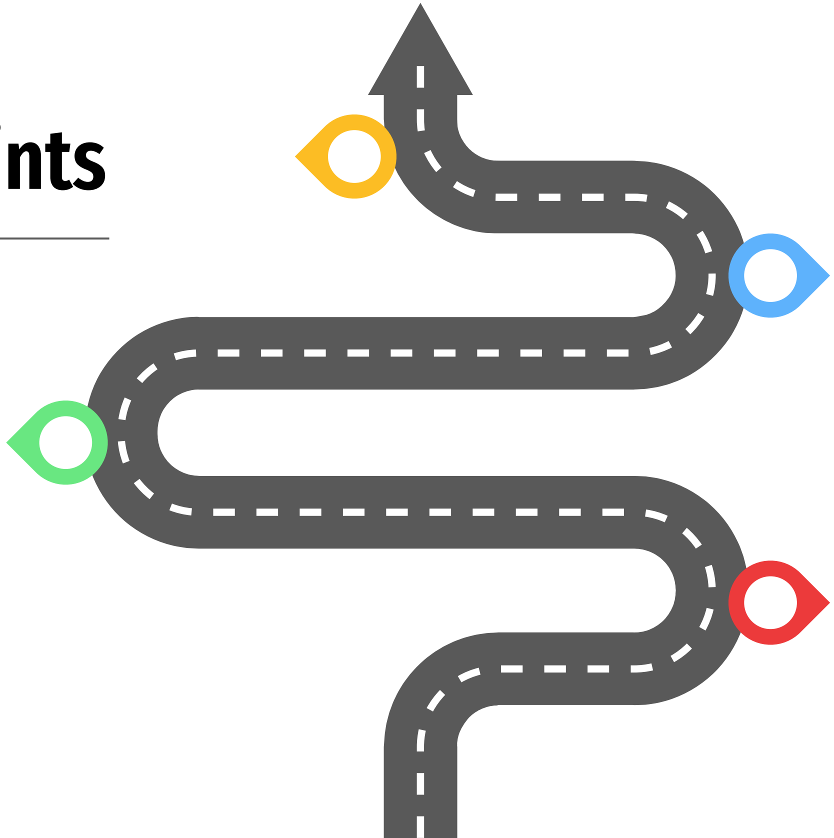
3D Reconstruction from Road Marker Feature Points

Team 6　你就繼續

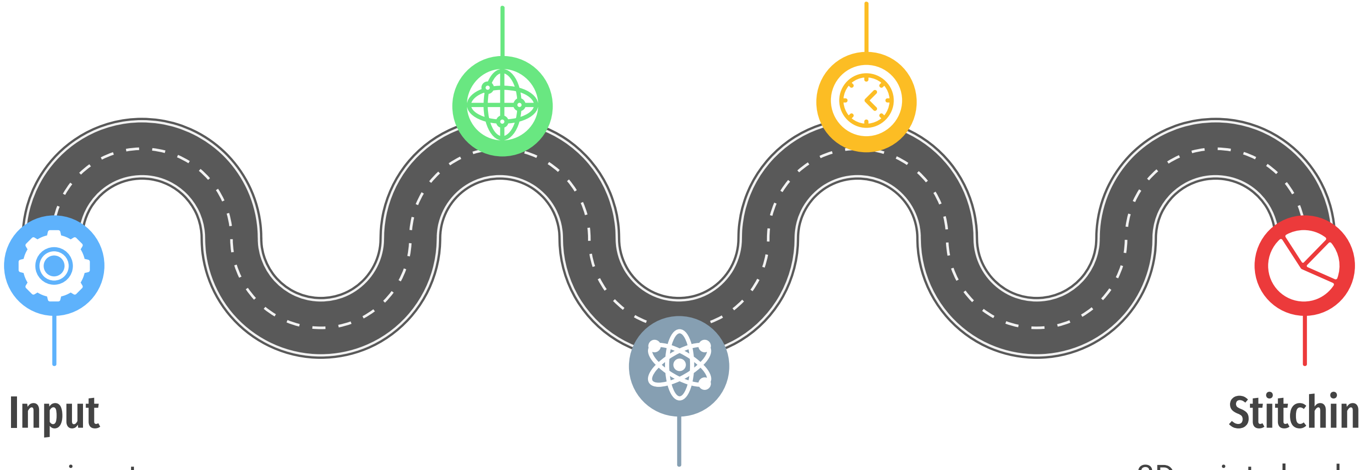
D11921B09	徐子程
B08901081	黃靖元
B08901057	吳瑋倫
B08901046	詹侑昕



Outline

- Literature review
- Method
  - Road marker identification
  - Point cloud reconstruction
- Results
- Conclusion

Pipeline



Input

Image input

Segmentation

Using 3rd party segmentation tool

Find Corners

OpenCV

Reconstruction

3D point cloud by pinhole model

Stitching

3D point cloud stitching

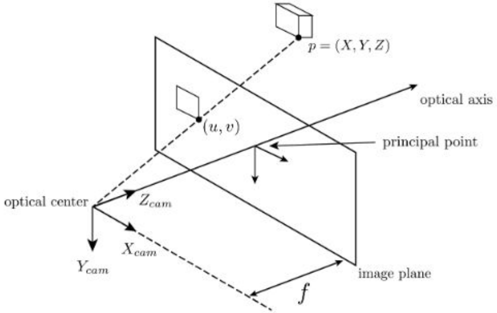
Literature review - Pinhole model

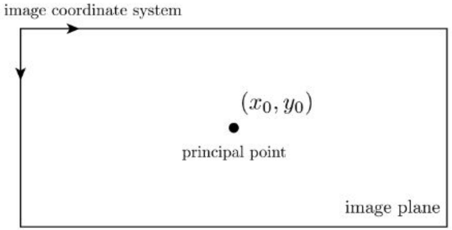
$$\begin{bmatrix} u' \\ v' \\ w \end{bmatrix} = \begin{bmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

