CS 231A Computer Vision
Problem Set 2
Due Date: Friday, Oct 26, 2012

1 Some Projective Geometry Problems

Suppose there are two parallel lines that extend to infinity in our world coordinates. The images of these two lines intersect at a point, v, on the image plane known as the vanishing point. Every line has a parallel line that goes through the origin. You can also think of a vanishing point as a point on the line that goes through the origin, and remember that the distance between a vanishing point in world coordinates and the origin is infinite. In this problem we choose to represent the lines goes through the origin as (using non-homogeneous coordinates)

$$l = \left\{ r \begin{bmatrix} \cos\alpha \\ \cos\beta \\ \cos\gamma \end{bmatrix} \;\middle|\; r \in \mathbb{R} \right\}$$

where α, β and γ are the angles from the x, y and z world axes respectively.

(a) Using homogenous coordinates show that the image coordinates of the vanishing point v can be written as $v = KRd$ where

$$d = \begin{bmatrix} \cos\alpha \\ \cos\beta \\ \cos\gamma \end{bmatrix}$$

K is the camera calibration matrix and R is the rotation matrix between the camera and the world coordinates.

(b) Express the line direction d in terms of K,R and v.

Remark : Any matrix inverse which does not hold in general, must be shown to hold in this case.

(c) Let d_1, d_2, and d_3 represent the directional vector for three lines, and let v_1, v_2, and v_3 represent the vanishing points for these lines respectively. Now consider the situation where these three lines intersect such that each line is orthogonal to the other two. That is, each line's directional vector d satisfies the following:

$$d_1^T d_2 = 0 \quad d_1^T d_3 = 0 \quad d_2^T d_3 = 0$$

Using this property, show that

$$(K^{-1}v_i)^T(K^{-1}v_j) = 0$$

for $i \neq j, 1 \leq i \leq 3$, and $1 \leq j \leq 3$

2 Fundamental Matrix (15 points)

In this question, you will explore some properties of fundamental matrix and derive a minimal parameterization for it.

(a) Show that two 3×4 camera matrices M and M' can always be reduced to the following canonical forms by an appropriate projective transformation in a 3D space, which is represented by a 4×4 matrix H. Here, we assume $e_3^T(-A'A^{-1}b + b') \neq 0$, where $e_3 = (0, 0, 1)^T$, $M = [A, b]$ and $M' = [A', b']$.
Note: You don't have to show the explicit form of H for the proof.
(Hint: The camera matrix has rank 3.)

$$\hat{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \text{ and } \hat{M}' = \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(b) Given a 4×4 matrix H representing a projective transformation in a 3D space, prove that the fundamental matrices corresponding to the two pairs of camera matrices (M, M') and $(MH, M'H)$ are the same.

(c) Using the conclusions from (a) and (b), derive the fundamental matrix F of the camera pair (M, M') using a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}, b_1, b_2. Then use the fact that F is only defined up to a scale factor to construct a seven-parameter expression for F. *(Hint: The fundamental matrix corresponding to a pair of camera matrices $M = [I \mid 0]$ and $M' = [A \mid b]$ is equal to $[b]_\times A$.)*

3 Efficient Solution for Stereo Correspondence

In this exercise you are given two or more images of the same 3D scene, taken from different points of view. You will be asked to solve a correspondence problem to find a set of points in one image which can be identified as the same points in another image. Specifically, the way you formulate the correspondence problem in this exercise will lend itself to be computationally efficient.

(a) Suppose you are given a point (p_1, p_2) in the first image and the corresponding point (p_1', p_2') in the second image, both of the same 3D scene. Write down the homogenous coordinates x and x'. In addition, write down the necessary and sufficient conditions for the points x and x' to meet in the 3D world, using what we know about fundamental matrix F.

