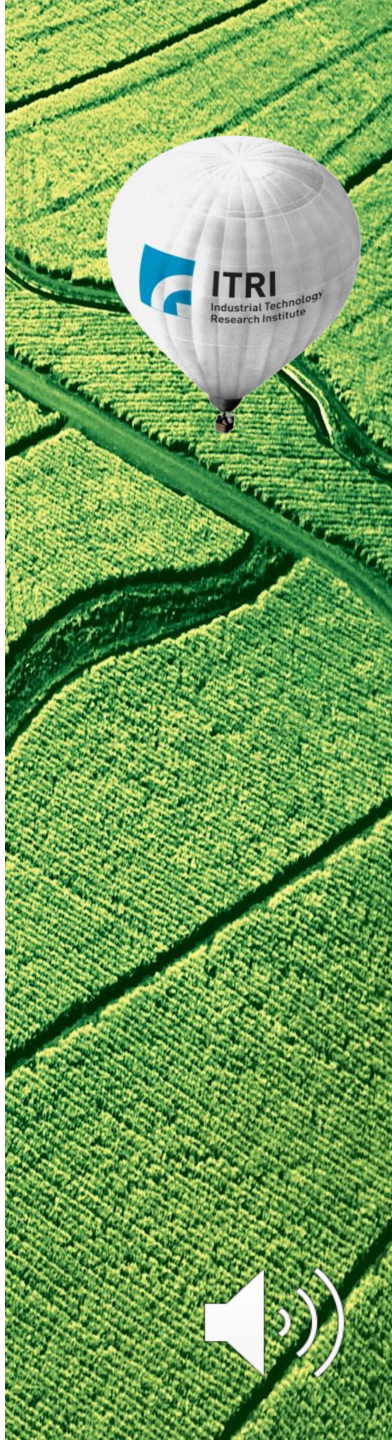
工業技術研究院
Industrial Technology
Research Institute