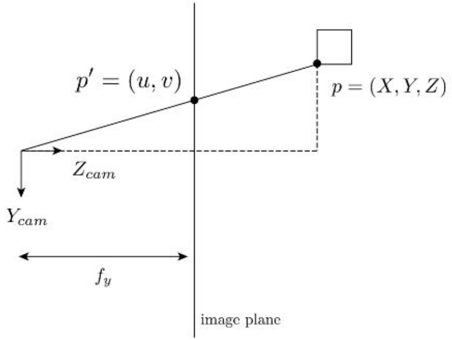
f_x, f_y : Focal lengths

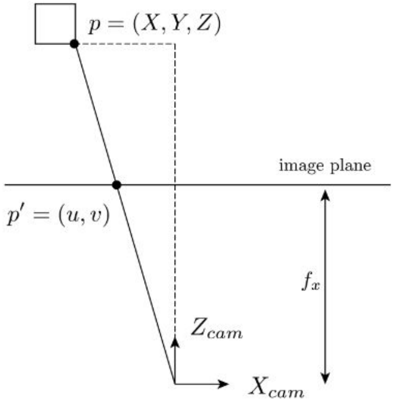
x_0, y_0 : Center of the image

X, Y, Z : 3D space coordinates

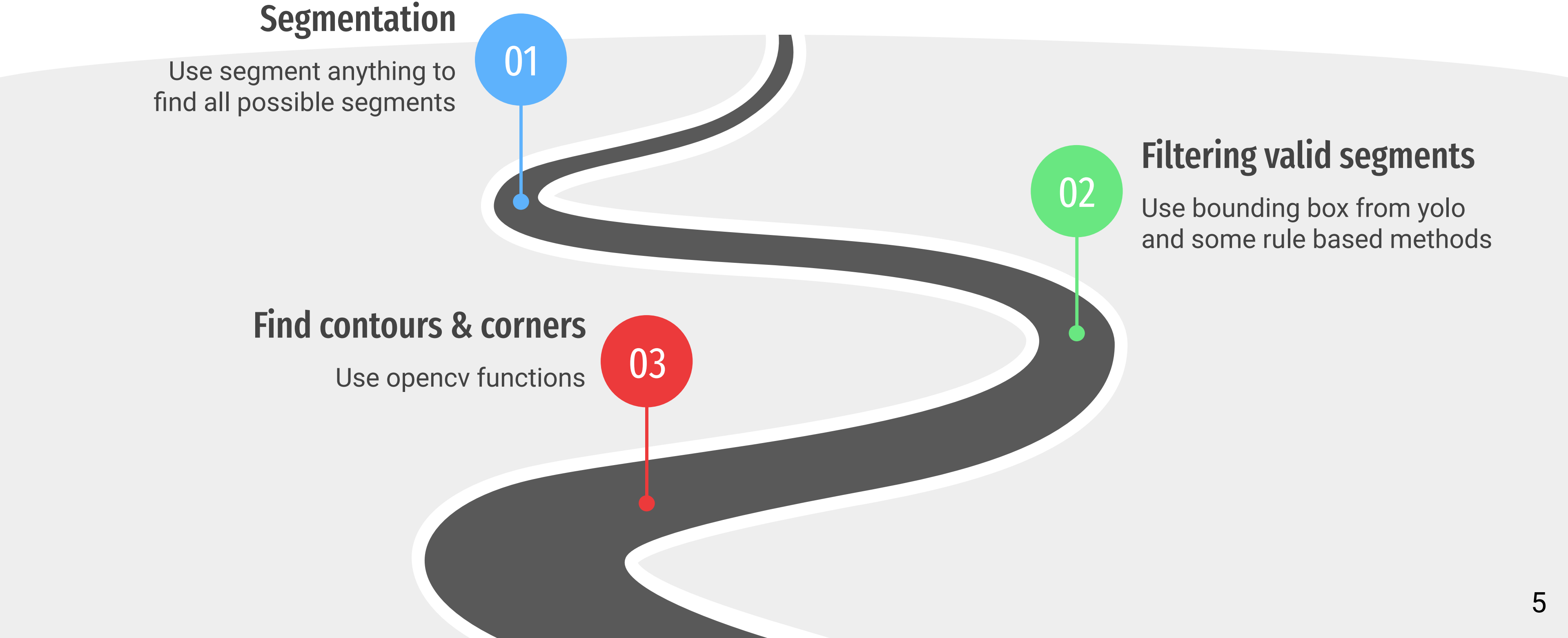








Road marker identification



Segmentation

Use segment anything to find all possible segments

Filtering valid segments

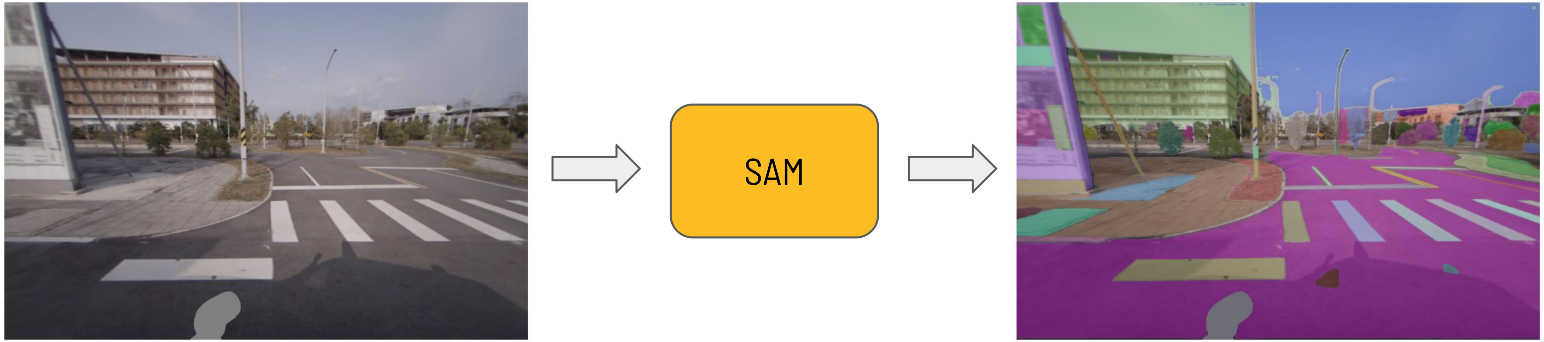
Use bounding box from yolo and some rule based methods

