Computer Vision

Alessia Paccagnella

November 21, 2019

1 Structure-from-Motion

1.1 Feature extraction and initialization with epipolar geometry

For this task we use the first and the last image. The features in the images were extracted using vl_sift. Then I used vl_ubcmatch to save the matches between the two images.

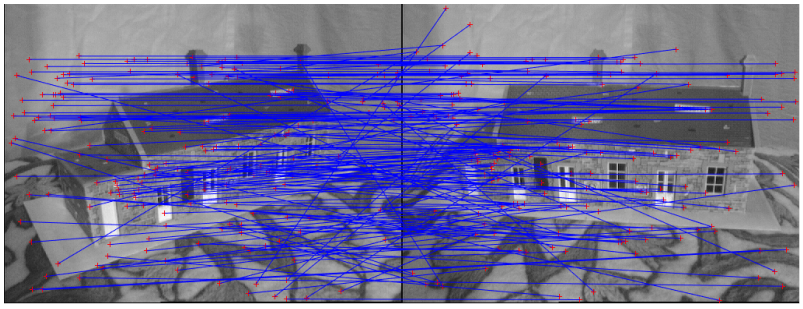

Figure 1: Feature matching images 0-4.

After that I computed the fundamental matrix with the 8-point ransac algorithm to find inliers and outliers. Here the inliers and outliers are shown. I used a function written by me to plot them in the same image with different colors.

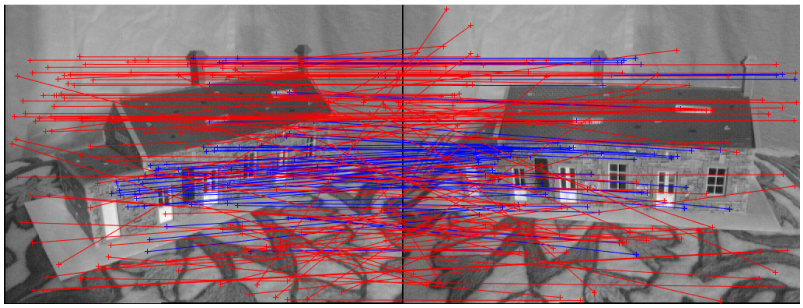

Figure 2: Inliers and outliers matches

Here the epipolar geometry is shown.

