Bachelor-Project
SS 2021

Julian Jandeleit

Camera Pose Estimation for Ushichka Dataset
Status Report 1

Structure from Motion Pipeline

My approach was to first build a working pipeline that is able to reconstruct 3D points using the pinhole model. Figure 1 shows artificially generated 3D points,

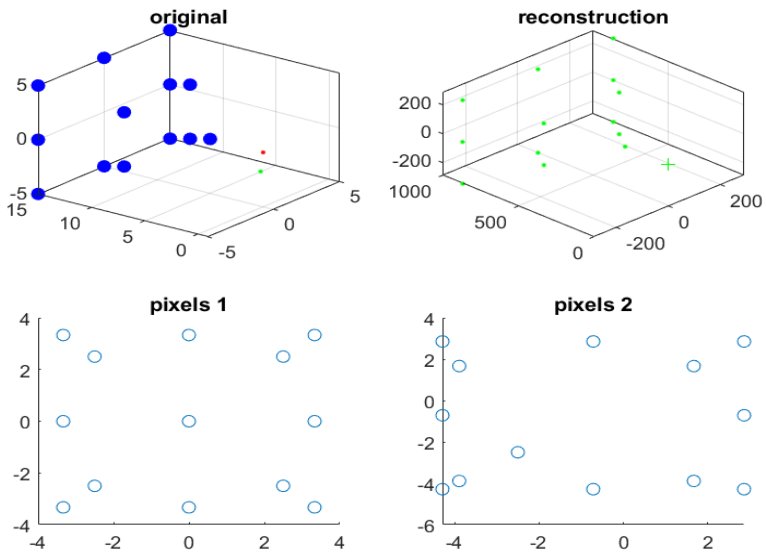

Figure 1: Artificial Data Reconstruction

their projection onto pixel coordinates of 2 pinhole cameras that were placed into the scene and the up-to-scale reconstruction of the 3D structure.
I chose a pyramid shape as the planar surface I used first resulted in reconstruction problems. It seems they represent a limit in the estimation algorithm for the fundamental matrix (no reference at the moment).
When looking at the result, the pipeline seems to be working. I also tested it on pictures of objects I took by myself and while not perfect, I was able to at least recognize the objects I photographed. However, I did not spend too much time to optimize the result as I wanted to quickly move to the Ushichka Dataset.
I used functions I wrote myself, the vl_feat library and inbuilt MATLAB methods.

Preprocessing

Due to the nature of the thermal cameras and the cave environment, there also often are ghost structures detected, that appear and disappear between frames without maintaining shape. I identified them to be similar to Salt & Pepper noise and removed it using a small median filter. I also applied a small Gaussian filter to be safe but as of now there wasn't a specific reason for it. Figure 2 shows the result.
For easier use and faster computation, the lidar scan is reduced to the region of interest, shown in figure 3. The dataset also gets down-sampled to an average in the grid size of 0.01. I work on the *2018-08-17* thermal recording and the *RecordingChamberMeshHighQuality.ply* lidar point cloud.

