COMPUTER VISION ASSIGNMENT 4

JUSTIN DE WITT* (21663904)

22 September 2022

1 ESTIMATING A CAMERA MATRIX PAIR

1.1 Simple Distance Based SIFT Match Filtration (Procedure Justification)

We are provided a set of SIFT feature match pairs, however there are unfortunately "pairs" which do not correspond to true feature matches. Some of the incorrect matches are obvious, others are a bit more elusive. When removing incorrect matches, there are different techniques available. We can use a RANSAC based method (discussed in the previous assignment) or distance based methods. Although effective, RANSAC based techniques are slow; they require many iterations before a suitable sample is randomly chosen. Consider the combination formula:

$$ {}_{n}C_{r} = \frac{n!}{r!(n-r)!} $$

In order to obtain a decent inlier set, we need all r randomly selected points to be "good matches". If there are relatively few *true* matches compared to total matches, it will take many iterations to randomly choose r true matches from the set. As an optimization, we can remove obviously incorrect matches using distance-based techniques. Doing so will increase the ratio between n and r, and increase the likelihood of randomly choosing r good matches, reducing the total iterations needed to obtain a decent inlier set. However, we do not want to decrease the quantity of good matches in the set, this is counterproductive for two reasons:

- We want as many true matches as possible, this increases the accuracy of the Fundamental matrix estimation, which is estimated using a set of point correspondences. The more correct matches we have in the set, the more accurate the matrix estimation.
- If the quantity of good matches is reduced, we reduce the size of the set that r must be an element of. All the sampled r points (during RANSAC) should be elements of the "good match set". It is intuitively counterproductive if the "good match set" length is reduced.

To do the distance based filtering, the entire set of matches is iterated. During the iteration, the Euclidean distance between each match pair is determined. The largest and smallest Euclidean distance (in the set) is recorded. The set is thereafter traversed again, this time a pair is removed from the set if its length exceeds

$$ \mathrm{MinDist} + C(\mathrm{MaxDist} - \mathrm{MinDist}) $$

C ($\in [0, 1]$) is a tunable constant, small values C increase the distance-based filtration intensity. I used this method because it is robust to the image dimension, and the magnitude of camera movement, as opposed to using a constant value threshold only. $C = 1$ implies no filtration.

1.2 Graphic Display Of The SIFT Match Set

The following images display the set of SIFT matches throughout the filtration process. The location of a feature with respect to the local image is signified by a colored circle. A line protrudes from the shaded circle towards the location of the same feature in the foreign image. For example, consider a feature pair at location $[x, y]$ in image 1, and $[x', y']$ in image 2. The shaded circle is placed at $[x, y]$ in image 1, and at $[x', y']$ in image 2. The line is drawn from the circle towards $[x', y']$ in image 1, and towards $[x, y]$ in image 2. To make the distinguishment more blatant, blue is used when referring to image 1, and yellow is used when referring to image 2. The Images are shown below.

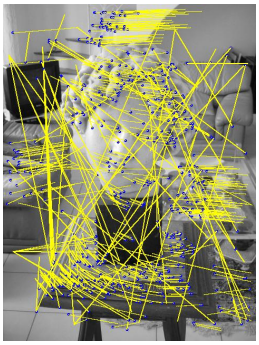


Figure 1: All Matches

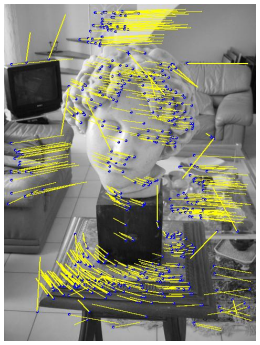


Figure 2: Distance Filtering

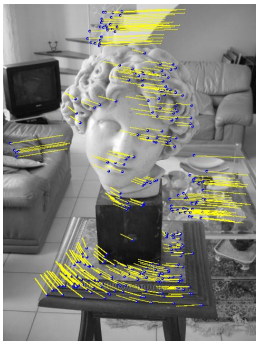


Figure 3: RANSAC Filtering

Figure 4: Feature Filtration Pipeline Performed on Image 1.

