Lucas Versini
lucas.versini01@gmail.com
lucas.versini@polytechnique.edu

Practical session n°2
October 9, 2024

3D Computer Vision

Master MVA

Practical session n°2

Remark: Sections 1 and 2 are only here to describe my understanding of the maths and of the code. They are useful to me, but may not be relevant for you to read.
Some results are presented in 3.

1 Objective

We are provided with two images of the same scene (that is, a stereo image pair). The goal is to display the corresponding epipolar line on the second image when the user clicks on a pixel on the first image.

2 Technical explanations

2.1 Mathematical foundations

In stereo vision, given a point $\mathbf{p}$ in the first image, it is not possible to find the exact corresponding point $\mathbf{p}'$ in the second image. However, it is possible to compute a line on which $\mathbf{p}'$ belongs, called the epipolar line associated with $\mathbf{p}$.
More precisely, this line is determined by the *fundamental matrix* F, which is a function of the calibration matrices of the two cameras, as well as the rotation matrix and translation vector between the cameras.
Then, for a point $\mathbf{p}$ in the left image, the corresponding right epipolar line is given by

$$\ell_R = F^T \mathbf{p},$$

and for $\mathbf{p}'$ in the right image, the corresponding left epipolar line is given by

$$\ell_L = F\mathbf{p}'.$$

The question is to know how to compute F for a given pair of images.

The idea is to use pairs of corresponding points $(\mathbf{p}_i, \mathbf{p}'_i)$, which satisfy the epipolar constraint

$$\mathbf{p}_i^T F \mathbf{p}'_i = 0.$$

If $\mathbf{p}_i = (x_i, y_i, 1)^T$ and $\mathbf{p}'_i = (x'_i, y'_i, 1)^T$, then this constraint can be rewritten as

$$A_i^T f = 0,$$

where $A_i = (x_i x'_i, x_i y'_i, x_i, y_i x'_i, y_i y'_i, y_i, x'_i, y'_i, 1)$ and $f = (F_{11}, F_{12}, F_{13}, F_{21}, F_{22}, F_{23}, F_{31}, F_{32}, F_{33})$.

F has 9 coefficients, but is determined up to scale, so 8 parameters are needed. Since each pair $(\mathbf{p}_i, \mathbf{p}'_i)$ gives us one equation, 8 pairs (in general position) are enough to compute F.

So we construct the matrix A with rows A_i described above, and want to solve $Af = 0$. In practice, we will solve $\min_{\|f\|=1} \|Af\|$ using the Singular Value Decomposition (SVD) of A. Note that if we have 8 pairs $(\mathbf{p}_i, \mathbf{p}'_i)$, we may add a row full of zeros to A to make it square.

However, we know that F is rank 2, while the computed F may not be. So once F is computed, we compute its SVD and replace the smallest singular value by 0, resulting in a matrix F that has indeed rank 2.

2.2 Code explanation

- **Keypoints matching:** First, Scale-Invariant Feature Transform (SIFT) is applied to both images to find keypoints. Each keypoint is associated with a feature vector of fixed dimension. By comparing the feature vectors, matches can be established: for two keypoints k_1 and k_2, the match (k_1, k_2) is accepted if the feature vectors associated to k_1 and k_2 are close enough.

- **RANSAC algorithm to refine matches:** However, these first matches are not precise enough. We then use the RANdom SAmple Consensus (RANSAC) algorithm. The main idea is the following:

  - Randomly select 8 pairs of matches;
  - Compute the matrix F from these matches as explained in 2.1;
  - For each match $(\mathbf{p}_i, \mathbf{p}'_i)$, check if the the epipolar constraint is verified (up to some error);
  - Count the number of *inliers*, and keep F and the inliers in memory if this is the highest number of inliers so far;
  - Repeat (the number of iterations being iteratively recomputed, as explained in the lecture).

  Note that the 8-point algorithm described previously is imprecise, because the coefficients of F have very different scales. So in practice, we scale the coefficient before applying the algorithm, and reverse the scaling afterwards.

- **Recomputing F:** Then, we recompute F based only on the inliers corresponding to the "best" matrix F found previously.

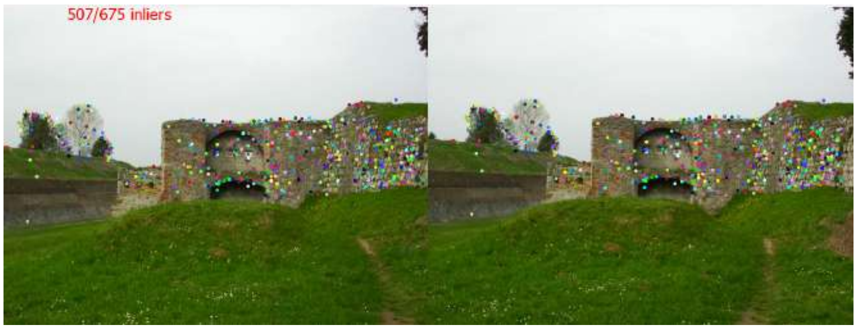
- **Displaying epipolar lines:** Finally, when the user clicks on a pixel in one image, we can easily display the corresponding epipolar line in the other image.

