Roy Orfaig, Prof Ben-Zion Bobrovsky. spring 2024

Project 1

Topics:

- Geodetic coordinate system and the KITTI dataset
- Probabilistic Occupancy Grid
- Sensor fusion and semantic segmentation

During this project, you will get experience with recorded data from a mobile lab and practice implementing the algorithm for robot mapping which you learned in the class. It is recommended to use the instructions in the attached Google Colab script, but please copy all the figures to the final report. Analysis of the results is a significant part of the project. To diagnose the details and explanations is what is expected from you.

Notice that each of you is given a different scenario to analyze, therefore your conclusions and results should differ from one to the other.

Data Category	**name**	**Last ID**
City	2011_09_26_drive_0104	0-1
City	2011_09_26_drive_0095	2-3
City	2011_09_26_drive_0064	4-5
City	2011_09_26_drive_0013	6-7
Residential	2011_09_26_drive_0061	8-9

** The KITTI vision benchmark suite

i. Link
ii. paper

Part A: Geodetic coordinate system and get familiar with the KITTI dataset (15%)

1. Download your recording data according to the table below from the KITTI dataset.

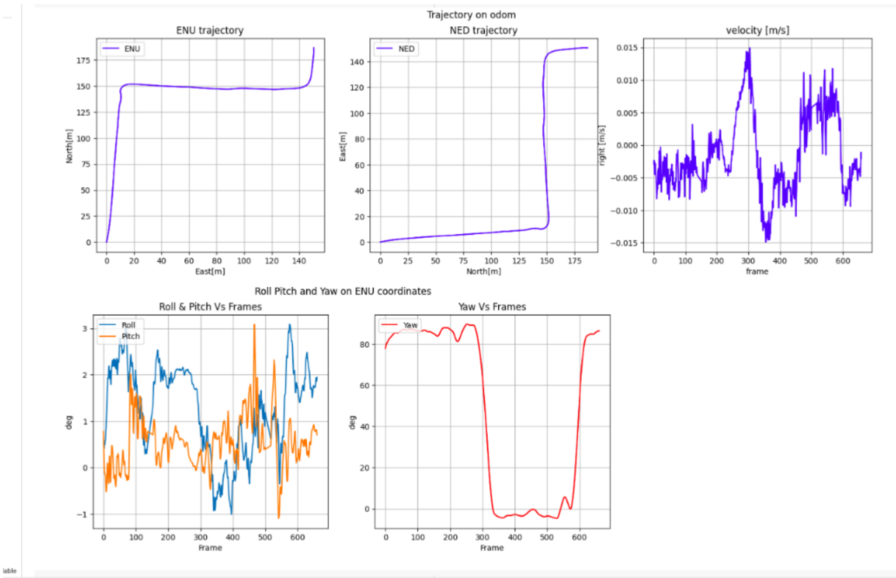
    a. Describe technical details about your scenario such as driving route, presence of dynamic objects (cyclists, cars and pedestrians), occlusions, turns etc.

    b. Use Google Maps to find the trajectory of the recorded data. Plot the results.

    Example: (the trajectory of the car is marked in red)

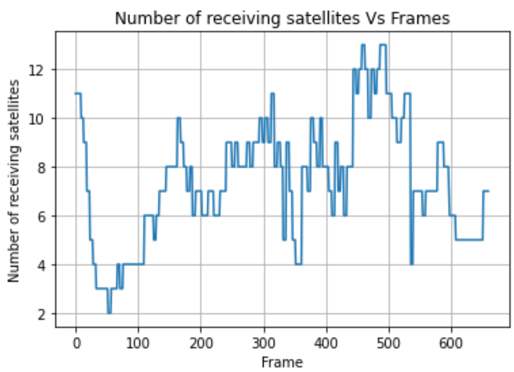
    c. Display the trajectory in ENU and NED representation of local coordinates, right side angular velocity (to the right), pitch, roll and yaw. Explain your results.

