2006 IEEE International Conference on
Systems, Man, and Cybernetics
October 8-11, 2006, Taipei, Taiwan

Multi-objects Tracking System Using Adaptive Background Reconstruction Technique and Its Application to Traffic Parameters Extraction

Chin-Teng Lin, *Fellow, IEEE,* Yu-Chen Huang, Ting-Wei Mei, Her-Chang Pu, and Chao-Ting Hong

***Abstract*—In this paper, we present a real-time multi-objects tracking system which can detect various types of moving objects in image sequences of traffic video obtained from a stationary video camera. Using the adaptive background reconstruction technique can effectively handle with environmental changes and obtain good results of objects extraction. Besides, we introduce a robust region- and feature-based tracking algorithm with plentiful features to track correct objects continuously. After tracking objects successfully, we can analyze the tracked objects' properties and recognize their behavior for extracting some useful traffic parameters. According to the structure of our proposed algorithms, we implemented a tracking system including the functions of objects classification and accident prediction. Experiments were conducted on real-life traffic video of some intersection and testing datasets of other surveillance research. The results proved the algorithms we proposed achieved robust segmentation of moving objects and successful tracking with objects occlusion or splitting events. The implemented system also extracted useful traffic parameters.**

I. INTRODUCTION

A tracking system is composed of three main modules: objects extraction, objects tracking and behavior analysis. Foreground segmentation is the first step of objects extraction and it's to detect regions corresponding to moving objects such as vehicles and pedestrians. The modules of objects tracking and behavior analysis only need to focus on those regions of moving objects. There are three conventional approaches for foreground segmentation outlined in the following: optical-flow, temporal differencing and background subtraction. Besides those basic methods, there are others or combined methods for foreground segmentation. It's a key process to recover and update background images from a continuous image sequences automatically. Friedman et al. [1] used a mixture of three Gaussians for each pixel to represent the foreground, background, and shadows with an incremental version of EM (expectation maximization) method. Li et al. [2] proposed a Bayesian framework that incorporated spectral, spatial, and temporal features to characterize the background appearance.

This work was supported by the Ministry of Economic Affairs, Taiwan R.O.C., under Grant 95-EC-17-A-02-S1-032.

C.-T. Lin is with the Department of Electrical and Control Engineering/Department of Computer Science, National Chiao-Tung University, Hsin-Chu, Taiwan, R.O.C., and also with the Brain Research Center, University System of Taiwan, Taiwan, R.O.C. (phone: +886-3-5712121 ext 31228; fax: +886-3-5729880; e-mail: ctlin@mail.nctu.edu.tw).

Y.-C. Huang, T.-W. Mei, and C.-T. Hong are with the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsin-Chu, Taiwan, R.O.C.(e-mail: yette@ms15.hinet.net ; twmei.ece93g@nctu.edu.tw).

H.-C. Pu is with with the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsin-Chu, Taiwan, R.O.C.(e-mail hcpu.ece87g@nctu.edu.tw).

Besides foreground segmentation, objects tracking is the another key module of surveillance systems. Hu et al. [3] presented there are 4 major categories of tracking algorithms: region-based tracking algorithms [4], active-contour-based tracking algorithms, feature-based tracking algorithms [5], and model-based tracking algorithms [6]. McKenna et al. [7] proposed a tracking algorithm at three levels of abstraction: regions, people, and groups in indoor and outdoor environments. Cucchiara et al. [8] presented a multilevel tracking scheme for monitoring traffic. Veeraraghavan et al. [9] used a multilevel tracking approach with Kalman filter for tracking vehicles and pedestrians at intersections.

Understanding objects' behavior and extracting useful traffic parameters are the main work after successfully tracking the moving objects from the image sequences. Jung et al. [10] proposed a traffic flow extraction method with the velocity and trajectory of the moving vehicles. They estimated the traffic parameters, such as the vehicle count and the average speed and extracted the traffic flows. Remagnino et al. [11] presented an event-based visual surveillance system for monitoring vehicles and pedestrian. Trivedi et al. [12] described a novel architecture for developing distributed video networks for incident detection and management. The networks utilized both rectilinear and omni-directional cameras. Kamijo et al. [13] developed a method by the results of tracking for accident detection which can be generally adapted to intersections. Hu et al. [14] proposed a probabilistic model for predicting traffic accidents using 3-D model-based vehicle tracking.

