

Vision Aided Navigation 2024 - Exercise 5

Amitai ovadia - 312244254

July 19, 2024

Amitai Ovadia 312244254

https://github.com/AmitaiOvadia/SLAMProject/blob/main/VAN_ex/code/ex5/Ex5.py

And

https://github.com/AmitaiOvadia/SLAMProject/blob/main/VAN_ex/code/utils/BundleAdjusment.py

Question 5.1

- For extrinsic camera matrix $[R|t]$, what would be the transformation from the camera's coordinate system to the global coordinate system?

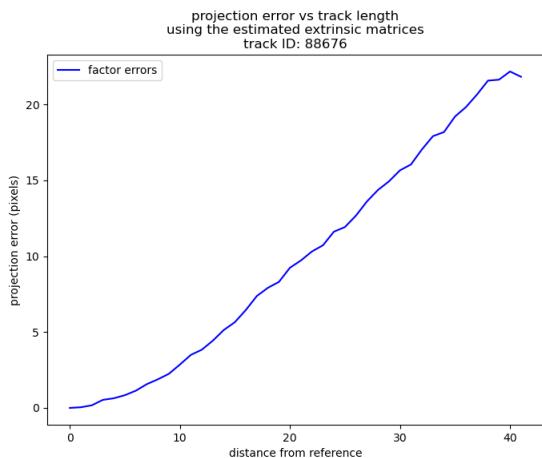
(Note the opposite direction here)

- give the transform in a 4×4 invertible matrix form $T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$ then this transform is from world coordinates to camera coordinates.

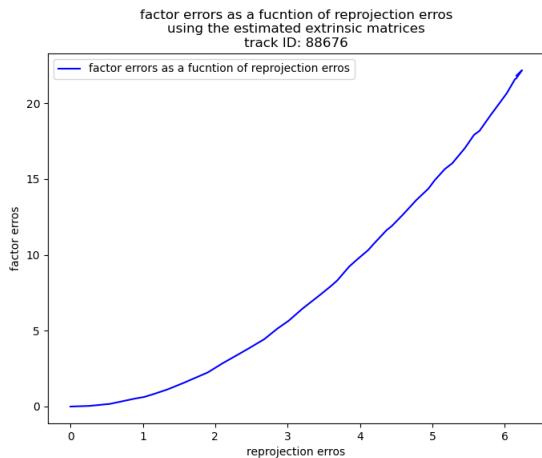
we want $T^{-1} = \begin{bmatrix} R^T & -R^T t \\ 0 & 1 \end{bmatrix}$ and so the new gtsam notation extrinsic camera matrix is $\begin{bmatrix} R^T & | -R^T t \end{bmatrix}$

- Now I created a factor for each frame projection.

Present a graph of the reprojection error size (L2 norm) over the track's images:



- I used a covariance matrix $\begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}$ and by that assumed that the distribution of the error is with $\sigma = 1$ over x, y
- here is the Factor error a function of the reprojection error:

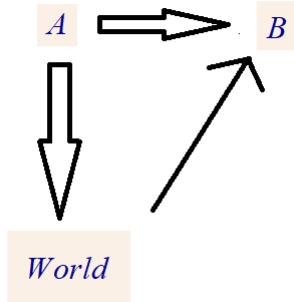


As you can see it looks like a parabula probably the factors are a sum of the squares of the reprojection errors under the mahalanubis norm. (suppose to look like $\frac{1}{2}x^2$ or x^2)

Question 5.2

Two cameras A, B have extrinsic camera matrices ($[R_A|t_A]$, $[R_B|t_B]$ respectively) in some common coordinate system. Express transformation $T_{A \rightarrow B}$ from the coordinates of camera A to the coordinates of camera B using R_A, R_B, t_A, t_B .

What would be the extrinsic camera matrix of B in the coordinate system of A ?



