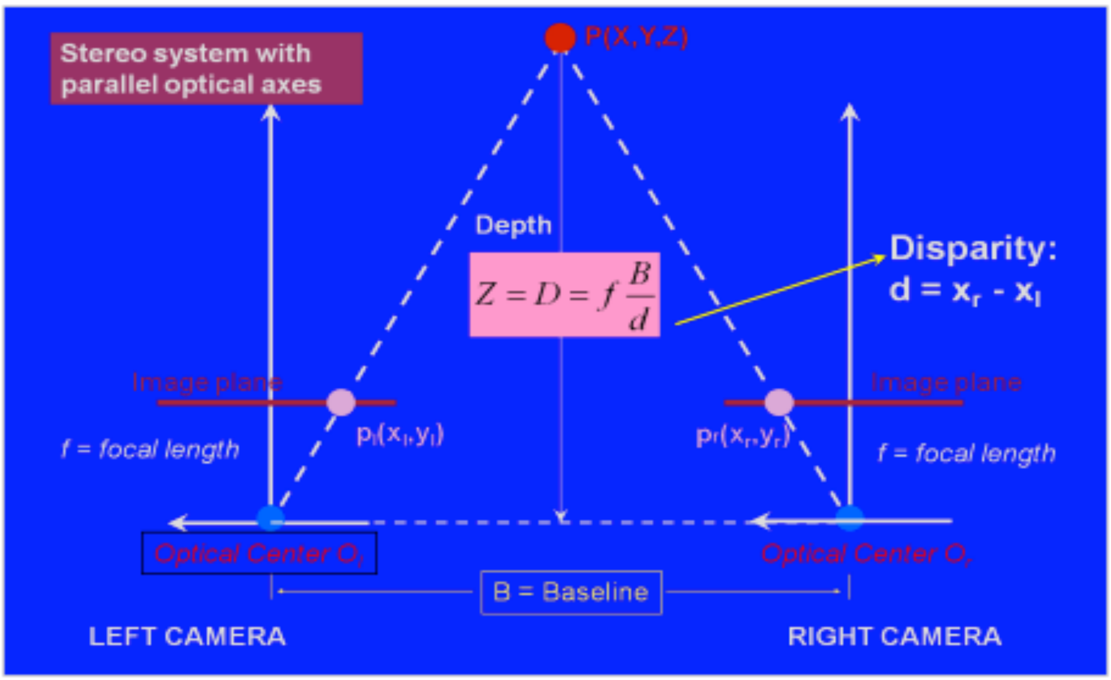
ASSIGNMENT – 4

CSc I6716
Computer Vision

Name – Jatin Jain
EMPLID # 24145250
PhD (Electrical Engineering)

1. (Stereo- 20 points) Estimate the accuracy of the simple stereo system (Figure 3 in the lecture notes of stereo vision) assuming that the only source of noise is the localization of corresponding points in the two images. Please derive (12 points) and discuss (8 points) the dependence of the error in depth estimation of a 3D point as a function of (1) the baseline width, (2) the focal length, (3) stereo matching error, and (4) the depth of the 3D point.
Hint: D = f B/d; Take the partial derivatives of D with respect to the disparity d.

Ans.



From the figure above, in order to find the position of a point in a disparity image, we can use the equation

$$Z = D = f\frac{B}{d}$$

Z = the depth, or distance along the camera z-axis
f = the focal length
B = the baseline width between the two cameras
d = disparity, or distance between two projected points

