Computer Vision 2 - Assignment 2

Gijs van Iterson, Levi Knigge, Bünyamin Çetinkaya

May 13, 2022

1 Introduction

This report will look at the Fundamental Matrix, Chaining, Structure from Motion, Additional improvements and Real-World Data and Industry Tool. In the Fundamental Matrix part, the Eight-point algorithm will be implemented. Thereafter the algorithm will be normalized. After this the Normalized Eight-point algorithm will be performed via a RANSAC-based approach. Finally, these algorithms will be analysed. In the Chaining part a point-view-matrix will be constructed and analysed. In the Structure from Motion part the point-view-matrix will be used for the affine structure from motion. The factorization and stitching part will be done. At last the 3D structure will be plotted and analysed. In the Additional improvements part we will try to improve the algorithm even further. Lastly in the Real-world data and industry tool part, COLMAP will be used and analysed.

2 Data

The dataset consists of 49 images of a house from different angles. The 2D image will be used to reconstruct the 3D structure from the motion between the images.

3 Fundamental Matrix

In this section, methods for computing the fundamental matrix will be implemented. First with just the eight-point algorithm. This will first be extended by adding normalization and finally by implementing the RANSAC algorithm.

3.1 Eight-point Algorithm

The standard eight-point algorithm is implemented. It takes at least 8 known point pairs n between two images and estimates the fundamental matrix by constructing a linear equation containing the given points, as seen in figure 1. The singular value decomposition of the eight-point matrix will contain the values for the fundamental matrix. The solution is the refined by taking the singular value decomposition of the fundamental matrix, setting the smallest singular value to 0 and recomputing.

$$\underbrace{\begin{bmatrix} x_1{x_1}' & x_1{y_1}' & x_1 & y_1{x_1}' & y_1{y_1}' & y_1 & {x_1}' & {y_1}' & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n{x_n}' & x_n{y_n}' & x_n & y_n{x_n}' & y_n{y_n}' & y_n & {x_n}' & {y_n}' & 1 \end{bmatrix}}_{A} \begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{bmatrix} = 0$$

Figure 1: Eight-point matrix

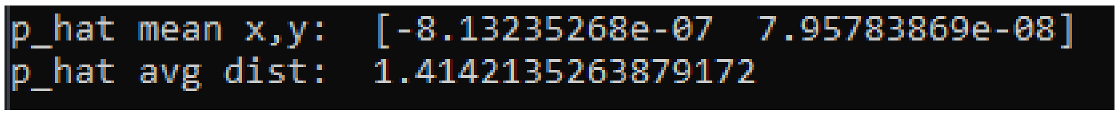
3.2 Normalized Eight-point Algorithm

We can improve the performance of the eight-point algorithm significantly by normalizing the input at very little computational cost. The chosen normalization is to set the mean of the points to 0 and the average distance to $\sqrt{2}$. This is done using transformation matrix T, seen in fig 2, with mean m and average distance d.

$$T = \begin{bmatrix} \sqrt{2}/d & 0 & -m_x\sqrt{2}/d \\ 0 & \sqrt{2}/d & -m_y\sqrt{2}/d \\ 0 & 0 & 1 \end{bmatrix}$$

Figure 2: Normalization matrix T

We verify the result of this transformation giving the desired result by applying it to the first pair of images. The resulting mean coordinates and average distance can be seen in figure 3 and correspond to the set requirements, though the mean is not exactly 0 due to rounding issues.



```
p_hat mean x,y:  [-8.13235268e-07  7.95783869e-08]
p_hat avg dist:  1.4142135263879172
```

Figure 3: Resulting mean and avg distance normalizing on first image pair

3.3 Normalized Eight-point Algorithm with RANSAC

Finally, RANSAC is added to the algorithm to improve the results of the algorithm. Here we perform the same method as before and use the resulting fundamental matrix on sets of points that were not in the eight-point matrix. We calculate the resulting image and amount of inliers using the Sampson distance. This process is repeated many times and the best fitting fundamental matrix with the largest set of inliers is selected.

