Practice Midterm Solutions
CMSC 427

Sample Problems (Note, some of these problems may be a bit more involved than ones I'd ask on a time-limited exam).

1. Create a matrix that rotates points 90 degrees about the point (1,1).

We can do this by translating to the origin, rotating, and translating back. We get:

$$\begin{pmatrix}1&0&1\\0&1&1\\0&0&1\end{pmatrix}\begin{pmatrix}0&-1&0\\1&0&0\\0&0&1\end{pmatrix}\begin{pmatrix}1&0&-1\\0&1&-1\\0&0&1\end{pmatrix}=\begin{pmatrix}1&0&1\\0&1&1\\0&0&1\end{pmatrix}\begin{pmatrix}0&-1&1\\1&0&-1\\0&0&1\end{pmatrix}=\begin{pmatrix}0&-1&2\\1&0&0\\0&0&1\end{pmatrix}$$

We can verify that this is right by checking that it takes the point (1,1) (written (1,1,1) in homogenous coordinates) to the point (1,1), and that it takes the point (2,1) to (1,2).

2. What is the distance from (3,2) to (7,5) in the direction (1,2)?

To do this we need to find the inner product between a vector from (3,2) to (7,5) and a unit vector in the direction (1,2). The first vector is (4,3). The unit vector we need is (1,2)/sqrt(5). This inner product is 10/sqrt(5)=2sqrt(5)

3. Provide any two rows of a 4x4 matrix that will transform 3D points as they would appear in the coordinate system of a viewer centered at (3,2,0) facing in the direction (1,2,3). The new *z* coordinate should describe the distance from the viewer to a point, in the direction that they are viewing it.

We are changing the origin and the viewing direction. We can do this by combining two matrices. If we just wanted to change origins, all we would need to do is create a 3D translation matrix that takes (3,2,0) to the origin. This would be:

$$\begin{pmatrix}1&0&0&-3\\0&1&0&-2\\0&0&1&0\\0&0&0&1\end{pmatrix}$$

However, we also want to change the viewing direction. To do this, we'll create a matrix in which the first three elements of the third row represent our new z direction. This direction is represented by the unit vector (1,2,3)/sqrt(14). So, the matrix we want will look like:

$$\begin{pmatrix} & & & \\ & & & \\ \frac{1}{\sqrt{14}} & \frac{2}{\sqrt{14}} & \frac{3}{\sqrt{14}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} & & & \\ & & & \\ \frac{1}{\sqrt{14}} & \frac{2}{\sqrt{14}} & \frac{3}{\sqrt{14}} & \frac{-7}{\sqrt{14}} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Notice that to get the bottom two rows we never had to come up with the top two rows of the first matrix on the left.

4. Suppose you have a camera with a focal point at (0,0,0) and an image plane of z=1. Give an example of a right triangle in 3D that will also appear in the image as a right triangle, assuming perspective projection. Do the same thing for orthographic projection.

Perspective projection scales points by the inverse of their distance. So, if we pick points that have the same z coordinate, they will all be scaled the same amount. So, for example, if we take a triangle with vertices at (0,0,2) (2,0,2), (0, 2, 2), this will appear in images with coordinates (0,0), (1,0), (0,1), and still be a right triangle.

With orthographic projection, the same triangle will appear as (0,0), (2,0), (0,2), also a right triangle.

5. Prove that parallel lines in the world do not always appear as parallel lines with perspective projection.

We just need an example to show this. Let’s consider lines that lie in the y=-5 plane, with our standard perspective camera. Consider a line with the equation z = mx+b. This contains the points (0,-5,b) and (1,-5, m+b). These appear in the image as points (0, -5/b) and (1/(m+b), -5/(m+b)). This line has a slope of (-5/(m+b) + 5/b)/(1/(m+b)) = -5 + 5(m+b)/b = 5m/b. Clearly this slope depends not just on m, the slope of the original line, but also on b. Two lines with the same value for m and different values for b will be parallel in the world, but will have different slopes in the image.

6. Explain how you would adapt DDA to discretely represent the boundary of a circle. What pixels would be filled in to represent a circle centered at (20,20) with a radius of 5?

First, let’s consider the simpler, naïve algorithm. We cannot just increment x from 15 to 25, and solve for two y values for each x, because in some parts of the circle y changes more rapidly than x, so we would get a circle with gaps. Instead, we can divide the circle into regions where either x or y is changing more rapidly. For example, at an angle between pi/4 and 0, y is changing more rapidly. This occurs in the region of the circle starting at (20+5/sqrt(2), 20+5/sqrt(2)) and ending at (25,20). Rounding off, this means we must consider y values of 24, 23, 22, 21, 20. For these, the corresponding x values are 23, 24, 25,25,25, so we fill in points (23,24), (24,23), (25,22), (25,21), (25,20).

