Computer vision (**CSC I6716)**

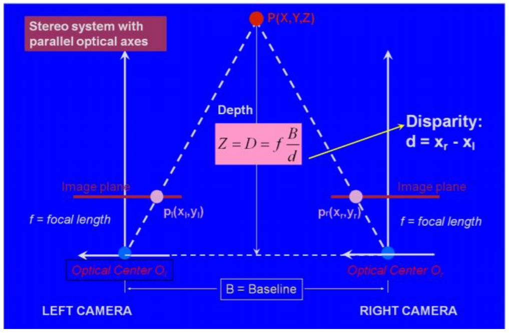
Assignment 4

Prof. Zhigang Zhu

Sonali Shintre

ID: 7602

1. **(Stereo- 15 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 (5 points) and discuss (10 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**.



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

Z = fB/d

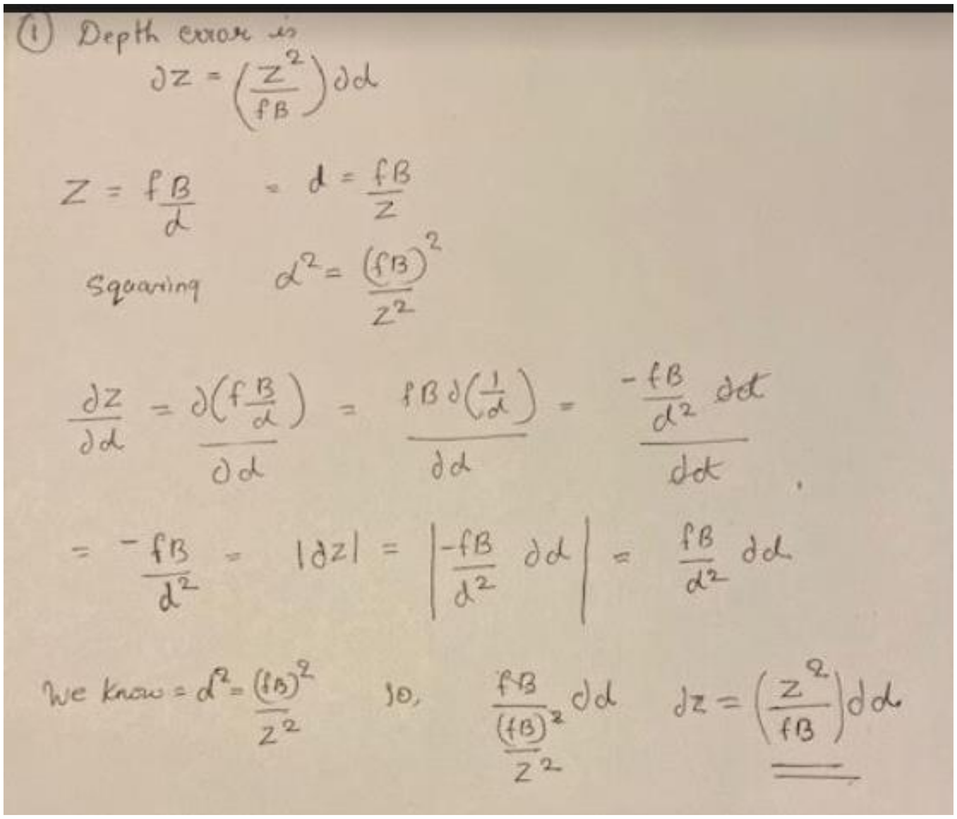
Z = the depth, or distance along the camera z-axis

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



(1) Depth error is

$$\partial Z = \left(\frac{Z^2}{fB}\right)\partial d$$

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

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

$$\frac{\partial Z}{\partial d} = \frac{\partial\left(f\frac{B}{d}\right)}{\partial d} = \frac{fB\,\partial\left(\frac{1}{d}\right)}{\partial d} = \frac{-\frac{fB}{d^2}\,\partial d}{\partial d}\ .$$

