Spatio-Temporal Vehicle Tracking Using Unsupervised Learning-Based Segmentation and Object Tracking

Shu-Ching Chen, Mei-Ling Shyu, Srinivas Peeta, Chengcui Zhang

Introduction

Recently, Intelligent Transportation Systems (ITS), which among others make use of advanced sensor systems for on-line surveillance to gather detailed information on traffic conditions, have been identified as the new paradigm to address the growing mobility problems. With the exponential growth in computational capability and information technology, traffic monitoring and large-scale data collection have been enabled through the use of new sensor technologies. One ITS technology, Advanced Traffic Management Systems (ATMS) [1], aims at using advanced sensor systems for on-line surveillance and detailed information gathering on traffic conditions. Robotic vision [2], especially 2D imaging based vision (2D image processing, object tracking, etc.), can be applied to traffic video analysis to address queue detection, vehicle classification, and vehicle counting. In particular, vehicle classification and vehicle tracking have been extensively investigated [3][4]. Issues associated with extracting traffic movement and accident information from real-time video sequences are discussed in [5].

For traffic intersection monitoring, digital cameras are fixed and installed above the area of the intersection. A classic technique to identify the moving objects (vehicles) is background subtraction [6]. Various approaches to background subtraction and modeling techniques have been discussed in the literature [4][5]. In the proposed framework, an unsupervised video segmentation method called the Simultaneous Partition and Class Parameter Estimation (SPCPE) algorithm is applied to identify the vehicle objects in the video sequence [7]. In addition, we propose a new method for background learning and subtraction to enhance the basic SPCPE algorithm in order to generate more accurate segmentation results, so that more accurate spatio-temporal relationships of objects can be obtained. Experiments are conducted using real-life traffic video sequences from road intersections. The experimental results indicate that almost all moving vehicle objects can be successfully identified at a very early stage of the processing, thereby ensuring that accurate spatio-temporal information of objects can be obtained through object tracking.

Learning-Based Vehicle Object Segmentation and Tracking for Traffic Video Sequences

Traffic video analysis at intersections can provide a rich array of useful information such as vehicle identification, queue detection, vehicle classification, traffic volume, and incident detection. The proposed unsupervised spatio-temporal vehicle tracking framework includes background learning and subtraction, vehicle object identification and tracking.

Background subtraction is a technique to remove non-moving components from a video sequence, where a reference frame of the stationary components in the image is created. Once created, the reference frame is subtracted from any subsequent images. The pixels resulting from new (moving) objects will generate a non-zero difference. The main assumption for its application is that the camera remains stationary. However, in most cases, the non-semantic content (or background) in the images or video frames is very complex. Therefore, an effective way to obtain background information can enable better segmentation results.

In the proposed framework, an adaptive background learning method is proposed. Our method consists of the following steps as illustrated in Fig. 1:

1. Subtract the successive frames to get the motion difference images.
2. Apply segmentation on the difference images to get the estimation of foreground regions and background regions.
3. Generate the current background image based on the learned information seen so far.
4. Perform background subtraction and object segmentation for those frames that contribute to the generation of the current background. Meanwhile, the extracted vehicle objects are tracked from frame to frame. Upon finishing the current processing, go back to Steps 1-3 to generate the next background image. Repeat this process until all the frames have been segmented.

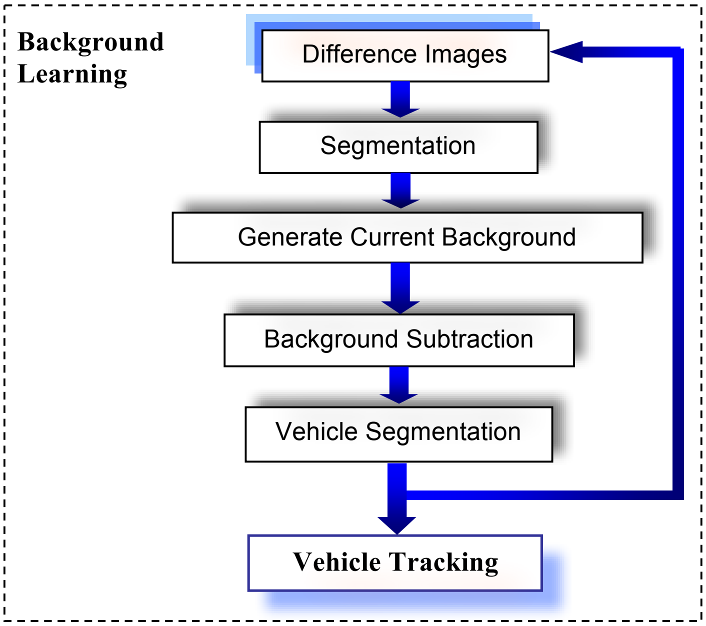


Figure 1. The basic workflow of the proposed method.

