Yihao Qian

Team A: Aware

Teammates: Amit Agarwal Harry Golash

Menghan Zhang Zihao (Theo) Zhang

ILR09

March.23, 2016

Individual Progress

During the last week, I was in charge of building the object tracking system.

Object Tracking:

The requirements for object tracking are as follow:

1. Tracking multiple objects
2. Work in real time
3. Track pedestrians and vehicles

I searched online for different kinds of object tracking algorithms. I compared the result using different algorithm (MIL, boosting, KF) for object tracking. Based on our requirements, I choose Boosting as our object tracking algorithm. This is an algorithm that can work in real time. You may find how algorithm works in the following paragraph.

1. For each object’s bounding box, the algorithm first extracts distinct features in the bounding box.
2. The algorithm builds a decision tree to learn those distinct features.
3. For t+1 frame, the algorithm sends multiple potential ROIs to decision tree, and judges whether the object is in the ROI or not.
4. Update position of the bounding box
5. Learn new feature in the ROI region
6. Repeat procedure 2-5

I applied this algorithm on all the test videos in MOT (Multiple object tracking) bench mark, in conclusion, there are several disadvantages and advantages.

Disadvantages:

Scale variant: This is due to the limitation of algorithm. The algorithm can't change ROI's bounding box's size during tracking. That's the reason why the size of bounding box stays the same during the tracking process, and only cover a small portion of car. I do believe this kind of limitation would result error when the car changes size. You may find some experiment result in Figure 1-3.

Rotation variant: This is due to the limitation of algorithm. The algorithm uses decision tree to decide whether the object is in the object. The decision tree is a kind of weak classifier, which means it is sensitive to noise and feature change. In order to solve this problem, we can substitute decision tree with other strong classifiers, however, this would cause huge drop for FPS.

Advantages:

Fast: This algorithm could work for more than 20 FPS.

Multiple object tracking: This algorithm could track multiple object at the same time.

Solution:

The algorithm is variant to scale and rotation, I planned to solve this problem using the following pipe line.

1. At frame i, detect the object using object detection algorithm (SSD) & Initial tracker position
2. From i to i+N frame, use Boosting tracking algorithm to track the object
3. At frame i+N+1, detect the object using object detection algorithm (SSD), refresh the ROIs' position and size
4. Repeat procedure 2 & 3

I do believe this way can solve scale and rotation variant. Our (SSD) object detection works very fast, the FPS could reach 50 FPS using a TITANX GPU. We can refresh the object tracker at a high frequency.

Desktop setup:

I installed Ubuntu 14.04.5 on our desktop and helped Zihao installed ROS and camera SDK on our desktop.

Challenges:

System integration:

None of the teammates in the team has a lot of knowledge in CS field. We might face some difficulties when integrating all the system together, especially when there are a lot of dependency libraries in our system. Also, different algorithms have different version of dependency. Let's say our GUI relies on OPENCV2.4. Stereo vision and object tracking system rely on OPENCV3.2, whether we could use both is a problem we are still trying to figure out.

Teamwork:

Zihao researched on integration design in ROS and building image capturing module in ROS. Menghan integrated our object detection system in ROS system. Harry and Amit worked on getting data from radar. They could get and parse data from radar.

In the next week. We will integrate object detection system with object tracking system. The first stage is to build ROS node to publish images captured from two cameras. Zihao and I are going to work together for the first stage. The second stage is to set up the node for our stereo vision system. I am planning to work on that. I am going to spend more time on integration object tracking system with object detection system. I need to write code for our pipeline system.

Figures:

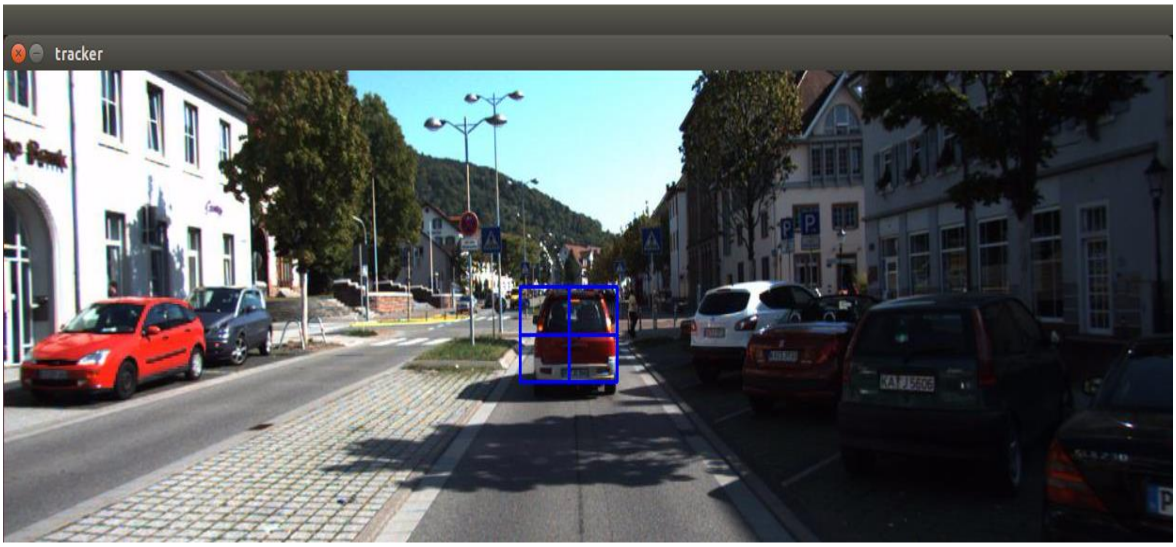

Figure 1. Initial tracker

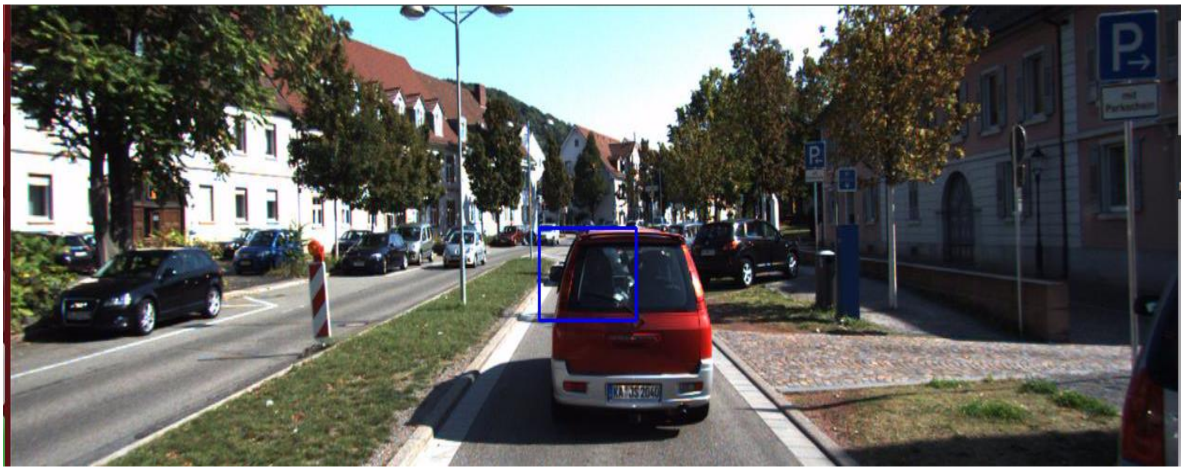

Figure 2. Scale variant

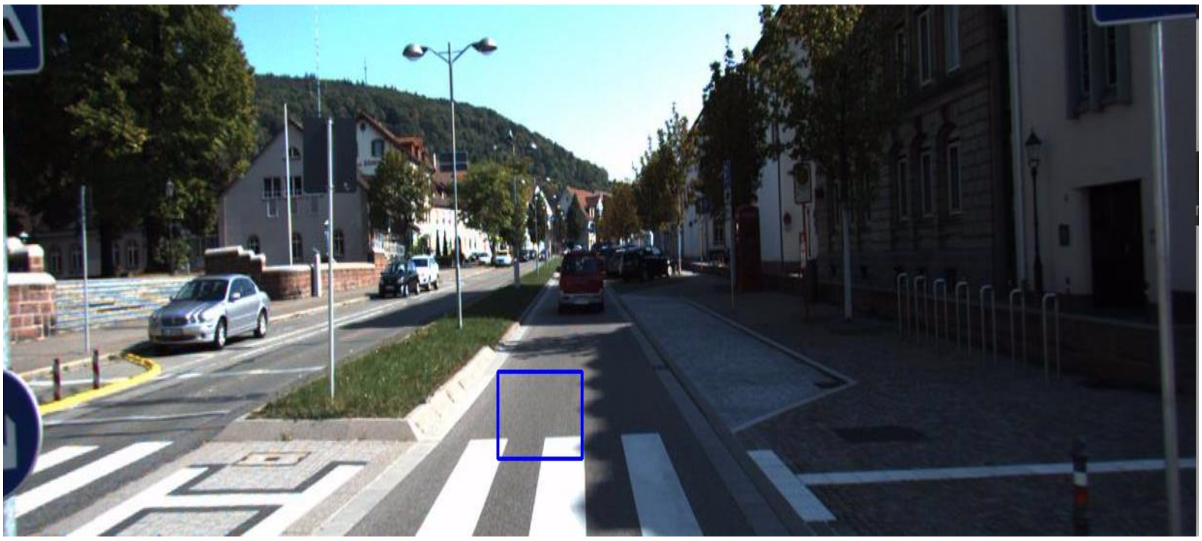

Figure 3. Failure case