Find contours & corners

Use opencv functions

Segmentation

Using segment anything from Meta AI to extract all possible segments

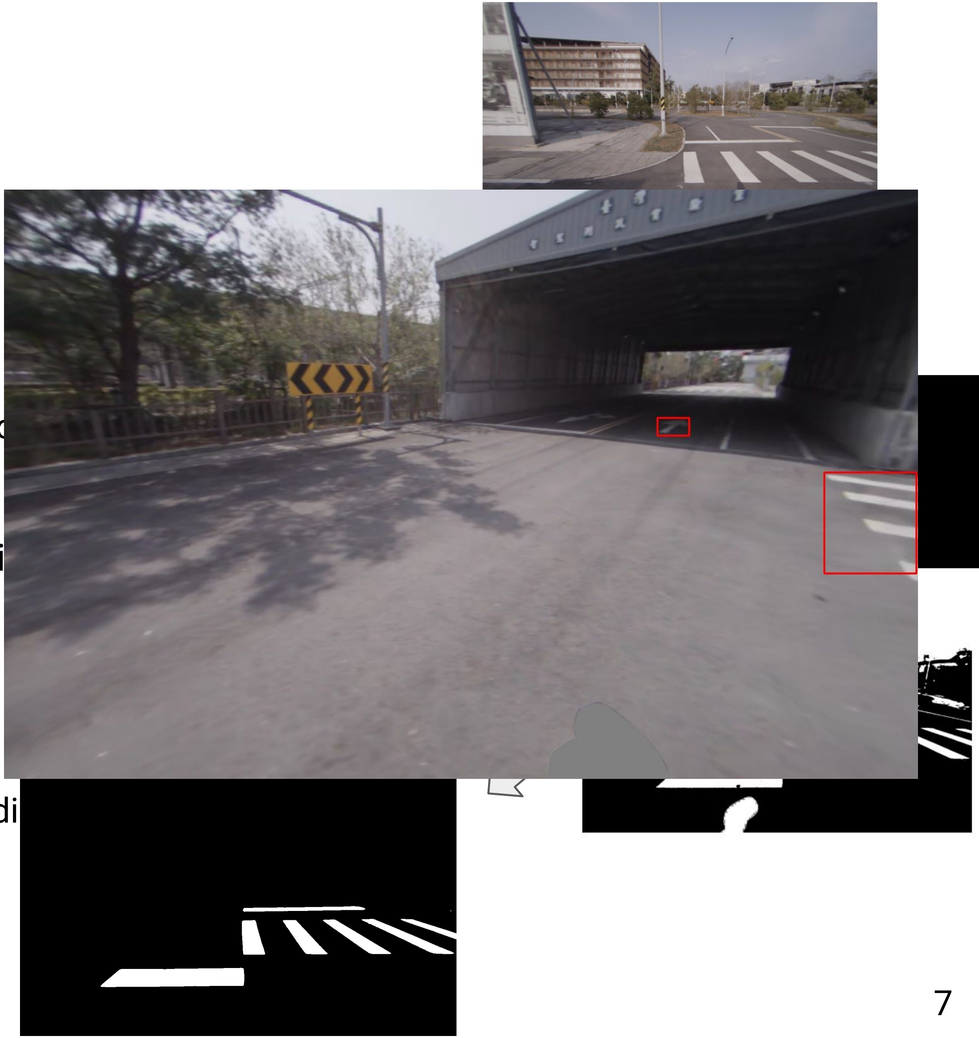


	seq1	seq2	seq3	test1	test2
Avg segments	81	72	77	89	76

Filtering valid segments

Steps:

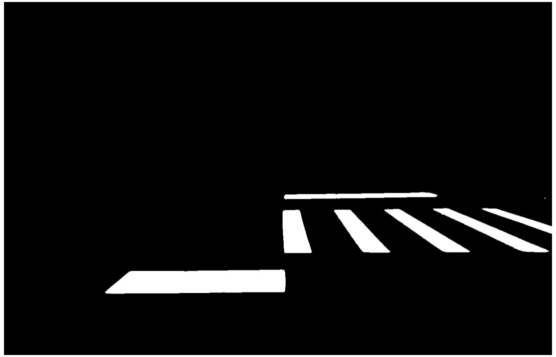
1. Read bounding box and camera mask
   a. If bounding box probability < thresh
2. Threshold raw image
3. For every segments, if following conditi
   a. It is in camera mask
   b. It is not in any bounding box
   c. If the segments in (2) is black and the percentage of segment in boundi



