Research Article

Transportation Research Record
1–11

DOI: 10.1177/03611981221076426
journals.sagepub.com/home/trr
SAGE

Traffic Behavior Recognition from Traffic Videos under Occlusion Condition: A Kalman Filter Approach

Junfeng Jiao[1] and Huihai Wang[1]

Abstract

Real-time traffic data at intersections is significant for development of adaptive traffic light control systems. Sensors such as infrared radiation and GPS are not capable of providing detailed traffic information. Compared with these sensors, surveillance cameras have the potential to provide real scenes for traffic analysis. In this research, a You Only Look Once (YOLO)-based algorithm is employed to detect and track vehicles from traffic videos, and a predefined road mask is used to determine traffic flow and turning events in different roads. A Kalman filter is used to estimate and predict vehicle speed and location under the condition of background occlusion. The result shows that the proposed algorithm can identify traffic flow and turning events at a root mean square error (RMSE) of 10. The result shows that a Kalman filter with an intersection of union (IOU)-based tracker performs well at the condition of background occlusion. Also, the proposed algorithm can detect and track vehicles at different optical conditions. Bad weather and night-time will influence the detecting and tracking process in areas far from traffic cameras. The traffic flow extracted from traffic videos contains road information, so it can not only help with single intersection control, but also provides information for a road network. The temporal characteristic of observed traffic flow gives the potential to predict traffic flow based on detected traffic flow, which will make the traffic light control more efficient.

Keywords

data and data science, computer vision, intelligent traffic system, sensor data analytics, traffic information, vehicle detection

In the process of urbanization, populations are increasing rapidly in urban areas. The consequent surge of vehicle numbers causes a series of problems, such as traffic congestion, air pollution, and crashes. Traffic control becomes vital to reduce traffic problems. However, increasingly complex traffic conditions make traditional traffic light control methods inefficient because of predetermined time intervals for light changing. In the domain of traffic control, neural networks and reinforcement learning are used to change traffic lights accordingly (*1–4*). Fog computing frameworks are also implemented to control traffic lights considering traffic conditions at multiple intersections (*5*, *6*). Even though this research makes traffic control more efficient and has the potential to reduce traffic congestion, lack of large scale and high quality, real-time traffic data limits the practicability of these methods.

For real-time traffic data acquisition, different kinds of sensor are used to measure traffic volume, speed, and congestion levels (*5*, *7–9*). Arduino and infrared radiation (IR) sensors are used to monitor congestion levels, and GPS is used to obtain vehicle trajectories (*10–12*). These sensors are not capable of providing researchers with detailed traffic conditions and it is difficult to analyze vehicle and pedestrian motions and interactions (*13*). Besides these sensors, surveillance cameras provide researchers and city planners with real scenes of traffic conditions. With traffic cameras, it is possible to not only extract vehicle count, density, and speed, but also vehicle motion patterns, and behavior and interaction between vehicle and pedestrians (*14–22*). With comprehensive and rich content, surveillance videos can provide the task

[1]Urban Information Lab, The University of Texas at Austin, Austin, TX

Corresponding Author:
Huihai Wang, hw9998@utexas.edu

of traffic control with detailed and accurate traffic information.

Methods of object tracking can be categorized into motion-feature-based tracking and detection-based tracking. Motion-feature-based tracking uses motion features at pixel level to recognize objects based on predefined rules. These algorithms have low computation complexity and are not capable of dealing with complicated traffic conditions, such as traffic congestion, multiple object detection, and occlusion. With the development of artificial intelligence and computer vision, models such as You Only Look Once (YOLO), single shot multibox detection (SSD), and faster region-based convolutional neural network (R-CNN), detect objects from images based on object features (*23–25*). These convolutional neural network (CNN)-based models can reach high accuracy and are capable of detecting different kinds of objects in a complex environment. Intersection of union (IOU) is one of the crucial parameters to track detected objects. Huang et al. calculated the IOU of currently detected objects with predicted bounding boxes of objects detected in the former frame to identify the same objects in different frames (*26*). This algorithm achieves a high accuracy and is efficient to track vehicles because of its low computation complexity. A Kalman filter is also used to eliminate the occlusion between vehicles at the tracking process, such as simple online and realtime tracking (SORT) (*27*, *28*). These algorithms are built and tested considering the occlusion between targets and achieved high tracking accuracy. However, few studies have investigated the performance of a Kalman filter under the circumstance of background occlusions, such as large trees and light poles making the algorithm lose tracking of vehicles. As for high-level behavior recognition, different shapes of splines are used to recognize vehicle turning events (*18*, *29*). The limitation of this method is that it is only suitable for specific types of intersections; however, there are many more complex intersections in the real world, and vehicle behaviors are not limited to forward, left, and right. Recognizing and classifying vehicle trajectories comprehensively is crucial to traffic control and transportation management.

To test the performance of the Kalman filter on eliminating the loss of tracking at background occlusion, a Kalman filter was compared with an IOU-based tracker and a stand-alone IOU-based tracker to track vehicles at the condition of background occlusion. Then, a predefined road mask is used to determine vehicles' motion behaviors. The contributions of this research are as follows:

- The Kalman filter was tested with an IOU-based vehicle tracking algorithm when tracking vehicles at intersections under background occlusion, such as trees and light poles. The algorithm has low computation complexity and can deal with large volumes of videos at a high speed. This makes the algorithm suitable for processing traffic videos and obtaining real-time traffic information.
- The proposed Kalman-filter-based traffic behavior detection algorithm provides detailed real-time traffic information for traffic control. Predefined road masks are used to determine high-level vehicle behaviors to provide traffic information for traffic control. The turning event can link to real-world roads, which makes it possible to analyze and combine traffic conditions at different intersections.

