Shadow maps
Midterm 2 analysis

Dinesh K. Pai

Textbook Chapter 15

Several slides courtesy of M. Kim

1

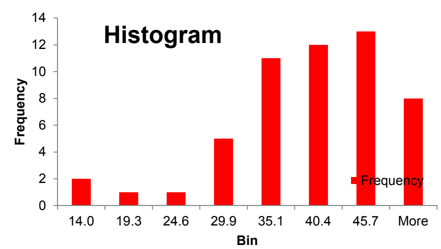
Announcements

- Midterm 2 results
- Shadow mapping

2

Statistics

Mean	36.83019
Standard Error	1.200428
Median	38
Mode	31
Standard Deviation	8.739248
Sample Variance	76.37446
Kurtosis	0.301182
Skewness	-0.72332
Range	37
Minimum	14
Maximum	51



3

Details

- Q1: Ans: 5,18,15,7,9,16,4,10
- Q2: Ans: F, F, T, T, T, F
- Q3: Blinn-Phong reflection model. See L20
  - Some wrote Phong, got partial credit
- Q4: Bernstein polynomials and Bezier curves.
  - Covered in BOTH L24 and L25. Straightforward, but many left it blank! Full credit if you picked coeff of C_1 or C_2.
- Q5: Almost everyone got this (median and mode = 5/5). Small variant of practice problem.

4

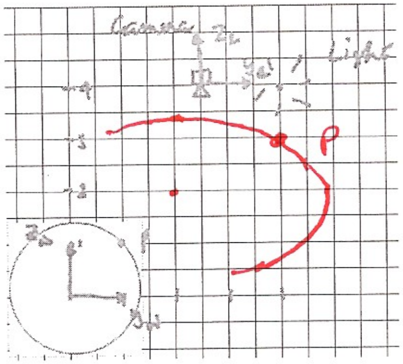
Model Matrix

$$O = \begin{bmatrix} 2 & \cdot & 2 \\ \cdot & 1 & 2 \\ \cdot & \cdot & 1 \end{bmatrix}$$

View Matrix

$$E = \begin{bmatrix} 1 & \cdot & 2.5 \\ \cdot & 1 & 4 \\ \cdot & \cdot & 1 \end{bmatrix}$$

$$E^{-1} = \begin{bmatrix} 1 & \cdot & -2.5 \\ \cdot & 1 & -4 \\ \cdot & \cdot & 1 \end{bmatrix}$$



Model-View

$$M = E^{-1} O = \begin{bmatrix} 2 & \cdot & -0.5 \\ \cdot & 1 & -2 \\ \cdot & \cdot & 1 \end{bmatrix}$$

Normal Matrix

$$N = \begin{bmatrix} 1/2 & \cdot \\ \cdot & 1 \end{bmatrix}$$

(a) What is the light vector $\vec{l}$ at the new position of $\tilde{p}$?

same in world or eye frame

$$\vec{l} = \begin{pmatrix} 4 \\ 4 \end{pmatrix} - \begin{pmatrix} 4 \\ 3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

[ok if you used homogeneous coords too]

(b) What is the normal vector $\vec{n}$ at the new position of $\tilde{p}$?

$$\bar{n} = \text{normalize}\left(N \begin{pmatrix} 1 \\ 1 \end{pmatrix}\right) = \text{normalize}\begin{pmatrix} 1/2 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{5}}\begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

(c) Compute the diffuse color of fragment at the new position of $\tilde{p}$, assuming that the object's color is (1,0.5,0) and the light's color is (0.5, 1, 0.3). In addition to the final answer, show your steps, either as a mathematical formula or as pseudocode.

(diffuse) $d = \max(0, \bar{l} \cdot \bar{n}) = \frac{2}{\sqrt{5}}$

(intensity) $I = \text{lightColor} * (\text{objectColor} * d)$

Note: vector x scalar as in GLSL

$$= \begin{pmatrix} 0.5 \\ 1 \\ 0.3 \end{pmatrix} * \left(\begin{pmatrix} 1 \\ 0.5 \\ 0 \end{pmatrix} * \frac{2}{\sqrt{5}} \right) = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \frac{1}{\sqrt{5}}$$

(d) Does the diffuse color of $\tilde{p}$ depend on the camera position? Why?

