Texture Mapping

Brian Curless
CSE 557
Fall 2015

Reading

<u>Required</u>

- Shirley, 11.1-11.2, 11.4-11.6

<u>Recommended</u>

- Paul S. Heckbert. Survey of texture mapping. **IEEE Computer Graphics and Applications** 6(11): 56--67, November 1986.

<u>Optional</u>

- Woo, Neider, & Davis, Chapter 9
- James F. Blinn and Martin E. Newell. Texture and reflection in computer generated images. **Communications of the ACM** 19(10): 542--547, October 1976.

Texture mapping

*Texture mapping (Woo et al., fig. 9-1)*

Texture mapping allows you to take a simple polygon and give it the appearance of something much more complex.

- ◆ Due to Ed Catmull, PhD thesis, 1974
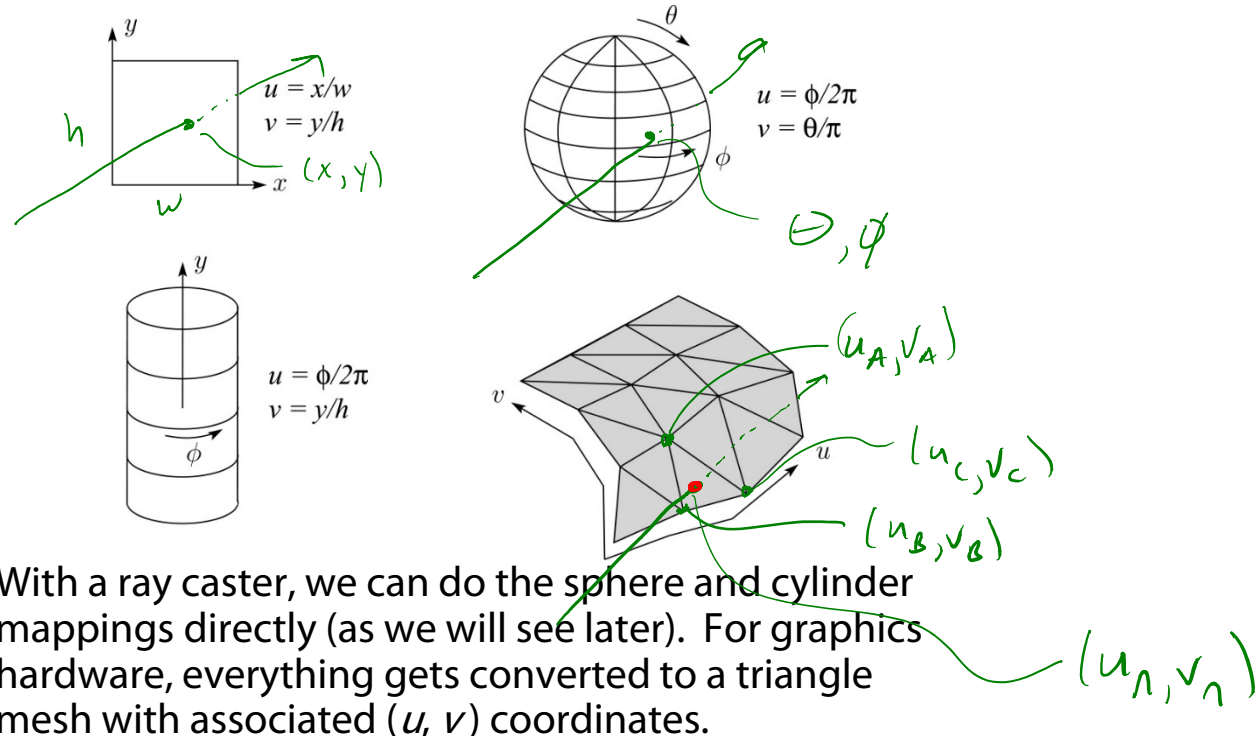- ◆ Refined by Blinn & Newell, 1976

A texture can modulate just about any parameter – diffuse color, specular color, specular exponent, …