CV Final Project

3D Reconstruction
from
Road Marker Feature Points

Outline

- Introduction
- Dataset
- Evaluation
- Schedule
- Submission

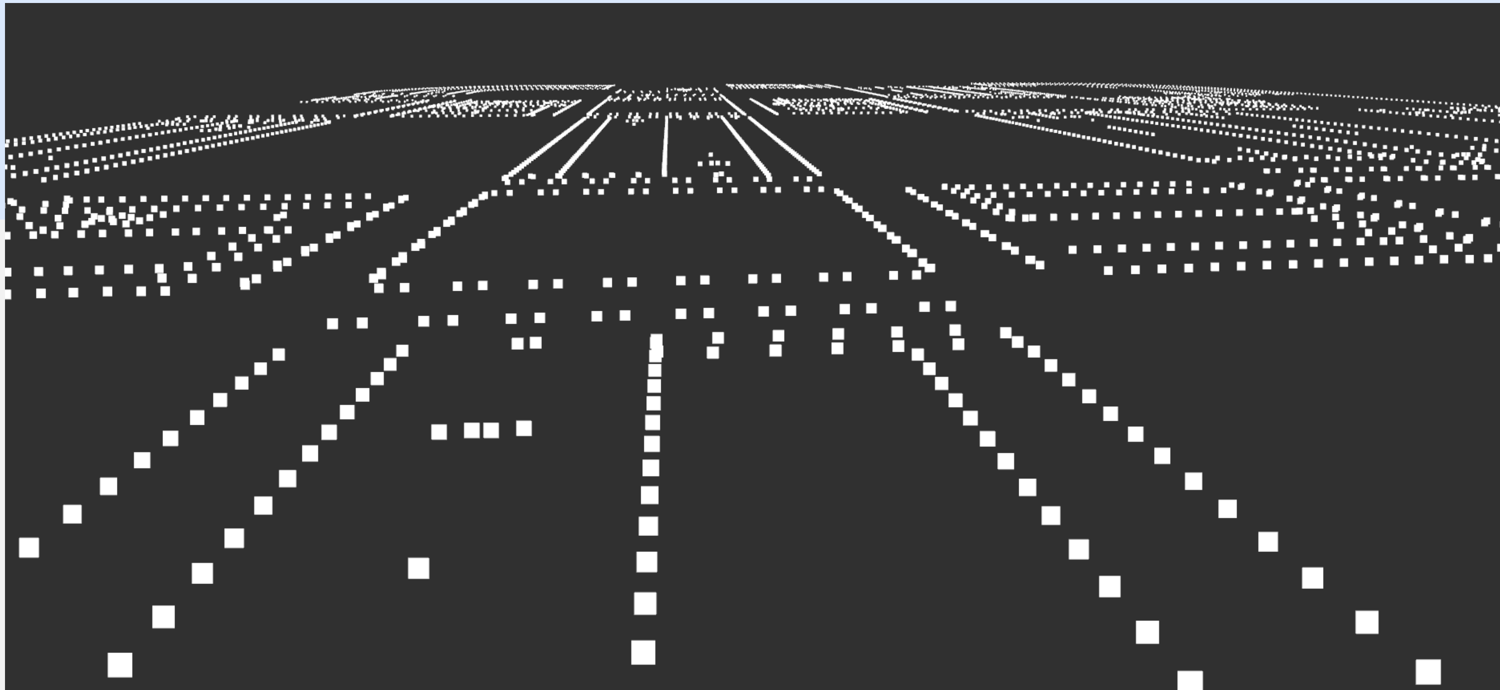
Outline

- Introduction
- Dataset
- Evaluation
- Schedule
- Submission

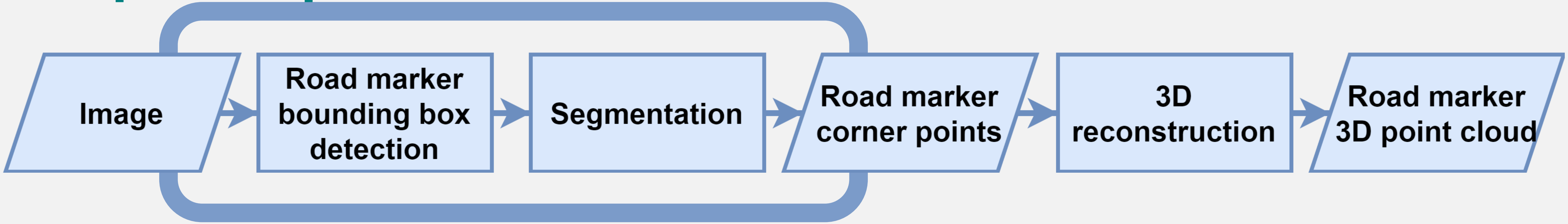
Road marker feature points 3D reconstruction