The proposed segmentation method can identify vehicle objects, but does not differentiate between them (into cars, buses, etc.). Therefore, *a priori* knowledge (size, length, etc.) of different vehicle classes should be provided to enable such classification. In addition, since the vehicle objects of interest are the moving ones, stopped vehicles will be considered as static objects and will not be identified as mobile objects until they start moving again. However, the object tracking technique ensures that such vehicles are seamlessly tracked though they "disappear" for some duration due to the background subtraction. This aspect is especially critical under congested or queued traffic conditions.

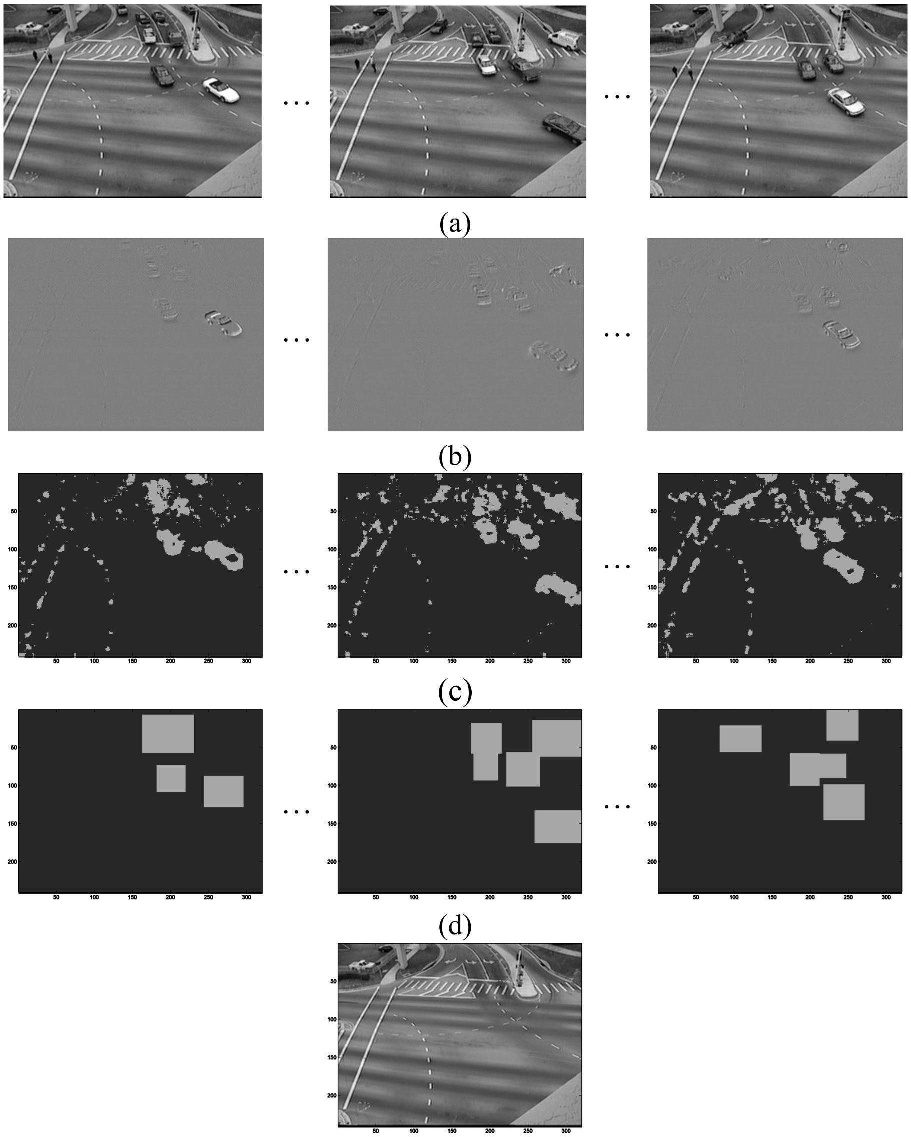
In a traffic video monitoring sequence, when a vehicle object stops in the intersection area (including the approaches to the intersection), our framework may deem it as part of the background information. In this case, since the vehicle objects move into the intersection area before stopping, they are identified as moving vehicles before they stop due to the characteristics of our framework. Hence, their centroids identified before they stop will be in the intersection area. For these vehicles, the tracking process is frozen until they start moving again; they are identified as "waiting" rather than "disappearing" objects. That is, the tracking process will follow the same procedure as before unless one or more new objects abruptly appear in the intersection area. Then, the matching and tracking of the previous "waiting" objects will be triggered to continue tracking the trails of these vehicles.

