Yue Pan
yuepan@student.ethz.ch

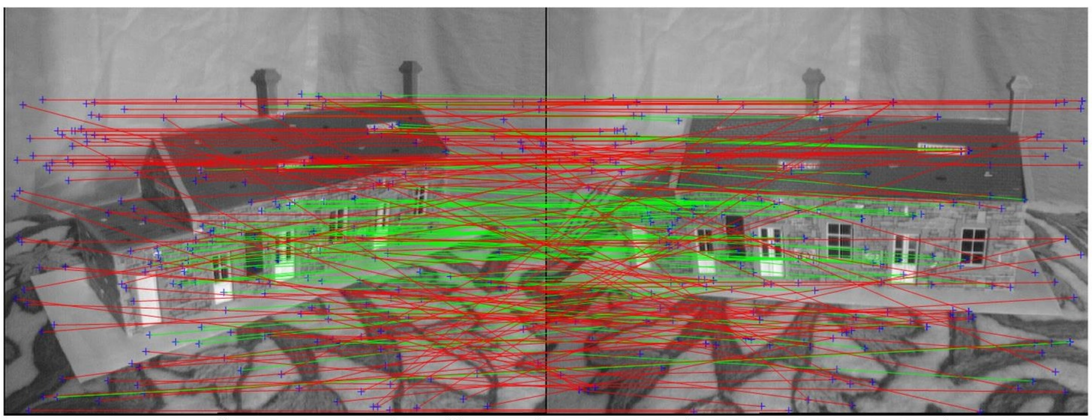
# Assignment 7

Computer Vision
November 22, 2019

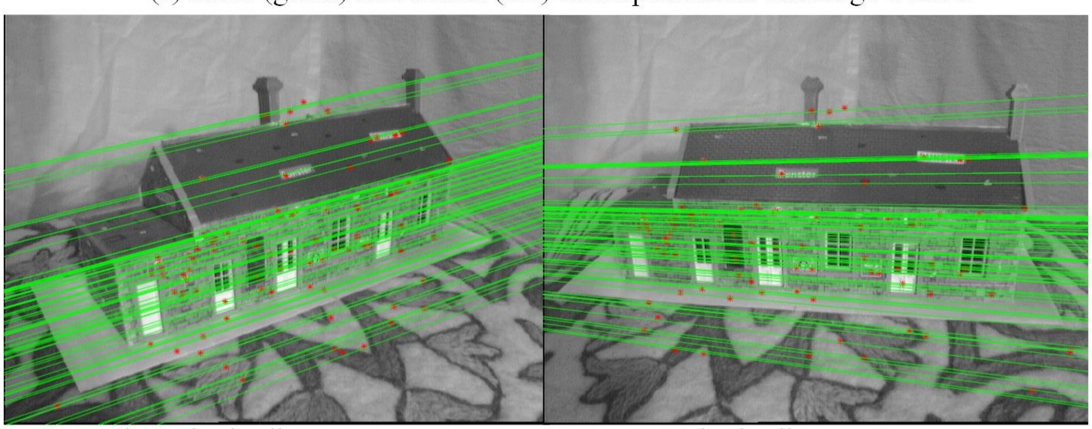
## Task 1: SFM initialization with epipolar geometry

As the first step of structure from motion, we should do the initialization by using the epipolar geometry of two images. These two images should not only share enough common feature points but also have a long enough baseline to achieve more accurate triangulation result (in this case, image 1 and image 5).

Firstly, with the help of VLFeat, SIFT feature points are detected from the image pair. Then we match the feature points with a second nearest neighbor over nearest neighbor ratio (set as 1.5). After that, we use the 8-point RANSAC algorithm in order to further filter the outliers and get the fundamental matrix as well. The results are shown in Fig.1, from which we can notice that about half of the initial matchings are regarded as outliers and filtered in this step.



