CS 231A Computer Vision Midterm

Out: 12:30pm, February 25, 2015

Due: 12:30pm, February 27, 2015

Full Name: _____

Question		Score
Transformations	(25 pts)	
Multiview Geometry	(25 pts)	
Hough Transform and Vanishing Points	(30 pts)	
Blob Detection	(20 pts)	
Total	(100 pts)	
Extra Credit	(10 pts)	

Welcome to the CS 231A Midterm Exam!

- The exam is designed to take 6 - 10 hours.
- The exam is open book and open notes. No collaboration is permitted either in person or online.
- The exam must be submitted to Scoryst by 12.30pm Friday 27th February. **The deadline is strict - make sure something is submitted by this time.** You may submit as many times as you like so submit early and revise your submission regularly. Scanning issues will not be accepted as an excuse for failing to submit. If you are having problems with Scoryst please email us your submission **before** the deadline.

I understand and agree to uphold the Stanford Honor Code during this exam.

Signature: _____ Date: _____

Good luck!

1 Transformations

In this question, we will explore some interesting low-level properties of transformations.

(a) In the lectures, we showed that a 3D rotation can be formed as the product of three matrices

$$R = R_x(\alpha)R_y(\beta)R_z(\gamma),$$

where

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}, R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}, R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

are rotations around the x, y, and z axis, respectively. This is called the X-Y-Z Euler angle representation, indicating the order of matrix multiplication.

However, representing rotations in this form can lead to ambiguities, since the result depends on the order in which the transforms are performed.

Let p' be the point obtained by rotating a point p with a rotation matrix R, so $p' = Rp$. Give an expression for p' obtained by rotating p in the following two ways:

- First rotate α around x axis, then rotate γ around z axis.
- First rotate γ around z axis, then rotate α around x axis.

Show that these rotations produce different values of p'.

(b) To avoid the representation ambiguity introduced in part a, we can fix the rotation order. However, even if the order of rotation is fixed, this rotation system may still result in a degenerate representation.

For instance, let α, β, and γ be three different angles. We can represent an arbitrary rotation with respect to the X-Y-X axes as the product $R_x(\alpha)R_y(\beta)R_x(\gamma)$ (note that the axes here are X-Y-X, not X-Y-Z). This representation of a rotation matrix should have three degrees of freedom. However if we set $\beta = 0$, something unusual happens. How many degrees of freedom are left? Hint: some trigonometric identities may be useful.

(c) To avoid the loss of a degree of freedom mentioned above, we can use an alternative rotation representation defined as follows:

$$R(\hat{\mathbf{n}}, \theta) = I + \sin\theta[\hat{\mathbf{n}}]_\times + (1 - \cos\theta)[\hat{\mathbf{n}}]_\times^2,$$

where $\hat{\mathbf{n}}$ is a unit vector and θ is the angle of rotation. Recall that $[\cdot]_\times$ is the matrix cross product operator (see Lecture 5).

Any valid rotation representation must satisfy some properties of rotation. Generally, for any rotation matrix R, we must have that $R^T R = I$. Prove that this equality holds for the rotation matrix R defined above. (Hint: Consider the rotation given by R^T.)

(d) Let us now consider points in 1-D. In homogeneous coordinates, a point in 1-D can be represented as $(x,\, 1)^T$, where $x \in \mathbb{R}$. More generally, such a point can be represented as $(kx,\, k)^T$ or generally as $(x_1,\, x_2)^T$.

A projective transformation of a point in 1D is represented by a 2×2 matrix,

$$\mathbf{x}' = H_{2\times 2}\mathbf{x},$$

where $H_{2\times 2}$ has a non-zero determinant.

