

CSE 152 Assignment 1
Spring 2006
Due Thursday, April 27, in class

Instructions :

- Attempt all questions.
- Please comment all your Matlab code adequately.
- There is a bonus question at the end of each question, answers to which will be awarded up to 3 points each, although more subjectively.

1 Homogeneous Coordinates and Vanishing Points

In class, we discussed the concept of homogeneous coordinates. In this example, we will confine ourselves to the real 2D plane. A point $(x, y)^\top$ on the real 2D plane can be represented in homogeneous coordinates by a 3-vector $(wx, wy, w)^\top$, where $w \neq 0$ is any real number. All values of $w \neq 0$ represent the same 2D point. Dividing out the third coordinate of a homogeneous point (x, y, z) converts it back to its 2D equivalent: $\left(\frac{x}{z}, \frac{y}{z}\right)^\top$.

Consider a line in the 2D plane, whose equation is given by $ax + by + c = 0$. This can equivalently be written as $\mathbf{l}^\top \mathbf{x} = 0$, where $\mathbf{l} = (a, b, c)^\top$ and $\mathbf{x} = (x, y, 1)^\top$. Noticing that \mathbf{x} is a homogeneous representation of $(x, y)^\top$, we define \mathbf{l} as a homogeneous representation of the line $ax + by + c = 0$. Note that the line $(ka)x + (kb)y + (kc) = 0$ for $k \neq 0$ is the same as the line $ax + by + c = 0$, so the homogeneous representation of the line $ax + by + c = 0$ can be equivalently given by $(a, b, c)^\top$ or $(ka, kb, kc)^\top$ for any $k \neq 0$.

All points (x, y) that lie on the line $ax + by + c = 0$ satisfy the equation $\mathbf{l}^\top \mathbf{x} = 0$, thus, we can say that a condition for a homogeneous point \mathbf{x} to lie on the homogeneous line \mathbf{l} is that their dot product is zero, that is, $\mathbf{l}^\top \mathbf{x} = 0$. We note this down as a fact:

Fact 1: A point \mathbf{x} in homogeneous coordinates lies on the homogeneous line \mathbf{l} if and only if

$$\mathbf{x}^\top \mathbf{l} = \mathbf{l}^\top \mathbf{x} = 0$$

Now let us solve a few simple examples:

- Give at least two homogeneous representations for the point $(3, 5)^\top$ on the 2D plane, one with $w > 0$ and one with $w < 0$. **[1 point]**
- What is the equation of the line passing through the points $(1, 1)^\top$ and $(-1, 3)^\top$ [in the usual Cartesian coordinates]? Now write down a 3-vector that is a homogeneous representation for this line. **[2 points]**

We will now move on to consider the intersection of two lines. We make the claim that: "The (homogeneous) point of intersection, \mathbf{x} , of two homogeneous lines \mathbf{l}_1 and \mathbf{l}_2 is $\mathbf{x} = \mathbf{l}_1 \times \mathbf{l}_2$, where \times stands for the vector (or cross) product".

- (c) In plain English, how will you express the condition a point must satisfy to lie at the intersection of two lines? Armed with this simple condition, and using **Fact 1**, can you briefly explain why $\mathbf{l}_1 \times \mathbf{l}_2$ must lie at the intersection of lines \mathbf{l}_1 and \mathbf{l}_2 ? [5 points]

In the following, we will use the above stated claim for the intersection of two lines.

- (d) Consider the two lines $x + y - 5 = 0$ and $4x - 5y + 7 = 0$. Use the claim you proved in question 1(c) to find their intersection in homogeneous coordinates. Convert this homogeneous point back to standard Cartesian coordinates. Is your answer compatible with what basic coordinate geometry suggests? [3 points]
- (e) Consider the two lines $x + 2y + 1 = 0$ and $3x + 6y - 2 = 0$. What is the special relationship between these two lines in the Euclidean plane? What is the interpretation of their intersection in standard Cartesian coordinates? [2 points]
- (f) Write the homogeneous representations of the above two lines and compute their point of intersection in homogeneous coordinates. What is this point of intersection called in computer vision parlance? [3 points]
- (g) Do questions 1(e) and 1(f) justify the claim in class, that homogeneous coordinates provide a uniform treatment of line intersection, regardless of parallelism? Briefly explain. [2 points]

Bonus question: Give (with justification) an expression for the homogeneous representation of the line passing through two homogeneous points \mathbf{x}_1 and \mathbf{x}_2 . [Hint: Try to construct an argument analogous to the one that justified the expression for the point of intersection of two lines in question 1(c).]

2 Histograms and Image Segmentation

Important note: In this problem, we will deal with only grayscale images. If you encounter an RGB color image anywhere, convert it to grayscale. (What is the Matlab command for this conversion?)

In this exercise, we will encounter a very useful construct called the *histogram* of an image. Informally, suppose you decide to divide the intensity range of an image into n bins. Then, $hist(j)$ is the number of pixels of a given image I that lie in the j -th bin, $1 \leq j \leq n$.

Matlab provides a function, $hist()$, that determines the histogram of a vector.

