Homework #1

Displays, Image Processing,

Affine Transformations, Hierarchical modeling

Assigned: Thursday, April 8th

Due: Wednesday, April 21st
at the beginning of class

Directions: Please provide short written answers to the following questions, using this page as a cover sheet. Feel free to discuss the problems with classmates, but please ***answer the questions on your own.***

Name:____

Problem 1: Short answers (10 points)

Provide short answers to each of the following questions:

a) (3 points) What is double-buffering and what is its purpose?

b) (3 points) How do you normalize a convolution filter, and what is the purpose of doing so?

c) (4 points) How would you compute the unit-length normal to a triangle in 3D with vertices A, B, and C, specified according to the right-hand rule (where curling the fingers of your right hand from A to B to C will leave your thumb pointing along the normal direction)? What happens if A, B, and C are co-linear?

Problem 2: Color LCD displays (20 points)

LCD displays operate on the principle of polarizing light at the entrance to each crystal, twisting the polarization by a voltage controlled amount, then passing the resulting light through a final polarizer. The end result is that the light coming out of the LCD has a particular intensity and (fixed) polarization. The color of the light reaching the viewer is controlled with a color filter (red, green, or blue) over each crystal.

Light coming out of the LCD has "linear polarization," characterized by an angle θ, with a corresponding polarization vector:

$$\mathbf{p} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$

A polarizing filter also has an orientation β with polarization vector:

$$\mathbf{f} = \begin{bmatrix} \cos\beta \\ \sin\beta \end{bmatrix}$$

The filter only passes the component of the input polarization aligned with filter's polarization. Specifically, it passes a fraction α of the light:

$$\alpha = |\mathbf{f} \bullet \mathbf{p}| = |\mathbf{f}^T \mathbf{p}| = |\cos\beta\cos\theta + \sin\beta\sin\theta|$$

Now, suppose, through a quirk of manufacturing, all the blue cells for an LCD display have polarization rotated 90 degrees with respect to all the green and red cells. Let's say that the blue polarization is horizontal ($\theta = 0^o$), and the green and red polarizations are vertical ($\theta = 90^o$). *Justify your answers* to each of the following questions. When giving a color as an answer, it is sufficient to just give the (R,G,B) coordinates.

a) (3 points) If we display solid white – (R,G,B) = (1.0, 1.0, 1.0) – and hold against the screen a polarizer in the vertical ($\beta = 90^o$) orientation, what color would we see? (We assume in this problem that each color channel ranges from 0.0 to 1.0, so (1.0, 1.0, 1.0) is bright white.)

b) (3 points) What color would we see if we then rotated the polarizer to a diagonal ($\beta = 45^o$) orientation?

c) (3 points) What color would we see if we then rotated the polarizer to a horizontal ($\beta = 0^o$) orientation?

d) (3 points) If we keep the filter oriented as in **c**), what color would we see if we now displayed solid green, i.e., (R,G,B) = (0, 1, 0)?

e) (8 points) Suppose now that you were to create a 3D LCD display (without the manufacturing quirk). Your display must work with glasses that are passive and based on linear polarization; i.e., one eye has a vertical polarizer and the other has a horizontal polarizer, no electronics in the glasses. How would you re-design the LCD to enable presenting stereo pair image streams (including renderings of moving 3D objects) to a viewer wearing these glasses? How would you modify the framebuffer? Note: there are multiple possible answers to this question; you only need to come up with one. Please be specific about your design.

Problem 3: Image processing (24 points)

In this problem, you will consider several convolution filtering operations and their behaviors. You do not need to worry about flipping filters before sliding them across images; i.e., assume filters are pre-flipped. In addition, assume that the *y*-axis points up, so that the line *x*=*y* is a diagonal line running from the bottom-left corner of an image to the upper right corner of the image.. *For each sub-problem, justify your answer.*

a) (2 points) The image you're editing is too dark, and you decide you need to amplify the value of each pixel by a factor of 2. Suggest a convolution filter that will double the value at each pixel of the image without changing it in any other way. (Technically, after scaling pixel values, they could be out of range; assume that any needed clamping will be taken care of later, after filtering).