    Example:



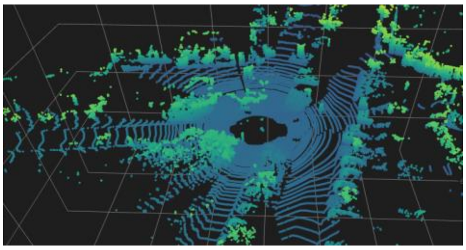
    d. Display the quality of GPS signal reception and any relevant details. Explain your results.

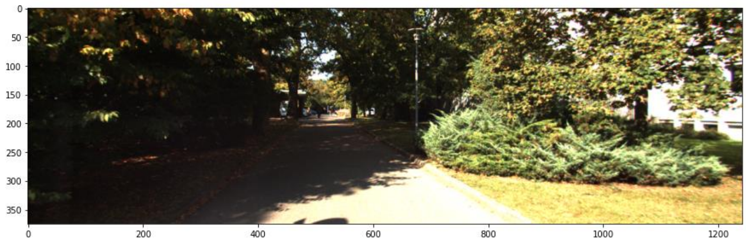
Example:



e. Display one example each of an image (from a camera) and its corresponding matched point cloud (from LiDAR) from your record. What are the advantages of each sensor in the scene related to perception task in autonomous driving?

Example:





Part B: Probabilistic Occupancy Grid (30%)

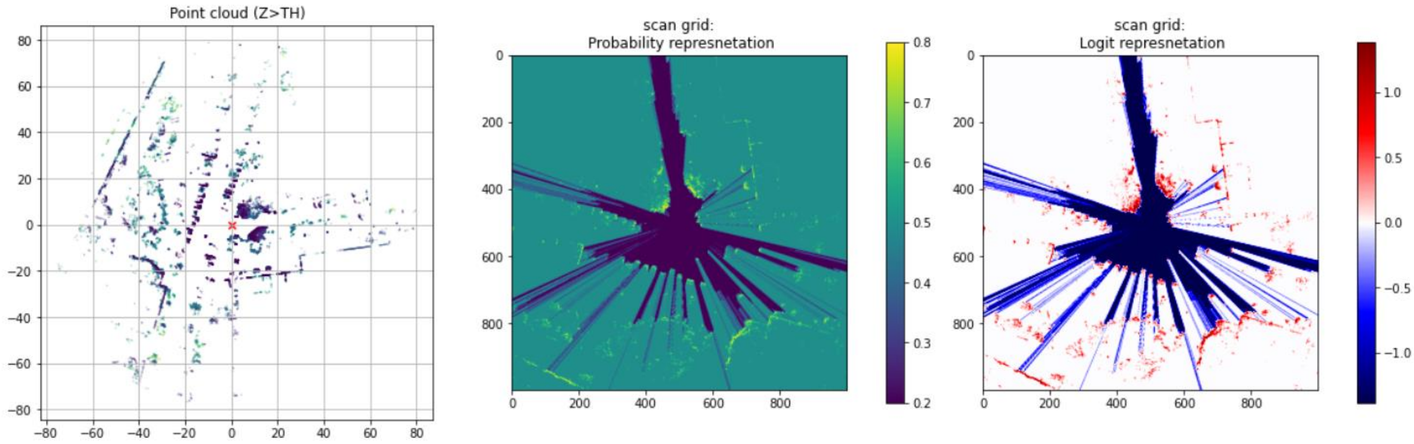
1. Implementation of a grid on a single scan based on the following instructions:
    a. To find a drivable path in the session, use a naïve role: filter all the point clouds above ~30 cm off the ground (assume flat surface).
    b. Recommend parameters.

Parameter	Value	Description
Resolution	0.2 (m)	cell size
ALPHA	0.2 (m)	The radial resolution when converting LiDAR data to grid map
BETA	0.1 (degree)	The angular resolution when converting LiDAR data to grid map
Max range	100 (m)	Maximum range of LiDAR points that will be converted to grid map.
Hit/Miss prob	0.8/0.2	Probabilities of hit and miss values
Probability Saturation	0.02/0.98	Minimum and maximum probability
Map width	100 (m)	Width of the map from side to side
SPHERICAL2CARTESIAN_BIAS	1 (pixels)	An adjustment needed due to some errors when converting the spherical grid map to the cartesian grid map.

   c. Add the figures of the first scan: filter point cloud, scan grid in probability and logit representations. Please describe your implementations in detail.

