Homework #1

Image Processing, Affine Transformations,

Hierarchical modeling, Perspective

Due: Wednesday, October 26th
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 (17 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) (3 points) Suppose, for each pixel of a grayscale image, you only average pixels in a 5x5 neighborhood that have values within +/-5% of the pixel being filtered. Could this be written as a convolution filter? Explain.

d) (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?

e) (4 Points) Consider a pair of three dimensional vectors, *u* and *v*, which are of non-zero length and not parallel to each other. Which of the following is True, False, or Nonsense (i.e., involving an operation that cannot be performed):

$$(v \times u) \times u = u \times (u \times v)$$

$$(v \bullet u) \times u = u \times (u \bullet v)$$

$$(v \times u) \bullet u = u \bullet (u \times v)$$

$$\frac{u}{\|u\|} \bullet v = u \bullet \frac{v}{\|v\|}$$

You do **not** need to justify your answer for part (e).

Problem 2: 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, the x-axis points to the right, and the lower left corner of the image is at (0,0). *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 4. Suggest a convolution filter that will quadruple 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 from left to right while the shutter is open. You discover this later when you see that vertical edges, in particular, have been blurred a bit (an effect called "motion blur"). You decide to filter the image so that vertical edges are sharpened, but horizontal edges are unchanged. Suggest a single convolution filter that does this.

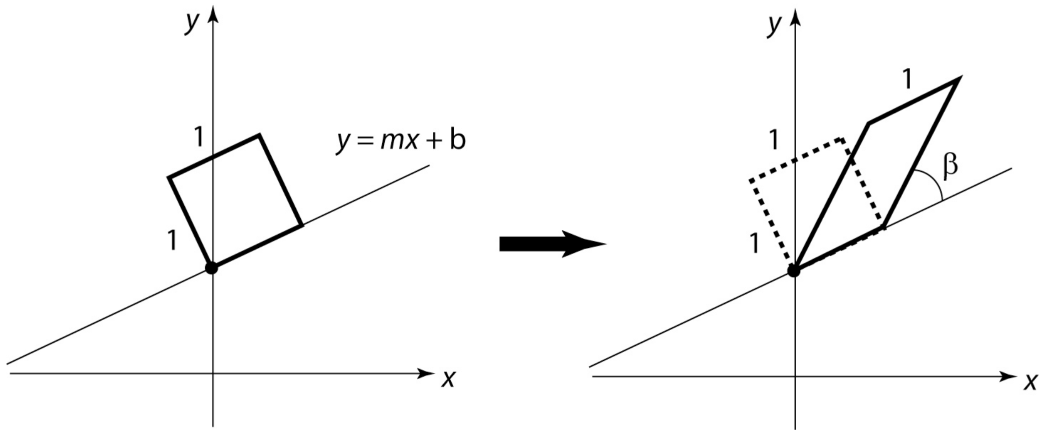
c) (4 points) After thinking a little more about the previous picture, you decide that motion blur is cool, and you want to apply it to another image. In this case, though, you want to simulate the effect of a camera translating diagonally along the $x = -y$ direction while the shutter is open. Suggest 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: Affine Transformations (20 points)

In this problem, you will determine the affine transformation needed to perform a shear with respect to the line $y = mx + b$, while holding fixed the point where the line intersects the y-axis, as illustrated here:



On the left is a unit square sitting on the line before shearing. After transformation, the unit square becomes the solid parallelogram shown on the right; the dotted outline is the original square, for reference. The 1's next to line segments (solid or dotted) indicate their lengths.

