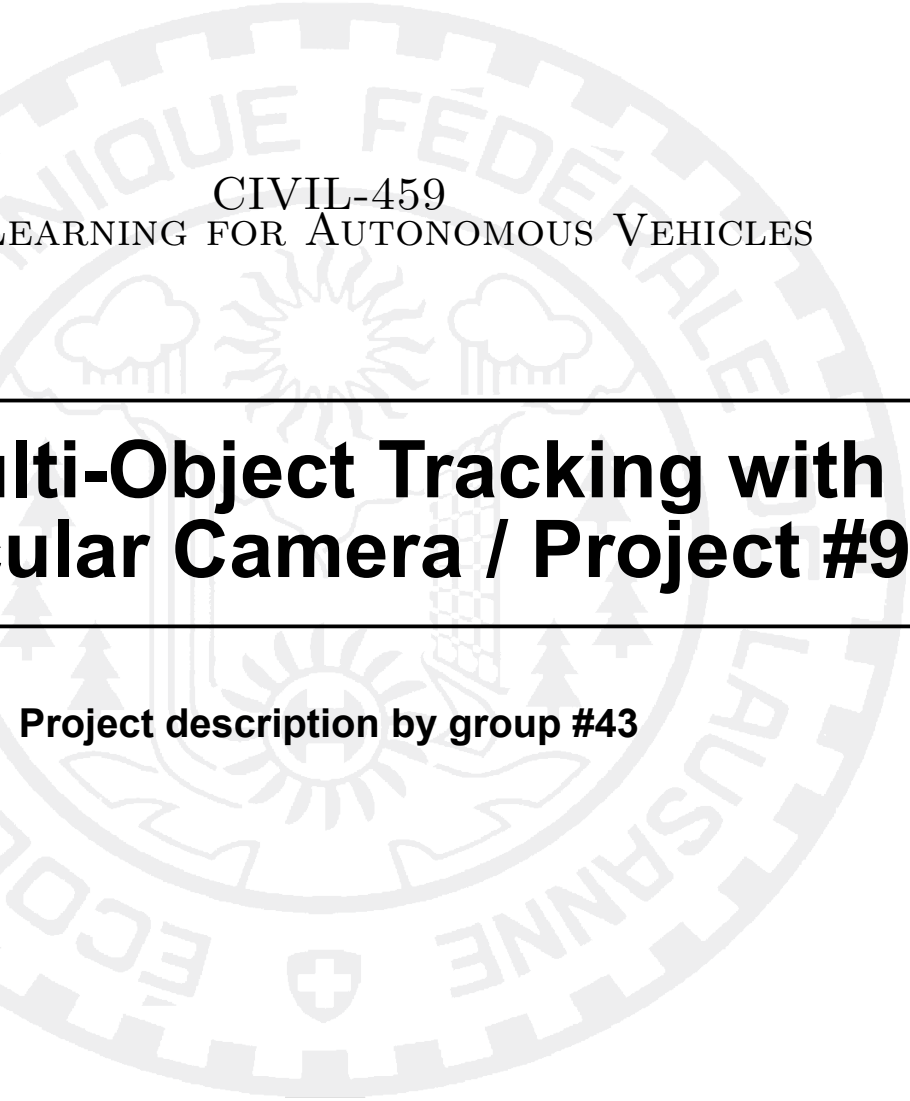

CIVIL-459
Deep Learning for Autonomous Vehicles

3D Multi-Object Tracking with Monocular Camera / Project #9

Project description by group #43

Elias William
SCIPER : 367106

Wilhelm Widlund Mellergård
SCIPER : 361948

April 6, 2023

Contents

Introduction

Tracking an object in 3D space with a monocular camera is a particularly difficult task due to the lack of depth information in an image. Most methods used today are based on accurate depth measurements using expensive sensors, for example LIDAR:s. Due to the high financial cost of these sensors there have been recent studies of using only one camera for both 3D detection and 3D tracking. The ill-posed problem of estimating distance from a monocular image, having no direct depth information, has led research to leverage deep networks as a solution, attempting to learn monocular depth estimation.

The goal of this project will be to combine a state of the art 3D detector with a tracking algorithm to get real-time performance 3D tracking. Also, it is proposed to add a visual appearance matching component to the tracking algorithm such that visual features of objects are taken into account when re-identifying objects from frame to frame.

