2023 spring CV final project - report

Group 13
r11528025 劉容綺
b05505006 林容與
r11942076 謝昌諭
b09705024 劉智心
r10h41003 游筑鈞

I. Abstract

In this report, we proposed a corner-detection based 3D reconstruction algorithm relating to the provided driving car scene dataset. Our proposed method utilizes basic morphological processing and simple mask filtering, realizing preliminary automatic traffic signal segmentation. And, appling corner detection algorithm on the segmentation result to get feature point of driving scene. Besides, we evaluate our method by three provided sequence data, doing the ablation study to determine optimal hyper parameters.

II. Hypothesis & System Architecture

In this section, we will present the basic idea and hypothesis of our proposed method. We separate it into three major parts, traffic signals detection, feature points acquiring and point cloud refinement. At the end of this paragraph, we will show the overall architecture of our reconstruction system.

2.1. Traffic signals detection

Our motivation to detect traffic signals from raw images directly is from the observation that almost every bounding box is located in the down half of the image, where the most frequent color is gray/black of the asphalt. At the same time, zebra cross, stopline, arrow, etc are in relatively bright color. Fig.1 shows the 3 color channel's pixel value distribution across all bounding patches of the training set. The peak that happened in 128 comes from the mask of car body. And the left peak indicates the relatively dark road and right means traffic signal.

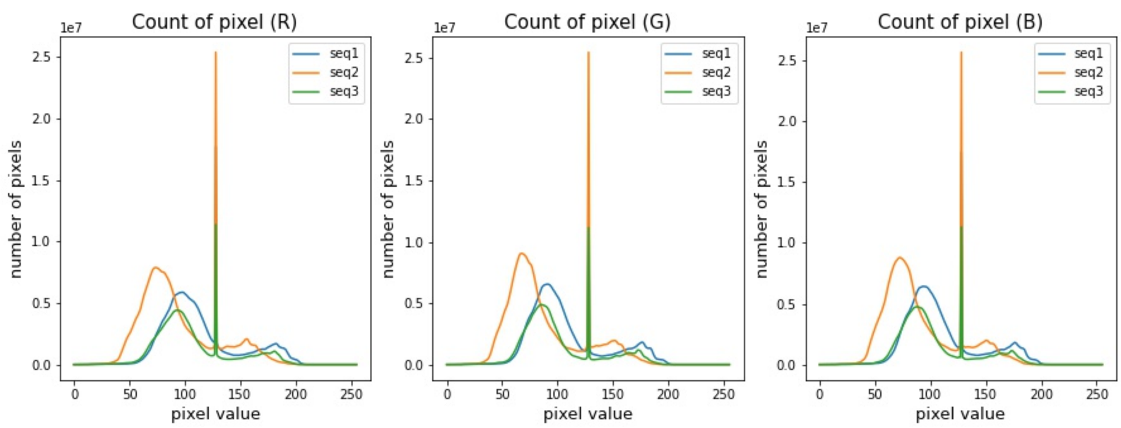

Fig.1 Distribution of 3 color channel's pixel value across three sequence in training set.

According to above result, we hypothesis that we can filter road region out by thresholding raw images. However, we also found that result of three sequence are similar but not totally match. We think it is because the time when the image was took. All three sequences were took in about 2023/04, but sequence 2 is shooted at about 17:00. Compare to sequence 1 (about 14:00) and sequence 3 (about 15:00). The sun incidence angle resulted in different average intensity. Therefore, we need to implement some adaptive filtering method to deal with this problem. Furthermore, we also add some more strict mask to focus on our interesting region.

2.2. Feature points acquiring

Ideally, the detection result after the former step should be a binary image with traffic signal to be 1 and the others to be 0. While, in reality, there might be some noise, like white building, sky regions which can also pass the threshold above. Thus, two key problems in this stage are (1)eliminating error detection regions and (2)extracting feature points. For the first problem, our method is using morphological opening to illminate small regions. For the second problem, we choose Shi tomasi ((Jianbo Shi et al. 1994)), for we think it is effective enough and easy to implement.