Assume you have the following transformations available to you:

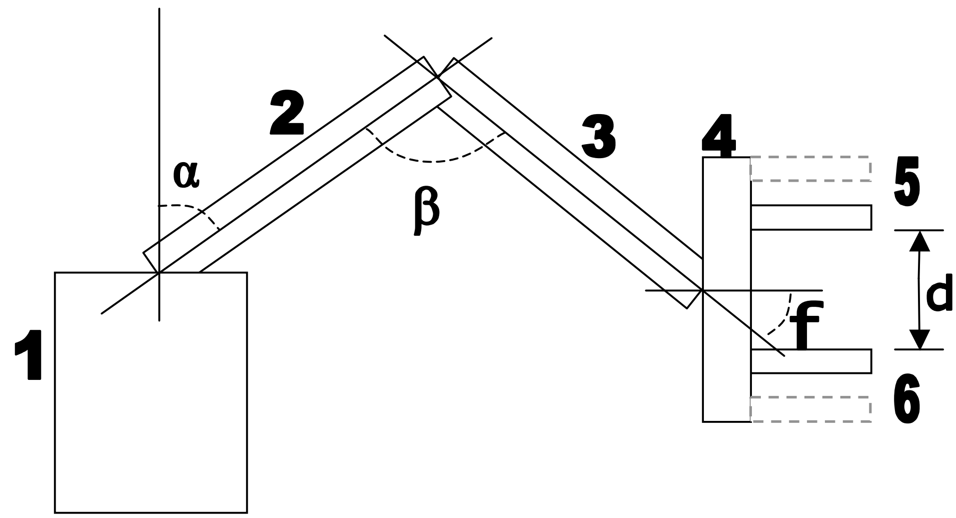
$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad T(c,d) = \begin{pmatrix} 1 & 0 & c \\ 0 & 1 & d \\ 0 & 0 & 1 \end{pmatrix} \quad Sh_x(b) = \begin{pmatrix} 1 & b & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad Sh_y(a) = \begin{pmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

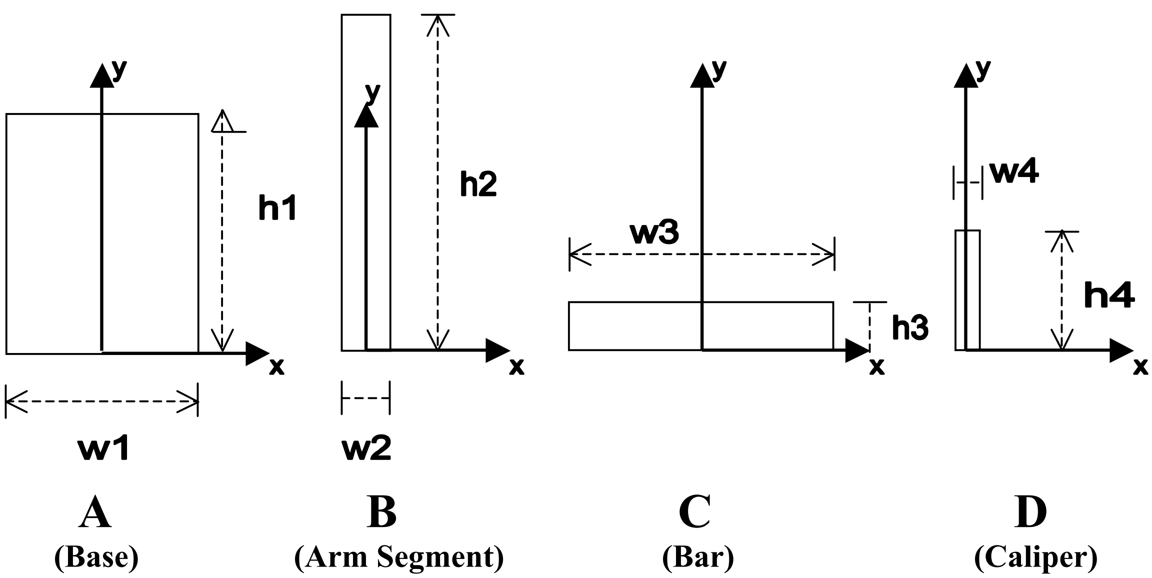
i.e., rotation, translation, and shears in the x-direction and y-direction.