Overall, the guiding principle of this project will be to improve re-identification performance without reducing inference FPS too much.

1. Problem definition

The project is divided into two main sub-tasks: 3D detection of objects, and re-identification of previously detected objects (tracking). The setting is that of dashcam video feed from a car in traffic being used to detect and track other road users in the immediate vicinity.

1.1 3D detection of objects

The purpose of this sub-task is to detect objects belonging to certain classes in 2D images, and estimate their size, 3D position and orientation. As this area is much more well-explored than 3D tracking, and as there is another project in the course which focuses on 3D detection, this project will use an off-the-shelf 3D detector, and focus on the tracking aspect.

1.2 Re-identification of objects: tracking

In order to track an object over several consecutive images (a video stream), it needs to be re-identified as being the same object as previously seen. Furthermore, objects need to be re-identified even after having not been seen for a number of images, for instance when a vehicle temporarily passes behind another before reappearing in the camera view, a phenomenon called occlusion.

1.3 Inputs

The input available to the project is RGB video, assumed to have a frame rate of 20 FPS.

1.4 Outputs

The outputs to be generated by the project are:

- An unique ID for each unique detected object.
- Key points for each ID:ed object, the nature of which vary depending on class.
- Bounding box, being the smallest possible rectangular cuboid that contains the entire object.

2. Method

Based on a literature study, the EagerMOT[1] method has been chosen as the framework for tracking. It has been chosen on merit of its suitability to the proposed structure of the project, availability of source code and documentation, and performance results.

2.1 Detection method

As the focus of this project will be the tracking aspect, the choice was made to use the 3D CenterPoint[2] and 2D MMDetectionCascade[3] detectors. This is because that combination was used by the EagerMOT team to set their benchmark score on NuScenes. See section 4.2 for further discussion on this topic.

2.2 Tracking method

During the literature study, it became apparent that the number of trackers with available open-source code, that also fulfill all the characteristics of this project, is limited. These characteristics are:

- Compatible with monocular camera detection

- Uses deep learning
- Open-source code available
- Outputs 3D bounding boxes

The lack of suitable off-the-shelf trackers led to the proposed contribution of this project, see section 2.3.

2.2.1 Considered methods for tracking

ByteTrackV2: 2D and 3D Multi-Object Tracking by Associating Every Detection Box.[4] This method combines a YOLOX detector with the authors' implemented tracker. Here, the tracker part alone is considered.

- (+-?) Very new, so ought to have taken recent advances into account, but also not thoroughly tested or replicated by others
- (+) Backward "prediction" complement to Kalman filter, capturing abrupt movements
- (+) Flexible pipeline compatible with other re-identification methods
- (+) Distinguishes between high-score and low-score detections, improving performance for occlusions
- (-) Does not take advantage of object visual attributes.
- (-) Ostensibly open source, but **source code unavailable** at the time of writing this report

TripletTrack: 3D Object Tracking using Triplet Embeddings and LSTM[5]

- (-) **Not open source**
- (+) LSTM motion-learning RNN used for motion prediction re-identification
- (+) Combination of 2D visual appearance (CNN learned features) and 3D appearance (bounding box size) used for appearance re-identification

GNN3DMOT: Graph Neural Network for 3D Multi-Object Tracking with Multi-Feature Learning[6]

- (-) **Not open source**
- (+) Combination of 2D visual appearance (CNN learned features), 3D appearance (bounding box size) and motion used for re-identification

EagerMOT: 3D Multi-Object Tracking via Sensor Fusion[1]

- (+) Open source, well documented
- (+) Adaptable to different detectors
- (-) Does not take visual appearance in to account

2.2.2 Chosen tracking model

The EagerMOT: 3D Multi-Object Tracking via Sensor Fusion[1] was chosen as the base tracking framework used in the project. There are not many 3D tracking algorithms with open-source code, especially for monocular camera setup. To take advantage of the available information in the video stream, the model should preferably leverage visual appearance. Also, to fit the course, deep learning should be present. TripletTrack would be the go-to example of such a model, if the source code was available.

EagerMOT is flexible with respect to how its input data (detected objects in 2D and 3D) are generated, due to its fusion step. The idea is that 2D and 3D data is matched by a Greedy algorithm, an example of sensor fusion being used to benefit from multiple sensors. As such a setup will typically lead to more 2D than 3D objects being detected, resulting in some 2D objects not being fused to any 3D object. For the scope of this project, the fusion feature doesn't add any actual functionality in online operation, as only monocular camera input will be available for the detector, meaning each 3D object will be directly based on a specific 2D object. For purposes of generalizability however, this feature is beneficial.

