Assignment 1: CS 763, Computer Vision

Due: 31st Jan before 11:55 pm

Remember the honor code while submitting this (and every other) assignment. All members of the group should work on and understand all parts of the assignment. We will adopt a zero-tolerance policy against any violation.

Submission instructions: You should ideally type out all the answers in Word (with the equation editor) or using Latex. In either case, prepare a pdf file. For assignment submission, follow the instructions for arrangement of folders and subfolders as given in `http://www.cse.iitb.ac.in/~ajitvr/CS763_Spring2017/HW1/HW1_CameraGeometry.rar`. Create a single zip or rar file obeying the aforementioned structure and name it as follows: A1-IdNumberOfFirstStudent-IdNumberOfSecondStudent-IdNumberOfThirdStudent.zip. (If you are doing the assignment alone, the name of the zip file is A1-IdNumber.zip). Upload the file on moodle BEFORE 11:55 pm on 31st January. Late assignments will be assessed a penalty of 50% per day late. Note that only one student per group should upload their work on moodle. Please preserve a copy of all your work until the end of the semester. If you have difficulties, please do not hesitate to seek help from me.

1. 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 $Z = f$ without any loss of generality. 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. What do you think will happen if the three lines did not pass through O? [10 points]

   Solution: 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 $OR - OQ$ (**As** $OS \cdot OR = OS \cdot OQ = 0$**, so** $OS \cdot (OR - OQ) = 0$). Also the optical axis Oo (where o is the optical center) is orthogonal to $OR - OQ$ (**We assume the optical axis, i.e.** Oo **to coincide with the Z axis. But** $OR - OQ$ **is contained in the** $Z = f$ **plane, and is in fact the same as segment** QR **by the triangle law of vector addition. Thus** Oo **will be orthogonal to** QR). Hence the plane formed by triangle OSo is orthogonal to $OR - OQ$ and hence line oS is perpendicular to $OR - OQ = QR$ (**As** OS **and** Oo **are both perpendicular to** QR**, we must have** oS **perpendicular to** QR **by triangle law.**). 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? **As parallel lines have the same vanishing point, there will no change to the proof.**

   Marking scheme: 2.5 points for each of the four questions to be answered in order to complete the proof.

2. Suppose you have acquired the image of a cricket pitch at the time instant that a ball thrown by the bowler landed on the ground somewhere on the pitch at some point say L_1. Given this image, your task

is to propose two different methods to determine the perpendicular distance from L_1 to the line containing the wickets on the batsman's side. Make use of the standard dimensions of a cricket pitch as seen on `https://en.wikipedia.org/wiki/Cricket_pitch#/media/File:Cricket_pitch.svg`. Assume that the ball and all the sides of the pitch were clearly visible in the image. For both methods, explain (with equations if required) why you do not need a calibrated camera for this calculation, ignoring errors due to discretization of the spatial coordinates. [10+10=20 points]

Solution: Consider the two long parallel sides of the pitch. They appear to intersect at their vanishing point (say V) in the image plane. Consider a line passing through L_1 parallel to these two long sides. It will also have the same vanishing point V and hence line VL_1 intersects the two pitch baselines (one containing the wickets and one after the wickets on the batsman side) at 90 degrees at points B_1 and B_2. This produces four collinear points B_1, B_2, L_1, V whose cross ratio is $\dfrac{B_1L_1 \times B_2V}{B_1V \times B_2L_1}$ which is equal to the cross ratio in 3D space given as $\dfrac{d \times \infty}{\infty \times (d-122)} = \dfrac{d}{d-122}$ where d is the perpendicular distance (in 3D space) from L_1 onto the line containing the batsman side wickets.

The above cross ratio inequality is valid if the points B_1, B_2, L_1, V were expressed in camera coordinates as we have seen in class. But it is also true when their coordinates are expressed in the image coordinate system. Let (x_c, y_c) and (x_{im}, y_{im}) be the coordinates of a point in the camera and image coordinate system respectively. These coordinates are related by an affine transformation. We have seen in class that an affine transformation preserves the ratio of areas and also the ratios of lengths of collinear line segments. This proves that a calibrated camera is not required (except for discretization errors in marking out point coordinates - which can be assumed to be negligible if the camera resolution is fine enough).

