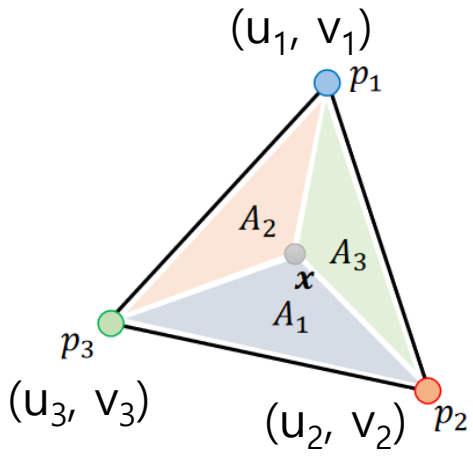
```
for each pixel:
    (u, v) = texture coordinates at pixel
    color = texture.get_color(u, v)
    pixel.color = color
```

- Texture coordinate functions defines texture coordinates for each vertex.

- How can the texture coordinates at each pixel be calculated?

Rendering Texture-Mapped Objects

- Texture coordinates are barycentric-interpolated across polygon.
  - (u, v) for each pixel is determined based on the positions of the pixel in the polygon.



$\alpha_1 = {}^{A_1}/_{A}$ percentage (u_1, v_1)

$\alpha_2 = {}^{A_2}/_{A}$ percentage (u_2, v_2)

$\alpha_3 = {}^{A_3}/_{A}$ percentage (u_3, v_3)

A ... area of whole triangle

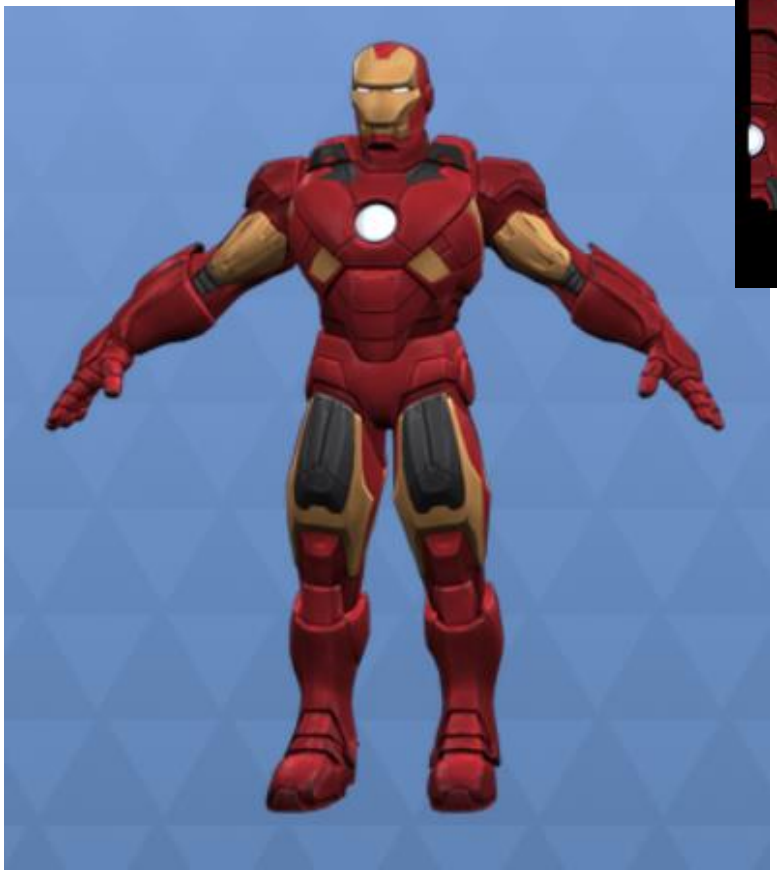
$$(u_x, v_x) = \alpha_1(u_1,v_1) + \alpha_2(u_2,v_2) + \alpha_3(u_3,v_3)$$

$$\alpha_1 + \alpha_2 + \alpha_3 = 1$$

Diffuse, Specular, Normal Maps

- Which surface property can be defined using a texture map?
  - **Diffuse color**, **specular color**
  - Specular exponents, transparency or reflectivity coefficients
  - **Surface normal**
  - Projected reflections or shadows
  - ...