The Unsupervised Video Segmentation Method (SPCPE)

The SPCPE (Simultaneous Partition and Class Parameter Estimation) algorithm is an unsupervised image segmentation method to partition video frames [7]. A given class description determines a partition, and vice versa. Hence, the partition and the class parameter have to be estimated simultaneously. In practice, the class descriptions and their parameters are not readily available. An additional difficulty arises when images have to be partitioned automatically without the intervention of the user: we do not know *a priori* which pixels belong to which class. In the SPCPE algorithm, the partition and the class parameters are treated as random variables. The method for partitioning a video frame starts with an arbitrary partition and employs an iterative algorithm to estimate the partition and the class parameters jointly. Since the successive frames in a video do not differ much, the partitions of adjacent frames do not differ significantly. Each frame is partitioned by using the partition of the previous frame as an initial condition to speed up the convergence rate of the algorithm. Usually, the number of iterations needed for convergence is around 2~3 except for the first frame for which a randomly generated initial partition is used.

Background Learning and Extraction

The basic idea of our background learning method is to generate the current background image based on the segmentation results extracted from a set of frame-to-frame difference images. The method described in [8] is probably closest to ours. In [8], binary thresholding on difference image is used as the segmentation method, and the background image is updated periodically based on the weighted sum of the current binary object mask and the previous background. They also conduct vehicle tracking and vehicle classification based on that. In our method, instead of using binary thresholding, we use the SPCPE algorithm to segment a difference image into a two-class segmentation map which can serve as a foreground/background mask, where class 1 includes the background points and class 2 records the foreground points. By just collecting a small portion of the continuous segmentation maps, we can reach a point where every pixel position has at least one segmentation map with its corresponding pixel value equal to 1 (background pixel). In other words, every background point within this time interval has appeared and been identified at least once in the segmentation maps.


(a)

(b)

(c)

(d)

(e)

Figure 2. Unsupervised background learning and subtraction in the traffic video sequence. (a) Image sequence from frame 1121 to frame 1228; (b) Successive difference (normalized) images for frames in (a); (c) Segmentation maps for difference images in (b); (d) Rectified segmentation maps for difference images in (c); (e) The generated background for this sequence.

Fig. 2 illustrates the process for background learning using an image sequence from frame 1121 to frame 1228. As shown in Fig. 2(b), the difference images are computed by subtracting successive frames and applying linear normalization. Then, these difference images are segmented into 2-class segmentation maps (Fig. 2(c)) using SPCPE, followed by a rectification procedure that can eliminate most of the noise and store the background information robustly. Fig. 2(d) shows such rectified segmentation maps for Fig. 2(c). The rectified segmentation maps are then used to generate a background image if the specified condition is satisfied. The step to extract the background information is done by taking the corresponding background pixels from individual frames within this time interval. Instead of simply averaging these background pixel values, we further analyzed its histogram distribution and picked the values in dominant bin(s) as the trusted values. With this extra sophistication, the false positives in background image due to noise or miss-detected motions can be reduced significantly. Two such examples are shown in Fig. 3 where the left column contains two constructed background images with lots of false positives, while the right column shows the improved results based on the principles described above. As can be also seen in this figure, the two generated background images contain some static vehicle objects. Such an example includes the gray car waiting in the middle of the intersection, and those cars that stopped behind the zebra crossing. Once they start to move, a new background image is created to reflect the motion changes in them.