- $A \Rightarrow B$ is basically $A \Rightarrow World$ and then $World \Rightarrow B$

given some extrinsic camera matrix:

$$X = RX_{world} + t$$

$$\text{And so } X_{world} = R^T(X_{camera} - t)$$

Camera A :

$$X_A = R_AX_{world} + t_A$$

From camera A to world:

$$X_{world} = R_A^T(X_A - t_A)$$

And so from $World$ to B :

$$X_B = R_B X_{world} + t_B$$

So we can substitute from X_{world} to A

$$X_B = R_B R_A^T(X_A - t_A) + t_B = R_B R_A^T X_A - R_B R_A^T t_A + t_B$$

And so $T_{A \rightarrow B}$ is composed of $R = R_B R_A^T$ and $t = -R_B R_A^T t_A + t_B$

and can be written as $\begin{bmatrix} R_B R_A^T & | -R_B R_A^T t_A + t_B \end{bmatrix}$

Question 5.3

For this question I optimized bundle adjustment to one window of size 12

- Print the total factor graph error before and after the optimization.

```
total factor graph error * before * optimization: 13056.435060811167
total factor graph error * after * optimization: 1802.2334798158668
Backend TkAgg is interactive backend. Turning interactive mode on.
```

- How many factors are there in the graph?

```
146
num_of_factors = first_bundelon.get_factor_graph()
>= {NonlinearFactorGraph} size: 11051\n\nFactor 0: PriorFactor on c0\n prior mean:
```

- What is the average factor error before and after the optimization?

```
average factor graph error * before * optimization: 1.1814709131129462
average factor graph error * after * optimization: 0.1630832938029017
```

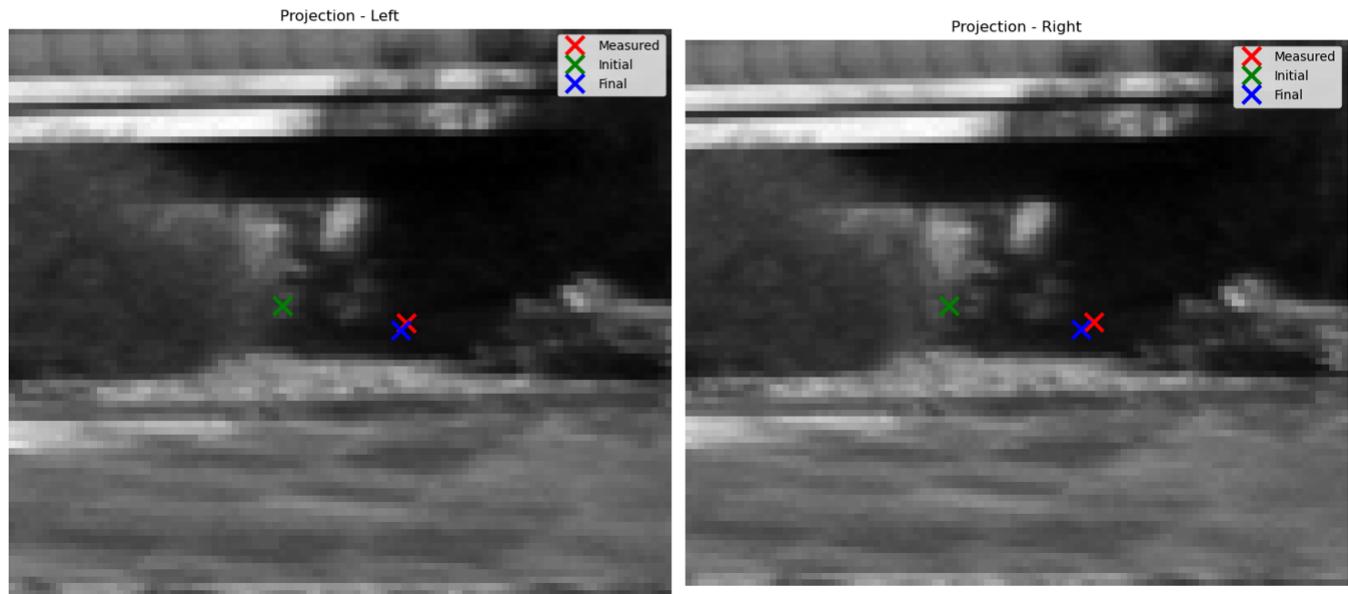
- for the factor with the biggest error:

```
initial factor error: 416.7200100577744
factor error after optimization: 2.732760437743385
```