No. ~~Above~~ formula doesn't need it, (unless p is occluded .. optional)

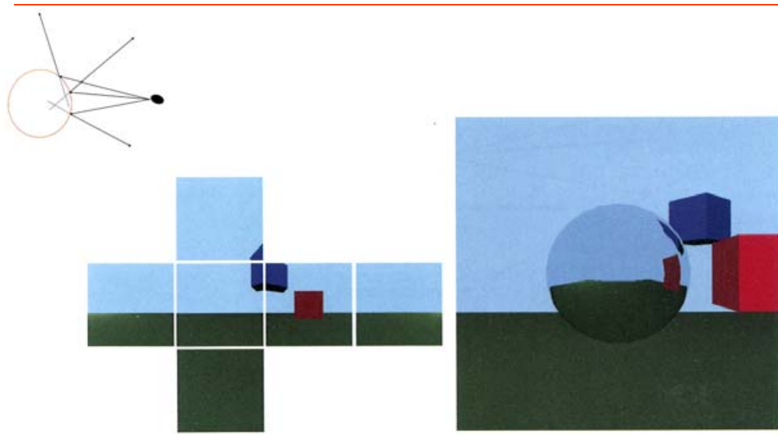
Details

- Q6: Mainly asked to see if you understand what projector textures do. With geometric reasoning answer is easy = 0.5. If you tried to compute it, more complex… go partial credit for being on right track.
- Q7: see attached

Multipass

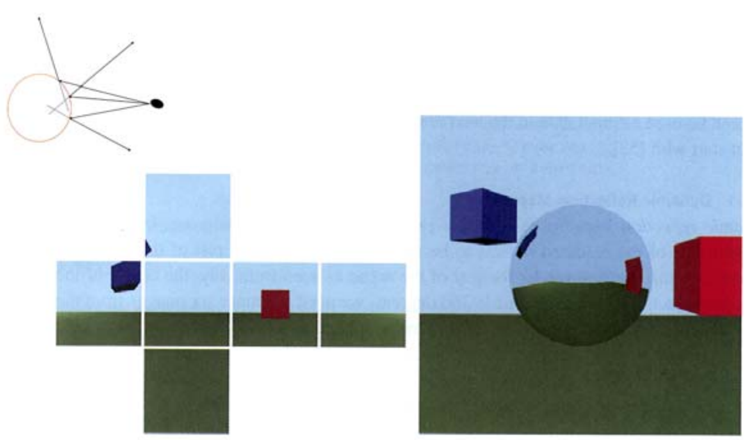
- More interesting rendering effects can be obtained using multiple rendering passes over the geometry in the scene.
- In this approach, the results of all but the final pass are stored offline and not drawn to the screen.
- To do this, the data is rendered into something called, a FrameBufferObject, or FBO.
- After rendering, the FBO data is then loaded as a texture, and thus can be used as input data in the next rendering pass.

Multipass



7

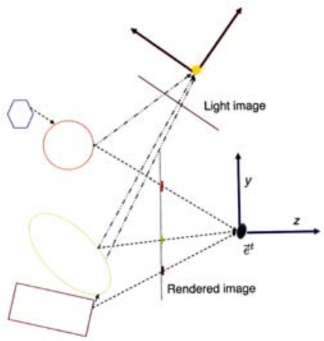
Multipass



8

Shadow mapping

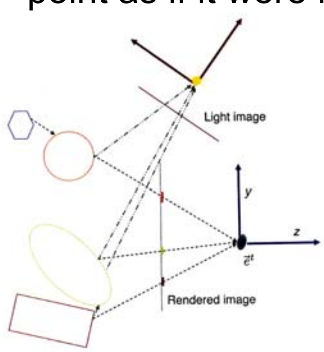
- First pass: create "shadow map", a z-buffer image from the point of view of the light
- Second pass: check if fragment is visible to the light using shadow map.



9

Shadow mapping

- If a point observed by the eye is not observed by the light, then there must be some occluding object in between, and we should draw that point as if it were in shadow.



10

Shadow mapping

- In a first pass, we render into an FBO the scene as observed from some camera whose origin coincides with the position of the point light source. Let us model this camera transform as:

$$\begin{bmatrix} x_t w_t \\ y_t w_t \\ z_t w_t \\ w_t \end{bmatrix} = P_s M_s \begin{bmatrix} x_o \\ y_o \\ z_o \\ 1 \end{bmatrix}$$

for appropriate matrices, P_s and M_s .

11

Shadow mapping

- During this first pass, we render the scene to an FBO using M_s as the modelview matrix and P_s as the projection matrix.
- In the FBO we store, not the color of the point, but rather its "depth value".
- Due to z-buffering, the data stored at a pixel in the FBO (depth value), is a monotone function of z_t. This FBO is then transferred to a texture.

12

Shadow mapping

- During the second rendering pass, we render our desired image from the eye's point of view, but for each pixel, we check and see if the point we are observing was also observed by the light, or if it was blocked by something closer in the light's view.
- To do this, we use the same computation that was done with projector texture mapping
- Doing so, in the fragment shader, we can obtain the varying variables x_t, y_t and z_t associated with the point $[x_o, y_o, z_o, 1]^t$.

13