Computer Graphics
CSE 457, Autumn 1998

Brian Curless, 318 Sieg
Head TA: Craig S. Kaplan

Homework 3: Perspective Tranformations, Hidden Surfaces, Shading, and Raytracing

Received: Monday, 2 November 1998
Due: Monday, 16 November 1998

Directions

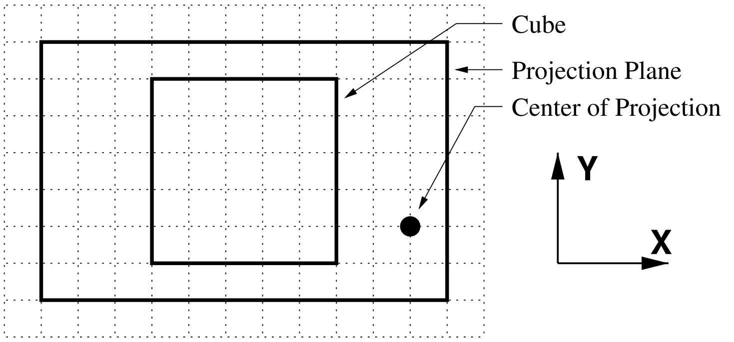
Please provide short written answers to the questions in the space provided. If you require extra space, you may staple additional pages to the back of your assignment. Feel free to talk over the problems with classmates, but please answer the questions on your own.

Name: ____

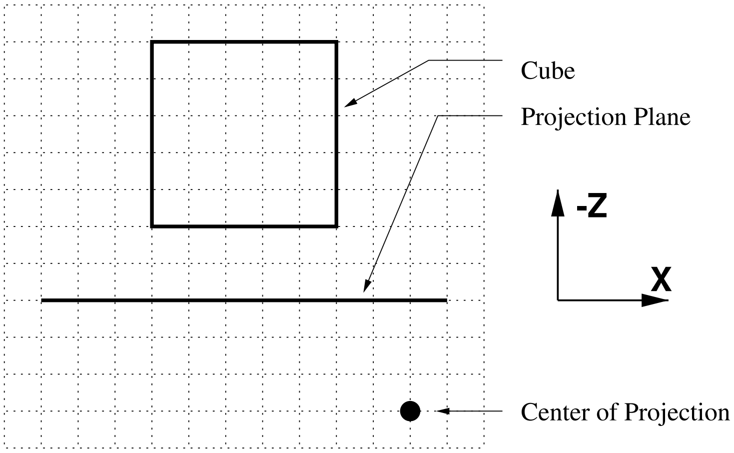
Problem 1. Shown below are two parallel projections of a cube, a projection plane, and a center of projection. The "top view" is looking down in the -y direction and the "front view" is looking in the -z direction.

(a) In the space provided, sketch the cube as it would be seen in perspective projection given the center of projection and projection plane. With a few contruction lines this can be done very accurately. Draw hidden lines as dashed, and identify any vanishing points. (It might be helpful to contruct the cube on top of the Front View Parallel Projection, before drawing it in the space provided)

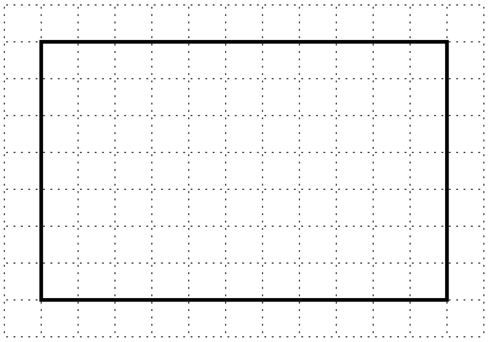
Front View Parallel Projection



Top View Parallel Projection



Fill in the perspective projection here



Problem 1 (continued)

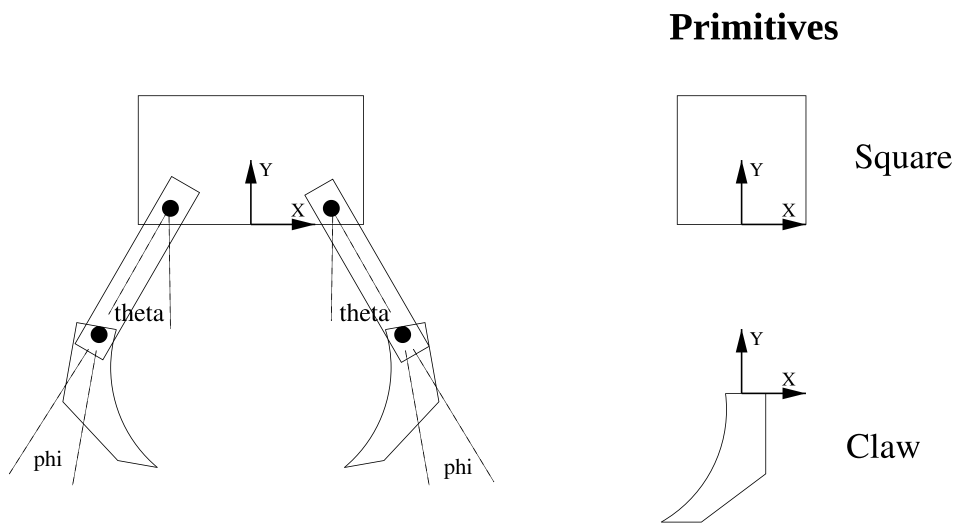
(b) How would the image of the cube change if the projection plane was moved to the right? (in the x direction)

