CSE 167:
Introduction to Computer Graphics
Lecture #18: Deferred Rendering

Jürgen P. Schulze, Ph.D.
University of California, San Diego
Spring Quarter 2015

Announcements

- 3rd final project blog due Sunday
- Final Project due Tuesday
  - Presentations start at 9am in CSE 1202
- TA Evaluation
- CAPE

Lecture Overview

- Deferred Rendering Techniques
  - Deferred Shading
  - Screen Space Ambient Occlusion
  - Bloom
  - Glow

Deferred Rendering

- Opposite to Forward Rendering, which is the way we have rendered with OpenGL so far
- Deferred rendering describes post-processing algorithms
  - Requires two-pass rendering
  - First pass:
    - Scene is rendered as usual by projecting 3D primitives to 2D screen space.
    - Additionally, an off-screen buffer (G-buffer) is populated with additional information about the geometry elements at every pixel
      - Examples: normals, diffuse shading color, position, texture coordinates
  - Second pass:
    - An algorithm, typically implemented as a shader, processes the G-buffer to generate the final image in the back buffer

Lecture Overview

- Deferred Rendering Techniques
  - Deferred Shading
  - Screen Space Ambient Occlusion
  - Bloom
  - Glow
- The Future of Computer Graphics

Deferred Shading

- Postpones shading calculations for a fragment until its visibility is completely determined
  - Only fragments that really contribute to the image are shaded
- Algorithm:
  - Fill a set of buffers with common data, such as diffuse texture, normals, material properties
  - For the lighting just render the light extents and fetch data from these buffers for the lighting computation
- Advantages:
  - Decouples lighting from geometry
  - Several lights can be applied with a single draw call: more than 1000 light sources can be rendered at 60 fps
- Disadvantages:
  - Consumes more memory, bandwidth and shader instructions than traditional rendering

Particle system with glowing particles. Source: Humus 3D

Reference

- Deferred Shading Tutorial:
  - http://gamedevs.org/uploads/deferred-shading-tutorial.pdf

Lecture Overview

- Deferred Rendering Techniques
  - Deferred Shading
  - Screen Space Ambient Occlusion
  - Bloom
  - Glow
- The Future of Computer Graphics

Screen Space Ambient Occlusion

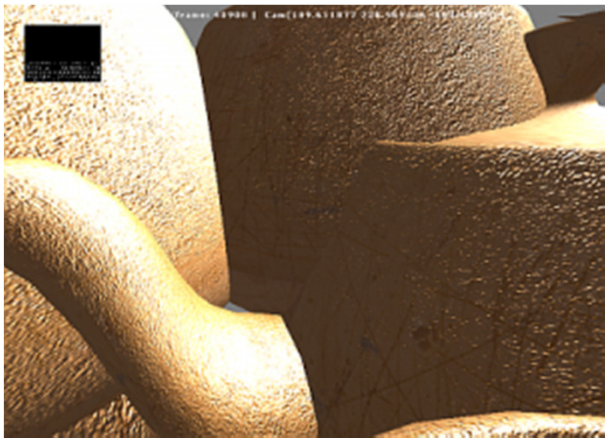
- Screen Space Ambient Occlusion is abbreviated as SSAO
- "Screen Space" refers to this being a deferred rendering approach
- Rendering technique for approximating ambient occlusion in real time
- Developed by Vladimir Kajalin while working at Crytek
- First use in 2007 PC game Crysis

Ambient Occlusion

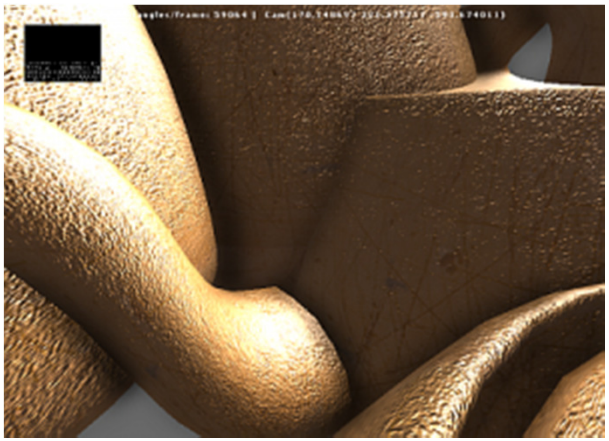
- Attempts to approximate global illumination
  - Very crude approximation
- Unlike local methods like Phong shading, ambient occlusion is a global method
  - Illumination at each point is a function of other geometry in the scene
- Appearance achieved by ambient occlusion is similar to the way an object appears on an overcast day
  - Example: arm pit is hit by a lot less light than top of head
- In the industry, ambient occlusion is often referred to as "sky light"

SSAO With Normals

- First pass:
  - Render scene normally and copy z values to g-buffer's alpha channel and scene normals to g-buffer's RGB channels
- Second pass:
  - Use normals and z-values to compute occlusion between current pixel and several samples around that pixel