2.3. Point cloud refinement

In stage 2, we got feature points in the image. Then, we convert it into the point cloud in world coordinates by intrinsic and extrinsic matrix. In this stage, we want to refine the point cloud result. We use the distance between world point and camera (that is the original point) to filter points with high probability to be noise. Fig.2 shows an example of distance based filtering. In Fig.2 blue points are those points whose horizontal distances with the camera are all smaller than 15 meters and the red ones are bigger. Distant points are more likely to be the noise point. For example, points on buildings have bigger height. Thus, after being projected into the ground, they have larger horizontal distance.

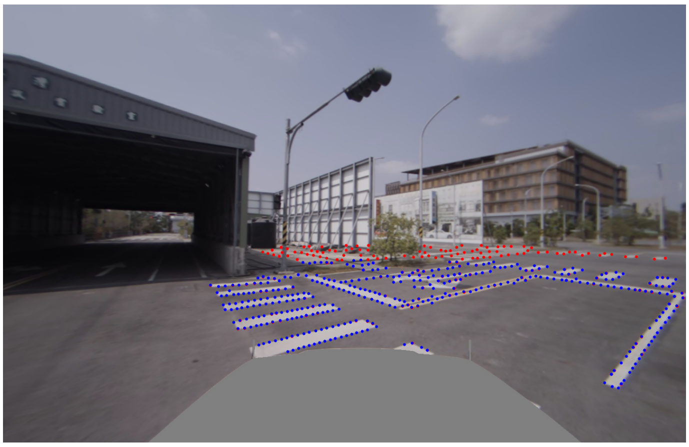

Fig.2 Example result of distance based filtering

2.4. Overall architecture

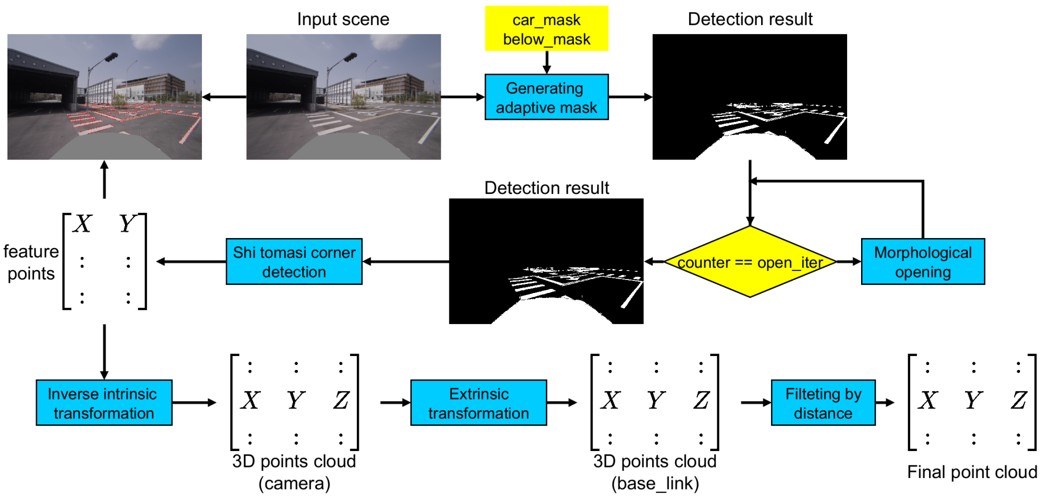

Fig.3 Overall architexture of proposed method

Fig.3 shows the architecture of our proposed method. At first step, we use two masks and input image to compute the adaptive mask (that is the traffic signal detection result), Then, we do some morphological opening to remove minor error segmentation regions, generating a more precise result. In the corner detection stage, we employ the Shi-Tomasi detection algorithm to find out feature points. Feature points would be converted into point cloud in base_link coordinates. Last, using distance based filtering to refine results.