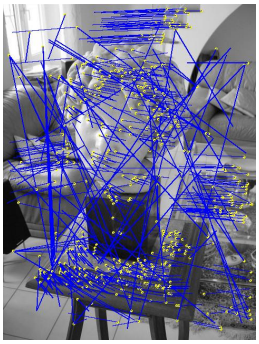


Figure 5: All Matches

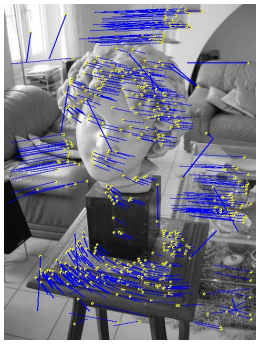


Figure 6: Distance Filtering

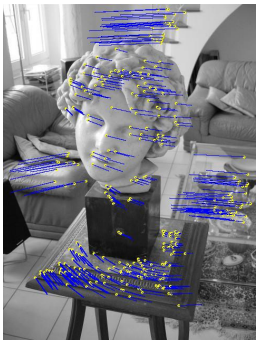


Figure 7: RANSAC Filtering

Figure 8: Feature Filtration Pipeline Performed on Image 2.

1.3 Re-Estimate the Fundamental Matrix

During RANSAC, I used a Sampson-Distance threshold of 0.1, and ran the procedure for 3,000 iterations. I arrived at these constants via a trial-and-error process, where I tested different thresholds and iteration lengths. After RANSAC the obtained inlier set is (hopefully) composed of "true" feature matches. We can use this set of inliers to obtain an accurate estimate of the Fundamental matrix F. The obtained fundamental matrix was:

$$F = \begin{bmatrix} -1.302e^{-8} & -1.911e^{-5} & -1.141e^{-3} \\ -6.814e^{-6} & 9.773e^{-7} & 1.241e^{-1} \\ -5.194e^{-4} & -1.175e^{-1} & 9.853e^{-1} \end{bmatrix}$$

There were 622 SIFT matches in the original set, which we filtered down to 549 matches using the distance based method discussed. This set was further reduced

down to 337 "true" matches using a RANSAC-based method. The entire inlier set of length 337 is used in the computation of the F matrix above. Recall from the previous assignment, the Fundamental matrix can be used to relate corresponding points in stereo images. We can use it to describe the geometric relationship between corresponding points, however the relationship is defined in terms of pixel coordinates. We use this matrix, and the intrinsic properties of each camera to obtain an *Essential Matrix*, which is defined in terms of *normalized image coordinates*. Normalized image coordinates have the origin at the optical center of the image. The essential matrix also has less degrees of freedom (5 compared to 7). We can use the Essential matrix to get the 3-dimensional position of a feature-pair, which makes determining the Essential matrix a worthwhile venture.

1.4 Obtaining the Essential Matrix

Now that an accurate estimation of the Fundamental matrix has been obtained, we will use the provided calibration matrices to determine the fundamental matrix using the following relationship:

$$E = K'^{T} F K$$

Once we have obtained the Essential matrix, we can obtain useful rotation matrices based on the spectral value decomposition (SVD) of the matrix. Theoretically the first two spectral values should be equal, and the third should be zero. A spectral value of zero corresponds to an eigenvalue of zero, which implies that the transformation leads to a drop in dimension. That is, the dimension of the input space is higher than that of the output space. I hypothesize that this dimension drop is associated with a 3D world-point mapping to a 2D image point. I also hypothesize that the equivalence of the first two eigenvectors quantifies the desire to maintain spatial distance between points in both spaces. Far away points in 3D space, should remain equally far away in 2D image space. In my implementation, the obtained spectral values are:

$$[6.820e^{-05}, 6.783e^{-05}, 1.397e^{-21}]$$

As mentioned, we will use the obtained U and V matrices (obtained by the SVD) to calculate the projection matrices of both cameras. Unfortunately, SVD may yield matrices which have a determinant of -1. A determinant of magnitude 1 implies that spacial size is maintained when applying the transformation, however a negative determinant implies that the transformation applies a translation to the output space, which is unwanted in this context. We can fix this by multiplying the offending matrix by -1 if necessary. We need not worry if both determines are -1, because we use a product of U and V. The following determinant properties are sufficient justification for this action:

$$|kA| = k^{n}|A|$$
$$|AB| = |A||B|$$

Our U and V matrices are 3x3, so applying the first property (k = -1) will flip the sign of our determinant. The corrected versions of these matrices are used to determine the camera matrices from the essential matrix.