3.4 Analysis

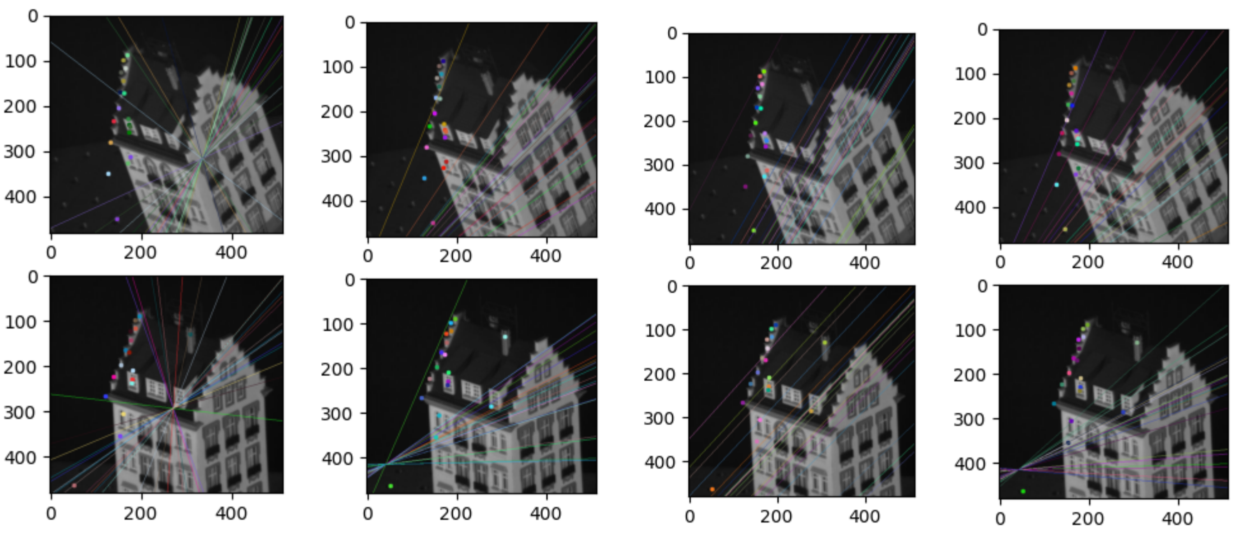
Here we will compare the results of the three versions of the algorithm. Figure 4 20 random epipolar lines between the first and last image. Although the first and last image have the biggest difference in camera position, there is little variation in the quality of the epipolar lines when taking different image sets. The first three columns show the three versions of the eight-point algorithm, while the final column shows the OpenCV implementation for reference.
The results of the standard eight-point algorithm do not look plausible4a. The camera coordinates that the point where all the lines meet, is right in the centre of the image despite this being the furthest two camera positions. This suggests only a slight perspective change, while this is the largest change in the dataset.

Adding normalization to the algorithm drastically improves the results, giving us a very plausible set of epipolar lines4b. The result of this algorithm is also closest to the reference set of OpenCV in 4d.
The results from the RANSAC improved algorithm seem to produce a very stable set of lines from a far away perspective4c. The results look plausible, although the fact that the lines are nearly parallel suggest a greater camera movement than is actually the case.

The epipolar constraint cannot always be satisfied due to the fact that the epipolar lines are not perfect. We see clear improvements in the quality of the epipolar lines. This also makes it possible to get closer to satisfying the epipolar constraint.

4 Chaining

Next, we chain multiple consecutive views into a point-view matrix. We take columns for each new point and rows for each new view. Below, we compare the result to a provided point-view matrix.
In figure 5 we see the resulting point-view matrix for the house dataset. We can see that the matrix is very sparse. Every row represents a view and every column represents a point. This means that vertical lines in



(a) eight-point (b) ep + normalization (c) ep + norm + ransac (d) opencv

Figure 4

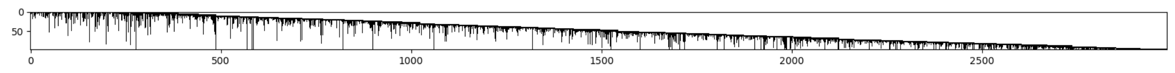


Figure 5: Caption

the graph correspond to points that are visible in consecutive views. We can see that most points are visible in multiple views, but the coverage of the graph is very limited. Seeing that the selection of interest points is somewhat randomized, it is still a plausible result that may need some refinement.
Compared to the provided point-view matrix, which is a dense matrix, we have very limited overlap between a large amount of consecutive views. This may limit the performance of possible reconstructions later in the project.
One difference is that the amount of points in our point-view matrix is significantly larger, so we can still make the matrix more dense by filtering out columns/points that occur only a few times in total.