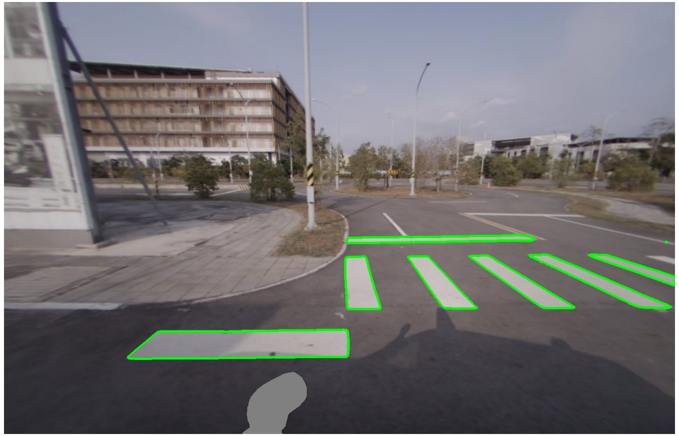
Find contours & corners

Apply opencv functions

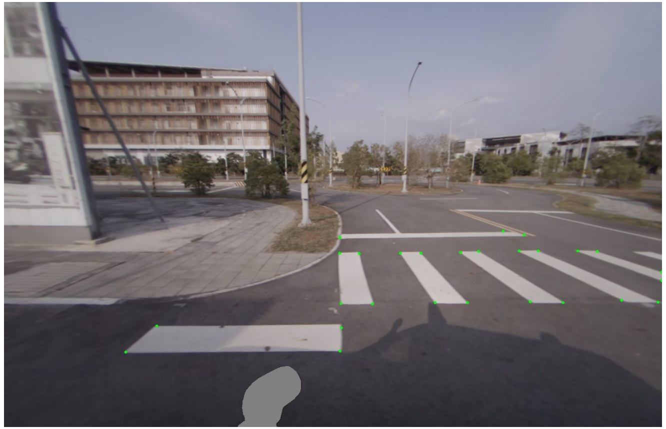
- findContours
- approxPolyDP



findContours

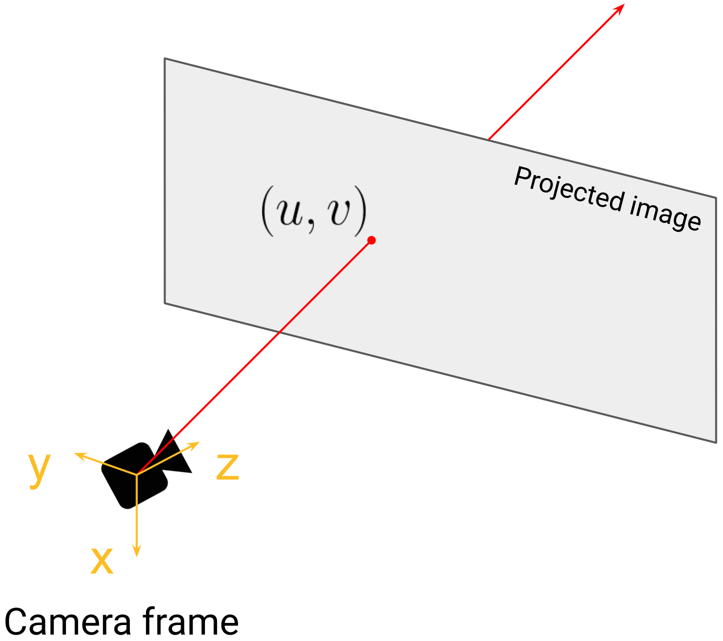


approxPolyDP

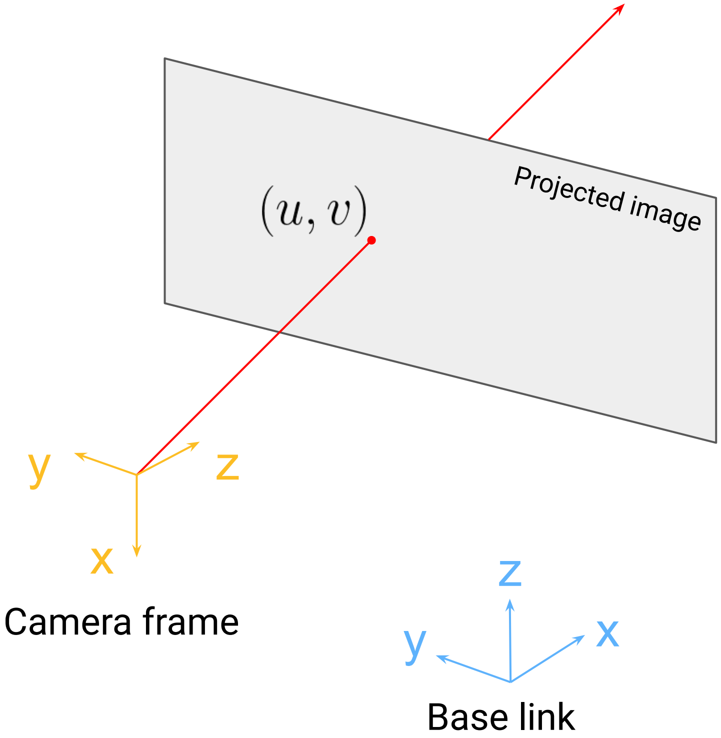


Point cloud reconstruction

From pinhole model

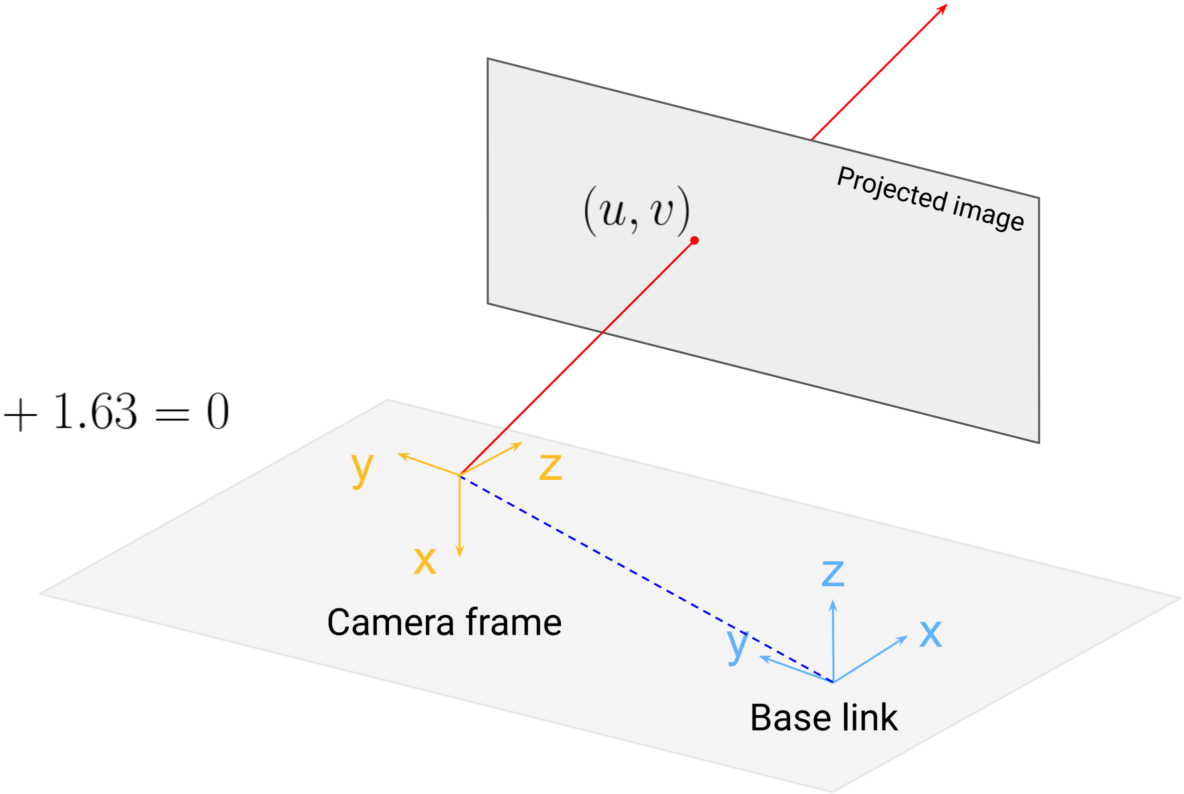


