Homework #1

Displays, Image Processing,

Affine Transformations, Hierarchical Modeling

Assigned: Thursday, Ocober 12th

Due: Thursday, October 26th
at the beginning of class

Directions: Please provide short written answers to the following questions on your own paper. Feel free to discuss the problems with classmates, but *please* ***follow the Gilligan's Island rule*, answer the questions on your own, and show your work.***

Please write your name on your assignment!

*** The Gilligan's Island Rule:** This rule says that you are free to meet with fellow student(s) and discuss assignments with them. Writing on a board or shared piece of paper is acceptable during the meeting; however, you should not take any written (electronic or otherwise) record away from the meeting. After the meeting, engage in a half hour of mind-numbing activity (like watching an episode of Gilligan's Island), before starting to work on the assignment. This will assure that you are able to reconstruct what you learned from the meeting, by yourself, using your own brain.

Clarifications / corrections posted to class

Problem 5

- **Part (a):** You should follow the structure that you see on slide 11 of the Hierarchical Modeling lecture. In that slide, the robot arm is rooted by the part "Base" and has an edge that connects it to its geometry "Cylinder" and another edge the connects to the rest of the model. "Base" is the same type as a "part number" in problem 5a (i.e., just a label), and "Cylinder" is the same as geometry (A or B) in this problem (i.e., the thing that actually gets drawn). Note that the transformation on the edge to the geometry is needed to transform the object both to get its proportions correct but also to move its origin as needed so that it correctly articulates when, e.g., later adjusting its rotation angle in the hierarchy.

- **Part (b):** The problem statement should read: "Write out the full transformation expression to be applied *to the geometry* associated with part **4**." This has been corrected in the re-posted pdf.

Problem 1: Alpha compositing (18 points)

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

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

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) (4 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 solve for 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) (2 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 solve for **F** and α under these circumstances?

(c) (6 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) (6 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 $\mathbf{C} = \alpha\,\mathbf{F} + (1-\alpha)\,\mathbf{B}$ and $\mathbf{D} = \alpha\,\mathbf{F} + (1-\alpha)\,\mathbf{G}$. 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 (16 points)

Suppose we have two filters:

Filter A

0	0	0
-1	0	1
0	0	0

Filter B

0	1	0
0	2	0
0	1	0

a) (3 points) In class, we described a simple and intuitive version of an x-gradient filter: [-1 1]. When applied, this filter computes the *finite difference* gradient in the x-direction, essentially solving for $\partial f / \partial x \approx \Delta f / \Delta x$, where $\Delta x = 1$ and pixels are one unit distance from their neighbors. Filter A, by contrast, is used to compute what is known as the *central difference* x-gradient. Although it cannot be normalized in the usual way, since its values sum to zero, it is usually multiplied by a coefficient of ½. Why?

b) (3 points) Normalize B. What effect will this normalized filter have when applied to an image?

c) (4 points) Compute $A*B$, using A and B from the ***original*** problem statement, i.e., ***without*** using the scale factors described in (a) and (b). You can treat B as the filter kernel and assume that A is zero outside of its support. You do *not* need to show your work. [Aside: convolution is commutative ($A*B=B*A$), so you would get the same answer by using A as the filter kernel. But, you would have to remember to "flip" the kernel to get $\widetilde{A}[i,j] = A[-i,-j]$. We've asked you instead to use B as the filter kernel, but since B is symmetric, i.e., $\widetilde{B}[i,j] = B[-i,-j] = B[i,j]$, you don't need to worry about flipping.]

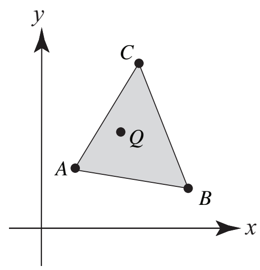
d) (2 points) Compute $A*B$, now using A and B after scaling them according to (a) and (b).

e) (4 points) If we apply the result of (c) or (d) to an image f, we are computing $(A*B)*f$. Convolution is associative, so we would get the same results as computing $A*(B*f)$. In other words, we're filtering the image with B, and then filtering the result with A.

- Why would it be desirable to apply B before computing the gradient (as opposed to not applying B at all)?
- Why might applying B be better than applying a filter B' that is filled with a full 3x3 set of positive coefficients (e.g., changing the 0's to 1's in the B filter given above), rather than just a single column of coefficients?