EagerMOT does not use deep learning, nor visual appearance information, but instead relies on motion estimation using a Kalman filter to make predictions of the position and orientation of the tracklets, as basis for re-identification. The predictions are compared with the detected objects, resulting in a score for each possible pair of tracklet-detection, which is stored in an affinity matrix. Finally, re-identification is done by matching tracklets with detected objects, using a Greedy algorithm on the affinity matrix. This process takes place twice, first for the fused 2D-3D objects, and then for the leftover 2D-only objects. In the 3D case, the score is based on spatial distance and angular difference, while the 2D case is based on IoU overlap.

2.3 Proposed contribution

2.3.1 Additional visual appearance matching in the tracking algorithm

Since EagerMOT does not use deep learning in any way, nor takes visual similarity into account, the proposed contribution is to implement an add-on module that does precisely these things, and combine it with EagerMOT. By adding this functionality, it should be possible to get a more accurate re-identification of the objects. This method is used with good results by other trackers, for example TripletTrack [5].

The idea is to pass the image of each detected object through a convolutional neural network to create a feature vector for that image. The feature vector would then be compared to all tracklets to determine the most likely match of objects. An affinity matrix would be generated, containing visual-based similarity scores for all possible pairs of tracklets-detected objects. This matrix would be returned by the module, and then be combined with the affinity matrix of EagerMOT. The result would be to get an affinity matrix that contains information about both the visual similarity and the position matching of the tracklets and detected objects. Primarily, this should serve to improve the accuracy in re-identifying objects that have been tracked previously, after a period of occlusion.

2.4 Overall structure

The overall pipeline resulting from the inclusion of the proposed module can be seen in figure 1. The only intrusion into the EagerMOT internal pipeline is the combining step of the affinity matrices, demonstrating the ease of implementing the module into any existing tracker. In a nutshell, the only requirement would be that a tracker stores possible matching scores in an affinity matrix, which is a very common practice.

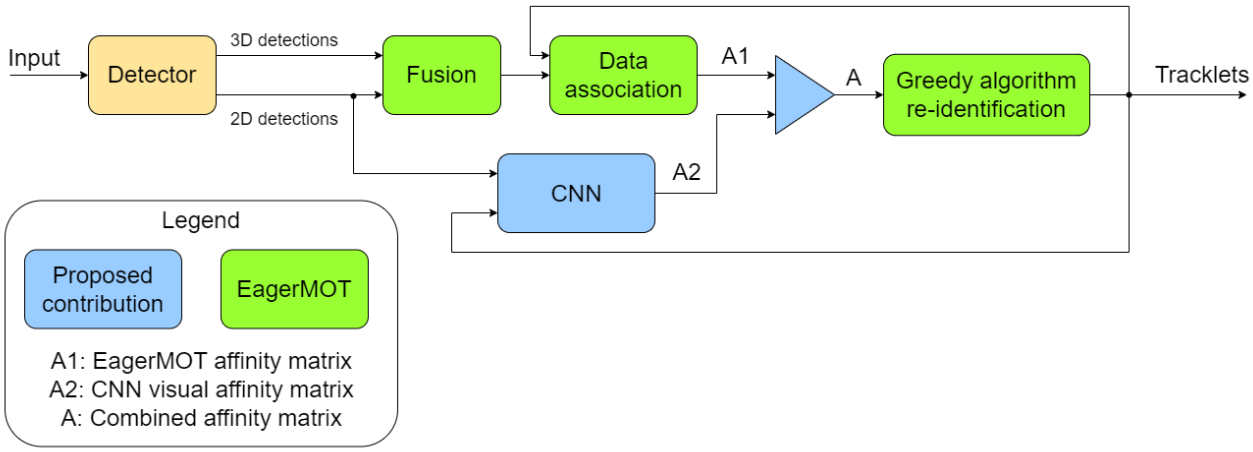

Figure 1: Overall pipeline

3. Dataset

The NuScenes dataset[7] is chosen for the project. It consists of 1000 driving scenes in Boston and Singapore. Each scene consists of a 20 seconds long video.

