Correspondence

Counting Vehicles from Semantic Regions

Rui Zhao, *Student Member, IEEE*, and
Xiaogang Wang, *Member, IEEE*

***Abstract*—Automatically counting vehicles in complex traffic scenes from videos is challenging. Detection and tracking algorithms may fail due to occlusions, scene clutters, and large variations of viewpoints and vehicle types. We propose a new approach of counting vehicles through exploiting contextual regularities from scene structures. It breaks the problem into simpler problems, which count vehicles on each path separately. The model of each path and its source and sink add strong regularization on the motion and the sizes of vehicles and can thus significantly improve the accuracy of vehicle counting. Our approach is based on tracking and clustering feature points and can be summarized in threefold. First, an algorithm is proposed to automatically learn the models of scene structures. A traffic scene is segmented into local semantic regions by exploiting the temporal cooccurrence of local motions. Local semantic regions are connected into global complete paths using the proposed fast marching algorithm. Sources and sinks are estimated from the models of semantic regions. Second, an algorithm is proposed to cluster trajectories of feature points into objects and to estimate average vehicle sizes at different locations from initial clustering results. Third, trajectories of features points are often fragmented due to occlusions. By integrating the spatiotemporal features of trajectory clusters with contextual models of paths and sources and sinks, trajectory clusters are assigned into different paths and connected into complete trajectories. Experimental results on a complex traffic scene show the effectiveness of our approach.**

***Index Terms*—Semantic region, surveillance, trajectory clustering, vehicle counting.**

I. Introduction

Counting vehicles in traffic scenes by video surveillance is of great interest for traffic management and urban planning. Many existing object counting approaches [10], [17], [22] rely on a pedestrian or vehicle detector based on the appearance of objects or background subtraction results. However, it is difficult to design a vehicle detector that robustly works in all kinds of complex traffic scenes since it has to handle different types of vehicles, such as sedans, trucks, vans, and buses, which are observed in different views. Vehicle detection is also challenging with the existence of occlusions and scene clutters. In recent years, some approaches [1]–[3], [7], [14], [15] have been proposed to count objects without relying on detectors. They tracked and clustered feature points or analyzed dynamic textures in the scenes. Some of them [2], [3], [7] required training data from the target scenes.

Manuscript received July 22, 2012; revised December 19, 2012; accepted February 7, 2013. This work was supported by the General Research Fund sponsored by the Research Grants Council of Hong Kong under Project CUHK417110 and Project CUHK417011 and by the National Natural Science Foundation of China under Project 61005057. The Associate Editor for this paper was S. Sun.

R. Zhao is with the Department of Electronic Engineering, the Chinese University of Hong Kong, Shatin, Hong Kong (e-mail: rzhao@ee.cuhk.edu.hk).

X. Wang is with the Department of Electronic Engineering, the Chinese University of Hong Kong, Shatin, Hong Kong, and also with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, 518055, Shenzhen, China (e-mail: xgwang@ee.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2013.2248001

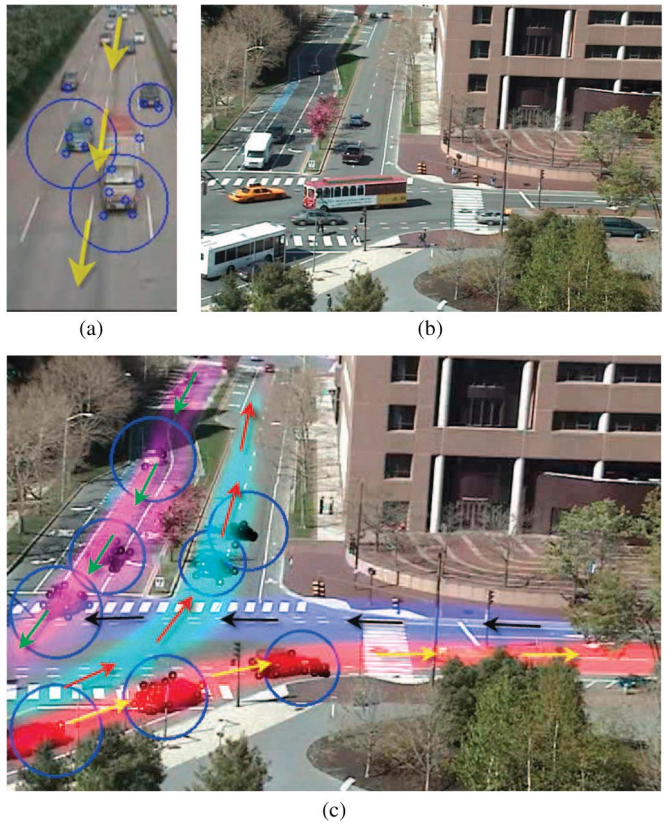


Fig. 1. Counting vehicles by tracking and clustering feature points. (a) Counting vehicles on a single path. (b) Complex traffic scene with multiple intersecting paths. (c) Counting vehicles on each of the intersecting paths separately. The spatial distributions of different paths are indicated by colors.

