Lane Detection in Surveillance Videos Using Vector-Based Hierarchy Clustering and Density Verification

Shan-Yun Teng† **Kun-Ta Chuang**† **Chun-Rong Huang**‡ **Cheng-Chun Li**†
Dept. of Computer Science and Information Engineering, National Cheng Kung University†
Dept. of Computer Science and Engineering, National Chung Hsing University‡
syteng@netdb.csie.ncku.edu.tw, ktchuang@mail.ncku.edu.tw
crhuang@nchu.edu.tw, johnbelieve2@gmail.com

Abstract

Automatic lane detection is known to facilitate the real-time traffic planning and identify traffic congestion. In this paper, we develop a visual surveillance trajectory clustering (VSTC) framework for automatic lane detection. Given a surveillance video, trajectories of vehicles are extracted at first. These trajectories contain behavior of vehicles on different lanes and are clustered by VSTC to retrieve candidate lanes. Finally, a density verification is applied to identify the correct lanes from candidate lanes. As shown in the experiments, our framework can identify the lanes by using trajectories without prior knowledge.

1 Introduction

In visual surveillance, automatic lane network analysis becomes an important research topic to provide traffic information of the road [8]. According to the analysis results, the traffic bottleneck can be identified and automatic traffic light optimization [9] can be performed to reduce the traffic congestion. To achieve automatic lane network analysis, automatic lane detection without prior knowledge is required.

Because GPS trajectory records of vehicles contain locations of vehicles, learning from GPS-based trajectories provides the possibility to detect lanes without the manual interpretation [2]. However, GPS trajectory records contain residual errors in meters. The inaccurate positions of GPS trajectory records cause false detection of lanes and limit the practical usage of identifying lanes of roads. Moreover, parsing long-term GPS trajectory records is time-consuming as shown in many GPS trajectory analysis methods [2].

To achieve tracking and monitoring real-time traffic situations in the city, we consider using video trajectories extracted from surveillance videos to detect lanes of roads. Recently, foreground object detection and tracking methods [5] enable the applications to automatically extract moving trajectories from videos. It becomes a cost-effective and feasible manner to analyze moving trajectories of vehicles and pedestrians from surveillance videos. In general, each vehicle drives on its own lane unless it changes lanes or turns. When sufficient trajectories are collected, the lanes of the roads can be identified.

After trajectories are extracted by [5], a naive idea is to directly apply traditional GPS-based clustering approaches [1] to segment the geographic space according to the measurement of similar trajectories in the Euclidean metric. The geographic area covered by trajectories within the same cluster is then considered as a lane. However, such approach will encounter the following problems:

- Due to the variety of the trajectory lengths, it is difficult to utilize the Euclidean distances as the similarity matrix between two trajectories.
- Vehicles have the speed-up intention in the straight line but decelerate while doing the lane change and turning. The highly diverse situations incur the difficulty of designing the similarity measurement between any complete trajectories.

To solve these problems, we propose a visual surveillance trajectory clustering *(VSTC)* framework. In *VSTC*, each moving trajectory is separated into several disjoint vectors according to the directional changes within a trajectory. The first step of the clustering is to generate clusters which tend to contain vectors of similar lengths and similar angles with close positions. The vector-based clustering is then applied to overcome problems in trajectory-based clustering with the Euclidean metric. Then, we develop a novel framework to hierarchically merge vector-based clusters so that the curved lanes can also be constructed in *VSTC*. Finally, a density verification is proposed to separate mis-merged adjacent lanes during clustering.

2 Related Work

Recently, several approaches [2] aim to build road maps based on GPS trajectory records of vehicles. These methods mainly focus on the lane direction identification and vehicle routing analysis. Because spatial resolutions of GPS data, these methods are difficult to be applied to retrieve the correct number of the lanes and the widths of each lane of the roads. To solve the lane detection problem, Chen *et al.* [3] proposed using Gaussian mixture models to identify centers and widths of lanes by assuming that the widths of lanes are the same and the center lines of lanes are given. However, the viewing angles of surveillance cameras are different, resulting in that the widths of lanes may be different. The solution in [3] cannot be used to the case of surveillance videos. Currently, automatic lane detection from surveillance videos remains unexplored.

