

# Object Tracker

Jama Hussein Mohamud

May 15, 2020

## 1 Introduction

Object tracking has been a long-term goal in computer vision and this project will focus on implementing an object tracking system by estimating the trajectory of objects in a sequence of consecutive frames. The idea will be to calculate the Euclidean distance between objects in subsequent frames.

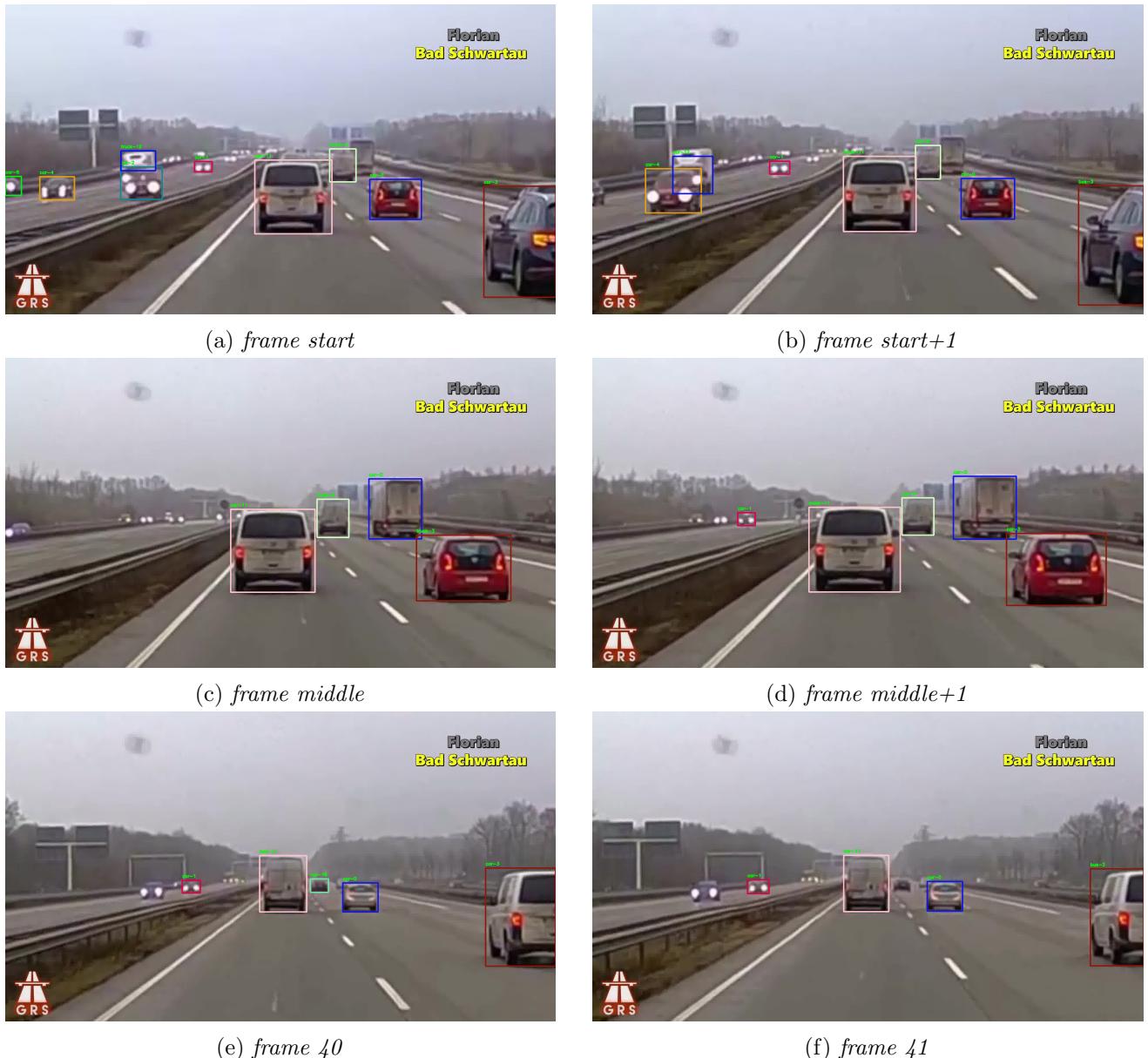


Figure 1: pairs visualization

## 2 Algorithm

Tracking objects could be done via Intersection of Union or Euclidean distance. However, there is a problem of the IOU giving a score of zero if there is no overlap between objects even if the tracked object moves a bit further from its original position. On the other hand, the Euclidean distance tends to calculate the minimum distance between the object in the previous frame and the current frame. Though both tracking strategies do not work with 100 % accuracy, the Euclidean distance turns to perform better.

### How the algorithm works:

- The euclidean distance between the current boxes and previous boxes are calculated.
- For each object in the current frame an object with minimum euclidean distance from the previous frame is chosen.
- Track ID and COLOR are used for tracking.
- Register new objects if they have not been seen before.
- Deregister the object that disappears after one frame, and register it as a new object if it appears again.

The baseline model (i.e. Detectron model) is able to detect an object in a frame but sometimes is unable to detect that same object in a different frame. Therefore, improvement in the Detectron model will highly improve the Euclidean distance algorithm as well. In cases the Detectron is unable to detect similar objects in different frames, the tracking algorithm de-register that particular object and registers it again if it reappears. This issue, therefore, can be resolved by setting a "maxDisappearance" variable and this value should be greater than 0. However, this is not an efficient method because the bounding box plot will sometimes get stuck on the previous frame even if the objects disappear. Aside from this, maxDisappearance of 0 works so well most of the time.



Figure 2: 10 consecutive frames from the video

### **3 Visualizations**

Figure 1 demonstrates a pair of objects from different locations of the video. Similarly, figure 2 shows 10 consecutive frames from the video. As can be seen from the color and the track ID, we can say that the algorithm is doing well on tracking the objects shown. Also, you can find here link a video for all the frames. As can be seen from the video, when color changes after sequence of frames, it is due to detectron's different prediction of the same object or detectron did not detect that object in one of the frames. Similarly, there sometimes exist a lot of overlapping objects, but luckily, thresholding based on detectron's prediction score and box area is used to alleviate this issue. Overall, the algorithm is working quite well with minimum configuration in all frames.

### **4 Conclusion**

Tracking objects in a video or live stream is a challenging problem, particularly when the speed of objects is so fast, or objects have occluded each other. In this lab we showed how euclidean distance helps in tracking objects in video. Finally, I can conclude that The project was fascinating, and I have learned a lot in object tracking.