II. SYSTEM OVERVIEW

A. System Flowchart

At first, foreground segmentation module directly uses the raw data of surveillance video as inputs. This sub-system also updates background image and applies segmenting algorithm to extract the foreground image. Next, the foreground image will extract individual objects. At the same time, object-based features are also extracted from the image with extracted objects. Main work of the third sub-system is to track objects. The tracking algorithm will use significant object features and input them into analyzing process to find the optimal matching between previous objects and current objects. After moving objects are tracked successfully in this sub-system, the consistent labels are assigned to the correct objects. Finally, objects behavior is analyzed and recognized. Useful traffic parameters are extracted and shown in the user interface. The diagram of global system is shown in Fig. 1.

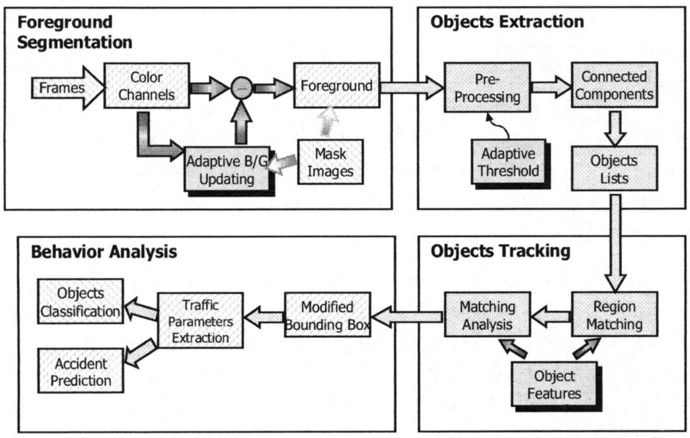


Fig. 1. Diagram of tracking system

1) Foreground Segmentation: At first, we input the raw data of surveillance video obtained from stationary video cameras to this module. And, the main processes of this sub-system are foreground segmentation and background reconstruction. The video camera is fixed and the background can be regarded as a stationary image, so the background subtraction method is the simplest way to segment moving objects. Besides, the results of frame differencing and previous objects condition are also used in for achieving the segmentation more reliably.

2) Background Initialization: Before segmenting the foreground from the image sequences, the system needs to construct an initial background image for further process. The basic idea of finding the background pixel is the high appearing probability of background. During a continuous duration of surveillance video, the level of each pixel appeared most frequently is almost its background level. We use a simpler method that if a pixel's value is within a criterion for several consecutive frames, it means the probability of appearing of this value is high locally or this value is locally stable. This method can build an initial background image even though there are moving objects in the view of camera at the duration of initialization.

3) Adaptive Background Updating: We introduce an adaptive threshold for foreground segmentation. The adaptive threshold includes two parts: one is a basic value and the other is adaptive value. And, we use the equation shown in (1) to produce the threshold.

$$Th_{FG} = Value_{basic} + 1.5 * Peak_{local} + STDEV_{local} \quad (1)$$

The two statistic data ($Peak_{local}$, $STDEV_{local}$) are calculated in the specific scope as shown in Fig. 2. This adaptive threshold will assist the background updating algorithm in coping with environmental changes and noise effects.

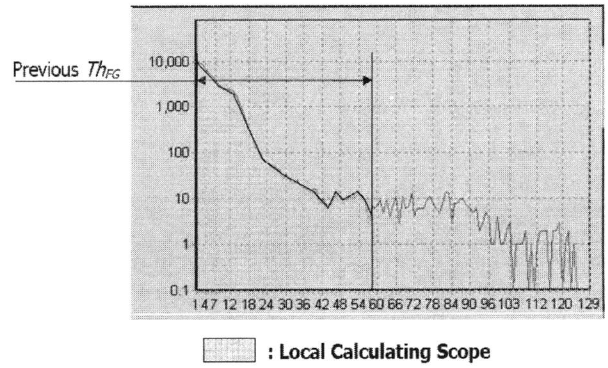


Fig. 2. Histogram of background subtraction image the caption.

