Computer Graphics **Instructor: Zoran Popović**

CSE 457, Autumn 2002

Homework #1

Display Devices, Image Processing, Affine Transformations, Hierarchical Modeling

Prepared by: Erik Curre and Mathew Chasan

Assigned: Friday, October 11th
Due: Friday, October 25th, **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)

Provide a short answer (typically one or two sentences) to each of the following questions. In each case, you must clearly justify your answer.

1. (2 points) Color lookup tables are useful even when used with a true color framebuffer. Give an example of what the lookup table might be used for.

2. (2 points) Name one reason why you might use a parallel projection instead of a perspective projection.

3. (2 points) ***True or False:*** If you convolve an image with the Laplacian filter, you will typically get an image that looks about the same except it will be a bit sharper. Justify your answer.

4. (2 points) Why do we use a 4x4 matrix for transformations in 3-space?

5. (2 points) How are the actual electrons coming out of the red, green, and blue electron guns of a color monitor different?

Problem 2: Display Devices (12 points)

In order to allow more time for transmitting and displaying each pixel, the American broadcast television standard (NTSC) uses an "interlaced" type of refresh, in which each video frame is broken into two "fields," each containing one-half of the picture. The two fields are "interlaced" in the sense that each field contains every other scan line: all odd-numbered scan lines are displayed in the first field, and all even-numbered scan lines are displayed in the second.

The purpose of an interlaced scan is to place some new information in all areas of the screen at a high enough rate to avoid flicker, while allowing the hardware more time for accessing and displaying each pixel.

1. (1 point) If the video controller displays each field in $1/60^{th}$ of a second, what is the overall frame rate for displaying the entire screen?

2. (2 points each) An interlaced refresh works well as long as adjacent scan lines display similar information. In which parts, if any, of the following images would you expect to see flicker on an interlaced display (and briefly mention why):

- A single pixel wide horizontal white line on a black background?

- A single pixel wide vertical white line on a black background?

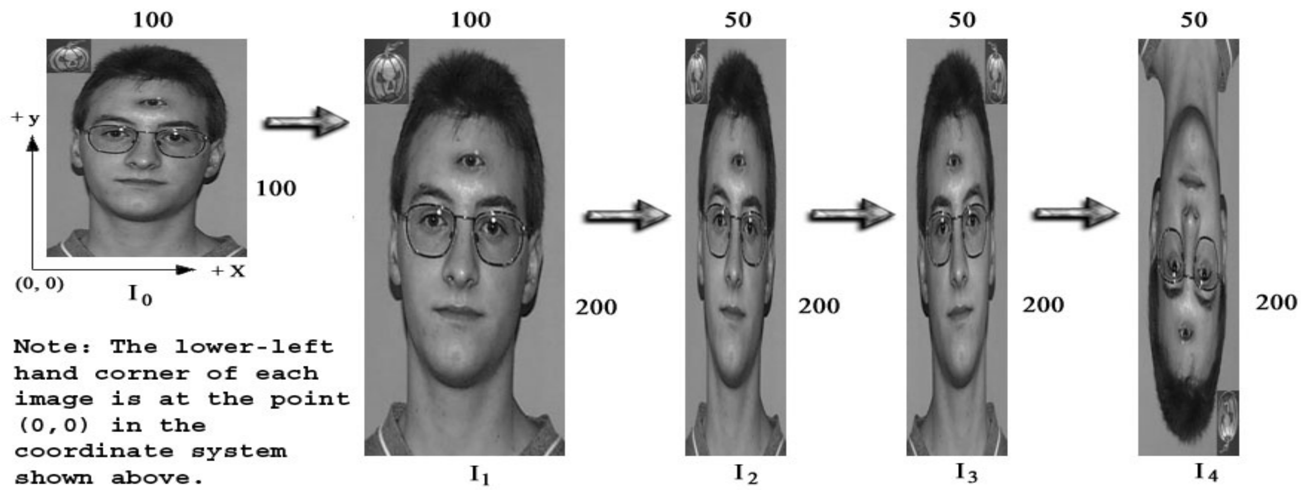
- A checkerboard of black and white, where each black or white square is 8 x 8 pixels?