Examples of Diffuse, Specular, Normal Map



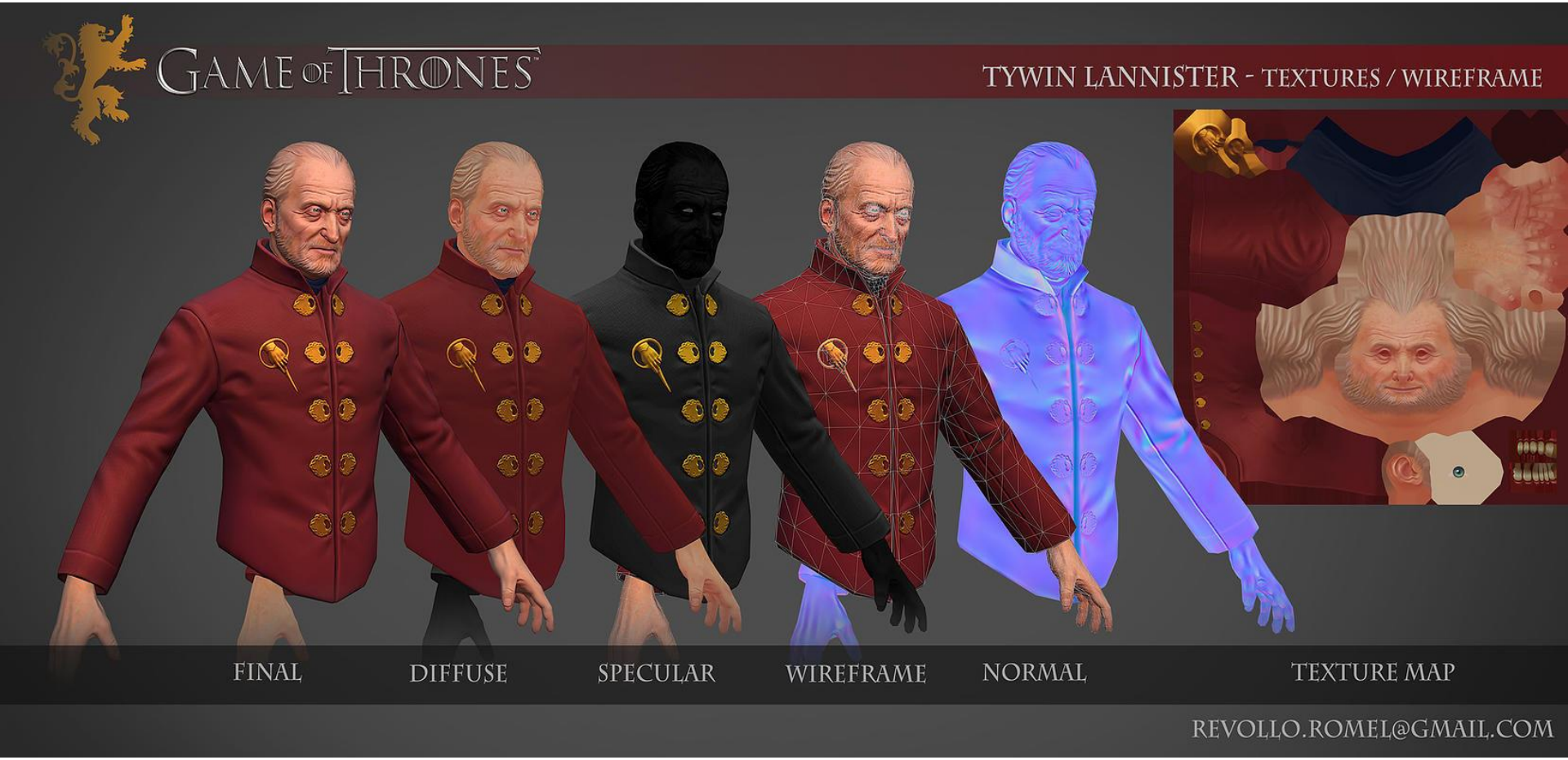
diffuse map

specular map

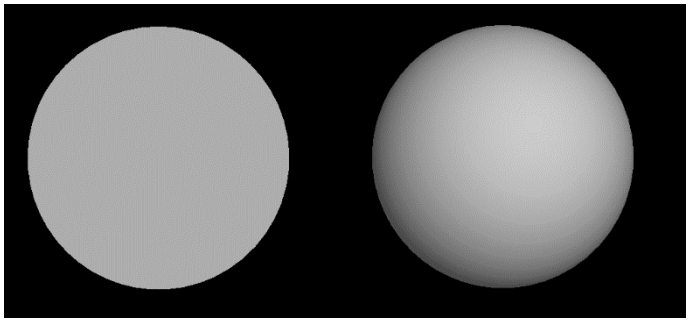
normal map

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

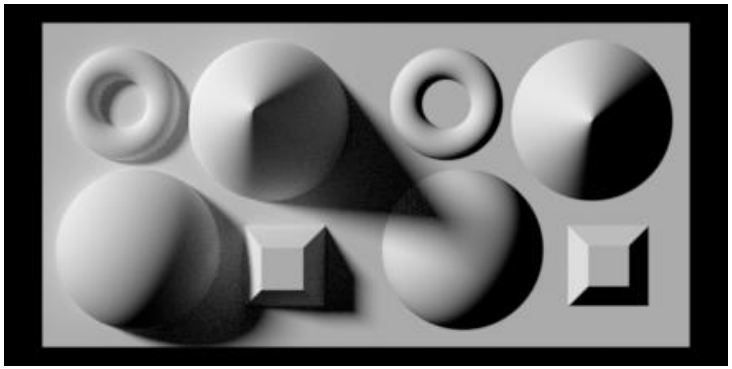
Examples of Diffuse, Specular, Normal Map



Normal Mapping - Motivation



circle sphere



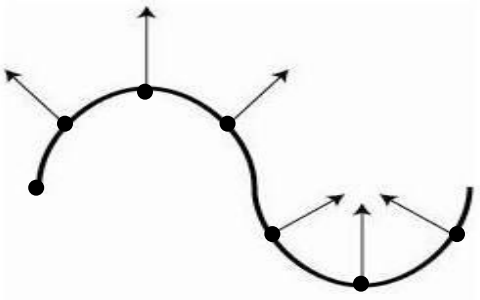
https://commons.wikimedia.org/wiki/File:Rendering_with_normal_mapping.gif

- Rendering a circle would be faster than a sphere, because a circle would have fewer triangles.
- The circle can appear like a sphere if it has the sphere's normal vectors!
- Because the human visual system infers the shape based on the patterns of **light and dark areas**, a process known as "shape from shading", and this is **determined by the normal** rather than the actual geometry of the model.

Normal Mapping - Basic Idea

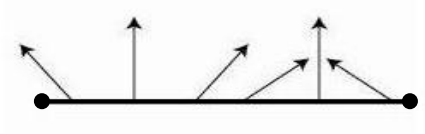
- Goal: Getting a low-poly model that "looks like" a high-poly model.

- 1. (Somehow) Encode the normal of a high-poly model into a texture image.