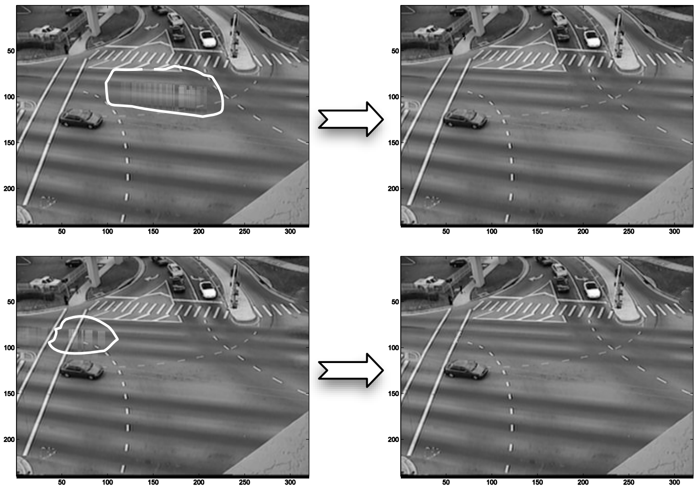


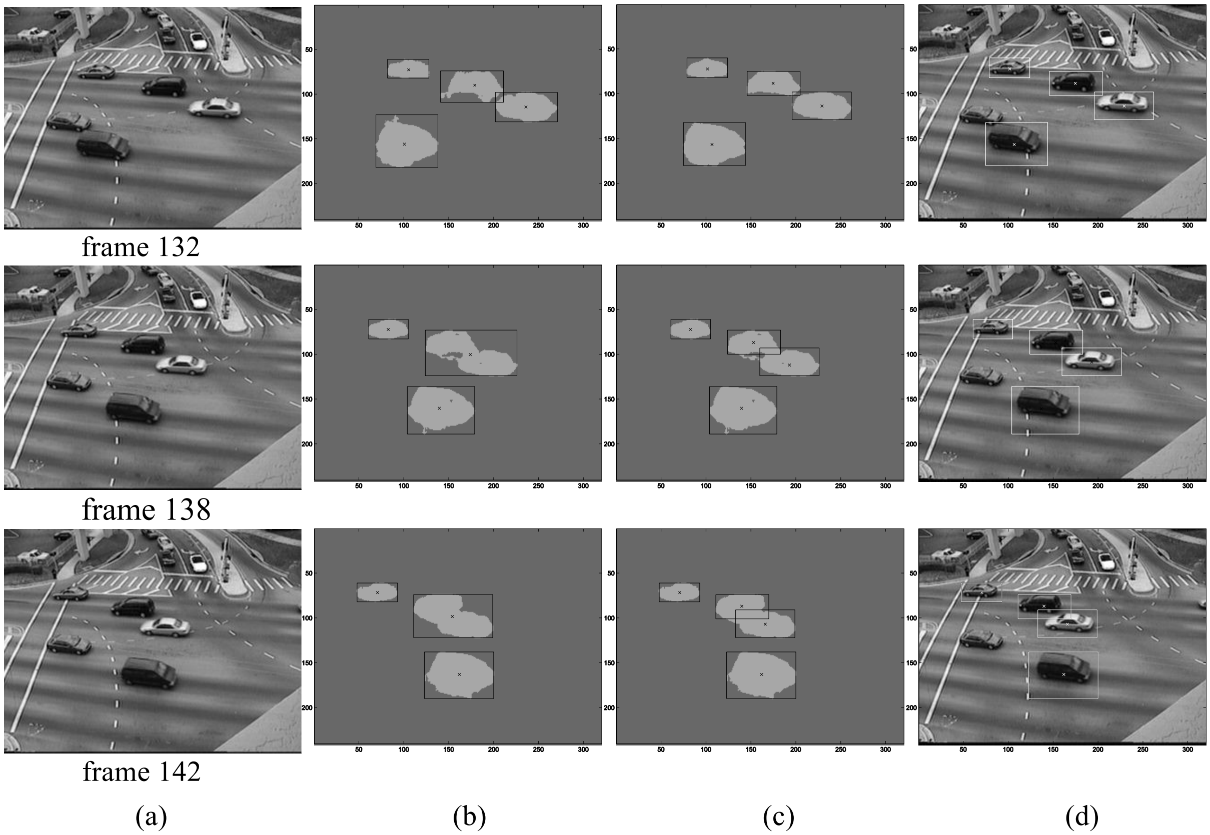
Figure 3. Improved background extraction. Left column are two generated background images with false positives marked by white circles. Right column shows the improved results.

The key point here is that it is not necessary to obtain a perfectly 'clean' background image for each time interval. In fact, including the static objects as part of the background will not affect the extraction of the moving objects. Further, once the static objects begin to move, the underlying background can be discovered automatically. Instead of finding one 'general background image', the proposed background learning method aims to provide periodical background images based on motion difference and robust image segmentation. In this manner, it is insensitive to illumination changes, and does not require any human efforts in the loop.

Vehicle Object Tracking

In order to index the vehicle objects, the proposed framework must have the ability to track the moving vehicle objects (segments) within successive video frames, which enables it to provide useful and accurate traffic information for ATMS. After video segmentation, the segments (objects) with their bounding boxes and centroids are extracted from each frame. Intuitively, two segments that are spatially the closest in the adjacent frames are connected. Euclidean distance is used to measure the distance between their centroids for vehicle tracking. However, it is still necessary to handle the occlusion situations in vehicle tracking.