- A checkerboard of black and white, where each black or white square is a single pixel?

3. (3 points) On interlaced displays, there is a very noticeable artifact when objects on the screen are moving fast. What is this artifact, and how will it appear if the video is of a white box moving horizontally on a black background?

Problem 3: Image Transformations (16 points)

Image I_0 below is a picture of a scary Halloween monster. I_4 is the result after several image transformations have been applied to I_0. I_1, I_2, and I_3 are the intermediate images. The numbers next to each image are its height and width in pixels.



Your job is to derive the intermediate transformations that were used to create the final image. Each intermediate transformation is of the following form:

$I_1(x, y) = I_0(\mathbf{a}(x, y)) = I_0(a_x(x, y), a_y(x, y))$
$I_2(x, y) = I_1(\mathbf{b}(x, y)) = I_1(b_x(x, y), b_y(x, y))$
$I_3(x, y) = I_2(\mathbf{c}(x, y)) = I_2(c_x(x, y), c_y(x, y))$
$I_4(x, y) = I_3(\mathbf{d}(x, y)) = I_3(d_x(x, y), d_y(x, y))$

Note that when we say a_x we are talking about the x-component of the vector-valued function $\mathbf{a}(x, y)$. We are NOT talking about the partial derivative of **a** with respect to x! And the notation $I_1(x, y)$ means "the intensity value of the pixel at coordinates (x, y) in image I_1". Hence when the transformation function $\mathbf{a}(x, y)$ is applied to a point in I_1, the result is another point in the image I_0. So we see that $\mathbf{a}(x, y)$ maps points in I_1 to points in I_0, $\mathbf{b}(x, y)$ maps points in I_2 to points in I_1, $\mathbf{c}(x, y)$ maps points in I_3 to points in I_2, and $\mathbf{d}(x, y)$ maps points in I_4 to points in I_3.

1. (1 point each) Now write out the x and y components of the four transformation functions:

$\mathbf{a}(x, y) =$ (____________________ , ________________________)

$\mathbf{b}(x, y) =$ (____________________ , ________________________)

$\mathbf{c}(x, y) =$ (____________________ , ________________________)

$\mathbf{d}(x, y) =$ (____________________ , ________________________)

2. (2 points) It should be evident that we can combine multiple image operations into a single transformation. Write out a transformation **h**(x, y) that combines the scaling operations represented by **a**(x, y) and **b**(x, y) so that $I_2(x, y) = I_0(\mathbf{h}(x, y)) = I_0(h_x(x, y), h_y(x, y))$:
Hint: What does the function a(b(x, y)) represent?

h(x, y) = (____________________ , _______________________)

3. (2 points) Now use the same technique you used in question 2 to derive a single transformation function **f**(x, y) that can be used to transform I_0 into I_4, such that $I_4(x, y) = I_0(\mathbf{f}(x, y)) = I_0(f_x(x, y), f_y(x, y))$:

f(x, y) = (____________________ , _______________________)

Bravo! You've managed to condense the four transformations into one new transformation that performs both scaling and horizontal/vertical flipping!

4. (3 points) ***In this example***, does the end result depend on the order in which the intermediate transformations are applied? Why or why not? Give an example to support your argument.
Hint: Don't think too hard about the equations you derived. Think about the actual operations that you are performing on the images.

5. (3 points) ***In general***, if you have a transformation that is composed of intermediate transformations (as was our example), does the order in which the intermediate transformations are applied matter? Why or why not? Give an example to support your argument. ***Hint: Affine transformations are a subset of the various transformations that can be performed in this manner.***

6. (2 points) Under what circumstances does the order in which the transformations are applied ***not*** matter? Justify your answer.