Second Method: You use planar homographies for this. Imagine an orthographic image of a cricket pitch. Mark out at least 4 pairs of corresponding points between the image you have and this orthographic image. These points can be the corners of the pitch rectangle. Compute the planar homography that takes you from the image you have to the orthographic image, and find the points corresponding to L_1 in the orthographic image. The distance computation follows easily. This method too won't require a calibrated camera as we have seen in class that homographies can be computed between image coordinates as well, since the transformation between image coordinates and camera coordinates is affine.

Marking scheme: 10 points for each method. For the cross-ratio method, 4 points for identifying that line VL_1 is parallel to the pitch sides in 3D space and 2 points for correct substitution of the cross ratios. 4 points for explaining why a calibrated camera is not required: 2 points for stating that the relationship between point coordinates in image and camera coordinate systems is an affine transformation and 2 points for stating that affine transformation preserve ratios of lengths of collinear line segments (Deduct 1 point if the word collinear is missing). Similarly, 10 points for the homography method - 6 points for stating the method, and 4 points for arguing why a calibrated camera is not needed. Some students may write a second solution using cross-ratios of a different kind - such methods should also be given full credit if they are correct and different from the first method.

3. Here you will develop an application of the concept of vanishing points to camera calibration. Consider two images (I_P and I_Q) of a non-planar scene taken with two pinhole cameras having unknown focal lengths f_p and f_q respectively. Both cameras produce images on a Cartesian grid with aspect ratio of 1 and unknown resolution s_p and s_q respectively. The orientations and positions of the two cameras are related by an unknown rotation (given by a 3×3 rotation matrix $\mathbf{R}$) and an unknown translation (given by a 1×3 vector $\mathbf{t}$). Note that 'position' here refers to the location of the camera pinhole, and 'orientation' refers to the XYZ axes of the camera coordinate system. In both I_P and I_Q, suppose you accurately mark out the corresponding vanishing points of three mutually perpendicular directions ℓ_1, ℓ_2, ℓ_3 in the scene. (Obviously, all three directions were visible from both cameras). Let the vanishing points have coordinates (p_{1x}, p_{1y}), (p_{2x}, p_{2y}) and (p_{3x}, p_{3y}) in I_P, and (q_{1x}, q_{1y}), (q_{2x}, q_{2y}) and (q_{3x}, q_{3y}) in I_Q. Note that these coordinates are in terms of pixel units and the correspondences are known. Given all this information, can you infer $\mathbf{R}$? Can you

infer $\mathbf{t}$? Can you infer f_p and f_q? Can you infer s_p and s_q? Explain how (or why not). (Hint: You can start off by assuming that you knew all the intrinsic parameters and work yourself upwards from there). [20 points]

Ans: Let (o_{px}, o_{py}) and (p_{qx}, o_{qy}) be the optical centers of the two cameras in their respective pixel coordinate systems. Corresponding to the three vanishing points, the direction vectors of the 3 lines in the coordinate system of the first camera are $\mathbf{p_1} = (s_p(p_{1x} - o_{px}), s_p(p_{1y} - o_{py}), f_p)$, $\mathbf{p_2} = (s_p(p_{2x} - o_{px}), s_p(p_{2y} - o_{py}), f_p)$ and $\mathbf{p_3} = (s_p(p_{3x} - o_{px}), s_p(p_{3y} - o_{py}), f_p)$ respectively. Corresponding to the three vanishing points, the direction vectors of the 3 lines in the coordinate system of the second camera are $\mathbf{q_1} = (s_q(q_{1x} - o_{qx}), s_q(q_{1y} - o_{qy}), f_q)$, $\mathbf{q_2} = (s_q(q_{2x} - o_{qx}), s_q(q_{2y} - o_{qy}), f_q)$ and $\mathbf{q_3} = (s_q(q_{3x} - o_{qx}), s_q(q_{3y} - o_{qy}), f_q)$ respectively. We will first assume that all the intrinsic parameters are known, i.e. $f_p, f_q, s_p, s_q, o_{px}, o_{py}, o_{qx}, o_{qy}$ are all known.
We will unit normalize all these vectors by dividing the vectors by their respective magnitude. Let us denote the resultant unit vectors are $\tilde{\mathbf{p}}_1, \tilde{\mathbf{p}}_2, \tilde{\mathbf{p}}_3, \tilde{\mathbf{q}}_1, \tilde{\mathbf{q}}_2, \tilde{\mathbf{q}}_3$ respectively. The corresponding direction vectors will now be related as follows:

$$\left(\tilde{\mathbf{p}}_\mathbf{1}^t | \;\; \tilde{\mathbf{p}}_\mathbf{2}^t | \;\; \tilde{\mathbf{p}}_\mathbf{3}^t \right) = \mathbf{R} \left(\tilde{\mathbf{q}}_\mathbf{1}^t | \;\; \tilde{\mathbf{q}}_\mathbf{2}^t | \;\; \tilde{\mathbf{q}}_\mathbf{3}^t \right) \tag{1}$$

Using this, you can determine $\mathbf{R}$ using matrix inversion (or transposition), assuming that the three sets of lines in 3D space were not coplanar (else the matrix on the RHS of the previous equation will not be invertible).
However, the translation parameters **cannot** be determined, because translation of a line without changing its direction does not change the coordinates of the vanishing point. The vanishing points therefore carry no information pertaining to translation.

Now what if the intrinsic parameters are unknown? In pixel coordinates, we can determine o_{px}, o_{py} and o_{qx}, o_{qy} from the property that the optical center is the orthocenter of the vanishing points corresponding to three mutually perpendicular lines. Also, the directions $\tilde{\mathbf{p}}_\mathbf{i}$ and $\tilde{\mathbf{p}}_\mathbf{j}$ are perpendicular for $i \neq j$, we have $s_p^2(p_{1x} - o_{px})(p_{2x} - o_{px}) + s_p^2(p_{1y} - o_{py})(p_{2y} - o_{py}) + f_p^2 = 0$, from which we can determine $\frac{s_p}{f_p}$, and likewise for $\frac{s_q}{f_q}$. Thus we can express s_p and s_q as constant multiples of f_p and f_q respectively. Now, as all three coordinates of an image point contain a term in f_p (or f_q), we can determine the directions of the three lines even if we don't know f_p (or f_q).
Marking scheme: 5 points for deriving the directions of the 3 lines in the coordinate systems of the two cameras (respectively) given the coordinates of the vanishing points in the camera coordinate system, i.e. assuming that the intrinsic coordinates were already known. 2.5 points for showing how the rotation matrix can be obtained (it can be obtained even from a single pair of lines as there are only 3 degrees of freedom). 2.5 points for explaining why the translation matrix cannot be obtained as the vanishing point coordinates are not affected by translation. 2 points for stating how to get the coordinates of the optical center in the respective image coordinate systems using the orthocentre property. 4 points for using the fact that the original lines were perpendicular, and 2 points to obtain an expression for s_p and s_q as direct multiples of f_p and f_q respectively. 2 points for stating that we cannot recover f_p and f_q but that does not pose any problem for us as the directions of the lines can still be found out.