A more sophisticated object tracking algorithm integrated in the proposed framework is given in [9]. It can handle the situation of two objects overlapping under certain assumptions (e.g., the two overlapped objects should have similar sizes). In this case, if two overlapped objects with similar sizes have ever separated from each other in the video sequence, they can be split and identified as two objects with their bounding boxes being fully recovered using the object tracking algorithm. The results are demonstrated in Fig. 4(a)-(d), where two vehicles have some overlapping in frames 138 and 142, but are identified as two separate objects in frame 132. Fig. 4(d) demonstrates the final results by applying the occlusion handling method proposed in [9]. The segmentation results accurately identify all the vehicles objects' bounding boxes and centroids.

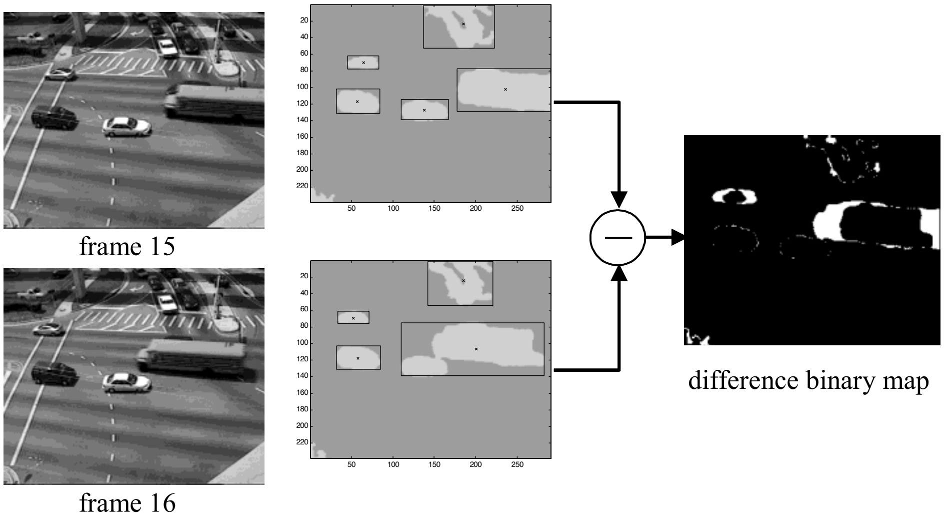

frame 132

frame 138

frame 142

(a) (b) (c) (d)

Figure 4. Handling two object occlusion in object tracking. (a) Video frames 132, 138, and 142; (b) Segmentation maps for frames in (a) without occlusion handling; (c) Results by applying occlusion handling; (d) The final results by overlaying the bounding boxes in (c) to frames in (a).


frame 15

frame 16

difference binary map

Figure 5. Handling object occlusion in object tracking.

However, there are cases where a large object overlaps with a small one. For example, as shown in Fig. 5, the large bus merges with the small white car to form a new big segment in frame 16 though they are two separate segments in frame 15. In this scenario, the car object and the bus object that were separate in frame 15 cannot find their corresponding segments in frame 16 by centroid-matching and size restriction. However, from the new big segment in frame 16, we can reason that this is an "*overlapping*" segment that actually includes more than one vehicle object. For this purpose, a difference binary map reasoning method is proposed in this article to identify which objects the "*overlapping*" segment may include. The idea is to obtain the difference binary map by subtracting the segment result of frame 15 from that of frame 16 and check the amount of difference between the segmentation results of the consecutive frames. As shown in the difference binary map in Fig. 5, the white areas in it indicate the amount of difference between the segmentation results of the two consecutive frames. Thereby, the car and bus objects in frame 15 can be roughly mapped into the area of the big segment in frame 16 with relatively small differences. Hence, the vehicle objects in the big segment in frame 16 can be obtained by reasoning that this segment is most probably related to the car and bus objects in frame 15. Therefore, for the big segment (the "*overlapping*" segment) in frame 16, the corresponding links to the car and bus objects in frame 15 can be created, which means that the relative motion vectors of that big segment in the following frames will be automatically appended to the trace tubes of the bus and car objects in frame 15.