Problem 4: Perspective Projections (8 points)

Recall the transformation matrix for perspective projection:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix}$$

1. (4 points) If we first apply a uniform scale to all points before performing the projection indicated by this matrix, how will the projected points be different from the case where no scale is applied? Show your reasoning mathematically and explain the result geometrically (in words and/or with a sketch). ***Hint: it may be useful to think about what this looks like in two dimensions instead of three.***

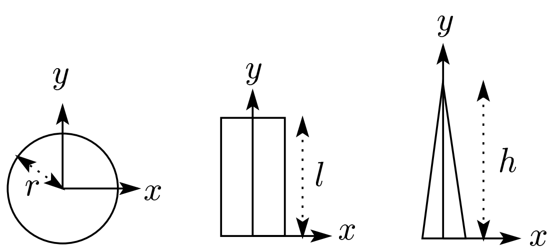
2. (4 points) Using the perspective projection matrix above, we observed in class that any set of parallel lines will converge and intersect at a vanishing point. In fact, some parallel lines will ***not*** intersect, i.e., they will remain parallel, even after undergoing this perspective projection. Which lines are these? Geometrically speaking, what do all sets of these lines have in common?

Problem 5: Hierarchies (12 points)

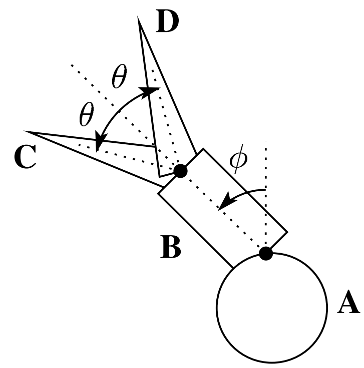
Suppose you want to model the pincher arm shown below. The pincher is made of four parts, labeled **A**, **B**, **C**, and **D** and each part is drawn as one of the three primitives shown below.

The following transformations are also available to you:

- R(t) – rotate by t degrees (counter clockwise)
- T(a, b) – translate by (a, b)



Primitives



Pincher diagram

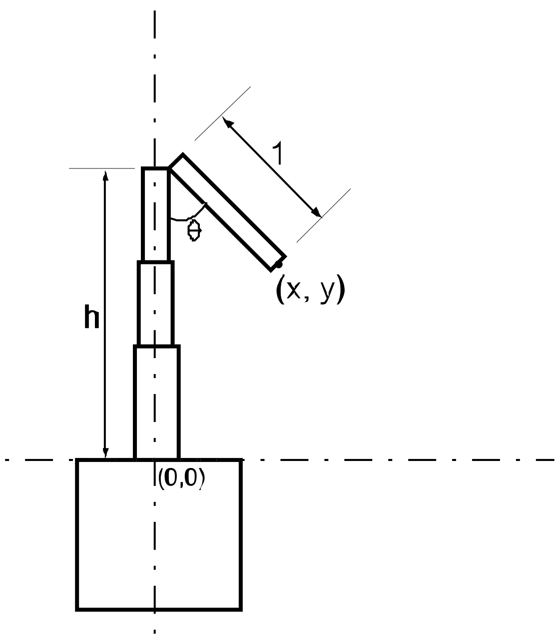
1. (9 points) Construct a tree to specify the pincher that is rooted at **A**. Along each of the edges of the tree, write expressions for the transformations that are applied along that edge, using the notation given above (you do not need to write out the matrices). Remember that order is important!

2. (3 points) Write out the full transformation expression for the part labeled **C**.

Problem 6: Inverse Kinematics (12 points)

In the hierarchies we have studied in class so far, we have dealt only with forward kinematics; that is, if we want to determine a particular position of a limb, we specify joint angles for all the joints such that the limb is positioned in the way we want.