(b) Since our points are measurements and susceptible to noise, x and x' may not satisfy the conditions in (a). Writing this statistically, our measurements are given by

$$x = \bar{x} + \Delta x \qquad x' = \bar{x}' + \Delta x'$$

where Δx and $\Delta x'$ are noise terms, and $\bar{x}$ and $\bar{x}'$ are the true correspondences we are trying to determine. Now we require that only $\bar{x}$ and $\bar{x}'$ satisfy the condition in (a). To find our best estimate of $\bar{x}$ and $\bar{x}'$, we will minimize

$$E = \|\Delta x\|^2 + \|\Delta x'\|^2$$

subject to the constraint on $\bar{x}$ and $\bar{x}'$ from (a). In addition to the constraint from (a), we also need to constrain the noise terms so that x and x' remain on the image plane, i.e., a constraint of Δx so that the last coordinate of x and $\bar{x}$ in homogeneous coordinate is identical. Similar for x' and $\bar{x}'$. For this part, write down the optimization problem (i.e. what you are minimizing and the constraints). Disregard the higher order terms of Δx and $\Delta x'$ in the constraint from (a).

Your answer should be in the following form:

minimize _____
subject to _____
_____
_____

Hint: To constrain the noise terms Δx and $\Delta x'$ *to lie on the image plane, the unit vector* $e_3 = [\ 0 \quad 0 \quad 1\ ]^T$ *will be useful. Also,* $\bar{x}$ *and* $\bar{x}'$ *should not appear in the optimization problem.*

(c) Once we drop these higher order terms, we can use Lagrange multipliers to solve the optimization problem in (b). Show that optimal Δx and $\Delta x'$ are given by

$$\Delta x = \frac{x^T F x' P_e F x'}{x'^T F^T P_e F x' + x^T F P_e F^T x} \qquad \Delta x' = \frac{x^T F x' P_e F^T x}{x'^T F^T P_e F x' + x^T F P_e F^T x}$$

Hint: The projection matrix $P_e = \mathrm{diag}(1, 1, 0)$ *will be useful to eliminate unwanted terms after you have taken the derivative and set it to zero.*

Remark: Once we have determined the optimal values for Δx *and* $\Delta x'$ *we can use these optimal values in conjunction with our measurements* x *and* x' *to determine our estimate of the true correspondences* $\bar{x}$ *and* $\bar{x}'$*.*

4 Affine Camera Calibration

In this question, we will perform affine camera calibration using two different images of a calibration grid. First, you will find correspondances between the corners of the calibration grids and the 3D scene coordinates. Next, you will solve for the camera parameters.

It was shown in class that a perspective camera can be modelled using a 3×4 matrix:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1}$$

which means that the image at point (X, Y, Z) in the scene has pixel coordinates $(x/w, y/w)$. The 3×4 matrix can be factorized into intrinsic and extrinsic parameters.

An *affine* camera is a special case of this model in which rays joining a point in the scene to its projection on the image plane are parallel. Examples of affine cameras include orthographic projection and weakly perspective projection. An affine camera can be modeled as:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2}$$

which gives the relation between a scene point (X, Y, Z) and its image (x, y). The difference is that the bottom row of the matrix is [0 0 0 1], so there are fewer parameters we need to calibrate. More importantly, there is no division required (the homogeneous coordinate is 1) which means this is a *linear model*. This makes the affine model much simpler to work with mathematically - at the cost of losing some accuracy. The affine model is used as an approximation of the perspective model when the loss of accuracy can be tolerated, or to reduce the number of parameters being modelled.