To determine the depth error, we can calculate the partial derivatives of Z with respect to the variable of uncertainty. Assuming the only uncertainty is in the disparity

$$\rightarrow \quad Z(d) = f\frac{B}{d}$$

$$\rightarrow \quad d = \frac{fB}{Z}$$

$$\rightarrow \quad d^2 = \frac{(fB)^2}{Z^2}$$

$$\rightarrow \quad \frac{\partial Z}{\partial d} = \frac{\partial\left(f\frac{B}{d}\right)}{\partial d}$$

$$= \frac{fB\,\partial\left(\frac{1}{d}\right)}{\partial d}$$

$$= \frac{-\frac{fB}{\partial^2}\,\partial d}{\partial d}$$

$$= -\frac{fB}{\partial^2}$$

$$|\partial Z| = \left|\frac{-fB}{d^2}\,\partial d\right|$$

$$= \frac{fB}{d^2}\,\partial d = \frac{fB}{\frac{(fB)^2}{Z^2}}\,\partial d = V\frac{Z^2}{fB}\,\partial d$$

1. The baseline width: Depth error is inversely proportional to the baseline width, meaning a larger baseline width will provide better depth accuracy but a smaller field of view (FOV)

2. The focal length: Depth error is inversely proportional to the focal length. A larger focal length will provide a better depth accuracy but a smaller FOV

3. The stereo matching error: Depth error is proportional to the stereo matching error, that is stereo matching error increases, then the error increases.

4. The depth of the 3D point: Depth error is proportional to the square of the depth, indicating that the depth error is a quadrate function of the depth itself. This means that the nearer the point is, the more accurate the depth estimation, that is Z increases then the error increases.

2. (Motion- 20 points) Could you obtain 3D information of a scene by viewing the scene by using multiple frames of images taken by a camera rotating around its optical center (5 points)? Discuss why or why not (5 points). What about translating (moving, not zooming!) the camera along the direction of its optical axis (5 points)? Explain. (5 points)

Ans. The camera has to physically move, and it can be rotated in order to gather 3D information. Another way of gathering 3D information is if the object is in motion. It could be said 3D Motion can be characterized by a rotation matrix and a translation matrix.

$$\begin{pmatrix} V_x \\ V_y \end{pmatrix} = \underbrace{\frac{1}{f}\begin{pmatrix} xy & -(x^2+f^2) & fy \\ y^2+f^2 & -xy & -fx \end{pmatrix}\begin{pmatrix} w_x \\ w_y \\ w_z \end{pmatrix}}_{\text{Rotational part : no depth information}} +$$

$$+ \underbrace{\frac{1}{Z}\begin{pmatrix} -f & 0 & x \\ 0 & -f & y \end{pmatrix}\begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix}}_{\text{Translation part: depth } Z.}$$

Under pure rotation $(T=0)$ the rotation field equation will be:

$$\begin{pmatrix} V_x \\ V_y \end{pmatrix} = \frac{1}{f}\begin{pmatrix} xy & -(x^2+f^2) & fy \\ y^2+f^2 & -xy & -fx \end{pmatrix}\begin{pmatrix} w_x \\ w_y \\ w_z \end{pmatrix}$$

Since z, is not included in the equa we cant obtain 3D information by viewing the ~~scene~~ scene by rotating the camera around, its optical center

using $(w=0)$ pure translation, the motion field can be simplified as

$$\begin{pmatrix} V_x \\ V_y \end{pmatrix} = \frac{1}{Z}\begin{pmatrix} -f & 0 & x \\ 0 & -f & Y \end{pmatrix}\begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix}$$

We can Obtain 3D Information through the above equation, even though camera is only translating along its optical axis.

So, there is no depth information, which means there is no Z value inside the formula, and it has no 3D information. This is the equation of rotational part. As we can see, there is no depth information. This is because we do not have a Z variable anywhere here. If we look at this from a practical perspective, we can see that if we were to take a camera and just rotate it, all we would get is a panoramic image. It would just be very wide image. There would be no depth information because we do not have 2 reference points to compare. For translation along the optical axis, there is no depth information. This is because you are simply changing the focal length if you move the camera forward or backwards. You need to have some distance in the X or Y direction in order to capture the 3D information.

3. (Motion- 20 points) Explain what the aperture problem (10 pts) is, and how it can be solved if a corner is visible through the aperture (10 pts).

