Video Surveillance for Road Traffic Monitoring
Universitat Politècnica de Catalunya

Daniel Fuentes
daniel.fuentes@e-campus.uab.cat

Daniel Yuste
daniel.yuste@e-campus.uab.cat

Sergi García
sergi.garciasa@e-campus.uab.cat

Isaac Perez
isaac.perezs@e-campus.uab.cat

Abstract

This report goes through the multi-target single-camera tracking and multi-target multi-camera tracking algorithms that we implemented during the project of Module6. In the field of Video Surveillance for Road Traffic Monitoring we can find a growing interest, in part due to the fact that challenges such as 2021 AI City Challenge [1, 2, 3, 4, 5, 6, 7]. This challenge provides a dataset that was mainly used to achieve our experiments with the different implementations. All the metrics used to evaluate the performance were mAP and IDF1 The code with the implementations explained in this paper and the plot of the different metrics can be found on GitHub [1].

1. Motivation

In the last years, an interest of how to build Smart Cities and use information from several devices has increase tremendously. This interest includes the capabilities of Internet of Things (IoT) devices in combination with 5G technologies to bring information received from different sensors and be able to make decisions based on real data.

One of the aims behind this sensorization of the cities is to make transportation more efficient and sustainable. With the information provided by on-the-market sensors, is it possible to make public transit systems safer, smarter at the time they can assist on traffic control related tasks.

This sensorization and understanding of what is happening at the moment might be beneficial in the future with the increase of self-driving cars in the roads.

Vehicle detection and tracking is a challenging task in which the aim behind is a better understanding of the flow of vehicles within the city's ecosystem of roads. Vehicle ReID is important for intelligent transportation systems (ITS) of the smart cities.

Furthermore, safety and infrastructure investment can also get the benefit of a better understanding of what happens on their roads in order to drive their decisions.

AICity Challenge

This challenge arises from the idea of making traffic systems smarter. The challenge tasks are proposed with different datasets provided as a set of videos or images.

- City-scale multi-camera vehicle tracking
- City-scale multi-camera vehicle re-identification
- Traffic anomaly detection: Leveraging unsupervised learning to detect anomalies such as lane violation, illegal U-turns, wrong-direction driving, etc...

Dataset

AICity Challenge provides a multiple traffic camera recordings from streets of cities in the United States.

- 3.25 hours of synchronized videos synchronously captured from multiple vantage points at various urban intersections and along highways.
- The videos are captured at 960p or better, 10fps. Detections for each frame provided by object detection Neural Networks: Mask RCNN, YOLOv3, SSD512.
- Data about the provided videos, including GPS location, camera calibration information and other derived data from videos.

[1] https://github.com/mcv-m6-video/mcv-m6-2020-team6

- Ground Truth for each frame and region of interest for each camera. 229,680 bounding boxes for 666 distinct annotated vehicle identities.

2. Related work

Re-identification (ReID) is widely studied in the field of computer vision. This task possesses various important applications. Most existing ReID methods are based on deep learning. Recently, CNN-based features have achieved great progress on vehicle ReID, outperforming any previous baseline using handcrafted features.

On the other hand, some methods focus on exploiting viewpoint-invariant features, e.g. 2D key-points features. One example is the work of Tang et al. [6] which embeds key-points, heat-maps and segments from pose estimation into the learning pipeline of vehicle ReID, guiding the network to pay attention to viewpoint-related information.

2.1. Multi-target single-camera (MTSC) tracking problem

When we talk about MTSC we can find quite interesting projects [8] like one that talks about Unmanned Aerial Vehicles (UAVs) which are still gaining popularity in civilian and military applications, as much as for personal use. Such emerging interest is pushing the development of effective collision avoidance systems. Such systems play a critical role UAVs operations especially in a crowded airspace setting. Because of cost and weight limitations associated with UAVs payload, camera based technologies are the de-facto choice for collision avoidance navigation systems. This requires multi-target detection and tracking algorithms from a video, which can be run on board efficiently.