The remainder of this paper is organized as follows. In the Literature Review, the related work in the field of vehicle tracking and high-level behavior recognition is described. In the Methods section, the algorithm is introduced in detail. The Results section shows the results of the research, and in the Discussion and Conclusion, the proposed method and potential further research are discussed.

Literature Review

Traffic cameras provide rich and real-time information for traffic management. Object detection and tracking algorithms are used to obtain object trajecotries from traffic videos, and high-level traffic behavior recognition methods are used to analyze traffic conditions with traffic data collected from traffic videos. This section introduces object tracking and behavior recognition algorithms in detail, and related research is discussed, as follows.

Object Detection and Tracking

Object tracking can be categorized into motion-feature-based tracking and detection-based tracking (*30*, *31*). For motion-based tracking, targets are detected by motion features based on fixed rules. This makes the method only applicable to tracking specific targets and have low robustness in complex situations. Object-tracking-based methods use deep-learning-based models to detect targets and are capable of detecting different kinds of objects in complex conditions. For the task of object tracking, IOU is a crucial parameter to determine whether a new detected object belongs to one of the existing trajectories. These object tracking methods are as follows.

Motion-Feature-Based Tracking. Majkowski et al. used morphological filtration to detect vehicles and a Kalman filter to track detected vehicles (*15*). This method uses predefined object features to detect vehicles, which makes it only capable of detecting objects with similar shapes, and it will be influenced by complex situations such as traffic congestion and occlusion. In Chowdhury et al., moving objects are extracted by identifying the foreground of the video input (*7*). Centroids of detected objects are then used from successive frames to form object trajectories. The Gaussian mixture model and a visual background extractor are used to initialize the background model, and feature points are calculated to match objects and contours in different frames to track objects. However, this frame requires the video in overhead view, and a change of optical conditions will have severe influence on the accuracy of the algorithm (*32*).

Object-Detection-Based Tracking. In Huang et al., the YOLO model is used to detect objects and generate bounding boxes (*26*). Object trajectories are determined based on the IOU of these boxes in consecutive frames. This method is efficient for tracking vehicles with lower computational complexity and is suitable for large volumes of traffic video processing. Yang et. al. proposed a framework consisting of a YOLO-based object detector and a CNN-based tracker (*33*). The result of the object detector is employed as the input for the tracker. A CNN is trained to extract image features and a correlation filter (CF) is used to determine the predicted object location in subsequent frames. Finally, the predicted location and IOU are used to track objects. For small object detection and tracking, Xu et al. used IOU and estimated subsequent location predicted by direction and velocity of the target to obtain the trajectory of pedestrians (*34*). The optical flow of targets and backgrounds are also used to predict vehicle location (*35*).

Real-time recurrent regression network (Re3) takes advantage of CNNs for image feature extraction and recurrent neural networks for temporal feature extraction from moving objects (*36*). Ren et al. uses SSD to generate initial vehicle bounding boxes and supply them to Re3 to obtain the trajectories (*18*).

SORT is also widely used in vehicle tracking. It combines the Kalman filter with IOU-based data association on top of a faster R-CNN detection framework (*27*). Then, a deep SORT is proposed by Xu and Wang; in their model, they add a deep association metric to judge whether two objects in two input images are the same, by matching deep features of target objects (*28*).

High-Level Traffic Behavior

High-level traffic behaviors, such as traffic safety, vehicle motion, and traffic conditions, are also extracted from traffic videos (*14*, *17*, *29*, *37–40*). Micro-level interaction of vehicles, such as lateral behavior, vehicle-following, and seeping behaviors, are recognized from traffic videos (*41*). Cui et al. built a highway imagery dataset with rich variance of scenes and road configurations using real-life traffic videos (*13*). AlexNet and GoogLeNet are employed to train classification models, and their performance shows that CNNs can achieve a high accuracy of 98% in recognizing highway traffic congestion. YOLO, deep convolution neural network (DCNN), and support vector machine (SVM) are also compared in the task of traffic congestion detection, and the results show that YOLO outperforms the other two models (*42*). Shah et al. uses faster R-CNN and an accident long short-term memory (LSTM) architecture to predict traffic accidents from traffic videos (*40*). The proposed architecture achieves an average of 1.359 s in relation to time-to-accident measure, with an average precision of 17.36%. Moradi et al. employed an unsupervised approach to extract traffic motion patterns from optical flow, which is not capable of recognizing movement patterns at object level (*19*). In Chen et al., trajectories are divided into different kinds of behaviors based on B-spline control points (*29*). Histograms of these behaviors and an SVM are then used to extract and classify dangerous behaviors. Ren et al. uses predefined cubic splines to determine trajectories' actions (*18*). The method is limited to a specific four-way intersection. However, there are much more complex intersections in the real world, and vehicle behaviors are not limited to forward, left, and right. How to recognize and classify vehicle trajectories comprehensively is crucial to traffic control and transportation management.

Methods

To achieve the goal of recognizing high-level vehicle behavior from traffic videos for traffic control at intersections, three steps are included in this research. First, vehicle detection and tracking. In this step, YOLOv5 was employed to detect vehicles for each frame in traffic videos (*43*). The YOLOv5 model is pretrained by the COCO dataset and can detect cars, buses, trucks, and motorcycles (*44*). For each detected vehicle, the coordinates of the bounding boxes and the confidence measure of the recognized vehicle are recorded. Then, vehicles are tracked based on the tracking algorithm in Huang et al., because of its high computation speed (*26*). However, the algorithm will lose objects under occlusion, and vibration of videos causes sudden changes of the detected vehicle velocity. So, a Kalman filter is used to estimate vehicle speed and predict locations of vehicles under occlusion. This process helps the tracker keep track of vehicles even though an occlusion occurs. In tracking processing, the same objects detected in the current frame and previous frame are identified based on IOU and the predicted locations and trajectories of each object are generated. At the second step, high-level vehicle behaviors are determined by a predefined road mask. The third step is to analyze and summarize vehicle behaviors to obtain traffic conditions at intersections.