Example:

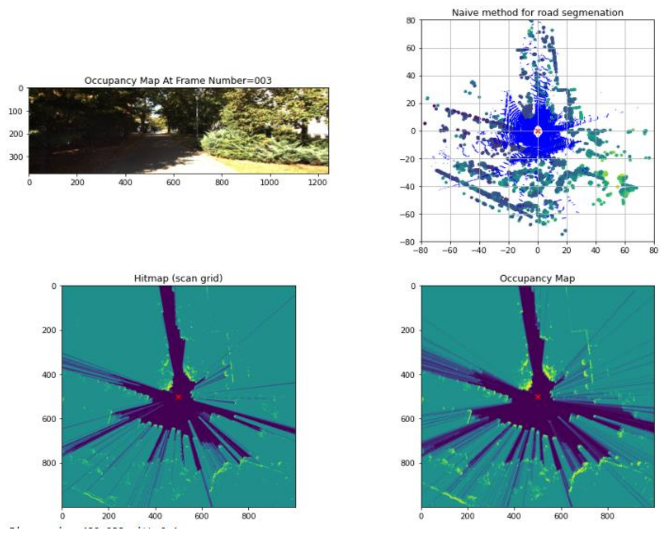
Expected results-



d. Update and shift the OGM (Occupancy Grid Map) according to the next pose of the car. Show an example of the first two frames.
e. Explain and analyze your results.

2. Implementation of a Probability Occupancy Grid over all the record:
   a. Pre-processing-
      i. Transform the point cloud recorded (in previous section) by the LiDAR to INS location.
      ii. Align the point cloud based on the IMU inputs.
   b. Build an OGM per LiDAR scan (use the function in section 1)
   c. Repeat the occupancy map process over all your scans/frames!
   d. Create an animation which includes the following sub plots like in the following

   example:



e. Describe your work process (explain flow + code).
f. Add to the report the last frame of the occupancy map.
g. Apply thresholds on the OGM (chose one frame from the record) to get segmented map: The segmented maps should describe 3 types of cells: free, occupied and unknown.

3. Analyze your results.
    a. Please show examples of good and bad examples (at least two of each) of mapping during the scene and explain what makes each example good or bad.
    b. Change any parameter** of the model according in any way you chose and run it again.
        i. Describe the change you made, final results and what you expect to happen in the mapping results.
        ii. Compare and analyze the effect of this change with examples in interesting cases.
        iii. Please add the new animation based on your experiential parameter.

***for examples:*

- *Hit/Miss probabilities to 0.6/0.4..*
- *Pixel size to 0.3..*
- *Probability saturation values..*
- *Alfa/beta..*

    c. Describe how objects of different types affect the mapping? For example:
    - Dynamic objects.
    - Parking cars /trees.
    - Sparse/dense objects.
    - Transparent materials (such as glass).
    - Different color objects of the same type..

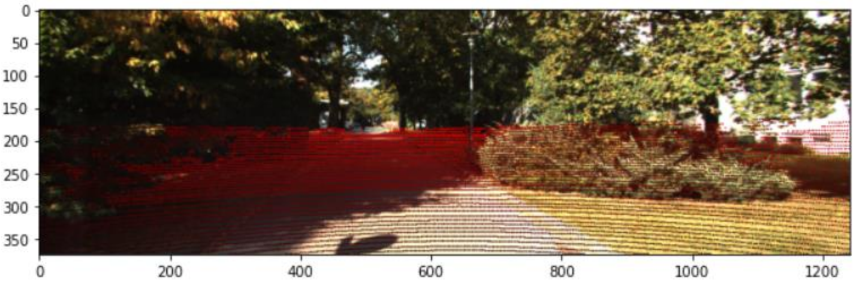
    Please explain and show examples from your record.