Ans. The aperture problem states that the motion of a spatial image cannot be determined if you see if through a small aperture. Because we cannot determine the direction in which the image is going. The motion of a 2D feature, such as a corner is very obvious. With a corner we can see exactly which direction it is going in. If we see a corner, then we are able to determine the direction it is going in. Therefore, solving the aperture problem requires us to find a corner in the image. It might appear to be going in a certain direction, but we cannot be sure. For example, an edge may appear to be moving diagonally, but it will be moving up and down.

Another method to solve the aperture problem can be: Constant Flow Method

Assume that pixel neighbors have the same (u,v)

If, we use a 5x5 window, that gives us – 25 eqn per pixel,

$$0 = I_t(P_i) + \nabla I(P_i) \cdot [u\,v]$$

$$\begin{bmatrix} I_x(P_1) & I_y(P_1) \\ I_x(P_2) & I_y(P_2) \\ \vdots & \vdots \\ I_x(P_{25}) & I_y(P_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} I_t(P_1) \\ I_t(P_2) \\ \vdots \\ I_t(P_{25}) \end{bmatrix}$$

$$\underset{25\times 2}{A} \; \underset{2\times 1}{d} = \underset{25\times 1}{b}$$

$$\underset{2\times 2}{(A^TA)} \; \underset{2\times 1}{d} = \underset{2\times 1}{A^Tb}$$

$$\underbrace{\begin{bmatrix} \sum I_xI_x & \sum I_xI_y \\ \sum I_xI_y & \sum I_yI_y \end{bmatrix}}_{A^TA} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \sum I_xI_t \\ \sum I_yI_t \end{bmatrix}}_{A^Tb}$$

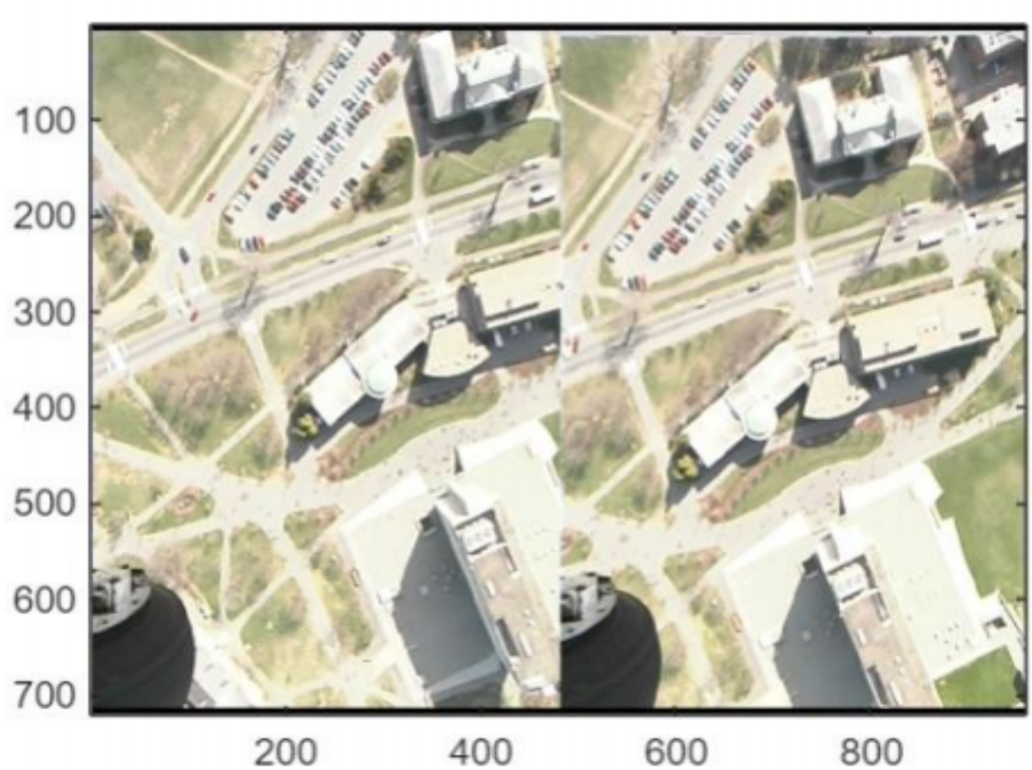
The summation are over all pixels in KxK window