At outdoor environment, there are some situations that result in wrong segmenting easily. Those include waving of tree leaves, light gradual variation and etc. Even there are sudden light changes happened when the clouds cover the sun for a while or the sun is revealed from clouds. We propose an adaptive background updating framework to cope with those unfavorable situations. Firstly, we introduce a statistic index which is calculated by the equation (2).

$$Index = Means_{local} + 3 * STDEV_{local} \quad (2)$$

According to this index, we adjust the frequency of updating the current background image adaptively and the updating frequency is defined as several phases. The background updating speed will increase or decrease with the updating frequency. These phases and their relevant parameters are listed in Table 1.

TABLE I
PHASES OF ADAPTIVE BACKGROUND UPDATING

Phase	Condition	Sampling rate	*Freq*
Normal	Index < 12	1/30	30
Middle I	12 ≤ Index < 18	1/24	24
Middle II	18 ≤ Index < 24	1/16	16
Heavy I	24 ≤ Index < 30	1/8	8
Heavy II	30 ≤ Index < 36	1/4	4
Extra Heavy	36 ≤ Index	Directly update	N/A

B. Objects Extraction

In this sub-system, we'll use the connected components algorithm to extract each object and assign it a specific label to let the system recognize different objects. Before the process of connected components algorithm, we will apply morphological operation to improve the robustness of object extraction. The result of this algorithm is the labeled objects image. Then we will build a current objects list with their basic features such as position, size and color according to the labeled objects image. We use a spatial filter to remove ghost objects or objects at the boundaries.

C. Objects Tracking

This sub-system is the main process of entire system, because it deals with objects tracking function. Inputs of this module are three lists: current objects list, previous objects list and overlap relation list. This sub-system can analyze the relation between current objects and previous objects and obtain other properties of objects, such as velocity, life, trajectory and etc. The tracking process is shown in Fig. 3.

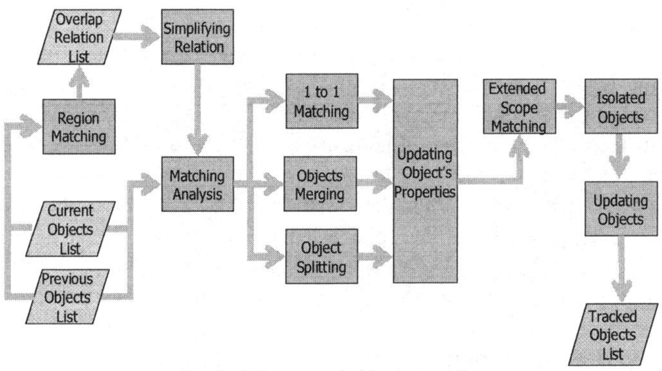


Fig. 3. Diagram of objects tracking

1) Region Matching: We use the size of bounding box and central position as input data and a simple method to calculate the size of overlap area. Then we calculate the ratio of the overlap area to the minimum area of two objects by (3).

$$Ratio_{overlap} = Area_{overlap} / Min(Area_{current\_obj.}, Area_{previous\_obj}) \quad (3)$$

If the overlap ratio is larger than a preset threshold, one relation of this current object and the previous object is established. This overlap relation list is an important reference list for objects tracking sub-system

2) Matching Analysis: Firstly, we need a simplifying relations process to let those relations limit to three categories: 1 to 1 matching, splitting and merging. In other words, we need to remove some relations to avoid merging and splitting associated with an object at the same time. Instead of finding all possible removing combination, we use the cost function to find one optimal relation to be removed at each evaluating cycle and run this process as a recursive process until the violation doesn't exist. The equations for the evaluation are shown in (4)and(5).

$$Ratio_{Diff} = \left| Area_{curr.} - Area_{pre.} \right| / Max(Area_{curr.}, Area_{pre.}) \quad (4)$$

$$Cost = Ratio_{Diff} / Ratio_{overlap} \quad (5)$$

After simplifying overlap relation list, the matching analysis can be separated into three processes. First process is 1 to 1 matching. This is a quite simple process of matching analysis and we only apply a matching equation to confirm this matching. Then remaining work is only to update object's features: label, type, status, velocity, life, child objects and trajectory.

3) Objects Merging: When multiple previous objects associated with one current object according to the overlap relation list, an objects merging event happened. During the process of objects merging, the main work is to reconstruct the parent-children association. We use the property of children list to present the objects which were merged into the parent object and those objects in children list keep their own properties. If the previous object is with children, we only append the children of this previous object to current object's children list. After appending all objects or their children, the current object becomes a group of those merged objects, like a parent object. The diagram is shown in Fig. 4.