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

We know = $d^2 = \frac{(fB)^2}{Z^2}$ so, $\frac{fB}{\frac{(fB)^2}{Z^2}}\partial d$ $\underline{\underline{\partial Z = \left(\frac{Z^2}{fB}\right)\partial d}}$

If the uncertainty is in the baseline width, the depth error with respect to the baseline is equal to ∂Z =(f/d) ∂B

If the uncertainty is in the focal length, the depth error with respect to the focal length is equal to ∂Z = (B/d) ∂f

1. 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.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. stereo matching error

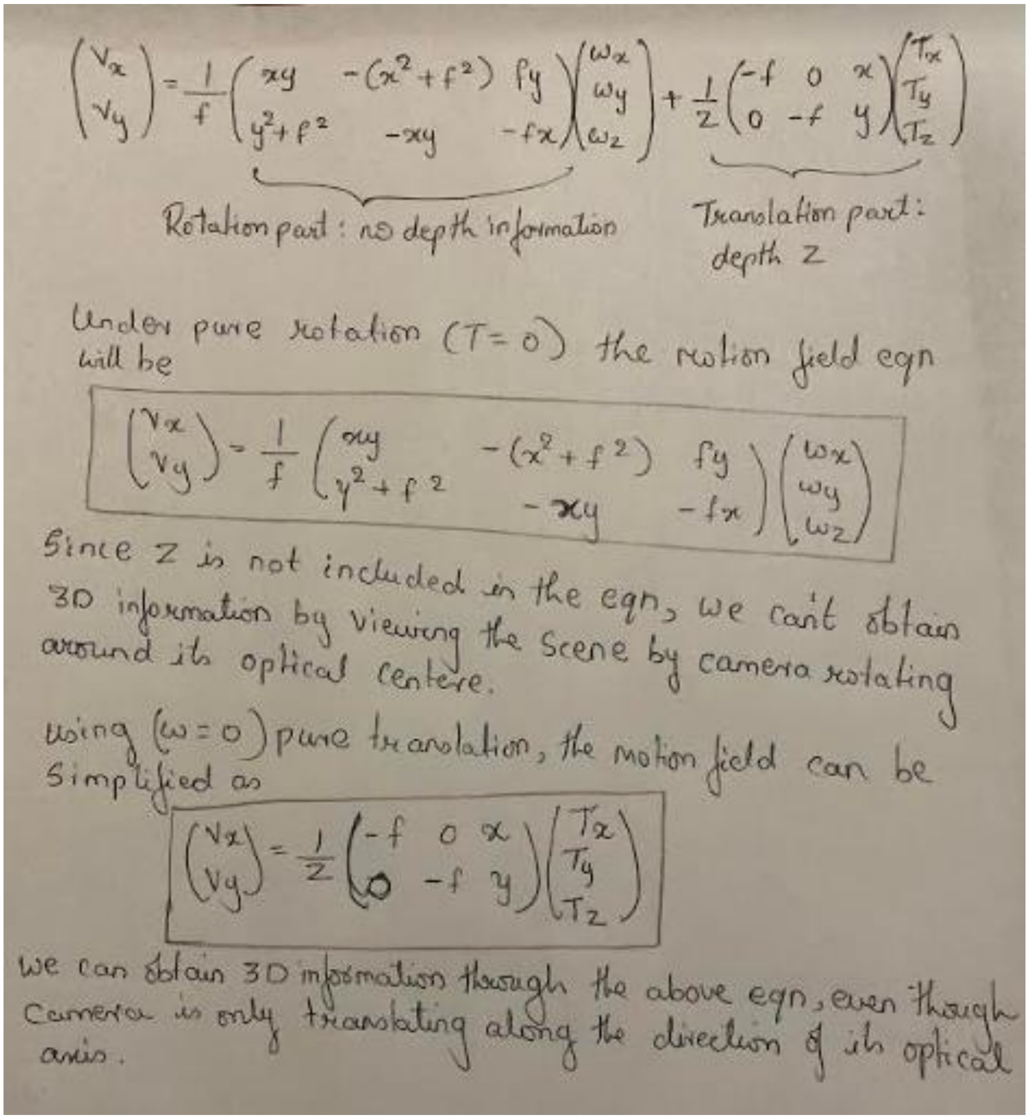
Depth error is proportional to the stereo matching error.

