Computer Vision: Lab6

Mamoru Ota   Ana Luísa Campos e Sampaio Melo

1 Part 1 - 8-Point Algorithm Function

The objective of this assignment is to implement the 8-Point Algorithm to estimate the fundamental matrix F from a set of corresponding points between two images. The fundamental matrix is a matrix 3×3 matrix that describes the geometric relationship between two images of a stereo pair.

The 8-point algorithm uses at least eight point correspondences between two images to estimate F.

1.1 8-Point Algorithm Function (version 1)

In this first version of the 8-point algorithm, we aim to estimate the fundamental matrix F by solving a system of linear equations constructed from point correspondences between two images. Each correspondence consists of a pair of points (x_1, y_1) in the first image and (x_2, y_2) in the second image. These correspondences are used to form the matrix A, where each row is defined as:

$$A_i = [x_1x_2, x_1y_2, x_1, y_1x_2, y_1y_2, y_1, x_2, y_2, 1]$$

These results in a matrix A of size $N\times9$, where N is the number of point correspondences. In this case, the algorithm requires at least 8 correspondences, so $N \geq 8$.

After constructing the matrix A, the next step is to compute the **Singular Value Decomposition (SVD)** of A. The SVD decomposition is performed as follows:

$$A = UDV^T$$

Here, U and V are orthogonal matrices, and D is a diagonal matrix containing the singular values of A. The solution f for the fundamental matrix F is derived from the last column of the V matrix.

After, the matrix F is reshaped from the vector f obtained from the SVD into a 3×3 matrix, and its rank is forced to be 2. To do this, we set $D(3,3) = 0$, where D is the diagonal matrix from the SVD of F:

$$F = UDV^T$$

By forcing the rank of F to be 2, we obtain the best possible approximation to the fundamental matrix given the input point correspondences.

1.2 8-Point Algorithm Function (version 2)

In the second version of the 8-point algorithm, normalization of the input points is introduced to improve numerical stability. The normalization process ensures that the coordinates of the input points are centered around the origin and scaled such that the average distance from the origin is $\sqrt{2}$. This step mitigates potential numerical issues when dealing with large or unnormalized coordinates. The algorithm provides a robust and reliable way to estimate the fundamental matrix, especially for datasets with varying scales or large coordinate values.

The algorithm is implemented as follows:

1. **Normalization of Points**: Both sets of points, $P1$ (from the first image) and $P2$ (from the second image), are normalized using the function `normalise2dpts`. This function ensures that:

   - The points are shifted so that their centroid is at the origin.
   - The points are scaled such that the average distance from the origin is $\sqrt{2}$.

   The normalization function also computes similarity transformations $T1$ and $T2$ for $P1$ and $P2$, respectively. The steps of normalization are:

   (a) Compute the centroid of the finite points.
   (b) Shift the points so that the centroid is at the origin.
   (c) Scale the points such that the average distance from the origin is $\sqrt{2}$.
   (d) Construct a similarity transformation matrix T that represents the scaling and translation.

   Using this function:

   $$[nP1, T1] = \text{normalise2dpts}(P1)$$
   $$[nP2, T2] = \text{normalise2dpts}(P2)$$

2. **Apply Version 1 of the Algorithm**: The normalized points $nP1$ and $nP2$ are used as input to the `EightPointsAlgorithm` function (version 1). This results in a normalized fundamental matrix F_n:

   $$F_n = \text{EightPointsAlgorithm}(nP1, nP2)$$

3. **De-normalization of the Fundamental Matrix**: The final fundamental matrix F is obtained by transforming F_n back to the original coordinate system using the inverse of the normalization transformations $T1$ and $T2$:

   $$F = T2^T \cdot F_n \cdot T1$$

1.3 Implementation and Results

1.3.1 Estimated Fundamental Matrices and Rank

The fundamental matrices - F_1 for the unnormalized algorithm and F_2 for the normalized algorithm - are as follow:

$$F_1 = \begin{bmatrix} -0.0000 & -0.0000 & 0.0050 \\ 0.0000 & 0.0000 & -0.0019 \\ -0.0034 & 0.0023 & -1.0000 \end{bmatrix} \quad F_2 = \begin{bmatrix} -0.0000 & -0.0000 & 0.0003 \\ 0.0000 & 0.0000 & -0.0042 \\ -0.0003 & 0.0042 & 0.0506 \end{bmatrix}$$

The ranks of F_1 and F_2 were confirmed to be 2, which is a necessary condition for valid fundamental matrices.

Point	Epipolar Constraint (F_1)	Epipolar Constraint (F_2)
1	-1.466148	-0.004987
2	-1.243624	0.002202
3	-1.251188	0.001504
4	-1.433921	-0.007141
5	-0.970212	0.008874

Table 1: Comparison of Epipolar Constraint Values for Selected Points.

