Computer Graphics

Lecture 5:

Texture mapping

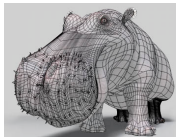
The Problem:

We don't want to represent all this detail with geometry

The Solution: Textures

- The visual appearance of a graphics scene can be greatly enhanced by the use of texture.
- Consider a brick building, using a polygon for every brick require a huge effort in scene design.
- So why not use one polygon and draw a repeating brick pattern (*texture*) onto it?

The Quest for Visual Realism



Texture Definition

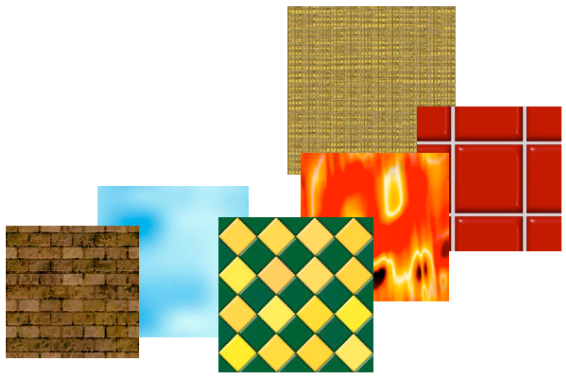
- Textures may be defined as:
  - – Pixmaps - Arrays containing the actual pixel values to be mapped to the polygon, e.g. photos.
  - – Procedures - Suitable for repeating patterns.

Procedural textures

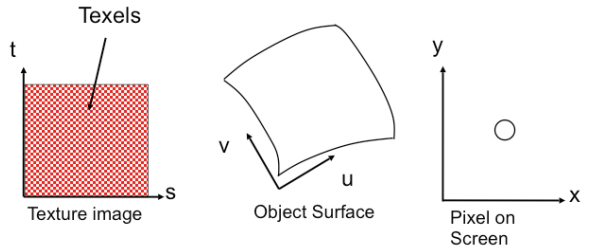
- Write a function: $F(\mathbf{p}) \rightarrow$ color

- non-intuitive
- difficult to match existing texture

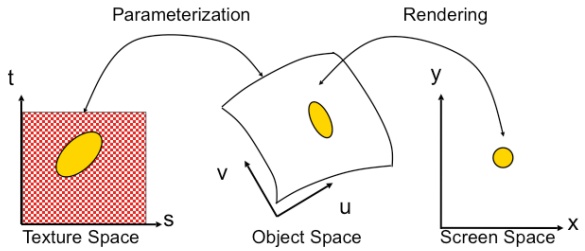
Photo textures



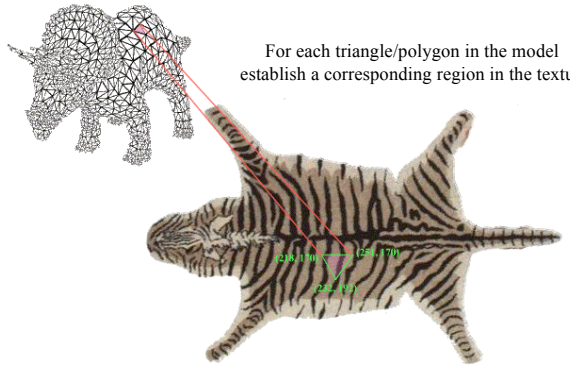
The concept of texture mapping



Texture mapping: Terminology



The concept of texture mapping

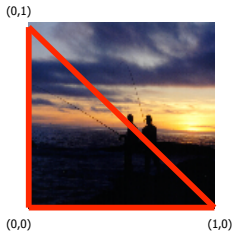


For each triangle/polygon in the model establish a corresponding region in the texture

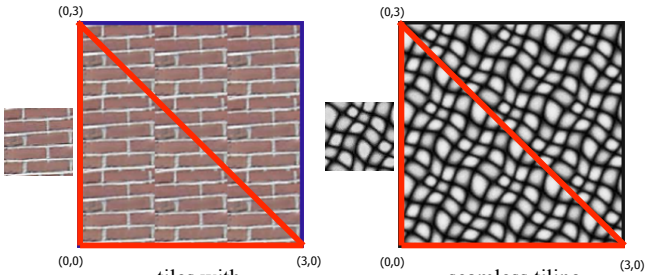
During rasterization interpolate the coordinate indices into the texture map

Texture Coordinates

- Specify a texture coordinate (*s, t*) at each vertex
- Canonical texture coordinates (0,0) → (1,1)
- Often the texture size is a power of 2 (but it doesn't have to be)
- How can we tile this texture?