- (a) Learn the properties of the $hist$ function using Matlab's help documentation. [1 point]

- (b) Notice that *hist* in Matlab needs its input to be a vector of type double. Learn about type conversion in Matlab, especially, converting between *uint8* and *double* types. In Matlab, how can you efficiently vectorize a matrix? [2 points]
- (c) Take an input image of a scene with a large range of intensities (say, a very colorful scene) and display its histogram. What is the default number of bins in a Matlab histogram? Describe in a sentence how the appearance of the histogram changes as you increase the number of bins? [4 points]

Now that we are familiar with a histogram in Matlab, we will use it for *image segmentation*. Intuitively, segmentation refers to separating out “different regions” of an image, in our case, we will simply count the number of different regions in an image. Turn in well-commented Matlab code for the following parts (c), (d) and (e) along with answers to any specific questions asked. Feel free to include any figures or explanations.

- (d) Write a Matlab function that takes as input a 2D array containing intensities of the image *rectangles.jpg* and counts the number of regions in the image, based on the histogram of the image. You may or may not want to count the background as a region, please state your decision explicitly. [3 points]
- (e) Now use the same function, without any changes, to count the number of regions of the image *circles.jpg*. [1 point]
- (f) Let us now look at a slightly more difficult segmentation problem. Consider the input image *gradients.jpg*. By visual inspection, how many regions do you think a reasonable segmentation algorithm should output for this image? What is the output of the function you had written in part (d) for this image? If it is not the same as you would expect from your visual inspection of the image, plot the histogram for this image with 100 bins. Now write a function to count the number of regions. (*Hint*: You might need to use a suitable thresholding, based on your observation of the histogram values.) [4 points]

We will have a look at better segmentation techniques later in the class. It is an important subject in computer vision, and designing good algorithms for image segmentation as well as evaluating them are intensely investigated problems. **Bonus question:** Can you suggest some reasons why segmenting images of the world around us is considered a very difficult problem in general?

3 Camera Matrices and Rigid-Body Transformations

Consider a world coordinate system W , centered at the origin $(0,0,0)$, with axes given by unit vectors $\hat{\mathbf{i}} = (1,0,0)^\top$, $\hat{\mathbf{j}} = (0,1,0)^\top$ and $\hat{\mathbf{k}} = (0,0,1)^\top$. Recall our notation where boldfaces stand for a vector and a hat above a boldface letter stands for a unit vector.

- (a) Consider another coordinate system, with unit vectors along two of the orthogonal axes given by $\hat{\mathbf{i}}' = (0.9, 0.4, 0.1\sqrt{3})^\top$ and $\hat{\mathbf{j}}' = (-0.41833, 0.90427, 0.08539)^\top$.

Find the unit vector, $\hat{\mathbf{k}}'$, along the third axis orthogonal to both $\hat{\mathbf{i}}'$ and $\hat{\mathbf{j}}'$. Is there a unique unit vector orthogonal to both $\hat{\mathbf{i}}'$ and $\hat{\mathbf{j}}'$? If not, choose the one that makes an acute angle with $\hat{\mathbf{k}}$. [2 points]

- (b) Find the rotation matrix that rotates any vector in the $(\hat{\mathbf{i}}, \hat{\mathbf{j}}, \hat{\mathbf{k}})$ coordinate system to the $(\hat{\mathbf{i}}', \hat{\mathbf{j}}', \hat{\mathbf{k}}')$ coordinate system. [2 points]
- (c) What is the *extrinsic parameter matrix* for a camera that is located at a displacement $(-1, -2, -3)^\top$ from the origin of W and oriented such that its principal axis coincides with $\hat{\mathbf{k}}'$, the x-axis of its image plane coincides with $\hat{\mathbf{i}}'$ and the y-axis of the image plane coincides with $\hat{\mathbf{j}}'$? [2 points]
- (d) What is the *intrinsic parameter matrix* for this camera, if its focal length in the x-direction is 1050 pixels, aspect ratio is 1.0606, pixels deviate from rectangular by 0.573 degrees and principal point is offset from the center $(0, 0)^\top$ of the image plane to the location $(10, -5)^\top$? [2 points]
- (e) Write down the projection matrix for the camera described by the configuration in parts (c) and (d). [2 points]
- (f) Consider a plane, orthogonal to $\hat{\mathbf{k}}$, at a displacement of 2 units from the origin of W along the $\hat{\mathbf{k}}$ direction. Consider a circle with radius 1, centered at $(0, 0, 2)^\top$ in the coordinate system W . We wish to find the image of this circle, as seen by the camera we constructed in part (e). The following questions are to be attempted in Matlab and the code for each part should be turned in along with any figures and answers to specific questions. Explain your variable names (with comments). Feel free to supply any additional description/explanation to go with your code.
 - (i) Compute 10000 well-distributed points on the unit circle. One way to do this is to sample the angular range 0 to 360 degrees into 10000 equal parts and convert the resulting points from polar coordinates (radius is 1) to Cartesian coordinates. Display the circle, make sure that the axes of the display figure are equal. [2 points]
 - (ii) Add the z coordinate to these points, which is 2 for all of them. Make all the points homogeneous by adding a fourth coordinate equal to 1. [1 point]
 - (iii) Compute the projection of these homogeneous points using the camera matrix computed in part (e). Convert the homogeneous projected points to 2D Cartesian points by dividing out (and subsequently discarding) the third coordinate of each point. [2 points]
 - (iv) Plot the projected 2D points, again ensure that the axes of your plot are equal. What is the shape of the image of a circle? [2 points]

Bonus points if you can write the Matlab code for all sub-questions in part (f) without using any for-loops. A command useful here is *repmat*. Look up the Matlab help documentation on *repmat* for more details.