A Moving Vehicle Segmentation Method Based on Clustering of Feature Points for Tracking at Urban Intersection

Yuexian ZOU, He ZHAO, Hang SHI, Yiyan WANG
Advanced Digital Signal Processing Laboratory, Shenzhen Graduate School of Peking University, PRC
zouyx@szpku.edu.cn
zhaoh07395@szcie.pku.edu.cn

***Abstract*—Video-based moving vehicle detection and tracking are important parts of modern intelligent transportation system (ITS). They can provide valuable information such as vehicle velocity and trajectory for ITS. However, vehicle tracking at urban intersection is more challenging than that at highway, due to the complicated scenarios, such as the variety of vehicle moving direction, inter-vehicle clustering and occlusion. Many successful vehicle tracking systems developed for high way vehicle tracking based on the blob-tracking approach failed to provide acceptable performance at urban intersections when there are heavy vehicle occlusion or vehicles close to each other. This paper proposes a novel vehicle segmentation method for moving vehicle segmentation at urban intersection by seeking the spatial-temporal matching of feature points. Experimental results show that feature point may be taken as an important cue for moving vehicle segmentation and tracking under sophisticated traffic situation.**

Keywords- segmentatin; feature points; clustering; intersection

I. INTRODUCTION

As digital cameras and powerful computers are less costly and have become wide spread, video-based traffic surveillance systems have been given more and more attention in the research of intelligent transportation system (ITS). Video-based moving vehicle detection and tracking techniques are used to estimate traffic flows and to provide detailed information of each vehicle, such as velocity and trajectory, for better understanding the operation of transportation networks as well as for road safety.

In the past few decades, various video-based detection and tracking methods and systems have been developed [1-5]. However, most of them focused on highways. A crucial part of transportation networks is at the urban intersection where many collisions and congestions occur. Due to randomness of vehicle direction, clustering and occlusion among various vehicles, tracking under this situation is more challenging than that at highway. In order to handle with those problems, multi-cue approaches often used, such as the color and texture of the vehicles. It is noted that the calculation of the vehicle texture asks for high computational complexity, and the vehicles with the same color cannot be segmented effectively using the conventional object detection methods. Considering the vehicles are rigid objects, and the motion of the feature points of the moving vehicles may provide useful information related to the vehicles in different movement trajectories, we find the method using feature points are a more proper one to solve the aforementioned problems at urban intersection.

This paper proposes a novel vehicle segmentation method for moving vehicle segmentation at urban intersection by seeking the spatial-temporal matching of feature points. Our method involves three major parts: foreground region extraction, feature point extraction, and feature point clustering. A systematic block diagram of the approach is illustrated in Fig 1. First, we extract the complete blob regions of the moving vehicles, upon which the feature points are calculated with neighboring window image of appropriate size as template. Second, repeat the first step to get new blob regions and feature points in the next frame and then perform template matching to associate the feature points with each other within the two consecutive frames. Third, repeat the second step till we get feature point trajectories long enough for clustering so as to segment adjacent vehicles of different motions.

In section II, the foreground region extraction method is introduced. The moving vehicle segmentation method based on the clustering of feature points for tracking is described in section III. Experimental results and conclusion are presented in section IV and V, respectively.