Tiling Texture



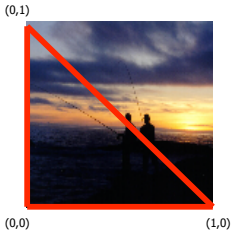
tiles with visible seams

seamless tiling (repeating)

Texture synthesis

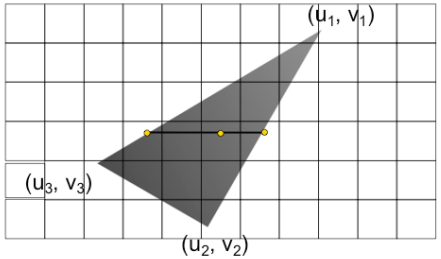
Texture coordinates

- Specify a texture coordinate (*s, t*) at each vertex
- Canonical texture coordinates (0,0) → (1,1)
- Linearly interpolate the values in screen space

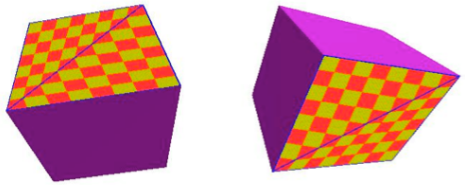


Mapping texture to individual pixels

- Interpolate texture coordinates across scanlines
- Same as Gouraud shading but now for texture coordinates not shading values



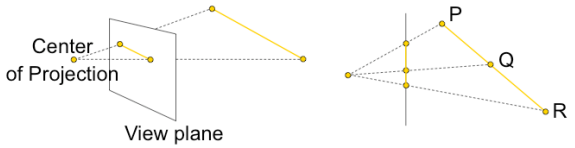
What Goes Wrong?



- Notice the distortion along the diagonal triangle edge of the cube face

Perspective projection

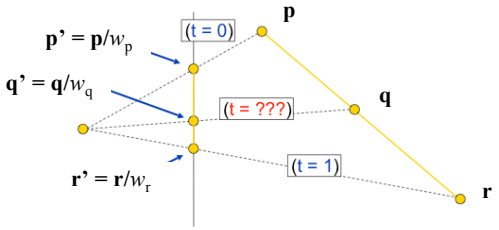
- The problem is that perspective projection does not preserve affine combinations of points!
- In particular, equal distances along a line in eye space do not map to equal distances in screen space



- Linear interpolation in screen space is not equal to linear interpolation in eye space!

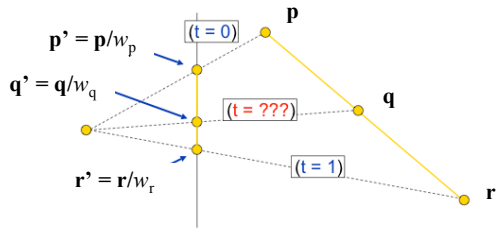
How to fix?

- Suppose we assign parameter t to vertices **p** and
- Suppose $t = 0$ at **p**, and $t = 1$ at **r**
- **p** projects to **p'** and **r** projects to **r'** (divide by w)
- What value should t have at location **q'**?



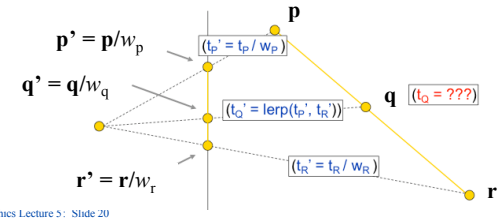
How to fix?

- We cannot linearly interpolate t between **r'** and **r**
- Only projected values can be linearly interpolated in screen space
- Solution: perspective-correct interpolation



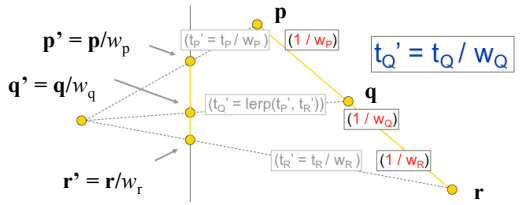
Perspective Correct Interpolation

- Linearly interpolate t / w (not t) between **p'** and **p'**.
  - Compute $t_{p'} = t_p / w_p$ and $t_{r'} = t_r / w_r$
  - Linearly interpolate (lerp) $t_{p'}$ and $t_{r'}$ to get $t_{q'}$ at location **q'**
- But, we want the (unprojected) parameter t_q, not $t_{q'}$