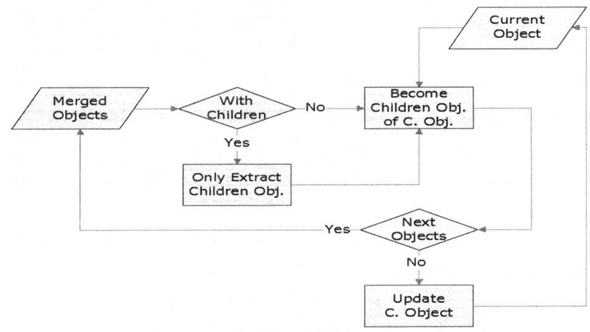


Fig. 4. Diagram of objects merging

4) Object Splitting: Object splitting happens when a single previous object associated with multiple current objects. Splitting events often result from occlusion and objects merging event is one situation of occlusion. Our algorithm deals with these occlusions by the property of children list. If the previous object doesn't have the child object, this splitting results from implicit occlusion. The process is shown in Fig. 5.

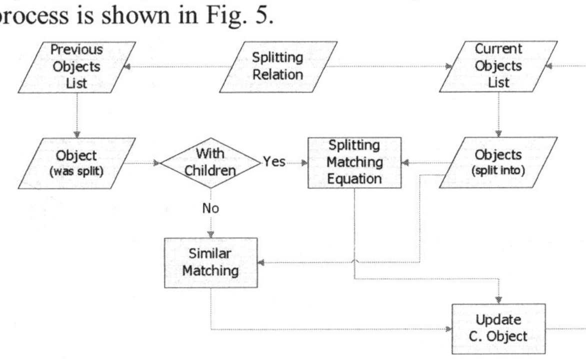


Fig. 5. Diagram of object splitting

We check the status of children objects of the previous object firstly. If none of children object, we do a similar matching process to assign the label of the previous object to the most similar object among these current objects and the others keep unmatched status. If the object has children objects, we will apply the matching algorithm to find the optimal matching between children objects of the previous object and the current ones. The algorithm is based on object's position, size and color. We use three preset thresholds to filter the matching candidates in advance and use the feature of object's position to find the best matching. After matching successfully, we update the matched objects' properties.

5) Other Matching Processes: After three previous matching process, there are still some objects unmatched. Main causes of those unmatched objects are new objects appeared or old objects had left out of FoV. Besides the two main causes, previous objects unmatched sometimes result from occlusion by background or long movement between frames. And, possible reasons of current objects unmatched are split from implicit occlusion, revelation from background occlusion or exceeding the thresholds when previous matching.

Our algorithm presents two additional processes to deal with those situations. One is extended scope matching. In this, we search unmatched current objects inside a preset scope of each unmatched previous object and use a matching function to match them. The other one is isolated objects process. Unmatched previous objects will keep their all properties except life and position. Their life feature will be decreased and their position will be updated with their estimating position. Finally, unmatched current objects will be regarded as new objects and be assigned with new labels.

6) Object's Life: In order to assist tracking framework, our algorithm exploits the temporal information and create a life feature. We use it to define the status of objects. The threshold of life feature can help filter out some ghost objects which usually appear and disappear very quickly. And, this threshold also defines the dead phase for filtering out those objects that can't be tracked anymore for a period time. The other threshold of appearing is used to confirm that the object is valid and our behavior analysis module will not deal with the invalid objects. We also design another type of life feature for the stopped objects. This feature records the consecutive duration of an object which has stopped. If this feature of an object is larger than a preset threshold, this object will become dead, and its region will be copied into background.

D. Behavior Analysis

The main purpose is to analyze the behavior of moving objects to extract traffic parameters. We introduce several modules to deal with high level objects features and output real-time information related with traffic parameters, such as objects classification and accident prediction.

1) Camera Calibration and Object Classification: We introduce a simple method to perform the function of camera calibration and the equations are shown in (6). This can compensate the dimensional distortion which results from different location at camera view.

$$\begin{bmatrix} Xw \\ Yw \end{bmatrix} = M_{calibration} \begin{bmatrix} Xi \\ Yi \\ 1 \end{bmatrix} ; \quad M_{calibration} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \end{bmatrix} \tag{6}$$

Before we apply the camera calibration to our system we need several data to calculate the calibrating matrix **M** in advance. We use the dimensions and the locations of some moving objects to calculate the matrix **M**. Besides, in order to simplify the computation of the camera calibration, we normalized those properties such as width, height and velocity to the calibrated values of central position of the image. Then, we use the calibrated feature to classify moving objects into large cars, cars, motorcycles and people.