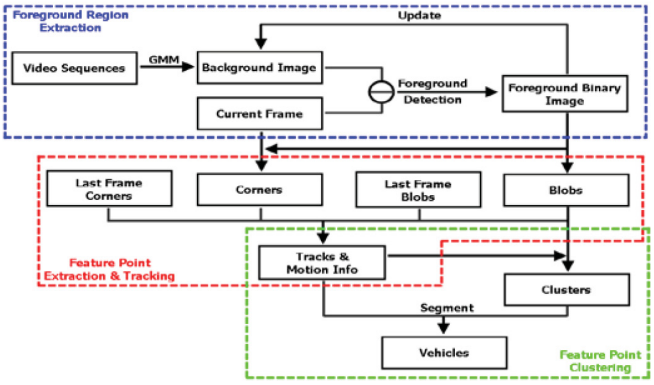

Fig. 1 The diagram of proposed vehicle segmentation method

II. FOREGROUND DETECTION

A. System Environment Setup

In our previous work [6], we carefully analyze and evaluate the intersection traffic characteristics by watching the real intersection traffic flow from the high and the high resolution traffic video (720×576) recorded from the roof of high building beside an urban intersection. It is obvious that different traffic applications require different techniques. For surveillance purpose using camera, a panorama view of the interested intersection detection region covered by one camera (as illustrated in Fig 2), is more helpful for eliminating the vehicle occlusions and increasing the performance of the system. In our approach, the region of interest is set to focus on the center region of intersection, the disturbance of people and other non-interest factors can be further reduced, as well as the computational complexity.

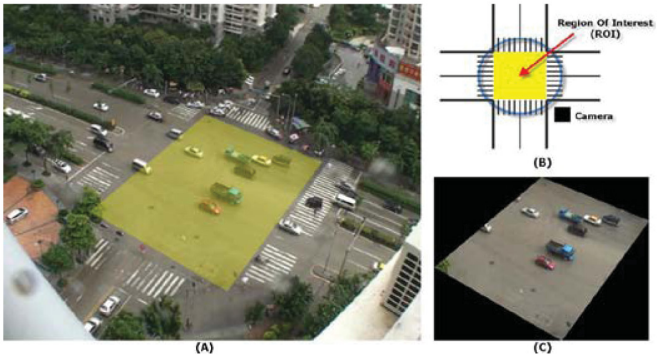

Fig. 2 Illustration of camera installation and ROI (where picture A is the original video image obtained from a camera on a high building, picture C is the ROI)

B. Object Region Extraction

In our approach, the vehicle detection is based on the same background subtraction algorithm [6] developed in our team where GMM is used as the background modeling method. To reduce the computational cost in extracting the moving object regions from subtraction binary image, instead of performing Canny operator, we introduce a new and faster procedure, where a Gaussian pyramid with morphological dilation to remove noises and fill holes and Laplacian pyramid with erosion to restore original size has been adopted. One experimental result is shown in Fig 3 to demonstrate the vehicle extraction performance using aforementioned pyramid algorithm. In Fig 3, row (A) and row (B) shows the object region extraction results for two different frames. More specifically, , column (a) is the original frame images, column (b) is the background images, column (c) is the subtraction binary images, and column (d) is the object region extraction results, respectively. From Fig 3 (B, d), it is clear to see that the expected entire region of moving vehicles have been extracted, where there are no black holes within vehicle blob regions and no noises turn out to be background.

III. MOVING VEHICLE SEGMENTATION

Feature point is the distinguishable point on the object of interest which can be robustly detected in the image. Besides the vehicle image itself, feature points of the vehicle are also a reliable tool for identification. Naturally, the motion of feature points on rigid vehicles can be considered as being in agreement with that of vehicles. Therefore, feature point based moving vehicle segmentation should be a reasonable approach that will offer reliable information for vehicle tracking.

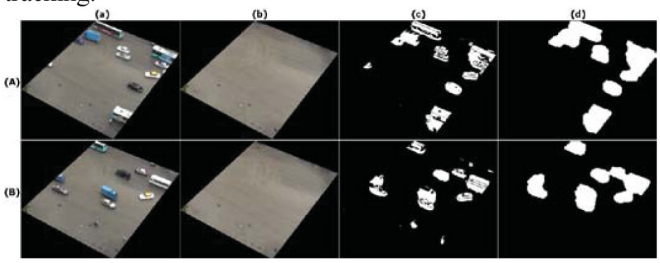

Fig. 3 An example result of object region extraction

A. Feature Point Extraction

In our method, the most commonly used Harris corner detector [7] is adopted upon every extracted foreground region derived from previous section. The Harris feature point detection algorithm can be summarized as follows: 1). Calculate the directional derivatives of the intensity of the foreground region image to obtain the directional derivative image; 2). Calculate the autocorrelation matrix of the directional derivative image over a small window around each point, which is defined as:

$$M_{acorr}(x,y)=\begin{bmatrix} \sum_{-K\le i,j\le K} w_{i,j} I_x^2(x+i,y+j) & \sum_{-K\le i,j\le K} w_{i,j} I_x(x+i,y+j) I_y(x+i,y+j) \\ \sum_{-K\le i,j\le K} w_{i,j} I_x(x+i,y+j) I_y(x+i,y+j) & \sum_{-K\le i,j\le K} w_{i,j} I_y^2(x+i,y+j) \end{bmatrix} \quad (1)$$

where $w_{i,j}$ is a weighting term that is often used to create a Gaussian weighting. Harris corners are allocated in the image associated with two large eigenvalues of the autocorrelation matrix.