Implementing texture mapping

A texture lives in it own abstract image coordinates paramaterized by (u, v) in the range ([0..1], [0..1]):
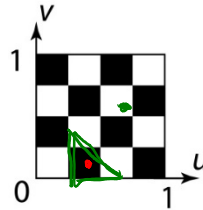
It can be wrapped around many different surfaces:

$u = x/w$
$v = y/h$

$u = \phi/2\pi$
$v = \theta/\pi$

θ, ϕ

(u_A, v_A)

$u = \phi/2\pi$
$v = y/h$

(u_C, v_C)

(u_B, v_B)

With a ray caster, we can do the sphere and cylinder mappings directly (as we will see later). For graphics hardware, everything gets converted to a triangle mesh with associated (u, v) coordinates.

(u_n, v_n)

Note: if the surface moves/deforms, the texture goes with it.

4

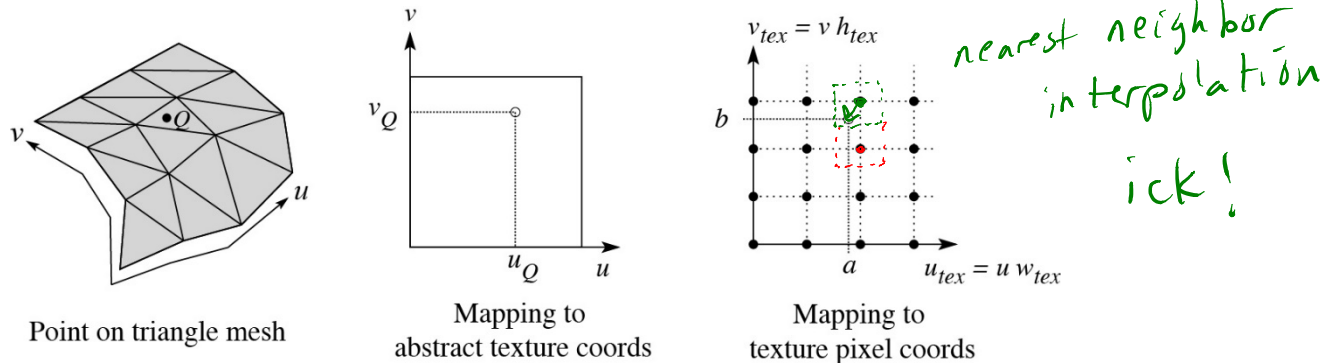## Mapping to texture image coords

The texture is usually stored as an image. Thus, we
need to convert from abstract texture coordinate:

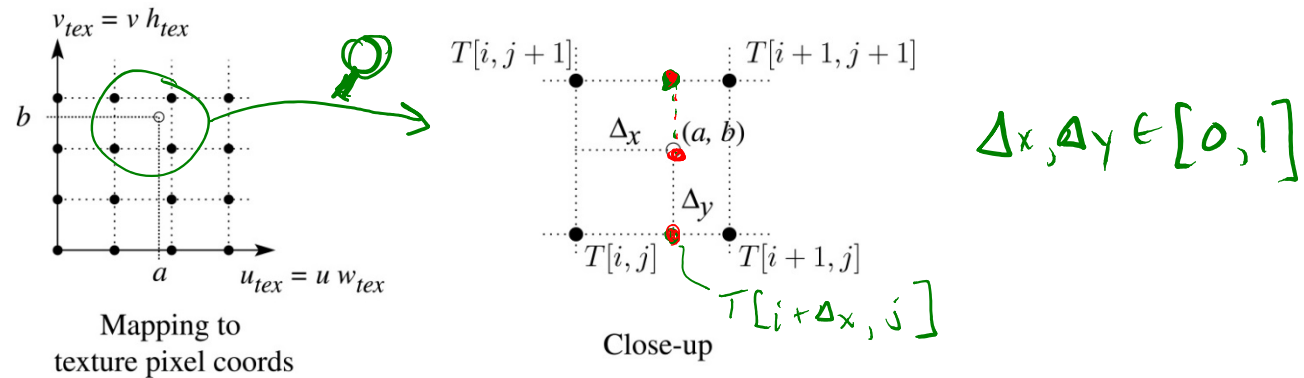(u, v) in the range $([0..1], [0..1])$

to texture image coordinates:

(u_{tex}, v_{tex}) in the range $([0.. w_{tex}], [0.. h_{tex}])$

Point on triangle mesh

Mapping to
abstract texture coords

Mapping to
texture pixel coords

nearest neighbor
interpolation

ick!

Q: What do you do when the texture sample you
need lands between texture pixels?

Texture resampling

We need to resample the texture:

$v_{tex} = v\, h_{tex}$

b

a

$u_{tex} = u\, w_{tex}$

Mapping to
texture pixel coords

$T[i, j+1]$ $T[i+1, j+1]$

Δ_x (a, b)

Δ_y

$T[i, j]$ $T[i+1, j]$

Close-up

$\Delta x, \Delta y \in [0, 1]$

$T[i + \Delta x, j]$

Thus, we seek to solve for: $T(a,b) = T\left(i + \Delta_x, j + \Delta_y\right)$

A common choice is **bilinear interpolation**:

$$T\left(i + \Delta_x, j\right) = \underline{(1 - \Delta_x)}\, T[i, j] \quad + \quad \underline{\Delta_x}\, T[i+1, j]$$

$$\rightarrow \quad T\left(i + \Delta_x, j+1\right) = \underline{(1 - \Delta_x)}\, T[i, j+1] \; + \; \underline{\Delta_x}\, T[i+1, j+1]$$

$$T\left(i + \Delta_x, j + \Delta_y\right) = \underline{(1 - \Delta_y)}\, T\left(i + \Delta_x, j\right) \; + \; \underline{\Delta_y}\, T\left(i + \Delta_x, j+1\right)$$

$$= \underline{(1 - \Delta_x)(1 - \Delta_y)}\, T[i, j] \quad + \quad \underline{\Delta_x(1 - \Delta_y)}\, T[i+1, j] \; +$$

$$\underline{(1 - \Delta_x)\Delta_y}\, T[i, j+1] \; + \; \underline{\Delta_x \Delta_y}\, T[i+1, j+1]$$

6

Displacement mapping

Textures can be used for more than just color.

In **displacement mapping**, a texture is used to perturb the surface geometry itself. Here's the idea in 2D:


$$\tilde{\mathbf{Q}}(u) = \mathbf{Q}(u) + d(u)\mathbf{N}(u)$$
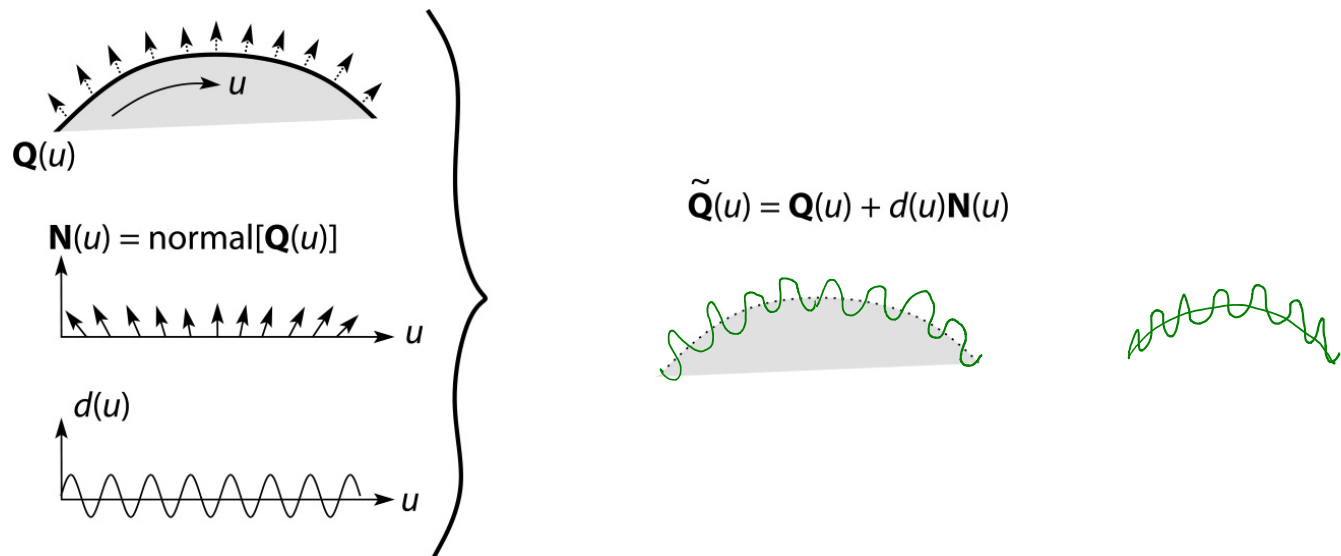
- ◆ These displacements "animate" with the surface
- ◆ In 3D, you would of course have (u, v) parameters instead of just u.

