CS 4495/7495 Fall 06 Computer Vision
Assignment 1 Projective geometry

29th August 2006

Introduction

In this assignment, you are going to play with simple 2D geometry and 3D geometry primitives (points, lines, and planes) and get used to their representation in homogeneous coordinates. There are four parts. Part 1 and Part 2 are a warm-up for 2D and 3D geometry. In Part 3, you will need to find the affine camera matrix associated with an image we provided, and calculate the projection of a few points. Finally, in Part 4, you will implement a simple matlab program to insert an synthetic object in an image of your choice.

Please partner up into **teams of 2 students**. Hand in Part 1 and 2 on Sept. 5, and Part 3 and 4 on Sept. 12. The final synthetic images you obtained in Part 4 should be uploaded to swiki (`http://swiki.cc.gatech.edu:8080/cs4495-fl04`). For the other parts, please provide a nicely formatted write-up that briefly describes what you did and what the results were.

1 Warm-up: 2D projective Geometry

In 2D projective geometry a 2D point p is represented using three numbers (x, y, w), and a line l is represented using three numbers (a, b, c) defining a set of points for which

$$ax + by + cw = 0$$

1. In Figure 1, sketch the line represented by its homogeneous coordinates $l_1 = (0, 1, 1)^T$. Please label it with l_1.

2. Sketch the line $l_2 = (4, 0, 1)^T$ and label it l_2.

3. **Manually** compute the intersection p_1 of the two lines using the cross-product, and label it p_1. Remember, points and lines are dual and they share the same homogeneous representation. The join of two lines is a point, and the join of the two points is a line. For example, for points $p = (x_1, y_1, w_1)^T$ and $q = (x_1, y_2, w_2)^T$ we have

$$p \times q = \begin{vmatrix} \bar{i} & \bar{j} & \bar{k} \\ x_1 & y_1 & w_1 \\ x_2 & y_2 & w_2 \end{vmatrix} = \begin{vmatrix} y_1 & w_1 \\ y_2 & w_2 \end{vmatrix} \bar{i} - \begin{vmatrix} x_1 & w_1 \\ x_2 & w_2 \end{vmatrix} \bar{j} + \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} \bar{k} \quad (1)$$

where for example the *minor* $\begin{vmatrix} y_1 & w_1 \\ y_2 & w_2 \end{vmatrix} = y_1 w_2 - y_2 w_1$. See also `http://mathworld.wolfram.com/CrossProduct.html`