To save the computation complexity, feature point extraction is only applied on those blobs we obtained from Section II. It is clear that we are only interested in the detected objects in the scene, so we only need to determine the corners belong to the objects of interest.

B. Feature Point Matching

Once the corners have been detected in the current frame, the next step is to find the corresponding corners in the following frames. Firstly, when we detect a new corner, not only its location, but also the intensity values in its neighbor (7×7 window area) will be stored. . Secondly, an effective algorithm needs to be developed to determine the correlation of the object corners, which means we need to match the corners in the adjacent frames. In our method, the template matching is used. The current frame is taken as the template, the following frame then is taken as the working frame. By calculating the intensity cross-correlation value between the template and the working frame which is defined as:

$$V_{ccorr}(x,y)=\sum_{x',y'}\left[T(x',y')I(x+x',y+y')\right]^2 \quad (2)$$

where *T(x, y)* is the template image, *I(x, y)* is working frame image. In order to reduce the effects of lighting differences between the template and the working frame image, the V_{ccorr} is divided by a normalization coefficient:

$$K(x,y)=\sqrt{\sum_{x',y'}T(x',y')^2\sum_{x',y'}I(x+x',y+y')^2} \quad (3)$$

As the result, we have:

$$V_{ccor\_normed}(x,y)=\frac{V_{ccor}(x,y)}{K(x,y)} \quad (4)$$

From (4), it is clear that the bigger V_{ccor_normed} is, the higher the probability we have finding the match point in the frames.

To reduce the calculation workload, we shall use a large searching window in the beginning. Once a feature point got matched, its motion is clear so that we can search for its matching position at the nearby in the next frame with a smaller window. Obviously, the matching results of a single corner for a series of the frames can be illustrated as a feature point trajectory, which is illustrated in Fig 4, where the 1st column shows two adjacent frames (A) and (C). The image (B) in the second column shows the feature points correspondingly; and the picture (D) in the second column shows the matching result. Here we can easily identify the feature points in different frames which are associated with the same vehicle. Some experimental results of the feature point trajectories are

C. *Feature Point Clustering*

As the feature point matching approach described in the previous section, the vehicle local motion information as well as the position of feature points have been acquired. The motion of feature points represents that of the vehicle they belong to. Different vehicles have different motions. Intuitively inspired by the experimental result shown in Fig 4, the feature point trajectory provides some vehicle moving and feature information, which can be explored to segment clustered or partially occluded vehicles through cluster analysis method. In this paper, we introduce the well-known k-means clustering method [8]. The k-means method aims to minimize the sum of squared distances between all sample points and the cluster center. The procedure consists of the following steps:

1) Choose K initial cluster centers $Z_1(1)$, $Z_2(1)$,..., $Z_k(1)$.
2) At the k-th iterative step, distribute the samples {x} among the K clusters using the relation, $x \in C_j(k)$ if $\| x-Z_j(k) \| < \| x-Z_i(k) \|$, for all i=1, 2,..., K; $i \neq j$; where $C_j(k)$ denotes the set of samples whose cluster center is $Z_j(k)$.
3) Compute the new cluster centers $Z_j(k+1)$, j=1, 2,..., K such that the sum of the squared distances from all points in $C_j(k)$ to the new cluster center is minimized. The measure which minimizes this is the sample mean of $C_j(k)$. Therefore, the new cluster center is given by

$$Z_j(k+1)=\frac{1}{N}\sum_{x\in C_j(k)} x\ ,\ j=1,\ 2,\ldots,\ K \quad (5)$$