No SSAO



With SSAO

References

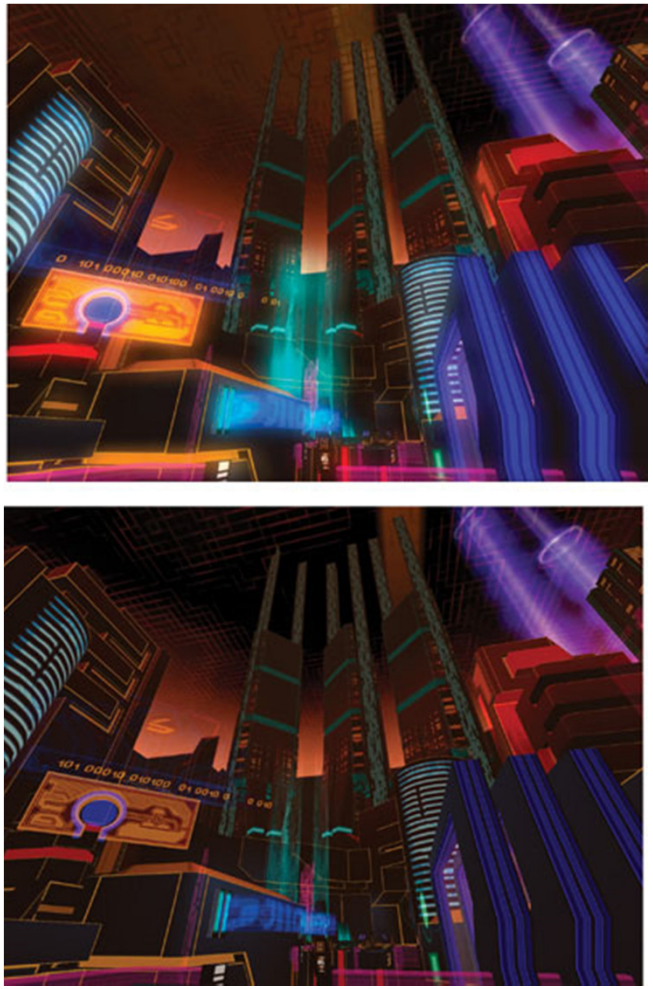
- Nvidia's documentation:
  - http://developer.download.nvidia.com/SDK/10.5/direct3d/Source/ScreenSpaceAO/doc/ScreenSpaceAO.pdf
- SSAO shader code from Crysis:
  - http://69.163.227.177/forum.php?mod=viewthread&tid=772
- Another implementation:
  - http://www.gamerendering.com/2009/01/14/ssao/

Lecture Overview

- Deferred Rendering Techniques
  - Deferred Shading
  - Screen Space Ambient Occlusion
  - Glow
  - Bump Mapping
- The Future of Computer Graphics

Glow Effects

- Glows and halos of light appear everywhere in the world
- They provide powerful visual cues about brightness and atmosphere
- In computer graphics, the intensity of light reaching the eye is limited, so the only way to distinguish intense sources of light is by their surrounding glow and halos
- In everyday life, glows and halos are caused by light scattering in the atmosphere or within our eyes



A cityscape with and without glow.
Source: GPU Gems

Glow Shader

- Render passes:
  - Render entire scene to the back buffer
  - Render only glowing objects to a smaller off-screen glow buffer
  - Apply a Gaussian blur filter to glow buffer
  - Compose back buffer and glow buffer together
- Glow example:
  - https://www.youtube.com/watch?v=kDOFM9Rj5dY

References

- GPU Gems Chapter on Glow
  - http://http.developer.nvidia.com/GPUGems/gpugems_ch21.html
- GLSL Shader for Gaussian Blur
  - http://www.ozone3d.net/tutorials/image_filtering_p2.php

Lecture Overview

- Deferred Rendering Techniques
  - Deferred Shading
  - Screen Space Ambient Occlusion
  - Glow
  - Bump Mapping
- The Future of Computer Graphics

