

# Final Review

15-462 Graphics I  
Spring 2003

April 30, 2003

## 1 Course announcement: *Animation Art and Technology*

Prof. Jessica Hodgins wanted us to mention that 15-505, *Animation Art and Technology* will be offered in the fall, meeting MW *in the evening* 6:30-9:20pm. The class will explore ways in which the production of animation can incorporate the enhancement and modification of software in combination with the objective of artistic expression or content development. Students will work on projects in cross-disciplinary teams. Student need permission of the instructor to enroll, so talk to Jessica if you are interested. The web site for the current quarter is at <http://www-2.cs.cmu.edu/~jkh/aat/>.

## 2 Overview

Here is a (mostly complete) list of concepts you should make sure you understand from the second half of the course (since the midterm).

- Line Clipping
  - Cohen-Sutherland
  - Liang-Barsky
- Polygon clipping
  - Sutherland-Hodgeman pipeline
- Rasterization
  - Bresenham's algorithm
- Image Processing
  - Applying a filter (by hand)
  - Understanding basic filters (blur, median)
- Ray Tracing
  - How to set up intersection calculation
  - Angle of reflection
  - Barycentric coordinates
  - Constructive solid geometry (CSG)
- Spatial Data Structures
  - Quadtrees/octrees
  - BSP trees

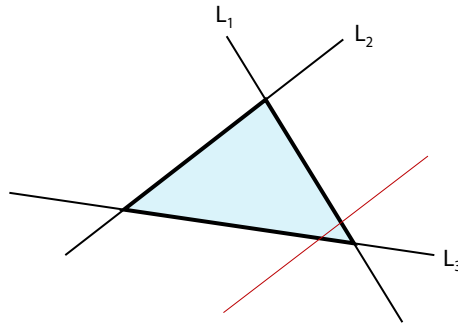
- Radiosity
  - Radiosity equation
  - Form factors
  - Classical vs. progressive radiosity
- Visualization
  - Marching squares/cubes
  - Transfer function/opacity/blending
- Non-photorealistic rendering (understand the concepts)
- Animation (same)
- Advanced global illumination (same)

### 3 Line clipping

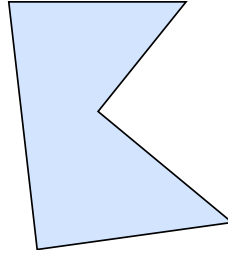
We covered two basic line clipping algorithms in class, Cohen-Sutherland and Lian-Barsky. You should understand them both,

#### 3.1 Cohen-Sutherland

Recall the Cohen-Sutherland algorithm works for clipping lines to rectangles. We can use an analogous algorithm for clipping to any *convex* polygonal shape. Let us consider the case of clipping to a triangle; consider the following diagram,

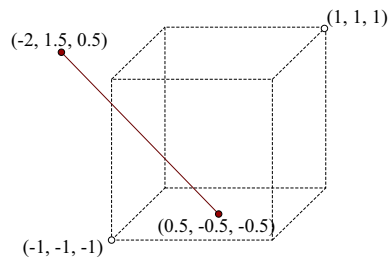


1. Recall that the Cohen-Sutherland uses *outcodes* to determine the clipping status of triangles. Give the outcode bits and the half-plane tests that you would need to clip to the triangle defined by  $L_1$ ,  $L_2$ , and  $L_3$ .
2. What are the trivial accept and trivial reject checks in terms of the outcodes you defined? Make sure you define any symbols that you use (e.g., logical or bitwise operators).
3. Given a line segment, what is the maximum number of subdivide steps that the Cohen Sutherland algorithm will perform? Draw a line segment on the triangle above that requires this maximum number of subdivisions. Label the line endpoints, and specify the outcodes.
4. We claim above that this approach will work for any convex polygon. Consider the following non-convex polygon. For this polygon, give an example of a line segment for which the trivial reject criterion will give an incorrect result. Why does it fail for this case?



### 3.2 Liang-Barsky

1. Consider the following line which we will be clipping to the canonical 3D viewing volum,



- (a) What is the parametric equation of the line? Write it in the form

$$\begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} + t \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

- (b) What are the six parametric coordinates (the  $t$  values) for the intersections of the line with each clipping plane? Label them as entering or leaving intersections.
- (c) What are the parametric coordinates of the end-points of the visible segment?
- (d) What are the  $(x, y, z)$  coordinates of the end-points of the visible segment?

## 4 Polygon clipping

### 4.1 Sutherland-Hodgeman pipeline

Remind yourself how this algorithm works.

## 5 Rasterization

### 5.1 Bresenham's algorithm

No questions here, just remind yourself how this algorithm works.

