
Exercise 7 - Structure from Motion

Computer Vision

Alberto Montes (malberto@student.ethz.ch)

November 25, 2016

Feature extraction and initialization with epipolar geometry

For the first task, first it was required to initialize the structure with two images (the first and last one) matching its features. First the SIFT features were extracted for both images and matched by the descriptors extracted. The resulting matching is on Figure 1.

Then the 8-point RANSAC algorithm has been used to compute the inlier-set and then compute the essential matrix. This provide a better matching and can be seen on Figure 2.

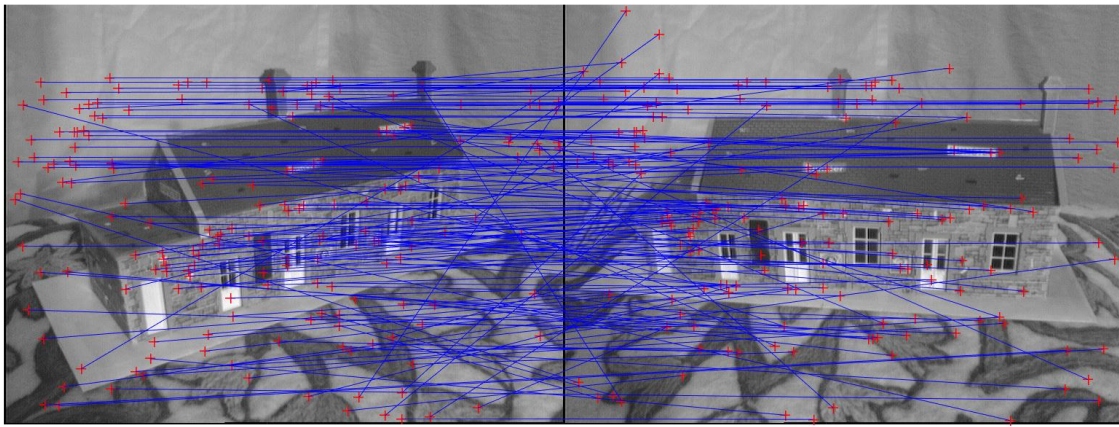


Figure 1: SIFT features matching

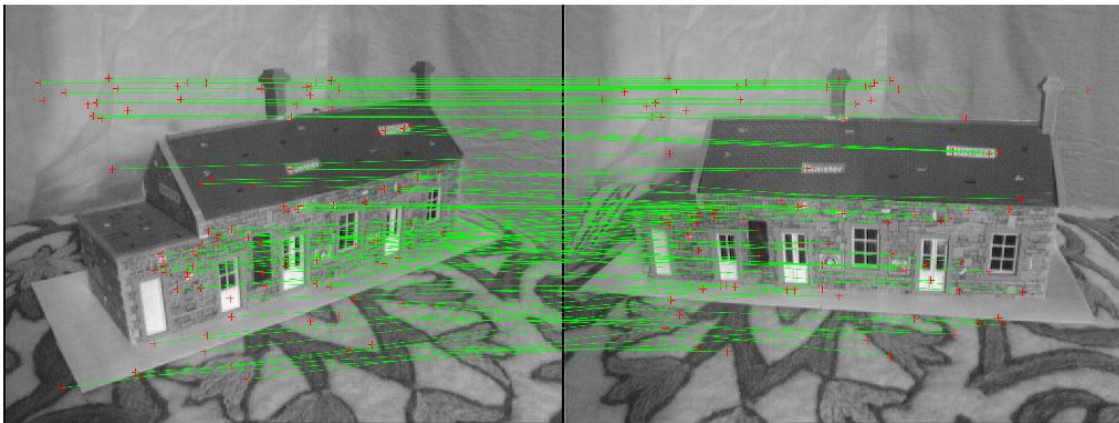


Figure 2: Inliers features matching using 8-point RANSAC algorithm

Then the image inliers points are calibrated using the K matrix and decompose the Essential Matrix to obtain the camera matrix. Finally the matched points are triangulated to obtain the 3D representation. The implementation is listed in the Listing 1.

```

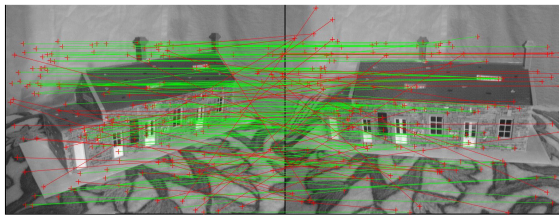
1 %% Compute essential matrix and projection matrices and triangulate
  matched points
2
3 % use 8-point ransac or 5-point ransac - compute (you can also optimize
  it to get best possible results)
4 % and decompose the essential matrix and create the projection matrices
5 t_F = 0.05;
6 [F, inliers_1] = ransacfitfundmatrix(fa(1:2, matches(1,:)), fb(1:2,
  matches(2,:)), t_F);
7
8 x1_normalized = K \ makehomogeneous(fa(1:2, matches(1, inliers_1)));
9 x2_normalized = K \ makehomogeneous(fb(1:2, matches(2, inliers_1)));
10
11 showFeatureInliers(img1, fa(1:2, matches(1,:)), img2, fb(1:2, matches
  (2,:)), inliers_1, 21);
12 E = K' * F * K;
13
14 Ps{1} = eye(4);
15 Ps{2} = decomposeE(E, x1_normalized, x2_normalized);
16
17 %triangulate the inlier matches with the computed projection matrix
18 [Xs{1}, ~] = linearTriangulation(Ps{1}, x1_normalized, Ps{2},
  x2_normalized);