2) Accident Prediction: We also present a prediction module for traffic accidents. We analyze the trajectories of any two objects and classify the relation of the objects into four types: both stopped objects, only one moving object, same moving direction and different moving directions. The equations of classifying are shown in (7). The prediction module only processed the later three types.

$$Type = \begin{cases} 0.(Bothstopped), & if\, V1=0 \cap V2=0 \\ I.(Onlyonemoving), & if\,(V1=0 \cap V2 \neq 0) \cup (V1 \neq 0 \cap V2=0) \\ II.(Samedir.), & if\, V1 \neq 0 \cap V2 \neq 0 \cap Vx1*Vy2-Vy1*Vx2=0 \\ III.(Differentdir.), & otherwise \end{cases} \tag{7}$$

Firstly, for the type I, the closest position is calculated and the distance between these two objects in that position is compared with a preset threshold to predict the occurrence of an accident. Secondly, we analyze the type of same moving direction further and only focus on the situation in which trajectories of the two objects are almost on the same line. We check whether the trajectories are almost on the same line with a threshold. Then the occurrence of accidents and the occurring position are predicted. Thirdly, if the moving directions of two objects are different, we use the equations shown in (8) to obtain the time for objects reaching the crossing of their trajectories. Then we can predict the occurrence position of accidents by the equation (9) and obtain the occurring position.

$$\begin{aligned} \Delta &= Vx2 * Vy1 - Vx1 * Vy2 \\ T1 &= (Vx2 * (Py2 - Py1) + Vy2 * (Px2 - Px1)) / \Delta \\ T2 &= (Vx1 * (Py2 - Py1) - Vy1 * (Px2 - Px1)) / \Delta \end{aligned} \tag{8}$$

$$T1,T2 = \begin{cases} T1,T2, & if\, T1>0 \cap T2>0 \cap |T1-T2| < (\frac{1}{2} Dist_{accident}) * (\frac{1}{V1} + \frac{1}{V2}) \\ Noaccident & otherwise \end{cases}$$

$$\begin{aligned} Px_{accient} &= T1*Vx1+Px1 \\ Py_{accidnt} &= T1*Vy1+Py1 \end{aligned} \tag{9}$$

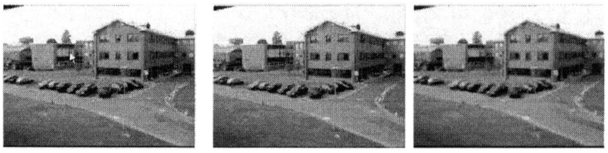
III. Experiment Result

We implemented tracking system on PC with Intel P4 2.4G and 512MB RAM. Inputs are image sequences, and resolution is 320 by 240 pixels. Datasets were captured by a DV or referred to testing samples used by other research.

A. Adaptive Background Updating

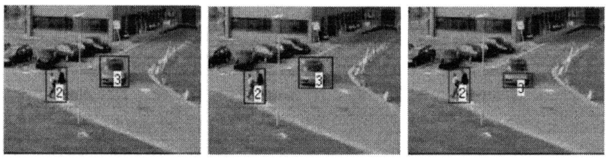
We use the datasets of PETS (Performance Evaluation of Tracking and Surveillance) to perform the adaptive background updating algorithm. The video is PETS2001 [15] dataset2 training video and it's prepared for its significant lighting variation. We also performed the tracking on this video using fixed background updating method with two different updating frequencies in order to compare with our proposed algorithm.

The experimental results are shown in Fig. 6 and 7. We focus on the right-bottom corner of each image. Because the car entered camera's FoV view slowly, some pixels of its region had been regarded as background. This symptom will become more deteriorated if we speeded up the updating frequency. We can find the wrong segmentation occurred in Fig. 7(c) image. It shows that fast updating frequency more easily results in wrong segmentation than slow updating frequency in stable environmental condition.