We propose an approach of counting vehicles by tracking feature points in the scene and clustering their trajectories into separate vehicles. If there is only a single vehicle path in the traffic scene, e.g., as shown in Fig. 1(a), the counting problem becomes relatively easy since the movements of vehicles are strongly regularized by the path. Vehicles must appear and disappear in fixed regions, which are called the source and the sink of the path. The trajectories of the feature points on vehicles must be within the region specified by the path. If two vehicles are on the same lane, their order of passing through the path cannot be changed. The average size of vehicles observed at the same location can be estimated. The problem becomes more challenging if the traffic scene is complex and has multiple intersecting paths, e.g., as shown in Fig. 1(b), since different moving patterns of vehicles are mixed. Existing approaches counted objects in the scene altogether, and the contextual information provided by the path models was not well exploited. This is partially due to the fact that it is not easy to learn automatically and accurately the models of paths in complex traffic scenes where object detection and tracking are not reliable.

A. Our Approach

Our approach automatically learns the models of scene structures and breaks the problem of counting vehicles in complex scenes into simpler problems, which count vehicles on each path separately. It first

learns the models of semantic regions, which are parts of paths, using our approach previously proposed in [20]. Wang *et al.* [20] learned that the models in the semantic regions are based on the cooccurrence of local motions. It did not require object detection and tracking. We propose a fast marching algorithm to automatically connect local semantic regions into complete global paths. The models of sources and sinks of the paths are also estimated from the models of semantic regions. Then, an algorithm is proposed to estimate the average vehicle sizes at different locations along each path and to cluster trajectories of feature points into objects according to the estimated average vehicle sizes. Due to occlusions among objects, trajectories of features points on the same object may be broken into multiple parts. Our approach assigns trajectory clusters into different paths and connected them by integrating multiple cues, including the spatiotemporal features of trajectories, the spatial distributions of vehicle paths, and the models of sources and sinks.

B. Related Work

In recent years, some approaches were proposed to count objects by clustering trajectories of features points or modeling dynamic textures. They are suitable for the cases when object detectors fail. Brostow and Cipolla [1] proposed an unsupervised Bayesian clustering algorithm to group probabilistically the trajectories of feature points into clusters, which independently represented moving entities. Rabaud and Belongie [14] proposed a conditioning algorithm to smooth and extend fragmented trajectories and clustered them into objects with an object model learned from training frames labeled with the ground-truth object count. Chan *et al.* [2], [3] used a Poisson regression and a Gaussian process to estimate the number of pedestrians by modeling dynamic textures. They first segmented the crowd into regions with homogeneous motion and then extracted low-level features from each segmented region for regression. The approaches of [2], [3], and [14] required manually labeled training frames.

Many approaches [5], [13], [21] learn path models through clustering complete trajectories of objects. They do not work well in complex traffic scenes where the trajectories of objects are highly fragmented and misassociated. Some approaches [9], [20] learned semantic regions, which are local subregions on the paths, by modeling temporal and spatial correlations of local motions. Without relying on object detection and tracking, they worked well in complex traffic scenes. Based on their results, we proposed a fast marching algorithm to connect local semantic regions into complete global paths.

Sources and sinks are the regions where objects appear and disappear. If the starting or ending points of trajectories are not observed in source/sink regions, it indicates tracking failures. To estimate sources and sinks, most approaches [12], [18], [21] used Gaussian mixture models (GMMs) to fit starting/ending points of trajectories. However, in complex scenes, trajectories are often fragmented because of object interactions, occlusions, and scene clutters. Therefore, the estimation was biased by the false entry/exit points on broken trajectories. We propose a new algorithm of estimating sources and sinks from the models of semantic regions. It works well in complex traffic scenes. Although path models, sources, and sinks have strong regularization on the movements of vehicles, they were not well exploited in object counting in the past work. Sources, sinks, scene structures, and velocity priors were used in some tracking algorithms [6], [11], [18].

II. Learning the Models of Scene Structures

To learn the models of scene structures, we first use the unsupervised algorithm proposed in [20] to learn semantic regions. Some examples of semantic regions learned from the Massachusetts Institute of Technology (MIT) traffic data set [20][1] are shown in Fig. 2. In total, there are 29 semantic regions learned. Because of space limitations, only 12 of them are shown. The sources and sinks of semantic regions are estimated in Section II-A. Local semantic regions are connected into global paths in Section II-B.

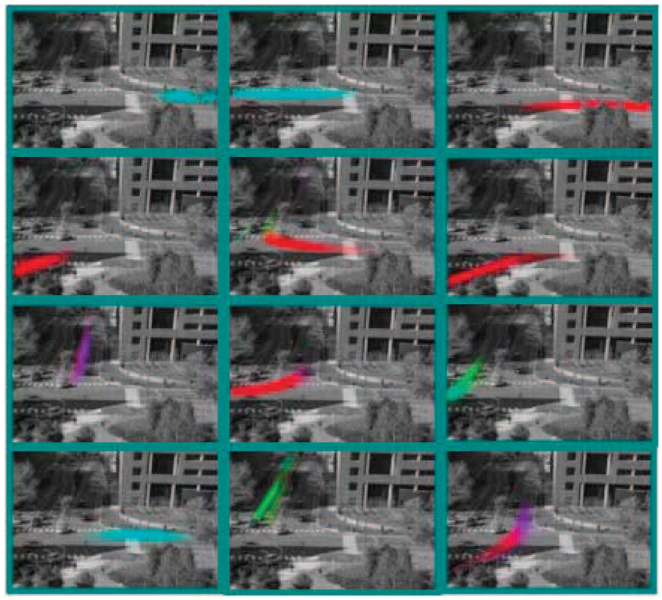
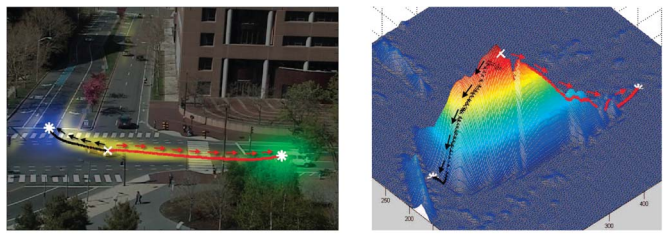


Fig. 2. Examples of the distributions of local semantic regions learned from the MIT traffic data set using the unsupervised algorithm proposed in [20]. Colors indicate directions of local motions: red (→), cyan (←), magenta (↑), and green (↓). Intensity indicates the density of distributions over space.