2.2. Multi-target multi-camera (MTMC) tracking

MTMC is a challenging problem in computer vision. By combining the trajectory information from multiple cameras, it overcomes the limitation of the field of view in a single camera and allows people to conduct a more detailed analysis of the target. This technology can be used in video surveillance, sports analysis and many other fields. However, compared to single camera tracking, multi-camera tracking has less location information and less strict time constraints, which makes it even more difficult [9].

Some previous solutions [10] to solve the multi-target multi-camera (MTMC) took into account the GPS-trajectory based on the camera coordinates and extra information obtained from the place where the camera was installed with Google Maps. One of the challenges we faced when evaluating cars trajectory is that images are not coordinated in time, so there's a little time-lapse between what is captured from one camera to another, and perspective is not fully coincident.

The proposed solution will need to consider:

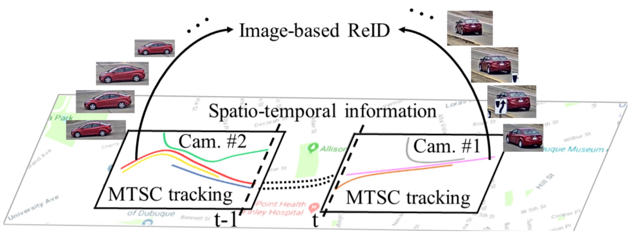

Figure 1. MTMC approach using previous MTSC algorithms

- Cars detection
- Tracking
- Re-identification (ReID)
- The spatial-temporal difference between cameras
- Time differences between recordings

3. Multi-Target Single Camera Tracking

In this paper we try to solve the MTSC tracking using a maximum overlap algorithm that is divided in the following steps:

- Get all the detections of the current frame
- Start a counter. If the counter equals to four we assume that it is a good detection, and we save the track. (To avoid false tracks)
- Compare the new detection with the previous tracks computing the IoU and save the best match
- If not match, start a second counter. If the counter equals to four we assume that the car in the previous frame is not in the new frame, and we remove the track.

But before executing it, data needs to be preprocessed. The reason is that ground-truth data takes into account only the car in movement and the cars that appears in more than one camera. The results increase between two and three times thanks to this preprocess.

To deal with this inconvenient it was necessary to establish a threshold to avoid the parked cars and remove the part of the frames in which the cars are only seen by one camera.

3.1. Optical Flow

To improve the tracking results with the maximum overlap algorithm, optical flow algorithms have been used to correct bounding boxes directly obtained.

The idea behind these techniques is to compute the optical flow between frames and make a correction on the four coordinates of the bounding box in the same direction as the movement goes, expecting to see a bounding box in the given direction based on previous frames knowledge.

It was implemented with the Gunnar-Farneback Optical Flow method that is a dense method based on Polynomial Expansion[11].

In dense optical flow, we take a look at all the points and detect the pixel intensity changes between the two frames, resulting in an image with highlighted pixels, after converting to HSV format for clear visibility.

DeepSort algorithm [12] has been used with Detectron2 [13]. Compared to others solutions based on flow network formulas [14] and probabilistic models [15], the implementation uses recursive Kalman filters and frame-by-frame data association.

Together with the Hungarian method, in charge of correlating all possible candidates in a efficient way, is it possible to improve the bounding box overlap between frames, thus leading to obtain better results.

4. Multi-Target Multi-Camera Tracking

The next step from single camera tracking is to transfer this algorithm to perform Multi-Object Multi-Camera Tracking. To do so, we implemented some changes on the previous method.

The sets of cameras were divided in pairs of contiguous cameras in order to perform the detection on each camera. Then, the following procedure is applied to analyse the videos:

First, the detections from different cameras and ground truth are read, and then an accumulator is created to perform tracking on these detections. This accumulator will be updated on each frame.

Next, while computing the detections for each detection on each camera we crop the inside part of the bounding box and pass it through a SIFT descriptor to extract its features. Ones extracted, these features are compared with the previous and actual crops of both cameras. If there is a match between the new crop and the previous stored, the detection is labelled with the ID of the match, see Figure 3.