3 Results

We show here some results.

For the images provided with the practical session, the result can be seen in Figure 1.
For other images (from the first practical session), the result can be seen in Figure 2.

These results are quite good: RANSAC finds more than 500 (sometimes 507, sometimes 570...) inliers in the first case (more than 400 in the second), and when clicking on a point on one image, the corresponding epipolar line in the other image seems to go through the corresponding point.



(a) Result of the RANSAC algorithm

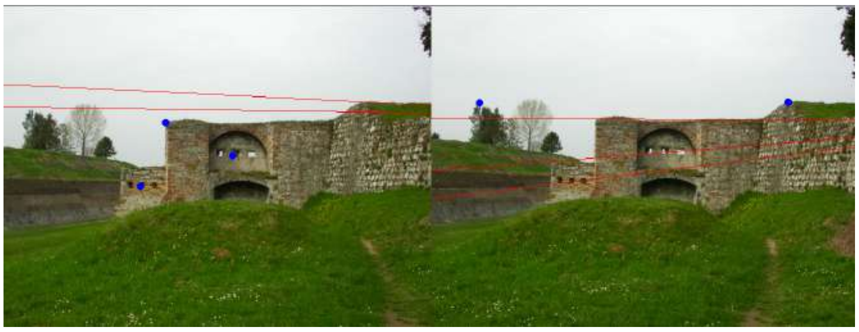
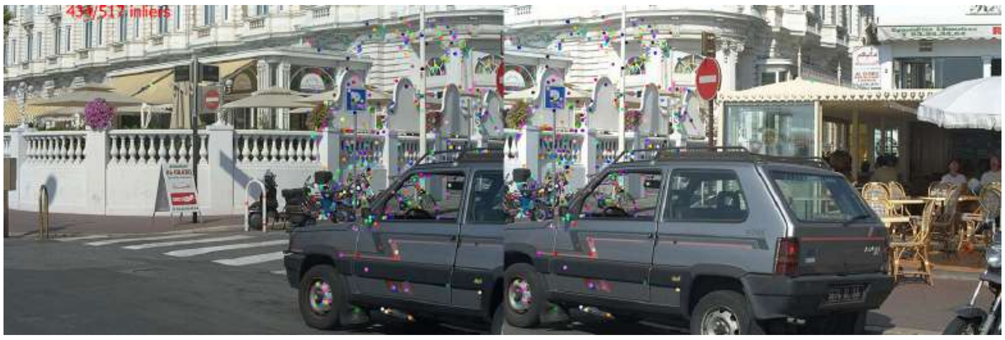
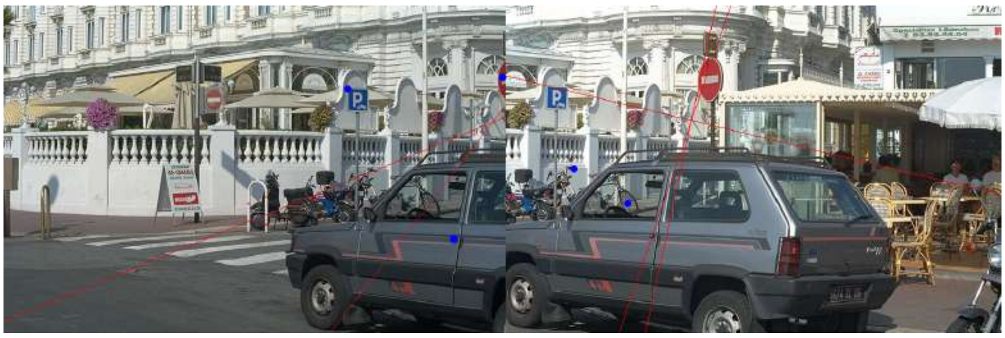

(b) Epipolar lines

Figure 1: Results for a first pair of images

However, the results may vary from one execution to another. Indeed, the RANSAC algorithm is stochastic, resulting in stochastic results. For example, we can compare Figure 1 and Figure 3: the epipoles are definitely not in the same place...



(a) Result of the RANSAC algorithm



(b) Epipolar lines

Figure 2: Results for a second pair of images

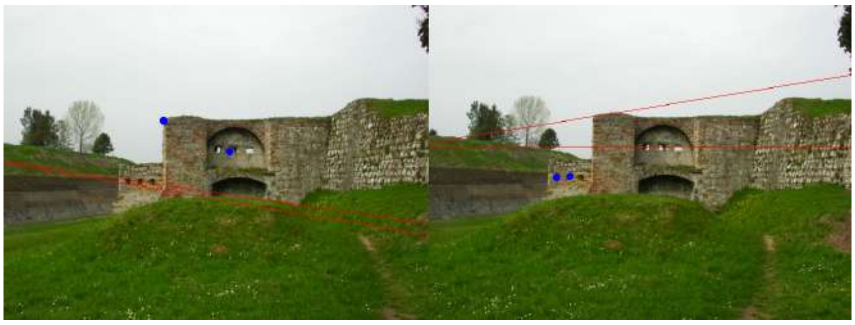


Figure 3: Other execution for the first pair of images