(a) (b)

Fig. 3. Estimate the source and sink of a semantic region. A starting point (indicated by '×') is selected as the point with the highest density in the semantic region. The source and sink points (indicated by '∗') are found by tracing the starting point. The starting point is driven by the principal local motion direction estimated from the model of the semantic region, and it keeps close to the ridge of the semantic region. (a) Example of a semantic region and its source and sink. The source is indicated by blue, and the sink is indicated by green. The black and red curves are the paths of tracing the source and the sink. (b) 3-D density map of the semantic region in (a). The starting, source, and sink points, and the tracing paths are also shown.

A. Estimation of Sources and Sinks

Let ϕ be a distribution of a semantic region over space and local motion directions. The scene is divided into small cells. ϕ_w^x and ϕ_w^y are the distributions along the x and y motion directions in cell w. ϕ_w^x and ϕ_w^y are learned from long-term observations with the algorithm in [20]. If ϕ_w^x (or ϕ_w^y) is negative, the average motion direction is on the opposite of the x-direction (or y-direction). $\phi_w = |\phi_w^x| + |\phi_w^y|$ is the overall motion density at w. As shown in Fig. 3, a starting point is selected as the point with the maximum ϕ_w in the semantic region. The source and sink points are found by tracing the starting point. During

[1]The MIT traffic data set is downloaded from http://www.ee.cuhk.edu.hk/~xgwang/MITtraffic.html.

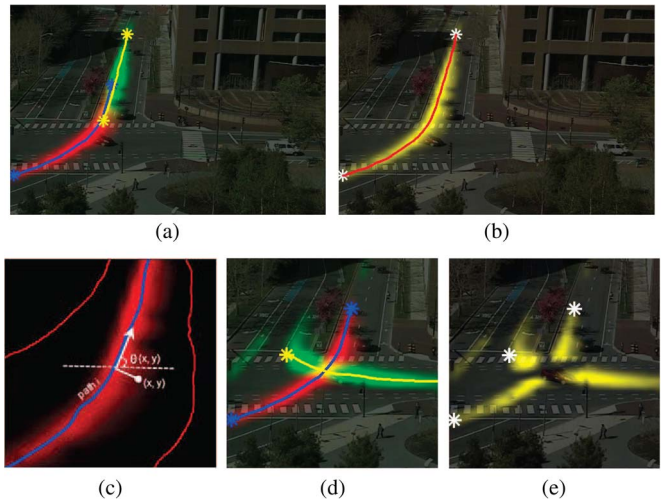


Fig. 4. Connect semantic regions into paths. (a) Two overlapping semantic regions with consistent ridge directions. Their distributions are represented by red and green. Their ridges are represented by blue and yellow curves. (b) Consistent density $\psi(x,y)$ of the two semantic regions in (a). They are connected into a complete path since the defined distance between their source and sink is small. (c) Define $\theta(x,y)$ using the ridge direction. (d) Two crossing semantic regions with different ridge directions. (e) Consistent density $\psi(x,y)$ of two semantic regions in (d). There is a region of low densities between sources and sinks. The defined distance between their source and sink is small, and they are not connected into one path.

the tracing process, the point is driven by the principal local motion direction, and it keeps close to the ridge, where the density is high, of the semantic region. Suppose that the current position of the point is (x,y). To find the sink point, the point moves to $(x+\triangle x, y+\triangle y)$ at the next step according to

$$\begin{aligned} \triangle x &= v_x + \alpha m_x, \quad \triangle y = v_y + \alpha m_y \\ v_x &= \frac{1}{n}\sum_{i=1}^{n}\phi_i^x, \quad v_y = \frac{1}{n}\sum_{i=1}^{n}\phi_i^y \\ m_x &= \frac{\sum_{i=1}^{n}\phi_i(x_i - x)}{\sum_{i=1}^{n}\phi_i}, \quad m_y = \frac{\sum_{i=1}^{n}\phi_i(y_i - y)}{\sum_{i=1}^{n}\phi_i}. \end{aligned} \tag{1}$$

There are n cells in the neighborhood of (x,y). (v_x, v_y) drives the point along the principal object moving direction. (m_x, m_y) keeps it close to the ridge of the semantic region. Without (m_x, m_y), the point may go astray beyond the path boundary before reaching the sink. The source point can be reached by replacing ϕ_i^x and ϕ_i^y with $-\phi_i^x$ and $-\phi_i^y$, respectively.

