Assignment 1: CS 763, Computer Vision

Sasank Chilamkurthy
Tharun Kumar Reddy
Rajeev Puppala

February 5, 2015

1. *In class, we have seen image formation on a flat screen (i.e. image plane) with a pinhole camera. Now suppose the screen was wrapped on the surface of a sphere and hence, the 3D points were projected onto a spherical surface. Derive a relationship between the coordinates of a 3D point $P = (X, Y, Z)$ and its image on such a screen (both in camera coordinate system). If you had to calibrate this sort of a system, what are the additional intrinsic parameters of the camera as compared to the case of an image plane ?* *[4 points]*

   Ans. Let origin be the pinhole and $(0, 0, c)$ be the centre and r be the radius of the screen sphere. Intersection of line passing through the pinhole and the point $P = (X, Y, Z)$ and screen sphere is the requried projection. Since general point on the sphere can be parametrized as $(r\sin\theta\cos\phi, r\sin\theta\sin\phi, c+r\cos\theta)$,

$$\frac{r\sin\theta\cos\phi}{X} = \frac{r\sin\theta\sin\phi}{Y} = \frac{c + r\cos\theta}{Z}$$

   From first part of the equation,

$$\frac{r\sin\theta\cos\phi}{X} = \frac{r\sin\theta\sin\phi}{Y} \tag{1}$$
$$\tan\phi = Y/X \tag{2}$$
$$\phi = \tan^{-1}\frac{Y}{X} \tag{3}$$
$$\sin\phi = \frac{Y}{\sqrt{X^2+Y^2}} \tag{4}$$

   using this in the second part of the equation,

$$\frac{r\sin\theta\sin\phi}{Y} = \frac{c + r\cos\theta}{Z} \tag{5}$$
$$\frac{r\sin\theta}{\sqrt{X^2+Y^2}} = \frac{c + r\cos\theta}{Z} \tag{6}$$

   Let $K = \sqrt{X^2 + Y^2}$

$$Zr\sin\theta = K(c + r\cos\theta) \tag{7}$$
$$r(Z\sin\theta - K\cos\theta) = Kc \tag{8}$$

   Divide either side by $\sqrt{Z^2 + K^2} = \sqrt{X^2 + Y^2 + Z^2} = R$. We get $\sin(\theta - \cos^{-1}(\frac{Z}{R})) = \frac{Kc}{R}$. Hence, $\theta = \cos^{-1}(\frac{Z}{R}) + \sin^{-1}(\frac{Kc}{R})$. We've thus determined θ and ϕ in terms of the r,c and X,Y,Z.
   Therefore the image of the point (X, Y, Z) on the image sphere is

$$(r\sin\theta\cos\phi, r\sin\theta\sin\phi, c + r\cos\theta),$$

   where $\theta = \cos^{-1}(\frac{Z}{\sqrt{X^2+Y^2+Z^2}}) + \sin^{-1}(\frac{Kc}{\sqrt{X^2+Y^2+Z^2}})$ and $\phi = \tan^{-1}\frac{Y}{X}$. Here, the other intrinsic parameters we are interested in are r, c.

2. *In this exercise, we will prove the orthocenter theorem pertaining to the vanishing points Q, R, S of three mutually perpendicular directions OQ, OR, OS, where O is the pinhole (origin of camera coordinate system). Let the image plane be $Z = f$. Recall that two directions v_1 and v_2 are orthogonal if $v_1^T v_2 = 0$. One can conclude that OS is orthogonal to $OR - OQ$ (why?). Also the optical axis Oo (where o is the optical center) is orthogonal to $OR - OQ$ (why?). Hence the plane formed by triangle OSo is orthogonal to $OR - OQ$ and hence line oS is perpendicular to $OR - OQ = QR$ (why?). Likewise oR and oQ are perpendicular to QS and RS. Hence we have proved that the altitudes of the triangle QRS are concurrent at the point o. QED. Now, in this proof, I considered the three perpendicular lines to be passing through O. How will you modify the proof if the three lines did not pass through O? [4 points]*