Point cloud reconstruction



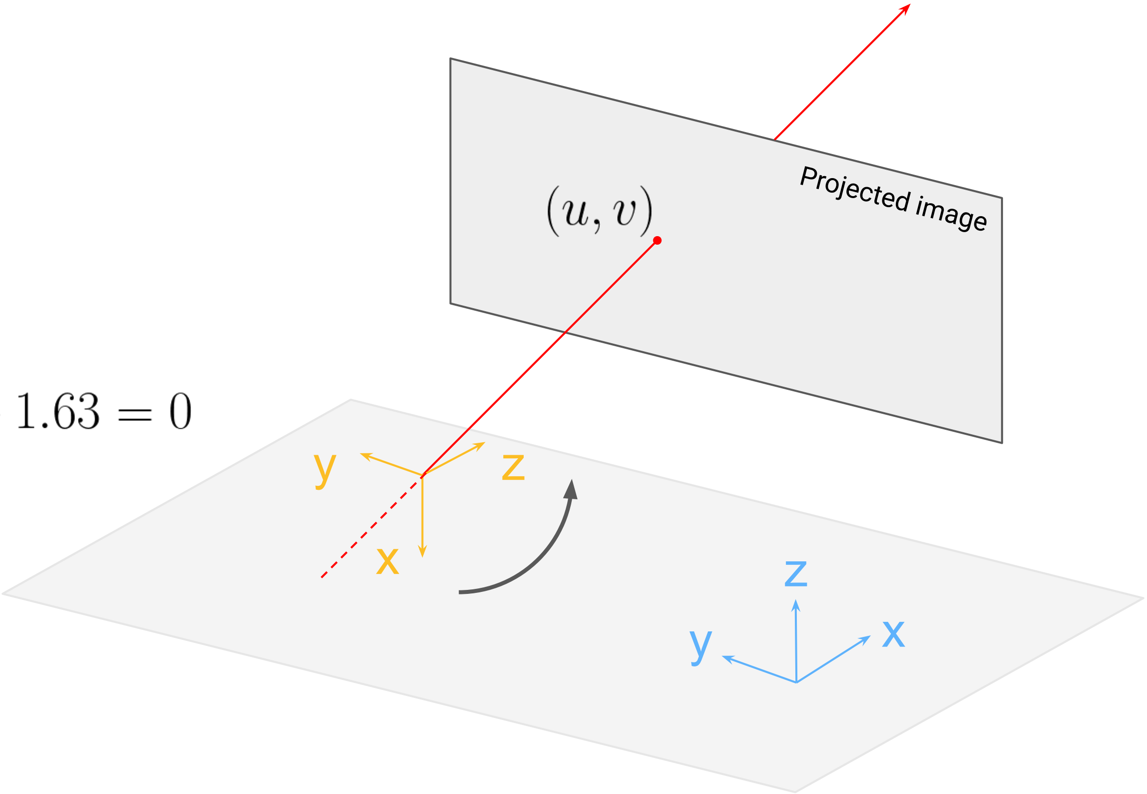
Point cloud reconstruction

$$E : z + 1.63 = 0$$



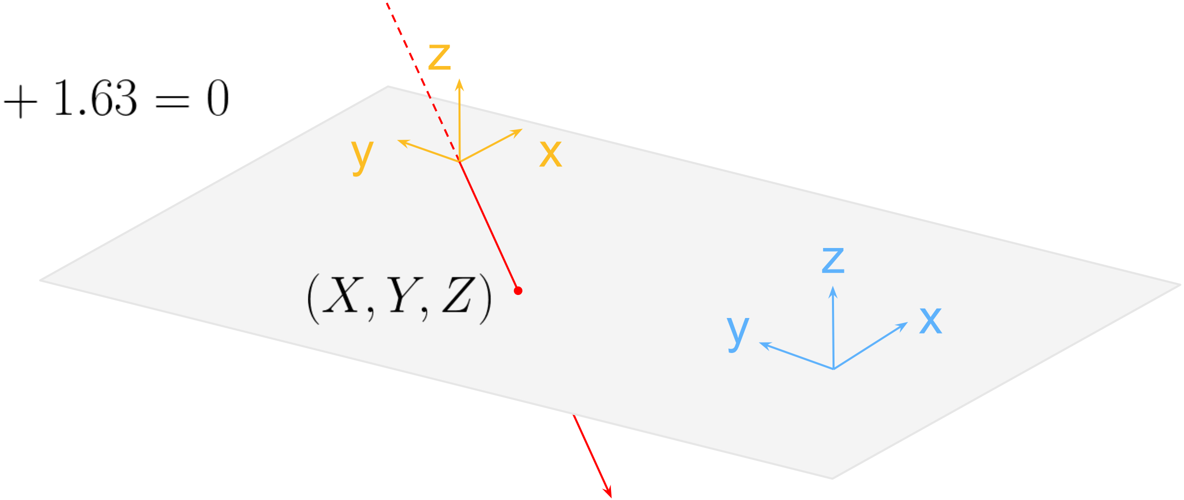
Point cloud reconstruction

$$E : z + 1.63 = 0$$



Point cloud reconstruction

$$E: z + 1.63 = 0$$

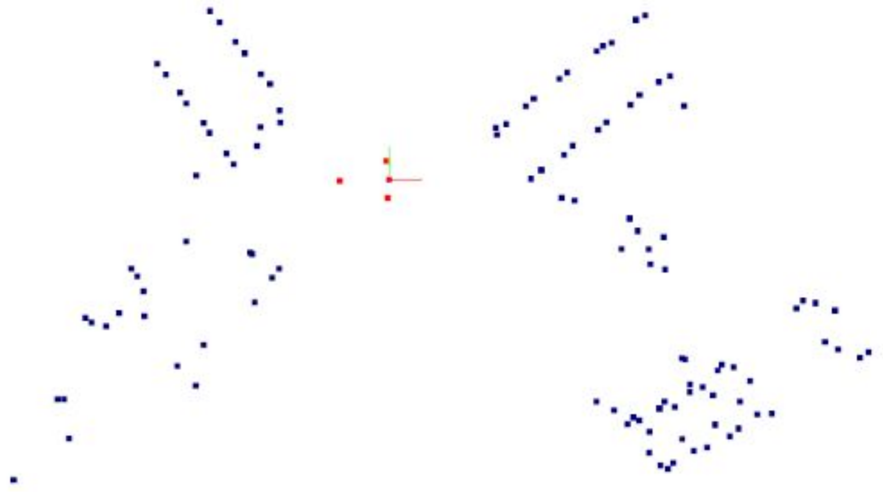


Point cloud blending from multiple cameras

1. Sort the frames by timeline