Experimental Analysis

Two real life traffic video sequences (one taken by us and the other downloaded from the website of KOGS/IAKS Universitat Karlsruhe [10]) are used to analyze spatio-temporal vehicle tracking using the proposed learning-based vehicle tracking framework. We label these two video sequences as video sequence #1 and video sequence #2, respectively. Both of them are grayscale videos that show the traffic flows on two different road intersections for some time duration. The background information can be very complex due to road pavement, trees, zebra crossing, pavement markings/signage, and ground. The proposed new framework is fully unsupervised in that it can enable the automatic background learning process that greatly facilitates the unsupervised vehicle segmentation process without any human intervention. During the segmentation, the first frame is partitioned with two classes using random initial partitions. After obtaining the partition of the first frame, the partitions of the subsequent frames are computed using the previous partitions as the initial partitions since there is no significant difference between consecutive frames. By doing so, the segmentation process will converge fast, thereby providing support for real-time processing.

To demonstrate the effectiveness of the proposed background learning process, the algorithm is coded in C++ and run on a 3.06 GHz Pentium 4 personal computer with 1 GB RAM. The total CPU time for processing 2,940 JPEG frames sized 240 by 320 pixels is about 1,639 seconds, i.e., less than 0.56 seconds per frame. This performance ensures that a long run is not necessary to fully determine accurate background information. In our experiments, the current background information can be usually obtained within 20~100 consecutive frames and is good enough for the future segmentation process. In fact, by combining the background learning process with the unsupervised segmentation method, our framework can enable the adaptive learning of background information.

Fig. 6 shows the segmentation results for a few frames (811 and 863) along with the original frames, background images, and the difference images. As shown in Fig. 6(a), the background image for frame 811 contains some static vehicle objects such as the gray car waiting in the middle of the intersection, and those cars that stopped behind the zebra crossing. Since they are identified as part of the background in this time interval as they lack motion, they will not be extracted by the segmentation process as shown in Fig. 6(a). However, the gray car (marked by white arrow in original frames) that was previously waiting in the middle begins to move around frame 860, triggering the generation of a new current background, shown in Fig. 6(b) labeled as #2. From Fig. 6(b), it is obvious that the gray car is fading, although not completely, from the background image. However, this fading is sufficient to result in the identification of the gray car in frame 863, as can be seen from the segmentation map and final result in Fig. 6(b). Moreover, as shown in frame 863 in Fig. 6(b), our method can successfully illustrate the slow motion effect, unlike many methods that have difficulties in dealing with it since it can be easily confused with noise-level information.

