Computer Vision: Stereo and Motion
Allen Lau

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.

   Derivation of Depth Estimation Error:

   The depth (Z) in the simple stereo system is defined as the following, where f is the focal length, B is the baseline length, and d is the disparity:

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

   To estimate the accuracy of the simple stereo system, assuming that the only source of noise is the localization of corresponding points in the two images (disparity d), the partial derivative of Z with respect to the disparity, d, will be calculated:

   Using the quotient rule and simplifying, we get:

$$\delta Z = \frac{d(1) - fB}{d^2}\,\delta d$$

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

   The depth equation will be rearranged to solve for disparity, and will be substituted into the above equation, and then the resulting function will be simplified:

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

$$\delta Z = \frac{-fB}{\left(\frac{fB}{Z}\right)^2}\,\delta d$$

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

The magnitude of the resulting function gives the wanted absolute error equation:

$$\delta Z = \frac{Z^2}{fB}\,\delta d$$

Discussion:

1) Error in Depth Estimation vs. Baseline Width
    a. Error in depth estimation is inversely proportional to baseline width. In other words, as the baseline increases, the error decreases. Conversely, as the baseline decreases, the error increases.
2) Error in Depth Estimation vs. Focal Length
    a. Error in depth estimation is inversely proportional to the focal length. In other words, as the focal length increases, the error decreases. Conversely, as the focal length decreases, the error increases.
3) Error in Depth Estimation vs. Stereo Matching Error
    a. Error in depth estimation is directly proportional to the stereo matching error. In other words, as the stereo matching error increases, the error in depth estimation increases. Conversely, as the stereo matching error decreases, the error in depth estimation decreases.
4) Error in Depth Estimation vs. Depth of the 3D Point
    a. Error in depth estimation is directly proportional to the depth of the 3D point. In other words, as the depth increases, the error increases. Conversely, and the depth decreases, the error decreases.

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)

The following equation describes the motion field equation:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \frac{1}{f}\begin{bmatrix} xy & -(x^2+f^2) & fy \\ (y^2+f^2) & -xy & -fx \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} + \frac{1}{Z}\begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix}\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

Pure Rotation Case

In the case where we are viewing the scene by using multiple frames of images taken by a camera rotating around its optical center, this would be a special case with pure

rotation. As a result, the second half of the motion field equation would be zero, resulting in the following:

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

It is observed that the depth (Z) term is not present in the above equation, which proves that 3D depth information cannot be obtained with this pure rotation case. Additionally, from the depth equation in equation 1, we know that depth is a function of the focal length, baseline, and disparity. A camera rotating around its optical center would result in no distance between the optical axes of the resultant cameras for each image, therefore, baseline becomes zero; therefore, the estimated depth becomes zero.

Pure Translation

In the case where the camera is translating along the direction of its optical axis, this would be a case of pure translation, or more specifically, radial motion. The motion field equation simplifies to the following, where it is observed that the depth (Z) variable is present in the equation:

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

In addition, we can define a vanishing point $p_0 = (x_0, y_0)^T$, which can be used to compute the 3D motion direction and use it to derive the depth estimation of in the radial motion field case, as seen below:

$$Z = \frac{|T_z|}{|\bar{v}|} \sqrt{(x - x_0)^2 + (y - y_0)^2}$$

3. (Motion- 10 points) (1) Explain what is the aperture problem, and how it can be solved if a corner is visible through the aperture (10 pts).

An aperture is defined as a small hole in which the camera is viewing the world through. The aperture problem is defined as the issue when only the component of the motion field in the direction of the spatial image gradient can be determined. This problem causes the motion of a moving object to be ambiguous due to the object being viewed being partially obscured by the aperture.

This problem can be solved if a corner is visible through the aperture because corners have sides what result in multiple spatial image gradients, which can be used to track

the motion of the object. As a result of this additional information, the ambiguity in motion is resolved and the motion of the object can be determined.

4. (Stereo and Motion – 10 points): (1) Give 5 examples when humans using stereo or motion in daily life or work (5 points) (2) Give another 5 examples that use computer vision techniques with stereo or motion in real applications.