4. (Stereo Programming – 40 points + 5 bonus points) Use the image pair (Image 1, Image 2) for the following exercises.

(1). Fundamental Matrix. – Design and implement a program that, given a stereo pair, determines at least eight point matches, then recovers the fundamental matrix (5 points) and the location of the epipoles (5 points). ***Check the accuracy of the result by measuring the distance between the estimated epipolar lines and image points not used by the matrix estimation (5 points). Also, overlay the epipolar lines of control points and test points on one of the images (say Image 1- I already did this in the starting code below). Control points are the correspondences (matches) used in computing the fundamental matrix, and test points are those used to check the accuracy of the computation.***

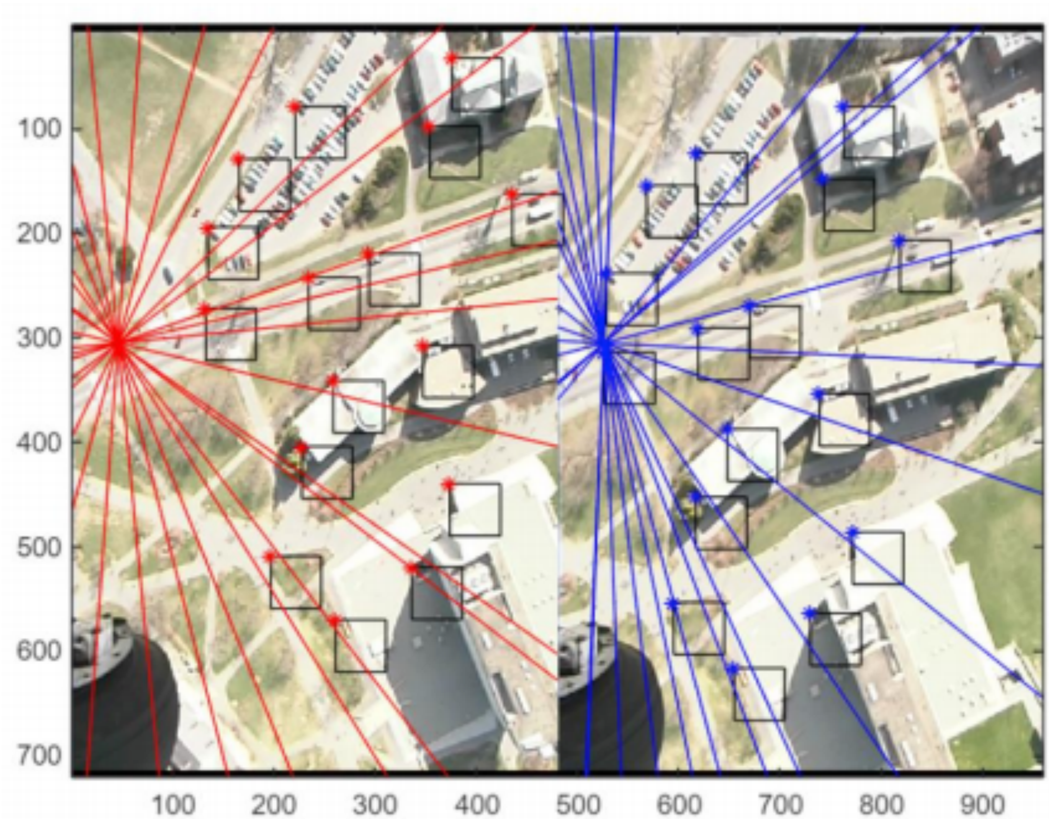
Hint: As a first step, you can pick up the matches of both the control points and the test points manually. You may use my matlab code (FmatGUI.m) as a starting point – where I provided an interface to pick up point matches by mouse clicks. The epipolar lines should be (almost) parallel in this stereo pair. If not, something is wrong either with your code or the point matches. Make sure this is achieved before you move to the second step – that is to try to search for point matches automatically by your program. However the second step is optional (for extra 5 points)*

Ans.