Given 4 points on a line, the cross ratio is defined as

$$\text{Cross}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4) = \frac{|\mathbf{x}_1\mathbf{x}_2||\mathbf{x}_3\mathbf{x}_4|}{|\mathbf{x}_1\mathbf{x}_3||\mathbf{x}_2\mathbf{x}_4|},$$

where

$$|\mathbf{x}_i\mathbf{x}_j| = \det \begin{bmatrix} x_{i1} & x_{j1} \\ x_{i2} & x_{j2} \end{bmatrix}$$

with point $\mathbf{x}_i$ given by $(x_{i1},\, x_{i2})^T$ and $\mathbf{x}_j$ given by $(x_{j1},\, x_{j2})^T$.

Prove that the value of the cross ratio is invariant under any projective transformation of the line: ie show that if $\mathbf{x}' = H_{2\times 2}\mathbf{x}$, then

$$\text{Cross}(\mathbf{x}'_1, \mathbf{x}'_2, \mathbf{x}'_3, \mathbf{x}'_4) = \text{Cross}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4).$$

Equivalently, the cross ratio is invariant to the projective coordinate frame chosen for the line.

(e) Given the previous result, does the cross ratio remain invariant under rotation? Give a brief explanation.

(f) **Extra Credit**

Show that the result in (d) holds even when $\mathbf{x}_i' = \lambda_i H_{2\times 2}\mathbf{x}_i$, where points are scaled by arbitrary scale factors λ_i.

2 Multiview geometry

Assume that we have two cameras with camera matrices M_1 and M_2. Suppose that the cameras generate image I_1 and I_2, respectively, where p_1, p_2, p_3, p_4 and p_1', p_2', p_3', p_4' are corresponding points across the two images.

(a) Suppose

$$M_1 = K_1 \begin{bmatrix} R_1 & T_1 \end{bmatrix} \qquad M_2 = K_2 \begin{bmatrix} R_2 & T_2 \end{bmatrix} \tag{1}$$

where

$$\begin{aligned} K_1 &= K_2 = K \\ R_1 &= I = \text{Identity matrix} \\ R_2 &= R \\ T_1 &= T_2 = [0, 0, 0]^T \end{aligned} \tag{2}$$

Prove that the homography H defined by

$$p_i' = Hp_i \text{ where } i = 1, 2, 3, 4 \tag{3}$$

can be expressed as $H = KRK^{-1}$.

(b) **Continuing with our setup in part (a)**. Do $N > 3$ corresponding points necessarily need to belong to the same planar surface in 3D in order for the above result in part (a) to be true? Justify your answer.

(c) Now we **change** our setup of two cameras M_1 and M_2 as follows.

$$M_1 = K_1 \begin{bmatrix} R_1 & T_1 \end{bmatrix} \qquad M_2 = K_2 \begin{bmatrix} R_2 & T_2 \end{bmatrix} \tag{4}$$

where

$$\begin{aligned} K_1 &= K_2 = K \\ R_1 &= I = \text{Identity matrix} \\ R_2 &= R \\ T_1 &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\ T_2 &= t = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T \end{aligned} \tag{5}$$

All four image correspondences p_i p_i' are projections of four points P_i which lie on the same planar surface Π in 3D. Π is defined as the set of points X such that $Z^T X = 0$, where $Z = \begin{bmatrix} v^T & 1 \end{bmatrix}^T$ (v is a 3×1 vector). An illustration of this step can be found in Figure 1. Express the homography H (defined by $p' = Hp$) in terms of K, R, t, v and explain your derivation.