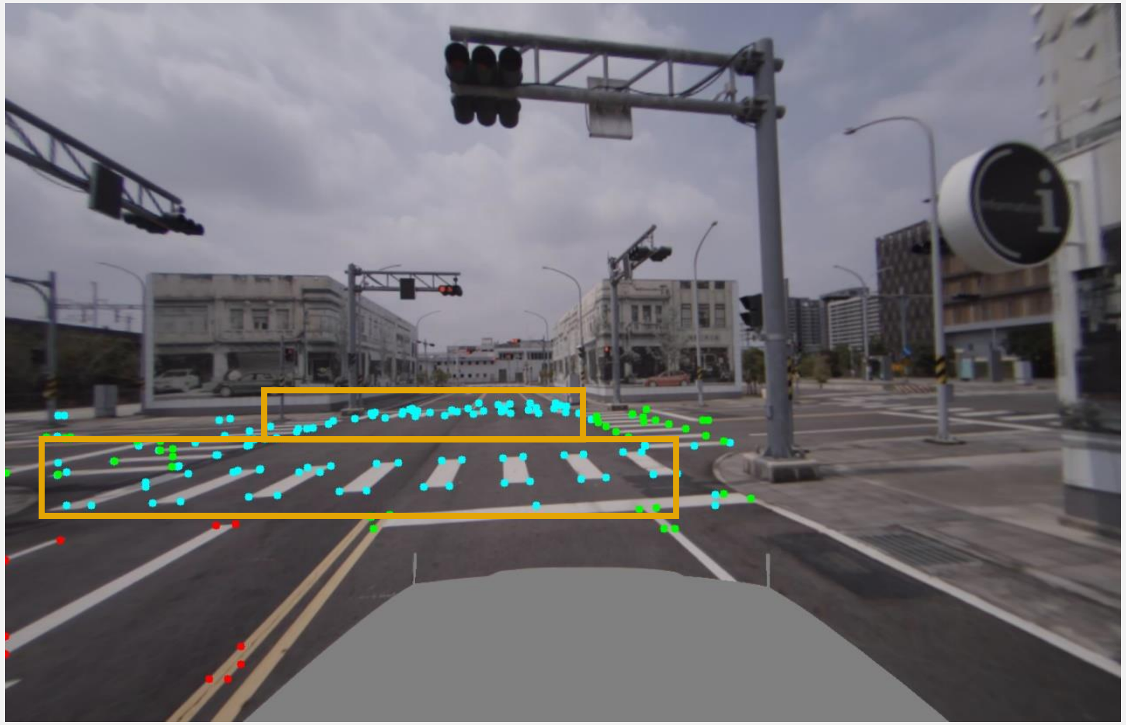
Output: 3D point cloud

Input: video



Simple Pipeline Reference

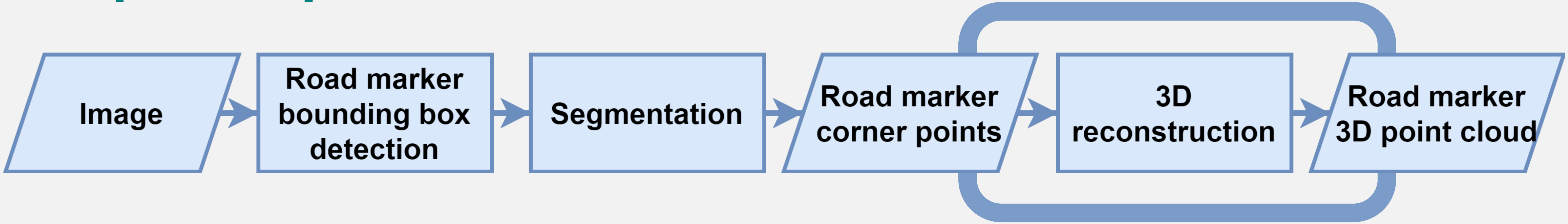


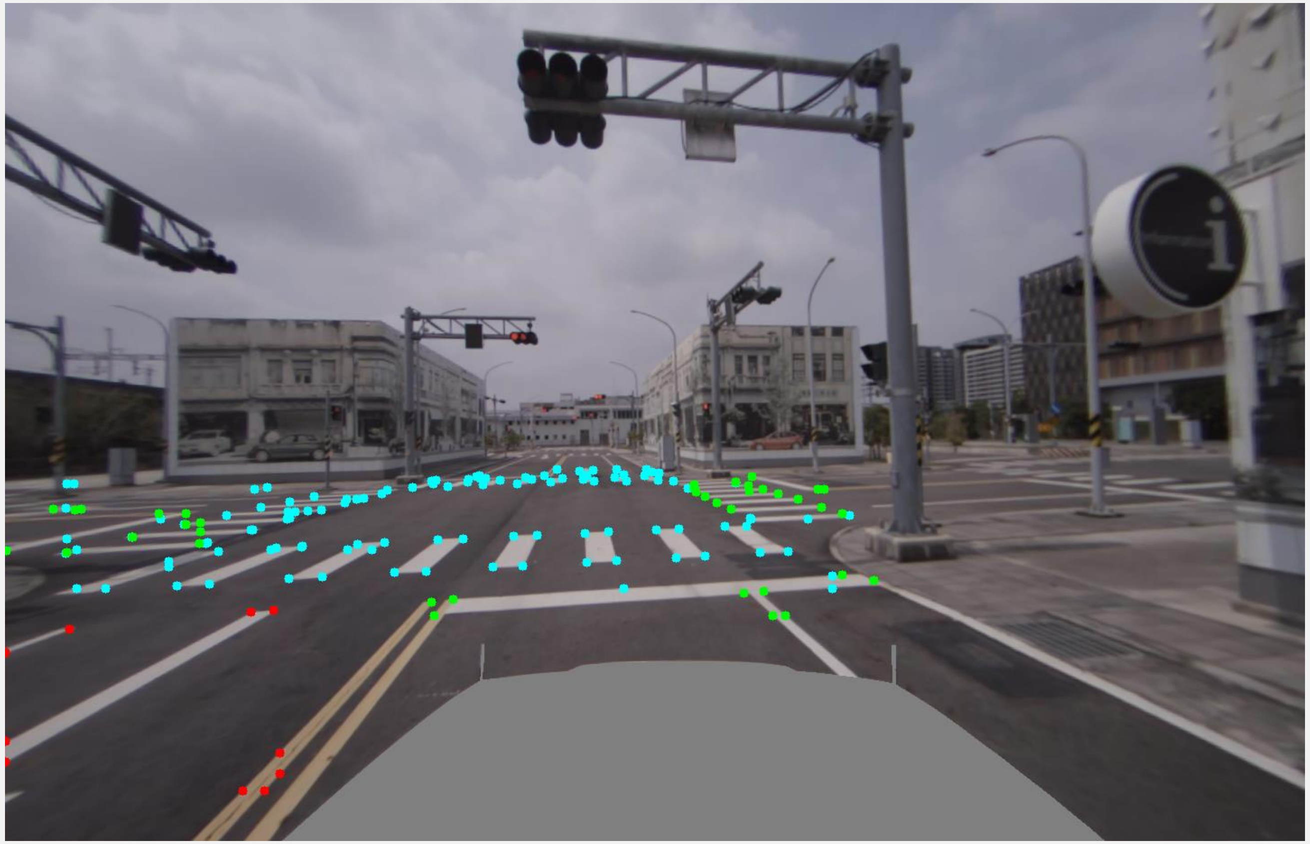


Reference methods:

- crop
- threshold
- inRange
- findContours
- approxPolyDP

Simple Pipeline Reference

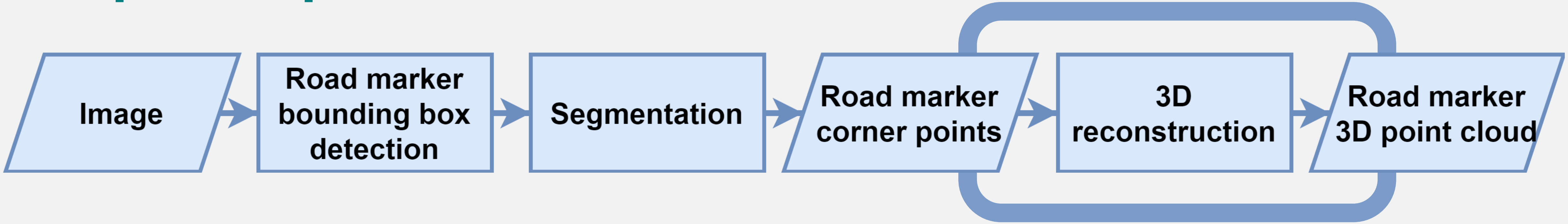


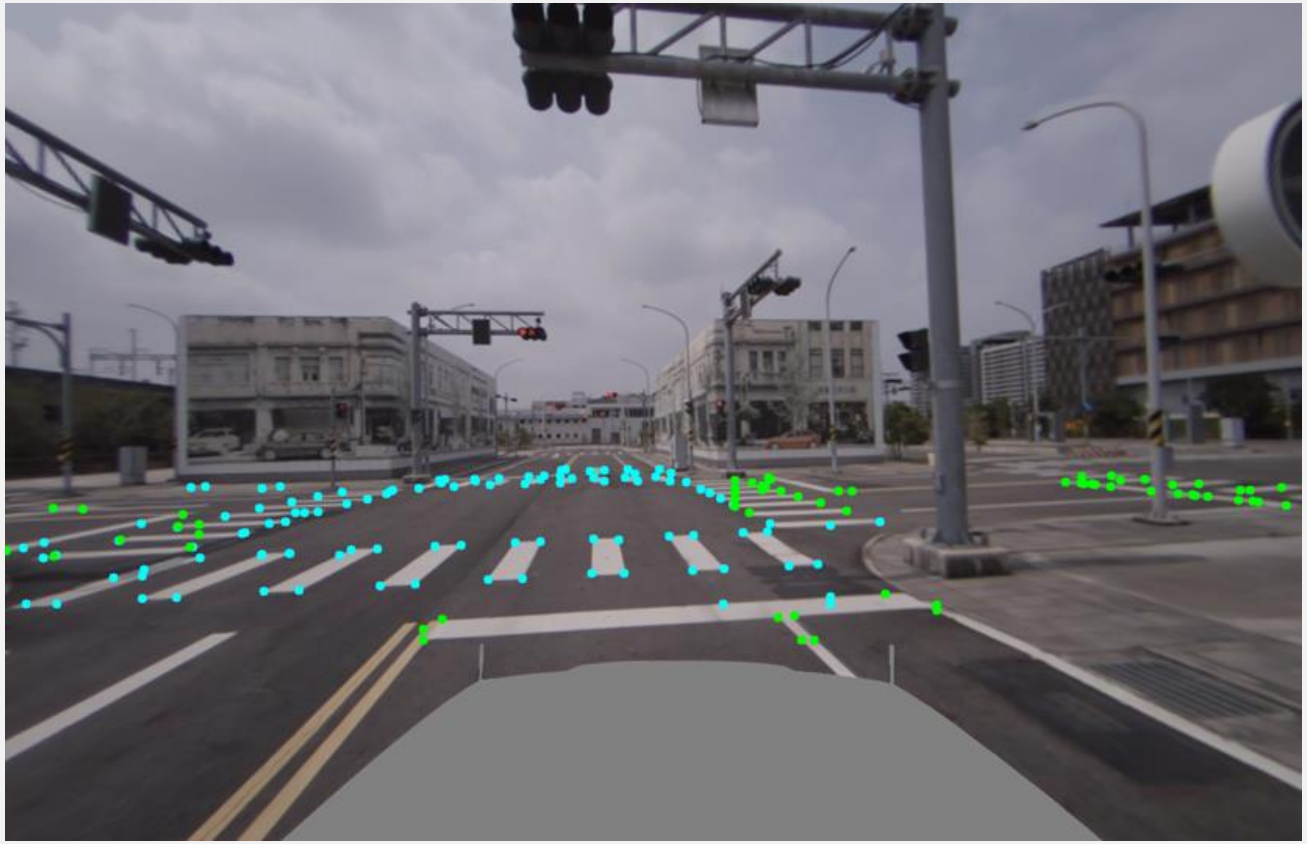


Reference methods:

- Pinhole model
- Structure from motion
- keyword:
    - SLAM
    - Visual-Inertial Odometry

Simple Pipeline Reference





Reference methods:

- Pinhole model
- Structure from motion
- keyword:
  - SLAM
  - Visual-Inertial Odometry

Outline

- Introduction
- Dataset
- Evaluation
- Schedule
- Submission

NEW!

Dataset Introduction

- Public Set
  - 3 video sequences: seq1, seq2, seq3
  - Download link: https://140.112.48.121:25251/sharing/Lw8QTICUf

- Private Set
  - 2 video sequences: test1, test2
  - Download link: https://140.112.48.121:25251/sharing/PyViYwNsv

NEW!

Dataset Structure

- ITRI_dataset
  - ReadMe.md
  - camera_info/
  - seq1, seq2, seq3/
  - eval.py, ICP.py
    - for evaluation
  - seq1, seq2, seq3/gt_pose.txt
    - the ground truth file for evaluation
  - test1, test2/

gt_pose.txt ×

ITRI_DLC > seq1 > gt_pose.txt

```
1   -4.47052 -83.9257
2   -4.46573 -83.9198
3   -4.43308 -83.8385
4   -4.39212 -83.7604
5   -4.3568 -83.6889
6   -4.32858 -83.6287
7   -4.3164 -83.6108
8   -4.31346 -83.6031
9   -4.28407 -83.5465
10  -4.2725 -83.5276
```

