Video Surveillance for Road Traffic Monitoring

Gemma Alaix Granell
Universitat Autònoma de Barcelona
gemma.alaix@e-campus.uab.cat

Josep Brugués i Pujolràs
Universitat Autònoma de Barcelona
josep.brugues@e-campus.uab.cat

Clara Garcia Moll
Universitat Autònoma de Barcelona
clara.garciamo@e-campus.uab.cat

Aitor Sánchez Abellán
Universitat Autònoma de Barcelona
aitor.sancheza@e-campus.uab.cat

Abstract

In recent years, video surveillance gained popularity in computer vision due to the wide range of applications it can carry on. Specifically, for urban traffic optimization, a necessity to advance in multi-target multi-camera (MTMC) tracking emerged. In this project, we propose and contrast different methods for multi-target single-camera (MTSC) and multi-target multi-camera (MTMC) vehicle tracking, tackling the third track of the AI City Challenge [9].

Our source code is public and can be found at: https://github.com/mcv-m6-video/mcv-m6-2021-team3.

1. Introduction

Video surveillance for road traffic calls for a reliable Multi-Target Multi-Camera tracking system to automatically track multiple cars through a network of cameras. In order to advance with the research focused on the development of intelligent transportation systems, a 5^{th} edition of the CVPR 2021 AI City Challenge proposed five tracks: vehicle counting using IoT Devices, City-Scale Multi-Camera vehicle Re-Identification, City-Scale Multi-Camera vehicle tracking, traffic anomaly detection, and Natural Language-based vehicle retrieval. The target of this project is the third track, City-Scale Single and Multi-Camera tracking.

The contributions supplied to the aforementioned problems are 1) Hand-crafted and deep learning feature-based detection and tracking methods within a single camera; 2) Re-identification color histogram-based, of targets across multiple cameras; 3) End-to-end detection, tracking, and re-identification deep learning approach for Multi-Target Multi-Camera (MTMC) tracking.

2. Related Work

In this section we cover some of the approaches that have already been implemented in regards of this project.

2.1. Single Camera Tracking

Few years ago, single camera tracking was a hot topic, especially due to the difficulty of dealing with occlusion and the challenge presented to do multi-target matching between frames. To face the problem of partial occlusions, Chu et al. [3] came up with constrained weighted kernel histograms. However, due to the irregular occlusions presented on vehicle surveillance, other methods should be introduced to address the task. Therefore, 3D deformable vehicle models are used to define 3D kernels by Lee et al. [6], which was later improved by Tang et al. [13] introducing camera calibration information. Furthermore, most recent approaches are based on machine learning, such as Deep SORT by Wojke et al. [14].

2.2. Multi Camera Tracking

Nowadays, with the constant development of urban intelligence, the study of multi camera tracking is increasing. Some of the methods proposed are based on re-identification. For instance, Peilun et al. [7] propose an algorithm where, by using a spatio-temporal consistency and hierarchical matching method, tries to overcome the issue. Moreover, other challenges such as low resolution, variation of illumination and complex background produce the emergence of other methods like NCA-Net [12], which is implemented to overcome the multiple objects tracking problem across multiple cameras.

3. Dataset

The dataset used to fulfill both tasks proposed in this project is the one suggested in the **AICity Challenge**, specifically the dataset for *Track 3: City-Scale Multi-Camera Vehicle Tracking* [9].

In particular, this dataset is set up by videos captured from multiple traffic cameras placed in a city in the United States. In each of the sequences there are multiple cameras in which the scene is captured from different points of view. Moreover, for each camera it is provided a ground truth file with the different annotations for the bounding boxes and tracking ids, as well as its GPS location and camera calibration information.

4. Multi-Target Single-Camera Tracking

Following the state-of-the-art of most MTSC tracking methods, this project is built on the tracking-by-detection paradigm. In the performed experiments, the problem is addressed in a straightforward way. First, background subtraction methods are used, including adaptive background subtraction, MoG (Mixture of Gaussians) [16, 17] and KNN (K-Nearest-Neighbours) methods. Then, a boost in performance is achieved by the introduction of well-known detectors such as YOLOv3 [10], SSD [4] and EffientDet [11]. For all detectors, the pre-trained model on COCO benchmark [8] is used as the starting point to fine-tune.