In addition, many approaches [8] were proposed to detect lanes from cameras set on vehicles for advanced driver-assistance systems. However, these approaches aim to detect the driving lane of the vehicle instead of identifying all lanes of roads. Other techniques to detect the traffic line from videos [6][7] are not the cure of the lane detection. It is believed that the traffic line is not always clearly revealed due to issues of the rain or the video resolution. Many relatively small pathways may lack for the traffic line inherently, making the line detection infeasible. These solutions are difficult to be

applied in the lane detection by using trajectories in surveillance videos.

3 The VSTC Framework

The *VSTC* framework is proposed to automatically identify the lane structure from trajectories of surveillance videos. Initially, we apply [5] to acquire trajectories of vehicles from surveillance videos. It is noted that the extracted trajectories usually contain noisy data, such as the false detection of dynamic background. Moreover, fragmented trajectories can also be observed due to visual obstacles or tracking failures. Identifying lanes from these trajectories becomes a challenging problem because these interferences are inevitable. With the increasing number of trajectories, these noisy data are hard to be effectively and manually filtered out. As a result, single-step clustering algorithms will be infeasible since they are difficult to distinguish correct trajectories from noises.

To solve aforementioned issues, we design *VSTC* as a cascaded framework. All trajectories, including noisy and correct trajectories, are individually segmented into disjoint sub-trajectories according to inertia of moving directions. Clustering is then performed based on sub-trajectories to obtain a large set of direction-diverse clusters, in which noisy or disordered data will be separated into relatively small clusters with a few instances. Afterward, direction-diverse clusters will be used to retrieve statistically connected clusters as candidates of lanes. Smaller disordered clusters will be removed in this step. Nevertheless, two or more lanes may be merged into the same cluster due to the over-clustering effect. To solve the problem, a density verification process is utilized to identify correct lanes.

Trajectory Transformation We first describe the flow to transform data extracted from videos to the trajectory format used in *VSTC*. Generally, a moving object o_i identified by foreground object detection algorithms [5] will be represented as a spatiotemporal list, i.e., $o_i = \{bb_{t_s}, bb_{t_{s+1}}, ..., bb_{t_{s+n}}\}$, where bb_{t_m} corresponds to the bounding box of o_i in the m^{th} video frame. Please note that all objects, either vehicles or pedestrians, have different sizes of bounding boxes. We thus transform each bounding box bb_{t_m} of o_i to a representative point $p_i(t_m)$, which is the central point of bb_{t_m}. Following the step of transformation, we have the set of moving trajectories T, in which each moving trajectory $t_i = \{p_i(t_s), p_i(t_{s+1}), ..., p_i(t_{s+n})\}$, representing the spatiotemporal point list of t_i.

Vector-Based Hierarchy Clustering In this step, we focus on resolving the issue of distance measurement between trajectories. Please note that moving trajectories have variant lengths due to the moving speed, traffic congestion, curved/straight lanes, and so no. Motivated by the nature of inertia in the object movement, we consider to segment each trajectory into a set of inertial sub-trajectories. Each sub-trajectory represents a list of sequential points with similar directions. To achieve this, we utilize the DouglasPeucker algorithm [4] to segment trajectories into inertial vectors. Due to limited space, please refer to [4] for detailed implementation. In this way, when a vehicle turns or drifts from its original lane, these situations can still be modeled by the inertial vectors which provide the moving behavior of objects during clustering.

After applying the segmentation, each trajectory t_i is transformed into a set of sub-trajectories. To represent the sub-trajectory, a representative vector rv_{i_j}, which is the vector computed from the head point to the tail point of each sub-trajectory, is applied. Each rv_{i_j} is the representation for a kind of moving behavior. Since the behavior of vehicles moving in the same lane should be consistent, to apply clustering on representative vectors is an effective way to identify the lane structure. In general, a surveillance video may contain more than ten thousands of trajectories. The calculation of the pair-wise similarity between all representative vectors is very time-consuming. In *VSTC*, we are inspired that the moving direction of representative vectors within a local region should be orderly, i.e. vectors are spatially separated based on the directions. To reduce the computational complexity, we devise the *VSTC* framework as a cascaded clustering methodology. The first step of the clustering is to generate clusters which contain directionally similar vectors. Then, clusters are merged by considering the connectivity between clusters to identify the spatial properties of clusters, such as clusters contain long or curved shapes of trajectories. Because noisy vectors are disordered, the size of clusters of noisy vectors will be small. Thus, clusters of noisy vectors are difficult to be merged when considering the connectivity with other clusters. Finally, the geographic shapes of these large clusters are transformed to the areas of lanes and form the candidate lanes for verification. We outline the details of the cascaded algorithms in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1 Directional Partitioning of Representative Vectors