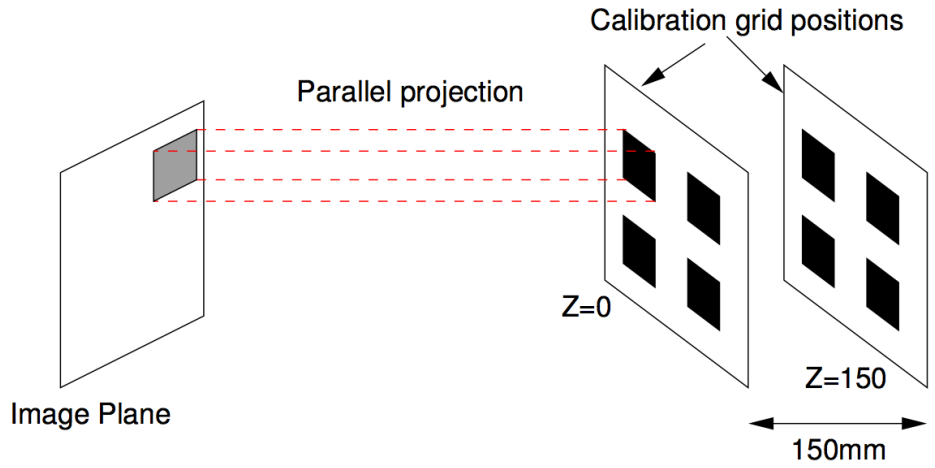
Calibration of an affine camera involves estimating the 8 unknown entries of the matrix in Eq. 2. (This matrix can also be factorized into intrinsics and extrinsics, but that is outside the scope of this homework). Factorization is accomplished by having the camera observe a calibration pattern with easy-to-detect corners.

Scene Coordinate System

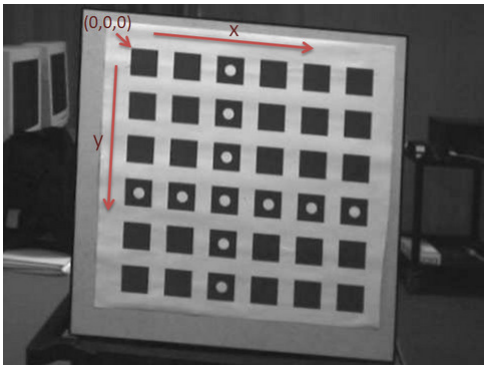
The calibration pattern used is shown in Fig. 1, which has a 6×6 grid of squares. Each square is $50mm \times 50mm$. The separation between adjacent squares is $30mm$, so the entire grid is $450mm \times 450mm$. For calibration, images of the pattern at two different positions were captured. These images are shown in Fig. 1 and can be downloaded from the course website. For the second image, the calibration pattern has been moved back (along its normal) from the rest position by $150mm$.

We will choose the origin of our 3D coordinate system to be the top left corner of the calibration pattern in the rest position. The X-axis runs left to right parallel to the rows of squares. The Y-axis runs top to bottom parallel to the columns of squares. We will work in units of millimeters. All the square corners from the first position corresponds to $Z = 0$. The second position of the calibration grid corresponds to $Z = 150$. The top left corner in the first image has 3D scene coordinates $(0, 0, 0)$ and the bottom right corner in the second image has 3D scene coordinates $(450, 450, 150)$. This scene coordinate system is labeled in Fig. 1.

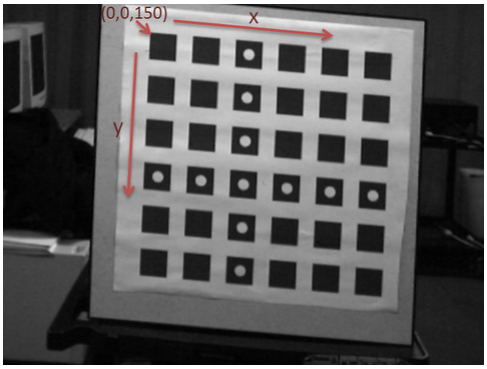
(a) Download the calibration images from the class website (`http://vision.stanford.edu/teaching/cs223b/hw/PS2_data.zip`). The images will be in the `affineCamera` direc-



(a) Image formation in an affine camera. Points are projected via parallel rays onto the image plane



(b) Image of calibration grid at Z=0



(c) Image of calibration grid at Z=150

Figure 1: Affine camera: image formation and calibration images.

tory. Find the feature correspondences manually for calibration. In other words, first calculate the scene coordinates of the square corners of the calibration grid. Next, find the corresponding pixel coordinates in the images manually.