5 Structure form Motion

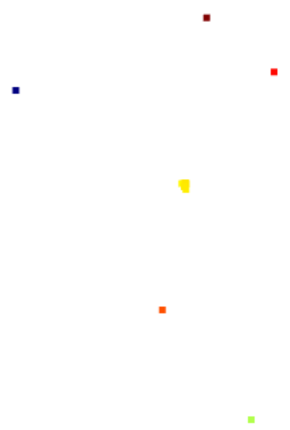
Finally, we will use the point-view matrix to attempt a 3D reconstruction of the house in the given image dataset. Firstly we select dense blocks from the point-view matrix with at least 3 points and 3 to 4 consecutive views.

5.1 Factorize and Stitch

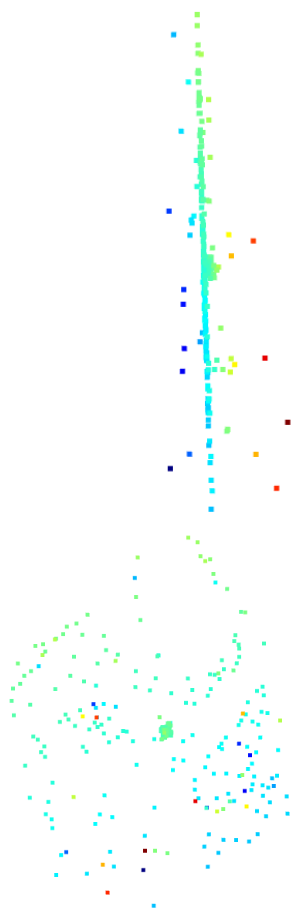
Factorization was done by obtaining either a dense matrix containing all views or by iteratively selecting dense matrices with fewer views. After obtaining the 3D points by factorization, they were stitched using Procrustes analysis.

Figure 6 shows several images of 3D point clouds obtained through with process with the point-view matrix as defined in Section 4. As can be seen in Figure 6a, using a dense matrix created with all views does not lead to a correct 3D reconstruction. As can be seen in Figure 5, not many points appear in all views and this fact will result in the given reconstruction.

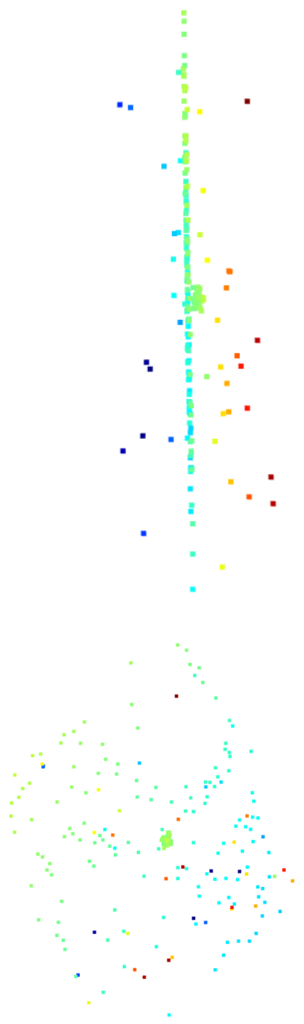
Iteratively reconstructing the 3D point cloud is better, but still does not result in a correct cloud. There is not a large difference in results when using sub-blocks containing 3 views (Figure 6b) or 4 views (Figure 6c). Mainly depth is missing, which can be seen in the top images. When viewing from the front (bottom images) it can be seen that the clouds slightly have the shape of the house in the original images.