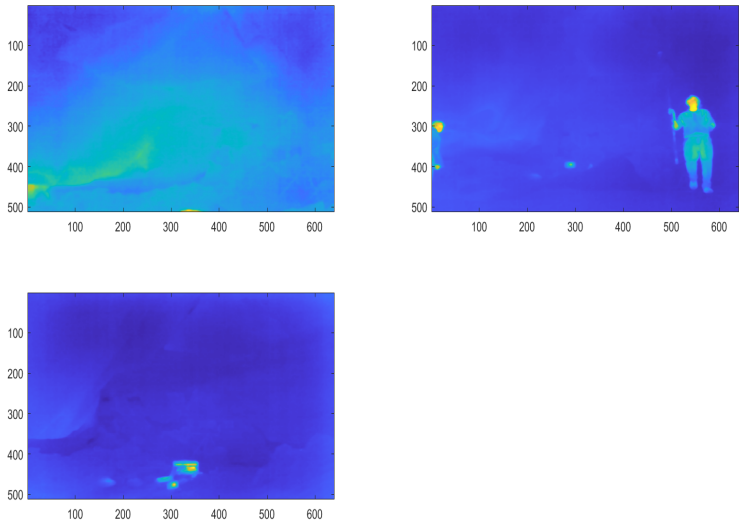

Figure 2: False Color representation of preprocessed Image

Feature Detection & Matching

When applying feature detectors like SIFT on the thermal images, they only provide few *meaningful* features. However, to construct a useful point cloud, I look at the video as an image stack and combine all SIFT features of a camera into a large set so that features from different frames can be matched.
As [1] proposed, I tried several versions of edge detection for SIFT but got similar results to the normal image so I stuck with it. However, I think an edge

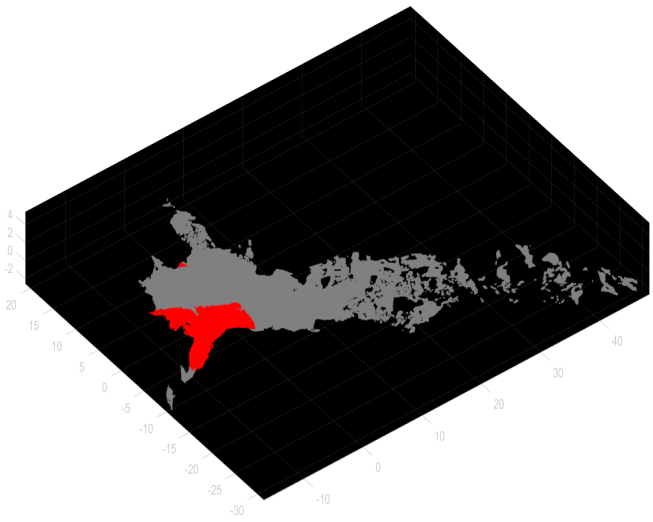

Figure 3: Lidar ROI

based approach could significantly improve the quality of matches, if a more specific method is used to directly compare specific edges and generate features from fitting comparisons.
Figure 4 shows all features above a certain threshold. As we use a complete image stack to generate features, there may be many features which match to one feature in second camera which can distort later calculation. Also, because there only is few visible structure, there are many mismatches which are mathematically as similar as correct ones. To tackle both problems, the matches get filtered for uniqueness and DBSCAN clustering is used to filter out outliers.

Point Cloud Reconstruction

The filtered feature correspondences are used to estimate the essential matrix directly using the MSAC scheme. Distortion is not yet considered in the camera parameters. Then, the relative Rotation and Translation is extracted from the Matrix.
Figures 5, 6 show the MSAC inliers. We only consider Camera pairs $(1, 3)$ and $(2, 3)$ as this configuration consistently shows best results. Figure 7 shows the epipolar lines for the camera pair $(2, 3)$. Somehow the essential matrix, and therefore also the relative camera position, is quite unstable. About every third time I get completely different epipolar lines. Maybe the features in 3D are to

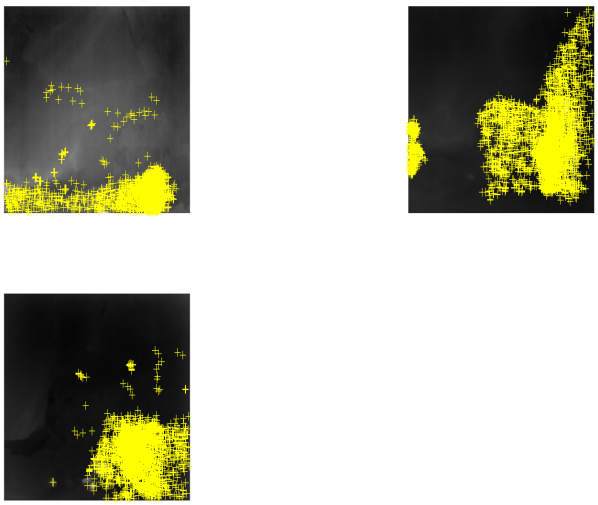

Figure 4: Raw SIFT Features

planar? I spent quite some time in experimenting with this step, but I'm not quite satisfied with the stability of the result.