Vehicle Tracking

The pseudo code of the vehicle tracking algorithm is shown as Figure 1. A pre-trained YOLOv5 model is used to detect vehicles in each frame. Four kinds of vehicle are detected in this research: car, bus, truck, and motorcycle. For each detected vehicle, the coordinates of the bounding boxes and the confidence measures of the recognized vehicles are recorded.

Trajectories are initialized by bounding boxes of detected vehicles in the first frame of traffic videos. Central points of lower boundaries are recorded as location of trajectories in the current frame, and coordinates of bounding boxes are defined as the rear of trajectories. Start frame and end frame indexes are also recorded for each trajectory. In the following frames, the IOU of detected vehicles and the rears of trajectories are calculated. Then, the maximum IOU for each vehicle is calculated, and a threshold value is used to determine which trajectory the newly detected vehicles belong to. For each trajectory whose rear is not updated, the predicted rear location is calculated based on historical trajectory data using the Kalman filter. The Kalman filter contains two stages (*45*). Equations 1 and 2 define the prior state estimate stage:

$$X_k = \begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} = AX_{k-1} + \mathrm{B}\begin{bmatrix} x_{k-1} \\ y_{k-1} \end{bmatrix}, \tag{1}$$

where

X_k = predicted state at frame k, including location and speed at x direction,

x_k and $\dot{x}_k$ = location and speed (in the 2D image coordinate system) at y direction y_k and $\dot{y}_k$, respectively,

x_{k-1} = acceleration (in the 2D image coordinate system) at x direction in frame $k-1$,

y_{k-1} = acceleration at y direction in frame $k-1$,

A = state transition matrix, and

B = control input matrix.

Then, the error covariance is calculated using Equation 2:

$$P_k = AP_{k-1}A^T + Q, \tag{2}$$

where

P_k = estimated error covariance matrix,

P_{k-1} = previous estimated error covariance matrix, and

Q = process noise covariance.

```
Algorithm 1 Pseudo code of the tracking step
F = {f_1, f_2, ...}: The video file is composed of a series of frames f_i, I = 1, 2, 3, ...
V_i = {v_i,1, v_i,2, ...}: Set of vehicles detected in the i th frame f_i
T = {t_1, t_2, ...}: The set of trajectories generated from detected vehicles
Starting with frame i = 1 and empty sets T
for frame i = 1, 2, 3, ..., until the end of video do
  V_i = YOLOv5(frame i)
  if T is empty then
    for the j th vehicle in V_i, j = 1, 2, 3, ... do
      create t_j with V_ij as the start point
    end for
  end if
  for the j th vehicle in V_i, j = 1, 2, 3, ... do
    for the m th trajectory in T, m = 1, 2, 3, ... do
      compute IOU value between V_ij and t_m
      if IOU(V_ij, t_m) is the maximum IOU for V_ij &  IOU(V_ij, t_m) > iou threshold then
        update t_m by V_ij
        calculate estimated tail location and speed of t_m based on Kalman Filter
      end if
    end for
  end for
  for the m th trajectory in T, m = 1, 2, 3, ... do
    if t_m is not updated then
      if i − last updated frame index > time threshold do
        end life of t_m and it will not be considered in the following calculation
        GOTO the next m
      end if
      predict tail location and keep vehicle box for IOU calculation in next video frame
    end if
end for
```

Figure 1. Pseudo-code of the vehicle tracking algorithm.

The initial state can be set randomly because errors will be corrected by Kalman gains and estimated error covariance.

The second stage of the Kalman filter is state update, which consists of three steps. The first step is Kalman gain generation, shown as Equation 3:

$$K_k = P_k H^T \left(HP_k H^T + R\right)^{-1}, \tag{3}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \tag{4}$$

$$R = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}, \tag{5}$$

where

K_k = Kalman gain,

H = measurement mapping matrix, and

σ_x (0.01) and σ_y (0.01) = magnitude of standard deviation of measurement in x direction and y direction, respectively.

Then, estimated state at frame k is calculated by Equation 6:

$$\hat{x}_k = X_k + K_k(Z_k - HX_k), \tag{6}$$

where

$\hat{x}_k$ = estimate state at frame k, and

Z_k = measured state at frame k.

Finally, estimated error convariance matrix is updated by Equation 7:

$$P_{k+1} = (I - K_k H)P_k, \tag{7}$$

where

P_{k+1} = estimated convariance matrix at frame $k+1$.

After the vehicle and trajectory matching process, new trajectories will be created by unmatched vehicles. Finally, rears and end frames for each trajectory are updated. Trajectories will be defined as finished if there are no newly detected vehicles appended to them beyond a time threshold and will not match vehicles in following frames.

High-Level Behaviors

In this step, the traffic behavior is defined as vehicle counts and turning movements. High-level behaviors for each trajectory generated from the last step are determined. The pseudo code is shown as Figure 2. Specifically, trajectories were classified to different turning actions between each pair of roads. Firstly, trajectories with short numbers of frames are deleted to remove noise caused by vibration of vehicle detections and occlusion by trees, traffic lights, or other vehicles. Then, road masks are defined for each road. Pixels in different roads are selected by the OpenCV package and are set to different values to distinguish them. The value for each pixel is road ID (from 1 to 3 in this case). The road mask is shown in Figure 3. Roads at right side of the image are the entrance to a plaza, so these entrances are excluded.

Then, the start and end locations for each trajectory are used to determine vehicle behavior. As shown in Figure 3, vehicle behaviors will be defined as: Road 1 to 2, Road 1 to 3, Road 2 to 1, Run in Road 2, Road 2 to 3, Road 3 to 1, Road 3 to 2, and Run in Road 3.