III. Method

In this section, we will give a detailed explanation of proposed method.

3.1. Strict mask - car_mask, below_mask

The provided masks contain only regions of the car body. Using them directly will result in many noise points on the edge of regions. Furthermore, feature points above the horizon won't fall in our interested region after correcting them to ground. Since they have a larger height value compared to the camera, the projection result will be located on the back of the camera, which is unreasonable.

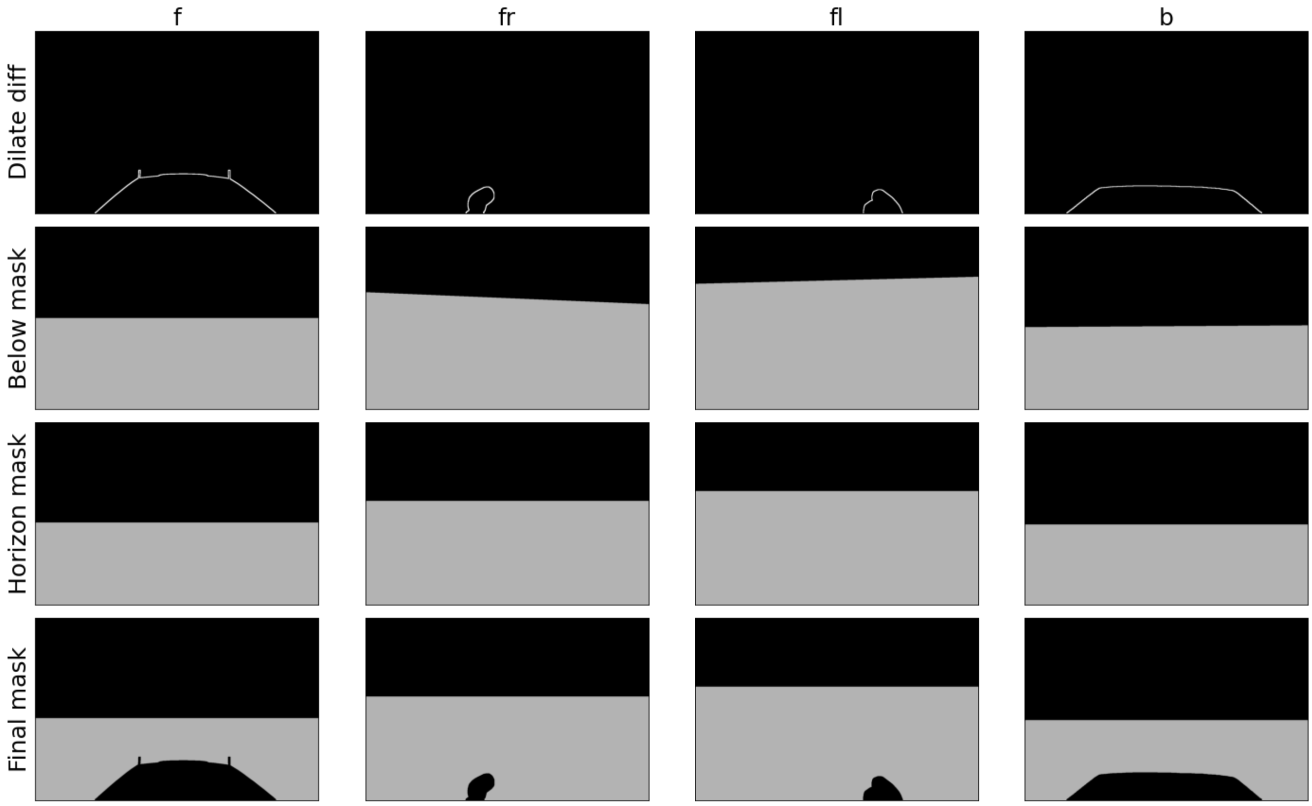

Fig.4 Masks of each camera.