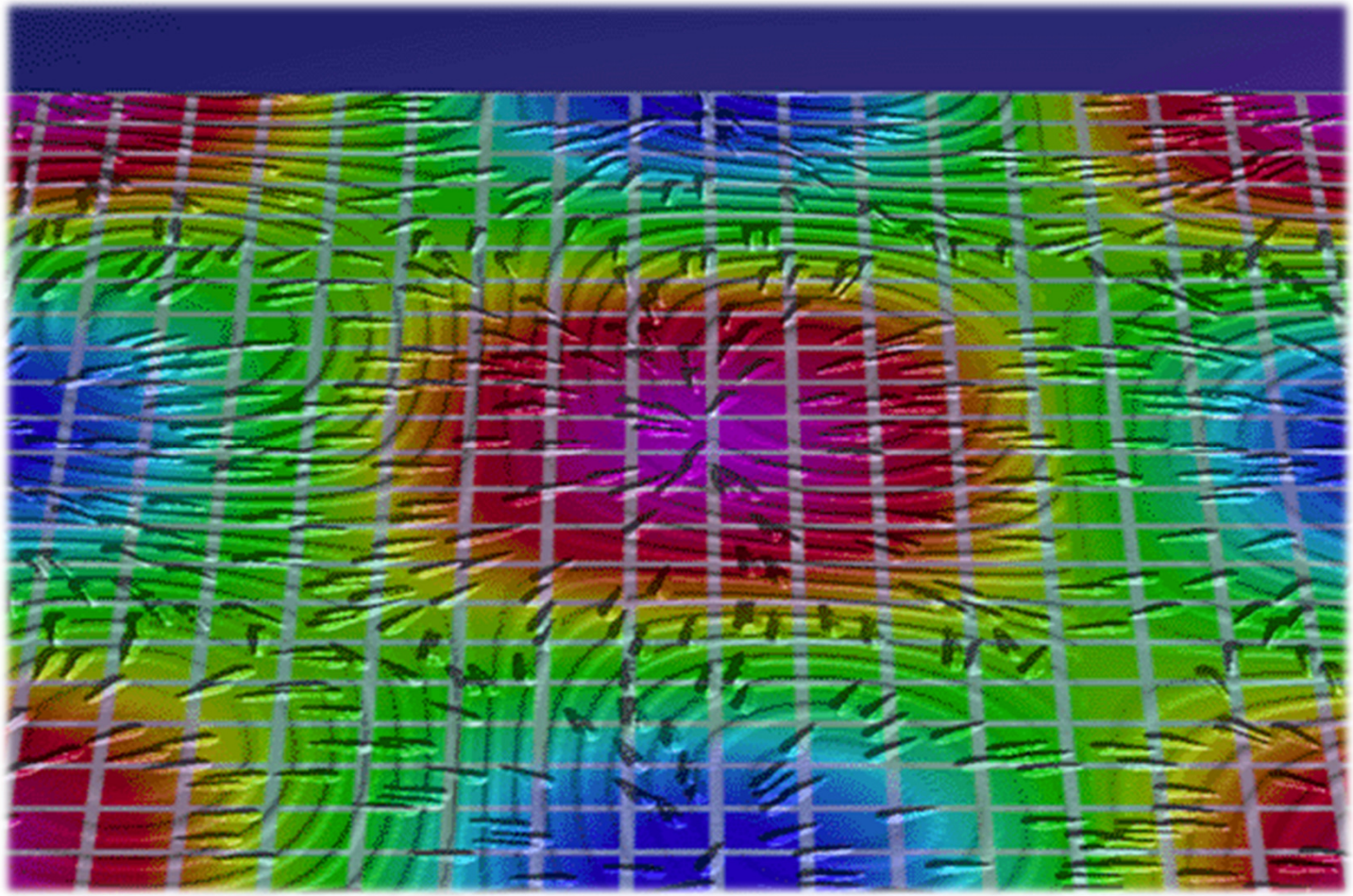
Bump Mapping

- Many textures are the result of small perturbations in the surface geometry
- Modeling these changes would result in an explosion in the number of geometric primitives.
- Bump mapping attempts to alter the lighting across a polygon to provide the illusion of texture.

[This chapter includes slides by Roger Crawfis]