B. Connecting Semantic Regions

We connect two semantic regions into a path if there is no significant gap between their density distributions and their ridges are along similar directions. Some examples are shown in Fig. 4. The goal is achieved by calculating a defined distance between the source of a semantic region and the sink of the other using the fast marching algorithm [16]. For each semantic region i, we define the gradient of distance at each location (x,y) as $\| \bigtriangledown D(x,y)\| = 1/\phi_i(x,y)$, where $\phi_i(x,y)$ is the density distribution of semantic region i at (x,y). Given distance gradient $\|\nabla D(x,y)\|$, the shortest path connecting any two locations and their shortest distance can be efficiently calculated using the fast marching algorithm. We find the shortest path connecting the source and the sink of a semantic region and define it as the ridge of the semantic region.

For any two semantic regions i and j, a consistent density distribution is defined as follows:

$$\psi_{ij}(x,y) = [\phi_i(x,y) + \phi_j(x,y)]^{\frac{1}{2}}\, e^{\frac{-\left|\theta_i(x,y) - \theta_j(x,y)\right|}{\pi/6}}. \tag{2}$$

For location (x,y), its nearest point (x',y') on the ridge of the semantic region i is found, and let $\theta_i(x',y')$ be the ridge direction at (x',y'). We choose $\theta_i(x,y) = \theta_i(x',y')$ in (2). $|\theta_i(x,y) - \theta_y(x,y)| \in [0,\pi]$. $\psi_{ij}(x,y)$ is low if both semantic regions have low densities at (x,y), which means that there is a gap between the two semantic regions, or their ridge directions are inconsistent, which means that the two semantic regions are along different directions. The gradient of distance at (x,y) is defined as $\|\nabla D(x,y)\| = 1/\psi_{ij}(x,y)$. The distance from the source of region i and the sink of region j is calculated using the fast marching algorithm. The two regions are connected into a path if the distance is small.

III. Clustering Trajectories of Feature Points

Feature points in the scene are detected and tracked using the Kanade–Lucas–Tomasi tracker [19]. Tracking stops if the feature points reach the scene boundaries, if there is large deviation of moving directions, or if there is no matching feature point nearby. Because of occlusions and scene clutters, the trajectories of feature points are highly fragmented. These fragments are call tracklets [6]. To detect individual vehicles, tracklets are clustered by mean shift [4] according to their spatial and temporal proximity. Because of projective distortions, vehicles appear in different sizes at different locations. The average size of vehicles at different locations are estimated and considered in clustering.

$T_i = \{(x_{ik}, y_{ik}, v_{ik}^x, v_{ik}^y, t_{ik})\}_{k=1}^{n_i}$ is a tracklet. (x_{ik}, y_{ik}), (v_{ik}^x, v_{ik}^y), and t_{ik} are the coordinate, velocity, and frame index of point k on T_i, respectively. The distance between T_i and T_j is

$$\begin{aligned} D(T_i,T_j) = \frac{1}{n_{ij}} \sum_{t_{ik_1}=t_{ik_2}} \Big[&(x_{ik_1} - x_{ik_2})^2 + (y_{ik_1} - y_{ik_2})^2 \\ &+ \gamma\left(v_{ik_1}^x - v_{ik_2}^x\right)^2 + \gamma\left(v_{ik_1}^y - v_{ik_2}^y\right)^2 \Big] \end{aligned}$$

where n_{ij} is the number of frames, and T_i and T_j temporally overlap.

Gaussian kernel $K(T_i,T_j) = \exp\{-D(T_i,T_j)/h\}$ is defined. The initial clustering results are obtained by mean shift with the defined kernel. Mean shift is an iterative algorithm. Assuming that $\bar{T}^{(t)}$ is the estimated mean of the cluster at the current step, the mean estimated at the next step is

$$\bar{T}^{(t+1)} = \frac{\sum_{T_i \in \mathcal{N}(\bar{T}^{(t)})} K(T_i, \bar{T}^{(t)}) T_i}{\sum_{T_i \in \mathcal{N}(\bar{T}^{(t)})} K(T_i, \bar{T}^{(t)})}$$

where $\mathcal{N}(\bar{T}^{(t)})$ is the neighborhood of $\bar{T}^{(t)}$. After the mean-shift clustering, the average size $\sigma_{(x,y)}$ of vehicles at each location (x,y) can be estimated as the average of the spatial sizes of clusters that cover (x,y). Given $\sigma_{(x,y)}$, the $D(T_i,T_j)$ can be refined by normalizing the sizes of vehicles as follows:

$$\begin{aligned} D(T_i,T_j) = \frac{1}{n_{ij}} \sum_{t_{ik_1}=t_{ik_2}} \Bigg[&\frac{(x_{ik_1} - x_{ik_2})^2 + (y_{ik_1} - y_{ik_2})^2}{\sigma_{(x,y)}^2/\sigma_0^2} \\ &+ \gamma\left(v_{ik_1}^x - v_{ik_2}^x\right)^2 + \gamma\left(v_{ik_1}^y - v_{ik_2}^y\right)^2 \Bigg] \end{aligned}$$

where $x = (x_{ik_1} + x_{ik_2})/2$, and $y = (y_{ik_1} + y_{ik_2})/2$. Then, mean shift is applied again based on the normalized kernel to get the final clustering results. This normalization helps to cluster objects into proper sizes.