b) (4 points) While taking a photograph with your digital camera, you fail to hold the camera steady, and it translates downward while the shutter is open. You discover this later when you see that horizontal edges, in particular, have been blurred a bit (an effect called "motion blur"). You decide to filter the image so that horizontal edges are sharpened, but vertical edges are unchanged. Suggest a single convolution filter that does this.

c) (4 points) After thinking a little more about the previous picture, you decide that motion blur is cool, and you want to apply it to another image. In this case, though, you want to simulate the effect of a camera translating diagonally along the *x*=*y* direction while the shutter is open. Suggest a convolution filter that would accomplish some diagonal blurring along that direction by averaging across *m* pixels.

d) (4 points) Describe a non-constant image that, when convolved with your diagonal blur filter from (c), would be unchanged by the filter. (You may ignore the boundaries.)

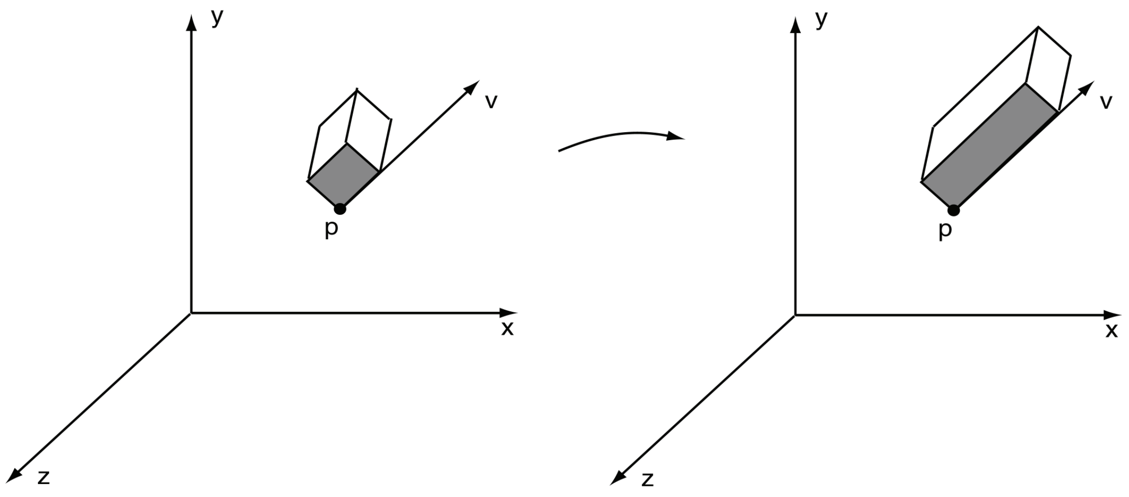
e) (10 points) Suppose you pad the boundary of an image in some way that allows you to compute output values for every pixel being filtered by a convolution filter. For an image of dimensions *n* x *n* and a filter of dimensions *m* x *m*, how many output pixels will be influenced by input pixels "hallucinated" beyond the boundary of the image? For simplicity, assume that *m* is odd. However, *m* and *n* may otherwise have arbitrary positive values.

Problem 4. Rotations and orthogonal matrices (8 points)

In class, we determined the number of degrees of freedom of a 3x3 rotation matrix by observing that its column vectors are each of length one and each orthogonal to the other column vectors. More concretely, for a rotation matrix $\mathbf{R} = [\mathbf{u}\ \mathbf{v}\ \mathbf{w}]$, we know that $\mathbf{u} \cdot \mathbf{u} = \mathbf{v} \cdot \mathbf{v} = \mathbf{w} \cdot \mathbf{w} = 1$, and $\mathbf{u} \cdot \mathbf{v} = \mathbf{u} \cdot \mathbf{w} = \mathbf{v} \cdot \mathbf{w} = 0$. However, going in the other direction, a matrix that meets these constraints is not necessarily a rotation matrix. Rather these length and orthogonality constraints are the constraints for an *orthogonal matrix*. All rotation matrices are orthogonal matrices, but not vice versa. In particular, an orthogonal matrix can include a reflection, so that the coordinate axes are no longer "right-handed" after the transformation. Using cross products and/or dot products among the **u**, **v**, and **w** vectors, devise a test to determine if a 3x3 orthogonal matrix is a rotation matrix. *Justify your answer* by explaining your reasoning.