If that's not the case, and the detection was not seen before, a new ID is generated and the crop is stored to be compared within the next detections.

5. Evaluation and Results

In this section is presented all the experiments and the evaluations that we were testing. Furthermore, we will expose some qualitative results.

5.1. MTSC

After solving the initial problems with the dataset applying the preprocesses we have encountered two other problems with the detections, see Figure 2 to see theses qualitative problems. The first problem, left image, appears when the cars are near to the camera and the displacement is parallel of the point of view. It provokes a big displacement between frames, the detections are quite bad and our algorithm reduce the IDF1 (The cars appears only in a few frames, and we need a counter of four to establish a track as a good track).

The second problem, right image, makes difficult to remove the parked cars. When a moving car is close to a parked one, it provokes a displacement in the bounding box of the parked car generating an IoU below the threshold that remove the parked cars.

The quantitative results can be seen in the Table 1 and 2. The detections obtained with YOLO3 and the maximum overlap algorithm generates better results in most of the cases that we have evaluated in this paper. In few times, Deepsort model generate better detections than YOLO.

*Optical flow was tested together with the models, but the results were worse than without it. As exists a computational cost increase, it motivates us to not include them in the results tables.

	S03 (IDF1)						
	c010	**c011**	**c012**	**c013**	**c014**	**c015**	**Average**
YOLO3	0,857	0,723	0,472	0,785	0,754	0,740	0,722
ssd512	0,794	0,663	0,524	0,794	0,731	0,68	0,698
mask_rcnn	0,76	0,632	0,475	0,783	0,715	0,7	0,678
deepsort	0,8313	0,649	0,553	0,736	0,658	0,727	0,692
deepsort + OP	0,777	0,479	0,524	0,743	0,646	0,727	0,649

Table 1. MTSC IDF1 score for each video in S03 for the different used methods.

	S01	**S04**
	Average	**Average**
YOLO3	0,555	0,556
ssd512	0,537	0,521
mask_rcnn	0,520	0,465

Table 2. MTSC IDF1 score for S01 and S04 for the different backbones used.

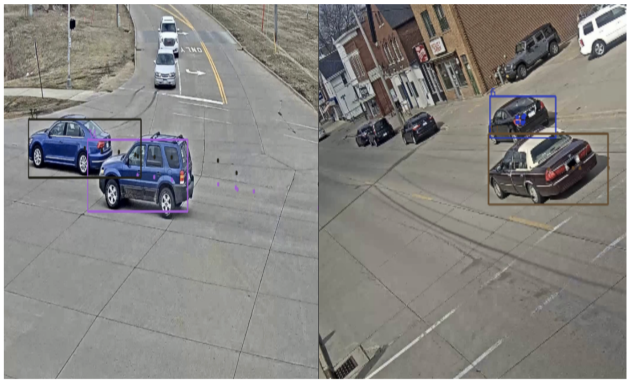

Figure 2. Qualitative examples of MTSC with our algorithm

5.2. MTMC

Once all the detections were executed, the different metrics were computed for the corresponding cameras. The experiments are documented on Tables 4, 5, 6. Then an overall average was computed for all the different experiments, see Table 3.

After analyze in details the results of Table 3 we were able to extract some conclusions and remarks, so in order to see different the remarks we list hereafter.

- Feature descriptors are not the best option to perform matches between cameras, as they depend too much on the angle of the camera and its point of view.
- Adjust ROIs for fixed cameras is a good option to improve detections.
- The method used needs a previous observation and study of the camera environment in order to improve the detections and to create masks and different ROIs.
- Semi-occlusions, like tree branches can make the descriptor perform wrong when detecting.
- Our Multi object Multi camera tracking provides decent results regarding the simplicity of the algorithms.