   5 Examples of Humans Using Stereo / Motion

   - Depth estimation to gauge how far or close an object is from the person when they pick it up.
   - Estimating the speed of a ball being thrown, so that it can be caught.
   - Parallel parking requiring the estimation of depth to gauge how far or close the adjacent cars or curb is to the car.
   - Determining the speed of an oncoming car that may be coming towards your path of motion.
   - Gauging the distance needed to jump over a puddle on the street when it is raining.

   5 Examples of Applications Using Computer Vision Techniques with Stereo / Motion

   - Using video of a person walking through a forest to obtain 3D structural / depth information for research and education purposes.
   - Autonomous vehicle driving using stereo and motion information to drive within human aid.
   - Assembly line robots using stereo information to help it perform a task on an industrial process.
   - Sports analytics using motion concepts to estimate the speed and trajectory of a ball thrown.
   - Security cameras that are tracking a moving object across its field of view.

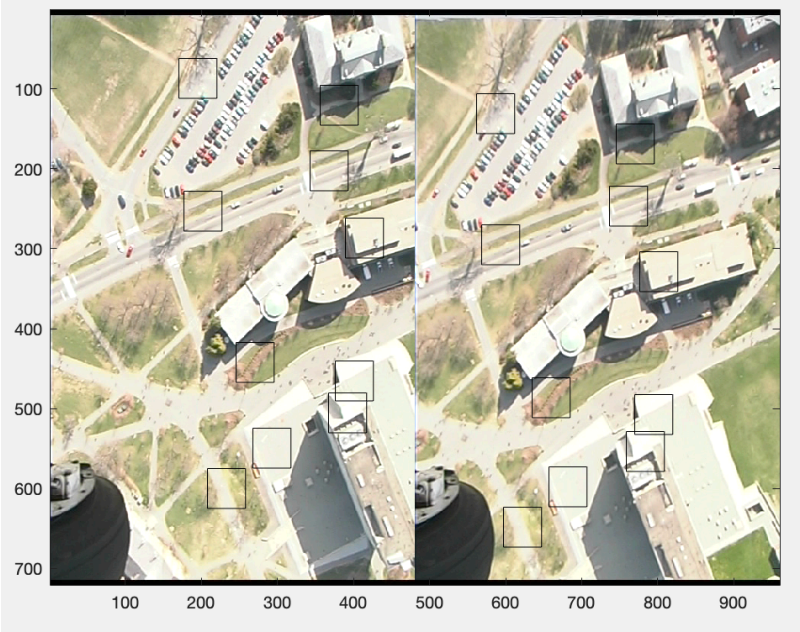
5. (Stereo Programming – 40 points + 5 bonus points)

(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.

This problem describes the scenario where the intrinsic and extrinsic parameters of the stereo camera system are unknown; therefore, the Eight-Point Algorithm will be used to estimate the Fundamental Matrix. The procedure to compute the Fundamental Matrix is described below:

1. Pick the correspondence point pairs between the stereo images, where we define 8 control points to derive the Fundamental Matrix. 2 test points are chosen to compute the accuracy of the result.
2. Normalize the points so that the coefficients of the linear equation system will be less ill-conditioned.
3. Estimate the Fundamental Matrix using the Eight-Point Algorithm:
    a. Given at least 8-point correspondences, construct the homogenous system Ax = 0 from the Fundamental Matrix Equation $\vec{p}_r^T F \vec{p}_l^T = 0$.
    b. Estimate $\hat{F}$ by taking the SVD of A.
    c. Enforce singularity constraint for F.
4. Denormalize F using the transformation matrices used in step 2 to obtain the finalized estimation of the Fundamental Matrix, F.

Figure 1: Stereo Image Pair w/ Correspondence Point Pairs Selected



For the stereo image pair in Figure 1 with the correspondences depicted as the selected for the points on the left image and right image, the aforementioned procedure is used to estimate the fundamental matrix, as described below:

$$Estimated\ Fundamental\ Matrix\ (F) = \begin{bmatrix} -8.5753e-06 & 1.7156e-06 & 0.00189 \\ 5.2702e-07 & 6.4762e-07 & -4.0828e-04 \\ 0.001472 & -4.5510e-04 & -0.26259 \end{bmatrix}$$

With the estimated Fundamental Matrix (F), the following procedure is used to estimate the epipoles:

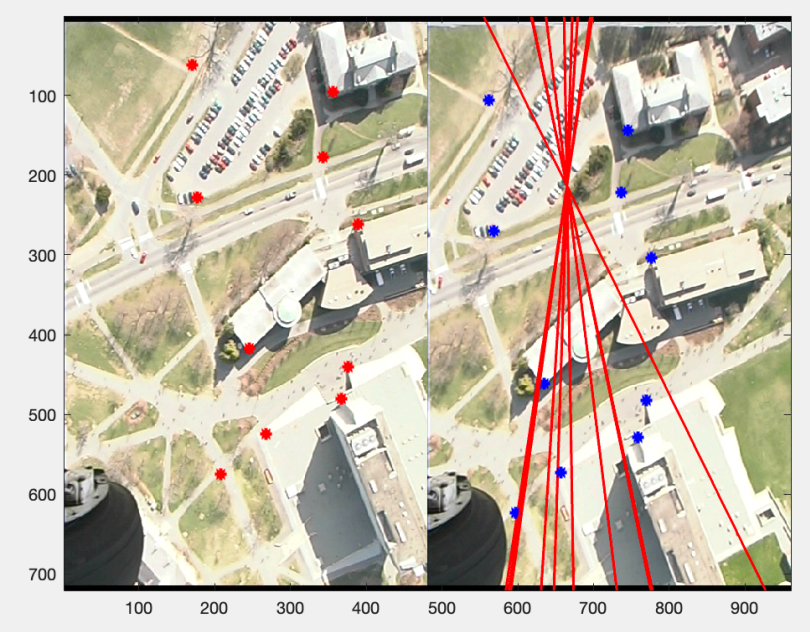
1. Find the SVD of F.
2. The epipole e_l is the column of V corresponding to the null singular value.
3. The epipole e_r is the column of U corresponding to the null singular value.

The estimated location of the epipoles for this system is listed below:

$$e_l = \begin{bmatrix} 0.6096 \\ 0.7927 \\ 0.0020 \end{bmatrix}, e_r = \begin{bmatrix} 0.6546 \\ 0.7559 \\ 0.0035 \end{bmatrix}$$

The Fundamental Matrix is then used to compute the epipolar lines for each reference point on the left image, as seen below:

Figure 2: Stereo Images, p_l and Derived Epipolar Lines



Left Image: Red Point = User Defined Reference Point
Right Image: Red Line = Derived Epipolar Line, Blue Point = True Correspondence Point

The correspondences of the p_l points should exist on the epipolar lines, assuming a perfectly estimated F matrix; however, due to error in estimation, the true location of the corresponding p_r point may exist off the epipolar line. The error for the two test points, measured by computing the Euclidean distance between the estimated epipolar lines and the image points is shown below:

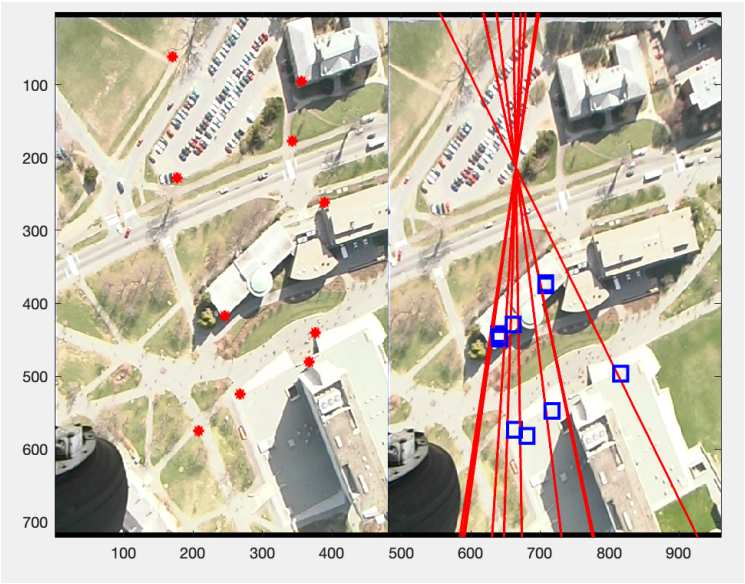
Test Point	Error (Distance btw Estimated Epipolar Line and Image Point in Pixels)
#1	42.244 pixels
#2	72.162 pixels

To automatically find the correspondences on the right image instead of manually selecting them, the procedure would be as follows:

1. Given the fundamental matrix, for each left image point, find its corresponding epipolar line on the right image.
2. Define a window size of 20x20 that will move along the epipolar line to find the correspondence.
3. For each step within the looping in the previous step, compute the cross correlation between the defined window on the left image, centered on p_l, and each window moving along the epipolar line.
4. The location of the max cross-correlation will be defined as the correspondence point.