Part C: Sensor fusion and semantic segmentation (50%)

1. Sensor fusion: Projection of LiDAR Point clouds onto image coordinates (camera) as follows:
    - Filter out point clouds behind the camera.
    - Project points onto the image.
    - Filter out points that are outside the image frame.

a. Describe your work process (explain the flow).
b. Attach 2-3 examples of the projection onto image coordinates to the report:

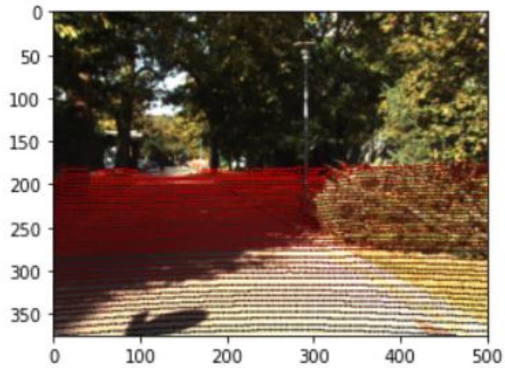
Example:



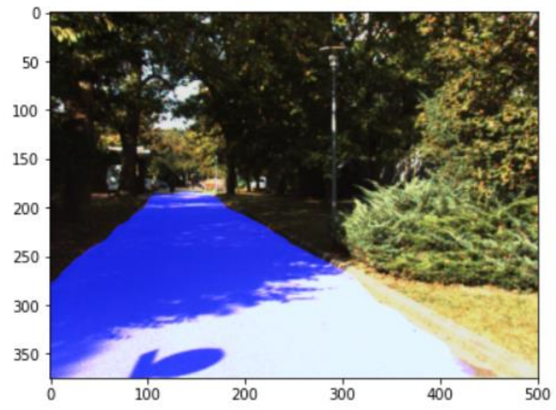
2. Road segmentation!
    a. Pre-processing: Image cropping- The deep learning model (DeepLabv3+) was trained with cropped images from the KITTI dataset with aspect ratio 4:3 (W:H), which was resized further to 513 x 513 images.

       crop the image (from the previous section) from the center and add it to your report.

       Example:

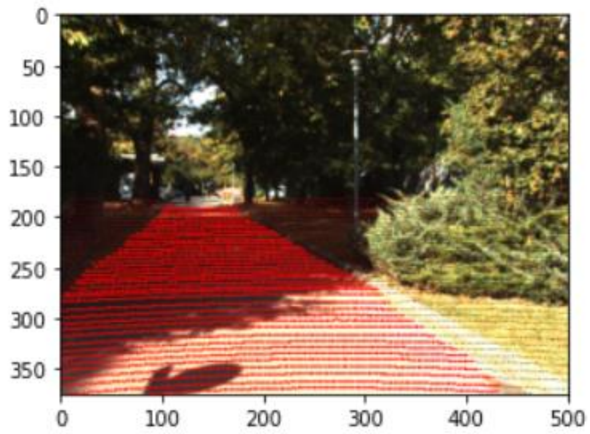


    b. Run DeepLabV3+ (use the pretrained model) on your image.
       Describe the process of using the algorithm and attach the results to the report:
       Example: (road is marked in blue)



c. Extract from the point cloud only the points that correspond to the road. Describe your process and attach the figure to the report:

Example:



3. LiDAR Road Filter

The road points set that are obtained from the previous step may contain false detections and obviously only contains points that are inside the camera field of view. We also need to detect road points that are outside this set!

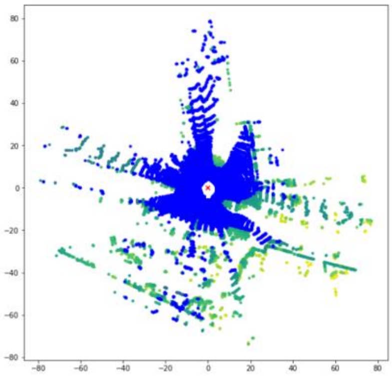
To do this, fit a plane model (which will represent the road model) Ax+By+Cz=1 to the current road points. Any points that are outside the camera's field of view and located near the road plane model can also be regarded as road points. RANSAC algorithm is applied when fitting the model to reduce the influence of outliers.