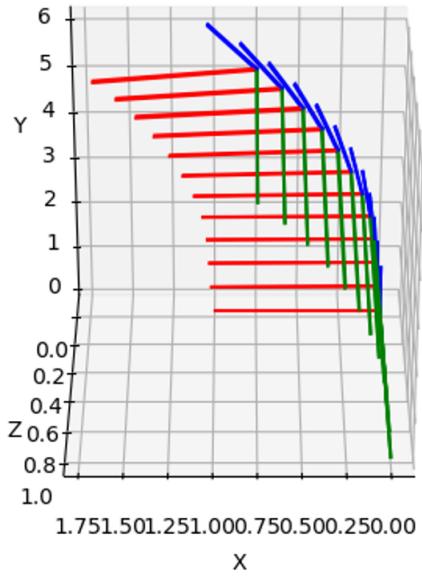
- In this plot:

Left and right projections on both images, along with the measurement, and their average reprojection errors.

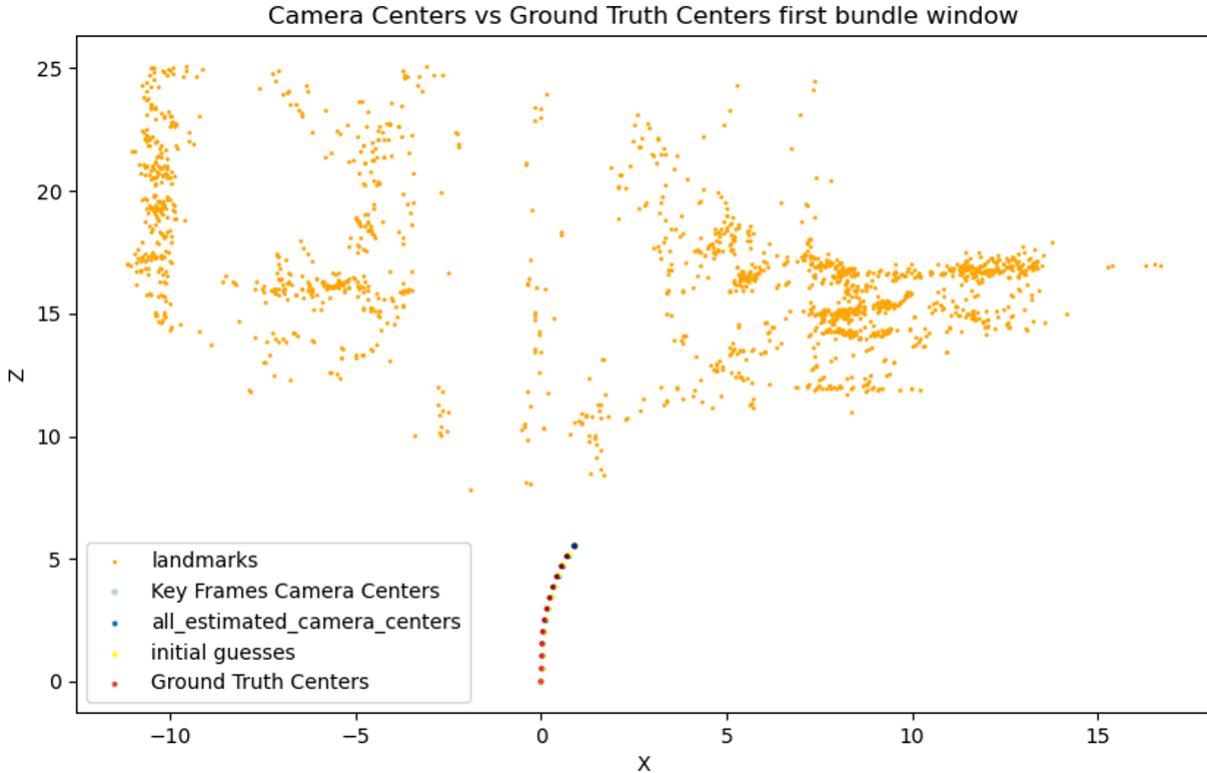
Stereo Points Tracking
Reprojection Error Initial: 18.45, Reprojection Error Final: 1.29



- This is the 3D gtsam based plot. it looks like there is only one trajectory plotted here but there are actually 2 very close together.
- Also the sigmas are not shown here as the `gtsam.utils.plot.plot_trajectory` didn't plot them.