4. 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.

2. (Motion- 20 points) Could you obtain 3D information of a scene by viewing the scene by a camera rotating around its optical center (5 points)? Discuss why or why not (5 points). **What about translating the camera along the direction of its optical axis (5 points)? Explain. (5 points)**

To collect 3D information, Camera has to be physically moved and rotated, If object is in motion then 3D motion is characterized by a rotation matrix and the translation matrix.
So, motion field equation under rotation around the camera's optical center can be written as



$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \underbrace{\frac{1}{f}\begin{pmatrix} xy & -(x^2+f^2) & fy \\ y^2+f^2 & -xy & -fx \end{pmatrix}\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}}_{\text{Rotation part : no depth information}} + \underbrace{\frac{1}{Z}\begin{pmatrix} -f & 0 & x \\ 0 & -f & y \end{pmatrix}\begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix}}_{\text{Translation part: depth } Z}$$

Under pure rotation $(T=0)$ the motion field eqn will be

$$\boxed{\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \frac{1}{f}\begin{pmatrix} xy & -(x^2+f^2) & fy \\ y^2+f^2 & -xy & -fx \end{pmatrix}\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}}$$

Since Z is not included in the eqn, we can't obtain 3D information by viewing the scene by camera rotating around its optical centere.

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

$$\boxed{\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \frac{1}{Z}\begin{pmatrix} -f & 0 & x \\ 0 & -f & y \end{pmatrix}\begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix}}$$

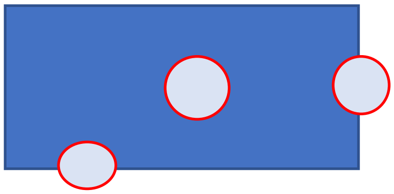
we can obtain 3D information through the above eqn, even though camera is only translating along the direction of its optical axis.

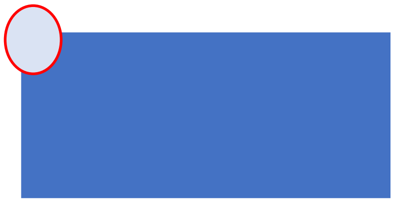
Z is not included in the equation and has no 3D information. The disparity between any two images under this rotation is 0. In order to be able to extract any 3D information, some translational operation must be applied instead. If we move the camera along its optical axis however, 3D information can be obtained because translational images can be captured at different time frames

There would be no depth information because we do not have 2 reference points to compare. For translation along the optical axis. This is because we are simply changing the focal length if you move the camera forward or backwards (zoom in or zoom out). You need to have some distance in the X or Y direction in order to capture the 3D information.

3. **(Motion- 10 points) Explain that the aperture problem can be solved if a corner is visible through the aperture**.

