

Assignment #1

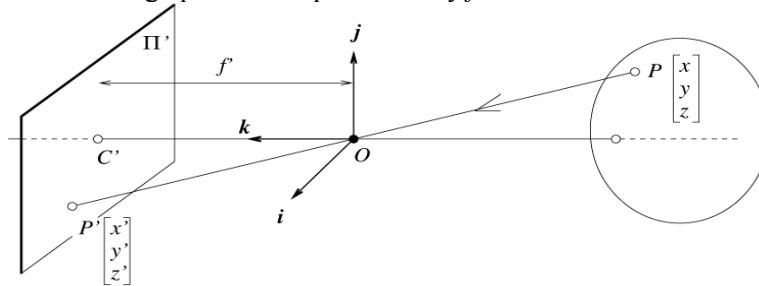
(Due date: 10/23/2012)

1. Perspective Projection [2pts]

Consider a perspective projection where a point

$$P = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

is projected onto an image plane Π' represented by $f' = -k$ as shown in the following figure.



The first, second, and third coordinate axes are denoted by i, j , and k respectively.

Consider the projection of a ray in the world coordinate system

$$Q = \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix} + t \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}$$

where $1 \leq t \leq \infty$. Calculate its two endpoints.

2. Rigid Transformation [3pts]

a) Consider a rigid transformation where a point A is rotated about the k -axis by the angle $\theta = 65^\circ$ (degrees) and then translated by

$$t = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}$$

to another point A' . When the two points A and A' are represented in the homogeneous coordinate system by

$$A = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \text{ and } A' = \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix}$$

Respectively, write a 4×4 matrix M such that

$$A' = MA$$

b) If this same point is A is translated by t then rotated by θ would the resulting M matrix be the same? Explain your answer.

3. Thin Lens Equation [3pts]

An illuminated arrow forms a real inverted image of itself at a distance $w = 40\text{cm}$, measured along the optic axis of a convex thin lens (see Figure 1). The image is just half the size of the object.

- How far from the object must the lens be placed?
- What is the focal length of the lens?
- If the index of refraction for the lens is $n = 1.5$, what is the lens radius?
- Suppose the arrow moves 5cm to the right while the lens and image plane remain fixed. This will result in an out of focus image; what is the radius of the corresponding blur circle formed from the tip of the arrow on the image plane assuming the diameter of the lens is d ?

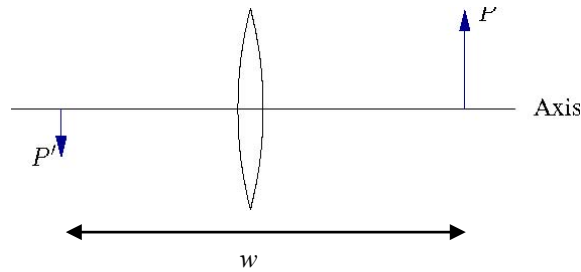


Figure 1: Problem 3. setup.

4. Affine Projection [2pts]

Consider an affine camera and two rectangles in 3D space. These two rectangles are coplanar and have two colinear sides. Consider the four corner points (A , B , C and D) of the larger rectangle. Now consider three additional points on the same plane (E , F and G). Two of these points (E and F) lie somewhere on the lines AB and AD respectively. The last point G lies inside the rectangle such that lines EG and FG are perpendicular and lies on the same plane as A , B , C , D , E and F . This will form two co-planar rectangles ($ABCD$ and $EBFG$). Now consider the area of the two rectangles on the image plane. Show that the ratio of areas between the two rectangles in the image plane is constant regardless of their orientation relative to the camera.