3.1 Training dataset

A subselection of the NuScenes training dataset will be made to fit the intended project structure. Not all the classes are considered, as some of them are outside the scope of this project. Since the focus is to detect and track other road users, some classes are ignored, and others are combined. The selection is done according to the class definitions in the NuScenes tracking benchmark[8], due to this being both an appropriate class set for the task at hand, and for purposes of evaluation, see 4.1. For training the CNN a re-identification dataset will be created from the NuScenes dataset.

3.2 Testing dataset

The NuScenes testing dataset will be used for evaluation, as this dataset is commonly used for this purpose, enabling comparison of performance for each of the chosen focus metrics in the following section.

4. Evaluation

4.1 Evaluation metrics

Several metrics are commonly used to test 3D MOT models, with Average Multi-Object Tracking Accuracy (AMOTA) being the main one in the NuScenes tracking benchmark[8]. This metric is the average of the Multi-Object Tracking Accuracy (MOTA), taken over different threshold levels for Recall, which reduces the effect of detection confidence boundary values. However, concerns have been raised that AMOTA focuses too much on detection and not enough on association. To ameliorate this, the HOTA[9] metric has been proposed. It aims to balance the effect of performing accurate detection, localization and association, providing a single metric for tracker evaluation.

Another tracking evaluation metric is IDS, which records the amount of identity switches: mistakenly

applying a new ID to a previously tracked object, for instance due to failure of recognizing the object after an occlusion. As speed during inference is important to an autonomous vehicle that depends on up to date information, another important metric for this project will be the FPS at which it operates.

4.2 Evaluation strategy

Due to the proposed contribution of extending the EagerMOT model, the evaluation strategy will be to monitor AMOTA and FPS, while targeting the HOTA and IDS metrics for improvement. AMOTA should ideally improve aswell, but FPS will invariably drop by some amount due to more computations being done in each cycle. A successful implementation would be characterized by maintained or improved AMOTA, insignificantly reduced FPS, and foremost by improvements to IDS and HOTA.

The AMOTA and IDS scores can be compared against those posted on the NuScenes leaderboard. This is the primary reason for selecting the off-the-shelf detector as the same one used by the EagerMOT team: for the comparison to make sense, everything except the added module must be the same. Since FPS depends on the hardware running the evaluation, there will have to be separate runs of both the modified and unmodified EagerMOT model on the same machine for a comparison of this metric to be made. As HOTA is not recorded on the NuScenes leaderboard, it can also be recorded from the same runs as for FPS, providing a benchmark for evaluation.

Bibliography

[1] Kim, Aleksandr, Ošep, Aljoša and Leal-Taixé, Laura *EagerMOT: 3D Multi-Object Tracking via Sensor Fusion* https://github.com/aleksandrkim61/EagerMOT, fetched April 4th 2023.

[2] Yin, Tianwei and Zhou, Xingyi and Krähenbühl, Philipp *CenterPoint: Center-based 3D Object Detection and Tracking* https://github.com/tianweiy/CenterPoint, fetched April 6th 2023.

[3] *MMLab Detection* https://github.com/open-mmlab/mmdetection, fetched April 5th 2023.

[4] Yifu Z. et al. *ByteTrackV2: 2D and 3D Multi-Object Tracking by Associating Every Detection Box* https://arxiv.org/pdf/2303.15334.pdf, fetched March 31st 2023.

[5] Marinello, Proesmans, Van Gool. *TripletTrack: 3D Object Tracking using Triplet Embeddings and LSTM* https://openaccess.thecvf.com/content/CVPR2022W/WAD/papers/Marinello_TripletTrack_3D_Object_Tracking_Using_Triplet_Embeddings_and_LSTM_CVPRW_2022_paper.pdf, fetched March 29th 2023.

[6] Xinshuo W. et al. *GNN3DMOT: Graph Neural Network for 3D Multi-Object Tracking with Multi-Feature Learning* https://arxiv.org/pdf/2006.07327.pdf, fetched April 4th 2023.

[7] Nutonomy*NuScenes Dataset* https://www.nuscenes.org/nuscenes#overview

[8] Nutonomy *NuScenes Tracking Task* https://github.com/nutonomy/nuscenes-devkit/tree/master/python-sdk/nuscenes/eval/tracking/README.me, fetched March 31st 2023.

[9] Luiten J. et al. *HOTA: A Higher Order Metric for Evaluating Multi-Object Tracking* https://arxiv.org/pdf/2009.07736.pdf