CS 231A Computer Vision, Winter 2015
Problem Set 1

Due Date: Jan 28$^{\text{th}}$ 2015 11:59 pm

1 Some Projective Geometry Problems [15 points]

Lines that are parallel in world space do not remain parallel when projected on to the image plane via a perspective projection. The point at which the projected lines meet, v, is known as a *vanishing point.* Now, every line has a parallel line that goes through the origin of the world coordinate system. Therefore, the vanishing point can be considered to lie on a line through the origin, at an infinite distance from the origin. In this problem we choose to represent lines that go through the origin as (using inhomogeneous 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.

1.a **[5 points]** 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.

1.b **[5 points]** 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.

1.c **[5 points]** 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:

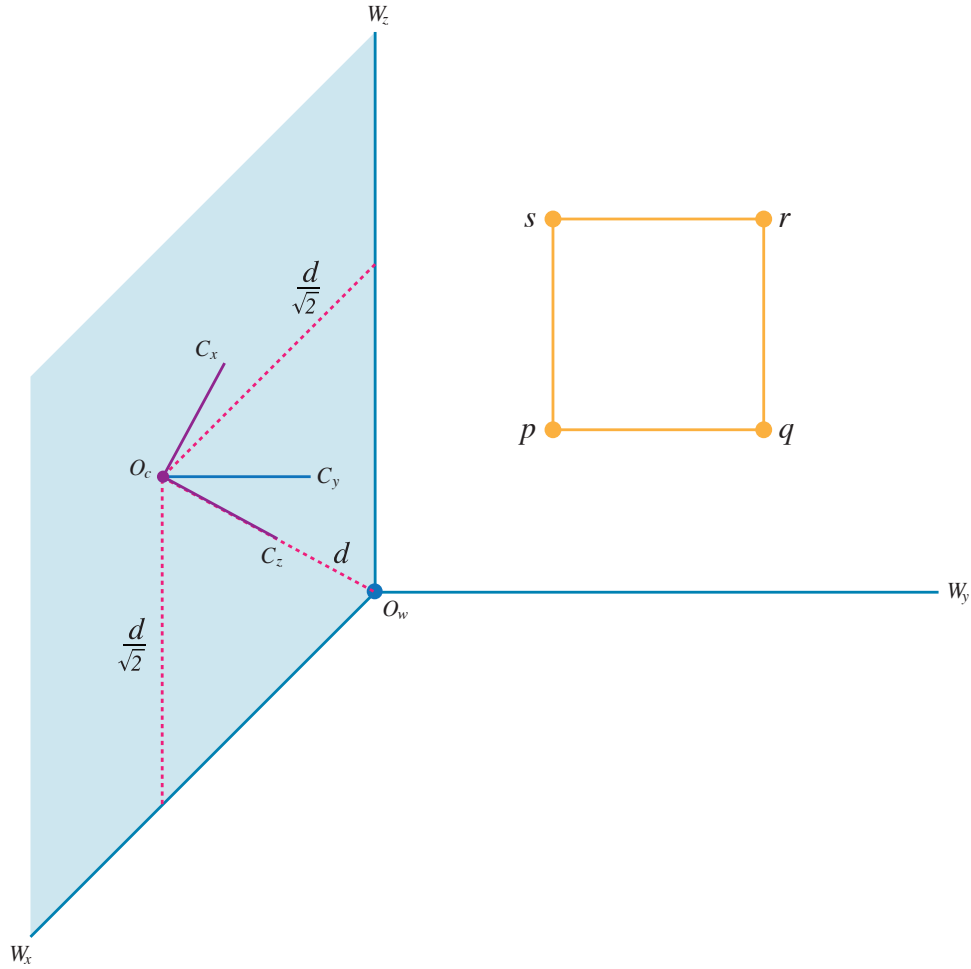$$d_1^T d_2 = 0 \quad d_1^T d_3 = 0 \quad d_2^T d_3 = 0$$

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 More Projective Geometry Problems [10 points]

In this question, we will examine more properties of projective transformations as we translate a simple shape from world coordinates to camera coordinates.

2.a **[2 points]** Derive the combined rotation and translation needed to transform world coordinate W into camera coordinate C as illustrated in the figure. Notice that C_z and C_x belong to the plane defined by W_z and W_x, while C_y is parallel to W_y.

2.b **[3 points]** Consider a square in the world coordinate system defined by the points p, q, r, s. Assume such a square has unit area. Show that the same square in the camera reference system still has unit area.

2.c [3 points] Are parallel lines in the world reference system still parallel in the camera reference system? Justify answer.

2.d [2 points] Does the vector defined from p to q have the same orientation in both reference systems? Justify answer.