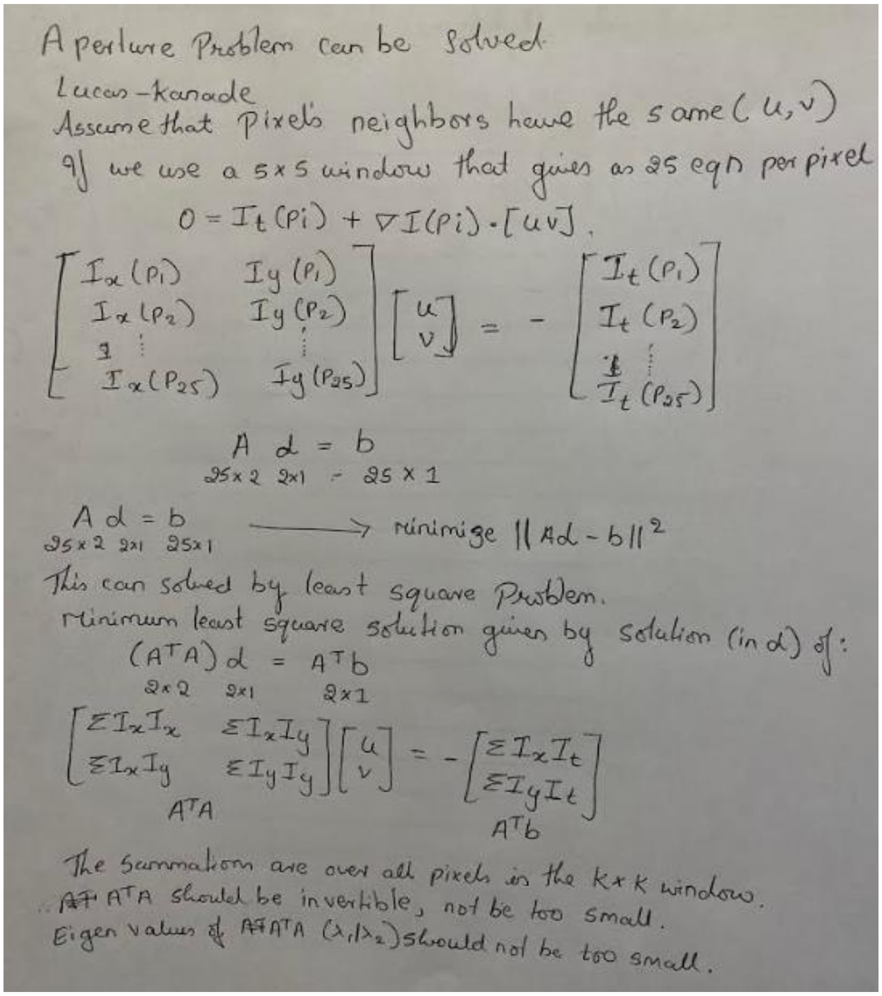
In computer vision we encounter this problem like motion estimation, considering the rectangular box, assume it moving back and forth in horizontal direction, when we try to find the pixels intensity from the middle of the rectangle or from the horizontal, we don't see the motion or change in pixel intensity. On vertical edge we see horizontal motion but vertical edge will not see the vertical motion same with the horizontal edges will not see the horizontal motion.





From the above fig 1 we cannot detect any motion but from the fig 2, image through the edge window we can detect. Since point on the edge is difficult to match then the corner

Best way is through the edges where we can see all kind of the motion both horizontal, vertical and corner feature can be detected, that why while doing motion estimation we first try to find corner edges and resolve the aperture problem, which can be solved through Lucas-kanade.



Aperture Problem can be Solved

Lucas-kanade

Assume that Pixel's neighbors have the same (u, v)

If we use a 5×5 window that gives as 25 eqn per pixel

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

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(P_{25}) & I_y(P_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(P_1) \\ I_t(P_2) \\ \vdots \\ I_t(P_{25}) \end{bmatrix}$$

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

$$\underset{25\times 2}{A}\ \underset{2\times 1}{d} = \underset{25\times 1}{b} \longrightarrow \text{minimize } \|Ad - b\|^2$$

This can solved by least square Problem.

Minimum least square solution given by solution (in d) of:

$$\underset{2\times 2}{(A^T A)}\ \underset{2\times 1}{d} = \underset{2\times 1}{A^T b}$$

$$\underbrace{\begin{bmatrix} \Sigma I_x I_x & \Sigma I_x I_y \\ \Sigma I_x I_y & \Sigma I_y I_y \end{bmatrix}}_{A^T A} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \Sigma I_x I_t \\ \Sigma I_y I_t \end{bmatrix}}_{A^T b}$$

The summation are over all pixels in the k×k window.