4.1. Detection processing

The annotations provided by the dataset only include moving objects information. In order to get rid of static objects from our detections, a processing over the bounding boxes needs to be added. The processing is assessed by two filters:

- **Close to ROI cars.** Cars out of the ROI, or close to its border, would not be taken into account.
- **Static (parked) cars.** Once tacking is computed, if the displacement of a car is less than a threshold, then it is considered to be parked.

4.2. Tracking

Once detection has been properly solved, the next step is to track them. The basic goal here is to assign IDs to detections. All the bounding boxes which belong to the same car should obtain the same ID. Moreover, IDs should be unique, meaning that detections which do not belong to the same vehicle should never coincide.

To tackle that task, different solutions are proposed, from simpler to more elaborated approaches.

4.2.1 Maximum Overlapping

As a first naive implementation for detection tracking, maximum overlapping is used. Firstly, a unique ID is assigned to each detection at the first frame of the sequence.

From that point the sequence is iterated frame by frame, for each of them:

1. At frame N, each detection is assigned with the ID with the highest overlap (IoU) in frame N-1 as long as this value is higher than 0.5.
2. If there is no match, previous frame is considered for comparison. This can be repeated up to frame N-5.
3. If there is match, the corresponding ID is assigned and, if necessary, detections are interpolated between the intermediates frames. Otherwise, assign a new ID to the bounding box.

4.2.2 Maximum Overlapping + Optical Flow

A brief variation from previous method is the one which optical flow information is used to refine detections. The idea is that by computing the optical flow between frame N and Frame N-1, that information can be used to anticipate the bounding box displacement along them.

MaskFlowNet [15] is used to compute the Optical Flow(OF). Then, for each detections, OF values within the bounding box are used to compute detection displacement. Considering this improved detection at frame N, the algorithm exposed on 4.2.1 is performed.

4.3. Kalman

Finally, the third implementation utilises the Kalman algorithm [2] in order to estimate the tracking. This method predicts the position of the bounding box at frame $t + 1$ by following a linear motion model. In order to do it, it is proposed to use Simple Online and Real-time Tracking (SORT) [1], which is an approach to multiple object tracking, where the objects are detected in each frame and represented using bounding boxes.

4.4. Enhancement

Not only interpolation is used to boost tracking performance, but also other post-process techniques are implemented as well.

- **Noise filtering.** After the tracking is performed, all the trajectories which less than 4 detections after interpolation are discarded.
- **Parked cars.** As mentioned in 4.1, distance between the first and the last detections of each trajectory is computed. Thus, filtering which are considered to be static cars.

5. Multi-Target Multi-Camera Tracking

5.1. Color Histogram

Color-based tracking had been widely used due to its efficiency and robustness in many cases. Our aim is to test its performance when it comes to multi-camera tracking. In order to assess the problem, a naive approach is proposed relying on color histograms. The approach can be split into two main sections: single-camera descriptors and multi-camera re-identification.

In the first place, as explained in section 4, multi-target tracking for every camera is computed. With the information that this provides, it is possible to obtain car descriptors for every single camera: mean values of 2D histograms at different color spaces (RGB, HSV, Lab and YCrCb).

The next step is focused on the process of matching cars among different cameras of the same sequence, i.e. re-identification. This problem has been tackle from two different perspectives:

1. **Distance matrix.** This method assumes that the camera with the largest number of ids contains all the possible ids present in the scene. Then, histograms from such camera are compared to the ones from the rest of the cameras, creating a distance matrix for each. Three different distance measurements have been tested: euclidean, L1 and Bray Curtis distance. Matches are then extracted from the generated matrices by minimizing the global distance as a linear sum assignment problem.

2. **Clustering.** The histograms from all cameras are stacked into a single space from which clustering is applied. The clusters used are either K-Means or Gaussian Mixture Models. The cluster determines the id assigned to every car.

5.2. Spatio-temporal Consistency and Hierarchical Matching

For a more advanced methodology we used the approach proposed by Peilun et al. [7] in the AICity Challenge 2018. The full pipeline consists on two parts: Single camera tracking and Multi-target multi-camera tracking with a vehicle re-identification (ReID) approach.