(a) (b) (c)

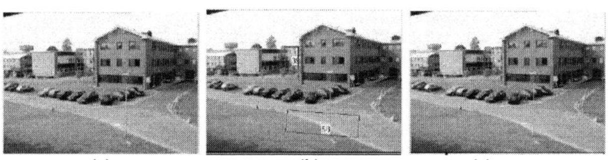
Fig. 6. Background images of frame 2234 (a: adaptive B/G updating; b: updated B/G per 30 frames; c: updated B/G per 10 frames)



(a) (b) (c)

Fig.7. Result images of frame 2234

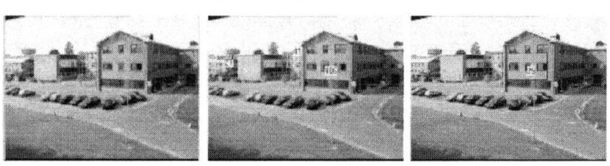
Next, we demonstrated the situation of significant lighting variation and this situation occurred in the later frames of this same video. In Fig. 8(b), it occurred wrong segmentation but the other two method updated background fast enough to cope with lighting changes.



(a) (b) (c)

Fig. 8. Result images of frame 3300

In Fig. 9(c), it also occurred some wrong segmentations and the slow one still tracked the wrong objects. These results prove that our proposed updating algorithm presents good balance and better performance among the normal environmental conditions, gradual change situations and rapid change situations.

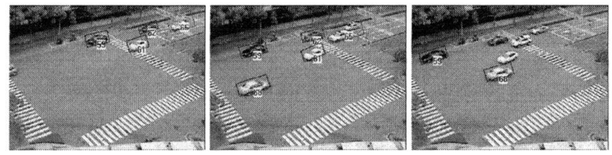


(a) (b) (c)

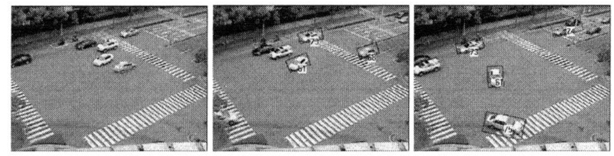
Fig. 9. Background images of frame3392

B. *Occlusion and Splitting of Multiple Objects*

Our proposed tracking algorithm can handle the occlusion and splitting events robustly even through there are various objects or more than two objects included the occlusion group. There are an occlusion of four vehicles occurred in Fig. 10. No. 65 is an object with an implicit occlusion so our system treats it as a single object. During frame #1258 to #1270, those vehicles merged together. No. 73 was created as a new object because it split from an object with implicit occlusion. Then No. 68 and No. 61 vehicle had split from the occlusion group and was tracked with correct label. No. 62 didn't split from the occlusion group because it still merged with another object when it left the camera's FoV.



#1200 #1240 #1258



#1270 #1294 #1340

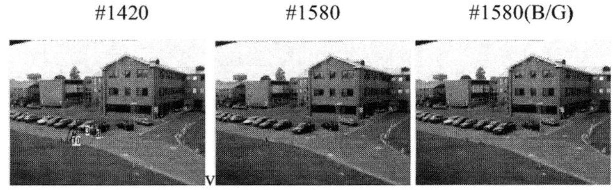
Fig.10. occlusion and splitting of multiple objects

C. *Stopped Objects*

There are often stopped objects or the moving background occurred during the tracking or monitoring. We propose a feature to cope with these special situations and can preset the threshold of this feature to let stopped objects or the ghost foreground be updated to background image. We demonstrated tracking results of stopped objects in Fig 11. In Fig. 11, there was a car (No.2) parked on campus originally and it intended to leave. Another car (No.3) intended to park on this campus. We can find a ghost object (No.8) occurred in frame #1420 because the car (No.2) was regarded as a region of the background previously. In frame #1640, the system duplicated the region of ghost object into the background image when the feature was larger than the threshold. In frame #1750, the car (No.3) which had parked on campus was also regard as the background.

#900 #1100 #1300

#1420 #1580 #1580(B/G)



#1640 #1640(B/G) #1750(B/G) Fig. 11. Tracking results of stopped objects

D. *Behavior Analysis*

We implement a simple but effective method to classify those moving objects into four categories: people, motorcycles, cars and large cars. This method must be processed with the correct information of camera calibration.

In Fig. 12, we show experimental results of objects classification. We designed four counters that present the accumulating quantity of different categories of moving objects and the bounding boxes of objects are drawn with the specific colors directly.