(a) Full dense matrix



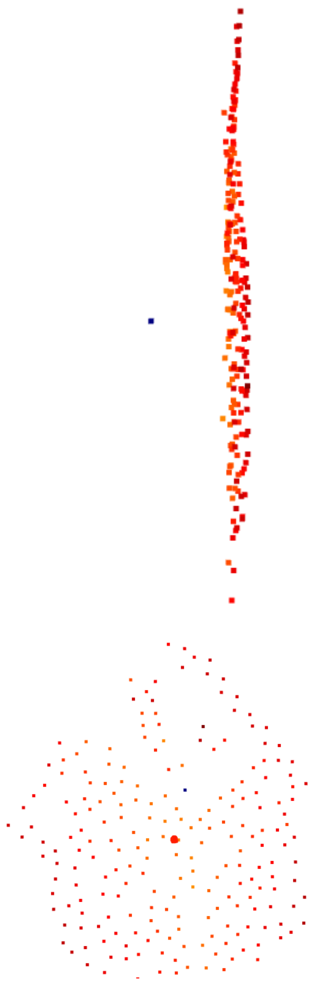
(b) Stiched after 3-view dense matrix selection



(c) Stitched after 4-view dense matrix selection

Figure 6: Results obtained with a SIFT point-view matrix

Figure 7 shows the results obtained with the provided point-view matrix and iterative reconstruction. Compared to sampling our own matrix with the same number of views, as shown in Figure 6, using the provided matrix leads to better results. The shapes resemble the house in the original images better, as can be seen in the bottom images. Like using our own matrix however, the depth is missing in the reconstruction, which is visible in the top images.


(a) Stiched after 3-view dense matrix selection

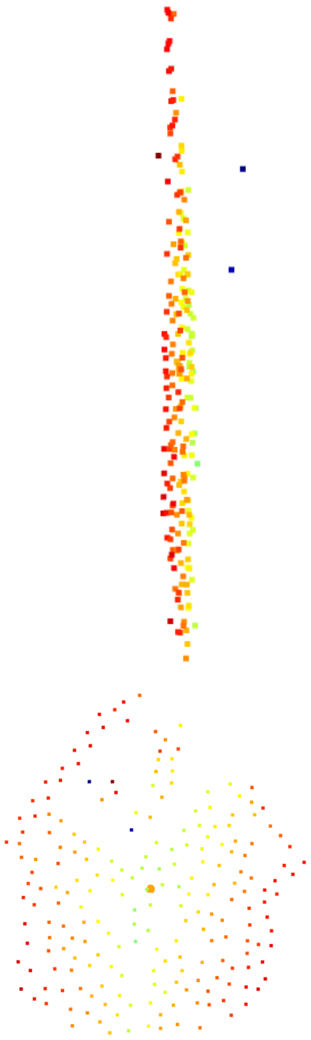

(b) Stiched after 4-view dense matrix selection

Figure 7: Results obtained with provided point-view matrix

As can be seen in Figure 8, the best results are obtained by directly factorizing the provided point-view matrix, i.e. using all views at once. Doing this results in a correct looking 3D reconstruction which appears to have the correct depth.