We can try to adapt DDA to this problem by noticing that the tangent to a circle at a boundary point is perpendicular to a vector from the center to the boundary point. So if we start at the point (25,20), a vector from (20,20) to (25,20) is (5,0), and the tangent at that point is 0/5. If we change y by 1, we should not change x at all, and we find the point (25,21). At this point the slope will be -1/5, so the next point on the circle will be (24 4/5, 22). Rounding this off, we would fill in the pixel (25,22). The slope at this point would be -2/(4+4/5) = -5/12. So the next point on the circle will be (24 23/60, 23), which rounds to (24,23). Continuing this way brings us to (23.7, 24), rounded to (24,24). Results are a bit less accurate. Is this worth it?

7. Suppose you have an image with vertical, black and white stripes that are five pixels wide, and you want to shrink this image to half its current size in each dimension. Is there a problem with aliasing? If not, why not? If yes, how would you prevent this by smoothing using averaging? How much would you need to smooth? There may not be a simple, exact answer to this question, so just think about it and describe the consequences of different possible choices.

If we shrink this image by sampling, then stripes will be represented by either two or three pixels, depending on where we choose our sample points. This won't be as bad as some aliasing effects, but it still will create significant artifacts. For example, suppose a white stripe extends from 1 to 5, and a black stripe from 6 to 10, and we sample at 2,4,6,8,10. The white stripe will appear to be two pixels wide, and the black stripe will be three pixels wide. A horizontal slice of the image will look like: (1 1 0 0 0 1 1 0 0 0 1 1 …).

If we average a great deal, the stripes will become a uniform gray. Now there will certainly be no aliasing, although we will have completely lost the structure of the image.

Now, suppose we replace each pixel by the average of it and its neighbors. Then, after averaging, a horizontal slice of the image will look like:

(2/3 1 1 1 2/3 1/3 0 0 0 1/3 2/3 1 1 1 2/3 1/3 0 0 0 1/3 2/3 1 1 1 …)

If we sample this, we get something like: (1 1 1/3 0 1/3 1 1 1/3 0 1/3 …) This still isn't perfect, but it will look similar to a white stripe that is 2 2/3 pixels wide, and a black stripe that is 2 1/3 pixels wide.

We get even better results if we use weighted averaging, so that each pixel is replaced by the average of its original value and that of half of each of its neighbors. Now, averaging produces something like:

(3/4 1 1 1 3/4 ¼ 0 0 0 ¼ ¾ 1 1 1 ¾ ¼ 0 0 0 …)

Sampling this produces (1 1 ¼ 0 ¼ 1 1 ¼ 0 ¼ ….). Now the average intensity in the image is still .5 (can you prove that this will always be true) and we have something that looks more like equal sized white and black stripes.

8. Suppose we apply the Efros and Leung texture synthesis algorithm to an image consisting of vertical black and white stripes. What do you think we might get? How might this depend on the parameters that we use?

The key issue here is how big a neighborhood we use when deciding whether things are similar. If we use a neighborhood, that is big enough to capture a full stripe, then we will reproduce stripes of equal length. If we use a smaller neighborhood, we'll still get black and white stripes, but they may not be of equal length.

9. If a pixel has RGB values of (.7, .3, .2), how would you describe its HSV values?

v will be the value of the largest component. So v = .7.

White light with the same value would be (.7, .7, .7). We can make light that is completely saturated, but has the same hue as our light by subtracting out the white, which would produce (.5, .1, 0). If we scale this light by 1.4 we get saturated light of the same hue, but with a value of .7, that is represented in RGB as (.7, .14, 0). Our light is therefore part way in between the fully unsaturated gray light (.7,.7,.7) and the saturated light at (.7, .14, 0). What fraction of the way are we from fully unsaturated to saturated? .4/.56, or .5/.7, both of which equal 5/7. Hue is represented by the angle between our light and red. For example, we can use (.7,.7,.7) as the origin, and find the angle between (.7,0,0) and (.7,.14,0). Things look a little simpler if we scale everything by 1/.7, in which case the origin is (1,1,1), red is at (1,0,0), and our hue is at (1, .2,0). This is about 6 degrees.

If a pixel has a hue of green, a saturation of .5, and a value of .4, describe it with RGB values.

Gray with a value of .4 would be (.4,.4,.4). Green with this value would be (0, .4, 0). Our color is halfway in between these two, so it's (.2, .4, .2).

10. Suppose you had a monitor that emitted light that was either cyan, magenta or yellow. How could you use this to create white light?

Representing these in rgb, they are (0 1 1), (1 0 1) (1 1 0). These are usually used in a subtractive system, where mixing them together will produce black. But in a monitor, small colored lights are added together, and so we will have equal parts of red, green and blue, which will appear white.