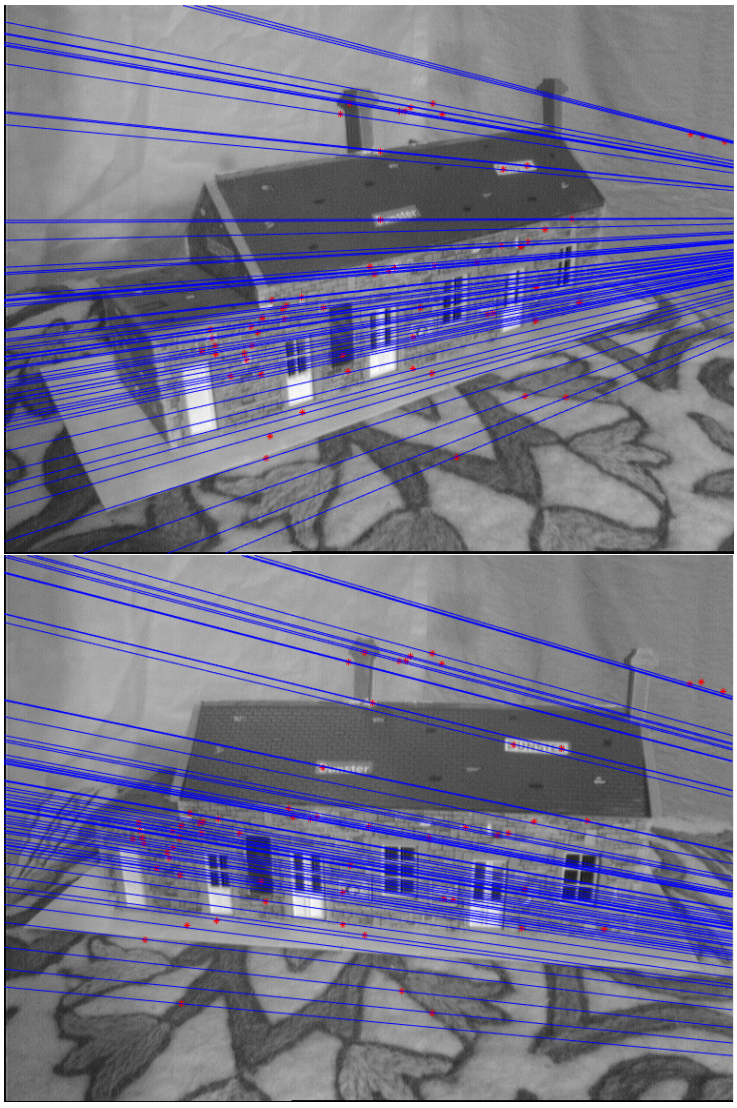

Figure 3: Epipolar geometry img 0 and img 4

After that, the task was to find the essential matrix. We know that `E = K'*F*K`. We calculate the calibrated inlier matches, using the calibration matrix `K`.

$$\mathtt{x}_{\mathtt{img1inlierscalibrated}} = K^{-1}\mathtt{x}_{\mathtt{inliersimg1}}$$
$$\mathtt{x}_{\mathtt{img2inlierscalibrated}} = K^{-1}\mathtt{x}_{\mathtt{inliersimg2}}$$

By considering that the first projection matrix is `[I|0]` we decompose the essential matrix calculated and obtain the projection matrix of the second camera (the view from the second image used).
After that it is possible to triangulate the inlier matches with the computed projection matrix.

1.2 Triangulation and adding new views

For this task I computed the features for the new image chosen and then I matched them with the inlier points from the first image ever used. I calibrated the new matches and ran 6-points ransac algorithm from the file given `ransacfitprojmatrix.m`. I checked if the matrix `R` had a positive determinant, otherwise I used the code provided in `fix.txt` to fix the otientation of the camera. I then triangulate the points with the calculated projection camera matrix.

