Homework #1

Framebuffers, Image Processing,

Affine Transformations, Hierarchical Modeling

Assigned: Thursday, April 5th

Due: Wednesday, April 18th
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 and show your work.***

Name:____

Problem 1: Alpha compositing (26 points)

The alpha channel is used to control blending between colors. The most common use of alpha is in "the compositing equation"

$$\mathbf{C} = \alpha\,\mathbf{F} + (1\text{-}\ \alpha)\,\mathbf{B} \quad \text{or} \quad \begin{bmatrix} C_R \\ C_G \\ C_B \end{bmatrix} = \alpha \begin{bmatrix} F_R \\ F_G \\ F_B \end{bmatrix} + (1-\alpha) \begin{bmatrix} B_R \\ B_G \\ B_B \end{bmatrix}$$

where α is the blending coefficient, **F** is the foreground color, **B** is the background color, and **C** is the composite color. In film production, compositing is a common operation for putting a foreground character into a new scene (background). The challenge faced with real imagery is to extract per pixel alpha and foreground color from a live action sequence, to enable compositing over a new background.

(a) (6 points) When filming an actor, a color **C** is observed at each pixel. If the three observed color channel values C_R, C_G, and C_B are the only knowns at a given pixel, how many unknowns remain in the compositing equation at that pixel? Treating each color channel separately, how many equations are there at the pixel? Is it generally possible to estimate all the unknowns under these circumstances? [Note: we are treating each pixel in isolation, so in each of these problems, you should just be thinking in terms of a single pixel.]

(b) (4 points) To assist the process of extracting the desired **F** and α values, the actor may be filmed against a known background, typically solid blue or green. If the components of **B** are known, how many unknowns remain at a given pixel? Is it possible, in general, to estimate **F** and α under these circumstances?

(c) (8 points) When filming the original Star Wars trilogy, the starships were assumed to contain only shades of gray and were filmed against a solid blue background. Thus, at a given pixel, the visual effects people could assume $\mathbf{F} = [L\ L\ L]^T$, where L is a shade of gray, and $\mathbf{B} = [0\ 0\ 1]^T$, where color channel values are in the range [0...1]. Given an observed color $\mathbf{C} = [C_R\ C_G\ C_B\ ]^T$ at a pixel, compute α and L in terms of the color components of **C**. You should comment on how to handle the case when $\alpha = 0$. Show your work. [Note: if the answer is not unique, just provide one possible solution.]

(d) (8 points) Suppose you had the luxury of two consecutive images of a stationary foreground subject against a blue and a green background in succession, $\mathbf{B} = [0\ 0\ 1]^T$ and $\mathbf{G} = [0\ 1\ 0]^T$, thus recording two colors, **C** and **D**, respectively, at each pixel. You would then have to consider two color compositing equations $\mathbf{C} = \alpha\,\mathbf{F} + (1\text{-}\ \alpha)\,\mathbf{B}$ and $\mathbf{D} = \alpha\,\mathbf{F} + (1\text{-}\ \alpha)\,\mathbf{G}$. Solve for α and the components of the foreground color, F_R, F_G, and F_B at a given pixel in terms of the components of **C** and **D**. Show your work. [Note: if the answer is not unique, just provide one possible solution.]

Problem 2: Image processing (26 points)

In this problem, you will consider several convolution filtering operations and their behaviors. *For each sub-problem, justify your answer.*

(a) (4 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.

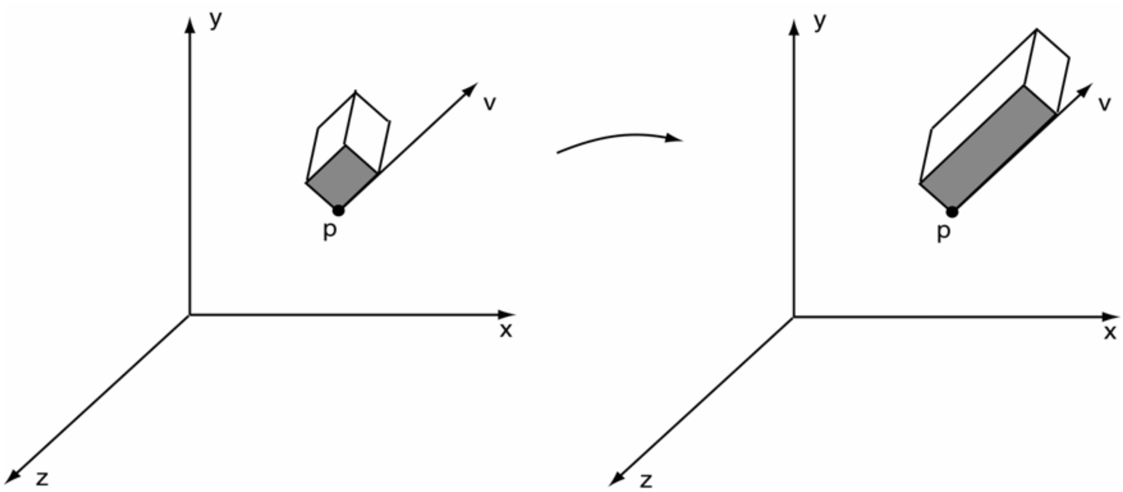
(c) (4 points) After thinking a little more about the previous picture, you decide that motion blur is cool, and you want to apply it to another image. In this case, though, you want to simulate the effect of a camera translating diagonally along the x=y direction while the shutter is open. Suggest a 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 3. 3D affine transformations (28 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 $\mathbf{p} = (p_x, p_y, p_z)$ and direction vector $\mathbf{v} = (v_x, v_y, v_z)$, 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 (a.k.a., Euler angle 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 $\mathbf{v}$ 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.

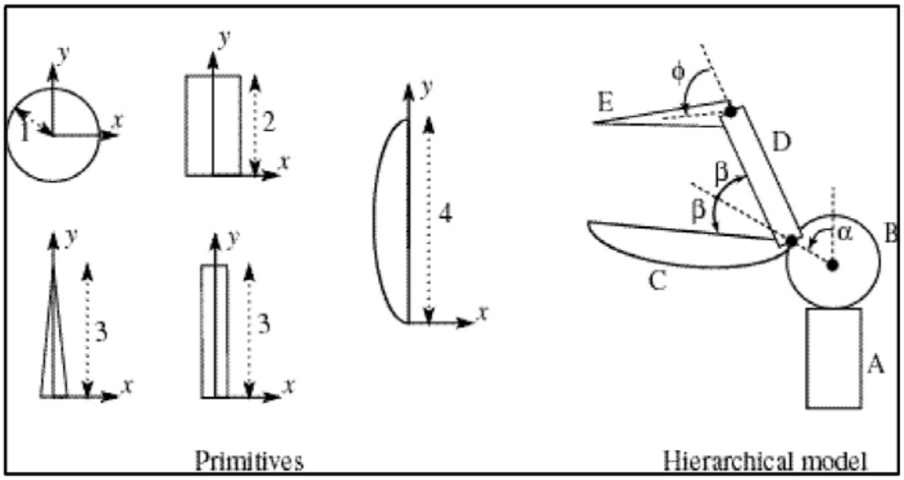
Problem 4. Rotations and orthogonal matrices (8 points)

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. A notable exception is a reflection matrix, which is an orthogonal matrix, but not a rotation matrix. Using cross products and/or dot products among the $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ vectors, devise a test for whether a 3x3 orthogonal matrix is a rotation matrix. *Justify your answer* by explaining your reasoning.

Problem 5. 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.

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