Load the image and Randomly select 16 points on the left and right images respectively For the fundamental matrix, I generated the A matrix and used SVD of A. Then I created the Fundamental Matrix using the A matrix.

F = -0.0000 -0.0000 0.0019
0.0000 0.0000 -0.0030
-0.0017 0.0001 1.0000



Epipolar lines are drawn

Pl =. 307.3537 394.1673 1.0000
359.2976 538.4559 1.0000
353.5261 272.9649 1.0000
236.1713 530.7605 1.0000
151.5220 409.5581 1.0000
199.6182 301.8226 1.0000
93.8066 344.1473 1.0000
149.5982 524.9890 1.0000
322.7445 653.8868 1.0000
143.8267 197.9349 1.0000
363.1453 163.3056 1.0000
270.8006 128.6764 1.0000

268.8768 465.3497 1.0000
282.3437 623.1052 1.0000
155.3697 661.5822 1.0000
51.4820 582.7044 1.0000

Pr= 153.3504 266.3643 1.0000
328.8333 144.7875 1.0000
351.8683 34.1956 1.0000
123.8248 201.1077 1.0000
100.5473 113.3542 1.0000
198.0080 25.3038 1.0000
40.0549 84.8250 1.0000
139.3363 54.4519 1.0000
240.2334 215.5270 1.0000
91.1743 111.2356 1.0000
347.8424 104.3081 1.0000
158.1443 219.8257 1.0000
259.0938 71.8686 1.0000
109.8592 260.0941 1.0000
153.8691 21.5418 1.0000
48.2841 17.8616 1.0000

The epipolar geometry is the intrinsic projective geometry between two views, it is independent of the scence structure and depends on the camera internal parameter Then, we can calculate the distance from a node to line based on the following form:

First getting the location left and the right peepholes
an2 = F*pr(cnt,:)'
0.0003
0.0003
-0.1024