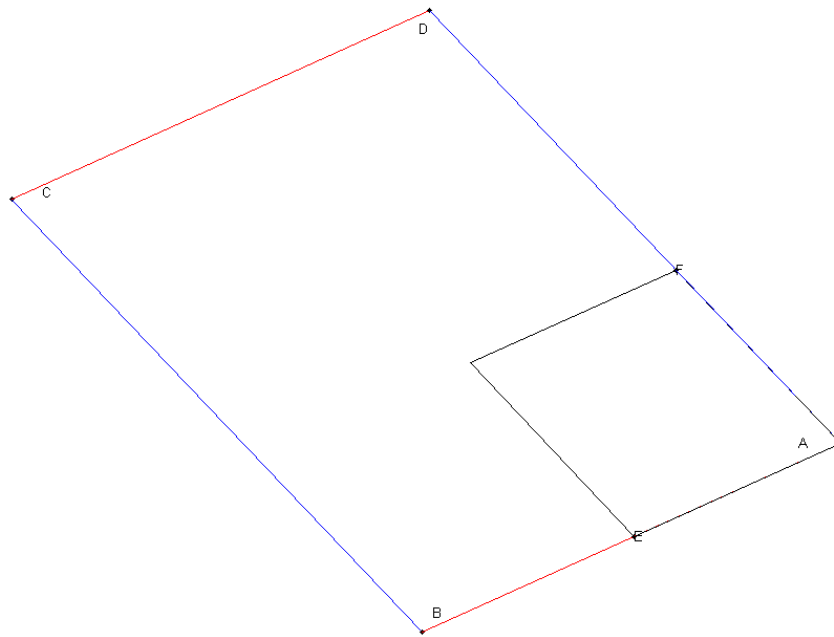


Figure 2: Problem 4.

5. Affine Pose Estimation [5pts]

An affine camera transforms 3D (homogeneous) points $\mathbf{x} = (x, y, z, 1)^T$ according to,

$$\mathbf{x}' = M\mathbf{x} \quad (1)$$

where

$$M = \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

is the camera matrix and $\mathbf{x}' = (x', y', 1)^T$ is the mapping onto the image plane.

Given a set of *corresponding* points $\mathbf{x}$ and $\mathbf{x}'$, your task is to recover the affine camera parameters.

- What is the minimum number of corresponding points needed to recover the camera parameters?
- Are there any conditions on the positions of these corresponding points for successful recovery of M ? List two degenerate configurations. Can you think of a necessary condition?
- Running `house_model.m` will display an image of an affine transformed house. A set of 8 corresponding points are contained in the matlab variables `Xcorr` (original points) and `Xcorr_` (affine transformed points). Write a matlab function that takes as input `Xcorr` and `Xcorr_` and returns the affine transformation matrix, M . What is the recovered affine transformation matrix?
- Repeat part (c) using only correspondence points 1,3,4, and 6. What happens when you try to estimate M ? Why?
- Write a second program that prompts the user to manually click the correspondence points from the

plot of the house model shown after running `house model.m`¹. Using the manually selected correspondences, compute the affine transformation matrix M . Do you get the same result as in part (c)? How sensitive is the estimation to errors in the correspondences (experiment by purposely mis-clicking some of the points)?

6. Programming Assignment: Image Warping and Image Merging [10pts]

This is a programming assignment, which should be done in Matlab as many of the necessary numerical routines are readily available (e.g., `eig` or `svd` for the computation of the eigenvalues and eigenvectors of a matrix). All data necessary for this assignment is available on the course web page.

6.1. Introduction

In this assignment, we consider a vision application in which components of the scene are replaced by components from another image scene.

Consider for a moment that you are watching a sporting event on television whose audience, which has a broad, possibly multinational audience. During these events, it is advantageous and profitable to use different advertisements in different markets to target more directly viewers in those regions. In this assignment you will implement a simple version of this algorithm using multiple advertisements on a single scene.

Given two scenes, the natural thing to do would be to compute how your object of interest is observed in the current scene and warp it to match the destination scene. This would allow you to paste together the two images where your new advertisement overlaps the particular billboard in the new scene even though these two images were obtained in different locations.

This digital replacement is accomplished by a set of points for each advertisement in both the target (scene) and advertisement images. The task then consists of mapping the points from the advertisement to their respective points in the target image. In the most general case, there would be no constraints on the scene geometry, making the problem quite hard to solve. If, however, the scene can be approximated by a plane in 3D, a solution can be formulated much more easily even without the knowledge of camera calibration parameters.

To solve this section of the homework, you will first derive the transformation that maps one image onto another in the planar scene case. Then you will write a program to find this warping and apply it to multiple advertisement images, and a single target image.

6.2. Projective Transformations of the Plane

To begin, we consider the projection of planes in images in the general case. Imagine two cameras C_1 and C_2 looking at a plane π in the world. Consider a point P on the plane π and its projections

$p = (u1, v1, 1)^T$ in one image and $q = (u2, v2, 1)^T$ in image in another image.