The results for the automatic location of the correspondence points are seen below:

Figure 3: Automatically Located Correspondence Points, P_r



Left Image: Red Point = User Defined Reference Point
Right Image: Red Line = Epipolar Line, Blue Square = Automatically Located Correspondence

It is observed that the results indicate that it was not able to accurately find the correspondences. This poor performance is due to the errors in estimating an accurate fundamental matrix since we are reliant on this result to derive the epipolar lines where the correspondence is searched for. Since the epipolar lines with the defined search radius (window size) might not contain the correspondence point, the algorithm will find

the closest match based on the cross correlation, which is what the results show in the above figure.

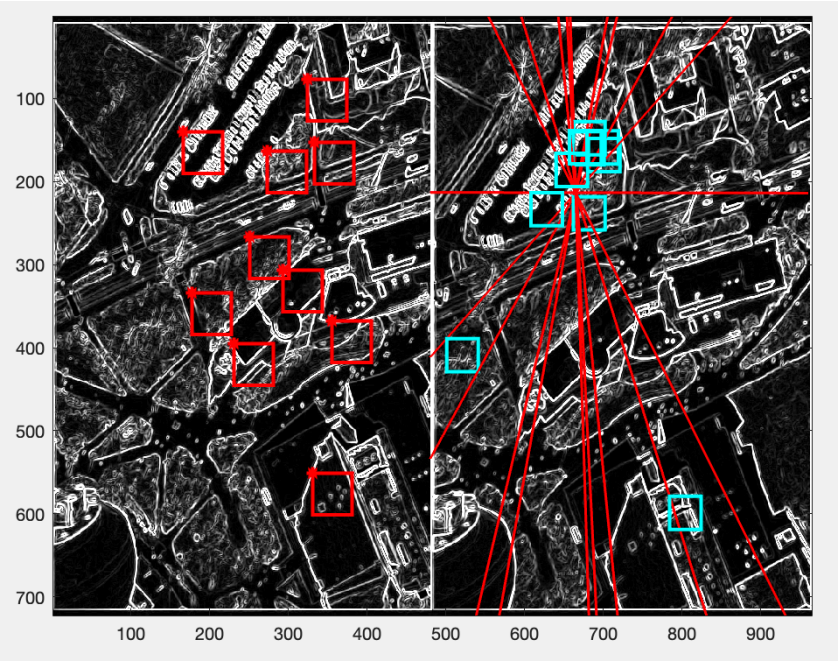
(2) Feature-based matching. – Design a stereo vision system to do "feature-based matching" and explain your algorithm in writing – what the feature is, how effective it is, and what are the problems (5 points). The system should have a user interface that allows a user to select a point on the first 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).

For feature-based matching, edges will be the focused-on feature. Using edges provides the advantages of being less affected by illumination changes or changes in appearance due to the change in viewpoint. However, using edges can present challenges in areas like the occlusion problem, where an edge may be partially occluded due to the difference in viewpoint between the images.

The algorithm for feature-based matching, based on the edges of the structures in the images, is described below:

1. Convert the stereo pair image to grayscale by summing the pixel values in the color channels and dividing by 3.
2. Define the vertical and horizontal Sobel 3x3 kernels and apply the kernel using a convolutional operation on the grayscale images to detect edges.
3. Use the interface to define reference points (p_l) on the left reference image, which we are tasked to find the correspondences (p_r) for in the right image.
4. Define the Fundamental Matrix as the one derived from problem 5.1, which will be used to find the epipolar lines on the right image.
5. Define a window size of 40x40 that will move along the epipolar line to find the correspondence.
6. For each step within the looping in the previous step, compute the cross correlation between the defined window on the left image, centered on p_l, and each window moving along the epipolar line.
7. The location of the max cross-correlation will be defined as the correspondence point.