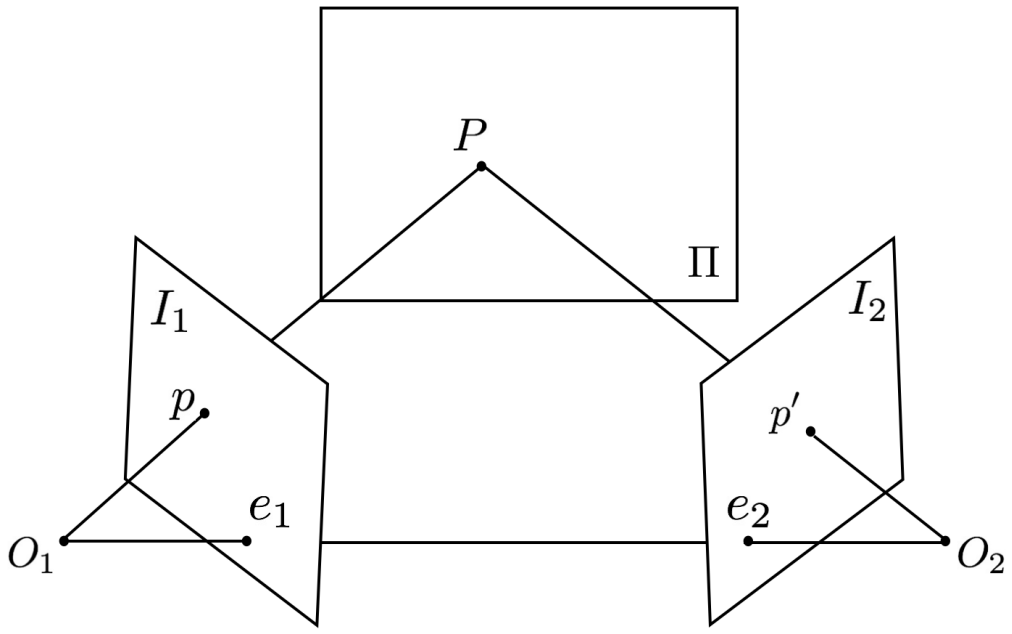


Figure 1: Camera setup used in part (c) (d) (e) and (f).

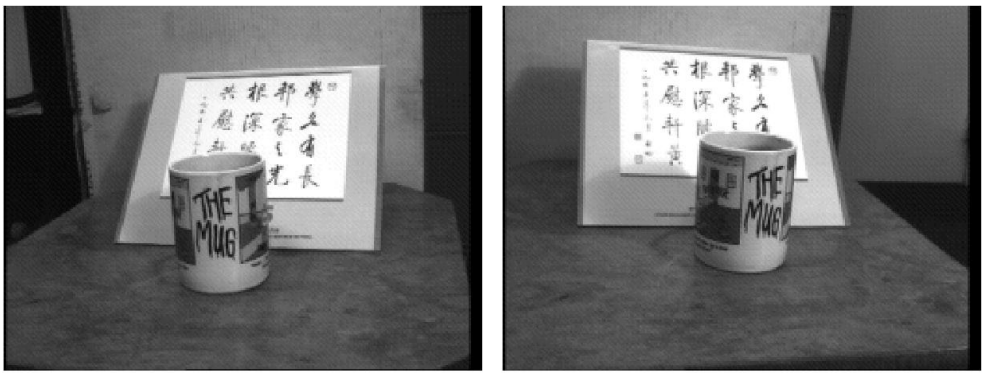
(d) **Continuing with our setup in part (c)**. Prove that the fundamental matrix F which satisfies $p^T F p' = 0$ can be expressed as

$$F = ([e_2]_\times H)^T$$

where e_2 (see Figure 1) is the epipole in the second image plane.

(e) **Continuing with our setup in part (c)**. In addition to the four correspondences on the plane, we are given two additional pairs of corresponding points $(p_5\, p_5', p_6\, p_6')$ that do NOT belong to the same planar surface Π. Show how to find epipole e_2 and explain your derivation. *Hint: e_2 can be expressed as the intersection of two lines.*

(f) **Continuing with our setup in part (c)**. From what you have shown in the previous parts, write a function `computeF.m` that estimates the fundamental matrix F using 6 correpondences (4 of them belong to the same 3D plane).



(a) Image 1 (b) Image 2

Figure 2: Two images from camera M_1 and M_2

We provide

- A dataset of six corresponding points from two images (Figure 2) under the camera setup shown in Figure 1. `p1.mat` includes six points from image 1 and `p2.mat` includes six corresponding points from image 2. The first four corresponding points are from the same 3D plane, and the last two corresponding points are out of that plane. You can run `loadData.m` to get familiar with the data.
- Function `computeH.m` that estimates a homography from at least 4 points.