1.3.2 Epipolar Constraint and Errors

The epipolar constraint $x_2^T F x_1 = 0$ was evaluated for all point correspondences. Ideally, these values should be close to zero. Table 1 shows some of the results for the unnormalized (F_1) and normalized (F_2) algorithms.

The mean epipolar error across all points was also calculated:

$$\text{Mean Epipolar Error (Unnormalized): } 1.2255$$

$$\text{Mean Epipolar Error (Normalized): } 0.0081$$

The results clearly show that the normalized algorithm significantly reduces the epipolar constraint error, demonstrating its effectiveness.

1.4 Visualization of Epipolar Lines

The fundamental matrices were used to generate epipolar lines for both images. Figures 1 and 2 illustrate the epipolar lines for F_1 and F_2, respectively.

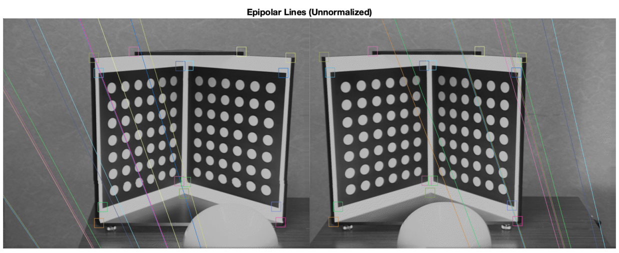

Figure 1: Epipolar lines using the unnormalized fundamental matrix F_1

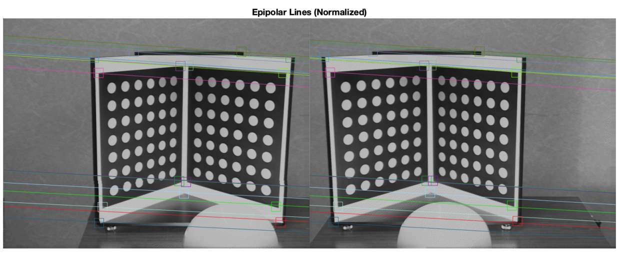

Figure 2: Epipolar lines using the normalized fundamental matrix F_2

The normalized version of the algorithm significantly outperforms the unnormalized version - it reduces numerical instability and achieves a lower epipolar error. The epipolar lines derived from F_2 align better with the corresponding points, as shown in the visualizations. These results confirm the importance of the normalization in achieving robust and accurate results with 8-point algorithm.

2 Part 2 - Acquire and match your own images

In Part 2, the goal is to process a pair of stereo images to estimate the fundamental matrix F. The process includes image matching, RANSAC-based fundamental matrix estimation, and the evaluation of the results.

2.1 Procedure

The following steps were implemented:

1. **Acquire and load a pair of stereo images:** Two images (`Books1.jpg` and `Books2.jpg`) were taken and loaded. These images were captured under appropriate conditions, ensuring overlapping regions with sufficient feature density for robust matching.

2. **Feature matching:** The `matchFeaturesBetweenImages` function was used to find correspondences between the two images. The steps performed by the function are as follows:

   - Convert both images to grayscale, simplifying feature detection and ensuring consistent processing.
   - Detect keypoints in each image using the SURF (Speeded-Up Robust Features) algorithm. Keypoints correspond to distinctive features in the image, such as corners or blobs.
   - Extract feature descriptors from the detected keypoints. These descriptors encode information about the local image structure around each keypoint.
   - Match the extracted descriptors between the two images to find corresponding points. This step uses a nearest-neighbor matching algorithm to identify pairs of descriptors that are most similar.
   - Convert the matched points into homogeneous coordinates, yielding two sets of matched points, $P1$ and $P2$, for further processing.

3. **Estimate the fundamental matrix using RANSAC:** The fundamental matrix F was estimated using the `ransacF` function. This function performs the following steps:

   - Iteratively selects subsets of 8 random point correspondences from $P1$ and $P2$ to compute a candidate F using the normalized 8-point algorithm (`EightPointsAlgorithmN`).
   - For each candidate F, computes residuals for all points using the epipolar constraint:
     $$r_i = |x_2^T F x_1|$$
     where $x_1 \in P1$ and $x_2 \in P2$.
   - Counts the number of inliers, defined as correspondences where $r_i <$ threshold (th), and updates F if the number of inliers increases.
   - Repeats the process until the desired confidence ($p = 0.999$) is achieved or the maximum number of iterations is reached.

   This robust estimation step ensures that noisy matches do not affect the accuracy.

