

Simplistic Approach

- Very finely tessellate the surface
- Determine a color for each patch
- Combined patch effects form a pattern

2

- Problems:
 - Lots of work
 - May not illuminate well

Texture/Pattern Mapping

- Describe a grid of color values
 - In texture space (r,s)
- Texture scanning
 - Texture space → object space → pixels
- Pixel-Order scanning
 - Pixels → object space → texture space

Texture Scanning Multi-step process Generate the texture pattern Choose how it will be "mixed"

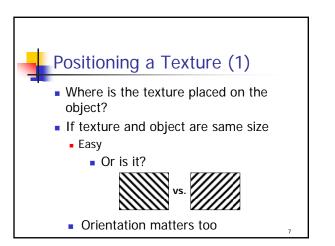
- 3. Position relative to the object face/image fragment
- 4. Apply to the fragment

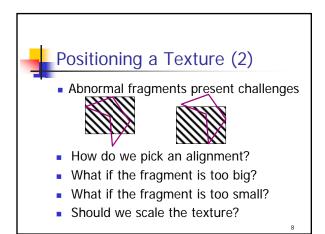
Specifying a Texture Typically a rectangular grid of values RGB or RGBA most common Color index has very poor applicability Size is restricted Powers of 2 are common Minimum size is reasonable

Often stored in a file



5





Positioning a Texture (3)

Repeating

- Should we tile with copies? Both directions?
- Do matching edges truly match?
- Clamp/Crop
 - Only display what overlaps

Positioning a Texture (4)

- Adjacent surface patches
 - How do the borders match?
 - Will it be OK for a "curved" surface?
- Possible solution
 - Small, tiled texture
 - Will this be too "busy"?

Viewing Distance Effects (1)

- Close range
 - Is the texture detailed enough?
- Will it look patchy?
- Long range
 - As the texture "shrinks," will it shimmer?

Viewing Distance Effects (2) Variations in pixel and texel size Specify texels, but must draw pixels Filtering strategy Nearest – use closest texel to final pixel Linear – weighted average of nearest texels Other – more complex interpolation (bicubic, ...)

12

10

11