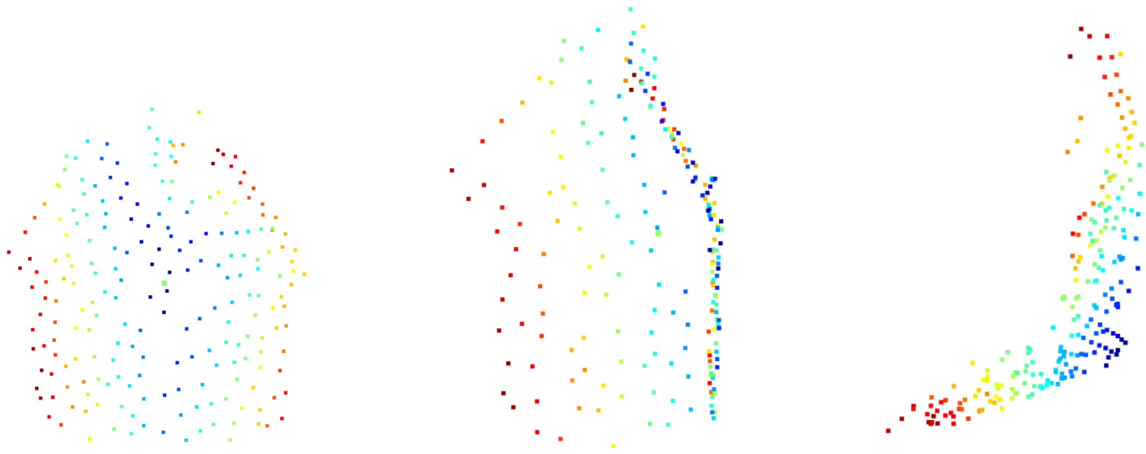


Figure 8: Results obtained with full provided point-view matrix

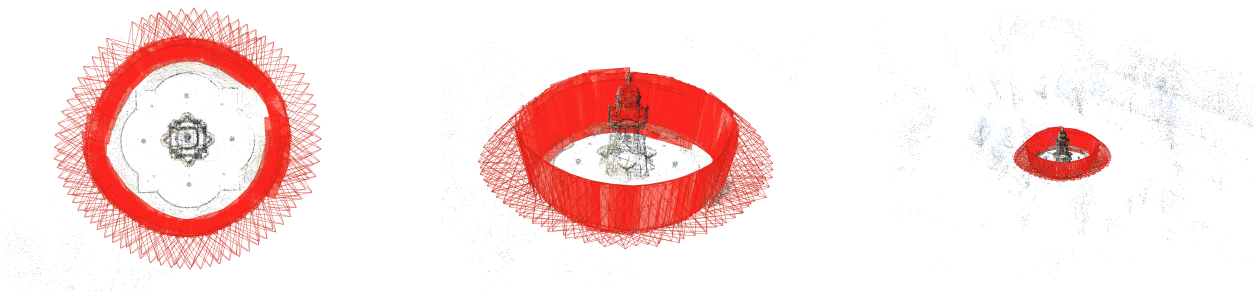
6 Additional improvements

Additional improvements could include filtering the point-view matrix in various ways. A possible way of filtering the point-view matrix could be to use features different from SIFT. Introducing constraints upon matching might also benefit results. Also a different factorization method could be considered. Due to insufficient time, these possible improvements unfortunately could not be implemented and tested experimentally.

7 Real-world Data and Industry Tool

1. The COLMAP tool is used for this part. With the Structure-from-Motion (SfM) [1] and Multi-View Stereo (MvS) [2] algorithms. The Windows pre-build binaries are used for these tests. To run the code, the automatic reconstruction is used. The data type is set to individual images. The quality is set to high. The number of threads is changed to 12 and one GPU is used.

2. Real-world images of the Sarphati monument are used. This data contains 116 images of going around the monument.



(a) Top view of the Sarphati monument.

(b) Side view of the Sarphati monument.

(c) Zoomed out side view of the Sarphati monument.

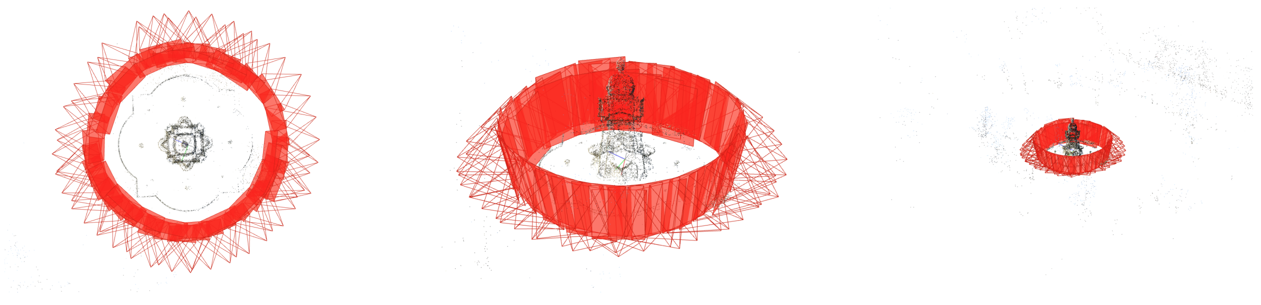
Figure 9: Image-based 3D reconstruction of the Sarphati monument with 116 images.