Problem 5. 3D affine transformations (26 points)

The basic scaling matrix discussed in lecture scales only with respect to the x, y, and/or z axes. Using the basic translation, scaling, and rotation matrices, specify how to build a transformation matrix that scales along any ray in 3D space. This new transformation is determined by the ray origin **p** = (p_x, p_y, p_z) and direction vector **v** = (v_x, v_y, v_z), and the amount of scaling s_v. For clarity, a diagram has been provided, showing a box being scaled with respect to a given ray. Your answer should work for *any* ray, not just the case shown in the picture.



You can use any of the following standard matrices (from lecture) as building blocks: canonical axis rotations $R_x(\alpha)$, $R_y(\beta)$, $R_z(\gamma)$, scales $S(s_x,s_y,s_z)$, and translations $T(t_x,t_y,t_z)$. You don't need to write out the entries of the 4x4 matrices. It is sufficient to use the symbols given above, supplied with the appropriate arguments. All scale factors are strictly positive. You must compute the angles of any rotations required. Note that you may require inverse trigonometric functions, and you should assume that $\cos^{-1}(x)$ outputs a range of $[0..\pi]$, and that $\sin^{-1}(x)$ and $\tan^{-1}(x)$ each outputs a range of $[-\pi/2..\pi/2]$.

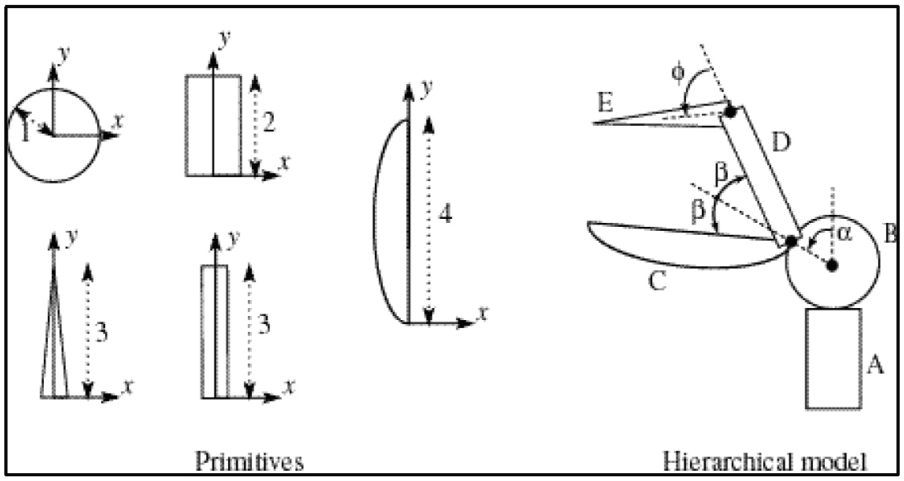
There are many possible solutions to this problem. To constrain the space of answers, and to give you a solution hint: you must cause the **v** direction to align with the y-axis at some stage of your solution.

Show your work, using words and drawings as needed to support your answer.

Problem 6. Hierarchical modeling (12 points)

Suppose you want to create the hierarchical model shown below. The model is comprised of five parts, labeled **A**, **B**, **C**, **D,** and **E**, and each part is drawn as one of five primitives given below (they are already scaled to the correct sizes). The following transformations are available to you:

- R(θ) – rotate by θ degrees (counter clockwise)
- T(a, b) – translate by $[a\ \ b]^T$



(a) (10 points) Construct a tree to specify the hierarchical model where the nodes of the tree should be labeled **A**, **B**, **C**, **D**, and **E**. Along each edge, write out the product of matrices that should be performed when traversing that edge. Insert numerical values (i.e., for primitive sizes) where available. You can use the matrix notation above; you do *not* need to write out any 3x3 (2D affine) matrices.

(b) (2 points) Write out the full transformation expression to be applied to the part labeled **E**.