Determine a matrix product that will perform the shear illustrated in the figure using some combination of these transformations. You do ***not*** need to write out the 3 x 3 matrices, just their symbolic references and the arguments they take. Your solution should be parameterized by the slope m, y-intercept b, and angle β. You may assume that m is finite (the line is not vertical). Justify your answer by drawing the result of each transformation step, before concatenating

Problem 4: Hierarchical modeling (19 points)

Suppose you want to model the robot arm with calipers, shown below. The arm is made out of six parts (**1-6**), and each part is drawn as one of the four primitives (**A-D**).





The following transformations are also available to you:

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

Note that the angle parameters in the illustration above are each positive in the current configuration in the illustration ($\alpha \approx +60^o$, $\beta \approx +110^o$, $\phi \approx +40^o$), though of course the model can be re-posed by changing these parameters.

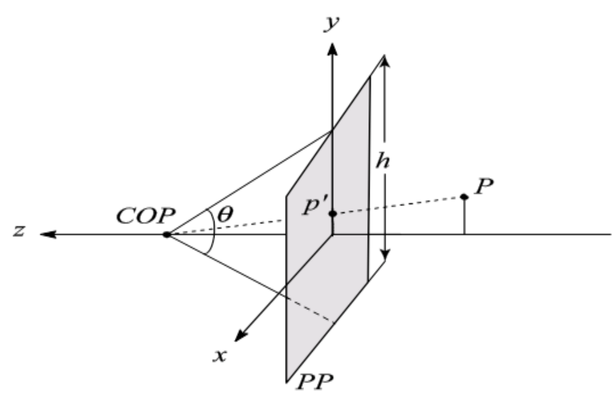
Problem 4: Hierarchical Modeling (cont'd)

a) (15 points) Construct a tree to specify the robot arm that is rooted at **1**. 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! Your tree should contain a bunch of boxes (or circles) each containing one part number (1…6); these boxes should be connected by line segments, each labeled with a corresponding transformation that connects child to parent. The calipers are distance *d* away from each other, and *equidistant from the center of the bar they are attached to*. (Note: Each caliper (objects **5** & **6**) is attached to the bar (object **4**) so you will need to reflect this fact in your tree.)

b) (4 points) Write out the full transformation expression for the part labeled **5**.

Problem 5: Projections (20 Points)

In class, we derived the matrix for perspective projection for a viewer sitting at the origin looking down the –*z* axis. Suppose now that we assume that the projection plane *PP* passes through the origin, and the viewer is at the *COP* somewhere on the +*z* axis, still looking in the –*z* direction, as illustrated in the figure below.



In addition, assume that we parameterize the projection in terms of viewing angle θ that measures the angle **from top to bottom** subtended by the image, which is of height *h*, and lies in the projection plane. The size of the image (and thus *h*) is constant throughout this problem.

a) (8 Points) What is the projection matrix that we would use to map a point $P = [x\ y\ z\ 1]^T$ to $p' = [x'\ y'\ 1]^T$. (The *w* coordinate for *p'* will be 1 after doing the perspective divide.) Your answer should be in terms of θ and *h* only. Show your work.

b) (5 Points) Solve for and write out the projection matrix when $\theta = 0$. Also solve for *p'* when $\theta = 0$. What kind of projection does this case correspond to?

c) (2 Points) What happens to *p'* as θ goes to 180^o? Why?

d) (5 Points) Suppose the point *P* starts at position $[x_0\ y_0\ z_0\ 1]^T$ and is initially imaged with viewing angle θ_0, so that its projection is $[x_0'\ y_0'\ 1]^T$. Now suppose we want the projection of *P* to remain constant while now varying the viewing angle. We can do this by moving the camera closer to or further from *P* as we vary θ. Moving the camera closer or further amounts to changing *P*'s *z*-coordinate (and no other coordinates of *P*). How must *P*'s *z*-coordinate vary as a function of θ to achieve the desired effect?