Ans. OQ, OR, OS are mutually perpendicular lines.
Let $\bar{s}, \bar{q}, \bar{r}$ represent unit vectors along OS, OQ, OR. Therefore, $s(q-r)^T = 0 - 0 = 0$.
Hence, OS is orthogonal to $OQ - OR$. Q, R, S lie on image plane $Z = f$. As Oo is orthogonal to this plane.
Hence, Oo is orthogonal with QR ,i.e., $OQ - OR$.
Hence, $OS \perp (OQ - OR)$ and $Oo \perp (OQ - OR)$. QR is normal to OSo triangle plane.
Therefore, oS is $\perp QR$. Similarly $oQ \perp SR$ and $oR \perp QS$.
$\implies$ o is orthocentre of $\triangle QRS$

Even if the lines don't pass through O, the lines connecting vanishing points Q, R, S to O will be parallel to the respective original lines and are still mutually perpendicular. Hence the proof redirects to the above case.

3. *Prove that the vanishing points of three coplanar lines are collinear. [2 points]*

Ans. Consider lines

$$\frac{X - X_i}{l_i} = \frac{Y - Y_i}{m_i} = \frac{Z - Z_i}{n_i}, i = 1, 2, 3$$

For these lines to be coplanar, it is neccesary that direction vectors of lines $v_i = (l_i, m_i, n_i)$ are coplanar or equivalently, linearly dependent. Thus,

$$\begin{vmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{vmatrix} = 0$$

Vanishing points of these lines can be computed as

$$(x_i, y_i) = \lim_{t \to \infty} \left(\frac{X_i + l_i t}{Z_i + n_i t}, \frac{Y_i + m_i t}{Z_i + n_i t} \right) = \left(\frac{l_i}{n_i}, \frac{m_i}{n_i} \right)$$

These three points in $X - Y$ plane are collinear as

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = \begin{vmatrix} l_1/n_1 & m_1/n_1 & 1 \\ l_2/n_2 & m_2/n_2 & 1 \\ l_3/n_3 & m_3/n_3 & 1 \end{vmatrix} = \begin{vmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{vmatrix} = 0$$

thus, proving the required statement.

4. *Consider a triangle (in 3D) whose side lengths are known to you. You capture an image of this triangle and mark out the positions of its three vertices in the image (assume all vertices were visible when you took the picture). Suppose, that you knew the 3D coordinates of exactly one vertex of the triangle, in the camera coordinate system. Explain how you will determine the 3D coordinates of the other two vertices, and write down the key equations (I do not expect you to solve the equations). Assume the pixel resolution of your camera to be 1 in both directions and the optical center to be $(0, 0)$. Do not assume that you knew the focal length. [4 points]*

Ans. Let $(X_i, Y_i, Z_i), i = 1, 2, 3$ be the coordinates of 3 points in camera coordinate system and suppose that coordinates of the first point is known. Correspoding points in the image $(x_i, y_i) = \left(\frac{fX_i}{Z_i}, \frac{fY_i}{Z_i} \right), i = 1, 2, 3$ are also known. We also know the distance between points, d_{ij} in the camera cordiate system

Equation $(x_1, y_1) = \left(\frac{fX_1}{Z_1}, \frac{fY_1}{Z_1}\right)$ gives the value of f. From the other two equations, we get

$$(X_i, Y_i, Z_i) = \left(\frac{x_i Z_i}{f}, \frac{y_i Z_i}{f}, Z_i\right), i = 2, 3$$

From the three equations,

$$d_{12} = \left\| (X_1, Y_1, Z_1) - \left(\frac{x_2 Z_2}{f}, \frac{y_2 Z_2}{f}, Z_2\right) \right\| \tag{9}$$

$$d_{13} = \left\| (X_1, Y_1, Z_1) - \left(\frac{x_3 Z_3}{f}, \frac{y_3 Z_3}{f}, Z_3\right) \right\| \tag{10}$$

$$d_{23} = \left\| \left(\frac{x_2 Z_2}{f}, \frac{y_2 Z_2}{f}, Z_2\right) - \left(\frac{x_3 Z_3}{f}, \frac{y_3 Z_3}{f}, Z_3\right) \right\| \tag{11}$$

we can find unknowns Z_2, Z_3 unambiguously. This will give the points $(X_i, Y_i, Z_i), i = 2, 3$.

5. Consider two sets of corresponding points $\{\mathbf{p}_{1i} = (x_{1i}, y_{1i})\}_{i=1}^n$ and $\{\mathbf{p}_{2i} = (x_{2i}, y_{2i})\}_{i=1}^n$. Assume that each pair of corresponding points is related as follows: $\mathbf{p}_{2i} = \alpha \mathbf{R} \mathbf{p}_{1i} + \mathbf{t} + \eta_i$ where $\mathbf{R}$ is an unknown rotation matrix, $\mathbf{t}$ is an unknown translation vector, α is an unknown scalar factor and η_i is a vector (unknown) representing noise. Explain how you will extend the method we studied in class for estimation of $\mathbf{R}$ to estimate α and $\mathbf{t}$ as well. Derive all necessary equations (do not merely guess the answers). [5 points]

Ans. Let

$$P = \begin{bmatrix} x_{11} & x_{12} & ... & x_{1n} \\ y_{11} & y_{12} & ... & y_{1n} \end{bmatrix}$$

and

$$Q = \begin{bmatrix} x_{21} & x_{22} & ... & x_{2n} \\ y_{21} & y_{22} & ... & y_{2n} \end{bmatrix}$$

We need to solve the optimization problem $Q = \alpha RP + T + N$ where

$$T = \begin{bmatrix} t_x & t_x & ... & t_x \\ t_y & t_y & ... & t_y \end{bmatrix}$$

is a repetition matrix of the translation column vector and N is the noise matrix.
Rotation and translation don't effect the standard deviation of the set of salient points but only scaling does effect. Hence, we can determine the unknown α by ratio of standard deviation of the two sets of points. We define the single dimensional SD as follows

$$SD1 = \sqrt{\frac{(x_{11} - \bar{x_1})^2 + (y_{11} - \bar{y_1})^2 + ...}{n}} \tag{12}$$

$$SD2 = \sqrt{\frac{(x_{21} - \bar{x_2})^2 + (y_{21} - \bar{y_2})^2 + ...}{n}} \tag{13}$$

where $\bar{x_i} = \frac{\sum_{j=1}^{n} x_{ij}}{n}$ and $\bar{y_i} = \frac{\sum_{j=1}^{n} y_{ij}}{n}$
We can thus obtain α as follows

$$\alpha = \frac{SD2}{SD1} \tag{14}$$

Define

$$\bar{p} = \begin{bmatrix} \bar{x_1} \\ \bar{y_1} \end{bmatrix}, \bar{q} = \begin{bmatrix} \bar{x_2} \\ \bar{y_2} \end{bmatrix}$$

and

$$\mu_p = \begin{bmatrix} \bar{x_1} & \bar{x_1} & ... \\ \bar{y_1} & \bar{y_1} & ... \end{bmatrix}, \mu_q = \begin{bmatrix} \bar{x_2} & \bar{x_2} & ... \\ \bar{y_2} & \bar{y_2} & ... \end{bmatrix}$$

Now, under the transformation $\alpha RP + T + N$, we've

$$\bar{q} = \alpha R\bar{p} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} = \alpha R\mu_p + \bar{t} \tag{15}$$

Here, the noise is assumed to be zero mean and hence doesn't appear in avg. Hence, $\bar{t} = \bar{q} - \alpha R\bar{p}$. Substituting this back in the transformation, we get

$$Q = \alpha RP + \mu_q - \alpha R\mu_p \tag{16}$$

$$Q - \mu_q = \alpha R(P - \mu_p) \tag{17}$$

Let $Q^{'} = Q - \mu_q$ and $P^{'} = P - \mu_p$. We now have $Q^{'} = R(\alpha P^{'})$. This is similar to the procrustes problem with $\bar{t} = 0$ that was discussed in the handout. We can determine R from there and t from the equation $\bar{t} = \bar{q} - \alpha R\bar{p}$. We thus have α, R, t.