(a) Inlier (green) and outlier (red) correspondences on Image 1 and 5



(b) Epipolar lines on Image 1 (c) Epipolar lines on Image 5

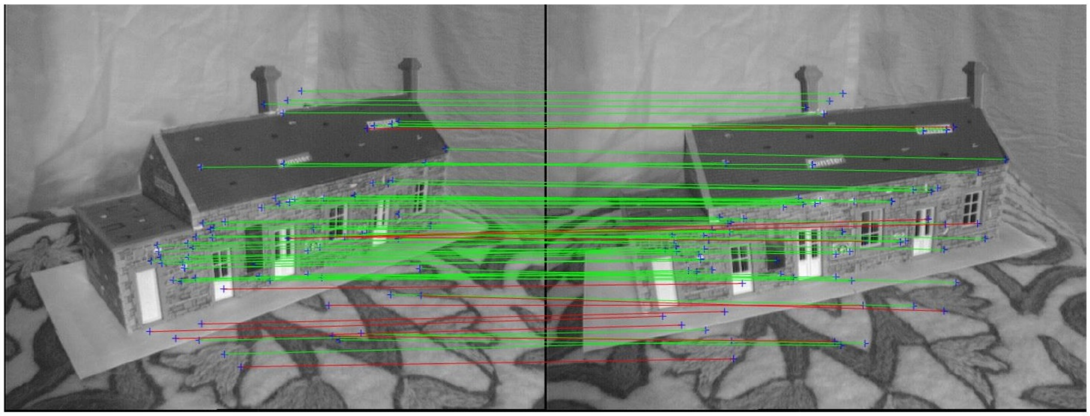
Figure 1: Initialization of SFM: epipolar geometry between image 1 and 5

With the fundamental matrix $F$, we can calculate the essential matrix $E$ according to Eq.1 and further decompose $E$ into relative pose of the camera $R, t$ according to Eq.2. Assuming the pose of the first camera as identity matrix, the pose of the fifth camera can be therefore determined.

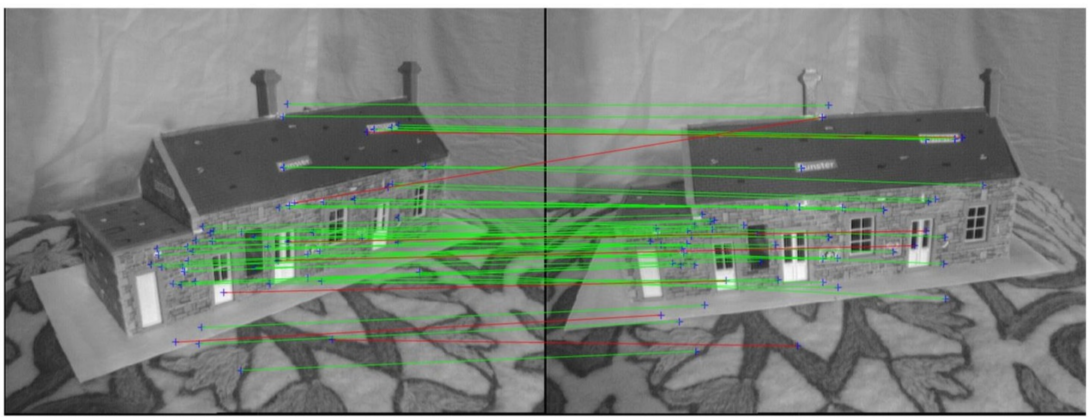
$$E = K^T F K \tag{1}$$

$$E = t^{\wedge} R \tag{2}$$

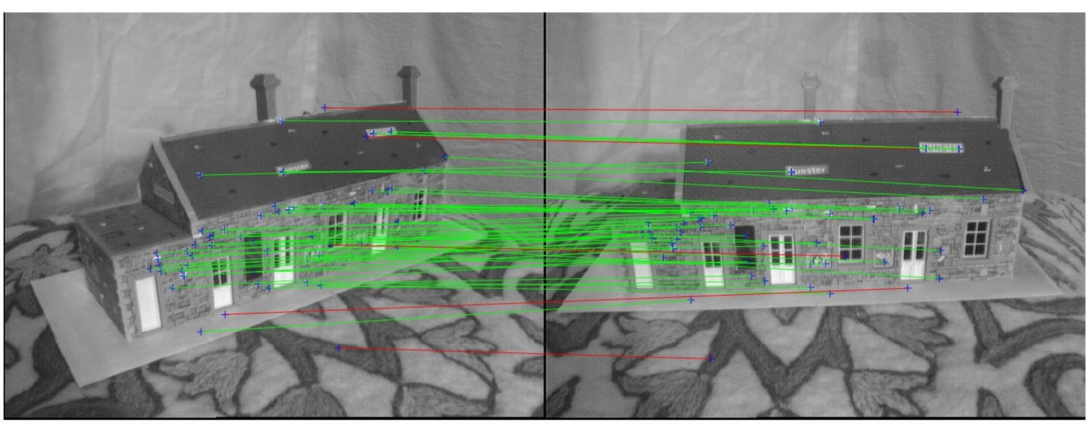
# Task 2: Triangulation and adding new views