2. Take adjacent 4 frames’ point clouds
3. Transform 4 point clouds to base_link and mix them to single point cloud set
4. Apply this mixed point cloud set to four frames

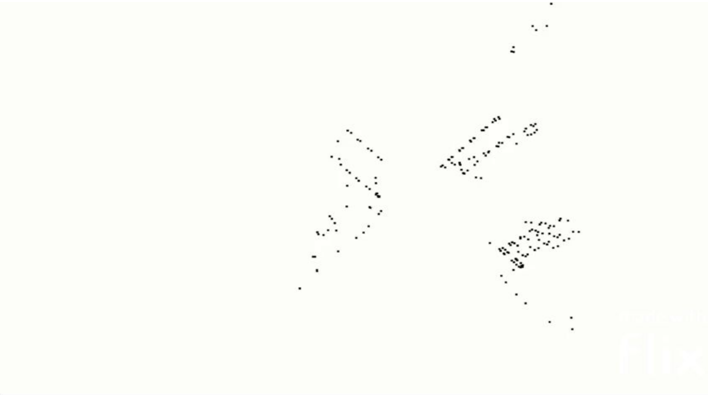
Results

Seq1:

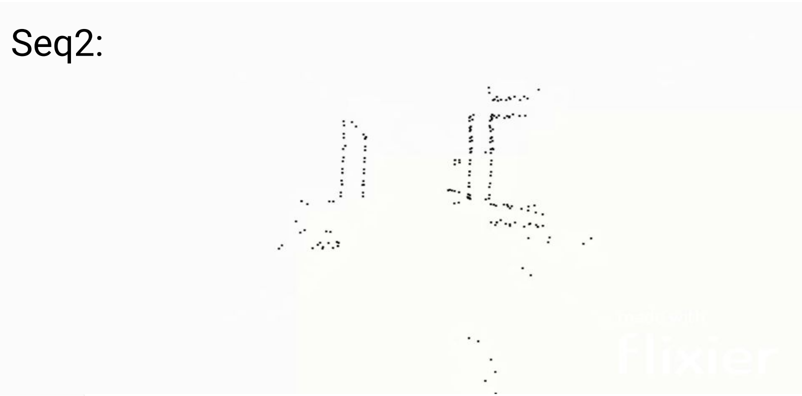


Results - Point cloud blending

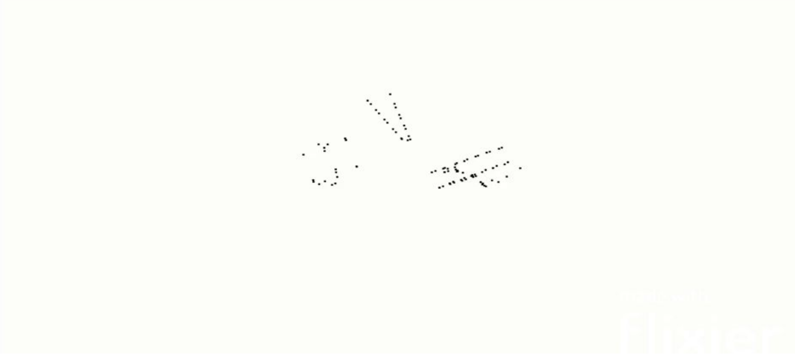
Seq1:



Seq2:

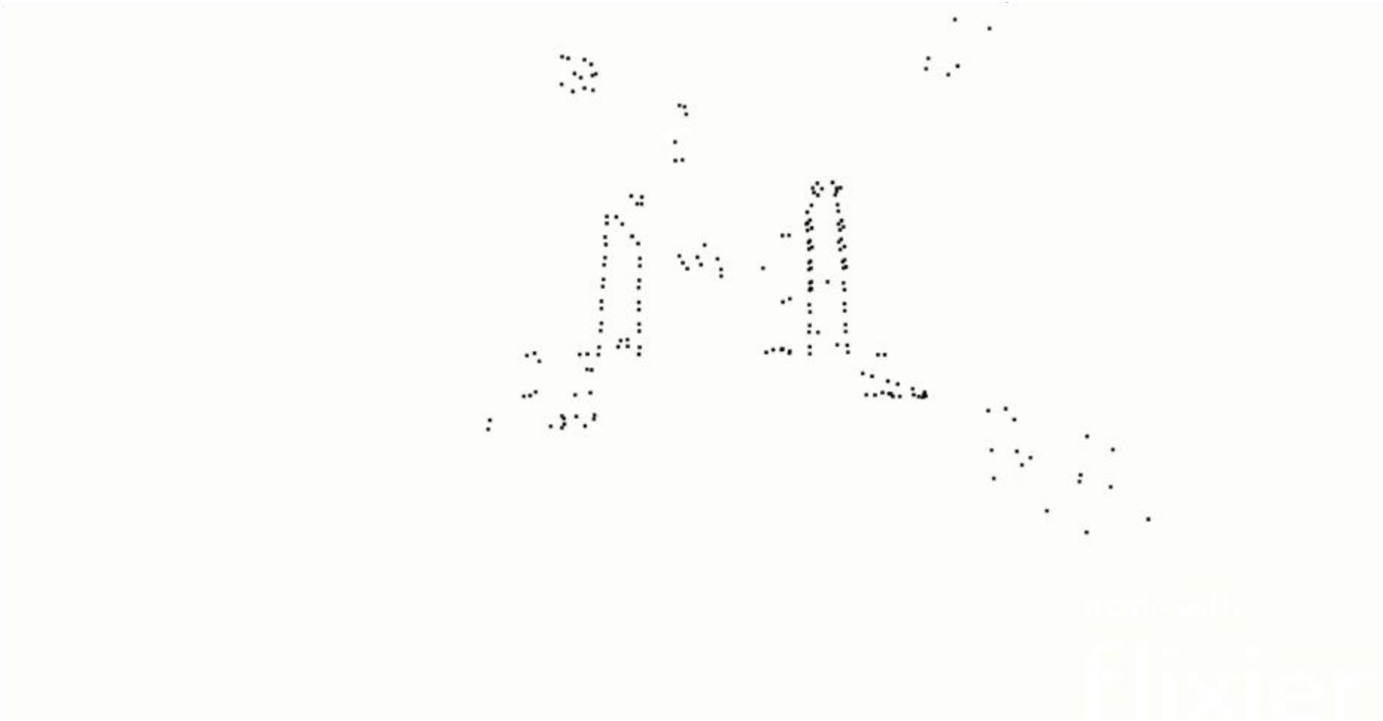


Seq3:



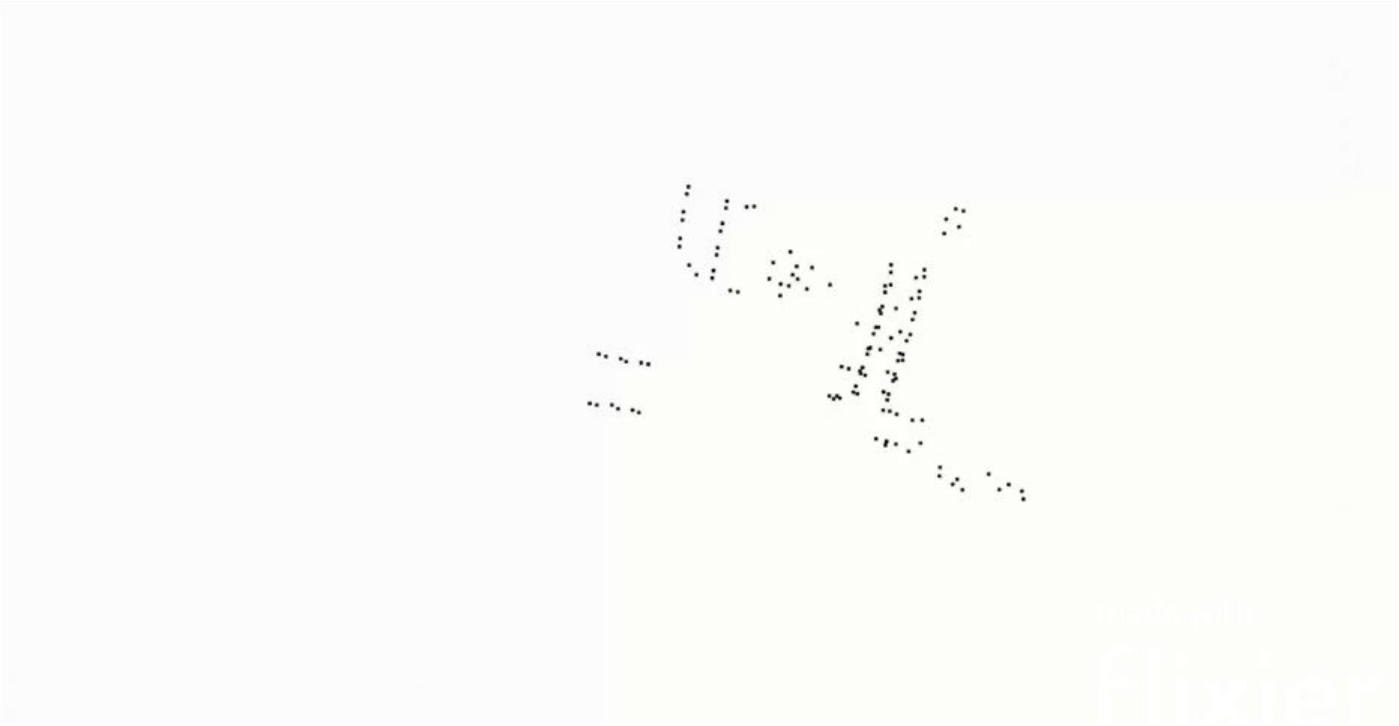
Results - Point cloud blending

Test1:

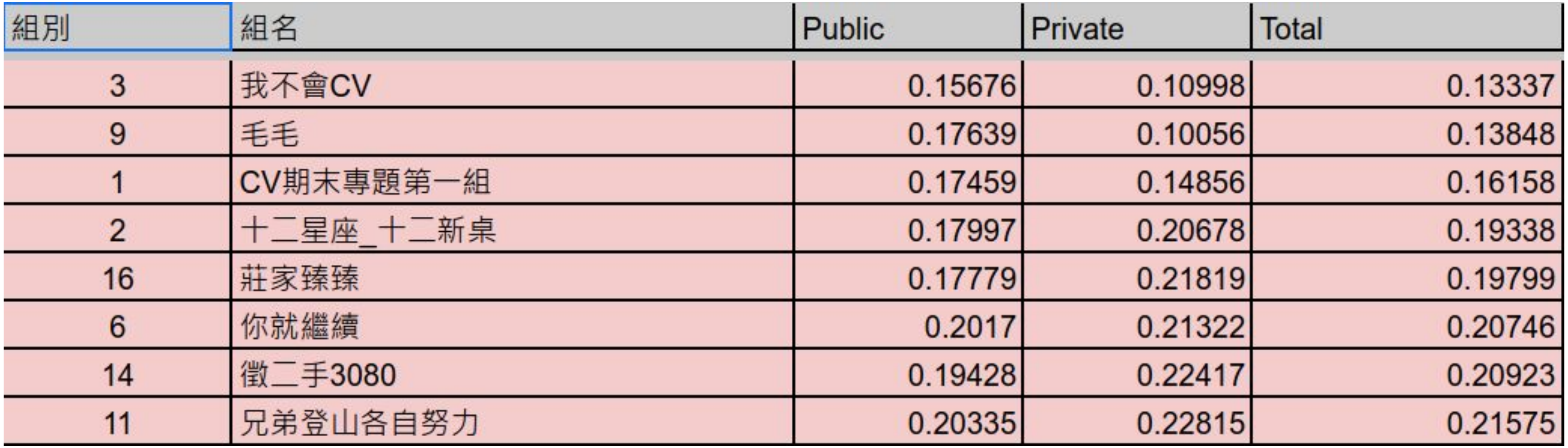


Results - Point cloud blending

Test2:



Results - leaderboard



組別	組名	Public	Private	Total
3	我不會CV	0.15676	0.10998	0.13337
9	毛毛	0.17639	0.10056	0.13848
1	CV期末專題第一組	0.17459	0.14856	0.16158
2	十二星座_十二新桌	0.17997	0.20678	0.19338
16	莊家臻臻	0.17779	0.21819	0.19799
6	你就繼續	0.2017	0.21322	0.20746
14	徵二手3080	0.19428	0.22417	0.20923
11	兄弟登山各自努力	0.20335	0.22815	0.21575

Conclusion

- Successfully generate point clouds by pinhole model
- Blending images to get more complete point clouds
- Threshold of ICP will influence the performance

Future work

- Adjust the corner points to avoid the misinterpretation on the boundaries of the road mark
- Ensemble on the same corner points given by different cameras
- Collect more information other than those given in the bounding box(for example, the double yellow lines)

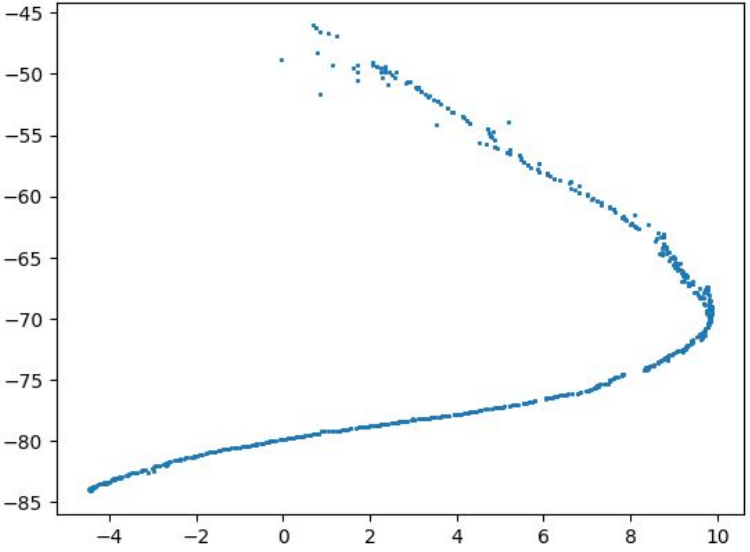
References

[1] “Lab 6: Computer Vision”, EECS C106A/206A | Introduction to Robotics, Fall 2021
https://pages.github.berkeley.edu/EECS-106/fa21-site/assets/labs/Lab_6__Computer_Vision.pdf
[2] Kirillov, Alexander, et al. "Segment Anything." *ArXiv*, 2023, /abs/2304.02643. Accessed 15 Jun. 2023.

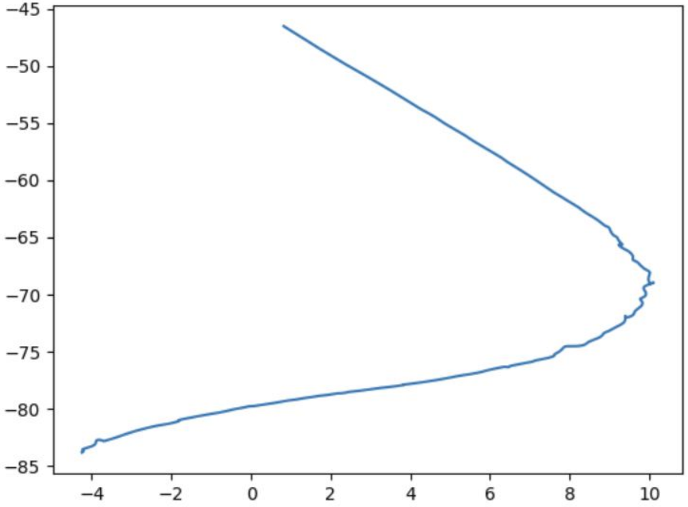
Thank you!

Appendix - Post processing

Method: Trajectory smoothing



−45
−50
−55
−60
−65
−70
−75
−80
−85
−4
−2
0
2
4
6
8
10



−45
−50
−55
−60
−65
−70
−75
−80
−85
−4
−2
0
2
4
6
8
10