```

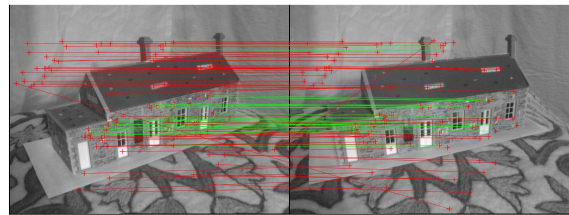
Listing 1: Compute essential matrix and projection matrices

Triangulation and adding new views

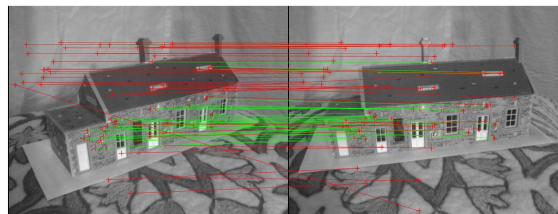
The next task has been to add a new triangulation from a new view. To do so, first it has been selected the descriptors of the first view which have projected in the linear triangulation. With this descriptors and the ones extracted to the new view, a feature matching has been performed. Now, having an initial correspondence between 3D points of the previous triangulation and the 2D points of the new view's features, a 6-point RANSAC algorithm is used to compute the projection matrix and the inliers points. The inliers matchings resulting from this algorithm at each of the views is plotted at Figure 3.



(a) house.000 ↔ house.001



(b) house.000 ↔ house.002



(c) house.000 ↔ house.003

Figure 3: Inliers features matching using 6-point RANSAC algorithm to find the Projection Matrix

The implemented code is shown in Listing 2 where all the steps are implemented with the given functions with the exercise statement.

```

1 %% Add an additional view of the scene
2
3 imgName3 = '../data/house.001.pgm';
4 img3 = single(imread(imgName3));
5 [fc, dc] = vl_sift(img3);
6
7 %match against the features from image 1 that where triangulated
8 fa_in = fa(:, matches(1,inliers_1));
9 da_in = da(:, matches(1,inliers_1));
10 [matches_2, ~] = vl_ubcmatch(da_in, dc);
11
12 % run 6-point RANSAC
13 x3n = K \ makehomogeneous(fc(1:2,matches_2(2,:)));
14 X_3 = Xs{1}(:, matches_2(1,:));
15 t_F = 0.01;
16 [Ps{3}, inliers_2] = ransacfitprojmatrix(x3n, X_3, t_F);
17 showFeatureInliers(img1, fa_in(1:2, matches_2(1,:)), img3, fc(1:2,
    matches_2(2,:)), inliers_2, 22);
18
19 %triangulate the inlier matches with the computed projection matrix
20 x1n = K \ makehomogeneous(fa_in(1:2,matches_2(1,:)));
21 [Xs{2}, ~] = linearTriangulation(Ps{1}, x1n(:,inliers_2), Ps{3}, x3n(:,
    inliers_2));

