Computer Graphics	Instructor: Zoran Popovic
CSE 457	Spring 2005

Homework #2

Hidden Surfaces, Shading, Ray Tracing,

Texture Mapping, Parametric Curves

Prepared by: Danny Wei & Diane Hu

Assigned: Monday, May 9th
Due: Friday, May 20th **at the beginning of class**

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 discuss the problems with classmates, but please ***answer the questions on your own.***

Name:____

Problem 1: Short Answer (10 points)

(a) (2 points) Antialiasing by adaptive supersampling samples rays at the corner of every pixel and subdivides the region recursively if the difference between neighboring sample colors is too great. When will this method fail?

(b) (2 points) Why is it easy for Gouraud interpolation to miss specular highlights? What can you do to make it more sensitive to specular highlights?

(c) (2 points) Can every third order Bezier curve be broken into two other third-order Bezier curves? If so, why?

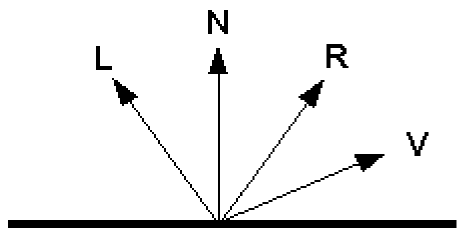
(d) (2 points) Is it possible to have a C2 continuous spline that is not also C1 continuous? Given an example or explain why it is not possible.

(e) (2 points) Does a curve that is C1 continuous imply that the curve is also G1 continuous (See Foley definition in text)? If so, why? If not, give an example of a curve that would be C1 continuous and G0 continuous.

Problem 2. Halfway Vector Specular Shading (10 points)

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

(a) (2 points) On the diagram below, assume that **V** and **L** are the new constant viewing direction and lighting direction vectors. Draw the new direction **H** on the diagram.



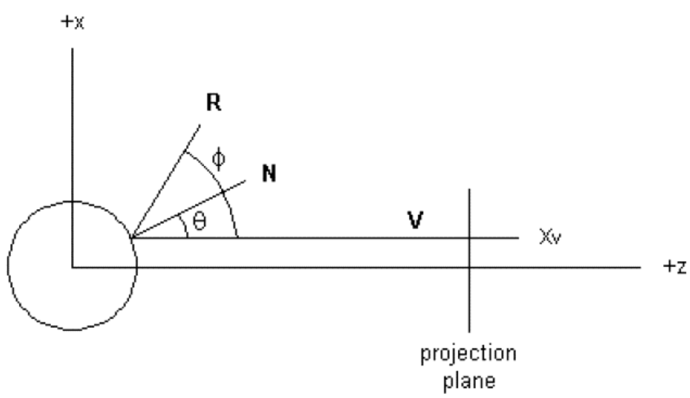
(b) (4 points) Under what circumstances or by making what approximations might **L** and **V** be assumed constant (or, at least, roughly so) for every point in the scene as seen through every pixel on the image plane?

(c) (4 points) Let's make the constant L, V assumption and use the halfway vector for shading. What is an advantage of this approach? For general lighting and viewing conditions, what is a disadvantage of this approach?

Problem 3. Environment Mapping (17 points)

One method of environment mapping (reflection mapping) involves using a "gazing ball" to capture an image of the surroundings. The idea is to place a chrome sphere in an actual environment, take a photograph of the sphere, and use the resulting image as an environment map. Let's examine this in two dimensions, using a "gazing circle" to capture the environment around a point.

Below is a diagram of the setup. In order to keep the intersection and angle calculations simple, we will assume that each view ray **V** that is cast through the projection plane to the gazing circle is parallel to the z-axis, meaning that the viewer is located at infinity on the z-axis. The circle is of radius 1, centered at the origin.



(a) (5 points) If the x-coordinate of the view ray is x_v, what are the (x,z) coordinates of the point at which the ray intersects the circle? What is the unit normal vector at this point?

(b) (3 points) What is the angle between the view ray **V** and the normal **N** as a function of x_v?

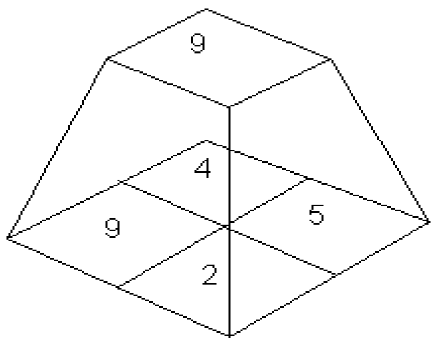
Problem 3. Environment Mapping (Continued)

(c) (5 points) Note that the angle φ between the view ray **V** and the reflection direction **R** is equal to 2θ, where θ is the angle between **V** and the normal **N**. Plot φ versus x_v. In what regions do small changes in the intersection point result in large changes in the reflection direction?

(d) (4 points) We can now treat the photograph of the chrome circle as an environment map. If we were to ray-trace a new, shiny object and index into the environment map according to each reflection direction, would we expect to get the same rendering as if we had placed the object into the original environment we photographed? Why or why not?