UCSD

Bump Mapping Example

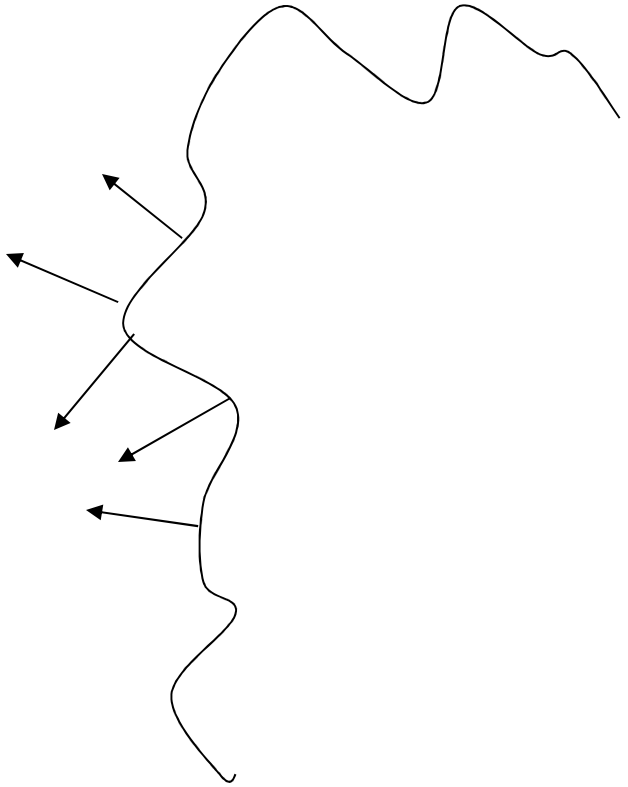


Crawfis 1991

UCSD

Bump Mapping

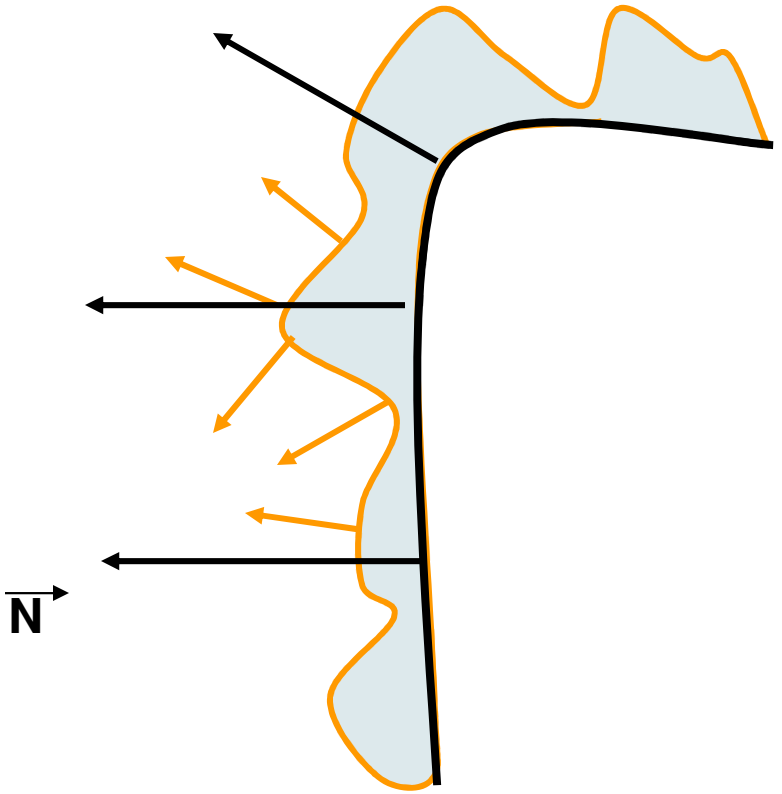
- Consider the lighting for a modeled surface.



UCSD

Bump Mapping

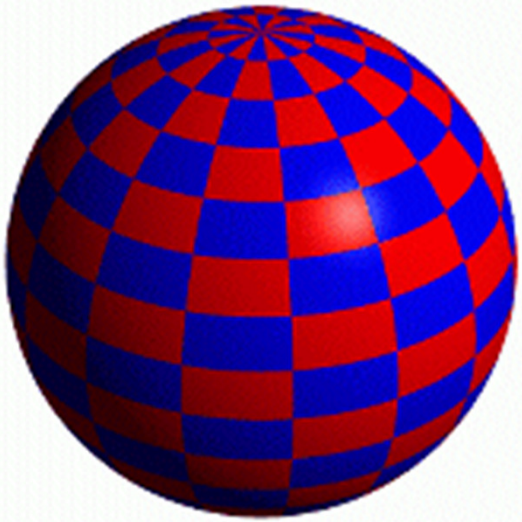
- We can model this as deviations from some base surface.
- The question is then how these deviations change the lighting.



UCSD

Bump Mapping

- Store in a texture and use textures to alter the surface normal
  - Does not change the shape of the surface
  - Just shaded as if it were a different shape