Perspective Correct Interpolation

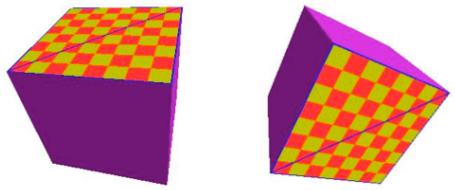
- The parameter $t_{q'}$ is related to t_q by a factor of $1/w$:
  - Lerp $1 / w_p$ and $1 / w_r$ to obtain $1 / w_q$ at point **q'**.
  - Divide $t_{q'}$ by $1 / w_q$ to get t_q



Perspective Correct Interpolation

- Summary:
  - Given parameter t at vertices:
    - Compute $1 / w$ for each vertex
    - Linearly interpolate $1 / w$ across the triangle
    - Linearly interpolate t / w across the triangle
    - Do perspective division:
      Divide t / w by $1 / w$ to obtain interpolated parameter t

Perspective Correct Interpolation



Mapping texture to individual pixels

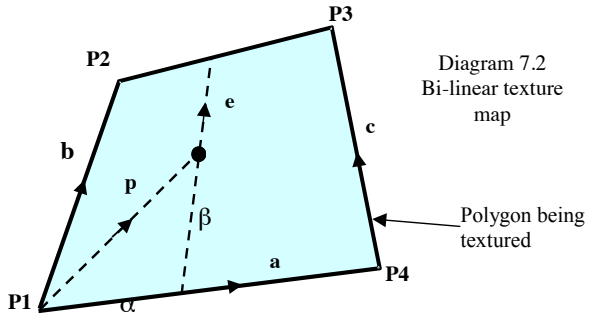


Diagram 7.2
Bi-linear texture map

Polygon being textured

Bi-linear Map - Solving for α and β

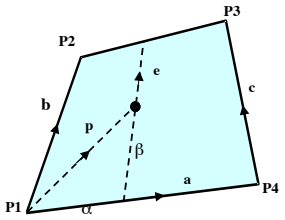
$\mathbf{p} = \alpha\mathbf{a} + \beta\mathbf{e}$

$\mathbf{e} = \mathbf{b} + \alpha(\mathbf{c} - \mathbf{b})$

so

$\mathbf{p} = \alpha\mathbf{a} + \beta\mathbf{b} + \alpha\beta(\mathbf{c}-\mathbf{b})$

Its Quadratic !



Non Linearities in texture mapping

- The second order term means that straight lines in the texture may become curved when the texture is mapped.
- However, if the mapping is to a parallelogram:

  $\mathbf{p} = \alpha\mathbf{a} + \beta\mathbf{b} + \alpha\beta(\mathbf{c}-\mathbf{b})$

  and

  b=c

  so $\mathbf{p} = \alpha\mathbf{a} + \beta\mathbf{b}$

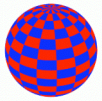
Texture Mapping & Illumination

- Texture mapping can be used to alter some or all of the constants in the illumination equation

$$L(\omega_r) = k_a + \left(k_d(\mathbf{n}\cdot\mathbf{l}) + k_s(\mathbf{v}\cdot\mathbf{r})^q\right)\frac{\Phi_s}{4\pi d^2}$$


Constant Diffuse Color

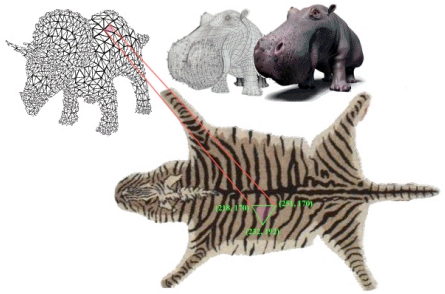

Diffuse Texture Color


Texture used as Label


Texture used as Diffuse Color

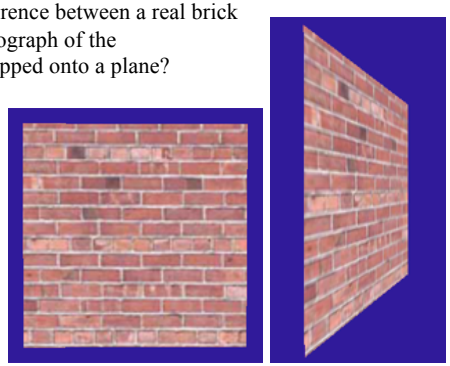
2D Texture Mapping

- Increases the apparent complexity of simple geometry
- Requires perspective projection correction
- Can specify variations in shading within a primitive:
  - Illumination
  - Surface Reflectance



