CSC 240 Computer Graphics
Day 18: More Mapping

Nick Howe
Smith College
Lab 10 Review

How do we set up the \((u, v)\) coordinates for the pyramid?
• Adjust coordinates for triangular faces (some outside 0-1 range)
• Split bottom face diagonally
Lab 10 Review

Order of vertex listings must match between `pyramidGeom.faces` and `pyramidGeom.faceVertexUvs`.

ABC is not the same as ACB, BCA, CBA, etc.
Q. I don't understand the projection diagram for perspective -correct. Can you explain why the span is bigger/smaller? Can you explain the reason and logics behind the interpolation and the math?

A. The span changes in size for the same reason that railroad ties decrease in size with distance from the camera.
Q. Why perspective makes x & y shrink, but not z? Why the affine image looks like that? (The image in the middle)

A. After perspective projection, there is no z. The affine image computes the texture mapping wrong; this is the point.

Q. I don’t quite understand how we get that formula for u* and v*. Would you please explain it in more detail?

A. As part of the perspective projection, we normalize the homogeneous coefficient. This introduces a 1/z factor in all the XY coordinates.

\[
P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} xf/z \\ yf/z \\ 1 \end{bmatrix}
\]

\[
u = \frac{au_1 + \beta u_2 + \gamma u_3}{\alpha + \beta + \gamma},
\nu = \frac{av_1 + \beta v_2 + \gamma v_3}{\alpha + \beta + \gamma}
\]

\[
u^* = \frac{u_1 + \beta z_1 + \gamma z_2 + \frac{u_3}{z_3}}{\alpha + \beta + \gamma},
\nu^* = \frac{v_1 + \beta z_1 + \gamma z_2 + \frac{v_3}{z_3}}{\alpha + \beta + \gamma}
\]
Q. Can you show visually what you mean when a polygon has the same z values? Is it just a 2D shape?

A. If all the z values are the same, the polygon is frono-parallel to the camera. There is no perspective correction; all distances are equal.
Q. Why is the RGB value range for blue (128-255) half that of red and green (0-255)?

A. Think of the origin at (128,128,128). Numbers less than 128 represent negatives. Since blue represents the z component, a negative value would be pointing away from the camera.
Q. When should we better use normal mapping? When should we better use bump mapping?

A. They are pretty interchangeable, so use whichever is convenient. Normal mapping might give you slightly more realistic results.

Q. My eyes are kind of deceiving me and I can't tell if the first black and white normal map is raised out or pushed in. Also can we create a normal map with pressed in texture if z can only be greater than 0?

A. White is out, black is in. All that matters is the height relative to neighboring points, since that gives the normal vector. Remember: everything is rendered on the flat polygon; we’re just shading.
Your Questions

Q. How does the bump map know then where each bump is if it doesn't have x or y?
A. The bumps/normal are mapped onto the polygon using (u,v) coordinates just like ordinary texture.

Q. How did you calculate the normal vector for the last question aka what is 127.5?
A. I used the equations from slide 6.

Q. I enjoyed the coding section of this video
A. I’m glad!
Q. Can you explain the difference in using the perspective and orthographic projections for computing shadows? How much is a significant difference for comparing camera-visible points and light-visible points? And does it depend on canvas size?

A. For DirectionalLight, rays are parallel and we use an orthographic projection. For PointLight, rays radiate and we use perspective.
Q. Please explain more about "compare depth of camera-visible point to light-visible point" on the slides.

A. That’s what I was demonstrating with the 2D example. You render the scene from both the camera and the light, and compute 3D coordinates of the visible points. Where they are the same, the pixel is lit; where they are different, the pixel is unlit.
Q. How to convert camera frame to light frame?
A. There are 3D transformation matrices that convert world coordinates into camera or light frame coordinates.

\[ P_L = T_L P_W \quad \text{and} \quad P_C = T_C P_W \]

\[ P_L = T_L T_C^{-1} P_C \]

Q. How does the program render shadows with different darkness levels when the shadow map seems to only tell whether there is a shadow?
A. There may be other lights on the surface (ambient, etc.)
Activities

• Handout 6: shading calculations
• Handout 7: texture
• Lab 11 (complete on your own): install Blender

Icebreaker: If you had a boat, what would you name it (and why)?