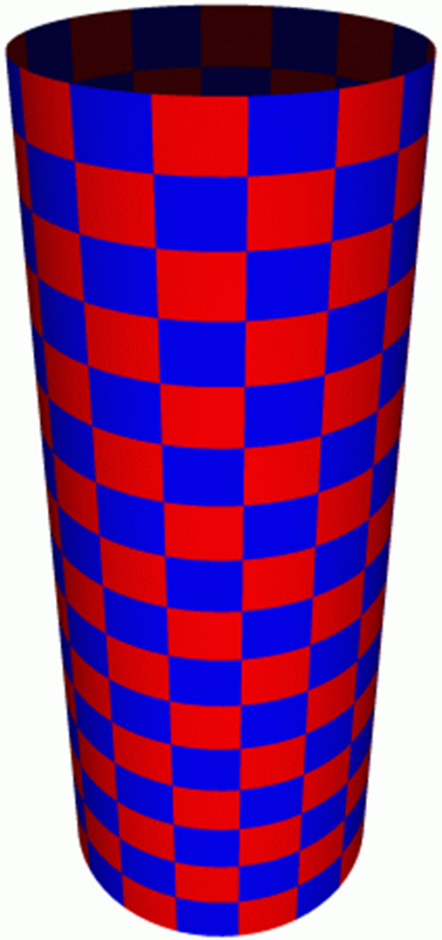
Sphere w/Diffuse Texture

Swirly Bump Map

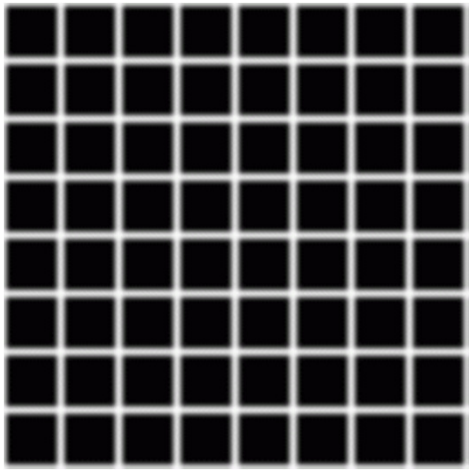
Sphere w/Diffuse Texture & Bump Map

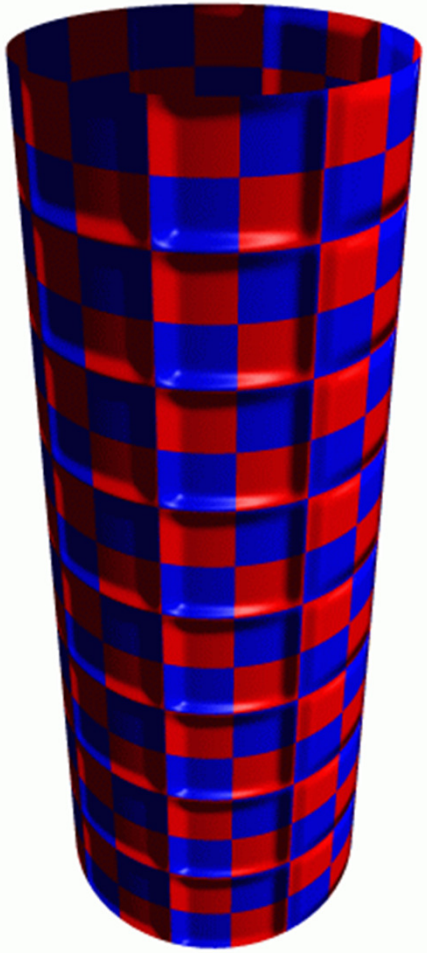
UCSD

Simple textures work great



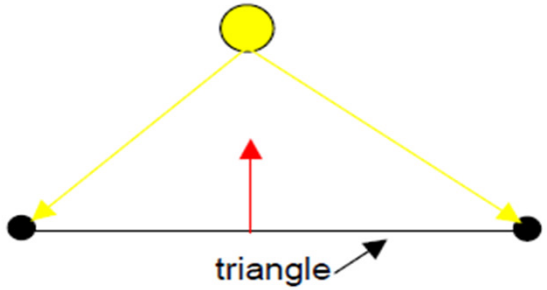
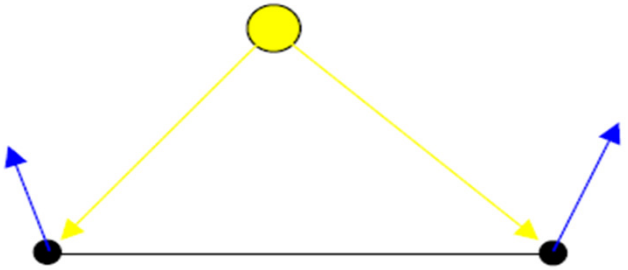
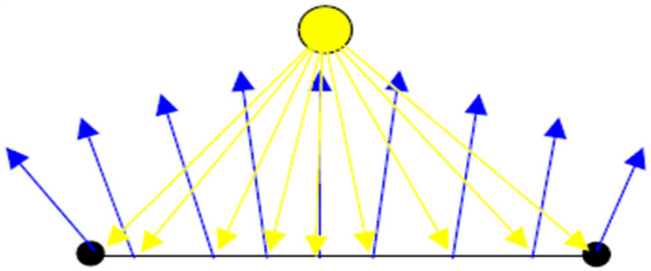
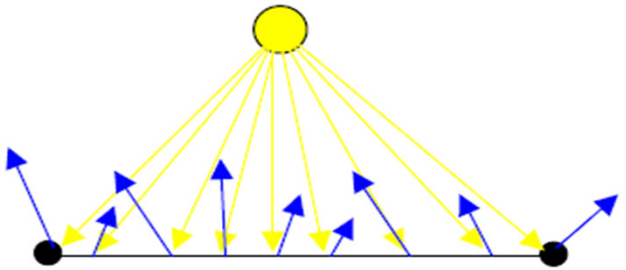
Cylinder w/Diffuse Texture Map