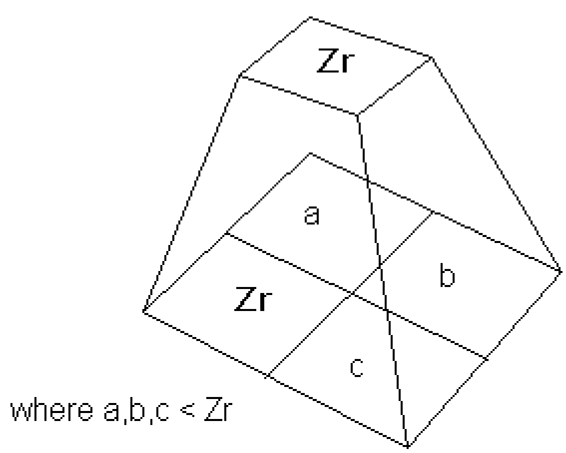
Problem 4. Z-buffer (13 points)

The Z-buffer algorithm can be improved by using an image space "Z-pyramid." The basic idea of the Z-pyramid is to use the original Z-buffer as the finest level in the pyramid, and then combine four Z-values at each level into one Z-value at the next coarser level by choosing the farthest (largest) Z from the observer. Every entry in the pyramid therefore represents the farthest (largest) Z for a square area of the Z-buffer. A Z-pyramid for a single 2x2 image is shown below:



(a) (3 points) At the coarsest level of the pyramid there is just a single Z value. What does that Z value represent?

Suppose we wish to test the visibility of a polygon **P**. Let **Zp** be the nearest (smallest) Z value of polygon **P**. Let **R** be the smallest region in the Z-pyramid that *completely* covers polygon **P**, and let **Zr** be the Z value that is associated with region **R** in the Z-pyramid.



Problem 4. Z-buffer (Continued)

(b) (3 points) What can we conclude if **Zr** < **Zp**?

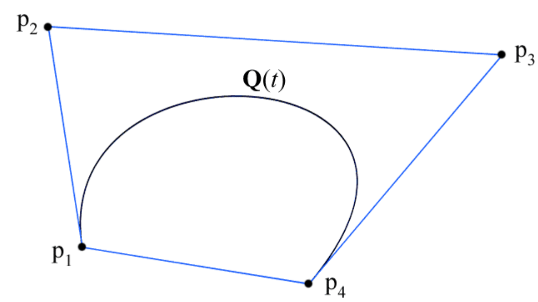
(c) (3 points) What can we conclude if **Zp** < **Zr**?

If the visibility test is inconclusive, then the algorithm applies the same test recursively: it goes to the next finer level of the pyramid, where the region **R** is divided into four quadrants, and attempts to prove that polygon **P** is hidden in each of the quadrants **R** of that **P** intersects. Since it is expensive to compute the closest Z value of **P** within each quadrant, the algorithm just uses the same **Zp** (the nearest Z of the *entire* polygon) in making the comparison in every quadrant. If at the bottom of the pyramid the test is still inconclusive, the algorithm resorts to ordinary Z-buffered scan conversion to resolve visibility.

(d) (4 points) Suppose that, instead of using the above algorithm, we decided to go to the expense of computing the closest Z value of **P** within each quadrant. Would it then be possible to always make a definitive conclusion about the visibility **P** of within each pixel, without resorting to scan conversion? Why or why not?

Problem 5. Bezier Properties (15 points)

A nice property of Bezier curves is that the curve itself will always remain within the *convex hull* of its control points. The convex hull of a set of points is defined as the smallest convex polygon containing all those points. Intuitively, you might imagine the convex hull of a set of points in two dimensional space to be the polygon defined by wrapping a rubber band around those points. In three dimensional space, imaging using a rubber sheet instead.



An intuitively true property about convex hulls is as follows. Suppose we are given n points; call these $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n$. Now suppose we are given n real numbers, $w_1, w_2, \ldots, w_n$. If $0 \le w_i \le 1$ for all $1 \le i \le n$ and $w_1 + w_2 + \ldots + w_n = 1$, then $\mathbf{q} = w_1\mathbf{p}_1 + w_2\mathbf{p}_2 + \ldots + w_n\mathbf{p}_n$ lies within the convex hull of the points $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n$. In other words, taking a weighted average of a set of points necessarily gives a point within the convex hull of those points.

(a) (3 points) A point on a cubic Bezier curve can be defined by the function

$$\mathbf{Q}(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \\ \mathbf{p}_4 \end{bmatrix}$$

where $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4$ are the control points of the curve and $0 \le t \le 1$. Write out the Bezier basis functions $f_1(t),\ f_2(t),\ f_3(t),\ f_4(t)$ such that

$$\mathbf{Q}(t) = f_1(t)\mathbf{p}_1 + f_2(t)\mathbf{p}_2 + f_3(t)\mathbf{p}_3 + f_4(t)\mathbf{p}_4 .$$

Problem 5. Bezier Properties (Continued)

(b) (4 points) Show that $f_1(t) \geq 0$, $f_2(t) \geq 0$, $f_3(t) \geq 0$, $f_4(t) \geq 0$ for all t, such that $0 \leq t \leq 1$.

(c) (3 Points) Show that $f_1(t) + f_2(t) + f_3(t) + f_4(t) = 1$ for all t, such that $0 \leq t \leq 1$.

(d) (2 points) Using the property about convex hulls stated previously, argue that any Bezier curve must lie within the convex hull of its control points. (Make sure you use the convex hull property exactly as it is stated)

(e) (3 points) Give an example of a situation in which the convex hull property of Bezier curves might be useful.

Problem 6. Bezier Properties (10 points)

Prove that when you have a straight Bézier curve where the distance between neighboring Bezier control points equals one, the velocity of the line (i.e. the first derivative of Q(t)) is constant.
(Hint: Look at the Hermite equation and the relation between Bezier and Hermite curves)

The equation for a general Bézier curve is:

$$Q(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}$$

The equation for a general Hermite curve is:

$$Q(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{bmatrix}$$