In the first stage of the pipeline, the feature extraction and the single camera tracking is performed. In more detail, the steps are as follows:

1. Detection of cars in all available frames in the sequence using an object detection neural network.

2. Feature extraction: for each car, compute its features: on the one hand, its GPS position (latitude, longitude); on the other hand, its ReID features from a ResNet50 CNN. The network has been trained using 333 different identities from the AI City Challenge dataset. From those, 149 have been used for training and 184 for validation.

3. For each camera in the sequence, perform single camera tracking. The Hungarian algorithm [5] is used to find the cars that have the smallest loss, i.e. that have the most similar features.

The final stage on the pipeline, which performs the multi-target multi-camera tracking, consists on two steps:

1. Hierarchical matching of tracks from different cameras based on ReID features. High similarity features are matched together. The algorithm also takes into account different scenarios, depending on the location of the cameras:

   - In crossroad scenes, the car should appear on multiple cameras at the same time.
   - On other scenes, it is not required for the car to appear in multiple cameras at the same time, as cameras are separated from one another.

2. Post processing of the tracks that have not been matched. Here, the GPS features are used to match the tracks with the highest overlapping on their trajectories.

6. Results

In this project only one aforementioned dataset is used to verify the effectiveness of the proposed methods. Since each scene of the dataset proposes different scenarios, the results obtained are really variable, making impossible to determine which is the method that clearly works better.

6.1. MTSC Tracking

In order to solve Multi-Target Single-Camera tracking task, it is followed the method advanced in Section 4, where different approaches to solve the tracking part are suggested. First, by using 2 of the 3 sequences of the dataset the Object Detection model (YoloV3 from ultralytics) was trained. Then, the model was evaluated with the remaining sequence in order to get the detections that will be used to address the tracking problem. Once this problem was solve, it was evaluated using IDF1 metric. The quantitative results obtained for each sequence are presented in Table 1, where it could be observed the higher similarity between the results obtained using only Maximum Overlap or using Maximum Overlap and Optical Flow, along the different sequences. However, the difference between these 2 methods is presented in terms of computational cost, where the second one is much more expensive than the first one. The

Technique	**Sequence 1**	**Sequence 3**	**Sequence 4**
Maximum Overlap	0.4297	0.7323	0.559
MO + OF	0.4196	0.7347	0.558
Kalman	0.5087	0.7303	0.541

Table 1. The average IDF1 results of each sequence using Yolo detections.

computational cost for each tracking estimation method is presented in Figure 1.

Since, the methods proposed by us work under the constraint of following a linear motion model, when the method is evaluated with Sequence 3, the results obtained are the highest. This is due to the fact that all the cameras in this sequence point to a single straight street, thus the motion of the detections would be almost linear. This is shown in 2, where the image in the left is a frame of the Sequence 1 that contains detections trajectories that do not follow a linear motion. Thus, if the detection is lost, when it is recovered the ID changes because the bounding box predicted (that follows a linear motion model) are not placed at the same position than the real detections. So, they are considered as different detections. However, in the right image, the trajectory of the detection follows a linear motion, thus the tracking is estimated accurately. This could explain the qualitative results presented in Table 1.

6.2. MTMC Tracking

Color histogram approach 5.1, although it is efficient, shows some weaknesses. Histograms result in a poor descriptor for a large number of cars, being sensitive to changes of illumination and perspective. Moreover, applying a mean over car instances does not generalize well features since it does not consider its 3D structure. Additionally, the camera with the largest number of cars does not necessarily include all the cars in the scene. And, as a final observation, spatio-temporal information should be taken into account, since the clustering technique in some situations cars can appear at the same time instant with repeated identification.

The approach introduced in Section 5.2 is tested using different configurations both for detecting cars on the images and for extracting deep features with a CNN. The IDF1 score, as well as the precision and recall scores are presented in Table 2 for each sequence in the dataset.