Cylinder w/Texture Map & Bump Map

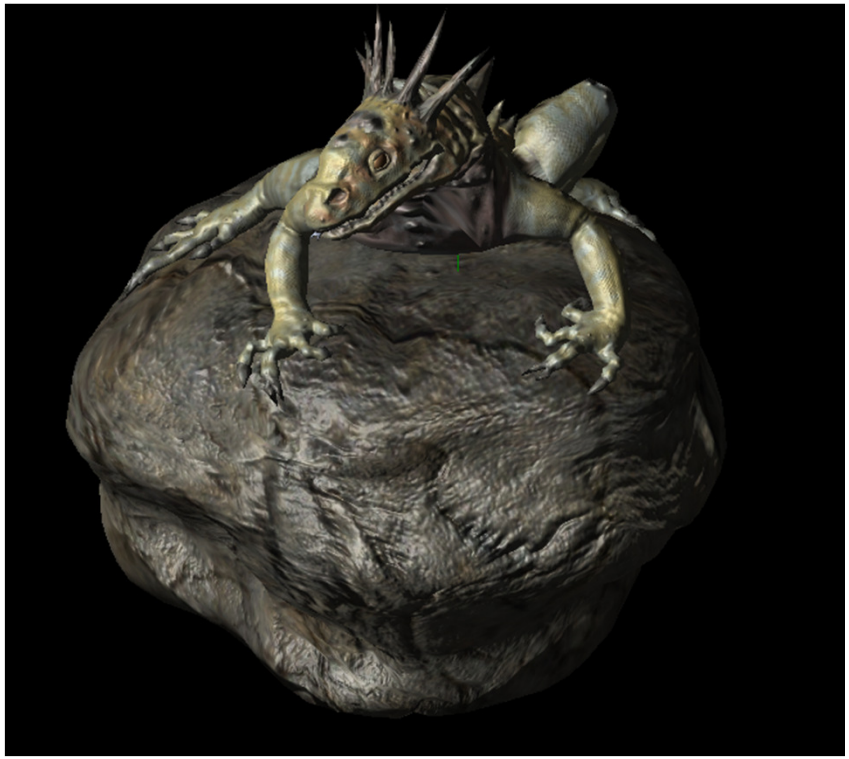
Normal Mapping

Flat shading	Goraud shading
	
Only the first normal of the triangle is used to compute lighting in the entire triangle.	The light intensity is computed at each vertex and interpolated across the surface.
Phong shading	**Bump mapping**
	
Normals are interpolated across the surface, and the light is computed at each fragment.	Normals are stored in a bumpmap texture, and used instead of Phong normals.