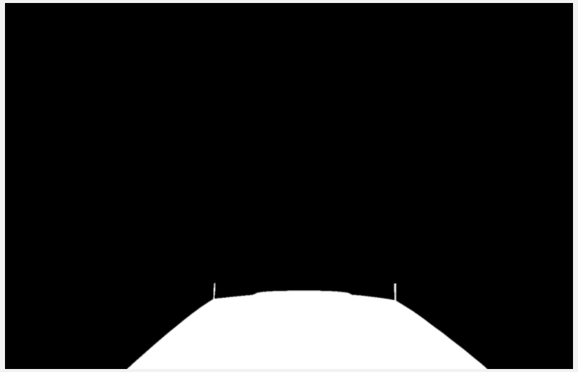
localization_timestamp.txt ×

ITRI_dataset > seq1 > localization_timestamp.txt

```
1   1681710717_541398178
2   1681710717_544461636
3   1681710717_580418005
4   1681710717_616719157
5   1681710717_649369147
6   1681710717_676552697
7   1681710717_685466899
8   1681710717_688825601
9   1681710717_715416966
10  1681710717_724763314
```

Dataset Structure

```
camera_info/
    {camera}/ (e.g. lucid_cameras_x00)
        {camera_name}_camera_info.yaml: (e.g. gige_100_b_hdr_camera_info.yaml)
            intrinsic parameters
        {camera_name}_mask.png:
            The mask for the ego car show in the image, it could help for decreasing some false alarms in detection.
        camera_extrinsic_static_tf.launch:
            transformation parameters between cameras
            key_word: tf2_ros, Robot Operating System (ROS)
```



f_mask.png

NEW!

Dataset Structure

- camera_extrinsic_static_tf.launch
  - tf2_ros
  - Robot Operating System

NEW!

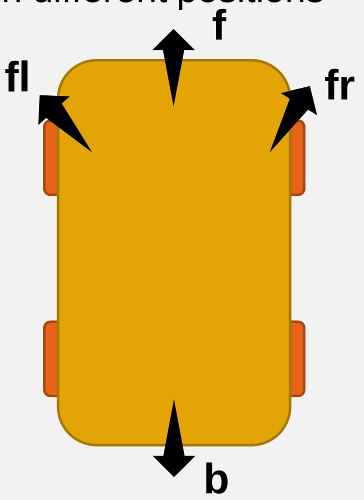
http://wiki.ros.org/tf2_ros

```
static_transform_publisher x y z qx qy qz qw frame_id child_frame_id
```

Publish a static coordinate transform to tf2 using an x/y/z offset in meters and quaternion.

base_link

the common reference frame for cameras installed in different positions




```
<launch>
    <arg name="main_camera_frame_id" value="/lucid_cameras_x00/gige_100_f_hdr"/>

    <rosparam param="/perception/main_camera_frame_id">
        /lucid_cameras_x00/gige_100_f_hdr
    </rosparam>

    <!-- All LiDAR to camera extrinsic parameters -->
    <node pkg="tf2_ros" type="static_transform_publisher"
      name="tf_main_camera__gige_100_fr_hdr"
      args="0.559084 0.0287952 -0.0950537 -0.0806252 0.607127 0.0356452 0.789699
      $(arg main_camera_frame_id) /lucid_cameras_x00/gige_100_fr_hdr" />

    <node pkg="tf2_ros" type="static_transform_publisher"
      name="tf_main_camera__gige_100_fl_hdr"
      args="-0.564697 0.0402756 -0.028059 -0.117199 -0.575476 -0.0686302 0.806462
      $(arg main_camera_frame_id) /lucid_cameras_x00/gige_100_fl_hdr" />

    <node pkg="tf2_ros" type="static_transform_publisher"
      name="velo2cam_tf__gige_100_fl_hdr_gige_100_fr_hdr_mix"
      args="-1.2446 0.21365 -0.91917 0.074732 -0.794 -0.10595 0.59393
      /lucid_cameras_x00/gige_100_fl_hdr /lucid_cameras_x00/gige_100_b_hdr" />

    <!-- tf about vehicle -->
    <node pkg="tf2_ros" type="static_transform_publisher"
      name="tf_main_camera__front_bump"
      args="0.06742502153707941 1.723731468585929 1.886103532139902 0.5070558775462676 -0.47615311808704197 0.4812773544166568 0.5334272708696808
      $(arg main_camera_frame_id) /front_bump"/>

    <node pkg="tf2_ros" type="static_transform_publisher"
      name="tf_main_camera__base_link_tmp"
      args="0.0 0.0 0.0 -0.5070558775462676 0.47615311808704197 -0.4812773544166568 0.5334272708696808
      /base_link $(arg main_camera_frame_id)"/>

</launch>
```

