Report No: 3
Team: BKBuilder
Date: 12/02/2023

1 Introduction

The third baseline includes details of BKBuilder's:

- **Obstacle & Lane detection:** Deploy and measure accuracy of traffic sign, pedestrian localization and lane segmentation
- **Motion Planning:** Deploy and integrate test complete architecture for steering angle and acceleration estimation via sensor and perception inputs. The system now can run many scenarios with minimal errors.
- **Software architecture real-world test:** Deploy and debug proposed system with 3 main blocks: perception, action, control via ROS.
- **Additional Hardware:** We additionally show that without a depth sensor, monocular depth estimation techniques can regress depth from video for processing.

2 Planned activities

Sprints 3/Contents	**Responsibility**	**Requirement Description**
Traffic dataset setup	Hai	Build a dataset for 8 kinds of traffic signs via Roboflow API.
Architecture definition (ROS-based)	Quang	Define the architecture in ROS format. Including: message type, constructing launch files, topics and software flow.
Sensor, Camera, IMU handling	Tien	Construct actionNODE to obtain sensor, cameras and IMU data and preprocess to subscribe to defined topics.
Object detection redesign	Hai	Redesign objectNODE to account for YOLOv5 architecture and implement area intrusion functionality for checking pedestrian crossing.
Lane detection migration and optimization	Quang	Construct perceptionNODE to send perception information such as steer angle, line types, etc. Redesign new optimal bird eye view for better calculation. Migrating previous source into ROS architecture.
Depth estimation integration	Hai	Research and integrate monocular depth estimation module (based on MiDAS model) to objectNODE.
Traffic light behaviour	Nhat	Implement a callback function which receives traffic light semaphore from the server and control the car accordingly to the traffic light colors.
Traffic sign behaviour	Nhat & Tien	Implement a callback function which receives traffic sign detection from object detection node and update the car's state accordingly.
Intersection behaviour	Hai	Implement logic to handle 8 traffic signs accordingly.
State machine for basic actions	Nhat	Implement an auto_control() function which defines the car's actions and state transition of every defined state of the car.
System test & integration	Tung	Proposed test-cases and scenarios to handle and debug.

Table 1: Planned activities

3 Status of planned activities

Below is the summary of status of activities in this baseline.

Sprints 3/Contents	Responsibility	Status
Traffic dataset setup	Hai	Passed
Object detection redesign	Hai	Passed
Depth estimation integration	Hai	Passed
Intersection behavior	Hai	Passed
Architecture definition (ROS-based)	Quang	Passed
Lane detection migration and optimization	Quang	Passed
Sensor, Camera, IMU handling	Tien	Passed
Traffic sign behaviour	Tien	Passed
Traffic light behaviour	Nhat	Passed
Traffic sign behaviour	Nhat	Passed
State machine for basic actions	Nhat	Passed
System test & integration	Tung	Passed

Table 2: Status of planned activities

4 General status of the project

Obstacle description	Priority	Completion	Status / Issues
Remote Car Control and Streaming	Critical	100%	N/A
Lane keeping	Critical	85%	The car can adjust steering angles and speed in ideal conditions. Curves are accounted and optimized. However, the model still sometimes have problems with dotted lines
Intersection crossing	Critical	60%	The car can recognized intersection via perception block and plan a path via action block. However, server information about V2X and semaphores haven't been tested.
Prove complete maneuverer after the following traffic signs: Stop, crosswalk, priority, parking	Critical	80%	The car can recognized all traffic signs and control the car accordingly. However, the perception sometimes provides false predictions (98% recall) and logic to handle those cases haven't been implemented.
Bring the car to a complete stop at the pedestrian signalling the will to cross the road on the crosswalk and wait until it crosses	Major	80%	The car can detect pedestrian and stop the car if it's in a pre-defined area. However, depth information of the pedestrian is not accounted and tracking module is not functional yet.
Overtake static car on road if the line signalling allows	Major	30%	The car can process location of other cars via V2X servers and recognized line signalling with object detection modules. However, the car logic to overtake is not implemented.
Do a parallel or perpendicular parking manoeuvre on an empty parking spot (Random position).	Major	60%	The car can recognized parking signs, segment parking locations and perform parking protocol. However, this module is not integrated with the whole system yet.

Continued on the next page ...

Continued from the previous page.

Obstacle description	Priority	Completion	Status / Issues
Pass the ramp	Minor	15%	The car can process IMU signals at a defined time interval. However, the data is still noisy and haven't been process yet.
Start with the green traffic light	Minor	15%	The car can received semaphores status via UDP. However, the car logic to process is not implemented.

Table 3: General status of the project

5 Upcoming activities

Sprint 3/Contents	Responsibility	Requirement Description
Motion planning	Tien & Nhat	**Roundabout handling:** The car can detect when a roundabout is encountered. New sequential logic will be implemented to ensure the shortest route and navigate roundabout. **Map graph traversal:** The map is provided in an XML graph. Besides the current behaviour logic, a graph traversing module need to be implement and optimizer to run the entire map. **Ramp handling:** With IMU data and depth information (monocular methods), a ramp handling should be implement with 2 main activities (acceleration and deacceleration on entering and leaving). Real world test case will have to be run to ensure the car can still follow the planned path after passing the ramp. **Intersection handling with sensor, server data:** The current intersection module only relies on visual data from cameras. This will be problematic when noise are introduced. A sensor based method will be investigate to ensure the visual data doesn't interrupt the action of the car. **Car overtaking:**
Object and lane detection	Hai & Quang	**Real-time optimization:** The system is currently processing both object and lane at 11fps. We intend to push this to 30fps via overclocking the TPU and optimizing the architecture. **Sensor fusion:** The current perception node only depends on image frame from camera. This can be unreliable as camera artifacts are introduced. We proposed investigating sensor fusion (IMU, GPS, LiDAR) to take advantage of geometric data, instead of just relying on photometric data. **Multi-task model investigation:** The current perception node includes many models working in parallel (YOLOv5, MiDAS) to localized and depth estimate various objects. A multi-task model may relieved computations by learning to directly regress the necessary information. This will be heavily investigate and prototype in the comming sprint.

Table 4: Upcoming activities description