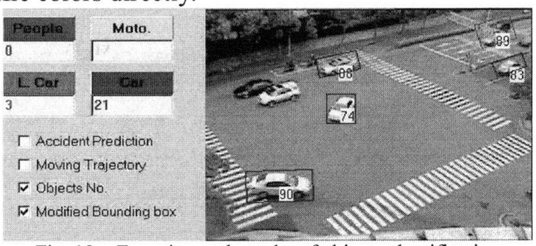


Fig. 12. Experimental results of objects classification

Next, we use the information of trajectory estimation of the objects to implement the prediction of traffic accidents. The results are shown in Fig. 13. We will draw an accident box and two predicting trajectories in the images of tracking results if our system predicts the occurrence of an accident. Because there isn't a real accident happened in frame #128, the accident alarm will disappear soon in the later frames.

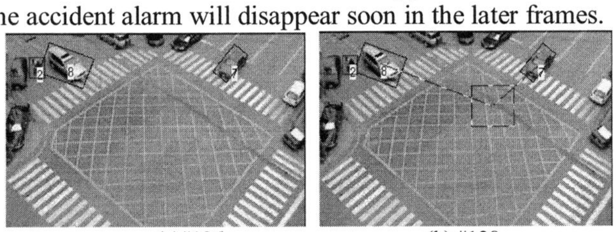

(a)#126 (b) #128

Fig.13. Experimental results of accident prediction

IV. CONCLUSIONS

According to the structure of the proposed algorithms, we implemented a tracking system including the functions of objects classification and accident prediction. Experiments were conducted on real-life traffic video of the intersection and the datasets of other surveillance research. We had demonstrated robust objects segmentation and successful tracking various objects with occlusion or splitting events. The system also extracted useful traffic parameters. Besides, those properties of moving objects or the results of behavior analysis are valuable to the monitoring applications and other surveillance systems.

REFERENCES

[1] N. Friedman and S. Russell, "Image segmentation in video sequences: a probabilistic approach," in *Proc. 13th Conference Uncertainty in Artificial Intelligence*, 1997, pp. 175–181.

[2] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1459 – 1472, Nov. 2004.

[3] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, vol. 34, no. 3, pp. 334 – 352, Aug. 2004.

[4] 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, Mar. 2002.

[5] A. Chachich, A. Pau, A. Barber, K. Kennedy, E. Olejniczak, J. Hackney, Q. Sun, and E. Mireles, "Traffic sensor using a color vision method," in *Proc. of SPIE: Transportation Sensors and Controls: Collision Avoidance, Traffic Management, and ITS*, vol. 2902, pp. 156–165, 1996.

[6] T. N. Tan, G. D. Sullivan, and K. D. Baker, "Model-based localization and recognition of road vehicles," *International Journal of Computer Vision*, vol. 27, no. 1, pp. 5–25, 1998.

[7] S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler, "Tracking groups of people," *Computer Vision and Image Understanding*, vol. 80, no. 1, pp. 42–56, 2000.

[8] R. Cucchiara, P. Mello, and M. Piccaidi, "Image analysis and rule-based reasoning for a traffic monitoring system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 2, pp. 119–130, June 2000.

[9] H. Veeraraghavan, O. Masoud, and N. P. Papanikolopoulos, "Computer vision algorithms for intersection monitoring," *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 2, pp. 78 - 89, June 2003.

[10] Y. K. Jung and Y. S. Ho, "Traffic parameter extraction using video-based vehicle tracking," in *Proc. of 1999 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems*, 1999, pp. 764 – 769.

[11] P. Remagnino, T. Tan, and K. Baker, "Multi-agent visual surveillance of dynamic scenes," *Image and Vision Computing*, vol. 16, no. 8, pp. 529-532, 1998.

[12] M. M. Trivedi, I. Mikic, and G. Kogut, "Distributed video networks for incident detection and management," in *Proc. of 2000 IEEE on Intelligent Transportation Systems*, 2000, pp. 155 – 160.

[13] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, "Traffic monitoring and accident detection at intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 2, pp. 108 – 118, June 2000.

[14] W. Hu, X. Xiao, D. Xie, T. Tan, and S. Maybank, "Traffic accident prediction using 3-D model-based vehicle tracking," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 3, pp. 677 – 694, May 2004.

[15] ftp://pets2001.cs.rdg.ac.uk.