Problem 3: Triangle coordinates (24 points)

Consider triangle Δ_{ABC} and a point Q depicted below:



A, B, C, and Q lie in the x-y plane, so, **neglecting the homogeneous coordinate**, we can write out their **3D** coordinates as:

$$A = \begin{bmatrix} A_x \\ A_y \\ 0 \end{bmatrix} \quad B = \begin{bmatrix} B_x \\ B_y \\ 0 \end{bmatrix} \quad C = \begin{bmatrix} C_x \\ C_y \\ 0 \end{bmatrix} \quad Q = \begin{bmatrix} Q_x \\ Q_y \\ 0 \end{bmatrix}$$

The last coordinate is the z coordinate in this case, and we know that $A_z = B_z = C_z = Q_z = 0$. Note that Q is depicted as lying inside of Δ_{ABC}, but you should not assume that it does unless stated otherwise in a sub-problem. Assume that $A \neq B \neq C$, i.e., the triangle is not "degenerate." Further assume that if you curl the fingers of your right hand from A to B to C, your thumb will point in the direction of the positive z-axis.

(a) (3 points) Using cross and/or dot products, devise a test (with the help of equations) to determine if point Q lies inside of Δ_{ABC}. You should assume that the edges and vertices of the triangle are part of the interior of the triangle. [You do not need to expand any cross or dot products in your answer, but you may do so if it helps you.]

(b) (2 points) Using cross and/or dot products, compute the unit-length normal to Δ_{ABC}. Your solution should work for 3D points in general, i.e., not depend on the fact that these points lie in the x-y plane. We will use the right-hand rule for triangles, which means that as you curl the fingers of your right hand from A to B to C, your thumb will point in the direction of the normal. [You do not need to expand any cross or dot products in your answer, but you may do so if it helps you.]

(c) (3 points) Using cross and/or dot products, compute the area of the triangle, Area(Δ_{ABC}). This time you **do** need to expand all cross and/or dot products based on the elements of A, B, and C. Multiply out all terms; e.g., an expression like $(a+b)(c+d)$ should be expanded to $ac+ad+bc+bd$.

For the remainder of the problem, we will safely ignore the z coordinate and work in **2D** only, but **now we will keep track of the affine coordinate** (w=1 for all points). We can now represent the triangle vertices and the point Q as:

$$A = \begin{bmatrix} A_x \\ A_y \\ 1 \end{bmatrix} \quad B = \begin{bmatrix} B_x \\ B_y \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} C_x \\ C_y \\ 1 \end{bmatrix} \quad Q = \begin{bmatrix} Q_x \\ Q_y \\ 1 \end{bmatrix}$$

Again, to be clear, the last coordinate is now the affine w coordinate, which, for affine points, is always 1; i.e., $A_w = B_w = C_w = Q_w = 1$.

(d) (2 points) Suppose we create a 3x3 matrix $[A\ B\ C]$, i.e., a matrix with columns filled by A, B, and C. Write out this matrix, explicitly filling in the elements of the matrix in terms of the elements of A, B, and C, and compute its determinant, $\det[A\ B\ C]$. Again, multiply out all terms.

(e) (1 point) Based on your answers to (c) and (d), what is Area(Δ_{ABC}) in terms of $\det[A\ B\ C]$?

(f) (1 point) In general, we can express Q as a weighted sum of A, B, and C; i.e., $Q = \alpha A + \beta B + \gamma C$, where α, β, and γ are scalars. In order for this to be a proper affine combination (a weighted sum of affine points that yields an affine point), what constraint is placed on α, β, and γ? Explain.

(g) (3 points) Now we will work on solving for α, β, and γ. Write out a matrix equation of the form $M\,\mathbf{p} = \mathbf{r}$:

$$\begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} r_0 \\ r_1 \\ r_2 \end{bmatrix}$$

where M is a 3x3 matrix, $\mathbf{p}$ is the column vector of unknowns, i.e., $\mathbf{p} = [\alpha\ \beta\ \gamma]^T$ and $\mathbf{r}$ is a column vector with three elements. I.e., explicitly write out the matrix equation, filling in the elements of M, $\mathbf{p}$, and $\mathbf{r}$, in terms of α, β, γ, and the elements of A, B, C, and Q. (Do not apply the matrix, just set up the equation.) Hint: you can expand $Q = \alpha A + \beta B + \gamma C$ explicitly in terms of the elements of A, B, C, and Q and the result should be equivalent to your matrix equation.