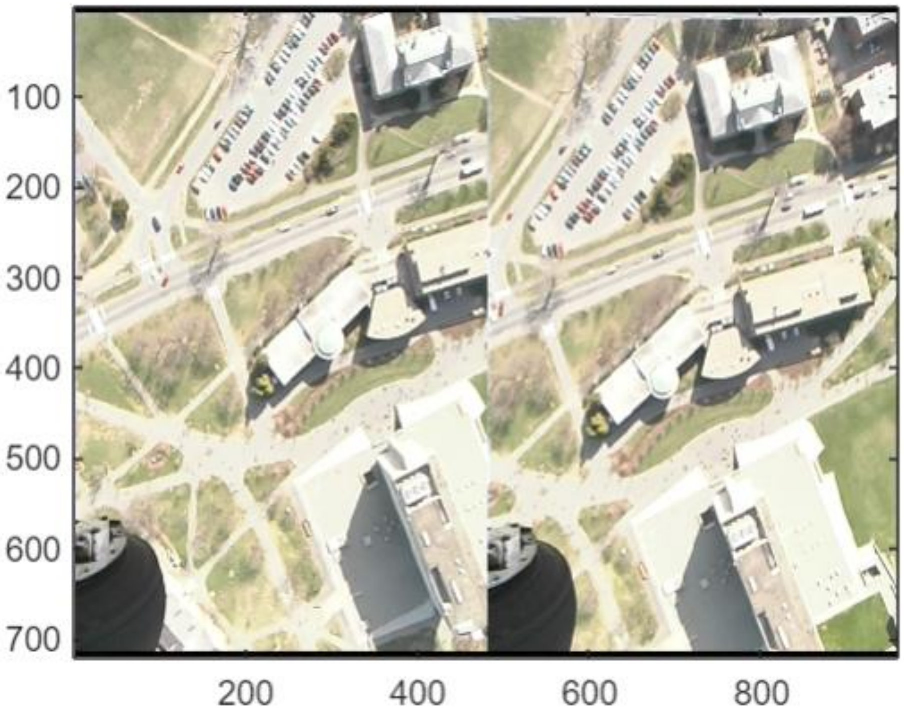
~~A^T~~ $A^T A$ should be invertible, not be too small.

Eigen values of ~~A^T~~ $A^T A$ (λ_1, λ_2) should not be too small.

4. (Stereo Programming – 55 points + 5 bonus points) Use the image pair (Image 1, Image 2) for the following exercises.

Fundamental Matrix.

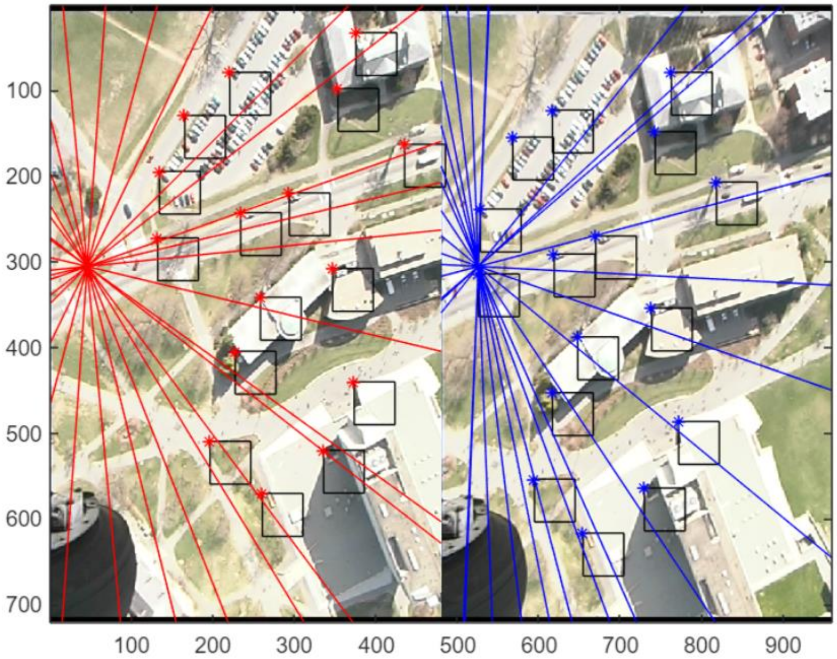
a. Design and implement a program that, given a stereo pair, determines at least eight point matches, then recovers the fundamental matrix (10 points) and the location of the epipoles (5 points).