IV. Tracklet Assignment and Association

Let CT be a cluster of tracklets. CT is assigned to one of the vehicle paths considering their spatial overlaps and motion consistency or removed as nonvehicles. For CT, the density maps of motions along the x-direction and y-direction are calculated as

$$M^x(x,y) = \sum_{(x_k,y_k)\in CT} v_k^x e^{-\frac{\|x-x_k\|^2+\|y-y_k\|^2}{2\sigma_1^2}}$$

$$M^y(x,y) = \sum_{(x_k,y_k)\in CT} v_k^y e^{-\frac{\|x-x_k\|^2+\|y-y_k\|^2}{2\sigma_1^2}}.$$

The density maps of CT are obtained by smoothing velocities of points in CT with a Gaussian kernel. ϕ^x and ϕ^y are the density distributions of the x and y motion directions of the path. The consistency between a tracklet cluster and a path is

$$\frac{\sum_{(x,y)} M^x(x,y)\phi^x(x,y)}{\left|\sum_{(x,y)} M^x(x,y)\right| \cdot \left|\sum_{(x,y)} \phi^x(x,y)\right|} \times \frac{\sum_{(x,y)} M^y(x,y)\phi^y(x,y)}{\left|\sum_{(x,y)} M^y(x,y)\right| \cdot \left|\sum_{(x,y)} \phi^y(x,y)\right|}. \quad (3)$$

A cluster of tracklets is assigned to the path with the largest consistency. If the largest consistency is below certain threshold, the cluster of tracklets is removed as a nonvehicle.

After this step, vehicle counting becomes relatively easy since all the remaining clusters of tracklets are on the same path whose sources and sinks are known. The starting/ending points of tracklet clusters of the same vehicle are associated using the Hungarian algorithm [8] by properly defining a transition matrix C, i.e.,

$$C = \left[\begin{array}{ccc|ccc} C_{11} & \cdots & C_{1n} & C_{1(n+1)} & \cdots & -\infty \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ C_{n1} & \cdots & C_{nn} & -\infty & \cdots & C_{n(n+1)} \\ \hline C_{(n+1)1} & \cdots & -\infty & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ -\infty & \cdots & C_{(2n)n} & 0 & \cdots & 0 \end{array}\right]$$

where n is the number of tracklet clusters on the path.

Let $\bar{T}_i$ be the mean of tracklet cluster CT_i. Its starting and ending points are $(x_{is}, y_{is}, v_{is}^x, v_{is}^y, t_{is})$ and $(x_{ie}, y_{ie}, v_{ie}^x, v_{ie}^y, t_{ie})$, respectively. The starting (ending) point of CT_i is either associated with the source (sink) of its path or connected to the ending (starting) point of a unique tracklet cluster on the same path. $C_{i(i+n)} (i \le n)$ defines the likelihood of the ending point of CT_i being associated with the sink as follows:

$$C_{i(i+n)} = \exp\left\{-\frac{(x_{ie}-x_{\text{sink}})^2 + (y_{ie}-x_{\text{sink}})^2}{\sigma_2^2}\right\}$$

where $(x_{\text{sink}}, y_{\text{sink}})$ is the center of the sink, and σ_2 specifies the spatial range of the source.

$C_{(n+i)i} (i \le n)$ defines the likelihood of the starting point of CT_i being associated with the source as follows:

$$C_{(i+n)i} = \exp\left\{-\frac{(x_{is}-x_{\text{source}})^2 + (y_{is}-x_{\text{source}})^2}{\sigma_2^2}\right\}$$

where $(x_{\text{source}}, y_{\text{source}})$ is the center of the source.

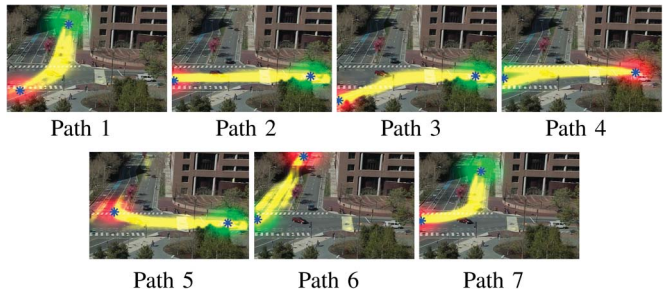


Fig. 5. Seven vehicle paths and their sources and sinks estimated from the 29 semantic regions learned by [20]. The star markers with red colors indicate the sources of paths. The star markers with green color indicate the sinks.

Fig. 6. Sources and sinks estimated from the starting and ending points of trajectories of feature points using the GMM [12]. The estimated sources and sinks are marked by yellow circles.

$C_{ij} (i \le n, j \le n)$ defines the likelihood of connecting the ending point of CT_i with the starting point of CT_j. It considers both motion affinity and temporal affinity as follows:

$$C_{ij} = C_{ij}^{\text{time}} \times C_{ij}^{\text{motion}}$$

where C_{ij}^{time} defines the temporal affinity. Let $\Delta t = t_{js} - t_{ie}$. Thus

$$C_{ij}^{\text{time}} = \alpha^{\Delta t - 1}$$

where $0 < \alpha < 1$. C_{ij}^{time} is lower if the temporal gap between CT_i and CT_j is larger. C_{ij}^{motion} defines the motion affinity, i.e.,

$$C_{ij}^{\text{motion}} = \exp\left\{-\frac{(x_{ie}+v_{ie}^x\Delta t - x_{js})^2 + (y_{ie}+v_{ie}^y\Delta t - y_{js})^2}{2\sigma_3^2}\right\} \times \exp\left\{-\frac{\left(x_{ie}+v_{js}^x\Delta t - x_{js}\right)^2 + \left(y_{ie}+v_{js}^y\Delta t - y_{js}\right)^2}{2\sigma_3^2}\right\}.$$

V. Experiments

We first present the results of connecting semantic regions into complete paths and estimating the sources and sinks of paths. Experiments are done on the MIT traffic data set [20]. As shown in Fig. 5, seven complete paths are estimated from 29 semantic regions learned by [20], and their corresponding sources and sinks are also well located. For comparison, Fig. 6 shows the results of using GMMs to estimate the sources and sinks [12]. This approach is popularly used in video surveillance. Sources are estimated from the starting points of trajectories, and sinks are estimated from the ending points. In Fig. 6, 400 trajectories are used for estimation. Due to occlusion and vehicles stopping and restarting, the trajectories of vehicles are fragmented. Such tracking failures bias the estimation. Some of the sources and sinks estimated by the GMM locate around occluders (street lamps and trees) and stopping lines. Therefore, sources and sinks estimated from semantic regions are more reliable.