## 6 Image processing

### 6.1 Applying filters

1. What is the difference between applying a filter and matrix multiplication?

2. Let the  $3 \times 3$  mean filter be defined by

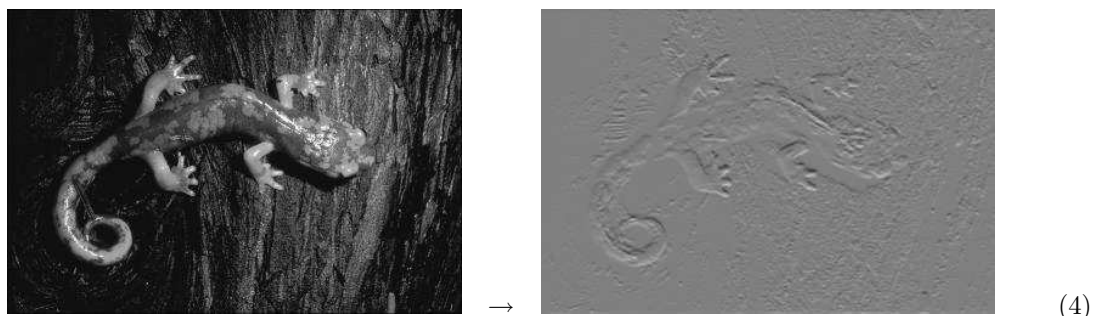
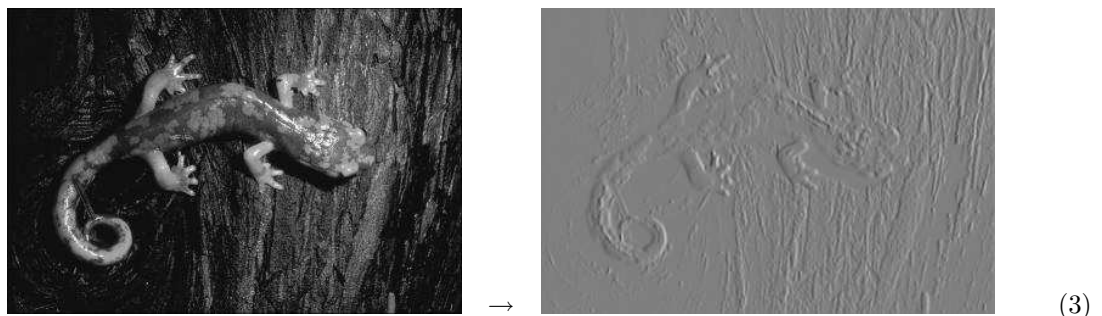
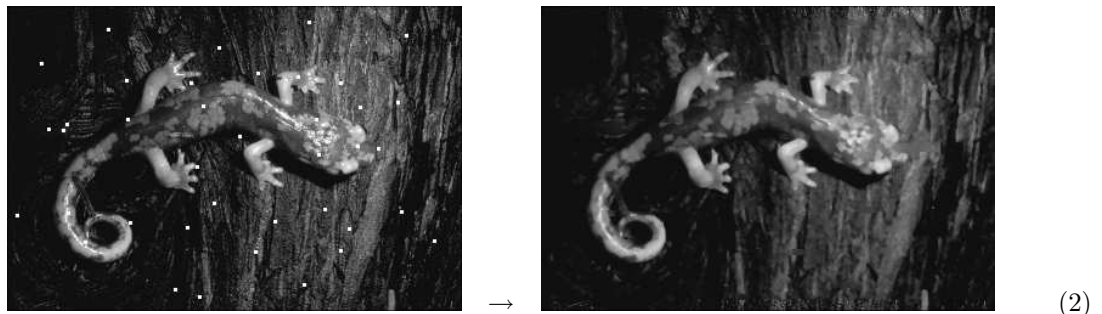
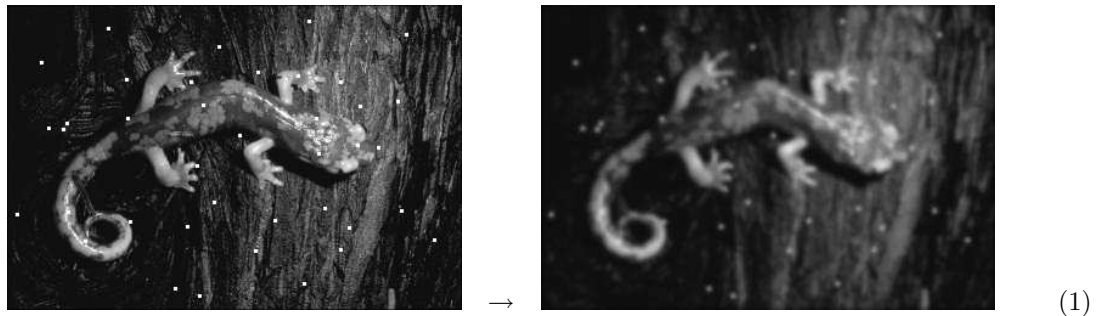
$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

What is the result of applying this filter to the following “image”? You can ignore the outermost pixels.

```
1 1 0 0 1 0
1 0 1 1 1 1
0 1 0 0 0 0
0 0 1 0 0 1
1 0 0 0 0 1
```

## 6.2 Understanding basic filters

1. Identify the filters used in the following (image on left is original, image on right is after filter is applied). Image is courtesy Duncan Parks via the Berkeley Image Database.