1.5 Determining The Camera Matrices From the Essential Matrix

The projection matrix associated with camera 1 is obtained as:

$$P = K[I \mid \mathbf{0}]$$

However, the matrix associated with the second camera is not so trivially defined. There are four possible choices of the second camera's projection matrix. Each choice is associated with a potential geometric camera positioning. We are interested in the positional configuration where point correspondences are triangulated to 3D points *in front* of both cameras. To determine which projection matrix is associated with this configuration, we conduct the following test:

$$\mathbf{z}_1^T(\mathbf{x}-\mathbf{c}_1) > 0 \quad \text{and} \quad \mathbf{z}_2^T(\mathbf{x}-\mathbf{c}_2) > 0$$

where $\mathbf{z}_i$ is the vector associated with camera i's principal axis, $\mathbf{x}$ is the real-world location of a triangulated point, and $\mathbf{c}_i$ is the real-world position of camera i. Intuitively this test works. The subtraction of the triangulated point and the camera location yields the vector connecting the camera to the triangulated point. Furthermore, recall a positive dot product implies that the two vectors point in approximately the same direction, whereas a negative dot product implies they point in approximately opposite directions. Therefore, if both the above dot products are positive, both camera's principle axes points in approximately the same direction as the vector connecting them to the triangulated point. For further insight into the intuition that this test yields the desired configuration, consider the 3D plots displayed later in the report.

To perform this test, we need to first devise a method of triangulating a true SIFT match, into a 3D world position. Fortunately, we can obtain a least squares solution using SVD, by solving the following 4x4 system:

$$\begin{bmatrix} y\mathbf{p}_3^T - \mathbf{p}_2^T \\ -\mathbf{p}_1^T - x\mathbf{p}_3^T \\ y'\mathbf{p}_3'^T - \mathbf{p}_2'^T \\ -\mathbf{p}_1'^T - x'\mathbf{p}_3'^T \end{bmatrix} \begin{bmatrix} W \\ X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

We can dehomogenize the obtained vector, which now represents a the 3D Euclidean coordinate position of the true SIFT match, given in the form $[x, y, x', y']$. Now that a triangulation method has been obtained, we can simply perform the test, and determine the projection matrices of each camera. The projection matrices of camera 1 (P) and camera 2 (P′) were determined to be:

$$P = \begin{bmatrix} 6.345e^{-1} & -4.775e^{-7} & 2.116e^{-1} & 0.000e^{0} \\ 0.000e^{0} & 6.345e^{-1} & 2.822e^{-1} & 0.000e^{0} \\ 0.000e^{0} & 0.000e^{0} & 8.818e^{-4} & 0.000e^{0} \end{bmatrix}$$

$$P' = \begin{bmatrix} 5.870e^{-1} & -4.759e^{-2} & 3.023e^{-1} & -5.995e^{-1} \\ 9.015e^{-3} & 6.297e^{-1} & 2.751e^{-1} & -6.284e^{-3} \\ -1.318e^{-4} & 1.202e^{-5} & 8.626e^{-4} & 9.747e^{-5} \end{bmatrix}$$

1.6 Plotting The Triangulated SIFT Matches

Now that we have a triangulation mechanism, we can triangulate all the SIFT pairs, and obtain a real world position for each of them. These positions can then be plotted by means of a 3D scatter-plot. On top of simply plotting the real world position of each feature match, we can also plot the cameras, and their orientations. This yields a wonderful visualization of our stereo camera setup.