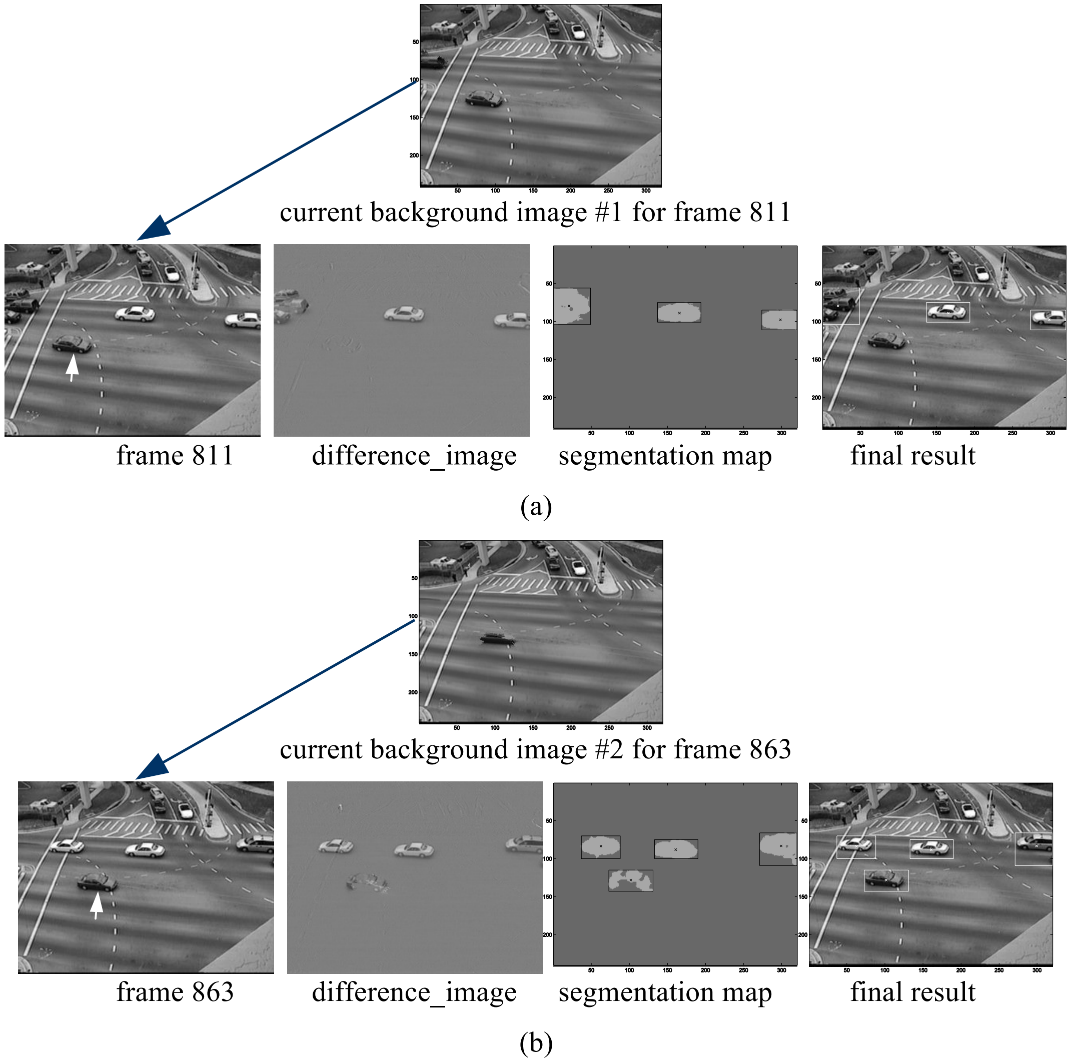
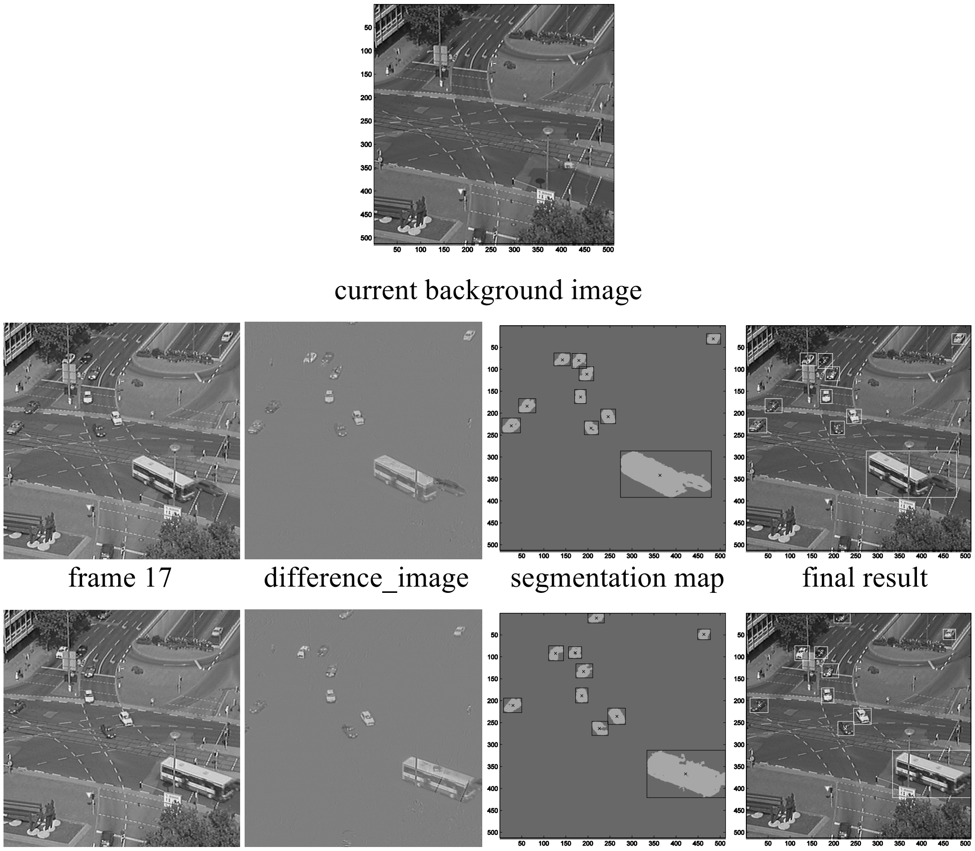


Figure 6. Segmentation results for video sequence #1. (a) Segmentation result for frame 811 using the background image #1; (b) Segmentation result for frame 863 using the background image #2.