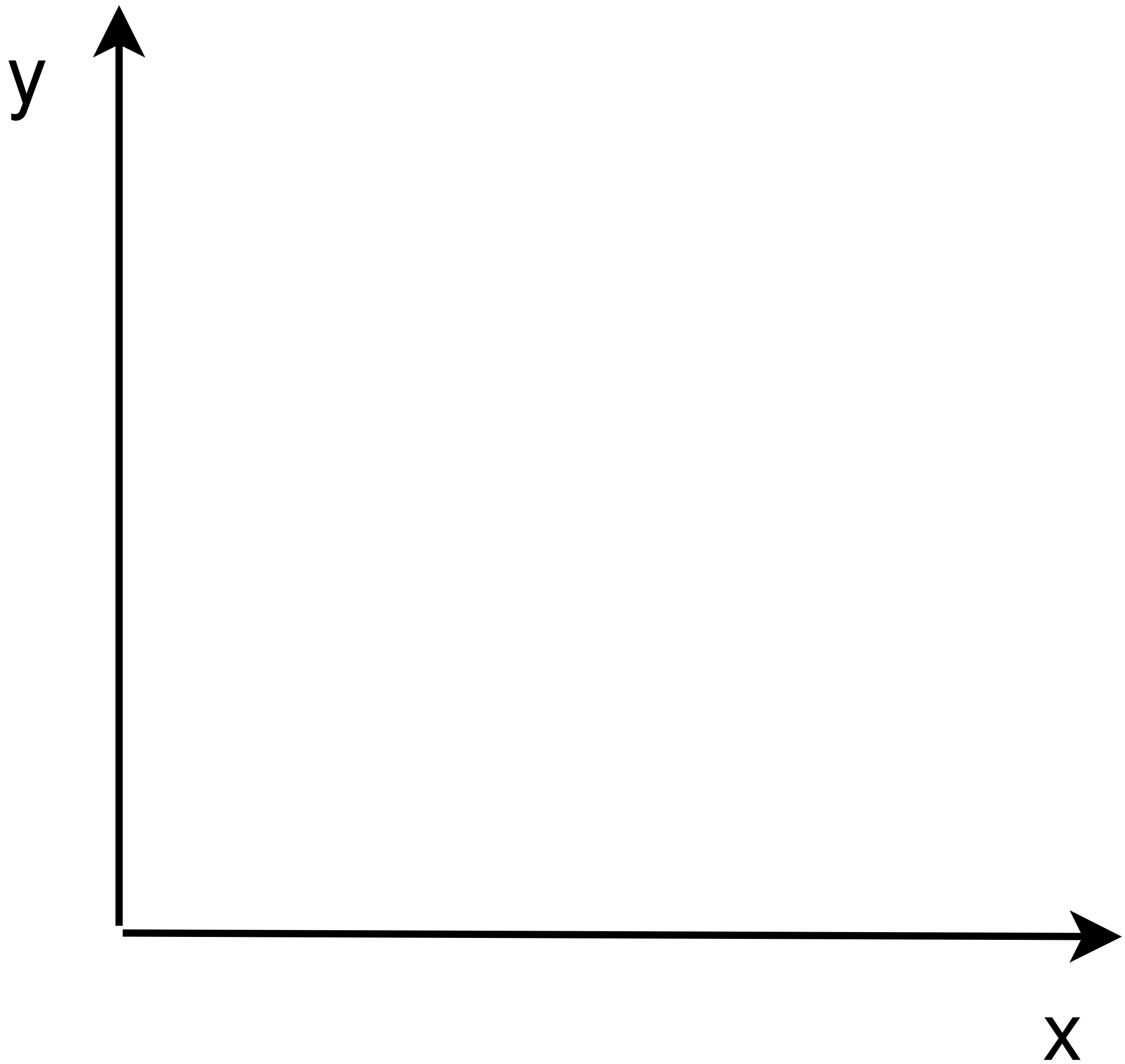

Figure 1: 2D Sketch

4. Now, sketch a third line $l_3 = (2, 0, 1)^T$ and label it l_3.
5. Compute the intersection p_2 of l_2 (from Q2) and l_3.
6. Where is that point ? Indicate where it is on your sketch.
7. You can also find the line between points in the same way by doing cross product between points: find the join l_4 of point $p_3 = (0, 0, 1)^T$ with the point p_1 from Q3. Sketch this line as well, and label it l_4.

2 3D Projective Geometry

In 3D projective geometry, we represent 3D points and **planes** with 4 numbers. In other words, in 3D planes and points are dual, not lines and points. In fact, lines are not so easy to describe in 3D.

1. Compute the join of three points $(1, 0, 0, 1)^T$, $(0, 1, 0, 1)^T$, and $(0, 0, 1, 1)^T$ and sketch it on Figure 2. The join of three points define a plane. Now, there does not exist a 3D cross-product, but we can generalize equation (1) by employing 3×3 determinant minors instead of 2×2 minors. In particular, for points $p = (x_1, y_1, z_1, w_1)^T$, $q = (x_2, y_2, z_2, w_2)^T$,$r = (x_3, y_3, z_3, w_3)^T$, their join can be computed by insisting that

$$\begin{vmatrix} x & y & z & w \\ x_1 & y_1 & z_1 & w_1 \\ x_2 & y_2 & z_2 & w_2 \\ x_3 & y_3 & z_3 & w_3 \end{vmatrix} = 0$$

i.e., the point (x, y, z, w) is linearly dependent on the other rows. Hence, the plane equation becomes