Results

Data Overview

The algorithm is implemented to analyze traffic behaviors at the intersection of Cooper Dr. and N. Lamar Blvd. (hereafter Lamar location) in Austin, Texas, U.S. Lamar location is a three-direction intersection, which is shown as Figure 3. Road 1 is Cooper Dr., Road 2 is Lamar Rd with vehicles driving north, and Road 3 is Lamar Rd with vehicles driving south. Consecutive traffic videos at Lamar locations are captured in seven 15 min intervals per day from 7:15 a.m. to 9:00 p.m. The size of each video is around 400 M with a resolution of 1920 × 1080 and FPS of 30. To obtain the traffic flow patterns at this intersection, 6 days of videos from Sunday to Friday (May 30, 2021, to June 4, 2021) are analyzed. The video data on June 04, 2021, is unavailable after 4:00 p.m. because of the angle change of the camera.

Algorithm 2 Pseudo code of the high level traffic behavior recognition algorithm

$T = \{t_1, t_2, \dots\}$: The set of trajectories generated from detected vehicles
$M = [m_{ij}, i = 0,1,2,\dots, j = 0,1,2,\dots]$: The road mask matrix
for the k th trajectory in T, $k = 1,2,3,\dots$ **do**
    calculate length of t_k
    if length of t_k < length threshold **then**
        GOTO the next k
    end if
    Start Road ID = m[x of t_k start point] [y of t_k start point]
    End Road ID = m[x of t_k end point] [y of t_k end point]
    Direction (Start_Road_ID, End_Road_ID) += 1
end for

Figure 2. Pseudo-code of the traffic behavior recognition algorithm.

Result of Object Tracking

Videos are processed with the proposed vehicle tracking algorithm. For Lamar location, the central part of the video was set as the study area. Vehicles driving through this area are detected and tracked. The IOU threshold is set as 0.5 based on the experiments. The object tracking results are analyzed in different conditions, which includes bad weather conditions, night conditions, crowded conditions, and low vehicle density conditions.

Figure 4 shows two good optical conditions at daytime with good weather. Two frames of the same video with a few seconds apart are shown in each row. Red boxes are bounding boxes of detected vehicles, and numbers inside boxes are IDs of corresponding trajectories. The first row of Figure 4 shows the condition of low vehicle density at Road 3 which has no occlusion between cars, and the second row shows crowded conditions at Road 3 with occlusions between cars. It can be seen that, at low density condition (Figure 4*a*), vehicles can be tracked with low occlusion by other vehicles and achieve a high accuracy. At crowded conditions, shown in Figure 4*b*, tracking accuracy is influenced by occlusion between vehicles. Cars with ID numbers 115 and 118 are detected and tracked correctly even between vehicles. But there was a failure to trace vehicle 122 at the second frame, which is because of the occlusion of vehicles behind it. Also, vehicles which are far away from traffic cameras are tracked with a lower accuracy.

Figure **5** shows two bad optical conditions, including bad weather and night-time conditions. Similar to Figure 4, two frames of the same video with a few seconds apart are compared and analyzed. The first row of Figure 5 shows bad weather condition, and the second row shows the performance of the tracking algorithm at night-time condition. From the results, it is possible to see that vehicles with ID numbers of 0, 18, 19, and 9 are tracked correctly, but the occlusion of the light pole at center of the video causes the loss of the truck 66 which is turning to Road 1. At night-time, because of the bad optical

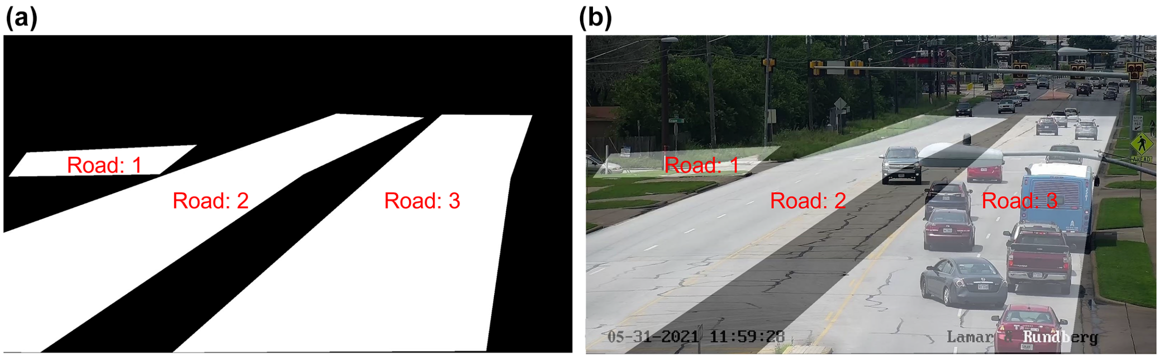

Figure 3. Road mask: (*a*) different values of each road (dark areas are set to 0), and (*b*) locations of roads in origin videos.

Figure 4. Tracking results at good optical conditions: (*a*) low density conditions (first row), and (*b*) high density conditions (second row).

conditions, the vehicle detection model cannot detect vehicles in areas which are too dark, or vehicles with very bright headlights. Also, features of vehicles with a long distance to traffic cameras are difficult to extract, which will influence the accuracy of the detection process.

These results show that occlusions between vehicles at crowded conditions make it hard to detect vehicles continuously, and the algorithm performs well when vehicles are blocked by other objects, such as light poles and trees.

Traffic Flow Analysis

Evaluation of Algorithms. From the tracking step, a list of trajectories for each video is obtained. Figure 6 shows trajectories at Lamar location from 9:30 a.m. to 9:45 a.m.

Figure 5. Tracking results at bad optical conditions.