(a) Inlier (green) and outlier (red) correspondences on Image 1 and 2



(b) Inlier (green) and outlier (red) correspondences on Image 1 and 3



(c) Inlier (green) and outlier (red) correspondences on Image 1 and 4

Figure 2: Inliers and outliers after 6-points RANSAC when adding new views

Given the projection matrix of two images and the filtered correspondences, we can use triangulation to calculate the coordinates of 3D points. Then we can add new views to the structure from motion system by estimation transformation from the 3D-2D correspondences. Since we already have the camera matrix $K$, the extrinsic element matrix $R, t$ can be estimated from the calibrated correspondences using DLT by 6-points RANSAC.The inlier and outlier matchings determined by RANSAC for each newly added view are shown in Fig.2. Besides, we need to guarantee the rotation matrix $R$ has a positive determinant. If not, we need to take the opposite of $R$ and $t$. Then, we can triangulate the correspondences between the newly added view and the reference image.

# Task 3: Plotting

For this task, the triangulated 3D points from the previous tasks and the camera poses are plotted as shown in Fig.3. Red points are the triangulated points from the initialization using image 1 and image 5. Green, Blue and yellow points are the triangulated points from image 2, 3 and 4 respectively taking image 1 as the reference.

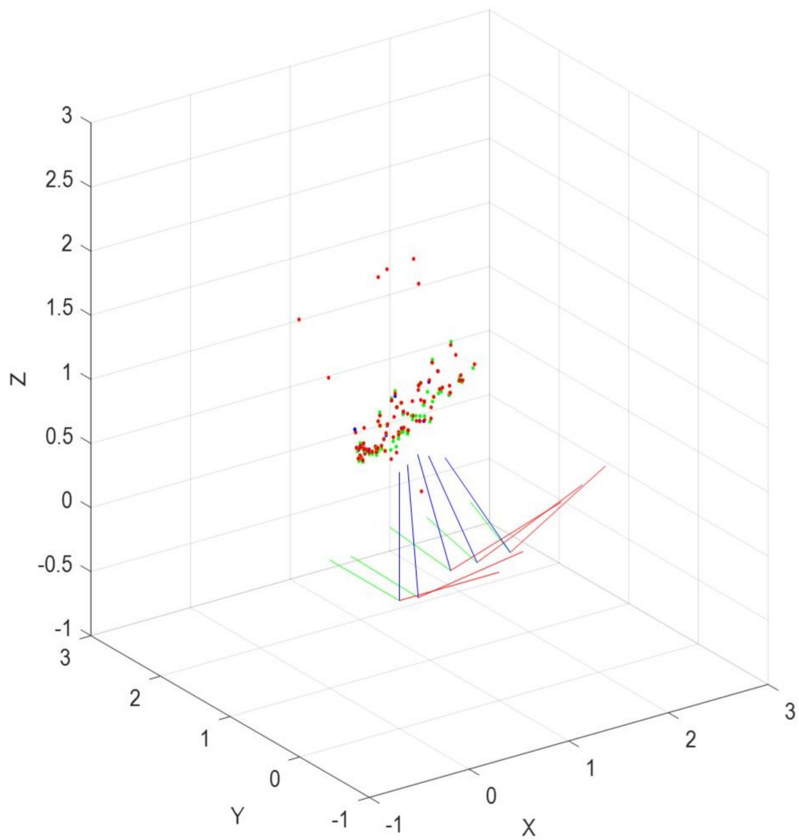



Figure 3: Triangulated points and camera poses plotted in 3D (Default perspective)

Since the first camera's pose is set as identity and the length of baseline for initialization is set as 1, the absolute orientation and scale of the structure from motion system is unknown. A more intuitive perspective of the 3D points and the camera pose is shown in Fig.4, in which the motion of camera and the sparse point cloud of the house model can be clearly seen.