To evaluate the performance of object counting, we conduct two experiments and manually label the objects in 3000 frames from the

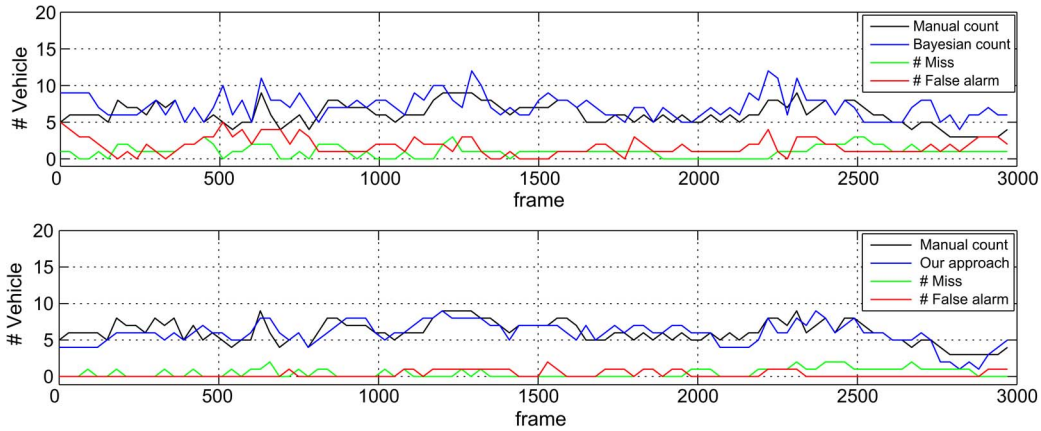


Fig. 7. Results of counting in frame over 100 samples from 3000 frames using the (top) Bayesian method and (bottom) our approach. The green lines show the missed counting mainly because of cluster merging, and the red lines show the false alarm due to cluster splitting and occlusion.

TABLE I
COMPARISON OF COUNTING QUALITIES FOR COUNTING IN FRAME

Method	MSE	ABS	MD	FA
Bayesian	3.68	1.52	15.19%	23.92%
Our approach	**1.43**	**0.93**	**9.81%**	**4.60%**

MIT traffic data set. In the first experiment, we compare with the Bayesian object counting approach proposed in [1]. Both [1] and our approach have some free parameters, and we selected them from a small validation set from the same scene. The study [1] counts the number of objects at every single frame and does not keep the identities of objects over time. We sample a frame every one second (the frame rate is 30 fps) and compare the counting results with the ground truth at every frame. Fig. 7 plots the number of objects counted by [1] and our method, the numbers of objects that they missed, and their false alarms. The count estimated by our approach varies from 1 to 9 at different frames. Table I report the MSE, the averaged absolute counting error, misdetection rate, and the false alarm rate (FA) over the 3000 frames, as described in the following:

$$\text{MSE} = \frac{1}{m}\sum_{i=1}^{m}(\hat{c}_i - c_i)^2$$

$$\text{error} = \frac{1}{m}\sum_{i=1}^{m}|\hat{c}_i - c_i|$$

$$\text{miss} = \frac{1}{m}\sum_{i=1}^{m} c_i^{\text{miss}}/\hat{c}_i \times 100\%$$

$$\text{FA} = \frac{1}{m}\sum_{i=1}^{m} c_i^{\text{fa}}/\hat{c}_i \times 100\%$$

where in the ith frame, $\hat{c}_i$ is the counting result, c_i is the manual counting result, c_i^{miss} is the number of missed counting, and c_i^{fa} is the number of false alarms, all averaged over the M sample frames.

For Bayesian counting, a threshold is used to remove the clusters of feature points in small sizes, which are more likely to be pedestrians. For our method, we add up the number of clusters of tracklets that are active in the current sample frame as the counting result. Our approach has much fewer misdetection occurences and false alarms. Both approaches count objects through clustering feature points. As shown in Fig. 8, in Bayesian counting, multiple vehicles detected in the previous frame may be wrongly merged in the next frame. In addition, feature points of a single vehicle detected as the previous frame may wrongly split into multiple clusters in the next frames. Our approach alleviate such problems to some extent because of the regularization added by path models and average vehicle sizes.