Fact 1 There exists a unique (up to scale) 3×3 matrix H such that, for any point P :

$$q \equiv Hp$$

(Here $\equiv$ denotes equality in homogeneous coordinates, meaning that the left and right hand side are proportional.) Note that H only depends on the plane and the projection matrices of the two cameras.

The interesting thing about this result is that by using H we can compute the image of P that would be seen in camera C_2 from the image of the point in camera C_1 without knowing its three-dimensional

¹ The matlab command “`ginput`” will be useful here.

location. Such an H is a *projective transformation* of the plane, also referred to as a *homography*.

Perhaps even more interesting, a homography can also be used to represent the transformation between images of a genuine 3-D scene for multiple images taken by camera that rotates in 3-D about the optical center.

6.3. Estimating Transformations From the Image Points

Given a set of points $\{(u_1^i, v_1^i)\}, i = 1 \dots N$ in image 1 and the corresponding set of points $\{(u_2^i, v_2^i)\}, i = 1 \dots N$ in image 2, show that H can be recovered from two sets of image points using *homogeneous* linear least squares (H is only defined up to scale). In this homework $N = 4$, and should be the 4 corners of the advertisements and the billboards. You should also note that four points should exactly solve for the homography which maps a square to an arbitrary quadrilateral and the solution for H should be unique.

Write a program which computes the matrix H using the method you just derived. Input arrays should be 2×4 matrices, listing a single image point's coordinates per column. Verify your results by manually selecting two sets of corresponding image points on two of the provided images, and applying the transformation H to them. Note that, throughout this estimation procedure, camera projection matrices did not come into play at all! Hence, no calibration was necessary.

Important hints: Recall that H is a projective transformation matrix and hence, defined only up to a scale. A good way to enforce this is by constraining the squared Frobenius norm (sum of the squared entries) of the matrix H to be equal to 1. Also remember that q and Hp are only *proportional* to each other, or equivalently we have $q \times Hp = 0$.

Here are a few implementation tips:

- Some of these matlab functions may interest you: *cpselect*, *ginput*, *svd*, *eig*, and *null*
- Image coordinates of the points are the corresponding row and column indices of the image array.
- Beware of numerical ill-conditions: Your estimation procedure may perform better if image coordinates range from -1 to 1 as opposed to from 1 to 800. Consider scaling your measurements to avoid numerical issues².

6.4. Image Warping and Image Merging

Write a program which takes as input images I_{ad} , I_{scene} and a 3×3 homography H then returns an image I_{merge} which has the input image I_{ad} warped according to H to be in the frame of the scene image I_{scene} and placed at the proper coordinates. In order to avoid aliasing and sub-sampling effects, consider producing the output by solving for each pixel of the *destination* image rather than mapping each pixel in the input image to a point in the destination image. Also note that the input and output images will be of different dimensions.

Finally, **write** a program to apply these methods to each of the two advertisements and two placements in the scene. Calculate new homographies as necessary. Use this program twice to place the both advertisements in two different locations in the scene, i.e. create two unique scenes (you may reuse coordinates in the scene image).

Here are a few implementation tips:

² Hartley normalization is a well known procedure for doing this. Given a set of homogeneous 2D points $\{x_1, \dots, x_n\}$, the Hartley normalized point set is given by $\hat{x} = Ax_i$ where A is a 3×3 linear transformation such that $\{\hat{x}_1, \dots, \hat{x}_n\}$ has zero mean and unit variance. In the context of homography estimation, we can write $q = Hp \Rightarrow A_q q = A_q H A_p^{-1} A_p p \Rightarrow \hat{q} = \hat{H} \hat{p}$. Thus, we can solve for $\hat{H}$ in term of $\hat{p}$ and $\hat{q}$ and then solve for H as $H = A_q^{-1} \hat{H} A_p$

- Some of these matlab functions may interest you: *interp2*, *meshgrid*

What to turn in:

Submit a hardcopy at the due date with the following

- a. Derivation of the algorithm for estimating H .
- b. Your code for: estimating H , image warping, image merging and any other code you wrote to complete this assignment.
- c. An image showing the set of corresponding points you selected plotted on the original images.
- d. An image showing the result of your first placement of both advertisements
- e. An image showing the result of your second placement of both advertisements

In addition, please email to jmerkow@ucsd.edu a copy of your code. If it is a single file, just send the *.m* file. If it is contained in multiple files, send a *zip* or *tar* file. In the subject line, please put the string: CSE252HW1.