(c) In terms of vanishing points, how would the image of the cube change if the center of projection was moved to the left? (in the -x direction)

(d) In general, if we rotate and translate the projection plane without moving the center of projection, how do the occlusion relationships change? That is, do surfaces that were occluded in one image become unoccluded in another? Explain.

Problem 2. You are modeling a hanging pincer as shown below. There are two primitives available to you, **Square** and **Claw**. The local coordinates for each primitive are shown. The transformations available are

- $\mathbf{R}(\theta)$ - rotate by θ degrees (counter clockwise)
- $\mathbf{T}(t_x, t_y)$ - translate by t_x, t_y
- $\mathbf{S}(s_x, s_y)$ - scale s_x and s_y in the x and y directions
- $\bar{\mathbf{R}}_{\mathbf{Y}}$ - reflect through the y axis



Problem 2 (continued)

(a) On the previous page, give names to the different pieces of the pincer. Now, in the space below, draw a tree hierarchy to specify the pincer. Use your naming convention for the nodes and label the transformations along the edges of the tree as in the robot figure example in class.

(b) Now write expressions for each of these edge transformations using only the symbols given on the previous page. Leave θ and ϕ as symbols; these are the parameters that you intend to animate. All other parameters should be definite numbers that you can estimate from the figure. Assume the **Square** is one unit in size.

Problem 2 (continued)

(c) What is the entire sequence of transformations applied to the claw drawn on the far left side of the figure?

(d) Redraw your hierarchy using a directed acyclic graph (DAG) to simplify the specification.

Problem 3. There are many issues to consider when choosing a hidden surface algorithm. In class we answered a number of questions about z-buffering and ray casting. In this question you will do the same for the **BSP tree algorithm**. Briefly explain your answers.

(a) Is it sort-first or sort-last? Or both?

(b) Is it image-precision or object-precision? Or both?

(c) Is it image-order or object-order? Or both?

(d) Do all polygons need to be rendered?

(e) Is it an online algorithm?

Problem 3 (continued)

(f) Can it handle transparency?

(g) Can it handle refraction?

(h) Can shading be done efficiently?

(i) What are the memory requirements proportional to?

(j) What types of 3D graphics programs are well suited to BSP trees?

Problem 4. The Phong shading model can be summarized by the following equation:

$$I = k_e + k_a I_a + \sum_i f(d_i) I_{li} \left[k_d (\mathbf{N} \cdot \mathbf{L}_i)_+ + k_s (\mathbf{V} \cdot \mathbf{R}_i)_+^{n_s} \right]$$

where

$$f(d_i) = \min \left\{ 1, \frac{1}{a_0 + a_1 d_i + a_2 d_i} \right\}$$

and the summation i is taken over all light sources.

The variables used in the Phong shading equation are summarized below:

$$I \quad a_0 \quad a_1 \quad a_2 \quad d_i \quad k_e \quad k_a \quad k_d \quad k_s \quad n_s \quad I_a \quad I_{li} \quad \mathbf{L}_i \quad \mathbf{N} \quad \mathbf{R}_i \quad \mathbf{V}$$

Which of the quantities above are affected if

(a) ...the viewing direction changes?

(b) ...the position of a light source changes?

(c) ...the orientation of the surface changes?

Problem 4 (continued)

Blinn and Newell have suggested that, when $\mathbf{V}$ and $\mathbf{L}$ are assumed to be constants, the computation of $\mathbf{V} \cdot \mathbf{R}$ can be simplified by associating with each light source a ficticious light source that will generate specular reflections. This second light source is located in a direction H halfway between $\mathbf{L}$ and $\mathbf{V}$. The specular component is then computed from $(\mathbf{N} \cdot \mathbf{H})^{n_s}$ instead of from $(\mathbf{V} \cdot \mathbf{R})^{n_s}$.

(d) Under what circumstances might $\mathbf{L}$ and $\mathbf{V}$ be assumed to be constant?

(e) How does the new equation using $\mathbf{H}$ simplify the shading computations?

Problem 4 (continued)

(f) In what way might the new equations change the appearance of the specular highlights? Justify your answer.

The ambient term in the Phong model is one way to guarantee that all visible surfaces recieve some light. Another possibility is to use the “headlamp” method in which a point light source is positioned at the eye, but no ambient term is used.

(g) Are these two methods equivalent? If so, explain why. If not, describe a situation in which the results would be clearly different.