As it can be seen from the table, Mask-RCNN outperforms in all the experiments the YoloV3. This may be due to two factors. On the one hand, due to different training processes. Since no information is provided on the training procedure of the Mask-RCNN, it is difficult to get conclusions from this. On the other hand, the ReID feature are extracted from a ResNet50 that has been trained with cars from the challenge dataset. Those cars are cropped from the frames using the detections obtained with the Mask-RCNN. By analyzing some images, we have seen that the YoloV3 networks makes better detections, i.e. the bounding boxes are more adjusted to the cars, thus making the cropped cars to be different from the ones that the ResNet50 has been trained.

In order to understand better the results obtained, some qualitative results are presented in Figure 3 of appendix B. What it is observed are 3 frames from different cameras of the same sequence. In the top left and in the bottom images, the same car is detected with the same ID, which is to be expected. Although, the ID in the car of top right image is the same of the other 2 frames, the car is completely different and this is one of the reasons of why the results are worse than in single camera tracking. Furthermore, as in MTSC tracking, some detections are lost in some frames, fact that is also reflected in the scores obtained.

7. Conclusions

The desired results have been obtained not only in what concerns tracking metrics but also in extensive experimentation with emerging methods in the field of video analysis. It has been shown that it is possible to perform MTMC tracking using different approaches successfully.

However some drawbacks and analysis need to be addressed:

- Depending on the Sequence, the methods that work better may differ. There is no method that is the clear best performing. Each situation requires a specific approach to maximize results.
- The limitations of using image sensors are clear. Although there is still room for improvement, there are situations where including more information from other sensors such as GPS, Radar of LiDAR, may improve the results.
- MTMC tracking is a challenging task. On the one hand, the tested hand-crafted techniques are way too simple for such a complex task. On the other hand, when applying the more advanced methods, different techniques must be applied depending on the scene, cameras, etc. in order to get good and consistent results across all situations.

Configuration	Scene	IDF1	Precision	Recall
MaskRCNN and ResNet50 (ReID)	Sequence 1	0.2112	0.4647	0.4358
	Sequence 3	0.4318	0.387	0.7703
	Sequence 4	0.2951	0.3478	0.7463
MaskRCNN and VGG16 (ImageNet)	Sequence 3	0.3377	0.4385	0.7664
	Sequence 3	0.4599	0.3446	0.9243
	Sequence 4	0.2778	0.326	0.8126
YoloV3 and ResNet50 (ReID)	Sequence 4	0.1248	0.5647	0.1979
	Sequence 3	0.3314	0.3653	0.4275
	Sequence 4	0.1968	0.3954	0.3932

Table 2. The average IDF1, precision and recall scores of each sequence using different configurations

References

[1] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pages 3464–3468. IEEE, 2016. 2

[2] FR Castella and FG Dunnebacke. Analytical results for the x, y kalman tracking filter. *IEEE Transactions on Aerospace and Electronic Systems*, (6):891–895, 1974. 2

[3] Chun-Te Chu, Jenq-Neng Hwang, Hung-I Pai, and Kung-Ming Lan. Tracking human under occlusion based on adaptive multiple kernels with projected gradients. *IEEE Transactions on Multimedia*, 15(7):1602–1615, 2013. 1

[4] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017. 2

[5] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. 3

[6] Kuan-Hui Lee, Jenq-Neng Hwang, and Shih-I Chen. Model-based vehicle localization based on 3-d constrained multiple-kernel tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(1):38–50, 2014. 1

[7] Peilun Li, Guozhen Li, Zhangxi Yan, Youzeng Li, Meiqi Lu, Pengfei Xu, Yang Gu, Bing Bai, Yifei Zhang, and DiDi Chuxing. Spatio-temporal consistency and hierarchical matching for multi-target multi-camera vehicle tracking. In *CVPR Workshops*, pages 222–230, 2019. 1, 3

[8] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2

[9] Milind Naphade, Shuo Wang, David C Anastasiu, Zheng Tang, Ming-Ching Chang, Xiaodong Yang, Liang Zheng, Anuj Sharma, Rama Chellappa, and Pranamesh Chakraborty. The 4th ai city challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 626–627, 2020. 1, 2

[10] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[11] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 2

[12] Yihua Tan, Yuan Tai, and Shengzhou Xiong. Nca-net for tracking multiple objects across multiple cameras. *Sensors*, 18(10):3400, 2018. 1