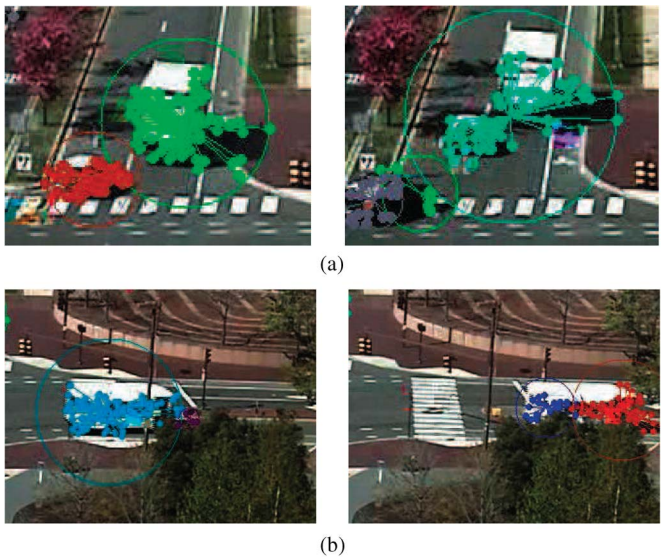


Fig. 8. Example of error merging and splitting in the result using [1]. (a) Two individual vehicles are detected in (left) the previous frame, and they wrongly merge together in (right) the later frame. (b) Individual vehicle is detected in (left) the previous frame, and it wrongly splits into two separate entities due to occlusion (right). Different entities are indicated by different colors.

In the second experiment, we use the whole 3000 frames and our approach to count the accumulated numbers of vehicles along each of the seven paths, respectively. When a vehicle enters the scene at the ith frame and is on the kth path, the number of accumulated count on path k increases by 1 at frame i. Fig. 9 shows that our counting result is close to the ground truth. From the dynamic variations of the accumulated numbers, we can observed different traffic patterns. Path 6 has the largest amount of traffic during this period. In total, there are 44 vehicles entering the scene during this period. Our counting result is 48. Our approach tends to overcount the number of vehicles. Error happens if a vehicle much larger than the normal size enters the scene, i.e., it may be counted as two, since under our model, there is no evidencing showing that the two clusters of tracklets actually belong to the same object. One possible solution is to consider appearance information. If two clusters of tracklets have similar color, they could be merged into one. Fig. 9 also shows the counting results without including path models, associating clusters of tracklets only based on their motion and temporal information. The result becomes much worse. In total, there are 59 objects counted with 15 false alarms.