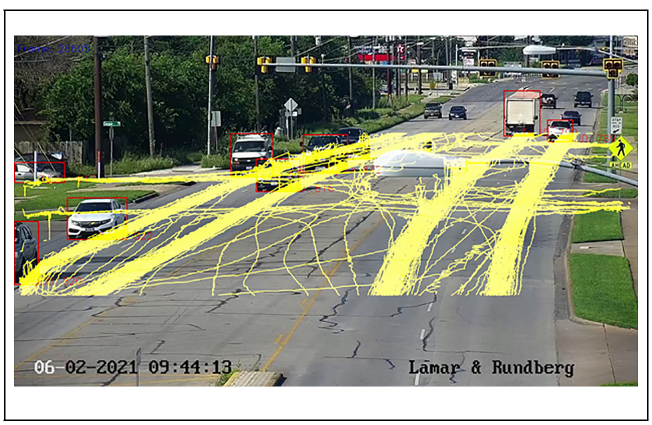

Figure 6. Vehicle trajectories.

on June 2, 2021. To remove noise caused by occlusion and vibrations of videos, the length between the start and end points for each trajectory was calculated, and the ones with a length smaller than 200 pixels were removed. Then, vehicle behaviors from these trajectories are extracted based on classification algorithms introduced in the Methods section.

To evaluate the performance of the Kalman filter with an IOU-based tracker under the circumstance of background occlusion, it is compared with the stand-alone IOU-based tracker. The vehicle behavior is extracted from the trajectories tracked by the algorithm, and the evaluation is based on the count of different vehicle behaviors, so this evaluation process tests both the tracking algorithm and the traffic behavior extraction process. A video with a length of 15 min is used to evaluate the algorithms. Counts of vehicles running on Road 2, Road 3, from Road 1 to Road 2, and from Road 1 to Road 3 are counted manually and compared with the detected results. Table 1 summarizes the counting results for each behavior, where KF_Based is the vehicle count of the Kalman filter with an IOU-based tracker at different directions, and IOU_Based is the vehicle count of the standalone IOU-based tracker at different directions. Dir2_2 is the behavior of continuing to drive in Road 2, Dir1_2 is the behavior of turning from Road 1 to Road 2, Dir3_3 is the behavior of continuing to drive in Road 3, and Dir1_3 is the behavior of turning from Road 1 to Road 3. The root mean square error (RMSE) is then calculated to evaluate these two algorithms. The RMSE of the Kalman filter and IOU-based algorithm is 10 and 40, respectively. The result shows that the Kalman-filter-based tracking and behavior algorithm outperforms the IOU-based algorithm.

Analysis of Traffic Flow. Figure 7 shows traffic flow at Lamar location from 7:15 a.m. to 20:59 p.m. in 6 days (May 30, 2021 to June 4, 2021). The time interval for each record is 15 min, the blue line shows traffic flow running on Road 2, and the orange line shows the traffic

Table 1. Vehicle Counts Summary

	Direction	KF_based	IOU_based	Real count
No occlusion	Dir2_2	133	167	152
	Dir1_2	5	9	6
Occlusion	Dir3_3	175	245	167
	Dir1_3	9	5	12

Note: KF_Based = vehicle count of the Kalman filter with an intersection of union (IOU)-based tracker at different directions; IOU_Based = vehicle count of the standalone IOU-based tracker at different directions.