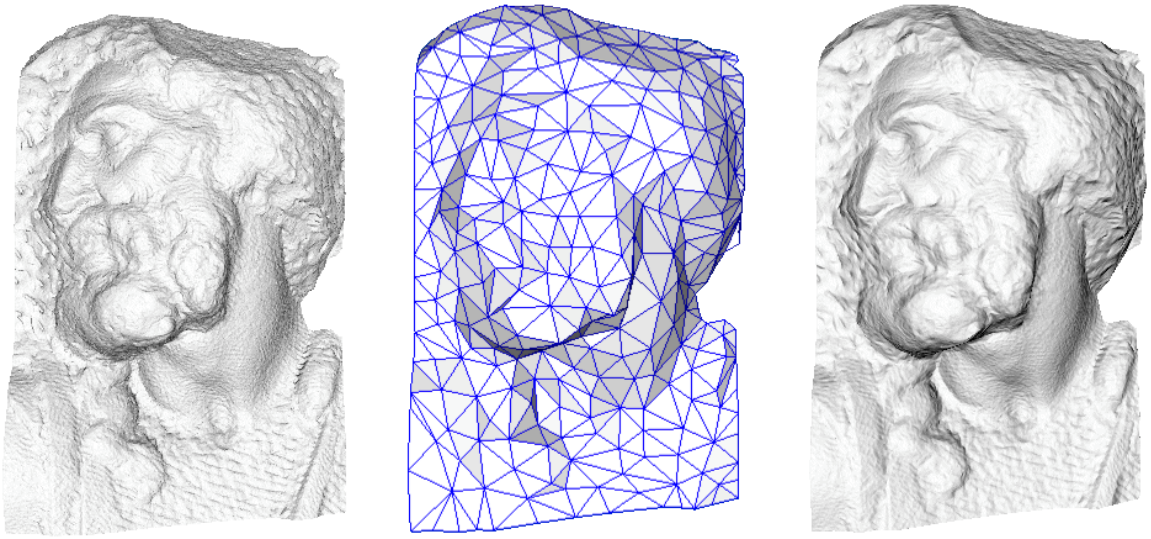
Original high-poly model

- 2. Map the texture image onto a low-poly model.



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

Normal Mapping - Example



original mesh
4M triangles

simplified mesh
500 triangles

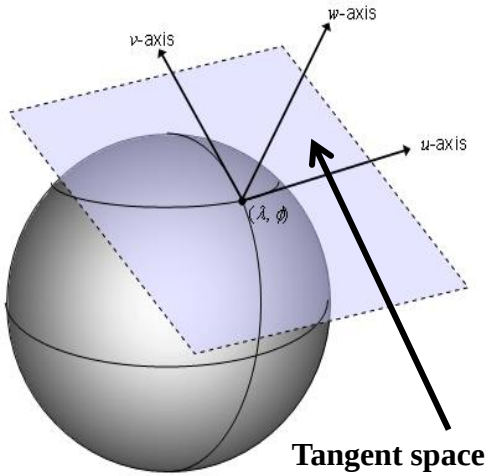
simplified mesh
and normal mapping
500 triangles

* The image is from the slides of Prof. Andy van Dam (Brown Univ.)
http://cs.brown.edu/courses/csci1230/lectures.shtml

Normal Mapping - Details

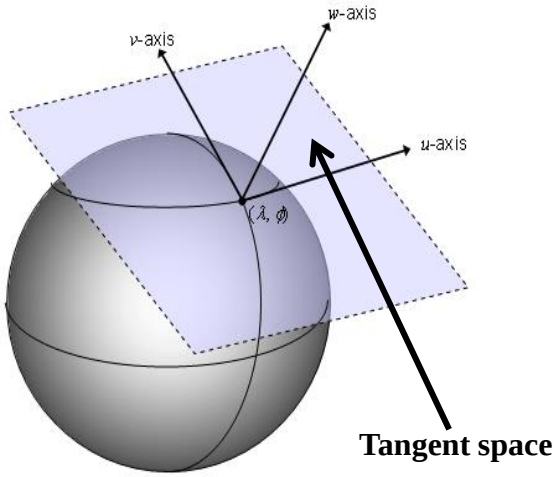
- Specifically, normal vectors of a high-poly model are encoded into RGB values of each pixel in a texture map.

- In *tangent space normal map*, RGB values of each pixel indicates the u, v, w component of the normal vector at the corresponding location on an object surface.
- i.e., $R = N_U$, $G = N_V$, $B = N_W$
  - u: u-axis of texture coordinates
  - v: v-axis of texture coordinates
  - w: surface normal at that point (of a low-poly model)
    - Refer http://foundationsofgameenginedev.com/FGED2-sample.pdf for the computation of u and v vector