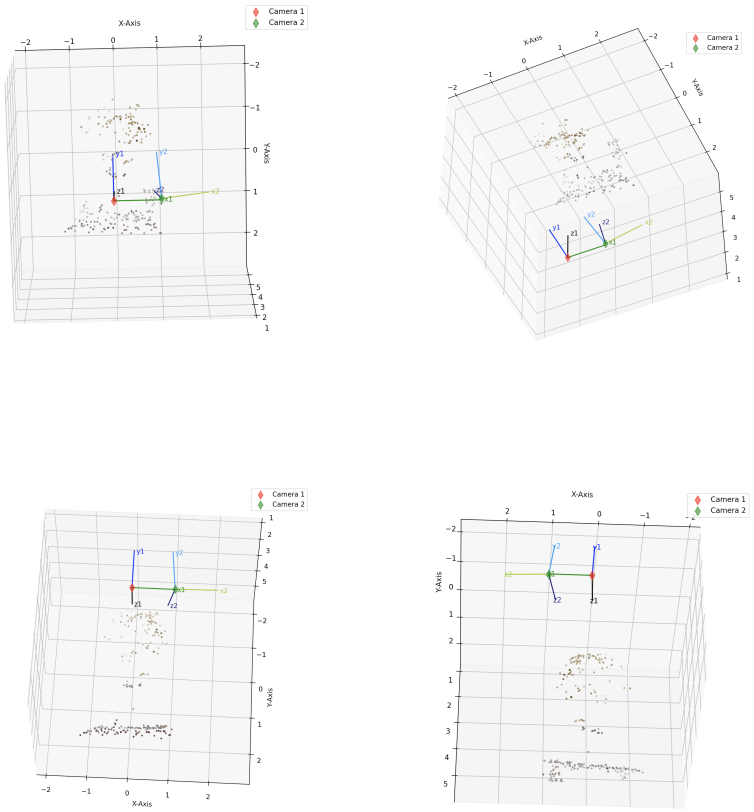


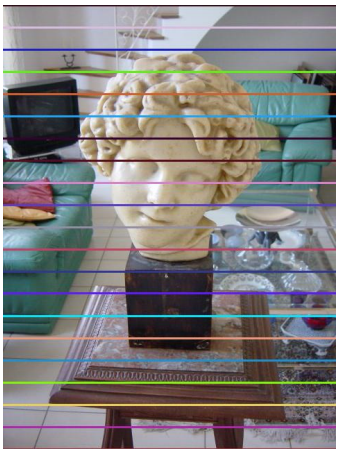
Figure 9: Visit This Link to see more.

Each triangulated point can be traced back to its location in the image by using the obtained projection matrices. Recall that a projection matrix allows for the mapping between real-world position, and image position. Otherwise, we can simply use the SIFT feature pair to obtain the location of the real-world position and its image location. We color each scattered point the color of the corresponding pixel in the image. We have removed some background points which appear far away from the 3D modeled bust. I hypothesize that these points end up being far away due to the relatively short spacial distance between the SIFT match pair, corresponding to these background points. To clarify, when the camera moves from position 1 to position 2, points near the camera appear to move far in the subsequent images, however background points do not move far. The concept is commonly referred to as *parallax*. Because we are only estimating our Fundamental matrix from inliers, we cannot guarantee infinite accuracy in the Essential matrix and projection matrix computation. This means our triangulation is not perfectly accurate. Inaccuracies are amplified when dealing with sensitive computation, such as the computation of background point triangulation. A small change in image location may correspond to a large change in triangulated distance. The noise in our estimation now poses a non-negligible influence.

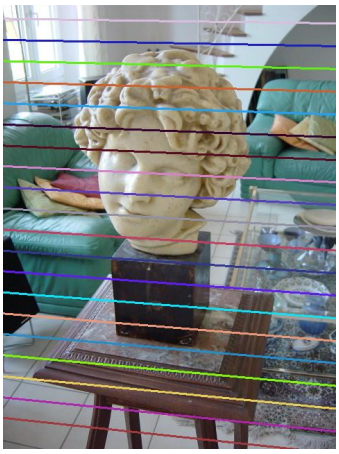
2 RECTIFYING AN IMAGE PAIR

Suppose we want to increase the density of our 3D model. We need to devise a method to quickly identify additional feature matches which were not spotted by the SIFT algorithm. From the previous assignment we learned that features in one image, can be mapped onto a line in the other image. If we wish to identify features between images, we need only search the corresponding line in the other

image. Unfortunately, traversing a line is somewhat challenging if the line has a non-zero slope. Consider the following images:



Epipolar Lines on Image 1



Epipolar Lines on Image 2

Notice that both purple lines contain the nose of the bust, but searching the epipolar lines of image 2 is a non-trivial task. We can use a homography to transform the images such that both their epipolar lines are horizontal, and vertically aligned. Thereafter, we can search for matching features by simply iterating the rows (consider including an error radius) of each image. Now that context has been provided, we will determine the homography that should be applied to each image, in order to vertically align the epipolar lines, as well as ensure they are horizontal in both images.