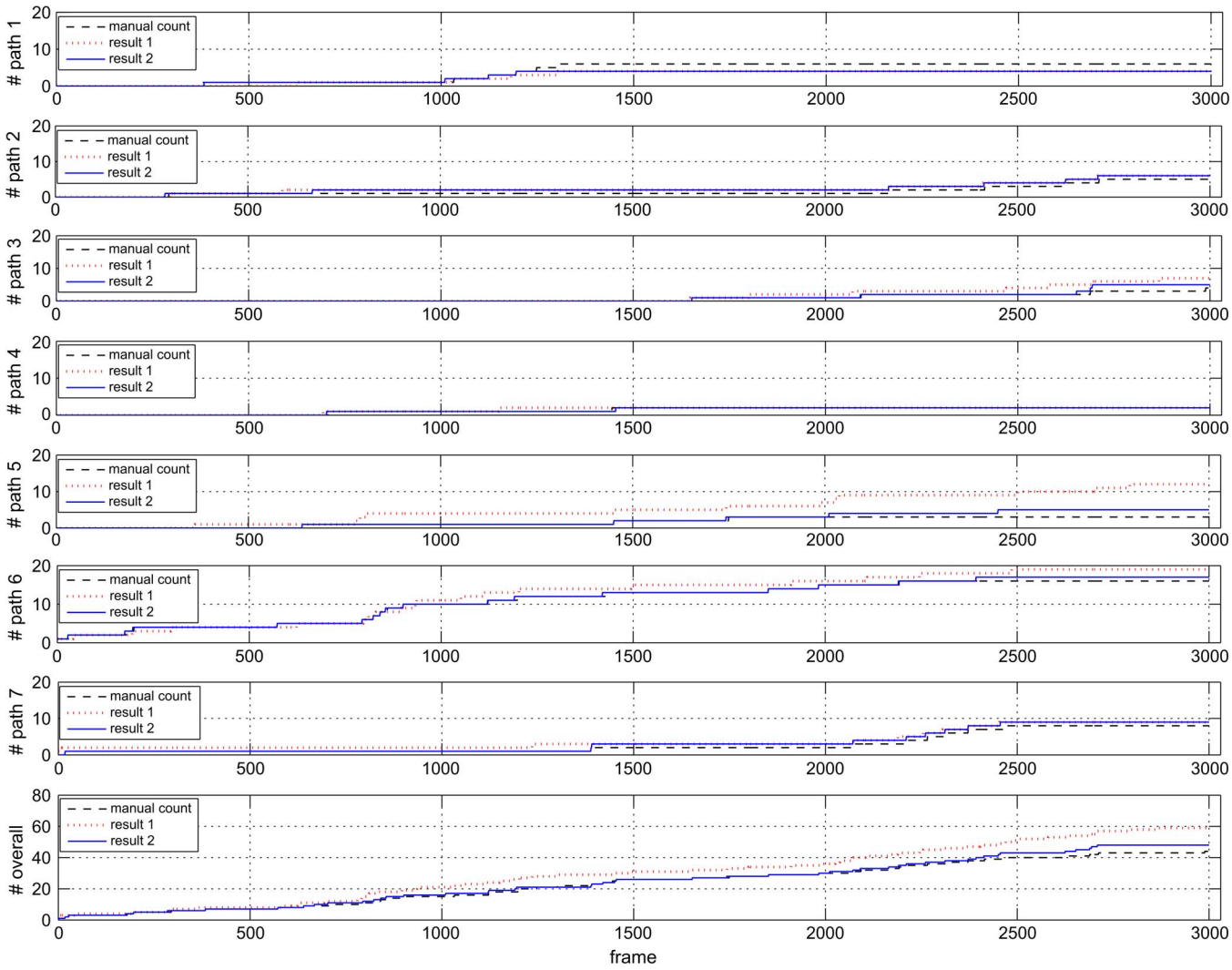


Fig. 9. Results of counting vehicles on paths: Seven figures show cumulative numbers of counted vehicles on seven different paths. The last figure is the overall counting result in the whole scene (i.e., it is the sum of the results of the first seven figures). Manual count: the ground truth. Result1: counting vehicles without using the models of paths and clusters of tracklets are associated only based on the motion and temporal information. Result 2: counting vehicles using the models of paths (our approach).

VI. Conclusion and Discussions

In this paper, we have proposed a new approach to count vehicles in complex traffic scenes by utilizing the information from semantic regions. By counting vehicles on each path separately, it breaks the challenging problem into simpler problems. Semantic regions, which are learned from optical flows, are appropriately connected into complete paths using the proposed fast marching algorithm, and their sources and sinks are well estimated. We propose an algorithm to estimate the average vehicle sizes at different locations along each path that helps to cluster feature points into objects in turn. The Hungarian algorithm is used to associate fragmented trajectories considering the contextual information added by the models of semantic regions, sources, and sinks. Experimental results show the effectiveness of our approach.

Our approach has some limitations and can be improved in several aspects. A semantic region could be detected if pedestrians frequently walk through a zebra crossing. Pedestrian paths and vehicle paths can be distinguished by simple human intervention or the distributions of object sizes and speed along the paths. In extremely crowded scenes, vehicles on adjacent lanes may be very close in space and move side-by-side with the same speed. It poses difficulty on trajectory clustering. This problem can be alleviated to some extent by first excluding trajectories outside a lane before clustering.

References

[1] G. J. Brostow and R. Cipolla, "Unsupervised Bayesian detection of independent motion in crowds," in *Proc. CVPR*, 2006, pp. 594–601.

[2] A. B. Chan, Z. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Proc. CVPR*, 2008, pp. 1–7.

[3] A. B. Chan and Vasconcelos, "Bayesian poisson regression for crowd counting," in *Proc. ICCV*, 2009, pp. 545–551.

[4] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[5] W. Hu, D. Xie, Z. Fu, W. Zeng, and S. Mayband, "Semantic-based surveillance video retrieval," *IEEE Trans. Image Process.*, vol. 16, no. 4, pp. 1168–1181, Apr. 2007.

[6] C. Huang, B. Wu, and R. Nevatia, "Robust object tracking by hierarchical association of detection responses," in *Proc. ECCV*, 2008, pp. 788–801.

[7] D. Huang and T. W. S. Chow, "A people-counting system using a hybrid rbf neural network," *Neural Process. Lett.*, vol. 18, no. 2, pp. 97–113, Oct. 2003.

[8] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Res. Logist. Quart.*, vol. 2, no. 1/2, pp. 83–97, 1995.

[9] J. Li, S. Gong, and T. Xiang, "Scene segmentation for behaviour correlation," in *Proc. ECCV*, 2008, pp. 383–395.

[10] S. Lin, J. Chen, and H. Chao, "Estimation of number of people in crowded scenes using perspective transformation," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 31, no. 6, pp. 645–654, Nov. 2001.

[11] R. Loveland, E. Rosten, and R. Porter, "Improving multiple target tracking in structured environments using velocity priors," in *Proc. SPIE Defense Secur. Symp.*, 2008, vol. 6969, p. 69 690H.

[12] D. Makris and T. Ellis, "Learning semantic scene models from observing activitiy in visual surveillance," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 3, pp. 397–408, Jun. 2005.

[13] B. Morris and M. Trivedi, "A survey of vision-based trajectory learning and analysis for surveillance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 8, pp. 1114–1127, Aug. 2008.

[14] V. Rabaud and S. Belongie, "Counting crowded moving objects," in *Proc. CVPR*, 2006, pp. 705–711.

[15] J. Rittscher, P. H. Tu, and N. Krahnstoever, "Simultaneous estimation of segmentation and shape," in *Proc. CVPR*, 2005, pp. 486–493.

[16] J. Sethian, *Level Set Methods and Fast Marching Methods Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Material Science*. Cambridge, U.K.: Cambridge Univ. Press, 1999.

[17] O. Sidla, Y. Lypetskyy, N. Brandle, and S. Seer, "Pedestrian detection and tracking for counting applications in crowded situation," in *Proc. IEEE Int. Conf. Adv. Video Signal Based Surv.*, 2006, p. 70.

[18] C. Stauffer, "Estimating tracking sources and sinks," in *Proc. IEEE Workshop Event Mining*, 2003, p. 35.

[19] C. Tomasi and T. Kanade, "Detection and tracking of point features," Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CS-91-132, 1991.

[20] X. Wang, X. Ma, and E. Grimson, "Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 539–555, Mar. 2009.

[21] X. Wang, K. Tieu, and E. Grimson, "Learning semantic scene models by trajectory analysis," in *Proc. ECCV*, 2006, pp. 110–123.

[22] T. Zhao and R. Nevatia, "Multiple humans in complex situations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1208–1221, Sep. 2004.