For this part, you need to turn in

- An outline of the algorithm that you are using to compute F.
- Your code for function `computeF.m`.
- Your numerical results of F.

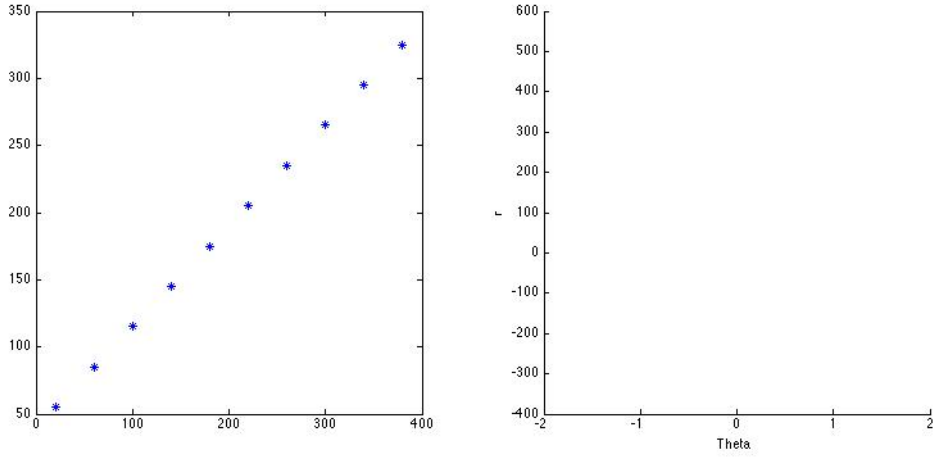
Hint:

- You don't need to (and should not) label additional points.
- You can verify your results by computing $p^T F p'$.

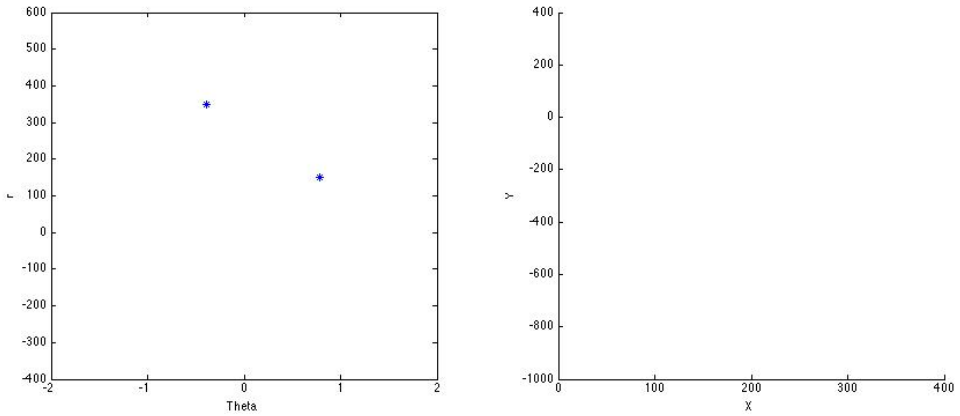
3 Hough Transform and Vanishing Points

For this problem, we wish to find the vanishing points of a scene by finding the intersection point of a set of parallel lines. In Problem Set 1, we selected a pair of parallel lines using 4 points and then extrapolated to find their intersection. This process is prone to noise and is sensitive to the selection of points (as many of you complained).
To improve this process, we will densely sample many points along a set of parallel lines and use the Hough Transform to better fit lines in the scene. If we parameterize a line using r and θ as presented in class, we can represent a line as $y = \frac{-cos(\theta)}{sin(\theta)}x + \frac{r}{sin(\theta)}$. As a result, each edge point will be mapped to a curve in the polar parameter space by the Hough Transform.