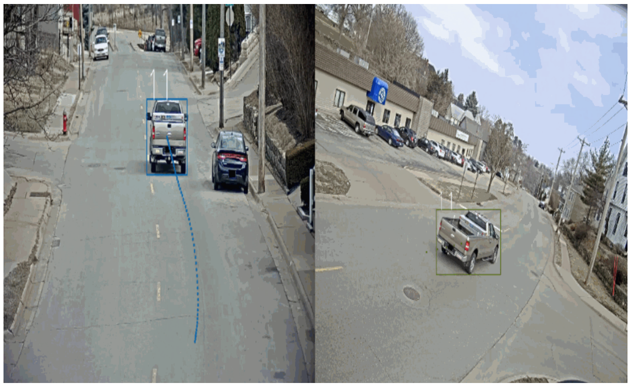

Figure 3. Time sequence asynchronous example. Although is properly identified, the time and place is not the same.

Metric	S03	S04	S01	AV
IDF	0,718	0,617273	0,5025	0,612591
IDP	0,8306	0,755909	0,868	0,81817
IDR	0,672	0,575364	0,35375	0,533705
Precision	0,8718	0,861091	0,96	0,89763
Recall	0,6844	0,653182	0,39125	0,576277

Table 3. MCMT IDF, IDP, IDR, Precision and Recall scores for S03 and S04.

6. Future work

The main objective for the future is to make MTMC able to deal with more than two correlative sequences. Usage of new deep learning techniques and compare it with the handcrafted techniques exposed in this paper might be a good future study.

Also, our future research should take into consideration the usage of neural networks and different models [YOLO3, MaskRCNN, ResNet 50] to improve inference of cars. Also, the option to retrain ReID models to deal with cars instead of persons.

A physical approach in which volume, speed and trajectories are taken into account has been also considered.

With some challenges still pending to be solved, the most difficult part when using a neural network would be the post-process phase related to refine the real object's path detected [5].

In order to do that we still envision the addition of some handcrafted features related to the concrete scenario of a city (typical car paths, crossing lines, distances, time synchronization between sequences). This would positively impact the overall given dataset and be beneficial for the final outcome without overfitting.

7. Conclusions

The intrinsics of the data available must be taken into account before trying to solve a problem. In the case mentioned in this paper, preprocess makes the difference.

It has been demonstrated that the use of handcrafted features and classic computer vision approaches in a consecutive cameras' scenario can be used for vehicles tracking with equivalent confidence levels to algorithms considering more cameras and using deep learning techniques.

Although obtaining equivalent results, problems to solve are similar: changes of light, perspective differences and occlusions. Handcrafted features was a good approximation when we have the objects, in our case the cars, from the same perspective.

Our first step was to accomplish the task of Single Camera tracking. In this task we achieved an average IDF1 of 68,6% of all the videos. From this results we have to take in consideration that some track IDs were mislead while using the method. This error was probably related with the embedding space studied.

On the other hand, the average IDF1 score obtained for Multi camera is 61,2%, the result was slightly worse than previous since was accumulated around the camera and the detection.

Finally, we notice that the post-process of the Bounding Boxes with the optical flow is too expensive, and we didn't get any benefit from using it in our scores.

References

[1] M. Naphade, S. Wang, D. C. Anastasiu, Z. Tang, M.-C. Chang, X. Yang, L. Zheng, A. Sharma, R. Chellappa, and P. Chakraborty, "The 4th ai city challenge," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020, p. 2665–2674. 1

[2] M. Naphade, Z. Tang, M.-C. Chang, D. C. Anastasiu, A. Sharma, R. Chellappa, S. Wang, P. Chakraborty, T. Huang, J.-N. Hwang, and S. Lyu, "The 2019 ai city challenge," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019, p. 452–460. 1

[3] M. Naphade, M.-C. Chang, A. Sharma, D. C. Anastasiu, V. Jagarlamudi, P. Chakraborty, T. Huang, S. Wang, M.-Y. Liu, R. Chellappa, J.-N. Hwang, and S. Lyu, "The 2018 nvidia ai city challenge," in *Proc. CVPR Workshops*, 2018, pp. 53—60. 1

[4] M. Naphade, D. C. Anastasiu, A. Sharma, V. Jagrlamudi, H. Jeon, K. Liu, M.-C. Chang, S. Lyu, and Z. Gao, "The nvidia ai city challenge," in *Prof. SmartWorld*, Santa Clara, CA, USA, 2017. 1