Parameter for RANSAC model:

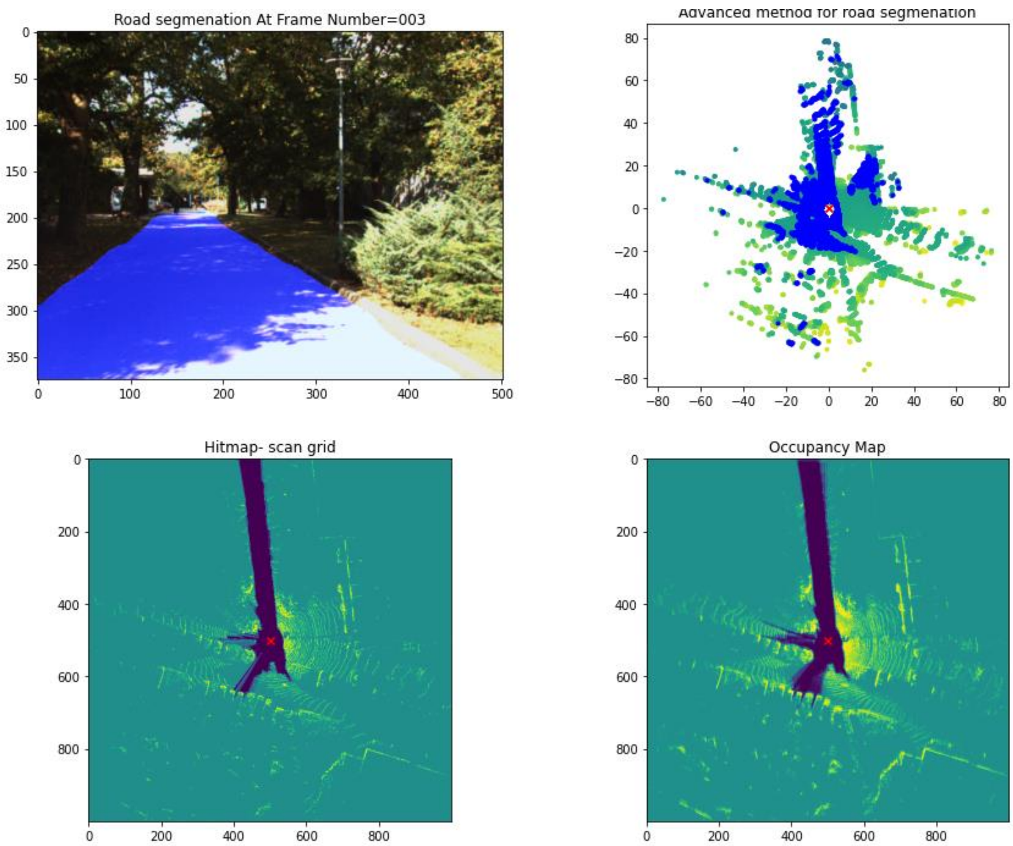
- ROAD_HEIGHT_THRESHOLD - The maximum distance (in height axis) between a point to the road model to be considered as road point

a. Calculate the plane model from the road points.
b. Predict the drivable path over all LiDAR point clouds based on the plane model you estimated in the previous section. Example:

In blue- the predicted path based on DeepLabv3+ and the projection mechanism



4. Implementation of a probability occupancy grid based on deep learning!
    a. Repeat the occupancy map process based on deep learning over all your scans/frames!
    b. Create an animation which include the following sub plots like in the following example:



    c. Describe your work process (explain flow + code).
    d. Add the animation to the report.