(h) (2 points) We can solve for $\mathbf{p}$ using Cramer's rule. In particular, for a matrix equation $M\,\mathbf{p} = \mathbf{r}$ as above, we can solve for $\mathbf{p}$ as ratios of determinants:

$$p_0 = \frac{\det\begin{bmatrix} r_0 & m_{01} & m_{02} \\ r_1 & m_{11} & m_{12} \\ r_2 & m_{21} & m_{22} \end{bmatrix}}{\det\begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & m_{22} \end{bmatrix}} \qquad p_1 = \frac{\det\begin{bmatrix} m_{00} & r_0 & m_{02} \\ m_{10} & r_1 & m_{12} \\ m_{20} & r_2 & m_{22} \end{bmatrix}}{\det\begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & m_{22} \end{bmatrix}} \qquad p_2 = \frac{\det\begin{bmatrix} m_{00} & m_{01} & r_0 \\ m_{10} & m_{11} & r_1 \\ m_{20} & m_{21} & r_2 \end{bmatrix}}{\det\begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & m_{22} \end{bmatrix}}$$

Note how the denominator is always the determinant of M, and the numerator is the determinant of a matrix that consists of M with one of the columns replaced with the elements of $\mathbf{r}$. Based on your answer to (g), re-write these same "Cramer's rule" equations using α, β, γ, and the elements of A, B, C, and Q.

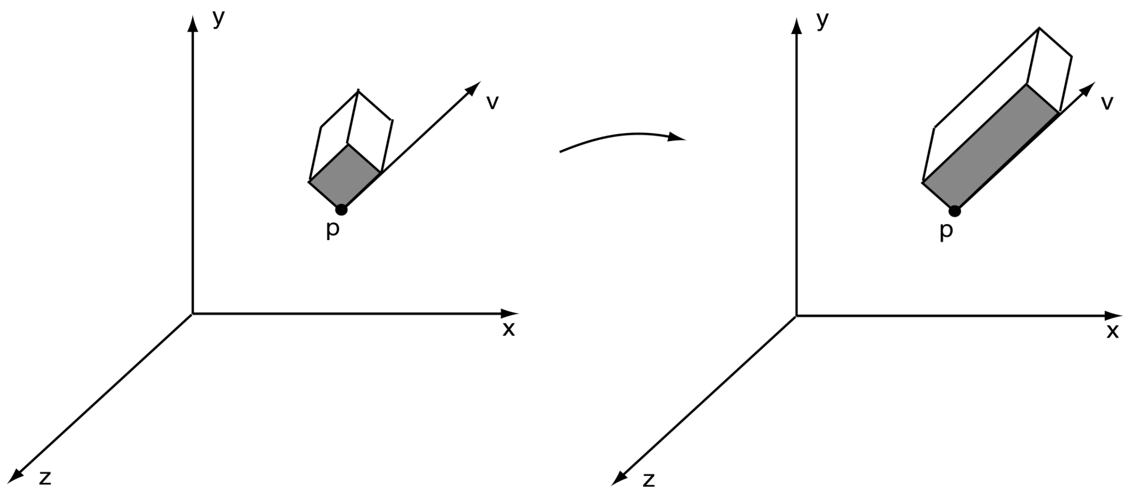
(i) (3 points) Assume Q is inside of Δ_{ABC}. In this case, all of the determinants in (h) are positive. Based on your answer to (e), re-write your answer to (h) in terms of areas of triangles.

(j) (2 points) Suppose $Q = B$. What should α, β, and γ be? Justify your answer in terms of areas of triangles.

(k) (2 points) Suppose Q is halfway between A and C. What should α, β, and γ be? Justify your answer in terms of areas of triangles.

Problem 4. 3D affine transformations (22 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 may *not* assume that **v** is of unit length.