(a) We will begin by building some intuition about the Hough transform. For the given Cartesian points in the left plot, sketch the corresponding curves in the Hough space. A rough sketch is ok; ensure your sketch captures the fact that there are **10 points** that all **lie along the same line**.



(b) Once we identify peaks in the Hough space, we can map these back to the Cartesian space. For the given peaks in the Hough space (left plot below), sketch the corresponding lines in the Cartesian space. The sketches can be approximate.



(c) Assuming we densely sample the parallel lines in the scene that correspond to the same vanishing point, we can use the Hough Transform to identify the most likely set of lines. Since there could be multiple parallel lines in the scene, there could be multiple peaks in the Hough space.

Let the vanishing point lie at x_0, y_0. The lines in the image that intersect the vanishing point will form peaks in Hough space. Describe the curve in (θ, r) Hough space that all peaks will lie along. Express your answer as a function of the form $r_i = r(\theta_i, x_0, y_0)$.

(d) If the points selected were from lines that corresponded to more than one vanishing point, what would the peaks in the Hough space look like? Give your answer qualitatively, referencing your solution to (c).

(e) If we took the locations of the peaks described in part (d) and transformed them back into Cartesian space, how could we use a procedure similar to the Hough transform to find the location of the vanishing points?

(f) In the previous parts of this problem we overlooked the size of the bins in the Hough space. If the Hough space had bins of finite size to act as accumulators for the transformed points, describe the effects of bin size on the procedure described in parts (d) and (e), assuming that the original point sampling of the parallel lines described in part (a) was subject to noise. Specifically discuss the trade offs that must be considered when selecting bin size.

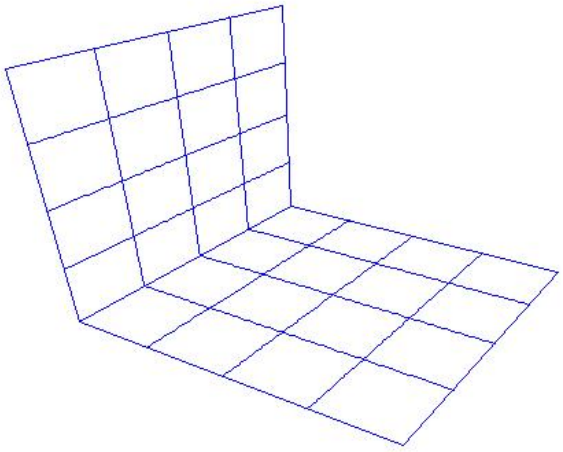


Figure 3: Scene with two planes containing parallel lines

(g) Suppose the sets of parallel lines described thus far in this problem belong to N planes, where the number of planes, N, is unknown. Each plane contains multiple sets of parallel lines with different directions. A simple illustration of this setup can be seen in Figure 3 with N=2 planes. How might we find N, the number of planes in the scene and their normal directions using techniques developed in earlier parts of this question? What additional information might you need? *Hint: What properties do vanishing points that belong to the same plane have? Are they geometrically related?*

(h) **Extra Credit**

One way to estimate the position (x_0, y_0) of the vanishing point is to fit a model of the form found in (c) to the n observed Hough peaks that lie at (r_i, θ_i). We could then use least squares to minimize the sum of squared errors between the measured r_i and that predicted by the model $r(\theta_i, x_0, y_0)$.