Input: Representative vector list $RV = rv_1, ..., rv_i, ..., rv_i$, where i is the total number of trajectories; angle range r to decide the directional sub-groups;
Output: Directional sub-groups S_g;
1: $S_g = (S_{g_1}, S_{g_2}, ..., S_{g_m})$, where $m = \frac{360}{r}$;
2: vectorStandard $= (0, 0, 0, 1)$;
3: **for** each $rv_i \in RV$ **do**
4: **for** each $rv_{i_j} \in rv_i$ **do**
5: calculate θ between rv_{i_j} and vectorStandard;
6: add rv_{i_j} to S_{g_m}, where $m = (\frac{\theta}{r})\%n + 1$;
7: **return** S_g

Definition 1. Given a list of represented vectors $RV = \{rv_1, rv_2..., rv_n\}$, the dominant direction of RV is defined as:

$$dir(RV) = \frac{rv_1 + rv_2 + ... + rv_n}{size(RV)}.$$

Lane Detection based on Density Verification After the cascaded clustering algorithm, a set of clusters is obtained. Each major cluster possibly represents trajectories on a lane because these trajectories have

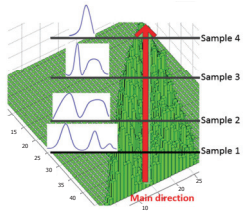


Figure 1: The illustration of density verification for adjacent lanes.

Algorithm 2 Vector-based Cascaded Clustering

Input: vector list $S_{g_i} \in S_g$ and $S_{g_i} = (rv_1, rv_2..., rv_n)$; threshold ϑ_m for the maximal spatial distance for two vectors; threshold θ_m for maximal angle for two vectors;
Output: cluster sets: $C = (C_1, C_2,...,C_n)$

initialize each $rv_j \in S_{gi}$ as a cluster c_j;
C.Add(c_j);
while $size(C) > 0$ **do**
  for each pair (c_x,c_y) **do**
    compute VSimilarity($c_x, c_y, \vartheta, \theta$);
  find the pair P=(c_x, c_y) with the highest VSimilarity;
  if P= $\emptyset$ **then**
    break;
  else
    merge c_y into c_x;
    C.remove(c_y); C.update(c_x);
P=(c_1, c_2);
while P$\neq \emptyset$ **do**
  for each pair (c_a,c_b) $\in C$ **do**
    compute $CD(c_a, c_b) : dis(RV_a, RV_b) = min(|RV_a.tail - RV_b.head|or|RV_b.tail - RV_a.head|)$;
  find the pair P=(c_a, c_b) with the minimum connected dissimilarity $CD(c_a, c_b)$;
  merge c_a into c_b;
  C.remove(c_a); C.update(c_b);
return C

Algorithm 3 VSimilarity: Direction Similarity

procedure VSimilarity($c_x, c_y, \vartheta, \theta$)
  θ =angle(rv_x, rv_y);
  if $\theta < \theta_m$ **then**
    Set $rv' = \overrightarrow{rv_y.startPos, rv_x}$;
    $crossValue = rv' \times rv_y$;
    $dist = \frac{|crossValue|}{lengrh(rv_y)}$;
  if $dist < \vartheta$ **then**
    return $dist$
  else
    return $NULL$

similar behavior and move at the same spatial directions. However, the cluster is likely to merge trajectories from two or more adjacent lanes because trajectories have the same direction and objects move across these lanes frequently. In addition, each cluster may still contain noisy trajectories which will reduce the precision of our lane detection results. To solve these problems, a density verification process is proposed to refine the lane detection results of the vector-based cascaded clustering.

Although trajectories in each cluster may contain noisy trajectories, the dominant moving directions of vehicles in a lane, which are identified by Definition 1 with the average of directional factors, can smooth the side-effect from noises. The dominant moving direction is determined to decide the direction when segmenting the cluster of more than two lanes. The next step is to quantify the frequency of every spatial grid by counting the number of vectors passing through the grid. As illustrated in Figure 1, the valley shows that there are few vehicles passing through the grid. By verifying the linkage among valleys, the boundary between adjacent lanes can be identified. Sometimes irregular moving behavior may move outside the boundary of lanes, causing clusters are enlarged due to the outlier effect. When we statistically remove exterior grids of the cluster with the less frequency, outliers in each cluster are also filtered. The process of density verification is outlined in Algorithm 4.