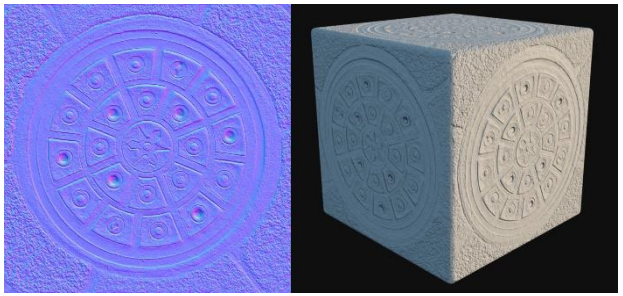


Normal Mapping - Details

- $R = N_U$, $G = N_V$, $B = N_W$
  - U: -1 to +1 → Red: 0 to 255
  - V: -1 to +1 → Green: 0 to 255
  - W: 0 to +1 → Blue: 128 to 255



- That's why a tangent space normal map is usually *bluish*.
  - Normals usually point outward, around (0, 0, 1) in tangent space, which is (128, 128, 255) in RGB space.



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. 2021123456: 4.0**

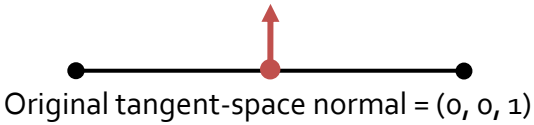
- Note that your quiz answer must be submitted **in the above format** to receive a quiz score!

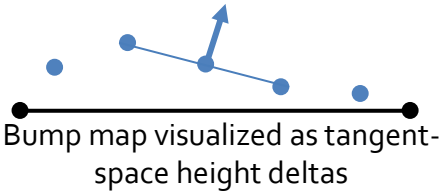
Bump Mapping

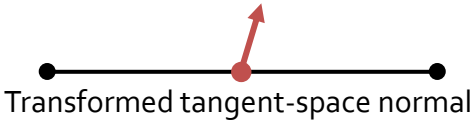
- Same goal as normal mapping

- Idea: Instead of encoding normal vectors in a texture map, encode **relative heights.**
  - Black: minimum height delta
  - White: maximum height delta

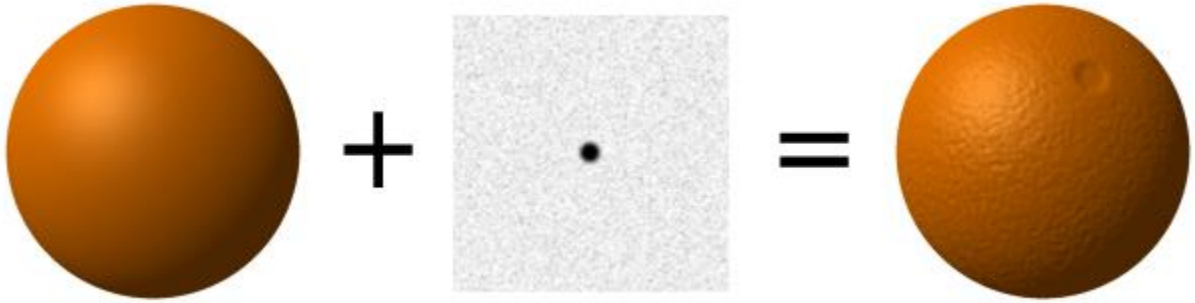
- Normals are computed from the height map, and then applied.







Bump Mapping Example

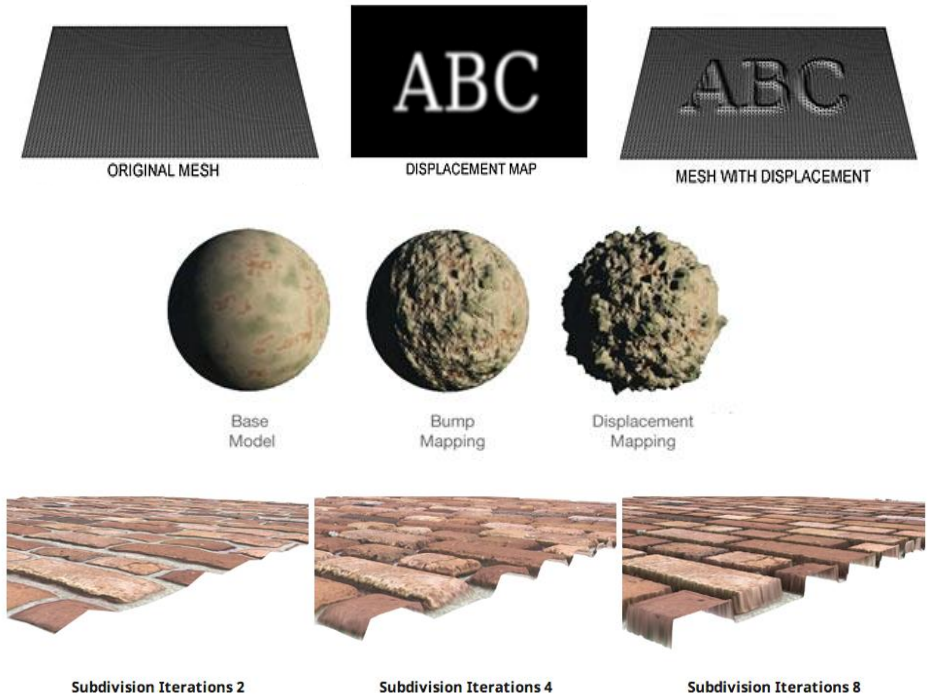


original object bump map bump-mapped object

Displacement Mapping

- *Actually move the vertices* by looking up height deltas in a height map.
  - Unlike bump/normal mapping, it produces **correct silhouettes** and self-shadowing.

- By default, it does not provide detail between vertices like normal/bump mapping.
  - To increase detail, we have to add more vertices, thus it 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

[Demo] Normal / Bump / Displacement Mapping

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

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

Light Map

- **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

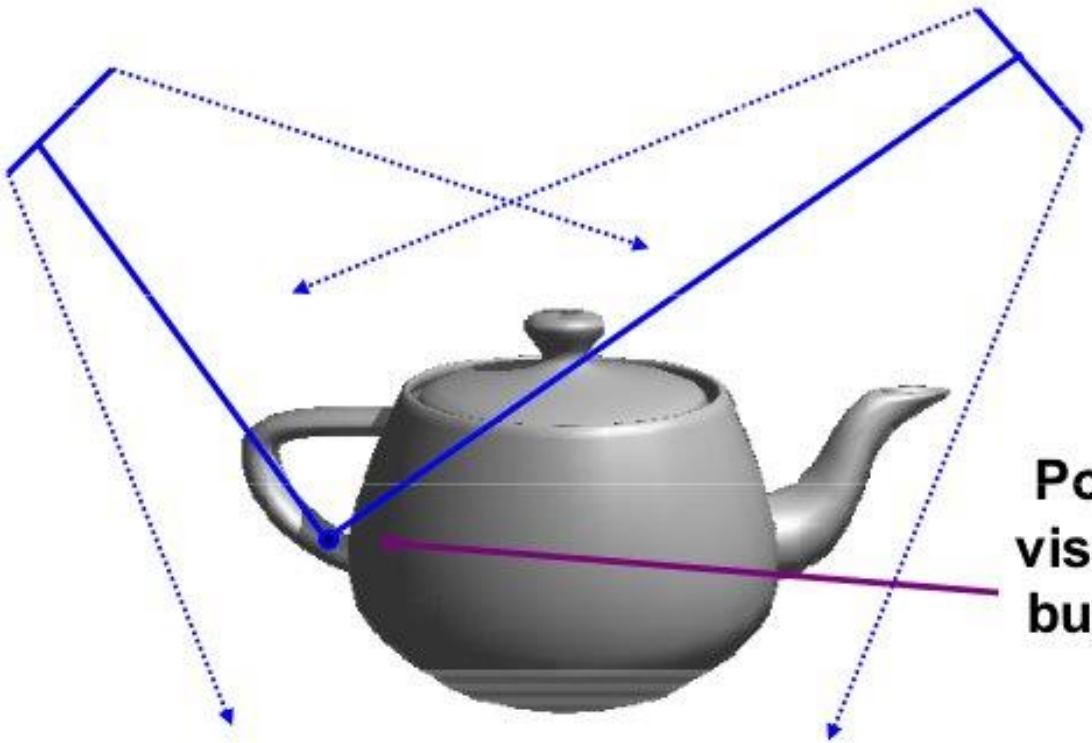
* This slide is from the slides of Prof. Sung-eui Yoon (KAIST)
https://sglab.kaist.ac.kr/~sungeui/CG/