Suppose **Q** is a simple surface, like a cube. Will it take more work to render the modified surface $\tilde{\mathbf{Q}}$? *Yes*

Bump mapping

In **bump mapping**, a texture is used to perturb the normal:

- Use the original, simpler geometry, $\mathbf{Q}(u)$, for hidden surfaces
- Use the normal from the displacement map for shading:

$$\tilde{\mathbf{N}} = \text{normal}[\tilde{\mathbf{Q}}(u)]$$
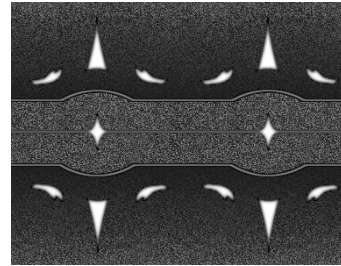

$\mathbf{Q}(u)$
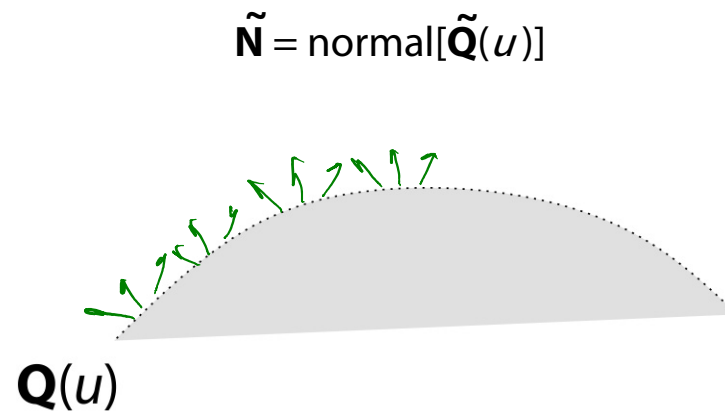
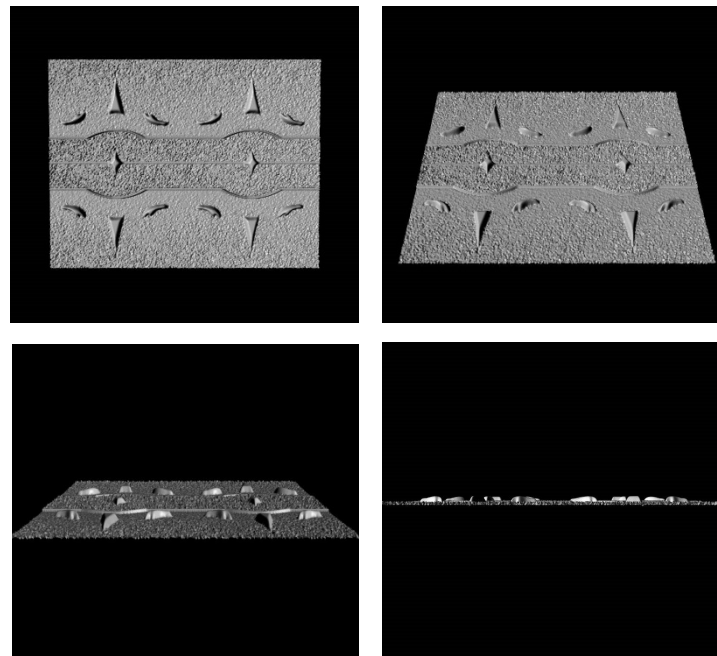What artifacts in the images would reveal that bump mapping is fake?