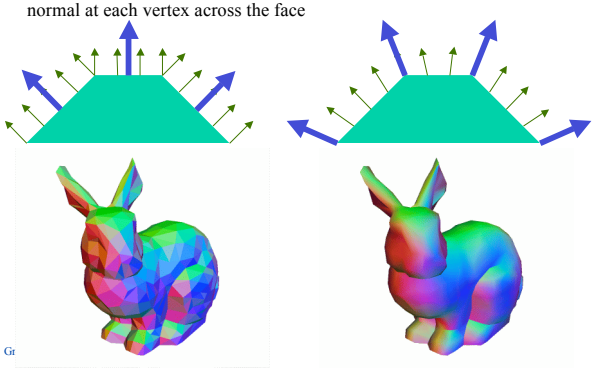
What's Missing?

- What's the difference between a real brick wall and a photograph of the wall texture-mapped onto a plane?
- What happens if we change the lighting or the camera position?



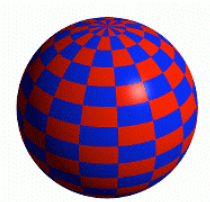
Remember Normal Averaging for Shading?

- Instead of using the normal of the triangle, interpolate an averaged normal at each vertex across the face



Bump Mapping

- Textures can be used to alter the surface normal of an object.
- This does not change the actual shape of the surface - we are only shading it as if it were a different shape!



Sphere w/Diffuse Texture

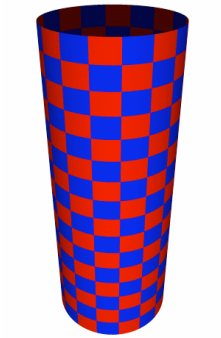
Swirly Bump Map

Sphere w/Diffuse Texture & Bump Map

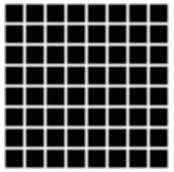
Bump Mapping

- The texture map is treated as a single-valued height function.
- The partial derivatives of the texture tell us how to alter the true surface normal at each point to make the object appear as if it were deformed by the height function.

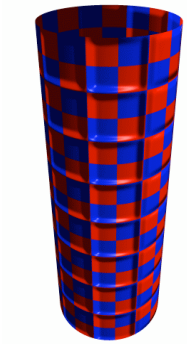
Another Bump Map Example



Cylinder w/Diffuse Texture Map



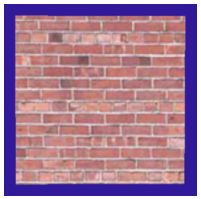
Bump Map

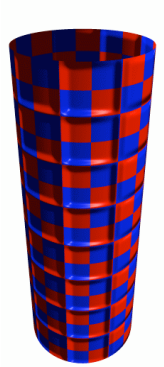


Cylinder w/Texture Map & Bump Map

What's Missing?

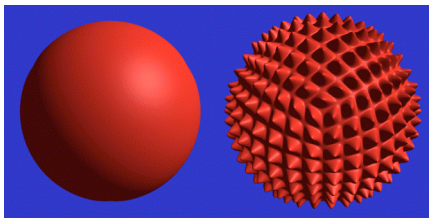
- What does a texture- & bump-mapped brick wall look like as you move the viewpoint?
- What does the silhouette of a bump-mapped sphere look like?





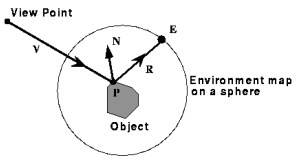
Displacement Mapping

- Use the texture map to actually move the surface point.
  - How is this different than bump mapping?
- The geometry must be displaced before visibility is determined.

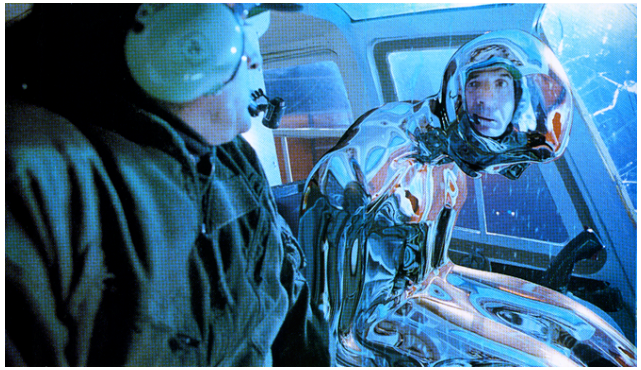


Environment Maps

- We can simulate reflections by using the direction of the reflected ray to index a spherical texture map at "infinity".
- Assumes that all reflected rays begin from the same point.



Environment Mapping Example



Terminator II

Questions?