3 Affine Camera Calibration [20 points]

In this question, we will perform affine camera calibration using two different images of a calibration grid. First, you will find correspondences 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 modeled 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:

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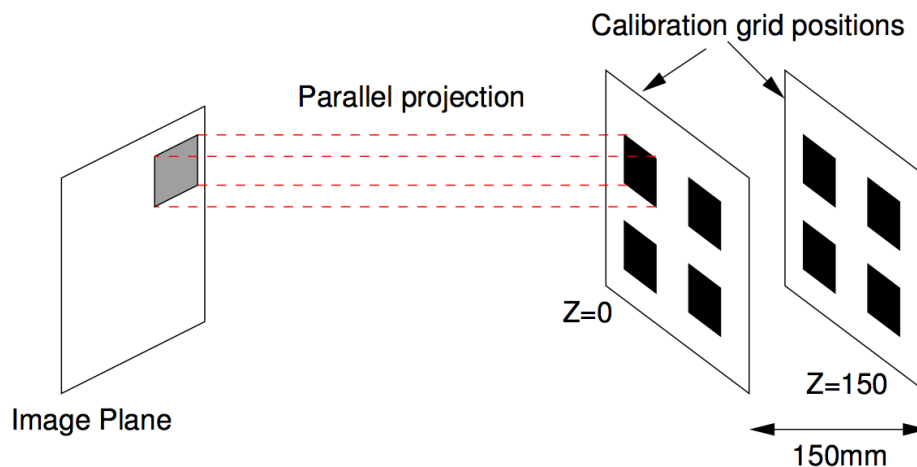\tag{2}
$$

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.
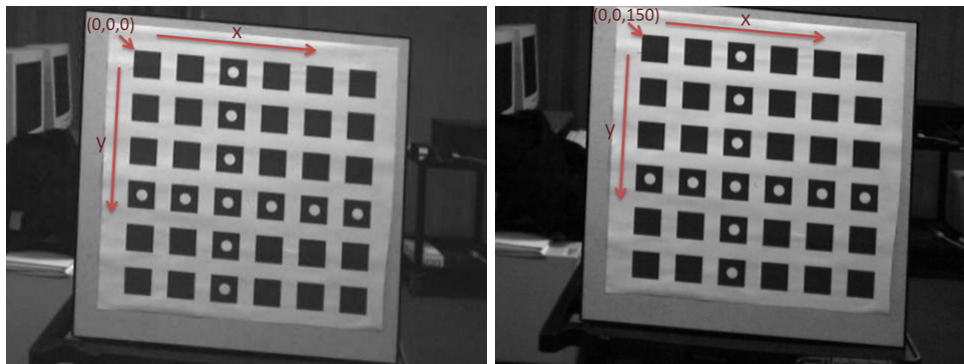
Scene Coordinate System

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) 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.

3.a **[5 points]** Download the calibration images from the class website (http://cvgl. stanford.edu/teaching/cs231a_winter1415/HW/PS1_data.zip). The images will be in the `affineCamera` directory. 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.

3.b **[10 points]** 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.

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

3.c **[5 points]** Could you calibrate the matrix with only one checkerboard image? Explain in words.

4 Single View Geometry [30 points]

(a) Image 1 (1.jpg)   (b) Image 2 (2.jpg)

In this question, we will estimate camera parameters from a single view and leverage the projective nature of cameras to find both the camera center and focal length from vanishing points present in the scene above.

4.a **[15 points]** Download the calibration images from the class website (http://cvgl.stanford.edu/teaching/cs231a_winter1415/HW/PS1_data.zip). The images will be in the `singleView` directory. Identify a set of N vanishing points from the Image 1. Assume the camera has zero skew and square pixels, with no distortion.

4.a.i Compute the focal length and the camera center using these N vanishing points. Show derivation and report numerical results.
Hint: Use the function getVanishingPoint.m to compute the Vanishing Point. Read the ReadMe.txt file for usage information.

4.a.ii Is it possible to compute the focal length for any choice of vanishing points?

4.a.iii Whats the minimum N for solving this problem? Justify answer.

4.a.iv The method used to obtain vanishing points (as explained in the ReadMe.txt) is approximate and prone to noise. Discuss approaches to refine this process.

This process gives the camera internal matrix under the specified constraints. For the remainder of the computations, use the following internal camera matrix:

$$K = \begin{bmatrix} 2448 & 0 & 1253 \\ 0 & 2438 & 986 \\ 0 & 0 & 1 \end{bmatrix}$$

4.b **[7 points]** Identify a sufficient set of vanishing lines on the ground plane and the plane on which the letter A exists, written on the side of the cardboard box, (plane-A). Use these vanishing lines to verify numerically that the ground plane is orthogonal to the plane-A.

4.c **[8 points]** Assume the camera rotates but no translation takes place. Assume the internal camera parameters remain unchanged. An Image 2 of the same scene is taken. Use vanishing points to compute camera rotation matrix. Report derivation and numerical results.

5 Efficient Solution for Stereo Correspondence [25 points]

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.

5.a **[3 points]** 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 condition for the points x and x' to meet in the 3D world, using what we know about fundamental matrix F.
Note: When writing down the necessary condition, make sure you express it in terms of the epipolar line corresponding to the point x'. This will be important for later parts of this question.

5.b **[15 points]** 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:

$$\begin{array}{ll} \text{minimize} & \text{_____} \\ \text{subject to} & \text{_____} \\ & \text{_____} \\ & \text{_____} \end{array}$$

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

5.c [7 points] 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 $P_e = \text{diag}(1, 1, 0)$ 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 $\bar{x}'$.