- Trajectory of the first bundle, plus the landmarks, as viewed from above



Question 5.4

- I took the first frame of last bundle in the bundle adjustment window, after optimization.

This is the camera pose in gtsam format:

```
camera pose is: R: [
  1, 5.25586236e-12, 3.16036896e-13;
 -5.25579147e-12, 1, 4.41383662e-11;
 -3.16038779e-13, -4.41383641e-11, 1
]
t: 2.17814153e-11 3.15848195e-13 -2.43645289e-08

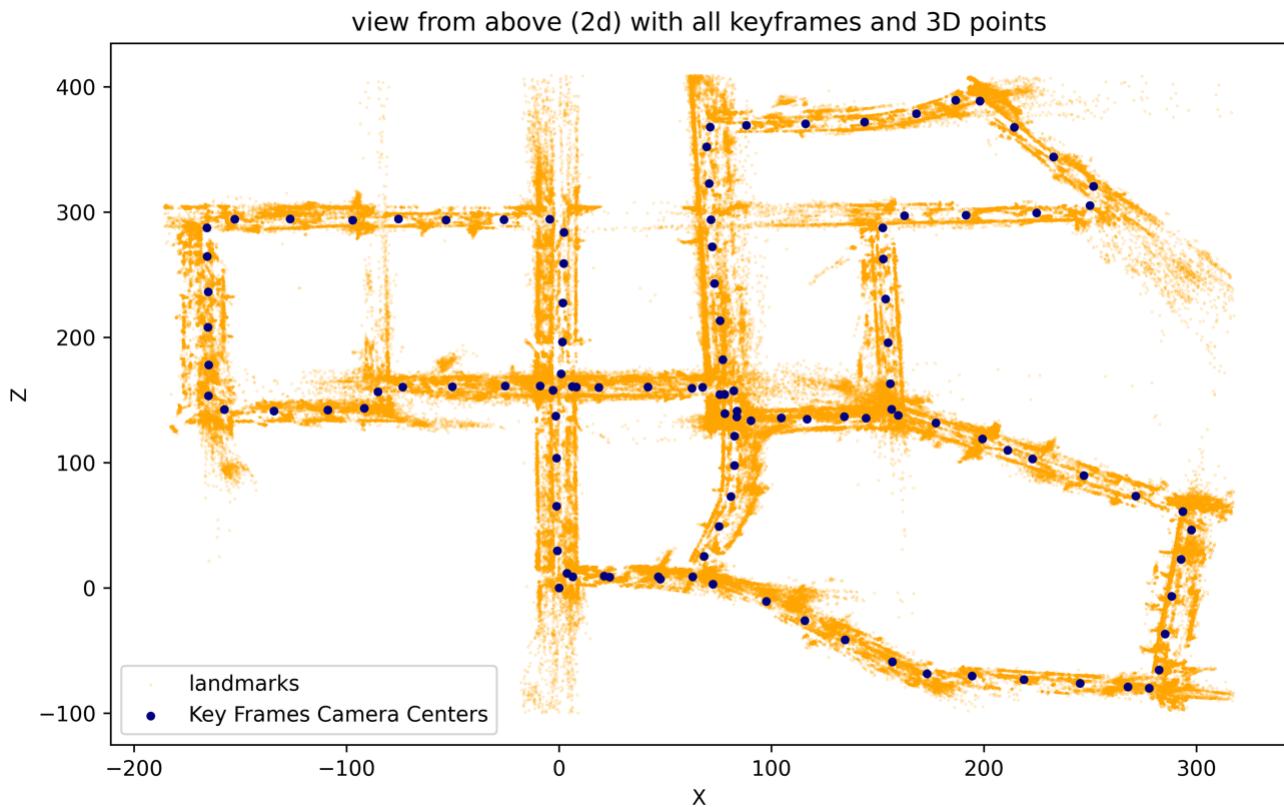
factor error is: 3.001e-16
```

- The location stayed at the prior $[I|0]$ with a small numerical error.