f-fr

f-fl

fl-b

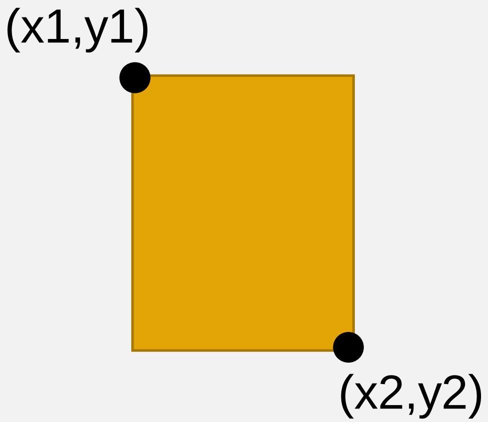
base_link-f

Dataset Structure

```
seq/ (e.g. seq1, seq2, seq3)
    dataset/
        {time_stamp}/ (e.g. 1681710717_532211005)
            1. camera.csv:
                camera name
            2. detect_road_marker.csv:
                a. detected bounding boxes, the bounding box are not always correct.
                b. format: (x1, y1, x2, y2, class_id, probability)
                c. class_id: (0:zebracross, 1:stopline, 2:arrow, 3:junctionbox, 4:other)
            3. initial_pose.csv:
                initial pose for ICP in "base_link" frame.
            4. raw_image.png:
                captured RGB image
            5. sub_map.csv:
                map points for ICP, (x, y, z).
            6. gound_turth_pose.csv: """not exist in all dirs"""
                x, y localization ground turth in "base_link" frame.

    other_data/
        {timestamp}_raw_speed.csv: (e.g. 1681710717_572170877_raw_speed.csv)
            car speed(km/hr)
        {timestamp}_raw_imu.csv:
            1st line: orientation: x, y, z, w
            2nd line: angular_velocity: x, y, z
            3rd line: linear_acceleration: x, y, z

    all_timestamp.txt:
        list all directories in time order
    localization_timestamp.txt:
        list all directories with "gound_turth_pose.csv" in time order
```



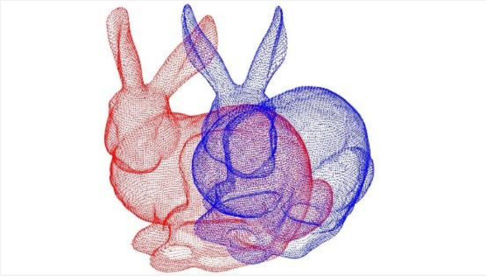
NEW!

Outline

- Introduction
- Dataset
- Evaluation
- Schedule
- Submission

NEW!

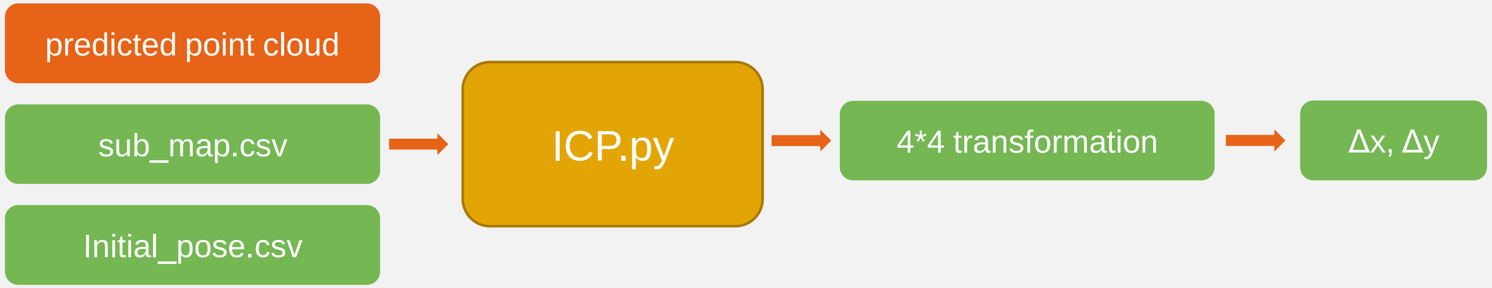
Evaluation Metric



- Iterative Closest Point (see ICP.py for details)
  - Fine local registration between predicted point cloud and ideal point cloud with initial pose information.
  - http://www.open3d.org/docs/release/getting_started.html
  - You may need to modify the code to implement the ICP function.
    - **Line 31:** Change the path of the directory here.
    - **Line 37~40:** Enter the filename of your point cloud.
    - **Line 45~49:** You can modify the threshold and iteration for ICP function for better alignment.
  - The output of the ICP function is a 4*4 transformation matrix (line46).
    - We extract the Δx and Δy from the matrix for evaluation.
    - You need to modify the code to save the values of Δx and Δy for the usage of eval.py.

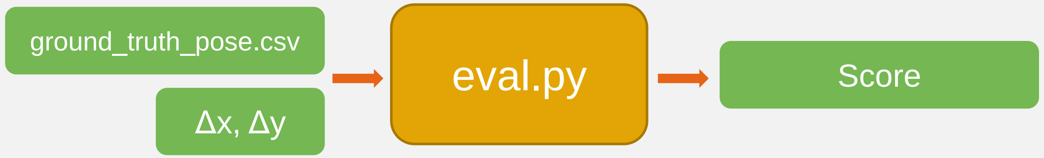
```
30 if __name__ == '__main__':
31     path_name = './seq1/dataset/1681710717_541398178'
32 
33     # Target point cloud
34     target = csv_reader(f'{path_name}/sub_map.csv')
35     target_pcd = numpy2pcd(target)
36 
37     # Source point cloud
38     #TODO: Read your point cloud here#
39     source = csv_reader(f'{path_name}/[[[your file name]]]')
40     source_pcd = numpy2pcd(source)
41 
42     # Initial pose
43     init_pose = csv_reader(f'{path_name}/initial_pose.csv')
44 
45     # Implement ICP
46     transformation = ICP(source_pcd, target_pcd, threshold=0.02, init_pose=init_pose)
47     pred_x = transformation[0,3]
48     pred_y = transformation[1,3]
49     print(pred_x, pred_y)
```

$$\begin{bmatrix} \begin{bmatrix} \text{3*3 rotation} \end{bmatrix} & \begin{bmatrix} \boxed{\begin{matrix}\Delta x \\ \Delta y\end{matrix}} \\ \Delta z \end{bmatrix} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}$$



NEW!