However using the reconstructed relative camerapositions, the 3D Point Cloud of all matched features can be interpolated. Figure 8 shows the reconstructed feature matches from the point of view of the 3. camera. Note that the reconstruction scale cannot be retrieved from the images alone and both point clouds have their own scale factor. That is to say that they are only displayed in the same coordinate system and not combined into a single cloud.

Point Cloud Registering

To find the position of the cameras in the lidar scan, we extract *(*FPFH-Features), which describe the relation between each point and its k nearest neighbors, from both clouds. They get matched and the thermal clouds get transformed by an rigid transformation into the lidar coordinates in a way that minimizes the matching error. In the final step they get registered using Iterative Closest point algorithm.

The same transformations are applied to the camera position, which leads to the result you see in figure 10. The reconstruction seems to be at roughly the

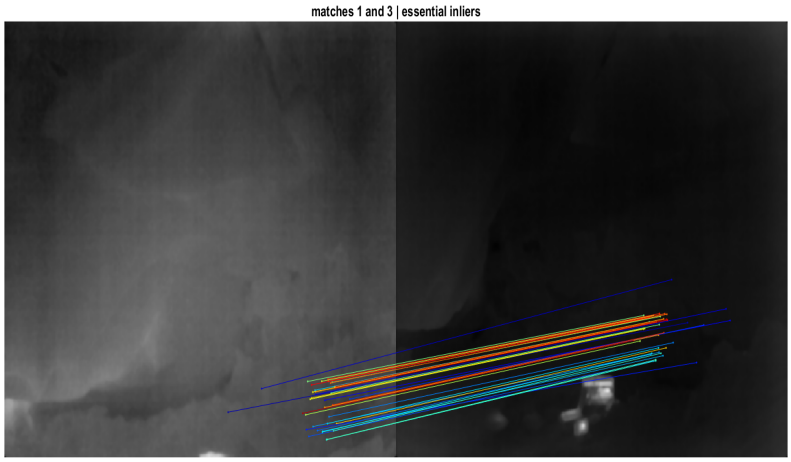

Figure 5: Essential Inliers K1,K3

right position, but the cameras are clearly not yet at the right position. Also, the matching of the FPFH features does not look optimal to me.

Note, that a rigid transformation (as used here) does not change the scale of the cloud. So the scaling factor between the cloud remains unknown and was guessed by eye manually. The paper [1] proposes a method using the camera baselines of both clouds. This is not applicable here, as I don't have the individual camera positions used to get the final lidar point cloud. As of now, I did not find a useful alternative, maybe I need to learn more about this special topic in future parts of the lecture. I think getting the scaling factor right is crucial for the 3D-matching and registration process. I'm not sure if it might be necessary to further increase the size/distribution of the thermal point cloud to capture more detail.