   Figure 9 shows the reconstruction from the images. The results look good. The monument is reconstructed correctly in 3D, as can be seen in Figure 9b. It is also in a good cirle, as can be seen in Figure 9a The background noise is however a lot when looking from a zoomed out perspective as shown in Figure 9c.

3. To further improve the results, a fixed-camera approach can be adopted. This will improve the measurement accuracy of the SfM-MvS pipeline [3]. With this method, only four images are needed instead of a couple of [3]. Using only these four images will reduce the processing time, while keeping an accurate result [3]. It also works with sub-optimal camera calibration [3]. This fixes the problem of all

the noise in the image with a lot of images, see Figure 9c. By reducing the number of images, we will also reduce the noise, as can be seen in Figure 11c, more on this in section 7.5.

4. To test the algorithm with adjacent frames reduced, half of the images are alternately removed, to the total images used for this result are 58. The same settings were used as the previous results.



(a) Top view of the Sarphati monument.

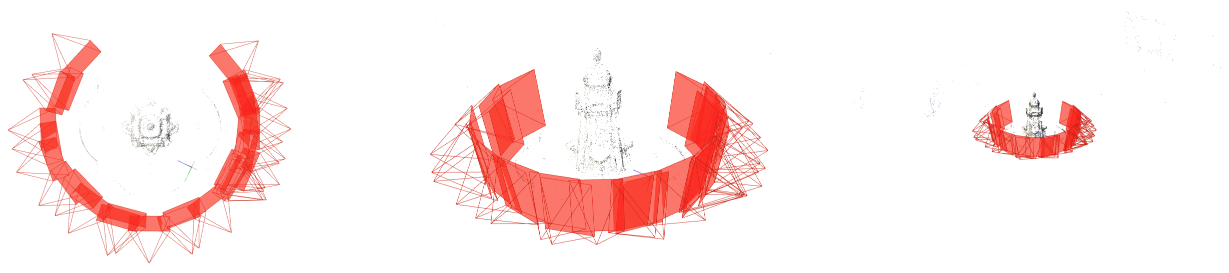
(b) Side view of the Sarphati monument.

(c) Zoomed out side view of the Sarphati monument.

Figure 10: Image-based 3D reconstruction of the Sarphati monument with 58 images.

As can be seen in Figure 10, the result of reduced number of frames is that there are less details. However the result is also that the noise is a lot less, as can be seen in Figure 10c compared to the previous result in Figure 9c.

5. To reduce the number of background pixels included in the reconstruction, we can reduce the number of images used for the reconstruction. Therefore the algorithm is run with only 29 images in this result. The algorithm however mistakes some images for other, so there is a part missing see Figure 11a. The images of the top part went to the right side of the image, as can also be seen in Figure 11a



(a) Top view of the Sarphati monument.

(b) Side view of the Sarphati monument.

(c) Zoomed out side view of the Sarphati monument.

Figure 11: Image-based 3D reconstruction of the Sarphati monument with 29 images.

When we compare Figure 11c with Figure 9c and Figure 10c we can see that the background noise is reduced when the algorithm is run with less images.

8 Self-Evaluation

Gijs did the implementation for sections 3 and 4, with Levi helping for section 5 and the visualization and analysis of 3 and 4. Bunyamin implemented and analyzed section 7 and wrote the introduction and conclusion. Everyone contributed to the details of the report.

Conclusion

In this report we looked at the Fundamental Matrix, Chaining, Structure for Motion, Real-world data and Industry tool. In the end we learned a lot about structure from motion. After all the steps we managed to implement the complete structure from the motion algorithm, where we used the point-view matrix to attempt a 3D reconstruction of a house. We found that the best results are obtained by directly factorizing the provided point-view matrix, where all views are used at once. Due to insufficient time we did not do some additional improvements. However, we made some possible recommendations that can be implemented next time, f.e. using a different factorization method. In the end we also looked at the real-world data and the industry tool COLMAP.

References

[1] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[3] Luigi Parente, Neil Dixon, and Jim Chandler. Optimising the quality of an sfm-mvs slope monitoring system using fixed cameras. *The Photogrammetric Record*, 09 2019.