If you wish to measure the pixel coordinates interactively in MATLAB, you may find the function `ginput` useful. It is strongly recommended that you save this matrix of measurements in a file for later use. You do not need to find *all* the corners, but the more corners you use the more accurate the calibration is likely to be. Be sure to include corners from *both* images. We recommend using at least 12 corners from each image. Report your feature correspondence pairs. You can find some helpful code for selecting points interactively in `ginputSample.m` file in the `affineCamera` directory.

(b) Having measured enough features, solve for the camera parameters using Eq. 2. Note that each measurement $(x_i, y_i) \leftrightarrow (X_i, Y_i, Z_i)$ yields two linear equations for the 8 unknown camera parameters. Given N corner measurements, we have $2N$ equations and 8 unknowns. Using your measured feature correspondences as inputs, write a script which constructs the linear system of equations and solves for the camera parameters which

minimizes the least-squares error. The inputs should be your feature correspondences. The output should be your calibrated camera matrix, and the RMS error between your N image corner coordinates and N corresponding calculated corner locations (calculated using the affine camera matrix and scene coordinates for corners). Please hand in your MATLAB code. Also, please report your 3×4 calibrated camera matrix as well as your RMS error.

$$RMS_{total} = \sqrt{\sum((x - x')^2 + (y - y')^2)/N}$$

(c) Could you calibrate the matrix with only one checkerboard image? Explain in words.

5 Image Stitching

In this problem, you will implement your own image stitching application. This technique is used to combine many photographs to produce a larger image than could be possible with one exposure alone. Refer to Fig. **??** for an example output from Adobe Photoshop's photomerge capability which produces highly refined panoramas using advanced techniques such as feathering, pyramid blending. Ignoring the "bells and whistles," the image-stitching portion of this algorithm reduces to finding the parameters for the affine transformation and homography matrix H to map one image onto the plane of the other.

(a) Let p correspond to point on one image and let p' correspond to the same point in the scene, but projected onto another image. Write a general equation for how a homography matrix H maps points from one image to another. How would H be restricted if it must describe an affine transformation?

(b) Our first implementation of image-stitching will use an affine approximation. This matrix transforms any point p_i in one view to its corresponding homogeneous coordinates in the second view, p_i', in accordance with your equation above. Note that $p_i, p_i' \in \mathbb{R}^3$.

Your task is to write a function that takes in $n \geq 4$ pairs of corresponding points from the two views, where each point is specified with its 2-D image coordinates, and returns the affine transformation matrix. Note: We have provided skeleton code. Please download the `stitching.m` script from the course webpage (`http://vision.stanford.edu/teaching/cs223b/hw/PS2_data.zip`). The `stitching.m` skeleton code will be in the `imageStitching` directory. You can confirm your result with MATLAB's `cp2tform` command which is given in `stitching.m`. Please hand in the resulting, stitched image, and your MATLAB code.

Big Hint: We can set up a solution using a system of linear equations $Ax = b$, where the 8 unknowns of H are put into a vector $x \in \mathbb{R}^8$, $b \in \mathbb{R}^{2n}$ contains image points from one view and the $2n \times 8$ matrix A is filled appropriately so that the full system gives us $\lambda p_i = Hp_i'$. Be sure to apply what you know about H. Solve for the unknown homography matrix parameters. In Problem 5, we assume that p=Hp', where p is the base point. And in your Matlab code, you also need to transpose H as the return value.

(c) Write a function that takes a set of corresponding image points ($n \geq 4$ pairs) and computes the associated 3×3 general homography matrix H. Use this function instead of your AffineTransformation function from part (b) to stitch the images together. As in (b),

Figure 2: Example panorama

please hand in the resulting, stitched image,and the MATLAB code. Which method (affine transformation, or homography) produces better results?

(d) Extra Credit (1%): Stitch a panorama using your own photos (number of images $n \geq 4$). You can choose either affine transformation or general homography transformation.