Fig. 7 shows our experimental results for video sequence #2. As illustrated by the figure, the background of this traffic video sequence is very complex. Some vehicle objects (for example, the small gray vehicles in the upper left part of the frame 33) can be easily ignored or confused with the road surface and surrounding environment. While there is an existing body of literature that addresses relatively simple backgrounds, our framework can address far more complex situations, as illustrated hereafter. The experimental results shown in Fig. 7 are very promising in that: first, the background is perfectly reconstructed by using 38 out of the total 50 consecutive frames; and second, a single class can capture almost all vehicle objects, even those vehicles that look small and obscure in the upper left area of the video frames. Also, the rightmost column in Fig. 7 shows that almost all vehicle objects are identified as separate objects.



current background image

frame 17 difference_image segmentation map final result

frame 33 difference_image segmentation map final result

Figure 7. Segmentation results for video sequence #2.

Based on our experimental experience on these two traffic video sequences, we have the following observations:

1. The segmentation method adopted is very robust and insensitive to illumination changes. Also, the underlying class model in the SPCPE method is very suitable for vehicle object modeling. Unlike some existing frameworks, in which one vehicle object is segmented into several small regions and later re-grouped and linked with the corresponding vehicle object, no extra effort is needed for such a merge in our method.
2. As described earlier, the long-run look ahead is not necessary to generate a background image in our framework. This implies that the moving objects can be extracted as soon as their motions begin to show up. Moreover, the background image can be quickly updated and adapted to new changes in the environment.
3. No manual initialization or prior knowledge of the background is needed.

Further, since the position of the centroid of a vehicle is recorded during the segmentation and tracking process, this information can be used in the future for indexing its relative spatial relations. The proposed framework has the potential to address a large range of spatio-temporal related database queries for ITS. For example, it can be used to reconstruct accidents at intersections in an automated manner to identify causal factors to enhance safety.

Conclusions

In this article, we present a framework for spatio-temporal vehicle tracking using unsupervised learning-based segmentation and object tracking. An adaptive background learning and subtraction method is proposed and applied to two real life traffic video sequences to obtain more accurate spatio-temporal information of the vehicle objects. The proposed background learning method paired with the image segmentation is robust under many situations. As demonstrated in our experiments, almost all vehicle objects are successfully identified through this framework. A key advantage of the proposed background learning algorithm is that it is fully automatic and unsupervised, and performs the generation of background images using a self-triggered mechanism. This is very useful in video sequences in which it is difficult to acquire a clean image of the background. Hence, the proposed framework can deal with very complex situations vis-à-vis intersection monitoring.

In our future work, we intend to: (i) perform a more comprehensive study under a wider range of conditions; (ii) index and store the vehicle tracking information; and (iii) fuse different types of media data from video data.

Acknowledgements

For Shu-Ching Chen, this research was supported in part by NSF CDA-9711582, NSF EIA-0220562, and the office of the Provost/FIU Foundation.

Keywords: Robotic vision, vehicle tracking, video analysis, segmentation, ITS.

References

[1] S. Peeta and H.S. Mahmassani, "Multiple user classes real-time traffic assignment for online operations: A Rolling Horizon Solution Framework," *Transportation Research,* vol. 3, no. 2, pp. 83-98, 1995.

[2] A.C. Kak and G.N. DeSouza, "Robotic vision: What happened to the visions of yesterday?" invited paper in *Proceedings of the 2002 Int. Conference in Pattern Recognition*, Quebec, Canada, Aug. 2002.

[3] S. Kamijo, Y. Matsushita, and K. Ikeuchi, "Traffic monitoring and accident detection at intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 2, 2000.

[4] C. Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999, pp. 246-252.

[5] D.J. Dailey, F. Cathey, and S. Pumrin, "An algorithm to estimate mean traffic speed using uncalibrated cameras," *IEEE Transactions on Intelligent Transportations Systems*, vol. 1, no. 2, pp. 98-107, June 2000.

[6] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Reading, Mass: Addison-Wesley, 1993.

[7] S.-C. Chen, S. Sista, M.-L. Shyu, and R.L. Kashyap, "An indexing and searching structure for multimedia database systems," in *IS&T/SPIE conference on Storage and Retrieval for Media Databases 2000*, San Jose, CA, USA, January 23-28, 2000, pp. 262-270.

[8] S. Gupte, O. Masoud, R.F.K. Martin, and N.P. Papanikolopoulos, "Detection and classification of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 37-47, March 2002.

[9] S.-C. Chen, M.-L. Shyu, C. Zhang, and R.L. Kashyap, "Object tracking and augmented transition network for video indexing and modeling," in *12th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2000)*, Vancouver, British Columbia, Canada, November 13-15, 2000, pp. 428-435.

[10] http://i21www.ira.uka.de/image_sequences/.