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
```

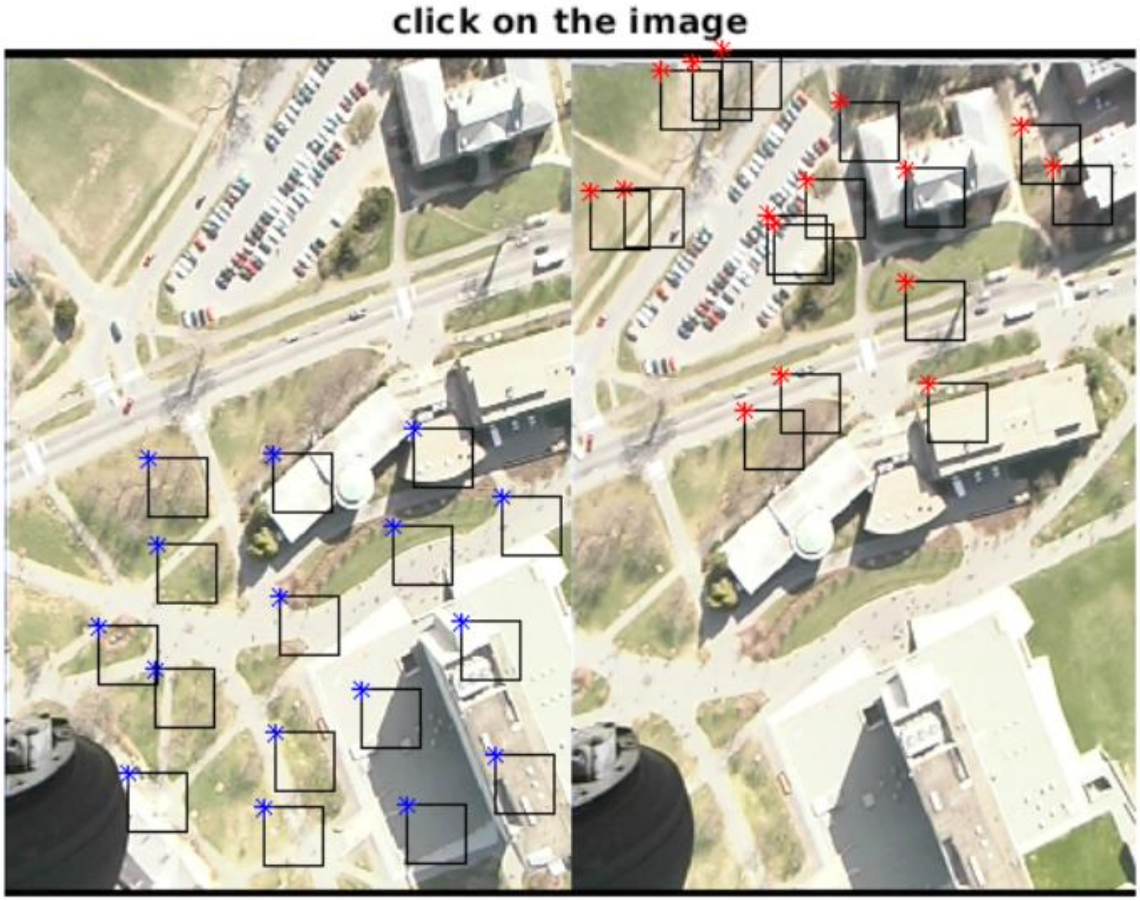
```
an = F*pl(cnt,:)'
    0.0001
   -0.0000
   -0.0014
```