First, we handled the edge problem by slightly dilating the original mask. We show the difference between original and dilated masks. Feature points which are located on the edge will be removed.

Second, we handled the wrong projection problem with adding Below mask and Horizon mask. The former ones are calculated from camera coordinate transformation. We labeled each pixel as 1 if its transformation result had a height value smaller than f-camera, 0 otherwise. So, the fr's and fl's results are slightly tilted. The latter ones are labeled by us. We observed the pictures from each sequence and determined the approximate horizon.

The final masks depict the interested region in each camera. When we did the traffic signal detection, only result falled in those regions would be kept.

3.2. Corner detection

We've applied Shi-Tomasi, which is a corner detection algorithm and a modification of the Harris Corner Detector. It scans an image, looking for corners by examining significant changes in all directions. It identifies corners by shifting a small window and observing large changes in appearance. The algorithm selects the N strongest corners based on a quality level threshold, rejecting corners below a specified minimum quality. This approach, being a modification of the Harris Corner Detector, provides reliable feature points representing corners in the image.

3.3. Distance based filtering

As we mentioned in Sec.2, to decrease noise points, we only retained points close to the camera. Feature points from each camera were transformed to base_link dirst. Projecting all points into ideal ground to make sure all points are coplanar. According to the projection result's horizontal distance to filtering. In our implementation, we use Euclidean distance. We also tried to use Manhattan distance, but the performance didn't have much difference.

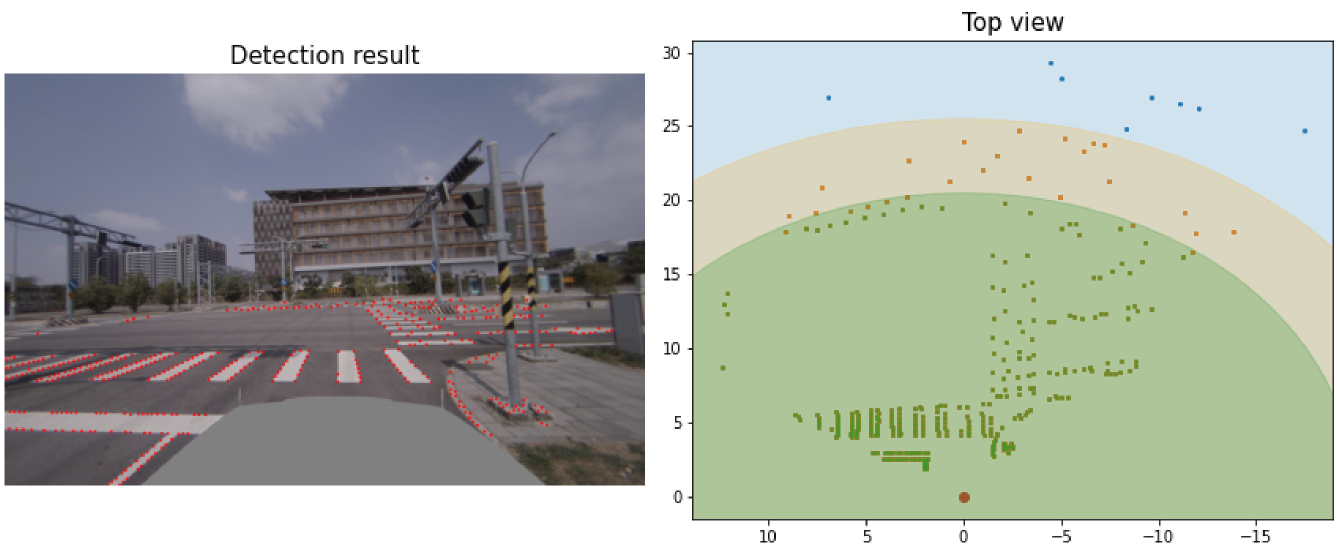

Fig.5 Example of distance based filtering. (a)Raw image with detected eature points. (b)The top view od corresponding point clouds. (For better visualization, the point density has been scaled up.)