4. Let $\boldsymbol{P}$ be the coordinates of a point in the world coordinate system. Let $\boldsymbol{r_1}, \boldsymbol{r_2}, \boldsymbol{r_3}$ respectively be the X, Y, Z axes of a camera's coordinate system expressed in the world coordinate system. Let the center of projection of the camera in the world coordinate system be vector $\boldsymbol{t}$. Express the coordinates of an orthographic image of $\boldsymbol{P}$ on the camera plane in the world coordinate system. [10 points]

   Answer: The coordinates of the orthographic image of $\boldsymbol{P}$ in the world coordinate system are (X, Y, f) where f is the distance between $\boldsymbol{P}$ and the screen. Roughly 3 points for each component. In the camera coordinate system the coordinates would have been $(\boldsymbol{r_1}^t(\boldsymbol{P} + \boldsymbol{t}), \boldsymbol{r_2}^t(\boldsymbol{P} + \boldsymbol{t}), 0)$.

5. You are given two datasets in the folder Q5 within `http://www.cse.iitb.ac.in/~ajitvr/CS763_Spring2017/HW1/HW1_CameraGeometry.rar`. 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 $\mathbf{M}$ such that $\mathbf{P_1} = \mathbf{MP}$ where $\mathbf{P}$ is a $4 \times N$ matrix containing the 3D object points (in homogeneous coordinates) and $\mathbf{P_1}$ is a $3 \times N$ matrix containing the image points (in homogeneous coordinates). Use the SVD method and print out the matrix $\mathbf{M}$ on screen (include it in your pdf file as well). Write a piece of code to verify that your computed $\mathbf{M}$ is correct. For any one dataset, repeat the computation of the matrix $\mathbf{M}$ 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 across all points) to every coordinate of $\mathbf{P}$ and $\mathbf{P_1}$ (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'. For ease of grading, use the 'uigetdir' command in MATLAB to allow a user to choose a folder containing your input data (i.e. the aforementioned mat files). [20 points]

Solution: In this section, we have to solve the equation $\boldsymbol{Am} = \mathbf{0}$ where $\boldsymbol{A}$ is a matrix of size $N \times 12$ (N = number of points) and $\boldsymbol{m}$ is a vector of 12 elements containing elements of the projection matrix $\boldsymbol{M}$ (up to an unknown scale factor). We solve using SVD and extract the singular vector (in $\boldsymbol{V}$ where $\boldsymbol{A} = \boldsymbol{USV}^T$) with the smallest singular value - or equivalently the eigenvector of $\boldsymbol{A}^T\boldsymbol{A}$ with the least eigenvalue. Note that the question did not ask you to solve for individual parameters. You can perform a sanity check on your computation by projecting the original 3D points using the computed matrix and then computing a mean squared error between the given 2D points and the computed points. Check the code in the homework folder.

Marking scheme: 12 points for the code estimating $\mathbf{m}$, 4 points for verifying that it is correct by checking that $\|\mathbf{p} - \mathbf{MP}\|_F^2$ is small, and 4 points for the noise case.

6. Consider the image in the folder Q6 within `http://www.cse.iitb.ac.in/~ajitvr/CS763_Spring2017/HW1/HW1_CameraGeometry.rar`. It is an image of two sheets of paper placed on the surface of a table. One (top right) is an A4 sheet of paper, i.e. it has width 21 cm and height 29.7 cm (these are its actual dimensions - obviously its apparent dimensions in the image are different). The second sheet of paper (bottom left) has unknown dimension. It is safe to assume that the surface of the table and both sheets of paper are completely coplanar. Your job is to design and implement a semi-automatic method (i.e. you can ask the user to click on some salient points) to determine the dimensions of the second sheet of paper using the fact that the first one has known dimensions. For ease of grading, use the 'uigetdir' command in MATLAB to allow a user to choose a folder containing your input data (i.e. the image provided). [20 points]

   Marking scheme: The method is based on planar homography between the given image of the A4 sized paper and an orthographic image of the same. The homography is to be computed between the 4 corner points of the A4 sheet. Identifying that this is the method to be followed carries 8 points. Actual implementation carries 8 points. 2 points for correctly stating the dimensions of the second sheet - in this case it is 22 cm by 14.1 cm (differences up to $\pm$ 1 cm are not to be penalized). 2 points for showing that the students verified the correctness of their implementation - for example, by trying on their own dataset with known dimensions and showing results for the same.