2.1 The Rectifying Homographies

Given our camera matrices P_1 and P_2, we wish to adjust the calibration and rotation matrices of each camera, while maintaining their real-world positions. The procedure involves decomposing each camera matrix into its calibration, rotation, and position components. A new calibration and rotation matrix is defined in terms of each camera's intrinsic parameters. The shared rotation matrix is composed from:

- $\mathbf{r}_1$: The normalized vector connecting the position of left camera, to the right camera.
- $\mathbf{r}_2$: The normalized vector orthogonal to $\mathbf{r}_1$ and principle axis of the left camera.
- $\mathbf{r}_3$: The unit vector orthogonal to both $\mathbf{r}_1$ and $\mathbf{r}_2$.

Notice the matrix obtained by stacking these row vectors is orthonormal, like all rotation matrices are. The shared calibration matrix is simply the average of each cameras calibration matrix. Define K_n as the shared calibration matrix, and R_n the shared rotation matrix, our homographies are given by:

$$T_1 = K_n R_n R_1^T K_1^T \quad \text{and} \quad T_2 = K_n R_n R_2^T K_2^T$$

where R_i is the rotation matrix of camera i, and K_i, its calibration matrix. We can apply these homographies to their respective images using techniques of previous assignments. The results are shown in the following images:

Rectified Image 1

Rectified Image 2

Note: The origin of the rectified images (top left corner) no longer coincides with the origin of the original image. This is because the edges of the original content are no longer perpendicular, but have been transformed during the homography. To fit this obscurely shaped image into another image, a bounding box is computed using the minimum x-value, and the minimum y-value of rectified image corners.

2.2 Displaying the Rectified Epipolar Lines

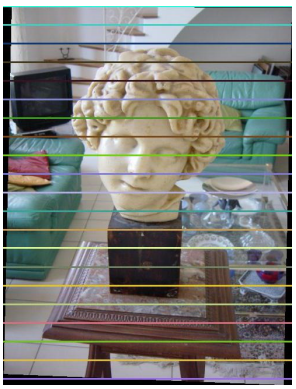
Now that the images have been rectified, we can determine the projection matrix of each (new) camera. Using this projection matrix, we can use the methods of the previous assignment to determine the corresponding Fundamental matrix, from which epipolar lines can be drawn. The new camera matrices P_1' and P_2' are given by:

$$P_1' = K_n R_n [I \mid C_1] \quad \text{and} \quad P_2' = K_n R_n [I \mid C_2]$$

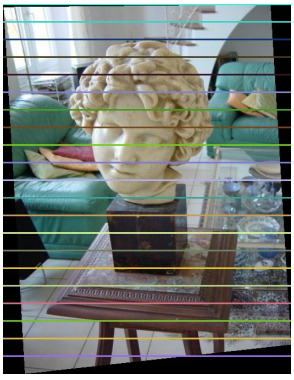
where C_1 and C_2 are the Euclidean-locations of camera 1 and camera 2 respectively. From the new camera matrices, the fundamental matrix used to draw the epipolar lines on the rectified images was determined to be:

$$F = \begin{bmatrix} 5.001e^{-37} & 4.833e^{-21} & -2.018e^{-19} \\ -8.805e^{-20} & -6.957e^{-19} & 6.312e^{-1} \\ -6.164e^{-17} & -6.312e^{-1} & -5.326e^{-14} \end{bmatrix}$$

We can use the same methods discussed in the previous assignment to draw the epipolar lines on the rectified images. The results of this procedure are shown in the following images:



Epipolar Lines on Image 1



Epipolar Lines on Image 2

Notice now that the epipolar lines are horizontal. We can now theoretically search each epipolar line for corresponding features, by simply iterating over the column

values of each respective row. After we identify more feature pairs, we could once more triangulate their real-world positions, and add them to our 3D model of the image space. **Note:** due to slight inaccuracy in the estimation of our fundamental matrix, we should not only traverse the line, but a small radius surrounding the line.