Fig.5 shows a case of distance based filtering. In Fig.5(b), green, orange, blue circle represent regions which have Euclidean distance smaller than 20, 25 and 999 meters respectively. In this case, the target features are mainly the zebra cross on the square area in front of car. We can observe that there are some points on road and pole. Because of the low angle of depression, their corresponding point clouds are very distant, mainly falling on blue region.

IV. Results and Ablation Study

In this section, we evaluate our method from two sides, quantitative and qualitative. In the qualitative part, we show the ablation study result and give the final best choice of hyper parameter. For the quantitative part, we present four corner detection results from different camera but close time stamp. From that, we can intuitively observe the effect of augmented point clouds.

We use abbreviations to refer to all hyper parameters in the following ablation study table. OpenIter means the iteration number of morphological openings. FiltDist is the distance threshold when doing the cloud point refinement. qualityLevel and minDistance are the built in parameters of `cv2.goodFeaturesToTrack`, characterizing the minimal accepted quality and minimum possible distance of image corners respectively. intensity_ratio is the percentile of sorted intensity in raw image, determining the threshold of filtering.

4.1. Quantitative result (Ablation study)

According to our experiment, OpenIter has less adjustable range. OpenIter over than 1 will cause no points detected in some image, leading to error in calculating ICP. Moreover, we found that performace is not sensitive to AssistTh. Since that we added other camera's result from the most close time stamp, AssistTh won't change overall result if it was not large enough. However, Big AssistTh is fundmentally unreasonable. Therefore, in below ablation study, we fixed OpenIter equal to 1 iteration and AssistTh equal to 0.2 sec.

- **FiltDist** (with maxCorners=2000, qualityLevel=0.01, minDistance=10)

FiltDist	**15**	**19**	**20**	**21**	**22**	**25**
Train score	0.43750 0.43541 0.42780	0.43358 0.43373 0.42752	0.43371 0.43448 0.42649	0.43324 0.43413 0.42530	0.43485 0.43538 0.42604	0.43401 0.43501 0.42575
Test score	NaN	NaN	NaN	NaN	NaN	NaN

- **minDistance** (with FiltDist=19, maxCorners=2000, qualityLevel=0.1)

minDistance	**10**	**15**	**17**	**19**	**20**	**21**	**22**
Train score	0.42982 0.42760 0.40546	0.43278 0.43024 0.41438	0.43220 0.42929 0.41335	0.43120 0.42862 0.40477	0.43129 0.42590 0.40353	0.43277 0.42617 0.40578	0.43004 0.42909 0.40613
Test score	NaN	NaN	NaN	NaN	0.42181	NaN	NaN

- **intensity_ratio**

1. Phase 1 (with FiltDist=19, maxCorners=2000, qualityLevel=0.1, minDistance=20)

intensity_ratio	**0.05**	**0.1**	**0.11**	**0.12**	**0.125**	**0.13**	**0.14**	**0.15**
Train score	0.62965 0.43389 0.42220	0.45478 0.41594 0.40978	0.43073 0.41534 0.41325	0.42807 0.41750 0.41692	0.42344 0.41819 0.41095	0.42941 0.41524 0.41416	0.43022 0.41593 0.41438	0.43263 0.41814 0.41728
Test score	NaN	NaN	NaN	NaN	0.39941	0.40042	NaN	NaN

2. Phase 2 (Using adaptive mask, with FiltDist=19, maxCorners=2000, qualityLevel=0.1, minDistance=20)

intensity_ratio	**0.121**	**0.122**	**0.123**	**0.124**
Train score	0.42939 0.41484 0.40928	0.42997 0.41963 0.40473	0.42863 0.41871 0.41026	0.42317 0.42194 0.41127
Test score	NaN	0.40227	NaN	0.39902