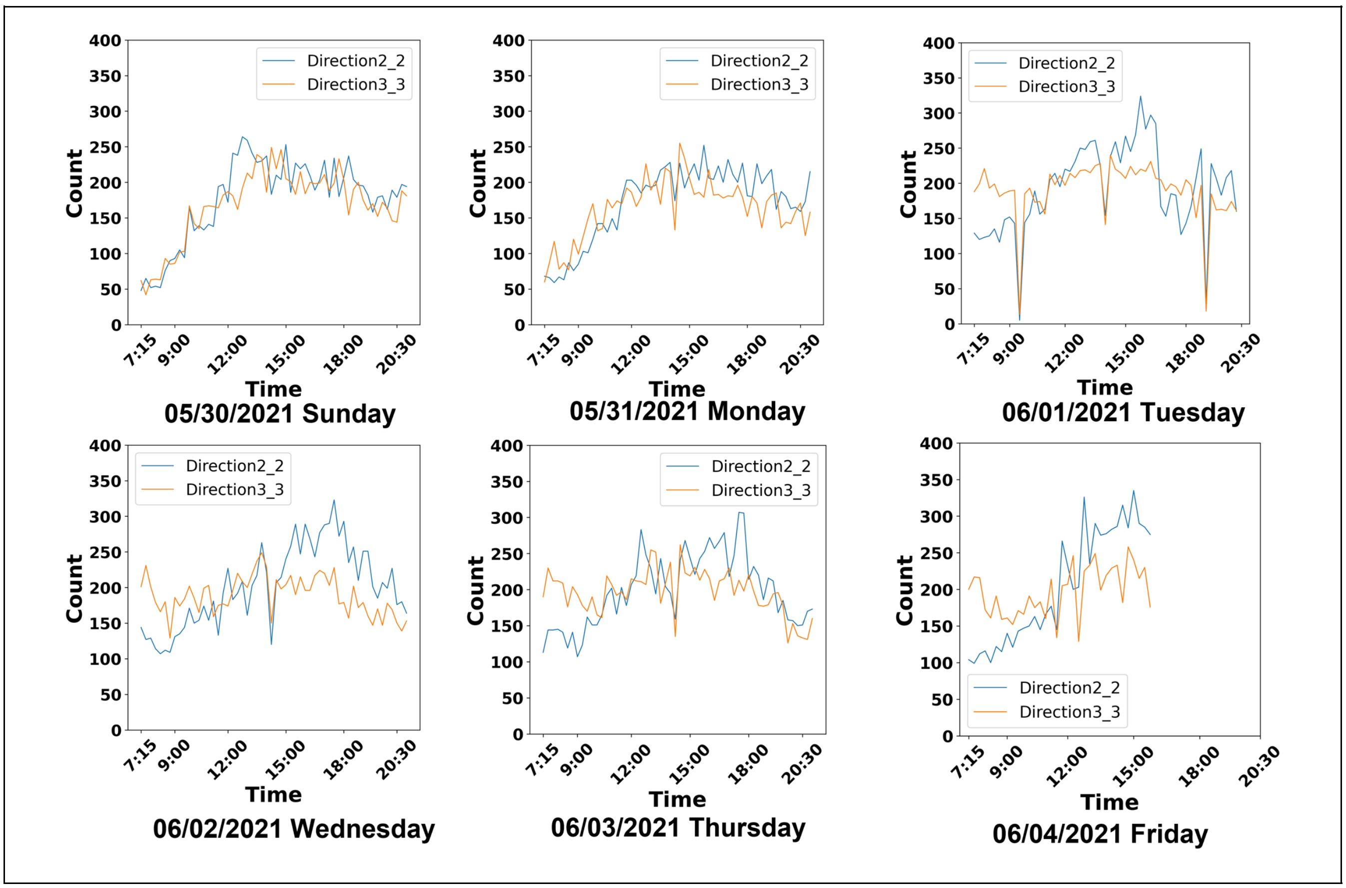

Figure 7. Traffic flow at Lamar location.

flow running on Road 3. From this figure, it can be seen that traffic flows at Road 2 and Road 3 have similar temporal characteristics. They both obtain peak traffic volume at around 3:00 to 5:00 p.m., and traffic volume starts to decrease at around 5:00 p.m. The change of traffic flow is continuous and self-dependent, so it is possible to predict traffic flow by the obvious data for the control of traffic lights. Traffic volume at Road 3 has a higher value than that of Road 2 in the morning (7:00 to 9:00 a.m.), an average of 170 and 130, respectively. In summary, Road 3 has heavier traffic than Road 2 in the morning, and traffic flow changes continuously and independently.

Discussion and Conclusion

In this study, a YOLO-based detector and a Kalman-filter-based tracker are tested at intersections which are occluded by background objects. The result shows that the Kalman filter is capable of tracking vehicles occluded by background objects for a relatively long time interval. Traffic flow in different roads and turning events are then recognized from trajectories based on predefined road masks. Under the circumstances of a rapid urbanization process and a surge in the number of vehicles, this real-time traffic data provides city planners and governments with precise and rich information for traffic management. Traffic flow prediction requires a high-quality historical traffic dataset, including trajectories collected from GPS devices and traffic count information from inductive loop devices (*46–48*). However, it is hard to obtain real-time data from these sources, and GPS devices require specific platforms and are not able to reflect real traffic flow information. Inductive loop sensors are widely used in traffic counting. However, they are not suitable for micro-scale traffic flow research as they are not able to reconstruct vehicle trajectories and recognize complex traffic behaviors. Compared with these data sources, the real-time traffic flow information extracted from traffic videos based on the proposed algorithms has the potential to predict traffic flow at real time and for a city-wise research. The proposed method links traffic turning behaviors to roads, which makes it possible to provide richer information by performing multi-intersection traffic flow analysis, and could provide input to adaptive signal control. The extracted vehicle trajectories contain detailed spatiotemporal information of vehicle movement. The speed can be calculated at a frequency of 30 times per second, which gives the potential to perform traffic safety analysis and traffic accident prediction.

From the results of the algorithms, three conclusions can be drawn. The Kalman filter with an IOU-based tracker performs well at the condition of backgroumd occlusion. Secondly, the proposed algorithm can detect and track vehicles at different optical conditions. Bad weather and night-time will influence the detection and tracking process in areas far from traffic cameras. The real-time traffic flow information extracted from traffic videos contains road information, so it can not only help with single intersection control, but also provide information for a road network. The temporal characteristic of observed traffic flow makes it able to train the model for predicting traffic condition based on detected real-time traffic flow, which will make traffic light control more efficient.

Our proposed method also has some limitations. Occlusion has a significant influence on the accuracy of the tracking process. Vehicles driving through the occlusion area will lose trajectory when performing complex behaviors, such as turning, changing speed, and waiting for traffic lights, which causes the overestimation of traffic flow and errors on turning event identification. The vision field of traffic videos also influences the algorithm. Some roads may not be captured by cameras, which causes the loss of information for specific roads. The authors' further work will focus on object tracking under the condition of occlusion and search for alternative data sources for real-time traffic flow detection at intersections.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: J. Jiao, H. Wang; data collection: J. Jiao, H. Wang; analysis and interpretation of results: J. Jiao, H. Wang; draft manuscript preparation: J. Jiao, H. Wang. All authors reviewed the results and approved the final version of the manuscript.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the University of Texas Good Systems Grand Challenge and the USDOT CM2 University Transportation Center at the University of Texas, Austin, and was supported by the National Science Foundation (NSF), ART-AI: Convergent, Responsible, and Ethical Artificial Intelligece Training Experience for Robotics (Award Number: 2125858), and National Science Foundation (NSF), SCC-CIVIC-FA Track A: Community Hub for Smart Mobility: A University-Government-Nonprofit Partnership (Award Number: 213302).