silhouette   shadows

Displacement vs. bump mapping

Input texture

Rendered as displacement map over a rectangular surface

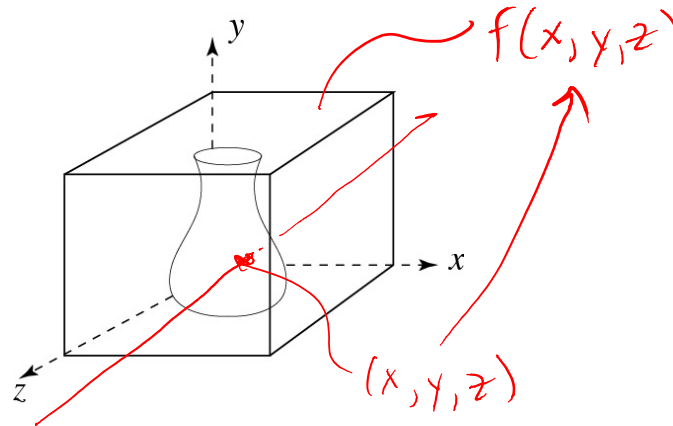# Displacement vs. bump mapping (cont'd)

Original rendering

Rendering with bump map
wrapped around a cylinder

Bump map and rendering by Wyvern Aldinger

Solid textures

Q: What kinds of artifacts might you see from using a marble veneer instead of real marble?

One solution is to use **solid textures**:

- Use model-space coordinates to index into a 3D texture
- Like "carving" the object from the material

One difficulty of solid texturing is coming up with the textures.

Solid textures (cont'd)