- qualityLevel (Using adaptive mask, with FiltDist=19, maxCorners=2000, qualityLevel=0.124, minDistance=20)

qualityLevel	**0.01**	**0.05**	**0.08**	**0.09**	**0.11**	**0.12**	**0.13**	**0.15**	**0.2**

qualityLevel	**0.01**	**0.05**	**0.08**	**0.09**	**0.11**	**0.12**	**0.13**	**0.15**	**0.2**
	0.42651	0.42655	0.42241	0.42228	0.42320	0.42454	0.42463	0.42367	0.47305
Train score	0.41933	0.41979	0.42179	0.42188	0.42194	0.42159	0.42158	0.41856	0.42486
	0.41144	0.41029	0.41158	0.41158	0.41132	0.41086	0.41084	0.41245	0.40970
Test score	NaN	NaN	0.39968	NaN	NaN	0.39871	0.39871	0.40052	0.40606

- Final optimal parameters
    - intensity_ratio = 0.124
    - OpenIter = 1
    - FiltDist = 19
    - AssistTh = 0.2
    - maxCorners = 2000
    - qualityLevel = 0.12
    - minDistance = 20

4.2. Qualitative result

Fig.6 Four results with distinct camera but close time stamp

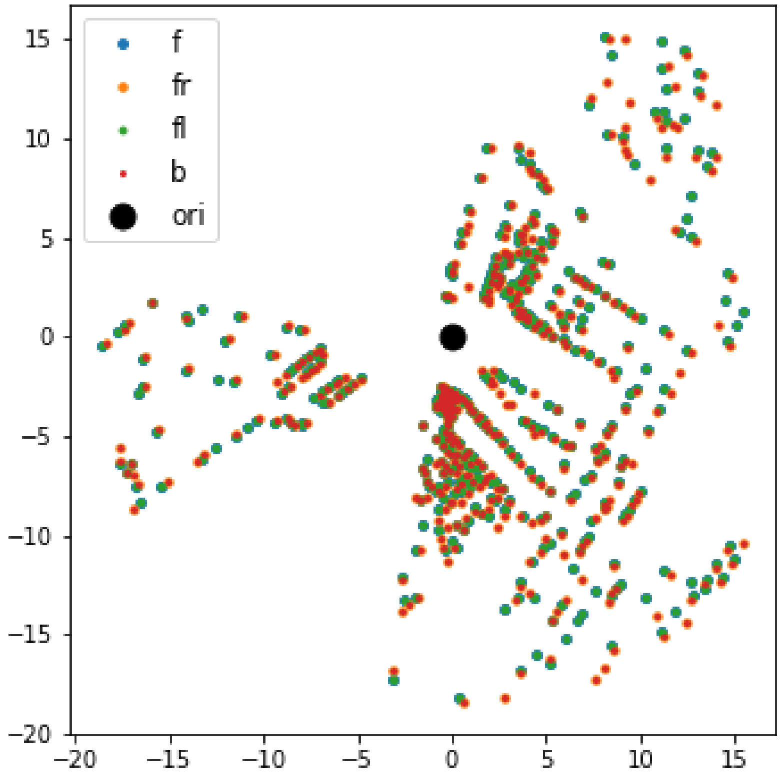

Fig.7 Correspong top view.

In Fig.6 and Fig.7, we augmented point clouds with the other cameras result. Apparently, although they are not completely matched due to the different augmented direction, we can still find that augmentation indeed help acquiring more comprehensive results.

V. Discussion

In the discussion section, we analyze the strengths and limitations of our proposed method and provide insights into the results obtained. We also discuss potential improvements and future directions for further enhancing the performance of the system.

5.1. Strengths of the Proposed Method

One of the key strengths of our method is the ability to effectively detect traffic signals in the driving car scene dataset. By utilizing basic morphological processing and mask filtering, we were able to achieve preliminary automatic traffic signal segmentation. Additionally, the use of the Shi-Tomasi corner detection algorithm allowed us to acquire reliable feature points representing corners in the image.