```

Listing 2: Compute essential matrix and projection matrices for new views

Plotting

The final part of the exercise consist on triangulate all inlier matches from all the views in a 3D representation. On Figure 4 there is the 3D plot of all the inlier points projected. On the plot can be observed how the number of red points (which correspond to the inlier features at the initialization) is much higher because than the rest of the views, as it was required to fit for each view the projection matrix using the RANSAC algorithm. Although, the projected points obtained look very consistent as can be observed a overlap between same features projections. Also the camera representation look to be correct as it plots the approximated locations where all the images where taken for this task.

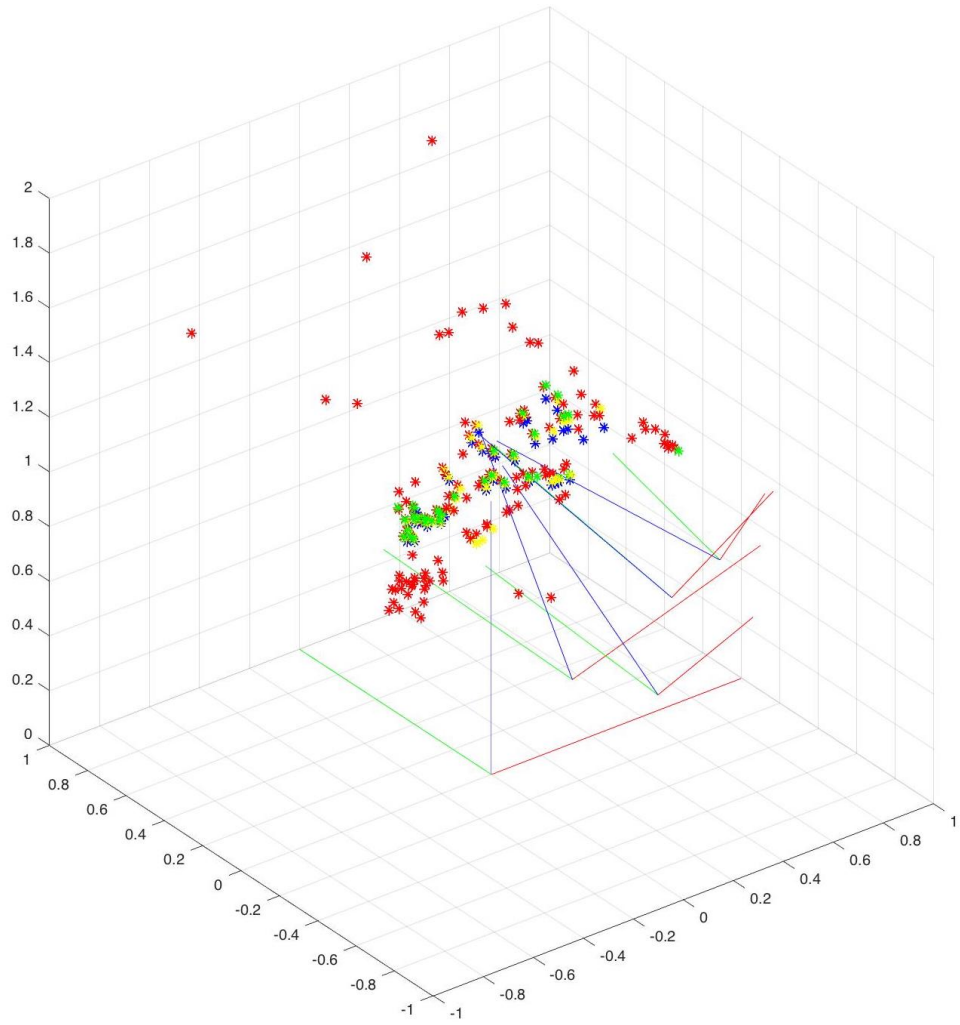


Figure 4: 3D projection of all the inlier features.