5. Analyze your results.
   a. Compare and analyze the new mapping based on deep learning to the naïve method (section B).
   b. Please show examples of good and bad examples (at least two of each) of mapping with deep learning during the scene and explain what makes each example good or bad.
   c. Describe at least three of the pros and cons to mapping based on deep learning vs the naïve method.
   d. Mention at least 2 algorithmic challenges in road segmentation for offroad.

<u>The final report should include the following details (5%)</u>**:**

- Analysis and explanations (Hebrew or English)- explain in detail and use examples to support your descriptions.
- Each graph/animation/figure must include titles, units, and descriptions.
- Attach the code in addition to your report.
- Attach the animations (AVI or MP4 format). Keep frame rate to a maximum of 10Hz.
- The code should be a script that displays and plots all the results. Verify that the code works properly and includes all the sub-functions. In case of a code which does not run/work properly/fail you might lose points.

Good Luck!

Appendix

- Please be honest, you may automatically lose points if you are caught copying including from the internet (code, results). The work is personal.

- Do not use prepared functions of Occupancy Grid. You should design the functions by yourself.

- See instructions about the recorded data in the Appendix.

- You are required to read the following paper for better understanding.
    - Vision meets Robotics: The KITTI Dataset/ Andreas Geiger

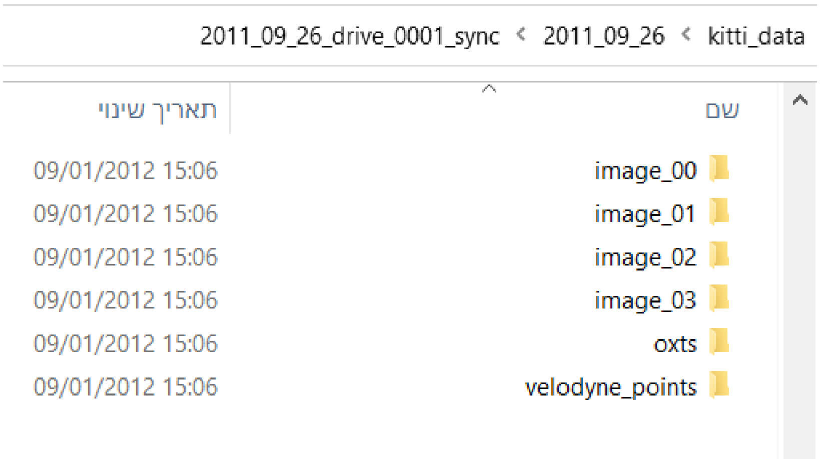
- SW – Python (recommended) / Matlab

- The final grade is given according to the quality of your analyses, descriptions, conclusions, explanations, the form of the results (plot, graphs, movies), understandable code with comments and explanations. It is possible that the final performance and results will not be as perfect as you desired, as this is real data and is part of the challenge of the autonomous driving field. Feel free to suggest solutions that could improve your results if this is the case.

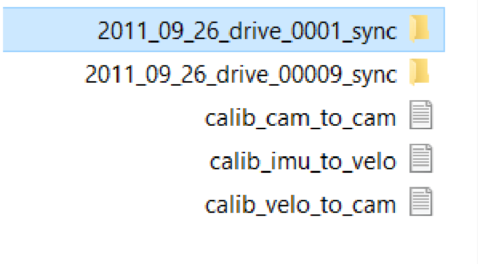
A. <u>The KITTI vision benchmark suite</u>

1. Link
2. Read the paper
3. Download the raw data development kit (1 MB)
4. Download your data according to the below table.
   download [synced+rectified data] [calibration]
5. Keep the format:
   The data format saved in <Folder of date>\<Title>
   For example:
   2011_09_26\ 2011_09_26_drive_0009_sync

6. Keep the following format:
   - You must save all the data in the same directory name as directory name



   - Save calibration files (transformation) in <Folder of date>, as follow:



7. Recorded Data (Go to raw data). (download [synced+rectified data] [calibration]).

B. Your final package should contain the following folders:

- **Report** contains the report
- **Code** contains all functions + sub-functions
- **Results** stores the resulting figures ,movies, etc.

Save the package as <name>_<ID>