6. You are given two datasets in the folder `http://www.cse.iitb.ac.in/~ajitvr/CS763_Spring2015/HW1/Calib_data`. The file names are Features2D_dataset1.mat, Features3D_dataset1.mat, Features2D_dataset2.mat and
Features3D_dataset2.mat. Each dataset contains (1) the XYZ coordinates of N points marked out on a calibration object, and (2) the XY coordinates of their corresponding projections onto an image plane. Your job is to write a MATLAB program which will determine the 3×4 projection matrix M such that $P_1 = MP$ where P is a $4 \times N$ matrix containing the 3D object points (in homogeneous coordinates) and P_1 is a $3 \times N$ matrix containing the image points (in homogeneous coordinates). Use the SVD method and print out the matrix M on screen (include it in your pdf file as well). Write a piece of code to verify that your computed M is correct. For any one dataset, repeat the computation of the matrix M after adding zero mean i.i.d. Gaussian noise of standard deviation $\sigma = 0.05 \times max_c$ (where max_c is the maximum absolute value of the X,Y,Z coordinate) to every coordinate of P and P_1 (leave the homogeneous coordinates unchanged). Comment on your results. Include these comments in your pdf file that you will submit. **Tips:** A mat file can be loaded into MATLAB memory using the 'load' command. To add Gaussian noise, use the command 'randn'. [5 points]

Ans. These are reults for dataset 2 when noise is not added:

```
dataset 2:
smallest singular value for A = 0.082374

M =

    0.0087    0.0011   -0.0039    0.9986
    0.0001    0.0092    0.0005   -0.0520
    0.0000    0.0000    0.0000    0.0027


maximum reconstruction error =

    2.4272
```

These are reults for dataset 1 when noise is not added:

```
dataset 1:
smallest singular value for A = 1.7407e-15

M =

    0.2905    0.0532   -0.1866   -0.6283
   -0.0881    0.3264   -0.0881   -0.6010
```

```
    0.0002    0.0002    0.0002   -0.0021


maxerror =

   2.0293e-11
```

Now, Noise has been added to dataset1. These are the results obtained:

```
dataset 1:
adding noise
smallest singular value of equation system = 0.13446

M =

    0.2728    0.0458   -0.1946   -0.6238
   -0.1055    0.3274   -0.0880   -0.6086
    0.0001    0.0002    0.0002   -0.0021


maximum reconstruction error =

   18.8168
```

Although M looks more or less the same, reconstruction error increased quite a bit after noise is added.

7. *In this exercise, you will estimate the homography between a pair of images using the method we studied in class. You should use the well-known SIFT algorithm to (1) detect salient feature points in both the images, and (2) determine pairs of matching points given the two point sets ('matching point pair' refers to points in the two images representing the same physical entity). The code for performing both these tasks is available at* ***http://www.cs.ubc.ca/~lowe/keypoints/****. We may study the internal details of how SIFT works in a separate set of lectures in class, but for this exercise, just assume this package is a magic blackbox. Now, given this set of matching pairs of points produced by the SIFT package, your job is to estimate the homography between the point sets. Write a routine of the form H = homography(im1,im2) where H is the homography matrix that will transform the first image. You will use data from the folder* ***http://www.cse.iitb.ac.in/~ajitvr/CS763_Spring2015/HW1/Homography/****. Do as follows:*

   (a) *Apply the homography transformation in the file 'Hmodel.mat' to the image 'goi1_downsampled.jpg' using reverse warping to generate a warped image. Now estimate the homography that transforms the first image into its warped version. Apply the estimated transformation to the first image (using reverse warping) and display all three images side by side in your report. Also print the model and estimated homography matrices (make sure you normalize both so that $H(3,3) = 1$ in both cases).*

   (b) *Determine the homography that transforms the image 'goi1_downsampled.jpg' to the second image 'goi2_downsam Warp the first image (using reverse warping) and compare it to the second. Display all three images side by side in your report. Also print the estimated homography matrix normalized so that $H(3,3) = 1$.*

   Note: *You may not get perfect answers for the motion estimate due to errors in SIFT, but you should get a reasonable alignment. While warping, crop off the portions of the image that do not fit into the original size. You may use the nearest neighbor method for interpolation during warping. I encourage you to try out this experiment on images of planar surfaces from different viewpoints that you should take with a real camera. You will notice that the warp estimate will often be very wrong due to several incorrect matches (called as 'outliers'). In a subsequent assignment, we will implement a method that will be reasonably immune to these outliers. At that point, we will attempt to mosaic together two or more pictures as well.* ***[6 points]***