2. Which of the following filters are linear?

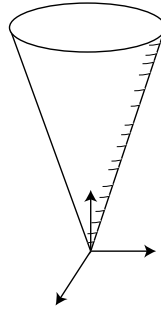
- Gaussian
- Sobel
- Gradient magnitude
- Mean
- Median

3. How can we use a Laplacian filter as a sharpening filter? A diagram would be especially helpful here.

## 7 Ray tracing

### 7.1 Setting up intersection calculations

1. Given an implicit function  $F(x, y, z)$  for an object, how do we determine whether a point  $(x_0, y_0, z_0)$  is inside, outside, or on the object?
2. Give an implicit function  $F(x, y, z)$  for the following cone, whose apex is at the origin and whose radius increases linearly in the positive Z direction (to infinity).



3. Given a ray  $\mathbf{p} + \mathbf{d}t$ , show how to derive its intersection with the above cone (it is sufficient to set up the equations, you don't need to solve them).
4. If we find an intersection with the cone, we'll need to do our local lighting calculation, and as you'll recall we will need a normal to the surface for this. Describe how to derive the normal to the surface at the point of intersection using the function  $F(x, y, z)$  described above (this one you should actually be able to calculate fairly easily).

### 7.2 Angle of reflection

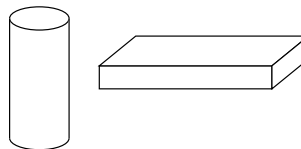
1. Derive the formula for the angle of reflection  $\mathbf{r}$  given an incoming ray direction  $\mathbf{v}$  and a surface normal  $\mathbf{n}$ .

### 7.3 Barycentric coordinates

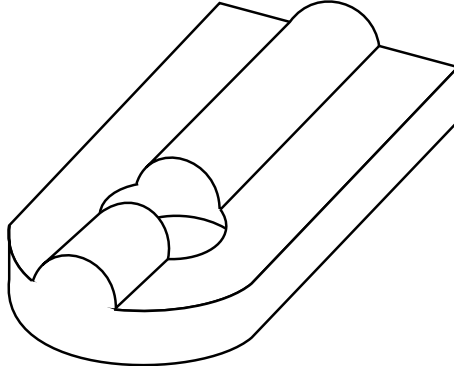
Be aware of how these are calculated and the way they are used in interpolating e.g. texture coordinates.

### 7.4 Constructive solid geometry (CSG)

1. Using only the (scaled) primitives of cylinders and boxes,



describe a CSG tree that would create the following gadget:



2. Given as input a CSG tree composed of unions, intersections, and differences, how would you detect if a particular ray  $\mathbf{p} + \mathbf{d}t$  has intersected it and where?

## 8 Spatial data structures

### 8.1 Quadtrees/Octrees

1. What is the most efficient way to use an octree for ray tracing?

### 8.2 BSP trees

First, remind yourself how to create a BSP tree on some arbitrary collection of polygons; you can use it to help answer the questions below.

1. When creating a BSP tree, we may choose to either use polygons as splitting planes, or we may just use arbitrary planes in space. Why might we choose one over the other?
2. How can we use a BSP tree to run the painter's algorithm?
3. How can we use a BSP tree to accelerate ray tracing?

## 9 Radiosity

### 9.1 Radiosity equation

Understand what the terms mean,

$$b_i a_i = e_i a_i + \rho_i \sum_{j=0}^n f_{ji} b_j a_j \quad (5)$$

$$b_i = e_i + \rho_i \sum_{j=0}^n f_{ij} b_j \quad (6)$$

$$(7)$$

### 9.2 Form factors

Covered in the last homework (gone over in class).

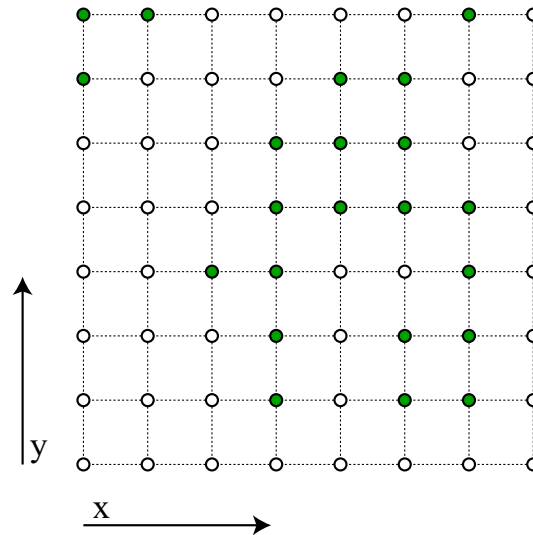
### 9.3 Classical vs. progressive radiosity

Understand the difference between these two.

## 10 Visualization

### 10.1 Marching squares/cubes

Suppose we have a function  $\phi(x, y)$  we have sampled only at the gridpoints  $(n\delta, m\delta)$  pictured below. Let dark-colored gridpoints represent positive function values ( $\phi(x, y) > 0$ ) and light-colored gridpoints represent negative function values ( $\phi(x, y) < 0$ ). Use the marching squares algorithm to (approximately) reconstruct the  $\phi(x, y) = 0$  level set.



Are there any ambiguities in the reconstruction?

### 10.2 Transfer function/opacity/blending

Blending was covered on assignment 6.