Moreover the factor error as displayed in the bottom, is also extremely low.

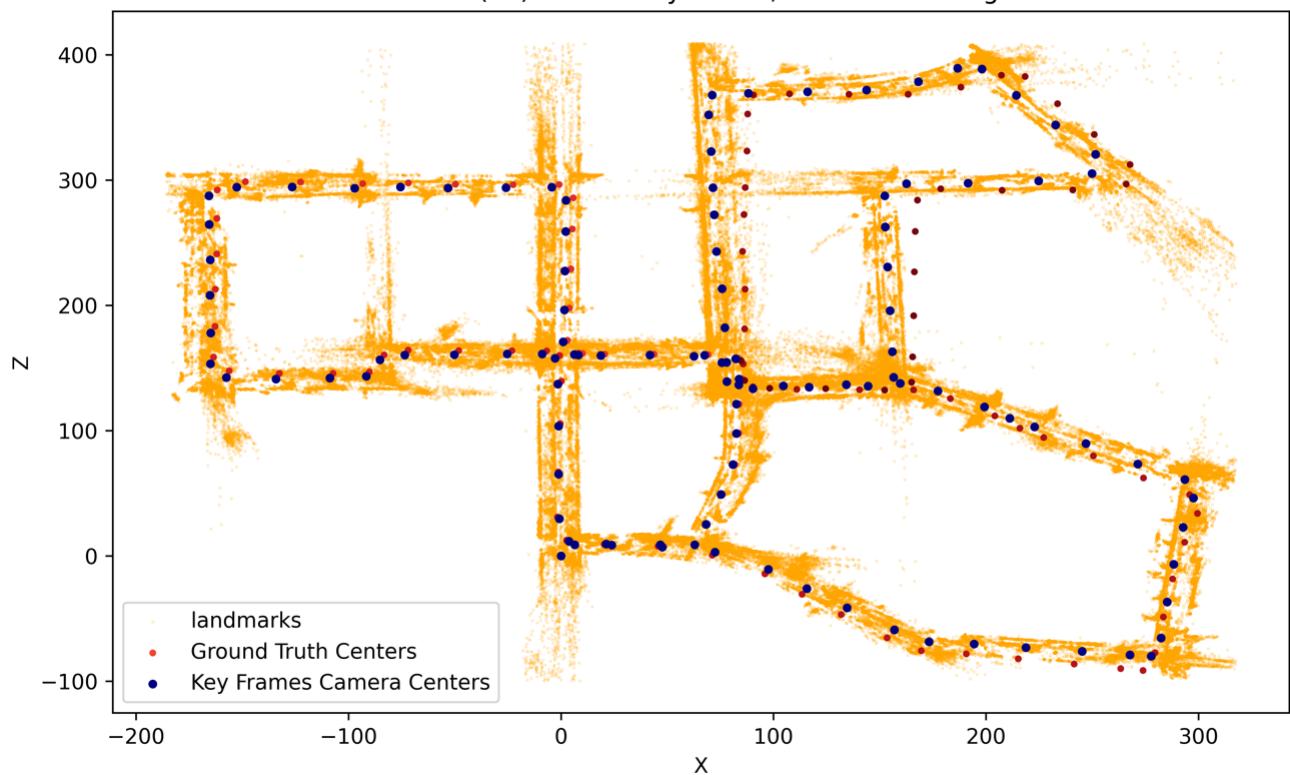
the reason is that the optimization left the prior untouched (it was fed as the prior)