Shadow Map

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

Eye

Light

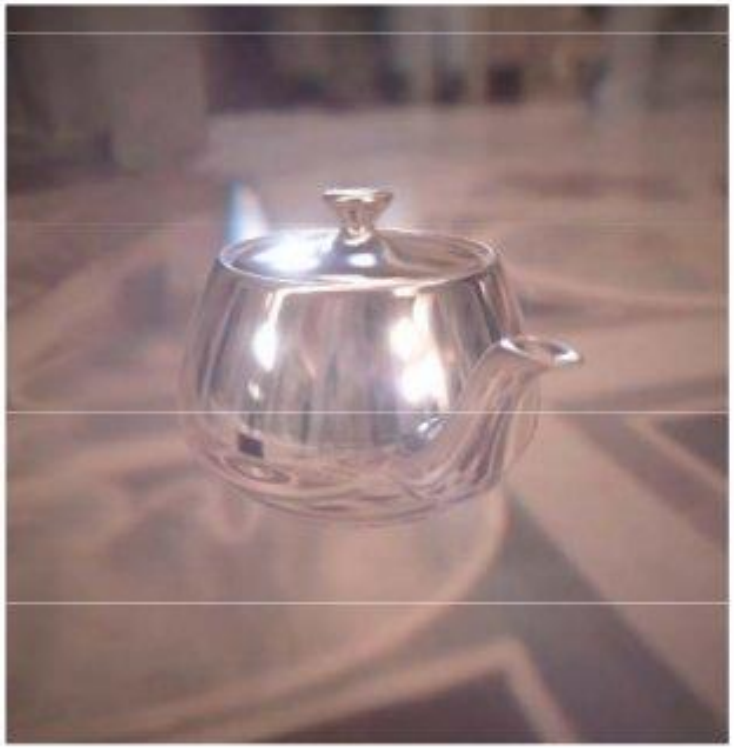


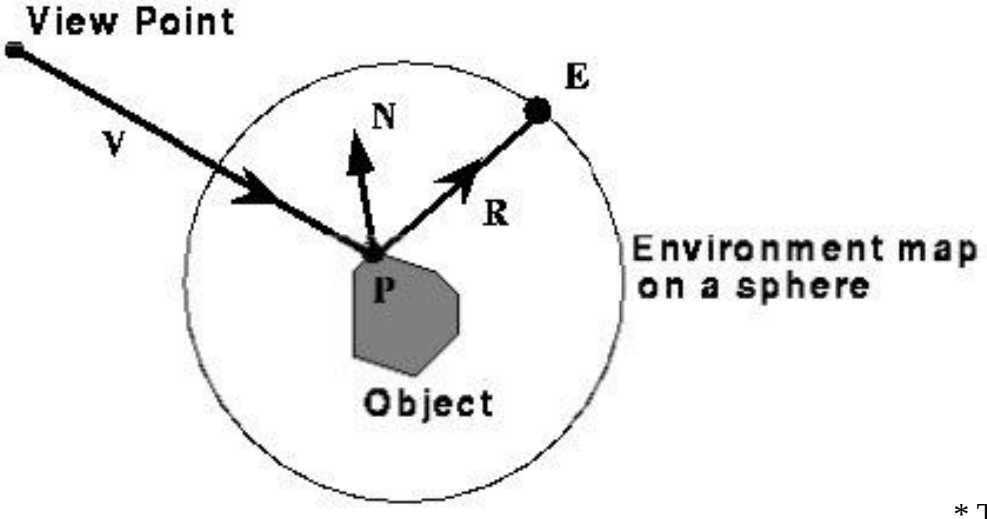
Point in shadow visible to the eye, but not visible to the light

* This slide is from the slides of Prof. Sung-eui Yoon (KAIST)
https://sglab.kaist.ac.kr/~sungeui/CG/

Environment Map

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





* This slide is from the slides of Prof. Sung-eui Yoon (KAIST)
https://sglab.kaist.ac.kr/~sungeui/CG/

[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/?q=refrac#webgl_materials_cubemap_refraction

Lab Session

- Now, let's start the lab today.