Evaluation Metric



- ## Evaluation (see eval.py for details)
  - Calculate the mean error (ME) between predicted Δx, Δy and the ground truth.
  - You need to modify the code to implement the evaluation function.
    - **Line 13~18:** Enter the filename of your prediction in Line 16.

```
13 def benchmark(dataset_path, sequences):
14     for seq in sequences:
15         label = np.loadtxt(os.path.join(dataset_path, seq, 'gt_pose.txt'))
16         pred = np.loadtxt(os.path.join(dataset_path, seq, 'gt_pose.txt'))  #TODO: Enter your filename to replace gt_pose.txt here#
17         score = calculate_dist(label, pred)
18         print(f'Mean Error in {seq}: {score:.4f}')
```

    - **Line 23~24:** Enter the path of dataset; you may also partially evaluate the dataset.

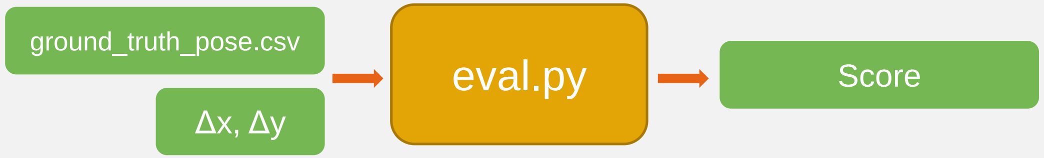
```
22 if __name__ == '__main__':
23     dataset_path = '/home/chenyukai/ITRI/TA_version'
24     sequences = ['seq1', 'seq2', 'seq3']
25     benchmark(dataset_path, sequences)
```

  - The output of the evaluation function is the mean error scores of the sequences, the lower the better.

```
def calculate_dist(label, pred):
    dist = np.sqrt(np.sum((label-pred)**2, axis=1))
    dist = np.mean(dist)
    return dist
```

NEW!

Evaluation Metric



- Evaluation (see eval.py for details)
  - For each sequence, you need to generate one text file (e.g. pred_pose.txt) to save your predicted Δx and Δy.
  - The numbers are separated with " " (one space).
  - The order of the prediction is the same as the order in "localization_timestamp.txt".
  - The format of generated text file is the same as the "gt_pose.txt", which is shown below.

gt_pose.txt — ITRI_DLC > seq1 > gt_pose.txt

```
-4.47052 -83.9257
-4.46573 -83.9198
-4.43308 -83.8385
-4.39212 -83.7604
-4.3568 -83.6889
-4.32858 -83.6287
-4.3164 -83.6108
-4.31346 -83.6031
-4.28407 -83.5465
-4.2725 -83.5276
```

Δx Δy

Same order

localization_timestamp.txt — ITRI_dataset > seq1 > localization_timestamp.txt

```
1681710717_541398178
1681710717_544461636
1681710717_580418005
1681710717_616719157
1681710717_649369147
1681710717_676552697
1681710717_685466899
1681710717_688825601
1681710717_715416966
1681710717_724763314
```

directory name

pred_pose.txt — TA_version > seq1 > pred_pose.txt

```
delta x1 delta y1
delta x2 delta y2
delta x3 delta y3
delta x4 delta y4
delta x5 delta y5
delta x6 delta y6
delta x7 delta y7
delta x8 delta y8
delta x9 delta y9
delta x10 delta y10
```

Δx Δy

NEW!