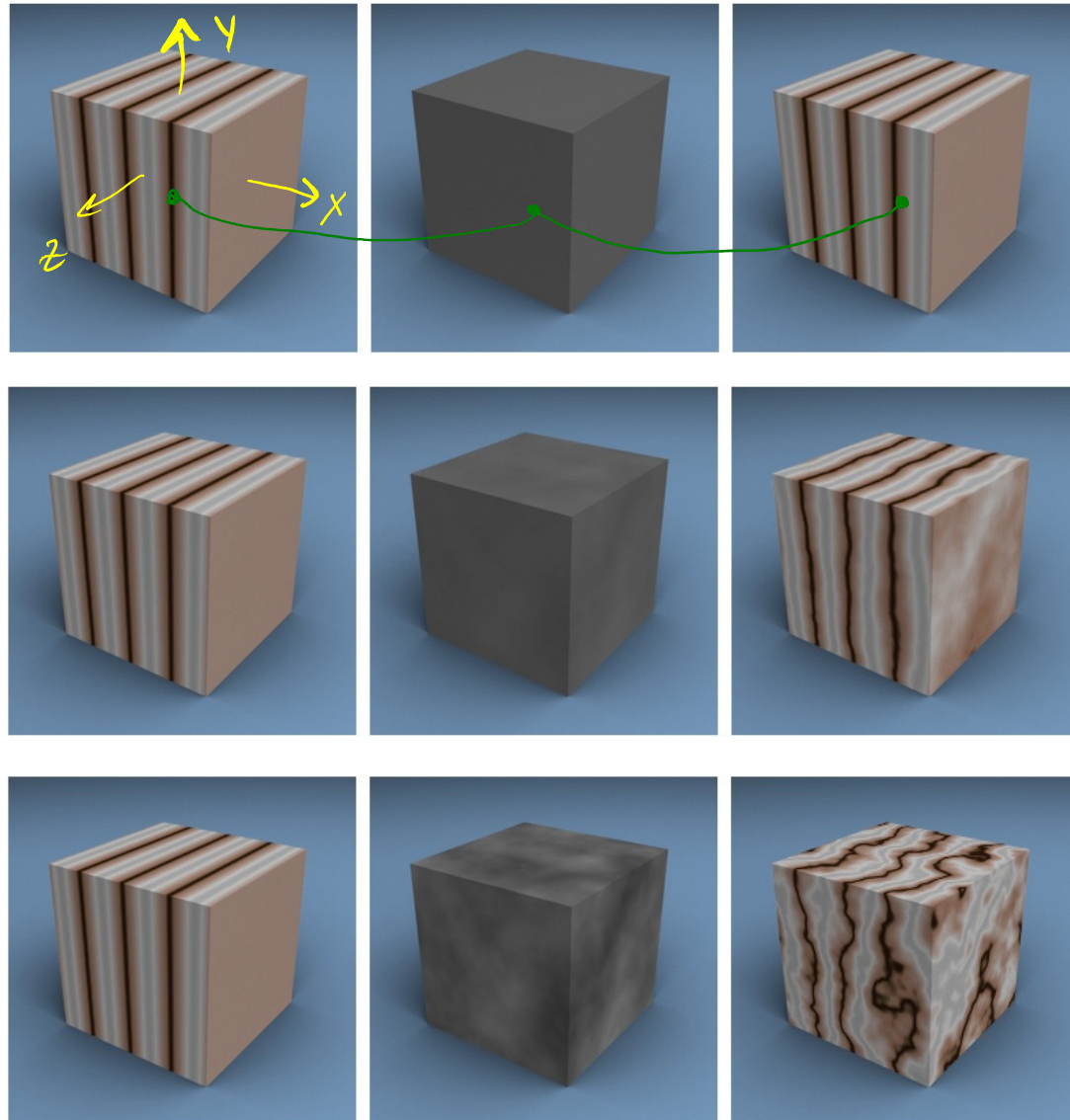Here's an example for a vase cut from a solid marble texture:

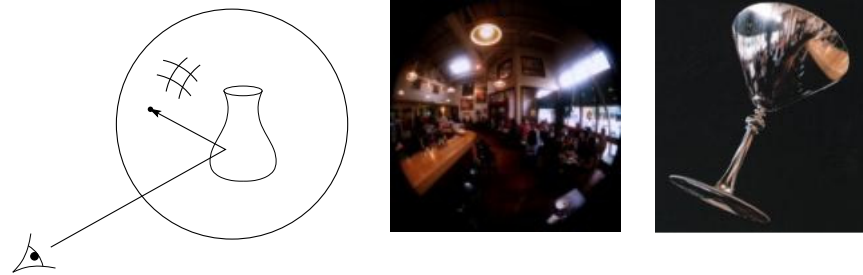*Solid marble texture by Ken Perlin, (Foley, IV-21)*

Solid textures (cont'd)

in$(x, y, z) =$
stripes(x)

shift$(x, y, z) =$
K•noise(x, y, z)

out$(x, y, z) =$
stripes$(x +$ shift$(x, y, z))$

Increasing K

Environment mapping

In **environment mapping** (also known as **reflection mapping**), a texture is used to model an object's environment:

- ◆ Rays are bounced off objects into environment
- ◆ Color of the environment used to determine color of the illumination
- ◆ Environment mapping works well when there is just a single object – or in conjunction with ray tracing

This can be readily implemented (without interreflection) in graphics hardware, where the texture is stored in a "cube map" instead of a sphere.

With a ray tracer, the concept is easily extended to handle refraction as well as reflection (and interreflection).

14