```
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 (**10 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**).**

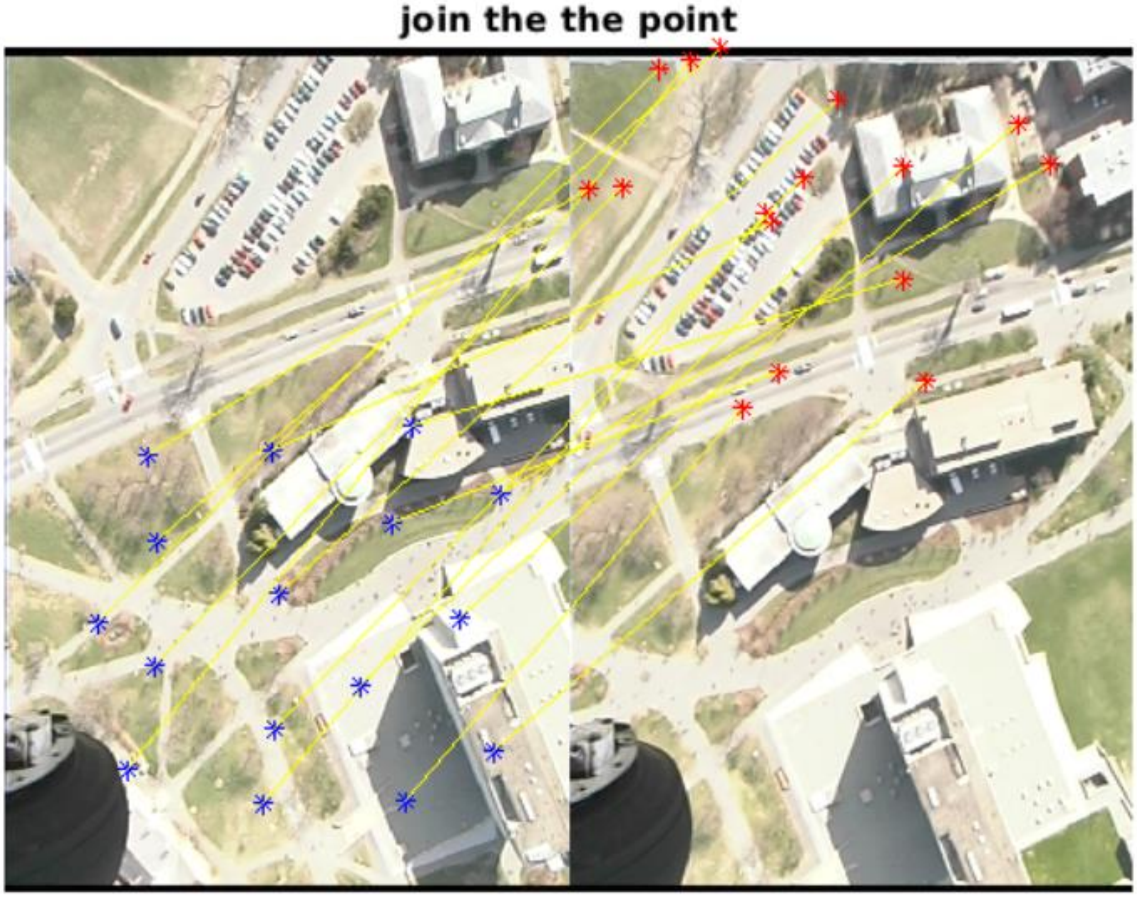


Feature based matching:
Algorithm：

Step 1. Extract corners in the stereo pair.
Step2. 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. Showed in the 3 plot.



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 (5 points). 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).**

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.