ORCID iDs

Junfeng Jiao https://orcid.org/0000-0002-7272-8805
Huihai Wang https://orcid.org/0000-0002-7343-3726

References

1. Garg, D., M. Chli, and G. Vogiatzis. Deep Reinforcement Learning for Autonomous Traffic Light Control. *Proc., 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, Singapore, 2018, pp. 214–218.
2. Liu, Y., L. Liu, and W.-P. Chen. Intelligent Traffic Light Control Using Distributed Multi-Agent Q Learning. *Proc., 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama, Japan, IEEE, 2017, pp. 1–8.
3. Arel, I., C. Liu, T. Urbanik, and A. G. Kohls. Reinforcement Learning-Based Multi-Agent System for Network Traffic Signal Control. *IET Intelligent Transport Systems*, Vol. 4, No. 2, 2010, pp. 128–135.
4. Wei, H, G Zheng, H Yao, and Z. Li IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control. *Proc., 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, UK, 2018, pp. 2496–2505.
5. Hossan, S., and N. Nower. Fog-Based Dynamic Traffic Light Control System for Improving Public Transport. *Public Transport*, Vol. 12, No. 2, 2020, pp. 431–454.
6. Liu, J., J. Li, L. Zhang, F. Dai, Y. Zhang, X. Meng, and S. Jian Secure Intelligent Traffic Light Control Using Fog

Computing. *Future Generation Computer Systems*, Vol. 78, 2018, pp. 817–824.
7. Chowdhury, M. F., M. R. A. Biplob, and J. Uddin Real Time Traffic Density Measurement using Computer Vision and Dynamic Traffic Control. *Proc., Joint 7th International Conference on Informatics, Electronics Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision Pattern Recognition (icIVPR)*, Fukuoka, Japan, 2018, pp. 353–356.
8. Alpatov, B. A., P. V. Babayan, and M. D. Ershov. Vehicle Detection and Counting System for Real-Time Traffic Surveillance. *Proc., 7th Mediterranean Conference on Embedded Computing (MECO)*, Budva, Montenegro, 2018, pp. 1–4.
9. An, C., Y. J. Wu, J. Xia, and W. Huang. Real-Time Queue Length Estimation Using Event-Based Advance Detector Data. *Journal of Intelligent Transportation Systems*, Vol. 22, No. 4, 2018, pp. 277–290.
10. Albatish, I. M., and S. S. Abu-Naser. Modeling and Controlling Smart Traffic Light System Using a Rule Based System. *Proc., International Conference on Promising Electronic Technologies (ICPET)*, Gaza, Palestine, 2019, pp. 55–60.
11. Díaz, N., J. Guerra, and J. Nicola. Smart Traffic Light Control System. *Proc., IEEE 3rd Ecuador Technical Chapters Meeting (ETCM)*, Cuenca, Ecuador, 2018, pp. 1–4.
12. Khattak, Z. H., M. J. Magalotti, and M. D. Fontaine. Operational Performance Evaluation of Adaptive Traffic Control Systems: A Bayesian Modeling Approach Using Real-World GPS and Private Sector PROBE Data. *Journal of Intelligent Transportation Systems*, Vol. 24, No. 2, 2020, pp. 156–170.
13. Cui, H., G. Yuan, N. Liu, M. Xu, and H. Song. Convolutional Neural Network for Recognizing Highway Traffic Congestion. *Journal of Intelligent Transportation Systems*, Vol. 24, No. 3, 2020, pp. 279–289.
14. Ashwin, S., R. A. Vasist, S. S. Hiremath, and H. R. Lakshmi. Automatic Control of Road Traffic Using Video Processing. *Proc., International Conference on Smart Technologies For Smart Nation (SmartTechCon)*, Bengaluru, India, 2017, pp. 1580–1584.
15. Majkowski, A., M. Kołodziej, T. Zabielski, P. Tarnowski, and R. J. Rak. Automatic Traffic Monitoring Using Images from Road Camera. *Proc., IEEE 21st International Conference on Computational Problems of Electrical Engineering (CPEE)*, Pińczów, Poland, 2020, pp. 1–4.
16. Eamthanakul, B., M. Ketcham, and N. Chumuang. The Traffic Congestion Investigating System by Image Processing from CCTV Camera. *Proc., International Conference on Digital Arts, Media and Technology (ICDAMT)*, Chiang Mai, Thailand, 2017, pp. 240–245.
17. Shokrolah, S. M., and B. T. Morris. Vision-Based Turning Movement Monitoring:Count, Speed Waiting Time Estimation. *IEEE Intelligent Transportation Systems Magazine*, Vol. 8, No. 1, 2016, pp. 23–34.
18. Ren, X., D. Wang, M. Laskey, and K. Goldberg. Learning Traffic Behaviors by Extracting Vehicle Trajectories from Online Video Streams. *Proc., IEEE 14th International Conference on Automation Science and Engineering (CASE)*, Munich, Germany, 2018, pp. 1276–1283.
19. Moradi, A., A. Shahbahrami, and A. Akoushideh. An Unsupervised Approach for Traffic Motion Patterns Extraction. *IET Image Processing*, Vol. 15, No. 2, 2021, pp. 428–442.
20. Morris, B. T., and M. M. Trivedi. A Survey of Vision-Based Trajectory Learning and Analysis for Surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 18, No. 8, 2008, pp. 1114–1127.
21. Lan, C. L., and G. L. Chang. A Traffic Signal Optimization Model for Intersections Experiencing Heavy Scooter–Vehicle Mixed Traffic Flows. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 16, No. 4, 2015, pp. 1771–1783.
22. Rahman, M., M. Islam, J. Calhoun, and M. Chowdhury. Real-Time Pedestrian Detection Approach with an Efficient Data Communication Bandwidth Strategy. *Transportation Research Record: Journal of the Transportation Research Board*, 2019. 2673: 129–139.
23. Redmon, J., S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. *Proc., IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016, pp. 779–788.
24. Liu, W., D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and C. Alexander. SSD: Single Shot MultiBox Detector. In *Computer Vision – ECCV 2016*. *Lecture Notes in Computer Science* (Leibe, B., J. Matas, N. Sebe, and M. Welling, eds.), vol. 9905. Springer, Cham, 2016, pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2.
25. Ren, S., K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv Preprint arXiv:1506.01497*, 2016.
26. Huang, L., W. Xu, S. Liu, V. Pandey, and N. R. Juri. Enabling Versatile Analysis of Large Scale Traffic Video Data With Deep Learning and HiveQL. *Proc., IEEE International Conference on Big Data (Big Data)*, Boston, MA, 2017, pp. 1153–1162.
27. Bewley, A., Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple Online and Realtime Tracking. *Proc., IEEE International Conference on Image Processing (ICIP)*, Phoenix, 2016, pp. 3464–3468.
28. Xu, Y., and J. Wang. A Unified Neural Network for Object Detection, Multiple Object Tracking and Vehicle Re-Identification. *arXiv Preprint arXiv:1907.03465*, 2019.
29. Chen, Z., C. Wu, Z. Huang, N. Lyu, Z. Hu, M. Zhong, Y. Cheng, and B. Ran. Dangerous Driving Behavior Detection Using Video-Extracted Vehicle Trajectory Histograms. *Journal of Intelligent Transportation Systems*, Vol. 21, No. 5, 2017, pp. 409–421.
30. Choi, W. Near-Online Multi-target Tracking with Aggregated Local Flow Descriptor. *Proc., International Conference on Computer Vision (ICCV)*. Santiago, Chile, 2015, pp. 3029–3037.
31. Yoon, J. H., M. H. Yang, J. Lim, and K. J. Yoon. Bayesian Multi-object Tracking Using Motion Context from Multiple Objects. *Proc., IEEE Winter Conference on Applications of Computer Vision*, Washington, D.C., 2015, pp. 33–40.