$$\begin{vmatrix} y_1 & z_1 & w_1 \\ y_2 & z_2 & w_2 \\ y_3 & z_3 & w_3 \end{vmatrix} x - \begin{vmatrix} x_1 & z_1 & w_1 \\ x_2 & z_2 & w_2 \\ x_3 & z_3 & w_3 \end{vmatrix} y + \begin{vmatrix} x_1 & y_1 & w_1 \\ x_2 & y_2 & w_2 \\ x_3 & y_3 & w_3 \end{vmatrix} z - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} w = 0$$

while the homogeneous representation of the line is

$$\left(\begin{vmatrix} y_1 & z_1 & w_1 \\ y_2 & z_2 & w_2 \\ y_3 & z_3 & w_3 \end{vmatrix}, - \begin{vmatrix} x_1 & z_1 & w_1 \\ x_2 & z_2 & w_2 \\ x_3 & z_3 & w_3 \end{vmatrix}, \begin{vmatrix} x_1 & y_1 & w_1 \\ x_2 & y_2 & w_2 \\ x_3 & y_3 & w_3 \end{vmatrix}, - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} \right)^T$$

2. Now given three planes $(a_1, b_1, c_1, d_1)^T$, $(a_2, b_2, c_2, d_2)^T$, and $(a_3, b_3, c_3, d_3)^T$, and suppose they intersect in a point, what are the coordinates of that point?

3 Projection from 3D to 2D

1. Please download the image `http://www-static.cc.gatech.edu/classes/AY2007/cs4495_fall/html/tokyo.jpg`. In this image, where objects are far from camera, we could use affine camera model which ignores depth variation. Notice that there is a color cube in the middle of the image. Suppose the origin of the world coordinate lies in the bottom corner of the cube and the x, y, and z axes coincide with the R, G, and B lines indicated in the image. An affine camera model projects a 3D world point to an image point by:

$$p = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix} = MP$$

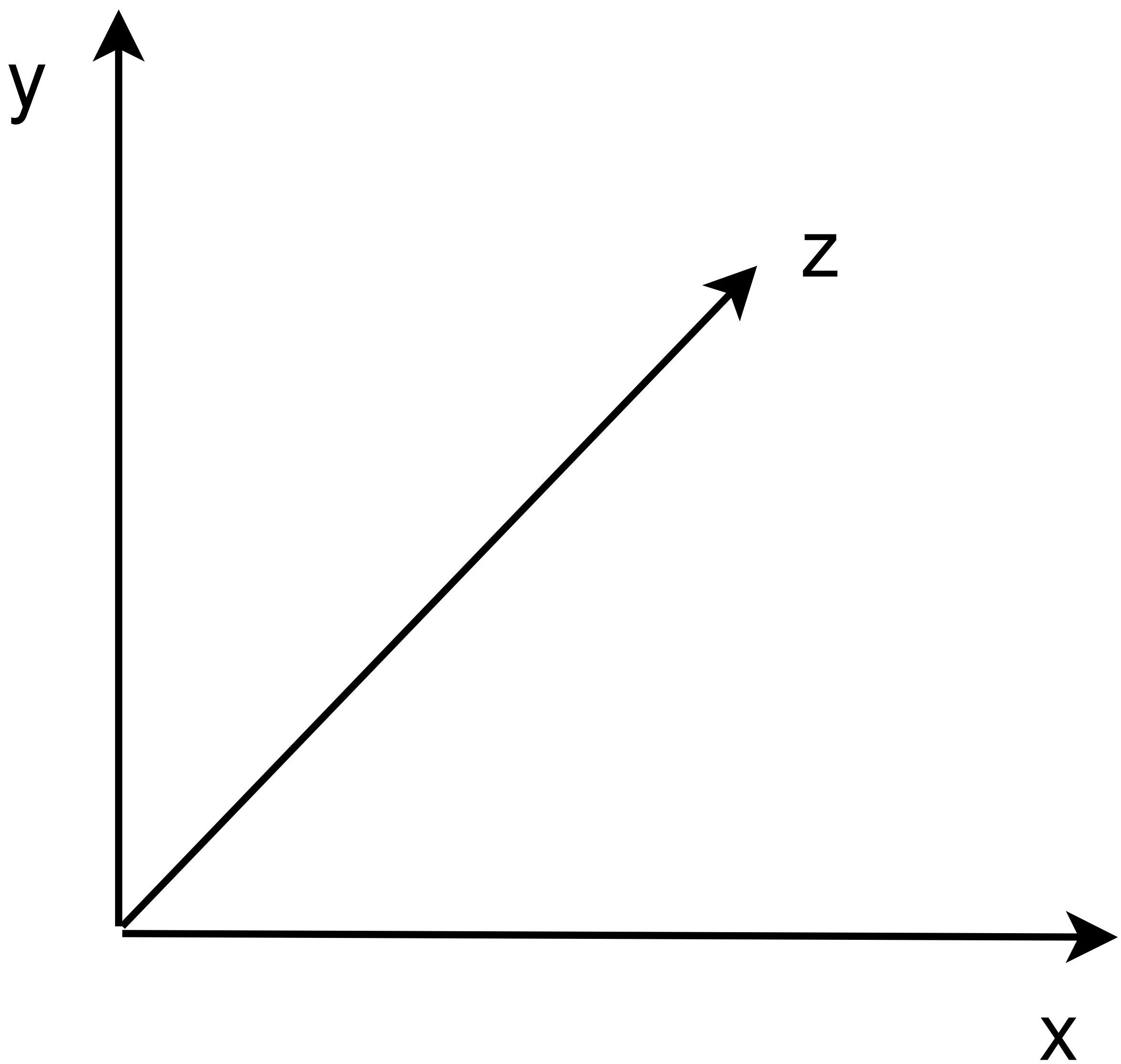

Figure 2: 3D Sketch

where $\begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix}$ is the 3D world homogeneous coordinate, $\begin{bmatrix} u \\ v \\ w \end{bmatrix}$ is the image coordinate, and $M = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ 0 & 0 & 0 & 1 \end{bmatrix}$ is the affine camera matrix, parameterized by 8 numbers $a...h$. If we normalize the homogeneous coordinate ($w = 1, t = 1$), given four corner points of the cube, that is $(0,0,0,1)^T$,$(1,0,0,1)^T$, $(0,1,0,1)^T$, and $(0,0,1,1)^T$ and their corresponding image pixel coordinates (u, v), we could get the affine camera matrix by solving 8 linear equations. Please obtain the image pixel coordinates first, list your equations and show the camera matrix that you solved for.

2. Project 3D points $p_1(1,1,1,1)^T$, $p_2(1,0,1,1)^T$, and $p_3(0,1,1,1)^T$ manually to find their image coordinates. You need to multiply the point's coordinate by the matrix you just computed. Are they in the right place? Show them on the image by indicating the position in a sketch on your paper.

4 Inserting Synthetic Objects

You already obtained the Affine camera matrix in Part 3, now you could project any 3D world objects onto the image. We encourage you to use Matlab programming in this part.

1. we define a pyramid with four 3D points on the base $(0,0,1,1)^T$, $(1,0,1,1)^T$, $(1,1,1,1)^T$,$(0,1,1,1)^T$ and one 3D point $(0.5, 0.5, 2, 1)^T$ as the apex. You should insert it onto the image, which is used in Part3. After getting the image pixel coordinates of the five vertices, you could use Matlab to draw line between the vertices to sketch the pyramid. Upload your synthetic image to Swiki.

2. Go crazy to define your own recognizable object (a Bunny) and insert it into an image which you choose yourself. Notice, you'd better choose an image suitable for an affine camera model, which means the depth variation can be ignored. Otherwise a more complicated camera calibration is needed. And also, it is easier to get the affine camera matrix if there's a cube in the image. Upload to Swiki the image you choose and the one after synthesizing.