5.2. Limitations and Challenges

Despite its strengths, our method has some limitations and faces certain challenges. One limitation is that the method may still produce false positives or false negatives in traffic signal detection due to variations in complex scenarios. Another challenge is the presence of noise and unwanted regions in the extracted feature points. While we applied morphological opening to eliminate small regions, some errors may persist, leading to inaccurate point cloud reconstruction. Exploring advanced techniques or incorporating additional information could improve the accuracy of feature point extraction.

5.3. Potential Improvements and Future Directions

To address the limitations and challenges mentioned above, several potential improvements and future directions can be considered.

1. **Advanced Filtering Techniques**: Investigating advanced filtering techniques, such as adaptive thresholding or machine learning-based methods, could enhance the accuracy of traffic signal detection. These techniques can better handle complex lighting conditions and improve the segmentation results.

2. **Restoration oriented preprocessing**: Besides enhacing the ability of filtering stage, using image restoration as preprocessing. For example, dehazing, shadow removal, super resolution, etc. With these restoration method as preprocessing, system can fit more complicated scene.

3. **Refinement of Feature Point Extraction**: Exploring alternative corner detection algorithms or incorporating additional cues, such as texture or edge information, could improve the accuracy and robustness of feature point extraction. This could help eliminate false positives and improve the overall quality of the reconstructed point cloud.

4. **Evaluation of Diverse Datasets**: Testing the proposed method on diverse datasets, including different driving scenarios, and road environments, would help assess its generalization capability and identify any potential limitations or areas for improvement.

VI. Conclusion

In conclusion, we have presented a corner-detection based 3D reconstruction algorithm for the driving car scene dataset. Our proposed method leverages basic morphological processing, mask filtering, and corner detection to achieve traffic signal detection and feature point extraction. Through an ablation study, we determined the optimal hyperparameters for the method.

While our method demonstrates acceptable results, there are still challenges to overcome and areas for improvement. The proposed method shows strengths in traffic signal detection and feature point extraction, but further refinements are needed to improve the accuracy and robustness of the system. By exploring advanced techniques, refining feature point extraction, and evaluating diverse datasets, we can enhance the performance and applicability of the method for real-world driving scenarios.

Overall, our work contributes to the field of 3D reconstruction in driving scenes for future research in this area.

VII. Reference

[1] Jianbo Shi and Tomasi (1994) ‘Good features to track’, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94 [Preprint]. doi:10.1109/cvpr.1994.323794.

[2] Harris, C. and Stephens, M. (1988) ‘A combined corner and Edge Detector’, Procedings of the Alvey Vision Conference 1988 [Preprint]. doi:10.5244/c.2.23.

[3] Point Cloud Processing with Open3D: A beginner’s guide to getting started (2021) YouTube. Available at: https://www.youtube.com/watch?v=zF3MreN1w6c&list=PLkmvobsnE0GEZugH1Di2Cr_f32qYkv7aN (Accessed: 14 June 2023).

[4] Implementations of a rather simple version of the iterative closest point algorithm in various languages., GitHub. Available at: https://github.com/pglira/simpleICP (Accessed: 14 June 2023).

[5] What is SLAM (simultaneous localization and mapping) – matlab & simulink (no date) What Is SLAM (Simultaneous Localization and Mapping) – MATLAB & Simulink - MATLAB & Simulink. Available at: https://www.mathworks.com/discovery/slam.html (Accessed: 14 June 2023).

[6] Tzeng, A. (2020) 針孔相機模型 Pinhole Camera Model, Math.py. Available at: https://allen108108.github.io/blog/2020/02/06/%E9%87%9D%E5%AD%94%E7%9B%B8%E6%A9%9F%E6%A8%A1%E5%9E%8B%20%20Pinhole%20Camera%20Model/ (Accessed: 14 June 2023).

[7] 程序员阿德 (2019) 角点检测：Harris 与 Shi-Tomasi, 知乎专栏. Available at: https://zhuanlan.zhihu.com/p/83064609 (Accessed: 14 June 2023).