an = F*pl(cnt,:)'
0.0001
-0.0000
-0.0014

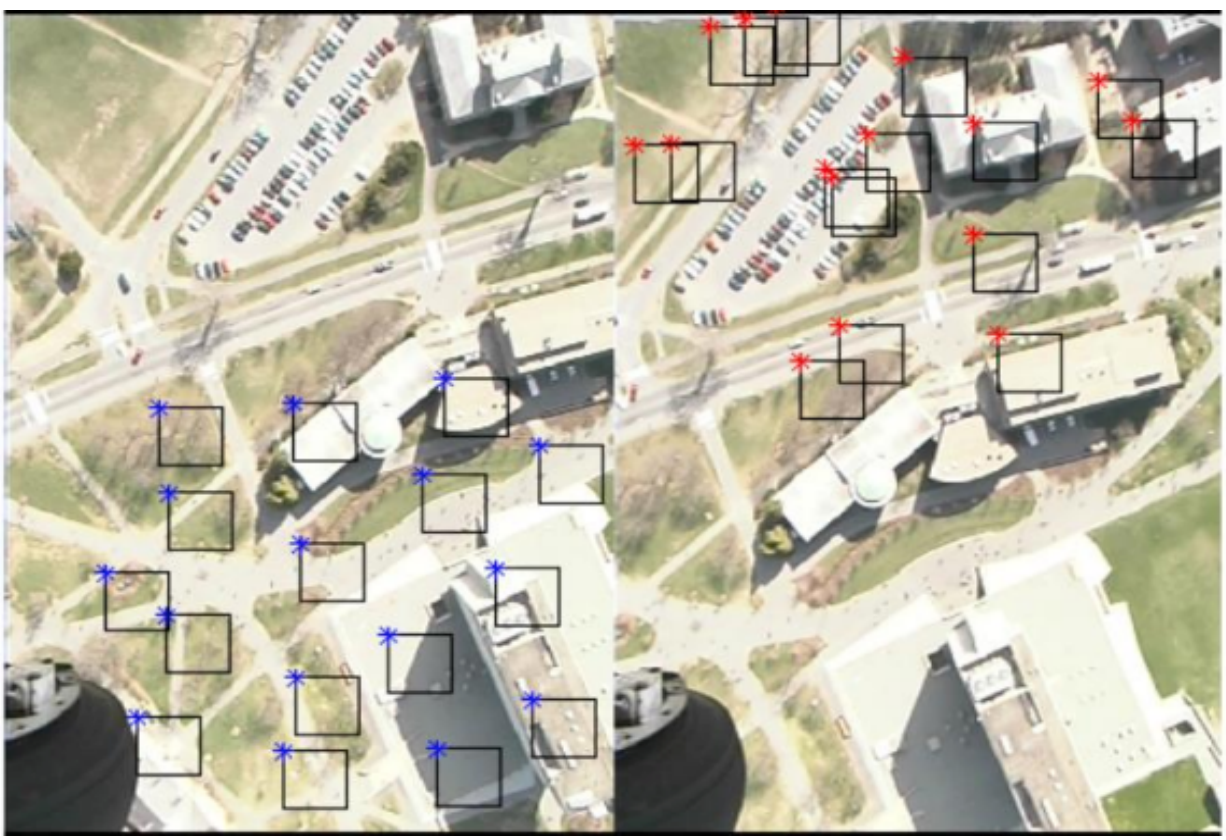
error(cnt) = abs(an(2)*pr(cnt,2)+an(1)*pr(cnt,1)+an(3))/sqrt(an(1)^2+an(2)^2)

2.9095
0.0367
0.5911
2.2249
2.4915
0.9727
1.5409

1.1674
2.8643
3.6046
1.5309
5.3113
0.3153
1.3596
0.4073
0.6003

*(2). Feature-based matching. – Design a stereo vision system to do "feature-based matching" and explain your algorithm in writing – what the feature is, how effect it is, and what are the problems (**5 points**). The system should have a user interface that allows a user to select a point on the first image, say by a mouse click (**5 points**). The system should then find and highlight the corresponding point on the second image, say using a cross hair points). Try to use the epipolar geometry derived from (1) in searching correspondences along epipolar lines (**5 points**).*

Hint: You may use a similar interface as I did for question (1). You may use the point match searching algorithm in (1) (if you have done so), but this time you need to constrain your search windows along the epipolar lines.



Feature based matching Algorithm

Step 1: Extract corners in the stereo pair.
Step 2: Define similarity measure.
Step 3: Search correspondences using similarity to measure and the epipolar geometry.

Problem with stereo vision is

1.It makes stereo system less accuracy and fails to find the matches points when some features are not in the both images.
2.It only creates a sparse depth map, therefore many points in the reference image may not have depth values.

In second plot I had to click on the left image, then corresponding points will be made on the other image after click 16 image the points are matched.

*(3) Discussions. Show your results on points with different properties like those in corners, edges, smooth regions, textured regions, and occluded regions that are visible only in one of the images. Discuss for each case, why your vision system succeeds or fails in finding the correct matches (**5 points**). Compare the performance of your system against a human user (e.g. yourself) who marks the corresponding matches on the second image by a mouse click (**5 points**).*

Ans. When some of the points don’t match, then vision system may fail. When the points are at the center of the images, the vision system will succeed. Most of the line are matched corresponding points in the images.