4 Experimental Results

In the experiments, three surveillance datasets were used. The video lengths, numbers of trajectories ($|T|$)

Algorithm 4 Density Verification

Input: cluster c_i; grid list CB of c_i; frequency list of grids SUP of c_i;
Output: dominant moving vector dv of c_i

D =**FindDenseRegions**(CB);
p_i=select top Dense element from D and get the list of separation point;
for each $cb_i \in CB$ **do**
  if IsDenseRegions(cb_i, sup_i) **then**
    $DR.add(cb_i)$;
calculate dv
for each $cb_i \in CB$ **do**
  if WhichGrid(cb_i) $\notin DR$ & IsOutlier(cb_i) = $ture$ **then**
    remove(cb_i);
return dv, p_i

and the resolutions of each dataset are shown in Table 1. The trajectories of vehicles were acquired by [5] in advance. The implementation of *VSTC* is based on JAVA on a 3.40 GHz Core i7 machine with 4 gigabytes of main memory, running on the Windows 7.

In the experiments, we show the lane detection results of the vector-based cascaded clustering and the density verification, respectively. The scene of video 1 shown in Figure 2(a) contains two curved lanes. The extracted trajectories in points after representative vector sampling are shown in Figure 2(b). Due to false foreground object detection of dynamic backgrounds such as waving trees, many noisy trajectories can be observed in Figure 2(b).

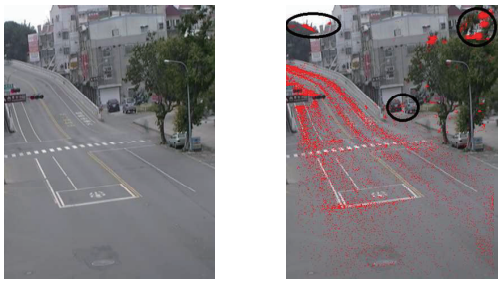
After applying the vector-based cascaded clustering, two dominant clusters were obtained as shown in Figure 3(a) and Figure 3(b), respectively. Although two dominant lanes in video 1 are detected, the areas of the lanes are enlarged unexpectedly. This is because some external noises generated from foreground extraction or irregular moving trajectories were merged into the major clusters. Moreover, the over-clustering effect is the inherent issue during clustering. As a result, the density verification is necessary to overcome the disadvantages mentioned above. As shown in Figure 3(c) and Figure 3(d), the shapes of detected lanes are more precisely to match the real lane range line drawing on the ground. With more trajectories, the results of the proposed approach can be more accurate.

Table 1: Summary of surveillance datasets

Dataset	video 1	video 2	video 3
Video Length	72 min.	58 min.	50 min.
$\|T\|$	42518	51774	72109
Resolution	370 × 480	720 × 480	857 × 481
Exec. time	905 sec.	697 sec.	1162 sec.

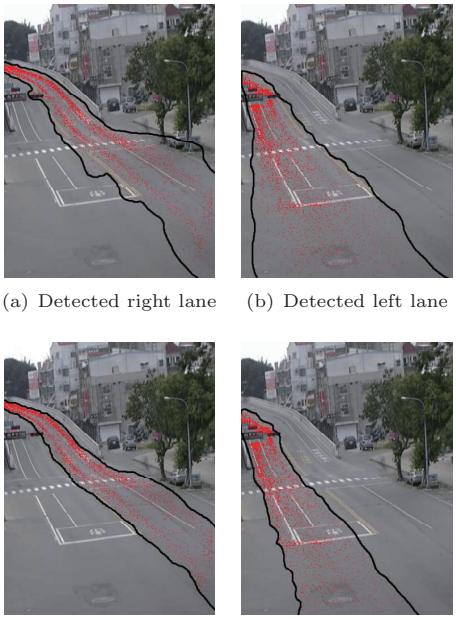
The results of lane detection in video 2 are shown in Figure 4(a) and Figure 4(b), respectively. This case shows that *VSTC* can precisely achieve the lane detection when the lane shape diversifies vertically due to the effect from visual angles. As shown in Figure 4(a), the small horizontal lane without traffic lines can still be identified. Moreover, multiple vertical lanes (marked by different colors) of the road can also be identified as shown in Figure 4(b).