Although no bundle adjustment is done to optimized the pose and 3D points, we can notice that points triangulated by different camera pairs are almost overlapped with each other. However, to achieve result with higher accuracy and consistency, bundle adjustment should be adopted.

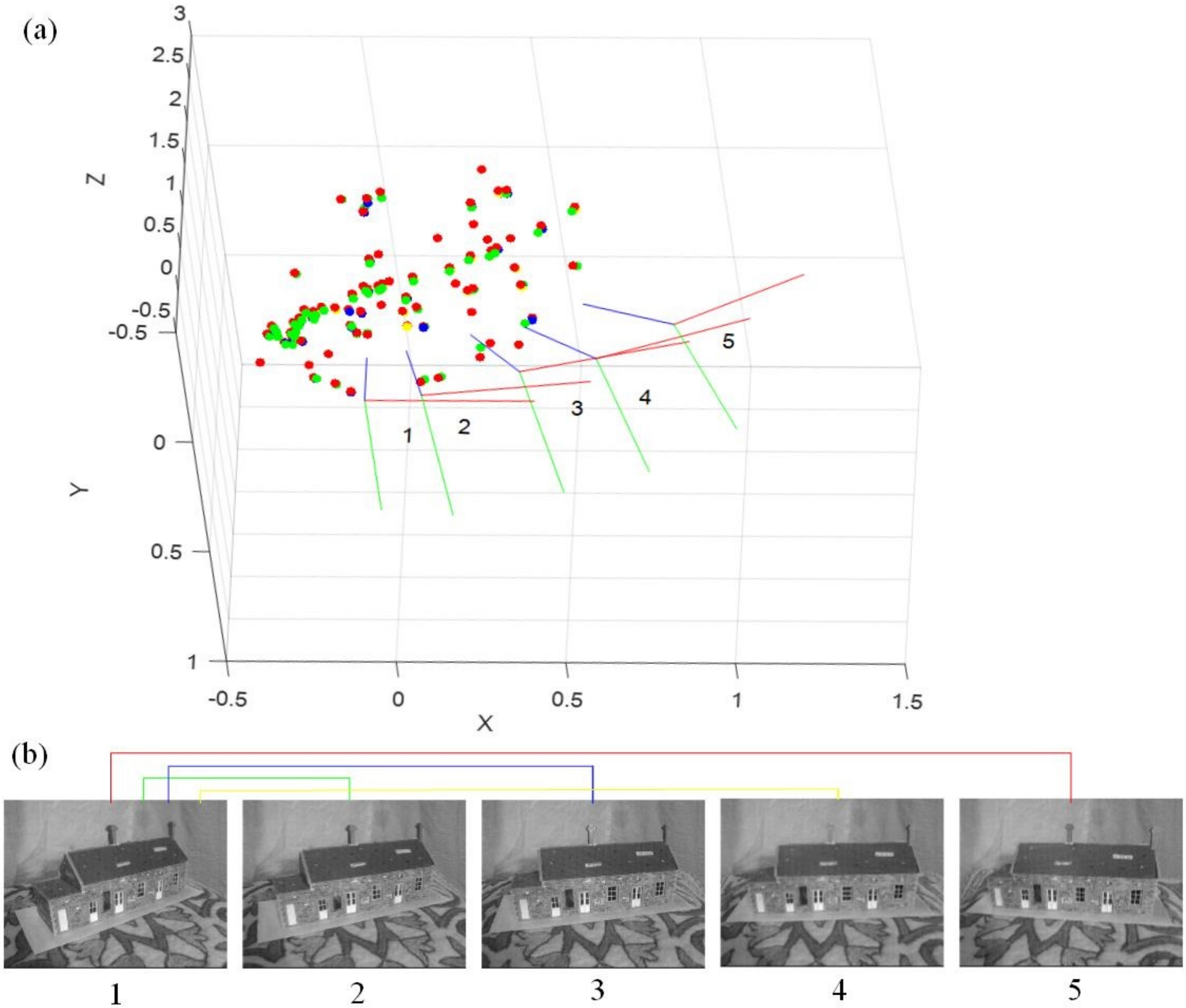



Figure 4: Triangulated points and camera poses plotted in 3D (easy-to-understand perspective): 3D points in different color are triangulated from different image pairs as shown in (b)

# Task 4: Dense reconstruction

Codes for the last exercise (MVS) are adopted to do the dense reconstruction in this task.