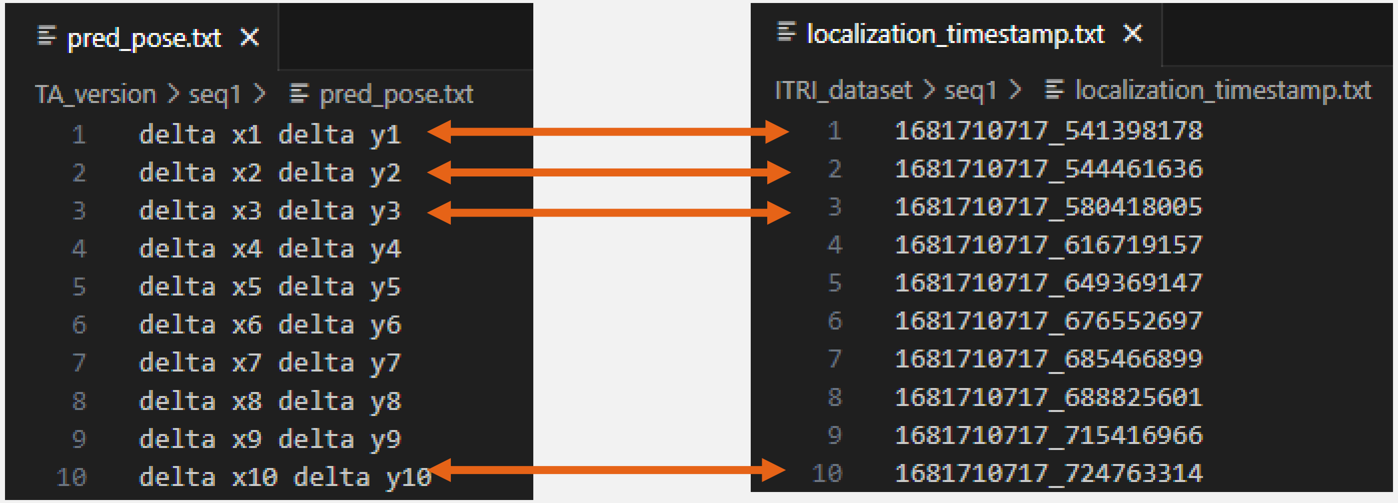
Evaluation Server

- Our final project challenge will be hold on Codalab competition server.
  - Link: https://codalab.lisn.upsaclay.fr/competitions/13369
- Please read all the rules written in the server carefully.
- **Maximum submissions: 80**
- **Maximum submissions per day: 10**

NEW!

Evaluation Server: Submission

- For each sequence, you need to generate one "pred_pose.txt" to save your predicted Δx and Δy. The format of "pred_pose.txt" is shown below.
  - For each sequence (test1 and test2), you need to generate one text file to save your predicted Δx and Δy.
  - The filename should be "**pred_pose.txt**".
  - The numbers are separated with " " **(one space).**
  - The order of the prediction is the same as the order in "localization_timestamp.txt" of test1 and test2.



NEW!

Evaluation Server: Submission

- We will evaluate the score on the sequences: test1 and test2 in the test set.
- Directory architecture:

solution/

└ test1/pred_pose.txt

└ test2/pred_pose.txt

- You need to **compress** “solution” into a zip file. **The file name of the zip is free**.
  - After unzipping it in Linux system, it should generate **one directory named “solution”**.
- If any of the file format is wrong, the evaluation process may be failed.

NEW!

Evaluation Server: Submission

- The ME scores on the leaderboard are only based on **50% of the testing set**.
- After we close the evaluation server, the team leaders need to upload **the final solution zip file to NTU COOL**.
  - Upload deadline: 2023/06/06 09:00
- We will announce the final leaderboard (50% data, 50% hidden data, 100% data), on NTU COOL at 2023/06/06 12:00.
  - The top 10 teams on the final leaderboard will be chosen for final presentation.
- **Note**: The zip file you upload should generate the same score as that on the leaderboard.

Grading

- Quantitative: 50%
  - according to the leaderboard
- Presentation: 50% (top 10 teams)
  - Novelty and technical contribution (20%)
  - Completeness (25%)
    - Ablation studies, visualization, experiments, analysis, etc
  - Presentation (5%)
- Report: 50% (other teams)
  - Novelty and technical contribution (25%)
  - Completeness (25%)
    - Ablation studies, visualization, experiments, analysis, etc

Score	# of teams
50%	1
49%	1
48%	1
46%	4
42%	4
40%	4
38%	3

Note

Only top 10 teams on the final leaderboard will be chosen for final presentation.

Outline

- Introduction
- Dataset
- Evaluation
- Schedule
- Submission

NEW!

Schedule

- Evaluation Server Open
  - 2023/05/20 00:00
- Evaluation Server Close
  - 2023/06/05 23:59
- Final zip file Submission
  - 2023/06/05 09:00
- Code Submission
  - 2023/06/06 23:59
- Oral Presentation
  - 2023/06/09 14:20~17:20 (Tentative)
- Report Submission
  - 2023/06/09 23:59

Outline

- Introduction
- Dataset
- Evaluation
- Schedule
- Submission

Code Submission

- Only the team leader needs to upload the code to NTU COOL.
- R07654321/
  - **README file**
  - **All your codes** (including training & testing)
  - **Model file** (which can reproduce the result on the leaderboard)
- Compress all files in a zip file named StudentID.zip (e.g. R07654321.zip)
  - After TAs run "unzip R07654321.zip", it should generate one directory named R07654321.
- In README file, you need to clearly describe how to set up the environment and the steps to run your code (training & testing), so that TAs can reproduce the results.
  - If we can not reproduce your result on the leaderboard, you will receive 0 point in the performance part. Minor errors are acceptable.
- Deadline: 2023/06/06 23:59

Report Submission

- Only the team leader needs to upload the report to NTU COOL.
- The teams who are selected for final presentation need to upload your presentation slide in *pptx* format.
- The teams who are not selected for final presentation need to upload your report in *pdf* format.
- Deadline: 2023/06/09 23:59

NEW!

Supplementary

- The init_pose.csv and sub_map.csv should not be used in your algorithm, they can only be applied in ICP.py.
- The filenames of ground_truth_pose.csv were typed wrong.
- 1681116137_203254447 should be removed in the “localization_timestamp.txt” of seq2.

- The height of cameras is 1.63 meter.
- The format of timestamp is {sec}_{ms}. (e.g. 1681116137_203254447)
- The sub_map represents the point cloud within 25 meters at that timestamp.

Contact

陳昱愷

- chenyukai@media.ee.ntu.edu.tw