- A view from above (2d) of the scene, with all keyframes and 3D points

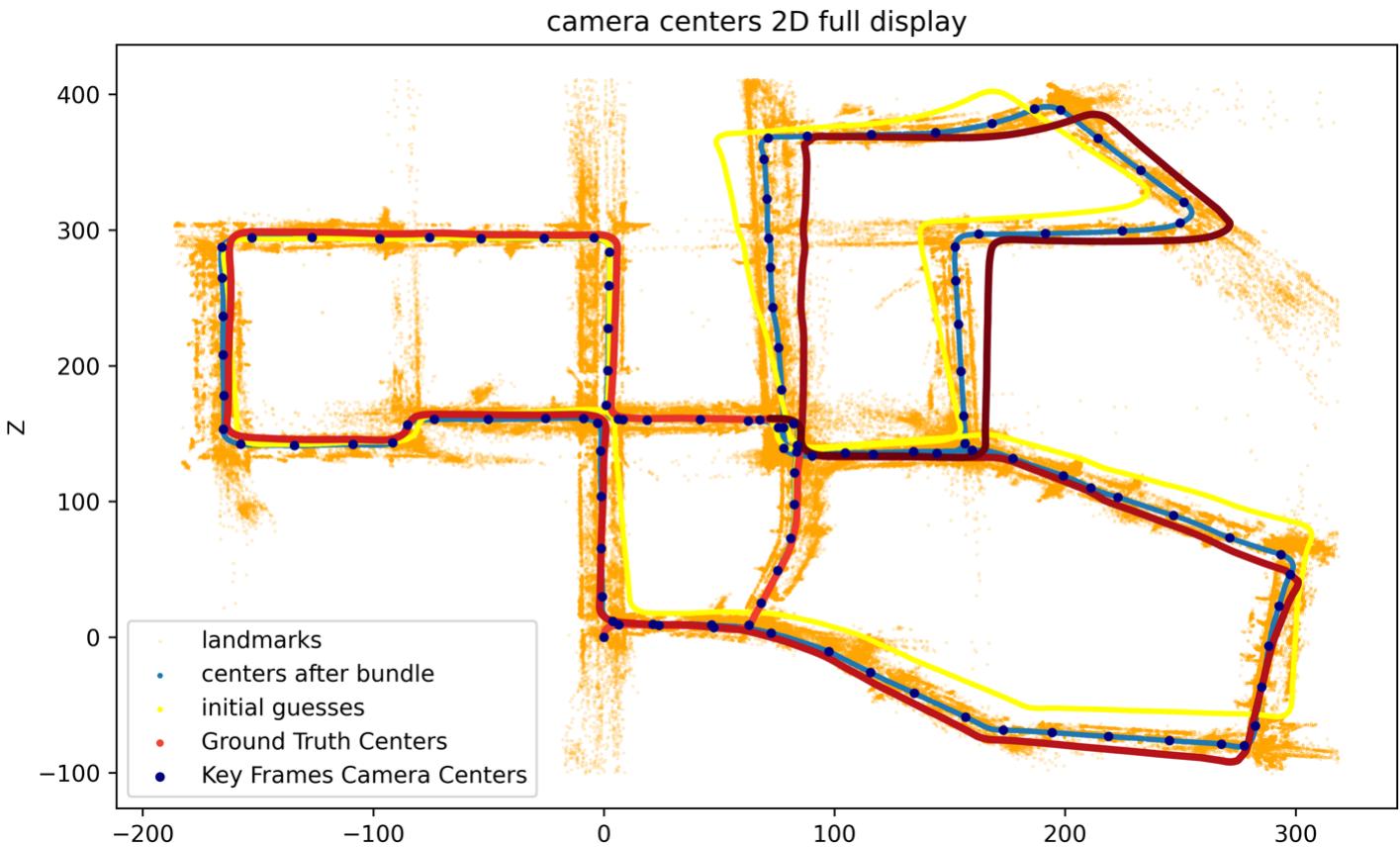


- estimated keyframes with the Ground Truth poses of the keyframes

view from above (2d) with all keyframes, landmarks and ground truth



- Display of all the frames including initial guesses



- This is the localization error from the ground truth in L2:

