Mobility21

A USDOT NATIONAL UNIVERSITY TRANSPORTATION CENTER

Carnegie Mellon University

Penn
University of Pennsylvania

The Ohio State University

CCAC

Vehicle and Pedestrian Trajectory and Gap Estimation for Traffic Conflict Prediction

Alexander Hauptmann
(https://orcid.org/0000-0003-2123-0684)

Final Research Report February 1, 2022

Contract # 69A3551747111

Contents

1. Overview

We have developed components of a system to identify and measure vehicle and pedestrian trajectories, speed, inter-vehicle gaps and using video feeds from arbitrary traffic surveillance cameras. The potential impact to transportation safety is the ability to detect crashes in real-time and capturing near crashes or accidents and their context. Real-time analysis allows for immediate notification, detecting traffic density and speeds. Alerting safety planners to near-miss crash events and related contextual information, could provide critical information for safety enhancement through appropriate infrastructure safety modifications.
This project's main research focus was to build a system to identify and measure vehicle trajectories, speed, inter vehicle gaps and vehicle-pedestrian gaps using video feeds from arbitrary traffic surveillance cameras, most of which will not have been specially calibrated and where no training data is available. This work showed that it is possible to create a system that can monitor a video stream in real-time for the purpose of traffic monitoring and improving safety.
The potential impact of such a system to transportation safety is in the ability to detect car crashes. This allows real-time capturing near crashes in real-time, allowing for immediate notification, detecting traffic density and speeds, and alerting safety planners to near-miss crash events and related contextual information, which could be mitigated through appropriate infrastructure safety modifications.
Over time, we also generalized this research to deal with large scale analysis of vehicle activities, and tracking of multiple cars with the ability to re-identify the same cars at different city roads from different cameras. This work resulted in a system that won first prize at the Road Challenge competition and also placed first at the NVidia AI City Challenge. Details can be found below.

In parallel, we also decided to focus on the prediction pedestrian routes and trajectories in a city as well as in typical road situations. We consider this work on pedestrian trajectories and path prediction an important complement to the prediction of vehicle in traffic. Details can be found at the end of the report,
.

2. Traffic danger recognition with surveillance cameras without training data.

We have developed components of a system to identify and measure vehicle and pedestrian trajectories, speed, inter-vehicle gaps and using video feeds from arbitrary traffic surveillance cameras. The potential impact to transportation safety is the ability to detect crashes in real-time and capturing near crashes or accidents and their context. Real-time analysis allows for immediate notification, detecting traffic density and speeds. Alerting safety planners to near-miss crash events and related contextual information, could provide critical information for safety enhancement through appropriate infrastructure safety modifications.

This project's research focus was to build a system to identify and measure vehicle trajectories, speed, inter vehicle gaps and vehicle-pedestrian gaps using video feeds from arbitrary traffic surveillance cameras, most of which will not have been specially calibrated and where no training data is available. This work showed that it is possible to create a system that can monitor a video stream in real-time for the purpose of traffic monitoring and improving safety.

The potential impact of such a system to transportation safety is in the ability to detect car crashes. This allows real-time capturing near crashes in real-time, allowing for immediate notification, detecting traffic density and speeds, and alerting safety planners to near-miss crash events and related contextual information, which could be mitigated through appropriate infrastructure safety modifications.

The implemented approach can automatically estimate the intrinsic and extrinsic camera parameters, which allows a 3-D reconstruction of the camera field of view. Given this reconstruction, exact estimates of vehicle or other traffic participant location, speed and spacing are now possible. Based on the 3D reconstruction of the road plane and prediction of trajectories, the method can achieve prediction, detection and mensuration without specific labeled training data of relevant vehicle to vehicle or vehicle-pedestrian interaction events. Since the gap distance between traffic vehicles is important for predicting vehicle flow, measuring the gap distance from a roadside camera aid with traffic operations can be very useful. By exploiting insights into the kinematic states in the monocular 3D representations of vehicles and pedestrians, our system has shown to have much better performance than a system based on hand-annotated training-data from specific cameras and views.

The system works in several steps. In step 1, our implemented approach robustly estimates the extrinsic and intrinsic camera parameters, which provides the system with the 'camera model' which also identifies the camera mount point, with a distance of its focal length from the observed images. Within the world coordinate system, the system estimates the road plane.Step 2 was the reliable detection and tracking of vehicles and pedestrians, for which several approaches have been proposed. Methods also have been implemented to estimate a 3-D bounding box around a vehicle or a pedestrian, which we can then transform into our world coordinate system. Step 3 is the trajectory prediction of the vehicle or the pedestrian. After the core system was implemented we proceeded to make it work robustly over several data sets of traffic accidents and real traffic cameras. We have also improved the efficiency of the analytics process to allow real-time results with minimal latency using a single computational device.

We developed an approach to identify and measure vehicle trajectories, speed, inter-vehicle gaps by simply using video feeds from arbitrary traffic surveillance cameras. The potential impact of such a system to transportation safety is in the ability in real-time to detect crashes and identifying near-misses in real-time, allowing for immediate notification, detecting of traffic

density and speeds, and alerting safety planners to near-miss crash events and related contextual information, which could be mitigated through appropriate infrastructure safety modifications. Based on this research, it appears possible to create a system that can monitor a video stream in real-time for this purpose.
The approach is to automatically estimate the intrinsic and extrinsic camera parameters, which allows a 3-D reconstruction of the camera field of view. Given this reconstruction, exact estimates of vehicle or other traffic participant location, speed and spacing are now possible. Based on the 3D reconstruction of the road plane and prediction of trajectories, the method can achieve prediction, detection and mensuration of traffic participants without specific labeled training data of relevant vehicle or vehicle-pedestrian interaction events. Since the gap distance is important for predicting vehicle flow, measuring the vehicle gap distance from a roadside camera could inform traffic operations managers of problems and potential solutions.

We have developed components o a system to identify and measure vehicle and pedestrian trajectories, speed, inter-vehicle gaps and using video feeds from arbitrary traffic surveillance cameras. The potential impact to transportation safety is the ability to detect crashes in real-time and capturing near crashes or accidents and their context. Real-time analysis allows for immediate notification, detecting traffic density and speeds. Alerting safety planners to near-miss crash events and related contextual information, could provide critical information for safety enhancement through appropriate infrastructure safety modifications.
This project's research focus was to build a system to identify and measure vehicle trajectories, speed, inter vehicle gaps and vehicle-pedestrian gaps using video feeds from arbitrary traffic surveillance cameras, most of which will not have been specially calibrated and where no training data is available. This work showed that it is possible to create a system that can monitor a video stream in real-time for the purpose of traffic monitoring and improving safety.
The potential impact of such a system to transportation safety is in the ability to detect car crashes. This allows real-time capturing near crashes in real-time, allowing for immediate notification, detecting traffic density and speeds, and alerting safety planners to near-miss crash events and related contextual information, which could be mitigated through appropriate infrastructure safety modifications.

The implemented approach can automatically estimate the intrinsic and extrinsic camera parameters, which allows a 3-D reconstruction of the camera field of view. Given this reconstruction, exact estimates of vehicle or other traffic participant location, speed and spacing are now possible. Based on the 3D reconstruction of the road plane and prediction of trajectories, the method can achieve prediction, detection and mensuration without specific labeled training data of relevant vehicle to vehicle or vehicle-pedestrian interaction events. Since the gap distance between traffic vehicles is important for predicting vehicle flow, measuring the gap distance from a roadside camera aid with traffic operations can be very useful .By exploiting into the kinematic states in the monocular 3D representations of vehicles and pedestrians, our system has shown to have much better performance than a system based on hand-annotated training-data from specific cameras and views.
The system works in several steps. In step 1, our implemented approach robustly estimates the extrinsic and intrinsic camera parameters, which provides the system with the 'camera model' which also identifies the camera mount point, with a distance of its focal length from the observed images. Within the world coordinate system, the system estimates the road plane .Step

2 was the reliable detection and tracking of vehicles and pedestrians, for which several approaches have been proposed. Methods also have been implemented to estimate a 3-D bounding box around a vehicle or a pedestrian, which we can then transform into our world coordinate system. Step 3 is the trajectory prediction of the vehicle or the pedestrian. After the core system was implemented we proceeded to make it work robustly over several data sets of traffic accidents and real traffic cameras. We have also improved the efficiency of the analytics process to allow real-time results with minimal latency using a single computational device.

Our traffic danger recognition model consists of five steps. Camera calibration provides geometry parameters and a transformation from image coordinates to road plane coordinates. Object detection and tracking algorithms provide the types, positions, and masks of vehicles and trace their histories. 3D bounding boxes are built to localize vehicles in the world space and then project to the road plane. Positions and speeds are calculated with adjacent frames plus smoothing and predicted for the future. Finally, we can recognize danger from vehicle distances and potential overlaps in the predictions.

We adopt a frequently used traffic camera model as shown in Figure 1. We follow the practice of setting up directions of three vanishing points U; V;W. With a known plane, points in an image can be re-projected to points on the plane in the

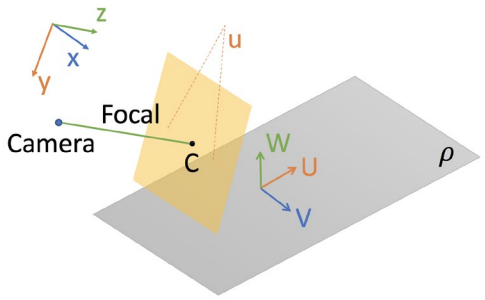


Figure 1. Traffic camera model: x; y; z defines a world coordinate system, where x-y plane is parallel to the image and z passes through its top left. Camera is on the x-y plane and points to the principal point C at the center of the image. *p* is the road plane. U; V; W are directions of vanishing points, U in the direction of traffic, V parallel to the road and perpendicular to U, and W perpendicular to *p*.

Although some automatic calibration methods have been developed, they do not achieve perfect performance in our model. So we remain using a 'manual' calibration which requires labeling two groups of parallel lines of each camera view. Then we derive two vanishing points in the image space using a least square error method. We rotate the world coordinate system to make the x-z plane parallel to the road plane, so we can get plane coordinates of a point by omitting the y axis. Rotation parameters are acquired by solving Equation 1.

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\gamma & \sin\gamma & 0 & 0 \\ -\sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{V}{\|V\|} \\ \frac{W}{\|W\|} \\ \frac{U}{\|U\|} \end{bmatrix} \quad (1)$$

Object Detection and Tracking

We select Mask R-CNN as our object detection model, which outputs detection scores, object types, bounding boxes, and object masks. We select three types of objects as targets: car, bus, and truck. Then we apply a filter to the detected objects as shown in Figure 2. The filter follows three rules:

1. Vehicles should not be too small in size.
2. Vehicles should be in the road area.
3. Vehicles should be completely visible.

We use Deep SORT to track vehicles across frames. Each vehicle is supposed to get a unique ID from the tracking model, and this is robust through brief loss of detection.

Figure 2. Object detection: raw detections (left), and filtered objects(right). The white car at the top left is filtered by rule 1, the red at the bottom right by 3, and the cars at the top right by 2.

To get the current location of a vehicle, we can find the bottom of the 3D bounding boxes and project them to the road plane according to Section 3.1. We build a 3D bounding box with tangent lines from vanishing points. Lines with subscript min denote the lines with the minimum tilt angle, and max the maximum. Position of the points are shown in Figure 3.

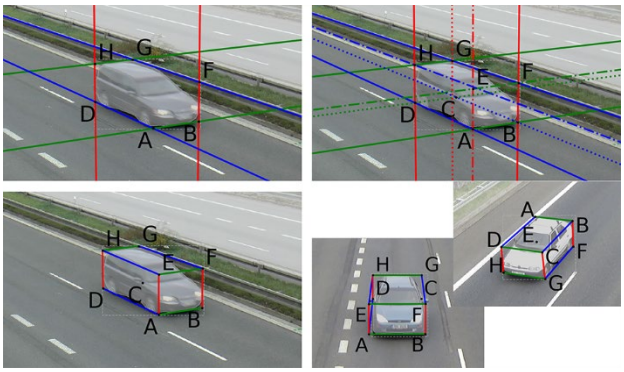


Figure 3. 3D bounding box: tangent lines of contour and their intersections (top left), derived lines and intersections (top right), the final result (bottom left), and vehicles in other angles of view (bottom right). Lines in colors of blue, green, and red pass through u; v;w respectively.

With an optional scale factor, we are able to know the real world value of the speed.
To predict the trajectories, we assume:

1. The future is divided into time slots with equal lengths.
2. The vehicle centers follow normal distributions.
3. The vehicle shapes do not change.

We predict speed, acceleration, center coordinates and variance for the beginning of each slot as a snapshot. Within a slot, we assume there are fixed acceleration and variance.

Then the speed and center coordinates can be calculated according to kinematics rules. In this way, predictions are available for an arbitrary time in the future. For now, we are using a simple linear prediction method with the real situation as the only one snapshot and assuming the acceleration is always zero.
We use two ways to recognize dangerous situations. The first one is the distance measurement between vehicles. It not only tells where cars are going to crash but provides a proactive safety check for areas where cars often get too close, as well. The second one is called danger map, which detects overlap of vehicles in the predictions that indicates crashes. The distance between two vehicles is defined as the minimum distance between two points from two quadrangles respectively.

We accumulate the probability of a car box based on the distribution of its center to get the heat map of a vehicle. It represents the probability of its position at a specific time in the future. Then we aggregate the heat maps of all the vehicles in a scene into a danger map. A danger map represents the probability of coexistence of two or more vehicles inthe same location. Figure 4 shows a sample result of danger recognition.

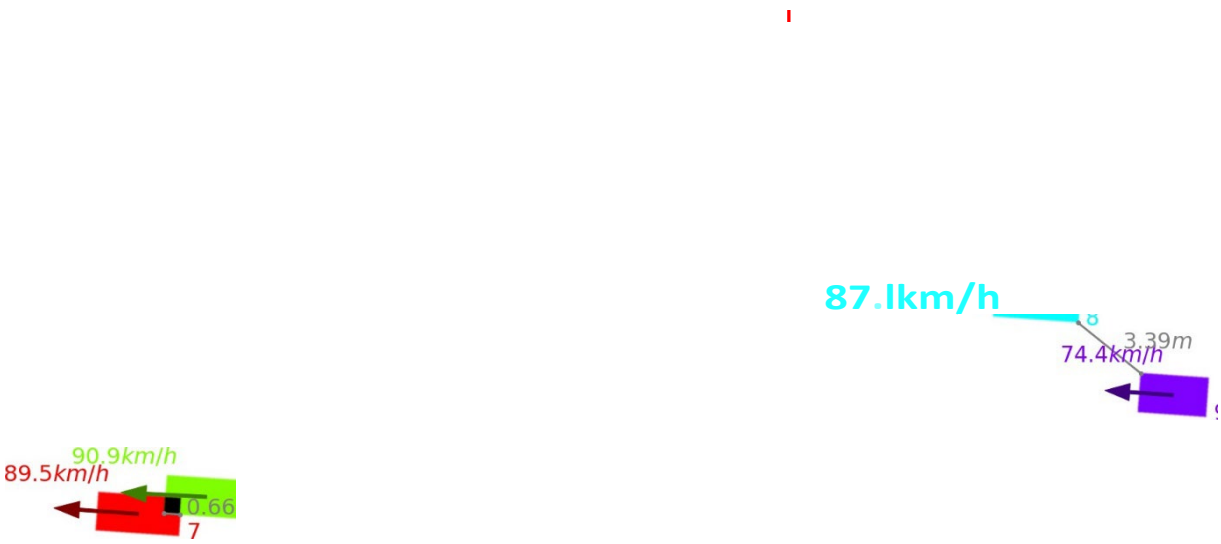


Figure 4. Danger recognition: four vehicles in a sample prediction of the road plane, with vehicle IDs at the bottom right and speeds at the top left. Distances are shown for nearing vehicles, and danger area is shown in black at the overlap of vehicle 7 and 10.

Although the prediction mechanism currently deployed is rather simple, it provides results much beyond our expectations. As a vehicle at 75 km/h would move 2.5 meters in 0.12 seconds, a mean error of 0.24 m for location prediction is well acceptable. The difference between the mean and median values indicates some outliers are harming the performance, but we can still see that most of the predictions are within an error of 2km/h. For traffic on highways, crashes usually happen within 0.12 seconds, so it is enough for the danger map to work. Moreover, another

prediction of +0.24s is there for more information beforehand, and it is reasonable to have a slightly larger error than +0.12s.

Conclusions

We propose a traffic danger recognition model that works with arbitrary surveillance cameras. It does not require any labeled training data of crashes. The model consists of five steps: camera calibration, object detection and tracking, 3D bounding box, trajectory prediction, and danger recognition. We measure the performance with experiments step by step, presenting that it is accurate at the estimation of speed and position of vehicles by projecting to a 3D reconstructed road plane. It is suitable for crash detection and proactive safety checks. A demo of our model working on a real crash scene can be found on Youtube[1].
(https://www.youtube.com/playlist?list=PLssAerj8zfUR5wBc7N6gmCFTm0azCHSIf)
In the future, a complete test set of video containing real crashes will be processed to report detection accuracy. Trajectory prediction model could be improved with conditional random fields or recurrent neural network. We will also test automatic camera calibration methods to obtain similar performance as manual calibration, then the system could function on arbitrary surveillance cameras with zero input

Publication:: Yu, L., Zhang, D., Chen, X., & Hauptmann, A. (2018, November). Traffic danger recognition with surveillance cameras without training data. In 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (pp. 1-6). IEEE

3. City-scale Vehicle Tracking and ReID

Multi-target multi-camera (MTMC) tracking aims to track the vehicles over large areas within multiple camera networks.
Different from classical multiple object tracking (MOT) which only focuses on tracking objects within a single camera, MTMC needs to resort to multiple cameras.
Moreover, the characteristics of moving vehicles bring unique challenges for multi-camera vehicle tracking.

Road Activity Detection

The CMU Argus++ [1] activity detection system has achieved leading performance in a series of benchmarks lately, including CVPR ActivityNet ActEV 2021, NIST ActEV SDL Unknown Facility and Known Facility, NIST TRECVID ActEV 2020-2021, etc. However, these benchmarks share the setting of extended videos from stationary cameras. To support road activity detection in an autonomous driving scenario, we adapted Argus++ into ArgusRoad for a moving camera mounted on a vehicle. ArgusRoad has taken first place at ICCV 2021 ROAD challenge.

The activity detection metrics for extended videos proposed by NIST, including nAUDC and Pmiss, measures the false positives at frame level while counting the true positives at the instance level. This has led to a design of output instances with a short temporal extent to achieve a higher score. However, if we also measure false positives at the instance level or care about the

[1] **Argus: Efficient Activity Detection System for Extended Video Analysis.** WACV Workshops 2020: 126-133. Wenhe Liu, Guoliang Kang, Po-Yao Huang, Xiaojun Chang, Lijun Yu, Yijun Qian, Junwei Liang, Liangke Gui, Jing Wen, Peng Chen, Alexander G. Hauptmann.

correct localization of an entire activity, this design does not fit. Therefore, in ArgusRoad, we proposed a connectionist temporal localization method to tackle this.

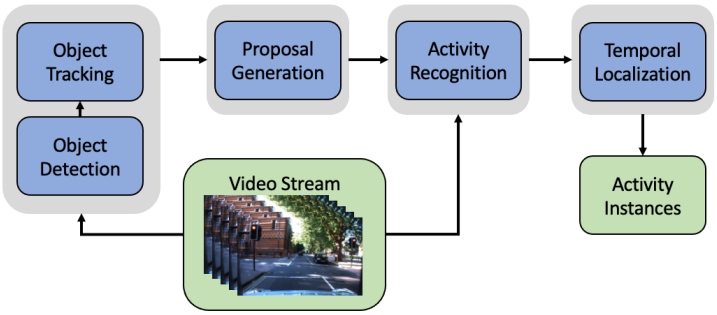


Figure 3.1 ArgusRoad Framework

Method

Figure 3.1 shows the architecture of the ArgusRoad framework. Like object proposals in two-stage object detection methods such as Faster R-CNN, ArgusRoad uses activity proposals as an intermediate concept. In Argus++, we have shifted from a non-overlapping proposal sample method to the overlapping format, which ensures the coverage of activity instances. To further achieve precise temporal localization, we attempt to use a dense sampling method, as illustrated below. In this sampling method, proposals are generated for every frame, with a fixed duration as temporal context. In the implementation, it is equivalent to using stride equals to 1 frame, as shown in Figure 3.2.

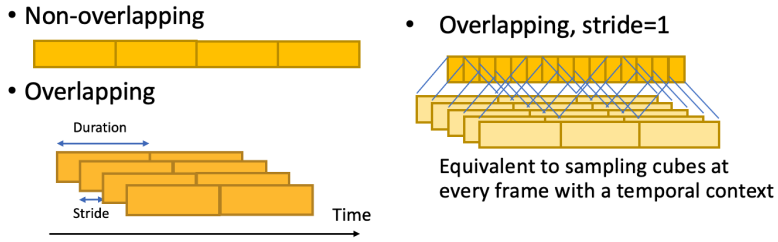


Figure 3.2 Proposal Sampling Methods

One important difference is in the label assignment aspect. Previously, the label of a proposal is generated based on the ground truth instances that overlap with any portion of a proposal. For the dense sampling method, however, the label is only about the center frame. In this way, the classifier effectively learns to predict activity confidence scores of each frame based on a temporal context. Then we can apply a simple yet effective merge method to obtain the final output instance. For each type of action, we select consecutive frames with scores above a certain threshold and length above a minimum duration as a positive prediction instance.

Achievements

ArgusRoad predicts each activity instance as a single output, rather than several separate cubes in Argus++. Therefore, it's applicable to real spatial-temporal localization with metrics such as mAP on 3DIoU. ArgusRoad won first place at ICCV 2021 ROAD challenge.

Figure 3.3 ICCV 2021 ROAD Challenge[2]

City-scale multi-camera vehicle tracking is an important task in the intelligent city and traffic management. It is quite challenging with large scale variance, frequent occlusion and appearance variance caused by viewing perspective difference. In this paper, we propose ELECTRICITY, an efficient multi-camera vehicle tracking system with aggregation loss and fast multi-target cross-camera tracking strategy. The proposed system contains four main modules. Firstly, we extract tracklets under single camera view through object detection and multi-object tracking modules which shared the detection features. After that, we match the generated tracklets through a multi-camera re-identification module. Finally, we eliminate isolated tracklets and synchronize tracking ids according to the re-identification results. The proposed system wins the first place in the City-Scale Multi-Camera Vehicle Tracking of AI City 2020 Challenge (Track 3) with a score of 0.4585. With the continuous expansion of the city scale, the man-agement of city has become more and more challenging. Thanks to the development of computer vision technol- ogy and surveillance network throughout the city, there are many new options for city management, especially in traffic management. Among them, multi-camera vehicle tracking is one of the important tasks. It aims to track the vehicles over large areas in multiple surveillance camera networks. Furthermore, it enables better transportation design and traffic flow optimization. Different from classical multiple object tracking (MOT) which only focuses on tracking objects within a single camera, multi-camera vehicle tracking needs to resort to multiple cameras.

2 The leaderboard is online at https://eval.ai/web/challenges/challenge-page/1059/leaderboard/2748

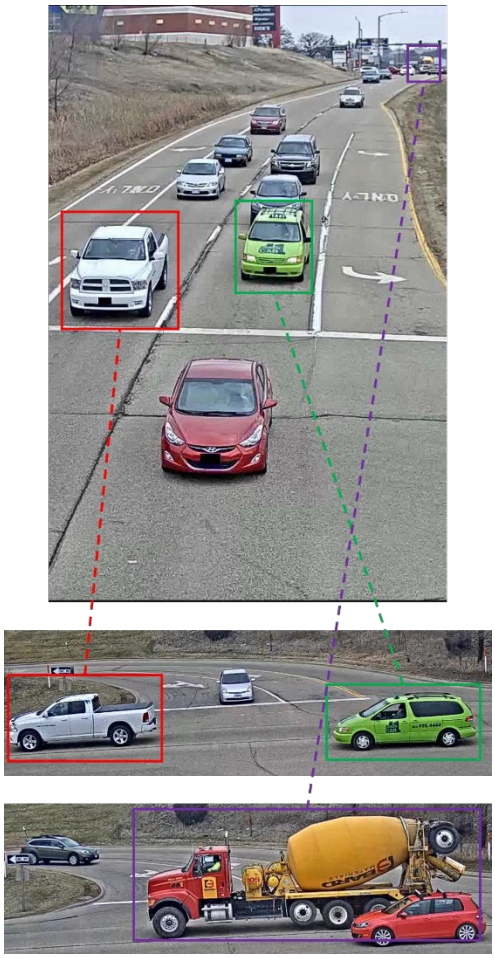


Figure 1: Multi-camera vehicle tracking needs to find out the same vehicles which appear in multiple cameras. Their appearance and size usually varies a lot due to the difference of viewing perspective and distance to cameras.

In fact, multi-camera vehicle tracking is a complicated task that includes detection, tracking, re-identification (ReID) and it is already a heated and leading-edge realm in computer vision research. Recent years have witnessed many successful works, especially after the release of some public datasets and challenges in this realm. Although the former methods have achieved great performance on the public datasets, they usually use feature aggregation which use lots of computational resources, such as GPUs. Moreover, most of them require annotations of large-scale datasets to train their models, which is difficult to collect. Meanwhile, there are still several challenges left in this realm:

1. The appearance of a specific vehicle usually varies a lot due to the difference of viewing perspectives and its distance to cameras. It brings challenges to accurate multi-camera ReID.
2. The synchronization procedure of tracking IDs tends to be a time-consuming work since it needs to match tracklets from different camera views.

To tackle these challenges, we proposed ELECTRICITY, an efficient and accurate multi-camera vehicle tracking system with aggregation loss and fast multi-target cross-camera tracking (MTMC) strategy. Given a set of videos under different camera views, the system firstly detects and tracks multiple vehicles within each single camera. After that, it re-identifies these tracklets among multiple camera views with a model trained by aggregation loss. Finally, the system synchronizes tracking results across multiple camera views and accelerate the procedure through using geometry information.

Tracking

Figure 3: Different from SOT, tracking ID in MOT usually change due to overlapping. Thus,rules, filters or smoothing functions are needed.

Tracking is another challenging computer vision task and there are many successful works in this realm. Generally, it can be split into three categories, single object tracking (SOT), or multiple object tracking (MOT).

To perform multi-camera vehicle tracking, the first step is to reliably detect vehicles within a single image. We adopt the state-of-the-art instance segmentation network Mask R-CNN as our frame-level vehicle detection model. It utilizes a powerful convolutional neural network as its backbone for feature extraction, such as ResNext-101 with feature pyramid network. With region proposal network and region of interest (RoI) alignment, it produces feature representations for candidate detections. Through multiple output heads, we can get the object class, confidence score, bounding box, and segmentation mask of each detection.

In the traffic scenarios, the objects we mainly focus on are vehicles. Here it is defined as the union of *Car*, *Bus*, *Truck* from the Microsoft COCO[12] dataset. The original Mask R-CNN only performs non-maximum suppression (NMS) within each class, which typically works fine. However, in our cases some vehicles could result in multiple detections, such as both *Car* and *Truck* for a pickup truck. Therefore, we additionally apply an interclass non-maximum suppression (NMS) on the detections. All detections are sorted in the descending order according to their confidence scores. Then we select the detection one by one and skip if there exists a detection with intersection over union (IoU) of at least IoUnms .

Samples of detected vehicles are shown in Figure 6.

To track multiple targets within a single view, we follow the tracking-by-detection paradigm to associate frame-level detection results into tracklets. We utilize two state-of-the art online multi-target tracking algorithms to associate detections into tracklets. Deep SORT is an online tracking algorithm which incorporates a Kalman filter with a constant velocity model to estimate the location and speed of objects from noisy detections. One of its great advantage is including the deep visual features as association criterions, which is much more expressive than simple bounding boxes. To reduce computational complexity, we directly reuse the RoI features from the backbone of Mask R-CNN as the features for Deep SORT.

It utilizes a powerful convolutional neural network as its backbone for feature extraction, such as ResNext-101[28] with feature pyramid network[11]. With region proposal network and region of interest (RoI) alignment, it produces feature representations for candidate detections. Through multiple output heads, we can get the object class, confidence score, bounding box, and segmentation mask of each detection. In the traffic scenerios, the objects we mainly focus on are vehicles. Here it is defined as the union of *Car*, *Bus*, *Truck* from the Microsoft COCO dataset. The original Mask R-CNN only performs non-maximum suppression (NMS)[5] within each class, which typically works fine. However, in our cases some vehicles could result in multiple detections, such as both *Car* and *Truck* for a pickup truck. Therefore, we additionally apply an interclass non-maximum suppression (NMS) on the detections. All detections are sorted in the descending order according to their confidence scores. Then we select the detection one by one and skip if there exists a detection with intersection over union (IoU) of at least IoUnms .

Samples of detected vehicles are shown in Figure 6.

Figure 6: Samples of detected vehicles in single cameras

To track multiple targets within a single view, we follow the tracking-by-detection paradigm to associate frame-level detection results into tracklets. We utilize two state-of-the art online multi-target tracking algorithms to associate detections into tracklets. Deep SORT is an online tracking algorithm which incorporates a Kalman filter with a constant velocity model to estimate the location and speed of objects from noisy detections. One of its great advantage is including the deep visual features as association criterions, which is much more expressive than simple bounding boxes. To reduce computational complexity, we directly reuse the RoI features from the backbone of Mask R-CNN as the features for Deep SORT.

Towards-Realtime-MOT is a recent successor of tracking algorithms, which unifies detection and feature into a single model. As we are already doing so by reusing the Mask R-CNN feature, we only utilizes its association algorithm. Basically, new detections are assigned to existing tracklets based on feature similarity and compliance with spatial constraints. It first attempts to match all detections with previously confirmed tracklets based on feature similarity. A match would be rejected if they are not spatially adjacent. Then its second try is using all remaining detections to match all remaining confirmed tracklets based on bounding box intersection over union (IoU). The third round includes unconfirmed tracklets in the matching, which are typically tracklets of length 1. After all these matches, remaining detections will be recorded as new tracks.

Figure 10: Visualization results of MTMC on five different kinds of vehicles. Each image represents a tracklet and each line represents these tracks from different camera views are classified by MTMC to be from the same vehicle. Yellow box represents false negative and red box represents false positive.

In this research task, we developed an efficient multi-camera vehicle tracking system which is accurate and easy to train without using supplementary data set. In the detection and tracking part, weighted inter-class non-maximum suppression and association algorithm are implemented to generate more accurate bounding boxes. In the Re-ID part, aggregation loss is used for training to overcome the appearance variance caused by different viewing perspectives. Finally, given the tracklets and distance matrix, we adopt fast multi-target cross-camera tracking strategies to generate final results. Our model ranks the first with score 0.4585 and outperforms other teams by a large margin. Meanwhile, it is easy to apply to real world large scale intelligent traffic and city management applications and be upgraded with future models.

Reference: Yijun Qian, Lijun Yu, Wenhe Liu, and Alexander G. Hauptmann. Electricity: An efficient multi-camera vehicle tracking system for intelligent city. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2020

4. Pedestrian tracking and path prediction

In this section we describe several research thrusts to track and predict the path of pedestrians in traffic scenes.

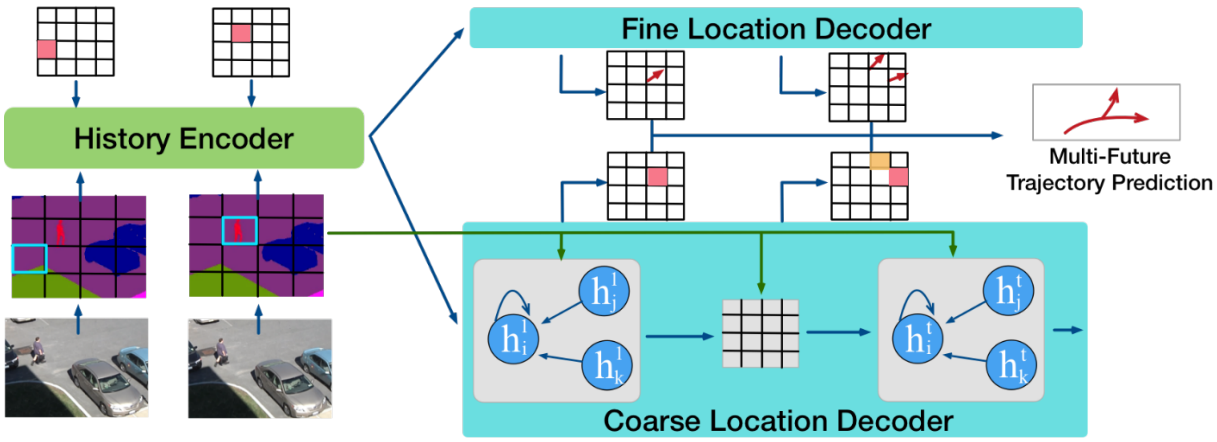
[Trajectory prediction] Liang J., Jiang L., Hauptmann A. (2020) SimAug: Learning Robust Representations from Simulation for Trajectory Prediction. In: Vedaldi A., Bischof H., Brox T., Frahm JM. (eds) Computer Vision – ECCV 2020. ECCV 2020. Lecture Notes in Computer Science, vol 12358. Springer, Cham.

This work studies the problem of predicting future trajectories of people in unseen cameras of novel scenarios and views. We approach this problem through the real-data-free setting in which the model is trained only on 3D simulation data and applied out-of-the-box to a wide variety of real cameras. We propose a novel approach to learn robust representation through augmenting the simulation training data such that the representation can better generalize to unseen real-world test data. The key idea is to mix the feature of the hardest camera view with the adversarial feature of the original view. We refer to our method as SimAug. We show that SimAug achieves promising results on three real-world benchmarks using zero real training data, and state-of-the-art performance in the Stanford Drone and the VIRAT/ActEV dataset when using in-domain training data.

4.1. Multiple Path prediction of Pedestrians in different scenes

This paper studies the problem of predicting the distribution over multiple possible future paths of people as they move through various visual scenes. We make two main contributions. The first contribution is a new dataset, created in a realistic 3D simulator, which is based on real world trajectory data, and then extrapolated by human annotators to achieve different latent goals. This provides the first benchmark for quantitative evaluation of the models to predict multi-future trajectories. The second contribution is a new model to generate multiple plausible future trajectories, which contains novel designs of using multi-scale location encodings and convolutional RNNs over graphs. We refer to our model as Multiverse. We show that our model achieves the best results on our dataset, as well as on the real-world VIRAT/ActEV dataset (which just contains one possible future).

This paper studies the problem of predicting the distribution over multiple possible future paths of people as they move through various visual scenes. We make two main contributions. The first contribution is a new dataset, created in a realistic 3D simulator, which is based on real world trajectory data, and then extrapolated by human annotators to achieve different latent goals. This provides the ***first benchmark*** for quantitative evaluation of the models to predict ***multi-future trajectories***.

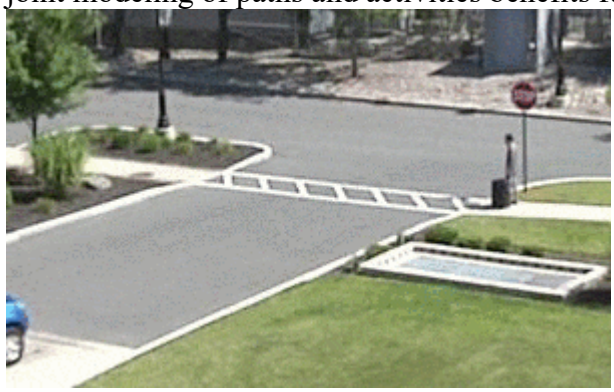


The second contribution is a new model to generate multiple plausible future trajectories, which contains novel designs of using multi-scale location encodings and convolutional RNNs over graphs. We refer to our model as ***Multiverse***. We show that our model achieves the best results on our dataset, as well as on the real-world VIRAT/ActEV dataset (which just contains one possible future).

The Forking Paths Dataset provides high resolution videos along with accurate bounding box and scene semantic segmentation annotations. For more details watch the video at https://www.youtube.com/watch?v=FJTJquN2Kj4&feature=emb_imp_woyt

4.2 Pedestrian path prediction with multiple futures

Deciphering human behaviors to predict their future paths/trajectories and what they would do from videos is important in many applications. Motivated by this idea, this research studies predicting a pedestrian's future path jointly with future activities. We developed an end-to-end, multi-task learning system utilizing rich visual features about the human behavioral information and interaction with their surroundings. To facilitate the training, the network is learned with two auxiliary tasks of predicting future activities and the location in which the activity will happen. Experimental results demonstrate our state-of-the-art performance over two public benchmarks on future trajectory prediction. Moreover, our method is able to produce meaningful future activity prediction in addition to the path. The result provides the first empirical evidence that a joint modeling of paths and activities benefits future path prediction.



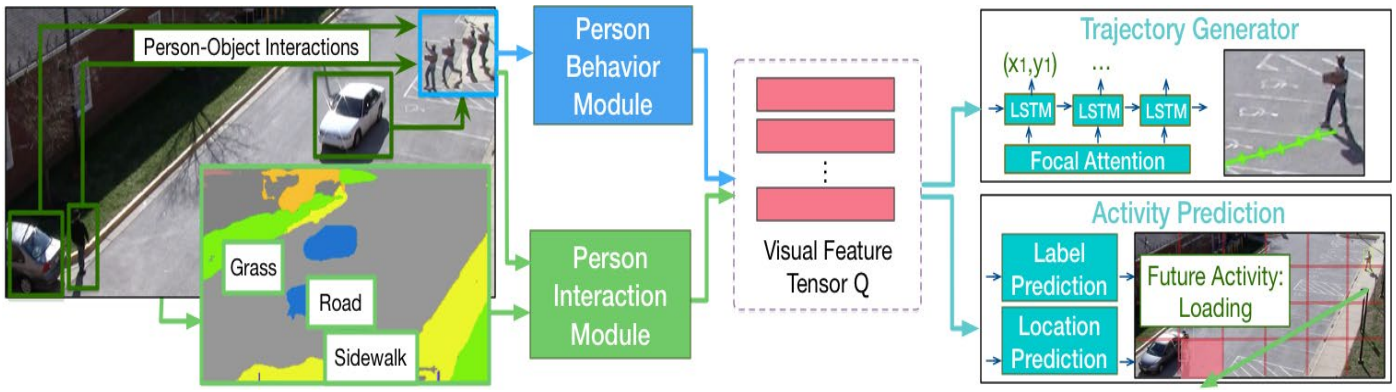
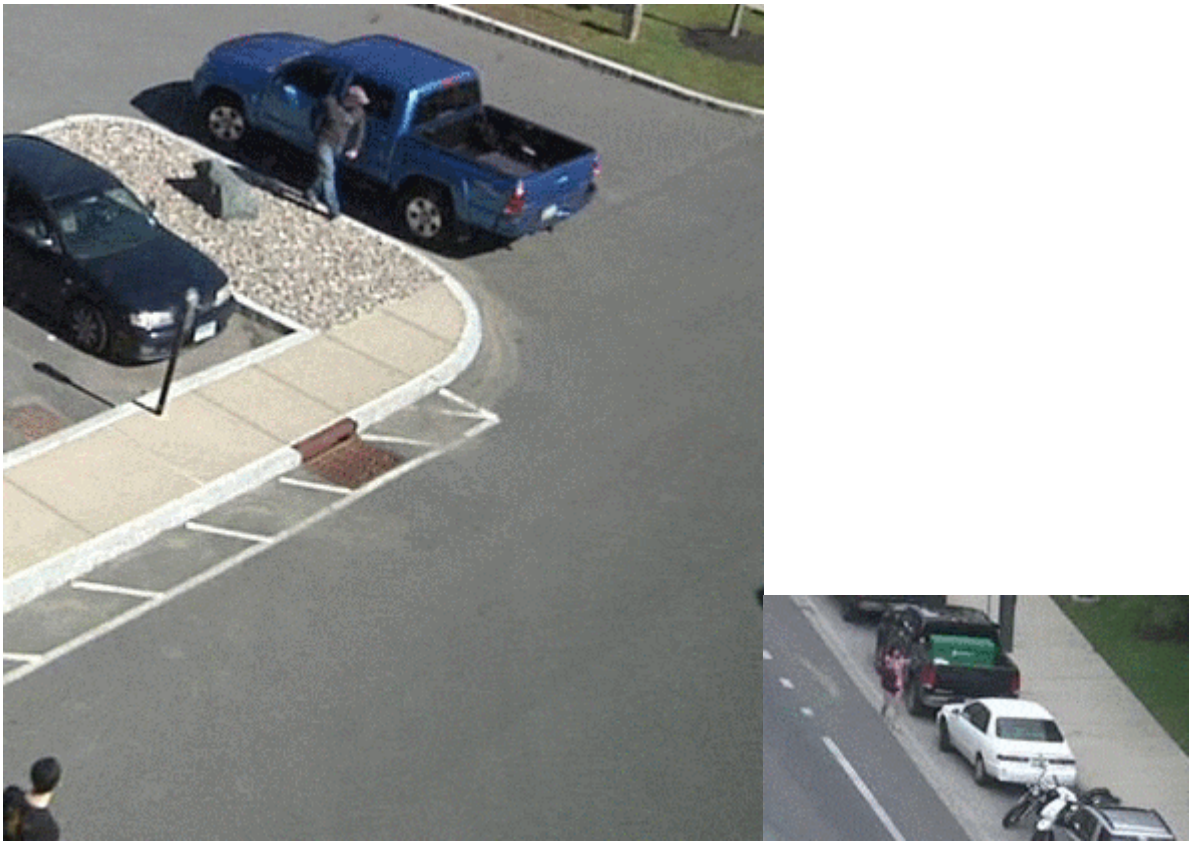


Figure: Our goal is to jointly predict a person's future path and activity. The green and yellow line show two possible future trajectories and two possible activities are shown in the green and yellow boxes. Depending on the future activity, the person (top right) may take different paths, e.g. the yellow path for "loading" and the green path for "object transfer".



Publications relevant to pedestrian path prediction

[Pedestrian path prediction] Liang J., Jiang L., Hauptmann A. (2020) SimAug: Learning Robust Representations from Simulation for Trajectory Prediction. In: Vedaldi A., Bischof H., Brox T.,

Frahm JM. (eds) Computer Vision – ECCV 2020. ECCV 2020. Lecture Notes in Computer Science, vol 12358. Springer, Cham.
This work studies the problem of predicting future trajectories of people in unseen cameras of novel scenarios and views. We approach this problem through the real-data-free setting in which the model is trained only on 3D simulation data and applied out-of-the-box to a wide variety of real cameras. We propose a novel approach to learn robust representation through augmenting the simulation training data such that the representation can better generalize to unseen real-world test data. The key idea is to mix the feature of the hardest camera view with the adversarial feature of the original view. We refer to our method as SimAug. We show that SimAug achieves promising results on three real-world benchmarks using zero real training data, and state-of-the-art performance in the Stanford Drone and the VIRAT/ActEV dataset when using in-domain training data.

[Pedestrian path prediction] Liang, J., Jiang, L., Murphy, K., Yu, T., & Hauptmann, A. (2020). The garden of forking paths: Towards multi-future trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10508-10518).
This paper studies the problem of predicting the distribution over multiple possible future paths of people as they move through various visual scenes. We make two main contributions. The first contribution is a new dataset, created in a realistic 3D simulator, which is based on real world trajectory data, and then extrapolated by human annotators to achieve different latent goals. This provides the first benchmark for quantitative evaluation of the models to predict multi-future trajectories. The second contribution is a new model to generate multiple plausible future trajectories, which contains novel designs of using multi-scale location encodings and convolutional RNNs over graphs. We refer to our model as Multiverse. We show that our model achieves the best results on our dataset, as well as on the real-world VIRAT/ActEV dataset (which just contains one possible future).

Publications related to Pedestrian path prediction

[Pedestrian Path Prediction] Liang, Junwei, Lu Jiang, Juan Carlos Niebles, Alexander G. Hauptmann, and Li Fei-Fei. "Peeking into the future: Predicting future person activities and locations in videos." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5725-5734. 2019.

[Pedestrian path prediction] Liang, J., Jiang, L., Murphy, K., Yu, T., & Hauptmann, A. (2020). The garden of forking paths: Towards multi-future trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10508-10518).

[Pedestrian path prediction] Liang J., Jiang L., Hauptmann A. (2020) SimAug: Learning Robust Representations from Simulation for Trajectory Prediction. In: Vedaldi A., Bischof H., Brox T., Frahm JM. (eds) Computer Vision – ECCV 2020. ECCV 2020. Lecture Notes in Computer Science, vol 12358. Springer, Cham.