The results of lane detection for video 3 are shown in Figure 5(a) and Figure 5(b). The surveillance camera captured the cross road from a high building to provide the bird's eye view of roads. Visual obstacles from trees lead to break the detection of a moving trajectory when performing the foreground object tracking. The result in Figure 5(b) shows that clustering by connectivity



(a) Background frame (b) Extracted traj. data

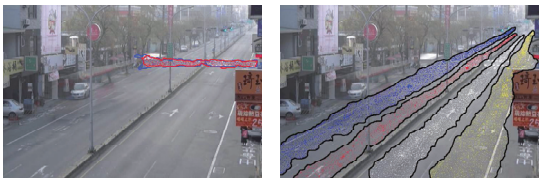
Figure 2: The background scene of video 1.



(a) Detected right lane (b) Detected left lane

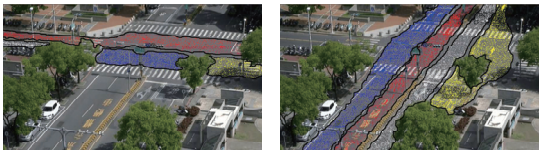
(c) right lane (d) left lane

Figure 3: The lane detection after vector-based cascaded clustering.



(a) Detected horizontal lanes (b) Detected vertical lanes

Figure 4: Results of lane detection in video 2.



(a) Detected horizontal lanes (b) Detected vertical lanes

Figure 5: Results of lane detection in video 3.

(in the second step of vector-based cascaded clustering) is able to partially resolve the problem if clusters can be spatially connected, and lanes can be precisely separated. In Figure 5(a), the detected lanes are broken because trajectories of clusters are not spatially connected. To resolve this, checking the spatiotemporal relationship between trajectories is necessary, which will be one topic of our future studies.

5 Conclusions

In this paper, we propose a novel *VSTC* framework for lane detection from surveillance videos. By considering the vector-based hierarchical clustering and the dense verification of trajectories, the lanes of roads can be identified. In the future, we will focus on linking broken trajectories due to visual obstacles. Moreover, we will apply the proposed method to detect lanes of more complicated scenes, such as roundabouts, and system interchanges with different ways.

Acknowledgement This paper was supported in part by National Science Council of Taiwan under Contract NSC-101-2221-E-005-086-MY3, and the Ministry of Economic Affairs under the grant 103-EC-17-A-02-S1-222. The authors would also like to thank Department of Industrial Technology, Ministry of Economic Affairs, ROC for the financial support under the grant, 1Z1030159, of Small Business Innovation Research Program.

References

[1] D. Ashbrook and T. Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, October 2003.

[2] L. Cao and J. Krumm. From GPS traces to a routable road map. In *Proc. of the ACM SIGSPATIAL*, 2009.

[3] Y. Chen and J. Krumm. Probabilistic modeling of traffic lanes from GPS traces. In *Proc. of the ACM SIGSPATIAL*, 2010.

[4] J. Hershberger and J. Snoeyink. An o (n log n) implementation of the douglas-peucker algorithm for line simplification. In *Proc. of the ACM Symp. on Computational geometry*, 1994.

[5] C.-R. Huang, P.-C. Chung, D.-K. Yang, H.-C. Chen, and G.-J. Huang. Maximum a posteriori probability estimation for online surveillance video synopsis. *Circuits and Systems for Video Technology, IEEE Trans. on*, 24(8):1417–1429, Aug 2014.

[6] J. Melo, A. Naftel, R. Bernardino, and J. Santos-victor. Detection and classification of highway lanes using vehicle motion trajectories. *ITS, IEEE Trans. on*, 7(2):188–200, June 2006.

[7] J. Ren, Y. Chen, L. Xin, and J. Shi. Lane detection in video-based intelligent transportation monitoring via fast extracting and clustering of vehicle motion trajectories. *Mathematical Problems in Engineering*, 2014.

[8] A. Saha, D. D. Roy, T. Alam, and K. Deb. Automated road lane detection for intelligent vehicles. *Global Journal of Computer Science and Technology*, 12(6), Mar 2012.

[9] A. Salkham, R. Cunningham, A. Garg, and V. Cahill. A collaborative reinforcement learning approach to urban traffic control optimization. In *Proc. of the WI-IAT*, 2008.