[5] Q. Feng, V. Ablavsky, and S. Sclaroff, "Cityflow-nl: Tracking and retrieval of vehicles at city scale by natural language descriptions," 2021. 1, 4

[6] Z. Tang, M. Naphade, M.-Y. Liu, X. Yang, S. Birchfield, S. Wang, R. Kumar, D. Anastasiu, and J.-N. Hwang, "Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, p. 8797–8806. 1, 2

[7] Y. Yao, L. Zheng, X. Yang, M. Naphade, and T. Gedeon, "Simulating content consistent vehicle datasets with attribute descent," in *The European Conference on Computer Vision (ECCV)*, August 2020, p. 775–791. 1

[8] J. Li, D. H. Ye, T. Chung, M. Kolsch, J. Wachs, and C. Bouman, "Multi-target detection and tracking from a single camera in unmanned aerial vehicles (uavs)," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4992–4997. 2

[9] Z. Zhang, J. Wu, X. Zhang, and C. Zhang, "Multi-target, multi-camera tracking by hierarchical clustering: Recent progress on dukemtmc project," December 2017. 2

[10] P. Li, G. Li, Z. Yan, Y. Li, M. Lu, P. Xu, Y. Gu, B. Bai, and Y. Zhang, "Spatio-temporal consistency and hierarchical matching for multi-target multi-camera vehicle tracking," 2016, supplied as additional material https://openaccess.thecvf.com/content_CVPRW_2019/papers/AI%20City/Li_Spatio-temporal_Consistency_and_Hierarchical_Matching_for_Multi-Target_Multi-Camera_Vehicle_Tracking_CVPRW_2019_paper.pdf. 2

[11] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," vol. 2749, 06 2003, pp. 363–370. 3

[12] N. Wojke and A. Bewley, "Deep cosine metric learning for person re-identification," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 748–756. 3

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969. 3

[14] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8. 3

[15] B. Yang and R. Nevatia, "An online learned crf model for multi-target tracking," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2034–2041. 3

Appendix

S01 CAM	001 / 002	002 / 003	003 / 004	004 / 005	Total	AV
IDF	0,539	0,512	0,502	0,457	2,01	0,5025
IDP	0,873	0,9	0,872	0,827	3,472	0,868
IDP	0,39	0,357	0,352	0,316	1,415	0,35375
PREC	0,958	0,989	0,967	0,926	3,84	0,96
REC	0,428	0,393	0,391	0,353	1,565	0,39125

Table 4. MTMC experiments results on S01

SO3 CAM	c10/c11	c11/c12	c12/c13	c13/c14	c14/15	Total	AV
IDF	0,871	0,647	0,567	0,813	0,692	3,59	0,718
IDP	0,963	0,616	0,639	0,946	0,989	4,153	0,8306
IDP	0,796	0,81	0,509	0,712	0,533	3,36	0,672
PREC	0,966	0,706	0,745	0,95	0,992	4,359	0,8718
REC	0,798	0,781	0,594	0,715	0,534	3,422	0,6844

Table 5. MTMC experiments results on S03

S04 CAM	24 / 25	38 / 39	38 / 40	38 / 39	19 / 18	19 / 20	31 / 32	30 / 31	30 / 32	28 / 27	36 / 37	Total	AV
IDF	0,274	0,828	0,68	0,68	0,671	0,787	0,526	0,466	0,464	0,678	0,736	6,79	0,61727
IDP	0,538	0,835	0,742	0,742	0,677	0,794	0,9	0,89	0,889	0,532	0,776	8,315	0,75591
IDP	0,182	0,821	0,627	0,627	0,665	0,779	0,37	0,31	0,314	0,934	0,7	6,329	0,57536
PREC	0,692	0,87	0,934	0,934	0,845	0,863	0,96	0,971	0,971	0,532	0,9	9,472	0,86109
REC	0,236	0,855	0,789	0,79	0,831	0,847	0,398	0,343	0,343	0,934	0,819	7,185	0,65318

Table 6. MTMC experiments results on S04