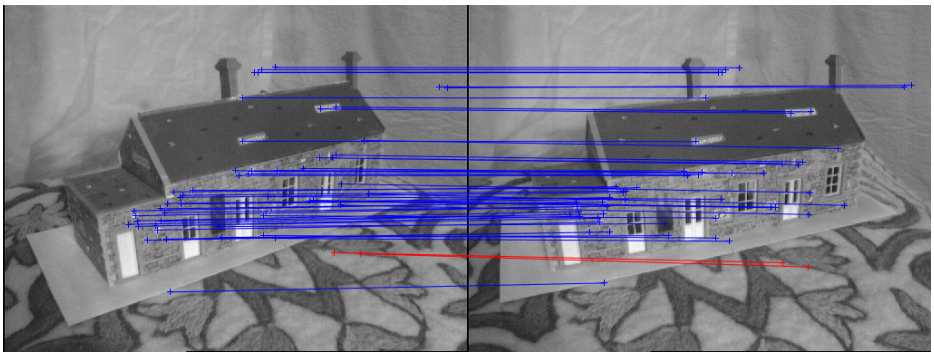

Figure 4: Feature matching with blue inliers and red outliers for images 0-1

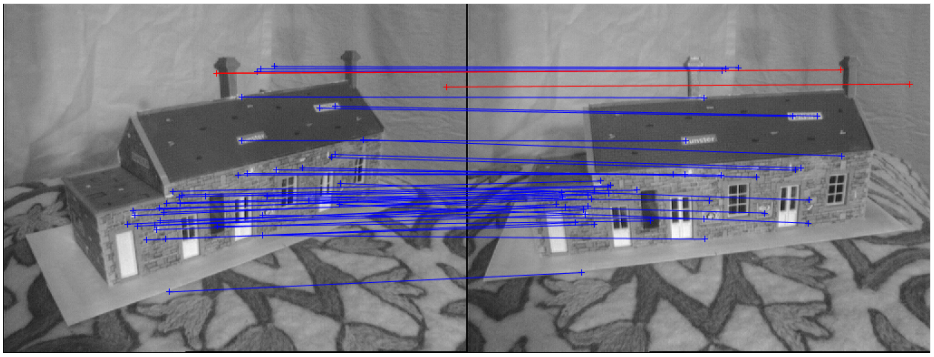

Figure 5: Feature matching with blue inliers and red outliers for images 0-2

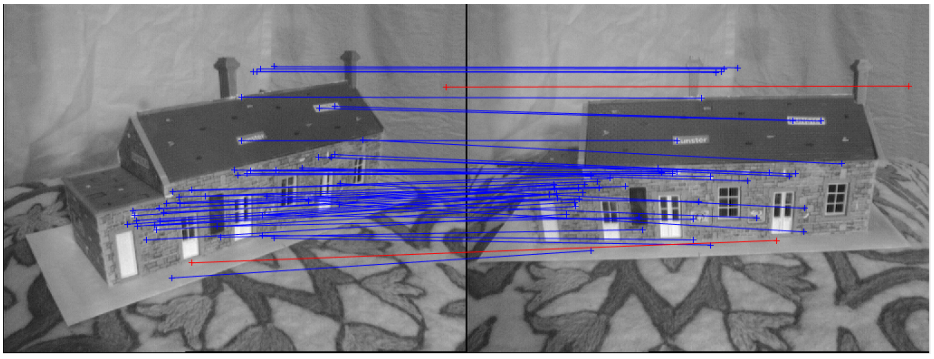

Figure 6: Feature matching with blue inliers and red outliers for images 0-3

1.3 Plotting

In the implementation of the exercise I computed the triangulated points of the inlier matches. Here the result along with the drawing of the camera poses is shown:

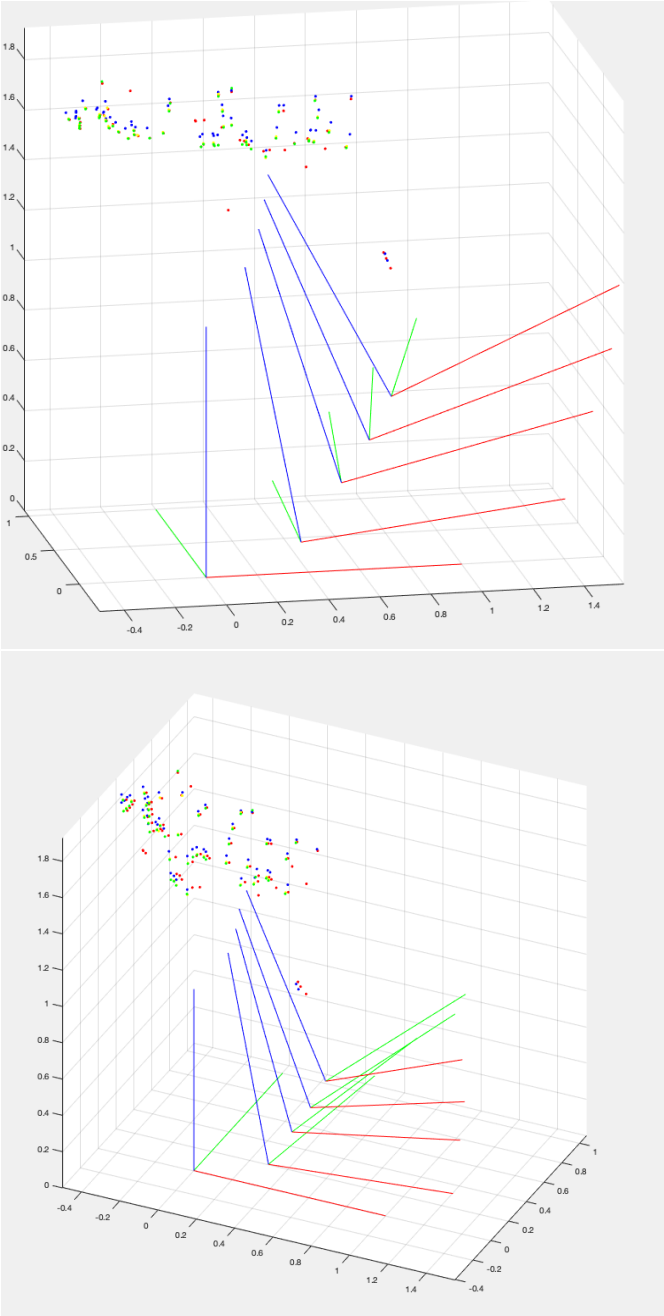

Figure 7: Here triangulated points imgs0-4 are shown in red, triangulated points imgs0-1 in blue, triangulated points imgs0-2 in yellow, triangulated points imgs0-3 in green.

The first camera is at the origin. The other cameras move up and rotate, which is expected because the views from the images that we used are moving in the same way.
I incremented the number of max trials for ransac from 2000 to 20000 as the code was often giving me warnings about ransac reaching the maximum number of trials.

1.4 Dense reconstruction

For this task I transformed the black and white image to RGB as requested in the assignment. After that I computed the depth choosing the z coordinate for all the inliers. I also tried to do it with the code from the stereo disparity calculation, by using the triangulated points obtained from there. The code is commented. I chose the second option implemented in `create3DModel` function. In these picture the height is representing the depth of the feature.

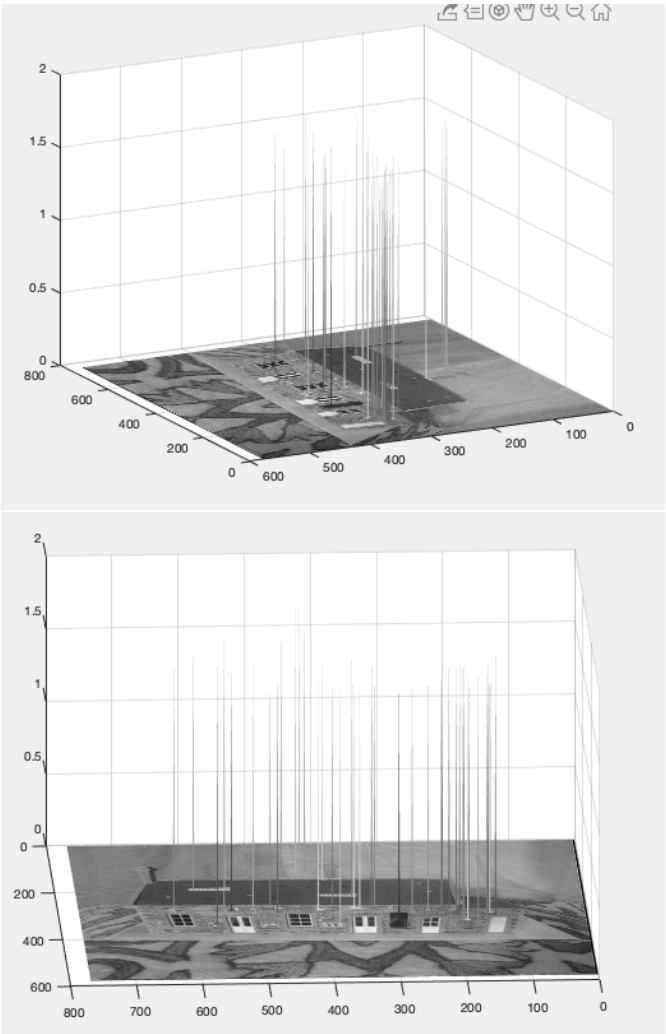

Figure 8: Dense reconstruction

I used this `http://www.maths.lth.se/matematiklth/personal/calle/datorseende13/notes/forelas6.pdf` to help me with the formulas for the assignment.