In certain cases, however, it would be useful to instead specify coordinate positions of the limbs; for example, if we wanted a character to pick up a coffee cup, it would be useful to specify the hand position in the same (x, y, z) world coordinates that the cup's position is defined in. This is known as inverse kinematics, and in order for it to work properly, we must be able to determine the joint angles that would result in the correct hand position. In this problem, we will explore such a solver for a simple case.



The robot arm at left has a hierarchy rooted at (0, 0). The upper arm (labeled h) is telescoping, while the lower arm has the fixed length 1. Assume also that there are no constraints on the length of the telescoping arm. Answer the following questions:

1. (3 points) Suppose we set the point (x, y) to be equal to $\left(\frac{1}{2}, 2-\frac{\sqrt{3}}{2}\right)$. What would the angle **θ** and the length h be then?

2. (3 points) Is this the only solution? Why?

3. (6 points) Suppose **(x, y)** is some arbitrary point, and invert the equations. That is, solve for the joint angle **θ** and the upper arm length h in terms of the **(x, y)** coordinates.

Problem 7: Affine Transformations (3 points)

$$\mathbf{A}=\begin{bmatrix}1 & 0 & 0 & 6\\ 0 & 1 & 0 & 3\\ 0 & 0 & 1 & 2\\ 0 & 0 & 0 & 1\end{bmatrix}\qquad \mathbf{B}=\begin{bmatrix}2 & 0 & 0 & 0\\ 0 & 2 & 0 & 0\\ 0 & 0 & 2 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\qquad \mathbf{C}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & .5 & -.9 & 0\\ 0 & .9 & .5 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

1. (1 point each) As discussed in class, any three-dimensional affine transformation can be represented with a 4x4 matrix. Match the matrices above to the following transformations (not all blanks will be filled):

____ Translation

____ Rotation about the x-axis

____ Rotation about the y-axis

____ Rotation about the z-axis

____ Shearing with respect to the x-y plane

____ Shearing with respect to the y-z plane

____ Uniform Scaling

____ Reflection

Problem 8: Image Filters (6 points)

1. (3 points) Fill in the entries of **G**, below. Leave the border pixels of **G** blank and just compute the center 3x3 block.

- **H** is a 3x3 kernel
- **F** is an input image
- **G** is the result of applying the filter **H** to the image **F** using cross-correlation

$$\mathbf{H}\ \begin{array}{|c|c|c|}\hline a & b & c \\ \hline d & e & f \\ \hline g & h & i \\ \hline \end{array} \quad \otimes \quad \mathbf{F}\ \begin{array}{|c|c|c|c|c|}\hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} \quad = \quad \mathbf{G}\ \begin{array}{|c|c|c|c|c|}\hline \ & \ & \ & \ & \ \\ \hline \ & \ & \ & \ & \ \\ \hline \ & \ & \ & \ & \ \\ \hline \ & \ & \ & \ & \ \\ \hline \ & \ & \ & \ & \ \\ \hline \end{array}$$

2. (3 points) Suppose we replaced cross-correlation with convolution. Below, give the result of convolving **H** with **F**.

$$\mathbf{H}\ \begin{array}{|c|c|c|}\hline a & b & c \\ \hline d & e & f \\ \hline g & h & i \\ \hline \end{array} \quad \star \quad \mathbf{F}\ \begin{array}{|c|c|c|c|c|}\hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline \end{array} \quad = \quad \mathbf{G}\ \begin{array}{|c|c|c|c|c|}\hline \ & \ & \ & \ & \ \\ \hline \ & \ & \ & \ & \ \\ \hline \ & \ & \ & \ & \ \\ \hline \ & \ & \ & \ & \ \\ \hline \ & \ & \ & \ & \ \\ \hline \end{array}$$

Extra Credit: Rotations as Shear Transformations (10 points)

You are ***not*** required to do this problem! But extra points are good things, so we encourage you to give it a try.

In 2D, a rotation transformation by angle θ can be specified as a series of shear transformation matrices. Give these matrices, or if it can't be done, prove it.