UCSD

Normal Mapping

Just texture mapped

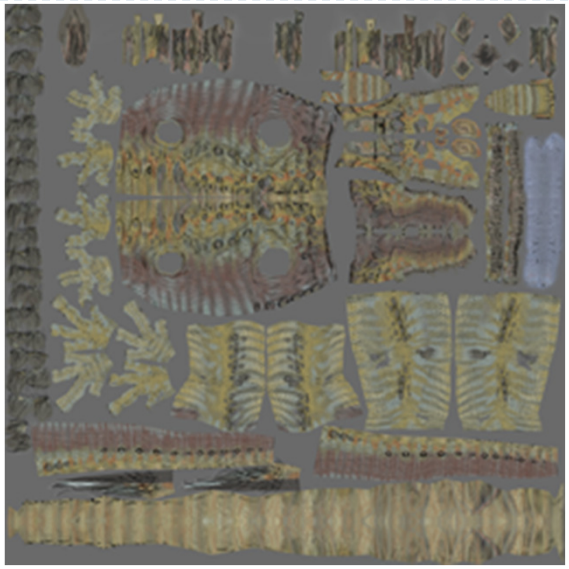


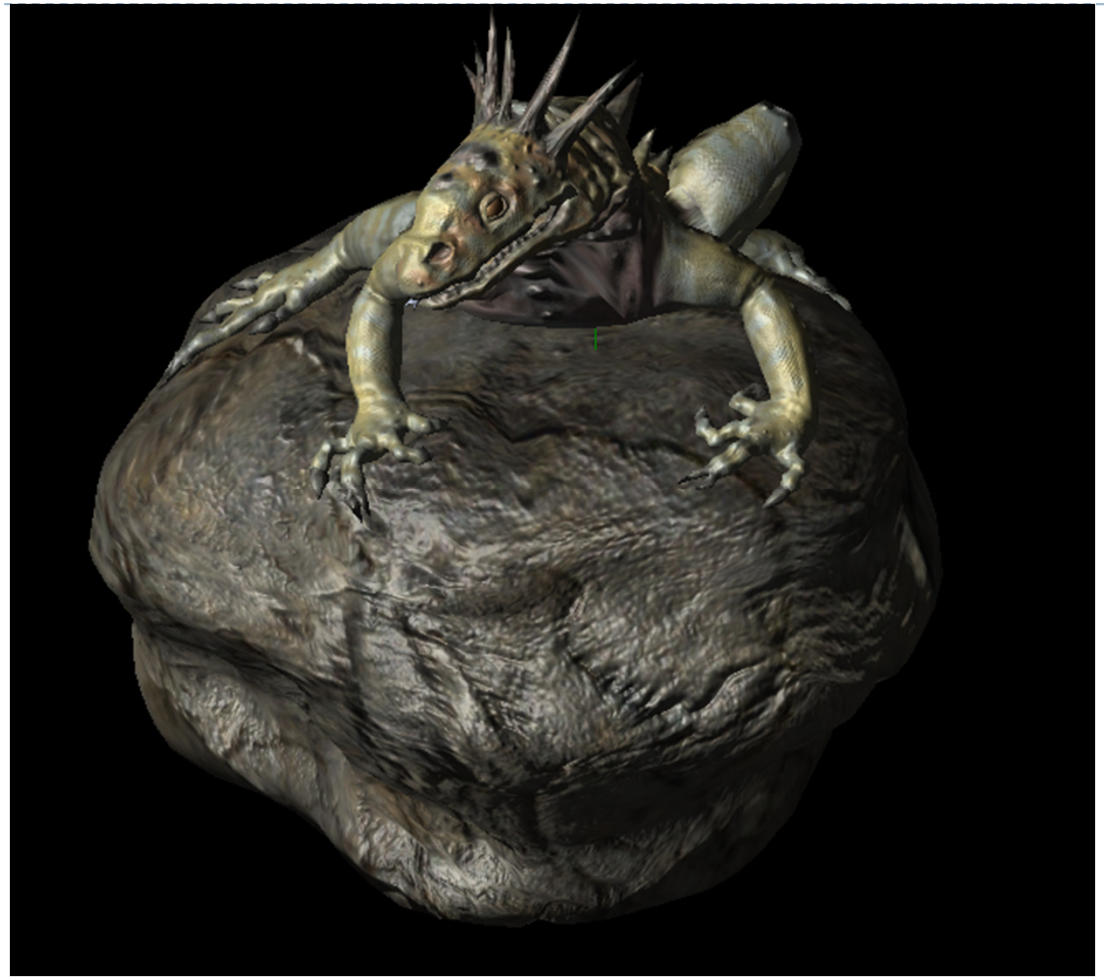
Texture and normal maps

Notice: The geometry is unchanged. There's the same number of vertices and triangles. This effect is entirely from the normal map.

UCSD

Normal Maps

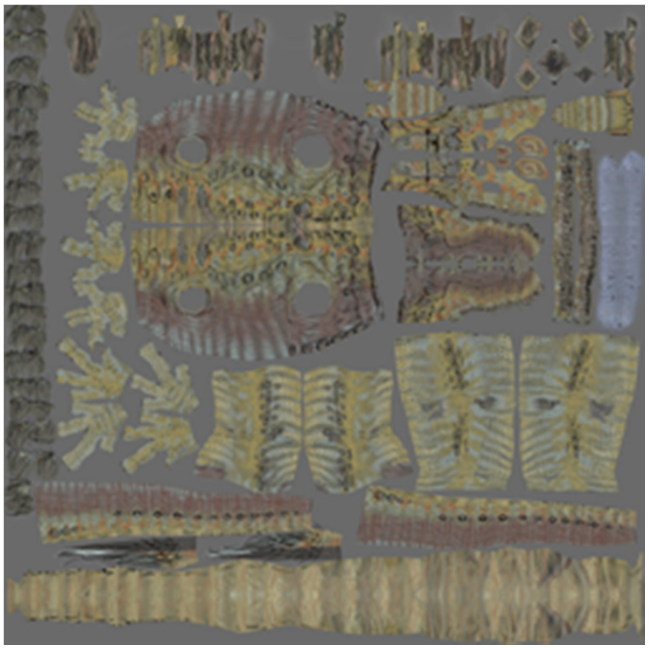




Store the normal directly in the texture.