$$\min_{x_0, y_0} \sum_{i=1}^{n} (r_i - r(\theta_i, x_0, y_0))^2$$

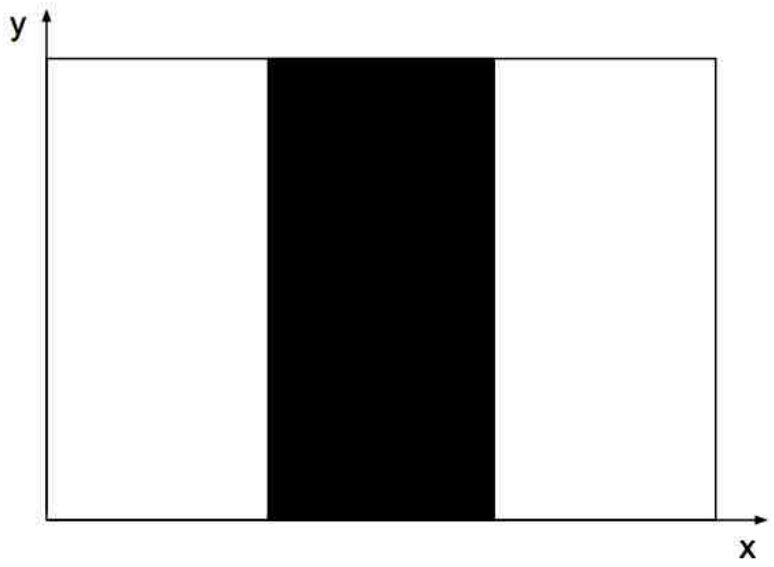
This is an elegant way to find a vanishing point just from the point measurements of the parallel lines in a scene.

Formulate an appropriate objective to describe this problem and solve it using least squares. For this part, you can assume that all peaks in Hough space correspond to lines intersecting at a single vanishing point (x_0, y_0) in the image. What is the expression for the optimal vanishing point? *Note: it is OK to leave matrix inverses in your solution, but try to simplify other terms as much as possible.*

4 Blob detection

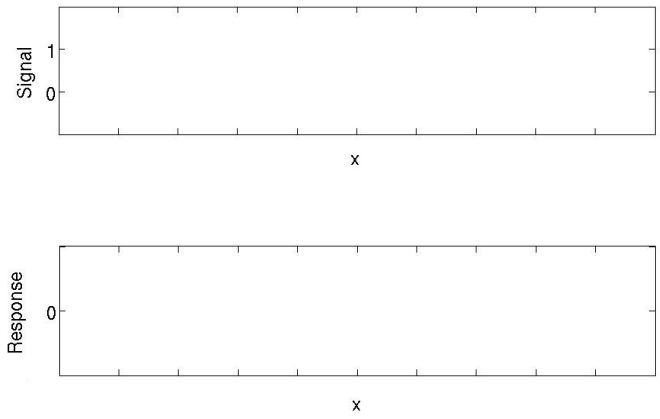
In lecture we learned how we can use a convolution with a Laplacian of a Gaussian to find edges in an image. In this question we will explore this idea in more detail.

(a) Suppose that you have the following black-and-white image. In our image encoding, white is represented by the value of 1 and black the value of 0.



Sketch a one-dimensional slice of the intensity (i.e. brightness) of this image in the x-direction. Then sketch the result of convolving this signal with the Laplacian of a Gaussian and mark where the edges are located in the plot of the convolution. Your sketches do not need to be scaled correctly because we have not specified the width of the Gaussian. Please ignore border-effects, i.e. ignore the effect of convolving the kernel with the borders of the image.

Draw your answer here:



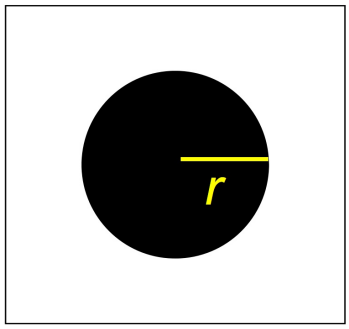
(b) Given the following formula for a 2D Gaussian centered at (0, 0):

$$g = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)}$$

take derivatives and compute the formula for the 2D Laplacian of a Gaussian.

(c) Suppose that you convolve a 2D Laplacian of a Gaussian with the following image, where the circle has radius r. Using your answer from parts a and b, find the characteristic scale of this image, in terms of the radius of the circle r.

Hint: At the characteristic scale, the convolution will have the same zeros as the Laplacian convolution kernel.



image