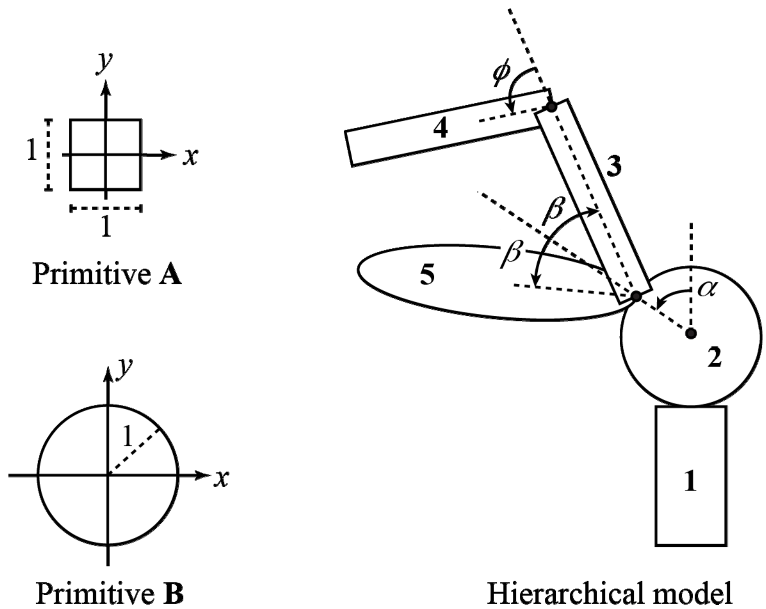
You can use any of the following standard matrices (from lecture) as building blocks: canonical axis rotations $R_x(\alpha)$, $R_y(\beta)$, $R_z(\gamma)$, scales $S(s_x,s_y,s_z)$, and translations $T(t_x,t_y,t_z)$. You don't need to write out the entries of the 4x4 matrices. It is sufficient to use the 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 5. Hierarchical modeling (20 points)

Suppose you want to model a robot arm "pincer" illustrated below. The model is comprised of 5 parts, using primitives **A** and **B** and is controlled by parameters α, β, and ϕ.



As part of your solution to this problem, you will need to define the instance transformations that should be applied to a given primitive on the left so that it is the same shape as a desired primitive on the right; all scale factors will be integers (1, 2, 3, …) or simple fractions (1/2, 1/3, 1/4, …); use a ruler to determine scale factors. Transformations available to you:

- R(θ) – rotate by θ degrees (counter clockwise)
- T(a, b) – translate by $\begin{bmatrix} a \\ b \end{bmatrix}$
- S(s_x, s_y) – scale the x-component by s_x and scale the y-component by s_y

(a) (14 points) Construct a tree to describe this hierarchical model using part **1** as the root. Each node will contain either a part number (**1**…**5**) or a reference to the canonical geometry that will be drawn (**A** or **B**). Along each of the edges of the tree, write an expression for the transformations that are applied along that edge to connect parent to child. Write all transformations using the notation above; you do not need to write out the matrices, just the symbolic references to them, and their arguments. Remember that the order of transformations is important! Show your work wherever the transformations are not "obvious." Assume that the center of part **1** sits on the origin in world coordinates. The tree must have one or more branches in it. If two parts are connected physically, then they should be connected in the tree, as long as you don't end up with any node with two parents.

(b) (2 points) Write out the full transformation expression to be applied *to the geometry* associated with part **4**.

Problem 5. (cont'd)

(c) (2 points) Suppose we now want to compute the location of the tip of the part labeled **5**, where the tip is at the end of ellipse **5** farthest from part **2**. One way to do this is to imagine adding one more primitive **6** that consists of an infinitesimal point at the origin (in its local coordinates, as was the case for each of the primitives in the drawing above). We can then add this primitive to the hierarchical model, attached to **5**, with a suitable transformation. What is the transformation expression to be applied to **6**?

(d) (2 points) To compute the location of the point at the tip of part **5**, we would just apply the transformation expression to the location of the point **6** (from part (c)) in its local coordinates, which is (in 2D affine coordinates) $[0\ 0\ 1]^T$. Suppose $\alpha = 270^{\circ}$ and $\phi = \beta = 0$. Evaluate the location of the tip of **5** using the matrices in part (c), writing out the matrices in full to compute the result. You may multiply the matrices together first before applying to point **6** or apply them to **6** sequentially to get your result. Show your work.