Figure 4: Feature-Based Matching



Left Image: Red Point/Red Square = User Defined Reference Points
Right Image: Red Line = Derived Epipolar Line, Cyan Square = Automatically Located Correspondences

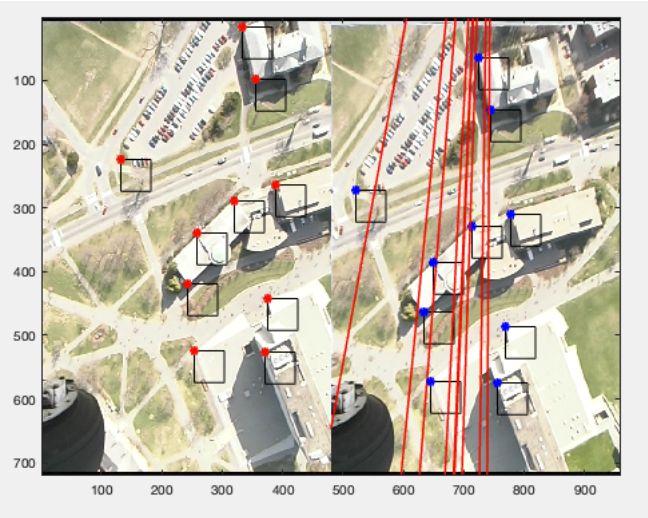
From the results in the feature-based matching algorithm above, it is observed that this algorithm does not perform very well, likely due to errors in the estimation of the fundamental matrix. Additionally, a potential downside to using edges as a feature is that it removes much of the defining characteristics of features/structures in the images like color which may help the cross-correlation similarity metric perform better in finding the best match between the defined reference windows.

(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).

The eight-point algorithm in 5.1 is used to determine the effect of choosing correspondence points with differing properties. In summary, the algorithm does ok when the properties of the reference points are rich in defining characteristics. For instance, the errors for using points with the properties of being a corner, edge, and texture region were less than that of the algorithm on points of smooth regions. This is because having defining characteristics will aid in the process of finding correspondences easier, when using a similarity metric. When comparing the aforementioned results to the case where there were occlusions, the error is significantly higher. This is expected since it is not possible to find the correspondence in the right image if it is occluded. These errors represent the errors that the computer vision system would result in during correspondence matching; however, when a human user marks the corresponding matches on the second image by mouse click, we'd expect a close-to-zero error in all cases, no matter the properties of the points (except for the case of occlusions). This is because humans can accurately determine the correspondences using our "biological" vision system and vision processing, which will outperform a simple computer vision system.

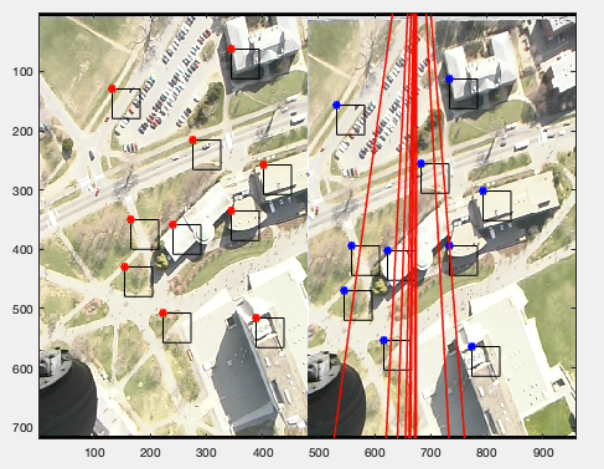
The below figures illustrate the results for the properties of corners, edges, smooth regions, textured regions, and occluded regions:

Figure 5: Eight-Point Algorithm w/ Corners



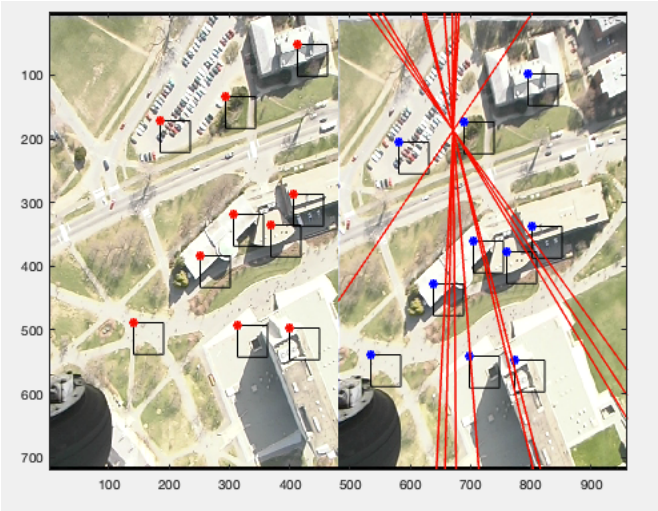
Test Point	Error (Distance btw Estimated Epipolar Line and Image Point in Pixels)
#1	59.965 pixels
#2	181.526 pixels

Figure 6: Eight-Point Algorithm w/ Edges



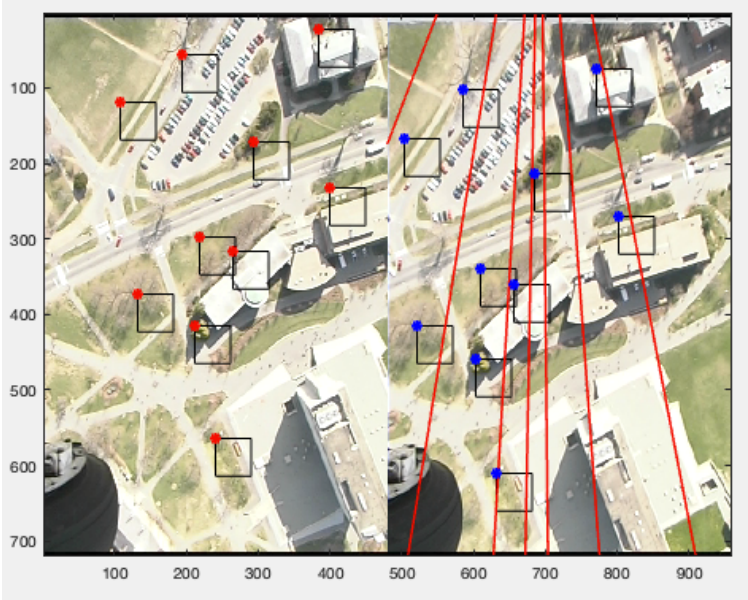
Test Point	Error (Distance btw Estimated Epipolar Line and Image Point in Pixels)
#1	119.102 pixels
#2	115.277 pixels

Figure 7: Eight-Point Algorithm w/ Smooth Regions



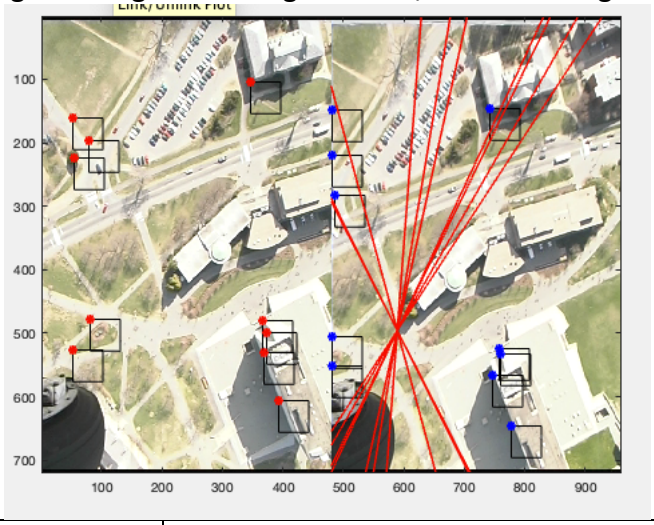
Test Point	Error (Distance btw Estimated Epipolar Line and Image Point in Pixels)
#1	129.490 pixels
#2	118.270 pixels

Figure 8: Eight-Point Algorithm w/ Textured Regions



Test Point	Error (Distance btw Estimated Epipolar Line and Image Point in Pixels)
#1	22.8275 pixels
#2	100.496 pixels

Figure 9: Eight-Point Algorithm w/ Occluded Regions



Test Point	Error (Distance btw Estimated Epipolar Line and Image Point in Pixels)
#1	164.061 pixels
#2	184.108 pixels