32. Zhang, Q., Z. Yang, and D. Sun. Automated Data Collection and Safety Analysis at Intersections Based on a Novel Video Processing System. *Transportation Research Record: Journal of the Transportation Research Board*, 2019. 2673: 136–144.
33. Yang, B., M. Tang, S. Chen, G. Wang, Y. Tan, and B. Li. A Vehicle Tracking Algorithm Combining Detector and Tracker. *EURASIP Journal on Image and Video Processing*, Vol. 2020, No. 1, 2020, p. 17.
34. Xu, W., N. Ruiz, K. Pierce, R. Huang, J. Meyer, and J. Duthie. Detecting Pedestrian Crossing Events in Large Video Data from Traffic Monitoring Cameras. *Proc., International Conference on Big Data (Big Data)*, Los Angeles, CA, 2019, pp. 3824–3831.
35. Ke, R., S. Feng, Z. Cui, and Y. Wang. Advanced Framework for Microscopic and Lane-level Macroscopic Traffic Parameters Estimation from UAV Video. *IET Intelligent Transport Systems*, Vol. 14, No. 7, 2020, pp. 724–734.
36. Gordon, D., A. Farhadi, and D. Fox. Re3: Real-Time Recurrent Regression Networks for Visual Tracking of Generic Objects. *arXiv Preprint arXiv:1705.06368*, 2018.
37. St-Aubin, P., L. Miranda-Moreno, and N. Saunier. An Automated Surrogate Safety Analysis at Protected Highway Ramps Using Cross-Sectional and Before–After Video Data. *Transportation Research Part C: Emerging Technologies*, Vol. 36, 2013, pp. 284–295.
38. Hu, W., X. Xiao, D. Xie, T. Tan, and S. Maybank. Traffic Accident Prediction Using 3-D Model-Based Vehicle Tracking. *IEEE Transactions on Vehicular Technology*, Vol. 53, No. 3, 2004, pp. 677–694.
39. Sun, Z., M. Yu, H. Wang, and X. Li. Effects of Traffic Cameras on Incident Management: Validation of Statistically Significant Effects Using Field Data. *Transportation Research Record: Journal of the Transportation Research Board*, 2015. 2484: 31–38.
40. Shah, A., J. B. Lamare, T. N. Anh, and A. Hauptmann. CADP: A Novel Dataset for CCTV Traffic Camera based Accident Analysis. *arXiv Preprint arXiv:1809.05782*, 2018.
41. Raju, N., P. Kumar, A. Jain, S. S. Arkatkar, and G. Joshi. Application of Trajectory Data for Investigating Vehicle Behavior in Mixed Traffic Environment. *Transportation Research Record: Journal of the Transportation Research Board*, 2018. 2672: 122–133.
42. Chakraborty, P., Y. O. Adu-Gyamfi, S. Poddar, V. Ahsani, A. Sharma, and S. Sarkar. Traffic Congestion Detection from Camera Images using Deep Convolution Neural Networks. *Transportation Research Record: Journal of the Transportation Research Board*, 2018. 2672: 222–231.
43. Thuan, D. Evolution of YOLO Algorithm and YOLOv5: The State-of-the-Art Object Detection Algorithm. 2021.
44. GitHub. Releases ultralytics/yolov5. https://github.com/ultralytics/yolov5/releases. Accessed November 13, 2021.
45. Li, X., K. Wang, W. Wang, and Y. Li. A Multiple Object Tracking Method Using Kalman Filter. *Proc., International Conference on Information and Automation*, Harbin, China, 2010, pp. 1862–1866.
46. Toan, T. D., and V. H. Truong. Support Vector Machine for Short-Term Traffic Flow Prediction and Improvement of Its Model Training using Nearest Neighbor Approach. *Transportation Research Record: Journal of the Transportation Research Board*, 2021. 2675: 362–373.
47. Ye, P., B. Wu, and W. Fan. Modified Betweenness-Based Measure for Prediction of Traffic Flow on Urban Roads. *Transportation Research Record: Journal of the Transportation Research Board*, 2016. 2563: 144–150.
48. Khan, Z., S. M. Khan, K. Dey, and M. Chowdhury. Development and Evaluation of Recurrent Neural Network-Based Models for Hourly Traffic Volume and Annual Average Daily Traffic Prediction. *Transportation Research Record: Journal of the Transportation Research Board*, 2019. 2673: 489–503.