הבינתחומי הרצליה
בית ספר אפי ארזי למדעי המחשב

Assignment 2: Geometry
Computer Vision, Fall 2018
Yael Moses

Submission date: **8.12.2018**
(Note, you may have overlapping with HW3.)
You can submit this assignment in pairs.

A: Hands on Projection & Epipolar Geometry

Follow the attached script, scr2.m, add the requested functions and answer the questions.

1. You are given partial parameters of two cameras in scr2. Fill the missing values as requested.
2. Given object points in the world coordinate system, P=(-140,50,1200) and Q=(30,100,2000),
   a. What are the coordinates (non-homogenous) of the points in the first camera coordinate system?
   b. What are the coordinates (non-homogenous) of the points in the second camera coordinate system?
3. Project and display P=(-140,50,1200) and Q=(30,100,2000) to the two images.
4. **Question – Not for submission:** Look at the projected of each of the points in the two images. One pair looks as expected, and the other not. Think what may be the reason.
5. Compute the epipoles, e_L and e_R, directly from M_L and $M_$R and the COPs.
6. Compute the fundamental matrix, F.
7. Check that the computed epipoles are consistent with F.
8. Choose a set of points from im1, compute the epipolar lines, and display them on the images.
9. We will use matalb functions to compute image features (SURF), match them, and display them. Follow the instructions in scr2.m Your part:
   a. Write a matching function.
   b. Remove all matching pairs from Points_L and Points_R that do not satisfy the epipolar constraints using the Sampson distance (attached matlab function – sampsonDistance.m).
   c. Display the new results using the function *showMatchedFeatures ()*
10. **Extra (not to submit)**
   a. Read any two images with overlaping regions (you can capture them yourself).
   b. Compute the matched features and display them as in **9c**
   c. Compute F using RANSAC:
   F = estimateFundamentalMatrix(M_L,M_R,'Method', 'RANSAC', 'NumTrials', 2000, 'DistanceThreshold', 1);

d. Repeat 9b and 9c using the computed F.

Usefull matlabe tips and things to note:

1. For displaying an epipolar line compute the y values of the two points q=[1,y] and p=[size(im,2)),y]. Then plot([q(1), p(1)],[q(2),p(2)].
2. Note that the y axis in Matlab is top down rather than bottom up - therefore to be consistent with the image coordinates, also the Y in the world coordinates is top down.
3. To compute the null space of a matrix, you can use the matlab function "null".

Part B: Hands on Triangulation

Triangulation

1. Write a stereo function that receives a matrix with a set of points location in the left image, p_L, and its corresponding points in the right image, p_R. The function returns a set of points, P, in 3D.
   P=stereo_list(ps1,ps2, M_L,M_R),
   The dimensions of p_L and p_R are $m \times 2$ where *m* is the number of points.
   The dimensions of P are $m \times 3$ where *m* is the number of points.
   Note: in class we learned how to solve for λ of each of the cameras. The solution can be two different 3D points, one on each ray. Use the average of these points to compute the 3D point in space.

2. To verify that your stereo function is correct:
   Choose a pair of corresponding points by hand, compute its 3D location, and project it back to the two images.

Usefull matlabe tips:

1. To solve the triangulation: you should find the lambdas that minimize ||Ax-b||^2 where $A = (U_L, U_R)$ and $b = C_L - C_R$ and $x = (\lambda_L, \lambda_R)^T$. To do so you can use the matlab function x=A\b. Then you should use the lambdas to compute the two points, one on each line, and average between the, to compute the desired P.
2. Note: if the results of $M^+\tilde{p} = (X, Y, Z, 0)^T$, it indicates that there is a point at infinity. In case $C = (0,0,0)^T$, the direction is given by u=(X,Y,Z).

Part C: Hands on Correspondence and simple Triangulation

In this part of the assignment, you will write a stereo algorithm, with a naïve matching.

a. Read the two images view1.png and view5.png

b. The image planes are co-planar. The distance between the cameras is 160mm.

c. Write an algorithm that receives two rectified images (coplanar and parralele to the line connecting the two COP, with the same focal length), and compute a naive disparity along corresponding epipolar lines:
For each pair of pixels (one from each image) compute the distance between a rectangles patches around it using the cosine distance
defined by $\frac{v_1}{\|v_1\|} \cdot \frac{v_2}{\|v_2\|}$, where v_i is a descriptor of the patch. For example, you can use just the intensity in a 3×4 patch, givne by:
$v_1 = reshape(I(i-1{:}\,i+1, j-1{:}\,j+1), [1,9])$.

The correspondence is defined to be the one with the minimal distance. The algorithm should accept as a parameters:

(i) The two images
(ii) The size of the patch (s_x, s_y).
(iii) The disparity range (d_{min}, d_{max})

The output of this algorithm is a matrix D that consists of the disparity of each pixel.

d. Apply your function to view1.png and view5.png, and display the disparity map as an image. Note, use imshow(D,[]);
The disparity range is [40-120] (the x location in view1 – the x location in view5).

e. To test your algorithm, you can back project the points of view1.png to view5.png, and look at the absolute value of their differences. By 'back project' we mean to compute view5.png using view1.png by using the computed disparity.
You can use the matlab finction imwarp(im,D2d) where D2d is an $m \times n \times 2$ matrix ($m \times n$ is the image size), and D2d(:,:,1)=0 and D2d(:,:,2)=D;
Display the result.
Answer: Why do we set D2d(:,:,1)=0?

f. Compute the depth map using the disparity. Add to your disparity depth map the value 100, since images were cropped. Note – simple triangulation can be applied here. Display it as an image.
Assume that scaled focal lengths (f in the presentation) are $\alpha\_x = \alpha_y = 1$.

g. Repeat (c-f) using the values of the gradients of the images instead of the intensities for computing the descriptors of $v_1$and v_2, to compute the disparity.

h. **Question:** discuss the difference in the results. How the patch size affects the results? How the assumption of order preserving affects the results?

i. **Question:** Which regions have more errors? Why?

j. **BONUS:** use dynamic programing to compute an optimal order preserving disparity.

Submit:
A zip file which includes the following files:

A. All of the documented functions you wrote and all the files required to run them (except for the input images and the toolbox).
B. Assignment2.doc :
Your name and id.
Answers to questions.
C. Scr2.m :
The script file filled with all the missing parts.
Note: When choosing points by hand – please save the list of points for the grader.

Good Luck