[13] Zheng Tang, Gaoang Wang, Tao Liu, Young-Gun Lee, Adwin Jahn, Xu Liu, Xiaodong He, and Jenq-Neng Hwang. Multiple-kernel based vehicle tracking using 3d deformable model and camera self-calibration. *arXiv preprint arXiv:1708.06831*, 2017. 1

[14] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649, 2017. 1

[15] Shengyu Zhao, Yilun Sheng, Yue Dong, Eric I Chang, Yan Xu, et al. Maskflownet: Asymmetric feature matching with learnable occlusion mask. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6278–6287, 2020. 2

[16] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 2, pages 28–31. IEEE, 2004. 2

[17] Zoran Zivkovic and Ferdinand Van Der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, 27(7):773–780, 2006. 2

A. Multi-target, single camera tracking

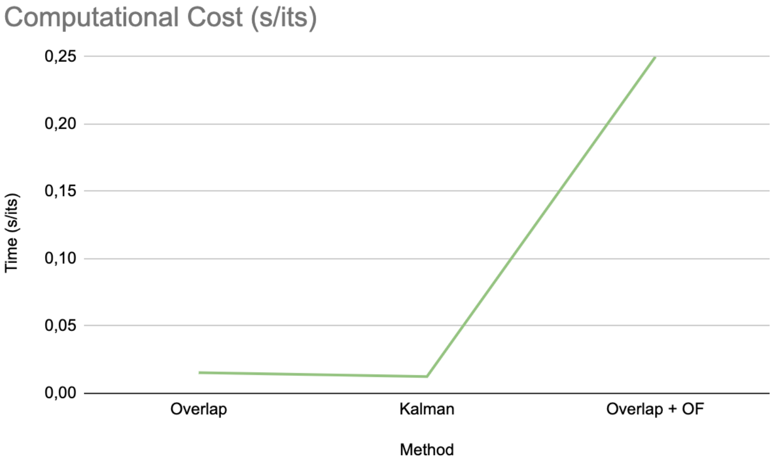

Figure 1. Computational cost for each estimation tracking method.

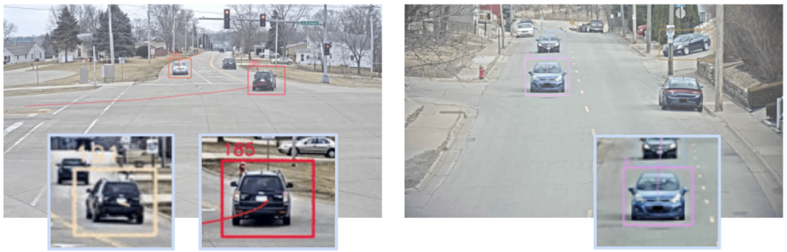

Figure 2. From left to right, frame of Sequence 1, where ID changes, and frame of Sequence 3, where the ID is the same along all the trajectory.

B. Multi-target, multi-camera tracking

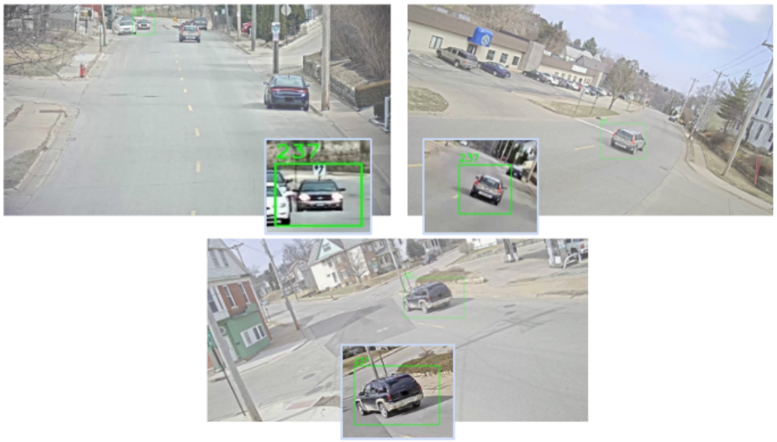

Figure 3. Cameras 10, 11 and 13 of sequence 3. The algorithm assigns correctly the same ID to the same car on cameras 10 and 13, but fails in camera 11.