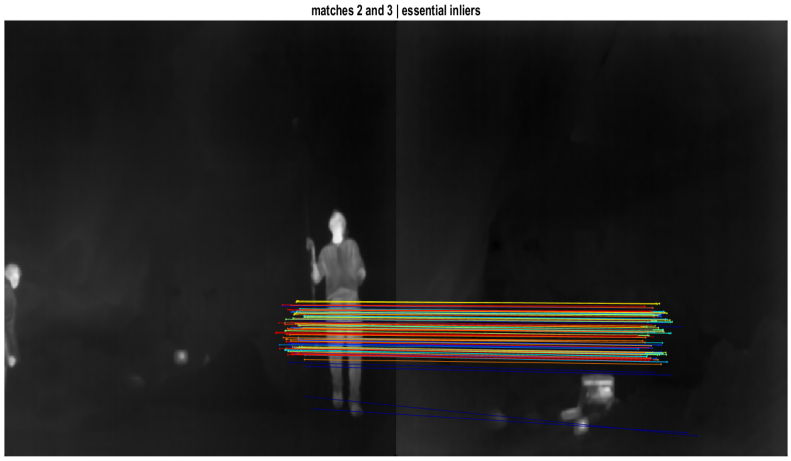

Figure 6: Essential Inliers K2,K3

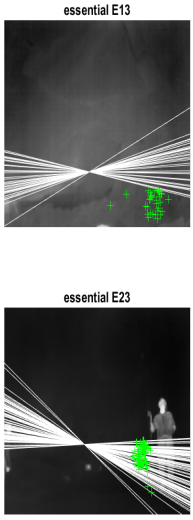

Figure 7: Epipolar Lines for K2,K3

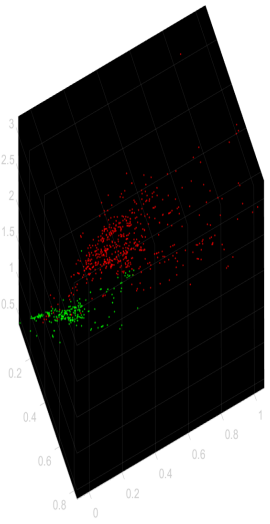

Figure 8: Reconstructed Point Clouds

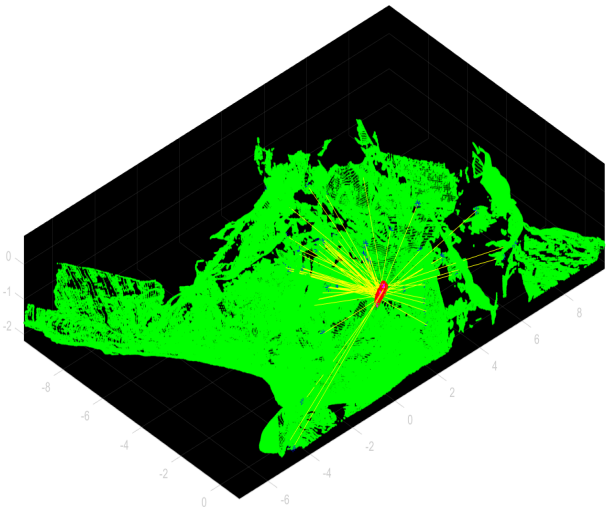

Figure 9: Point Cloud 2,3 Lidar matches

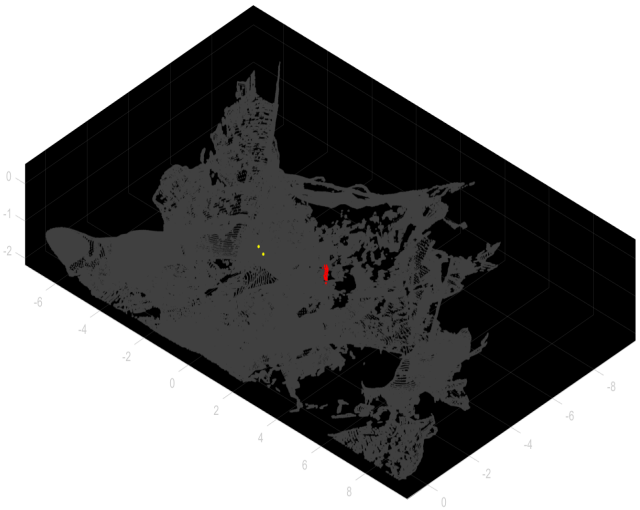

Figure 10: Final Reconstruction

Bibliography

[1] T. P. Truong, M. Yamaguchi, S. Mori, V. Nozick, and H. Saito. Registration of rgb and thermal point clouds generated by structure from motion.