3 VERIFYING THESE SOLUTIONS ON ANOTHER DATASET

We will now repeat the entire procedure, but on another dataset. We will repeat the SIFT feature filtration process, again using a combination of distance based filtering, as well as RANSAC. No changes should be made to the distance based filtering procedure, as I have implemented the process in a way that is resistant to changes in image scale (see section 1.1). The results of the SIFT feature filtration is shown in the following images:

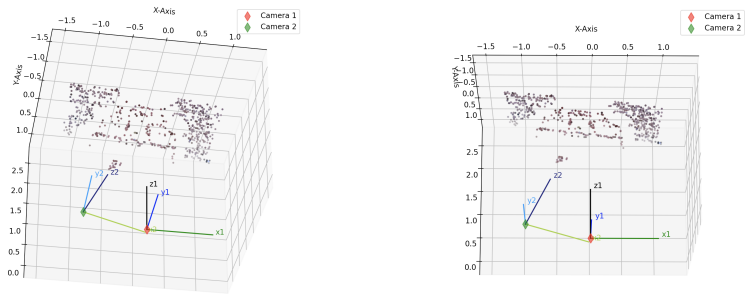
All Matches Distance Filtering RANSAC Filtering

Figure 10: Feature Filtration Pipeline Performed on Image 1.

All Matches Distance Filtering RANSAC

Figure 11: Feature Filtration Pipeline Performed on Image 2.

The subsequent fundamental, essential, and camera matrices were all obtained using identical procedures. The implementation did not need to be changed in any way. The same test was used to determine the projection matrix of camera 2, and the same triangulation process was performed. When plotting the 3D model, the distance threshold for removing the background points was updated. This is a result of a change in distance between the cameras, and the object on which the SIFT matches were detected. I had to simply determine a suitable threshold once more. The obtained 3D plot is shown below:



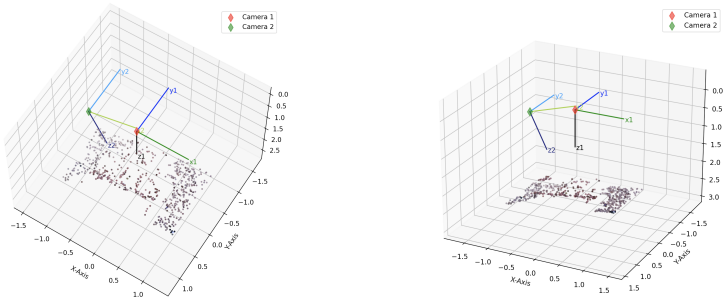


Figure 12: Visit This Link to see more.

The image rectification process was very similar, although slightly different. Notice that the perspective change associated with the cameras of this dataset, is different to that of the previous. In the first dataset, camera 2 moved rightwards, in relation to camera 1. In this dataset, camera 2 moved leftwards. If we ignore this adjustment, our rectified image appears upside down. This is likely a result of the cross products used to determine the new rotation matrix R_n yielding the orthogonal vector in the opposite direction. However, if we follow the procedure mentioned in section 2.1, we can ensure the rectification goes smoothly. The epipolar lines on both the original, and rectified images, are shown below. No adjustments other than those mentioned were needed to obtain these results.

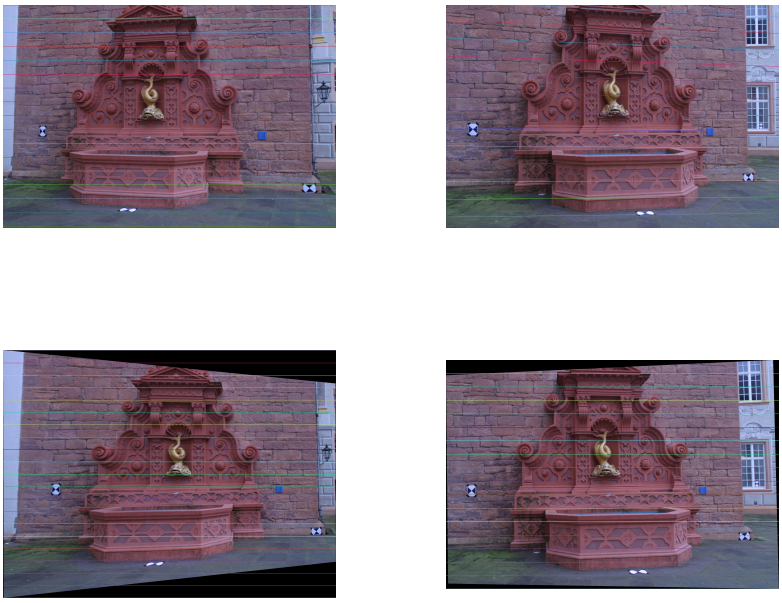


Figure 13: A Side-by-Side Comparison of the Epipolar Lines, and the Rectified Epipolar Lines