After rectify the image pair as shown in Fig.5, I apply the graph cut algorithm to generate the disparity map. Then we can calculate the depth $Z$ for each pixel from the disparity $d$ using Eq.3, in which $b$ is the length of baseline and $f$ is the focus length.

$$Z = \frac{bf}{d} \tag{3}$$

Using the function *create3DModel.m*, the result is shown in Fig.7. It seems that there's still some problem of the dense reconstruction point cloud.

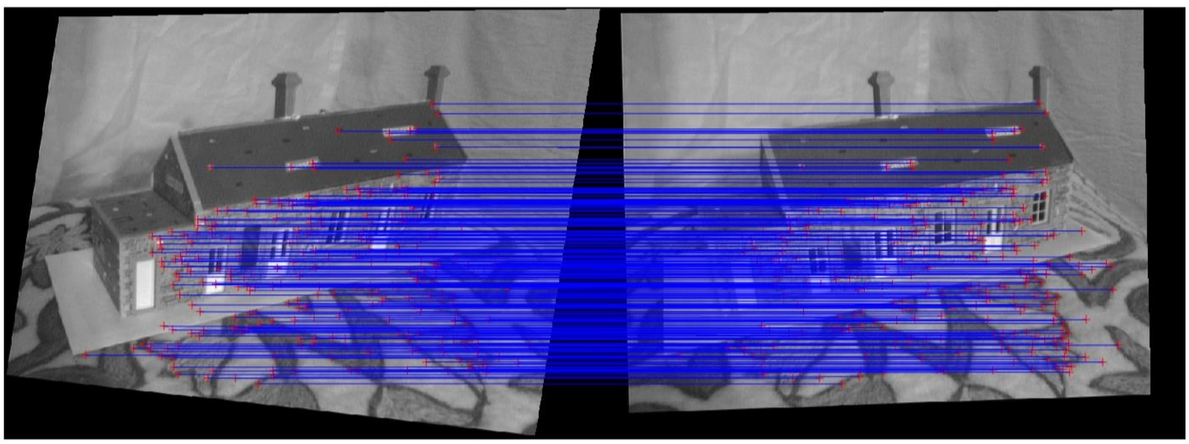


Figure 5: Rectified Image 1 and 2 for dense disparity estimation

Figure 6: Disparities map of Image 1 and 2 generated using Graph Cut

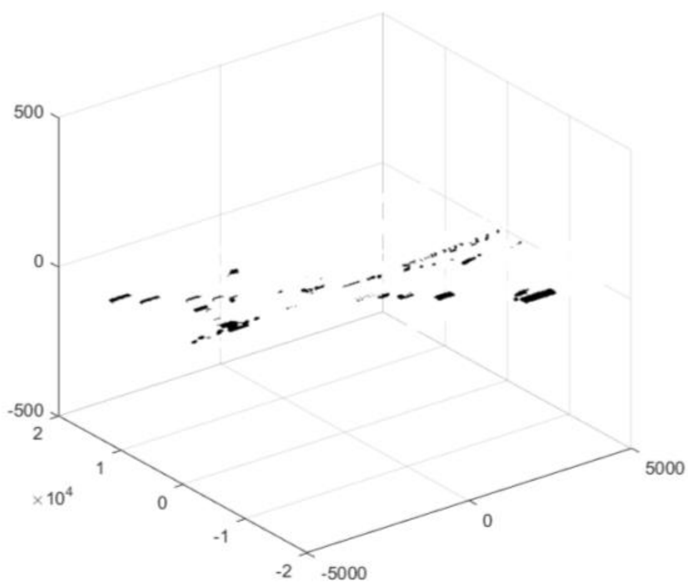



Figure 7: Reconstructed point cloud with texture