15-462 Computer Graphics I

Final Examination

May 6, 2002

Name: _____

Andrew User ID: _____

- This is an open book, open notes exam.
- Write your answer legibly in the space provided.
- There are 16 pages in this exam, including 5 worksheets.
- It consists of 4 questions worth a total of 250 points.
- You have 3 hours for this exam.

Problem 1	Problem 2	Problem 3	Problem 4	Total
70	**60**	**50**	**70**	**250**

1. Clipping (70 points)

The Sutherland-Hodgman clipping algorithm in two dimensions was described in lecture and in the textbook as an algorithm for clipping an arbitrary polygon against a clipping rectangle. In this problem you are asked to generalize the algorithm to clip an arbitrary (convex or concave) polygon against a convex polygonal clipping region, both still considered in two dimensions.

Assume a simple, convex, polygonal clipping region C is given by a sequence of vertices $c_0, \ldots, c_n$ where $c_0 = c_n$. Further assume the edges are enumerated in a clockwise order.

1. (50 points) Describe an algorithm derived from the Sutherland-Hodgman algorithm that clips a given arbitrary polygon with vertices $v_0, \ldots, v_k$ with $v_0 = v_k$ against C. Use pseudo-code if it helps to clarify the description.

2. (10 points) Give a counterexample showing that the algorithm in part 1 does not always work correctly for concave clipping regions.

3. (10 points) How would you implement clipping against a concave region?

2. Ray Tracing (60 points)

Ray tracing works well with constructive solid geometry (CSG). In this problem we explore how to perform the necessary ray/object intersection calculations.

We assume that a ray is given as $\mathbf{p}(\alpha) = \mathbf{p}_0 + \alpha\mathbf{v}$ for $\alpha \geq 0$. For a set of basic objects we already have implemented functions that take $\mathbf{p}_0$ and $\mathbf{v}$ as arguments and compute an *ordered intersection list* $\alpha_1 < \cdots < \alpha_n$ of parameter values such that the ray intersects the surface of the object at all α_i. If there is no intersection, the list is empty (that is, $n = 0$).

For simplicity we assume the ray originates outside all objects and ignore singularities, so the ray is inside the object between α_1 and α_2, outside the object between α_2 and α_3, inside between α_3 and α_4, etc.

1. (25 points) Show how to compute the intersection list for $T \cup S$ from the intersection lists for objects T and S.

2. (35 points) Show how to compute the intersection list for $T - S$ from the intersections lists for objects T and S.

3. Image Processing (50 points)

In this problem we consider edge detection in an image by filtering.

We can approximate $\frac{\partial f}{\partial x}$ with the so-called Sobel operator, written here in matrix form:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

1. (10 points) Give the Sobel operator that approximates $\frac{\partial f}{\partial y}$.

2. (20 points) We can use the magnitude of the gradient vector as an edge filter, approximating it as

$$|\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \simeq \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|.$$

Apply this technique to the following image.

0	1	1	1	0
1	4	4	4	1
1	4	4	4	1
1	4	4	4	1
0	1	1	1	0

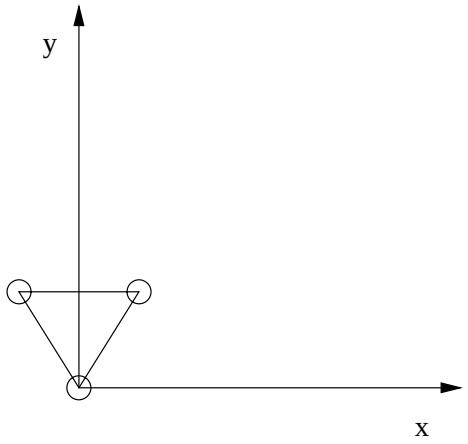
×	×	×	×	×
×				×
×				×
×				×
×	×	×	×	×

3. (20 points) In non-photorealistic rendering, we often want to enhance edges or draw outlines over a given image. Consider the case where we have rendered an image in OpenGL and now want to draw over edges with a specified color. Sketch in some detail how you could use the above edge filter together with OpenGL's blending facilities for this purpose.

4. Scientific Visualization (70 points)

In this problem you are asked to design and analyse a *marching triangles* algorithm for drawing contour lines in analogy with the marching cubes algorithm.

Assume we use only equilateral triangles where the length of every side is h, and the first one is placed as shown. Draw in at least 6 additional triangles to show your method of subdividing the plane.



1. (10 points) Devise a system of addressing the grid points with pairs of integer indices (i, j) and show how to compute the x and y coordinate of an arbitrary grid point p_{ij}.

2. (5 points) Assume we are given a function $f(x, y)$ and we are trying to draw the contour line $f(x, y) = c$. Describe how you would color each grid point either black or white as in the marching squares algorithm.

3. (15 points) Show all the possibilities for coloring the vertices of a triangle and indicate how the contour line crosses the edge of the triangle in each case. Up to symmetry, how many different situations are there?

4. (15 points) Give the formula for interpolation if we have determined that the contour line crosses the edge between two adjacent grid points (x_1, y_1) and (x_2, y_2).

5. (10 points) Explain how and when you would adaptively subdivide the grid in order to obtain a more accurate approximation of the contour line. Use pictures for illustration.

6. (5 points) Does the marching triangles algorithm you developed above have an inherent ambiguity problem as the marching squares algorithm?

7. (10 points) Assume we want to generalize our algorithm to three dimensions and use *marching tetrahedrons* instead of *marching cubes* in order to draw isosurfaces of a function $f(x, y, z)$. How many possible vertex colorings are there? How many unique cases remain once we account for symmetries?

Worksheet

Worksheet

Worksheet

Worksheet

Worksheet