4. **Evaluate the results:** The estimated fundamental matrix F was validated using the `evaluateFundamentalMatrix` function, which performs the following:

- **Epipolar constraint verification:** Computes residuals $r_i = x_2^T F x_1$ for all matched points and reports the mean residual. This step ensures that the estimated F satisfies the epipolar constraint.
- **Visualization of epipolar lines:** Visualizes the epipolar lines corresponding to matched points $P1$ and $P2$ on the respective images using the function `visualizeEpipolarLines`.
- **Epipole calculation:** Computes the left and right epipoles by performing Singular Value Decomposition (SVD) of F and identifying the null spaces of F^T and F.

2.2 Evaluation of the Results

To verify the estimated fundamental matrix F, the following tests were performed:

2.2.1 Epipolar Constraint Verification

The epipolar constraint was checked for all corresponding points $P1$ and $P2$. This constraint ensures that:

$$x_2^T F x_1 = 0$$

where $x_1 \in P1$ and $x_2 \in P2$. The residual $r = x_2^T F x_1$ was computed for all points, and their mean value was reported as 0.000442, indicating excellent consistency and accuracy in the estimated F.

2.2.2 Visualization of Epipolar Lines

The `visualizeEpipolarLines` function was used to compute and visualize the epipolar lines for the matched points $P1$ and $P2$ in the stereo images. The function operates in two modes:

1. **Manual mode:** If the input point sets ($P1$ and $P2$) are empty, the user can interactively select points in one of the images. The corresponding epipolar line is computed using the fundamental matrix F and visualized on the other image. The user can quit the selection by clicking outside the image boundaries.
2. **Automatic mode:** If the input point sets are provided, the function uses these points to compute and display the epipolar lines for all correspondences. The following steps are performed:
   - For each point in the first image, the epipolar line in the second image is computed using $L = F[x, y, 1]^T$. Similarly, for each point in the second image, the epipolar line in the first image is computed using $L' = F^T[x, y, 1]^T$.
   - The epipolar lines are plotted over the concatenated image pair for visualization. Each pair of corresponding points is assigned a random color for distinction.

Figure 3 demonstrates the epipolar lines for the `Books` stereo image pair. The alignment of the lines with the corresponding features confirms the correctness of the estimated fundamental matrix F.

2.2.3 Epipole Calculation

The left and right epipoles were calculated as the null spaces of F^T and F, respectively. By performing Singular Value Decomposition (SVD) of F:

$$F = UDV^T$$

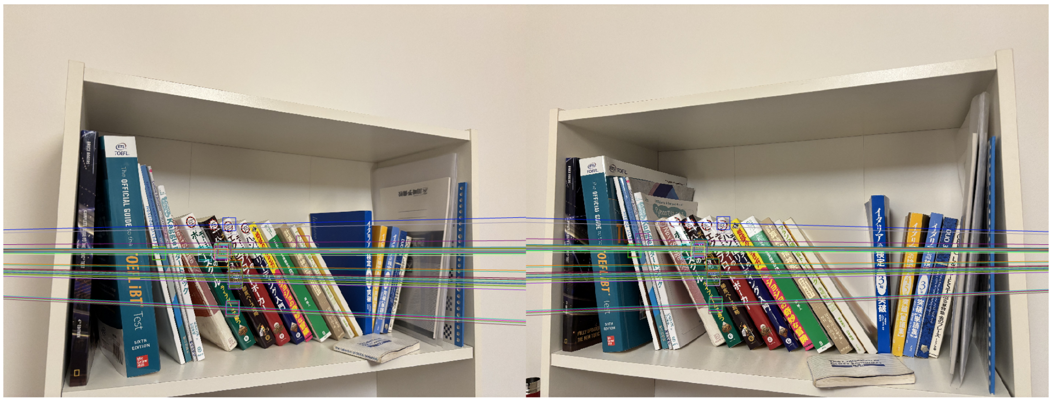

Figure 3: Visualization of epipolar lines for the `Books` images. The colored lines represent the epipolar geometry between matched points in the two images.

the left epipole is given by the last column of V, and the right epipole by the last column of U. The computed epipoles were:

$$\text{Left epipole: } [-10174.159, 2700.020, 1.000]$$

$$\text{Right epipole: } [15044.429, 2720.302, 1.000]$$

2.3 Conclusion

Through this experiment, the fundamental matrix F was robustly estimated using RANSAC, and its accuracy was confirmed through epipolar constraint validation, visualization of epipolar lines, and epipole computation. The `evaluateFundamentalMatrix` function played a critical role in this process by automating the validation and visualization steps. The extremely small residual and precise alignment of epipolar lines with matched features demonstrate the effectiveness of the implemented approach in estimating F for stereo image pairs.