UCSD

Normal Maps



Diffuse Color Texture Map

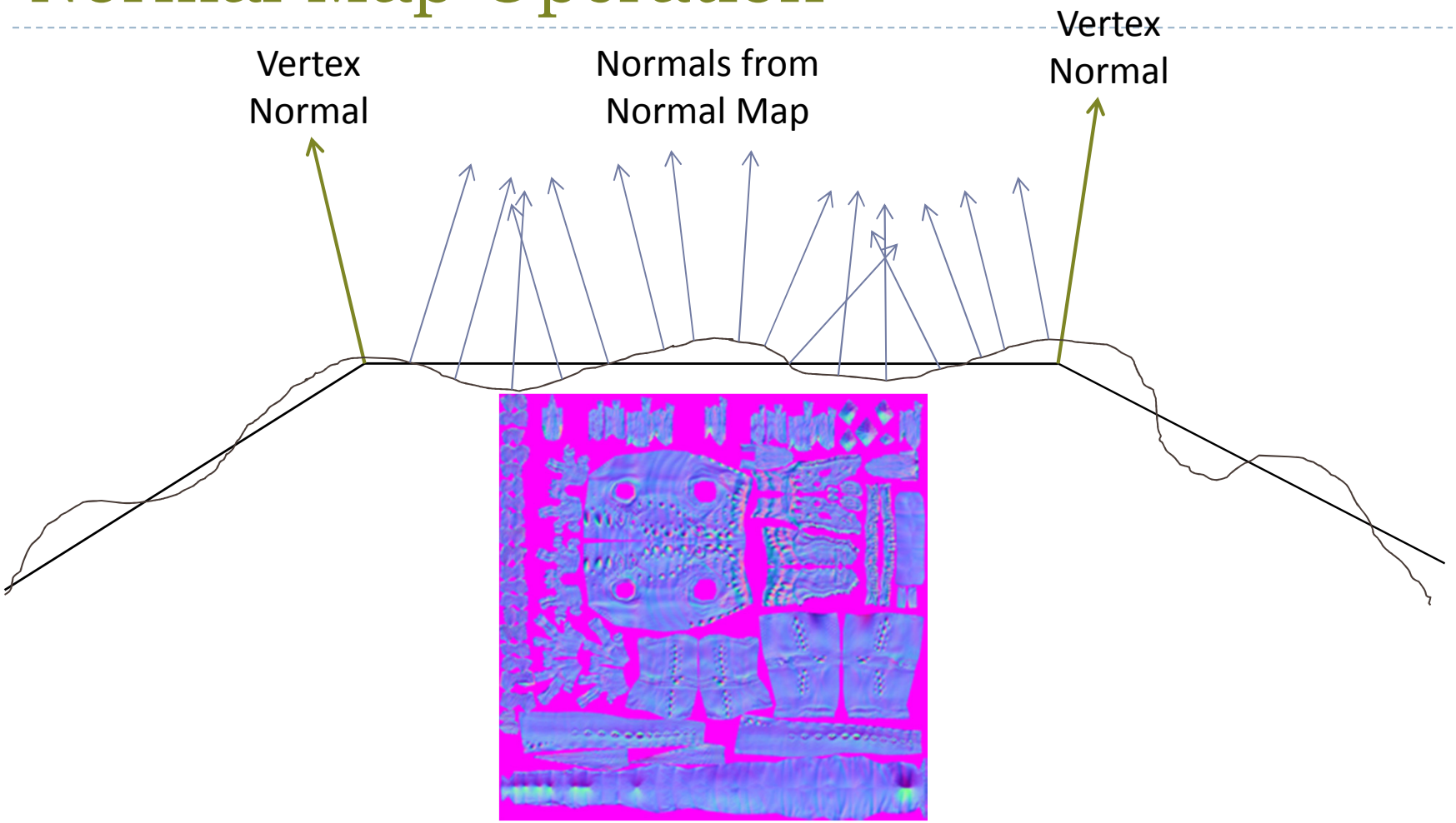
Normal Map

Each pixel RGB values is really a normal vector relative to the surface at that point.

-1 to 1 range is mapped to 0 to 1 for the texture so normals become colors.

UCSD

Normal Map Operation



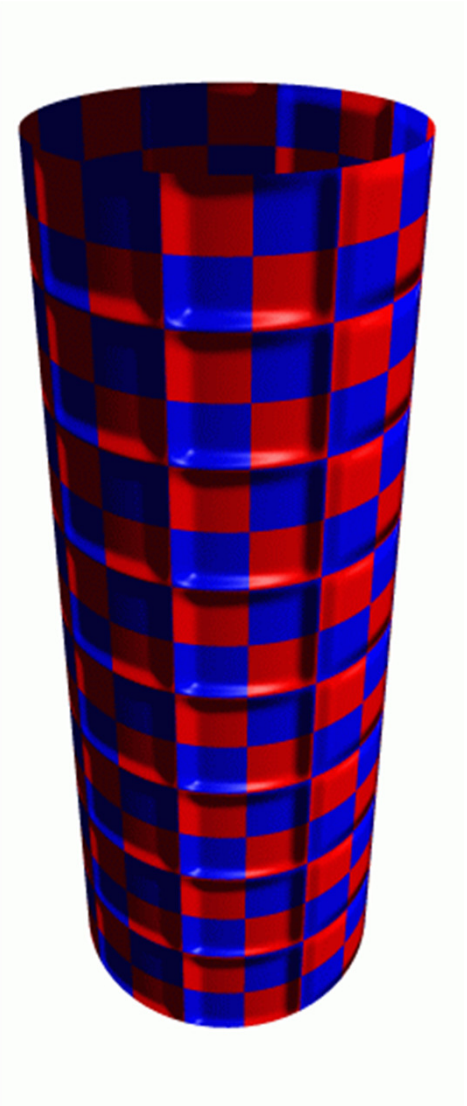
For each pixel, determine the normal from a texture image. Use that to compute the color.

UCSD

What's Missing?

- There are no bumps on the silhouette of a bump or normal-mapped object

→ Displacement Mapping



UCSD

Lecture Overview

- Deferred Rendering Techniques
  - Deferred Shading
  - Screen Space Ambient Occlusion
  - Glow
  - Bump Mapping
- The Future of Computer Graphics

The Future of Computer Graphics

- ACM SIGGRAPH 2015 Technical Papers (3:20)
  - https://www.youtube.com/watch?v=XrYkEhs2FdA
- SIGGRAPH 2015 - Computer Animation Festival Trailer (3:28)
  - https://www.youtube.com/watch?v=UH-mdAdT1BI
- Cryengine on Steam (2:41)
  - http://store.steampowered.com/app/220980/
- Top 5 Best Next Gen Game Engines Of The Future (12:16)
  - https://www.youtube.com/watch?v=vTF7wz0-AQs
- The Centrifuge Brain Project, 2013 (6:35)
  - https://www.youtube.com/watch?v=RVeHxUVkW4w

Good luck with your final projects!