2004 IEEE International Conference on Multimedia and Expo (ICME)

A Real-time and Color-based Computer Vision for Traffic Monitoring System

Mao-Chi Huang and Shwu-Huey Yen*
Department of Computer Science
and Information Engineering
Tamkang University, Taipei, Taiwan, ROC
*Corresponding author: shyen@cs.tku.edu.tw

Abstract - *This paper presents a vision-based traffic monitoring system via analyzing color image sequences of traffic scenes recorded by a stationary camera mounted on a tall building or pedestrian crossing bridge near a traffic light. The system algorithms consist of building initial background, segmenting foreground, shadow separation, vehicle location, vehicle tracking, and background updating. The YCrCb color space is adopted in our algorithm. It provides us an accurate and robust device for foreground and shadow detection. Our system can accurately locate and track vehicles from image sequences. It is fast enough to operate in real time, insensitive to lighting and weather conditions, and only requires a minimal amount of initialization. Experimental results from town scenes are provided which demonstrate the effectiveness of our system.*

Keywords: Traffic monitoring system, traffic scene, vehicle detection, shadow detection, vehicle tracking, background adaptation.

1 Introduction

Traffic management and vehicle information rely on a suite of sensors for estimating traffic parameters. To monitor the traffic condition and provide correct vehicles information is important in nowadays due to heavy traffic volume in our daily life. Currently, magnetic loop detectors and radars are most common devices to monitor traffic condition. The use of an automated system can be more accurate and lower the cost. In particular, the vision-based approach is promising since it requires no pavement adjustments and has more potential advantages such as a cheaper cost, larger detection areas, and it is also more flexible and suitable for vehicles identification.

There are some researches related to the topic of vision-based traffic monitoring system. In [1], it uses pixel wise difference to detect moving vehicles in image sequences. An initial background is not required, but it will have too much noise that is difficult to remove. In [2], a probabilistic approach to segmentation is described. They use the expectation maximization (EM) method to classify each pixel as moving object, shadow or background. In [3], it uses background subtraction to detect moving targets and does not require an initial background. It is fast enough to operate in real time, and insensitive to lighting and weather conditions. The drawback is, it is difficult to update background in traffic jam and high cost computation. In [5], it uses edge projection to remove shadows. It is high cost computation. In [8], it gave a knowledge-based method to separate shadows from vehicles using knowledge including results of edges detection, road direction, date, time, etc. The knowledge is built for a long time. In [10], a system for detecting lane changes of vehicles in a traffic scene is introduced. The approach is similar to the one described in [9] with the addition that trajectories of the vehicles are determined to detect lane changes. There are many methods in the tracking of moving vehicle: model-based method [4], feature-based method [11]. These methods are able to extract not only points, but also information on the motion direction and velocity, but they are generally very expensive in terms of computational time.

In our algorithms we take color information into account. It has many advantages such as it can distinguish two look alike colors, two colors with similar intensity, insensitive to lighting, as well as saving for all the color information of vehicles that can be used for future development, etc. Conventionally, vehicles are located using connective component analysis (CCA). The method uses segmented results and then to filter noise. Followed by labeling connected components (blobs) and analyzing the blob size for eliminating or merging blobs. Due to complicated and some unexpected traffic scenes, not only is CCA high cost in computation, but it may cause some significant information eliminated while filtering irrelevant noise. Instead we propose a projection method to locate vehicles. Projection method doesn't need to remove noises, and it can be processed very efficiently with high correct detection rate. In this paper, a scheme for finding an initial background based on mode is presented. Moreover, in order to remove shadows from foreground and overcome vehicles' shadow occlusion problem, a shadow confidence score method is also presented.

2 System description

The system's algorithms are based on YCrCb color space for its robustness to brightly light backgrounds and preciseness of shadows detection. As in [6], it concluded that YCrCb is the most suitable color space for the detecting of foreground and shadow in traffic image sequences and is robust to camera artifacts as well.

2.1 Initial setup and initial background

Our system requires inputting some information at the first setup for improving the system's efficiency and preciseness. The information includes lane lines, top line, and base line. Lane lines, top line, and base line are used to surround a region for observing behaviors of vehicles as in Figure 1.

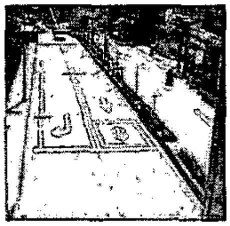

Figure 1. Background

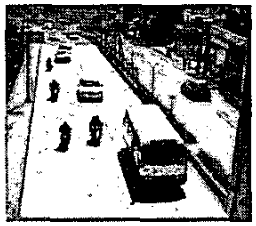

Figure 2. Foreground

Initial background means to locate a scene without any vehicles in the detection zone as in Figure 1. N consecutive frames (with vehicles) taken from video sequence are used and a mode method is proposed in the following to find the initial background. For each pixel located at (x, y) position in the detection zone and each color component respectively:

Step 1. Find a minimum value (*MIN_VAL*) from N frames at position (x, y).

Step 2. Get *INDEX_BASE*, the index value of interval in the corresponding color component, for each frame's pixel. The *INDEX_BASE* can be computed as in equation (1).

$$INDEX\_BASE = \frac{pixel\ (x,y) - MIN\_VAL}{BASE} \quad (1)$$

where *BASE* is the width of the unit interval.

Step 3. For every (x, y) position, find the mode of *INDEX_BASE* and then average all pixels (x, y) belonging to the mode *INDEX_BASE*. It is the value of initial background pixel (x, y).

We do step1 to step3 for every pixel in the detection zone and for its Y, Cr, and Cb values respectively to obtain the initial background. The Bases of Y, Cr, and Cb have been empirically determined to be 10, 5 and 5.

2.2 Foreground segmentation and shadow separation

We use the current background image and the YCrCB range to find the foreground objects from a current image. It is similar to a color slicing method by cube. By comparing the pixel (x, y) of the current image with the pixel at the same location (x, y) on the background, if any of the Y, Cr, Cb component is outside the YCrCb range then it will be classified as the foreground or background otherwise. The range of Y, Cr, and Cb have been empirically determined to be 50, 7 and 7. Figure 2 shows the result of the foreground objects.

Figure 3. Shadow separation

Figure 4. Locating

Foreground objects include vehicles and cast shadows. We adopt the idea from [7] and propose a method based on shadow confidence score (SCS) to separate vehicles and cast shadows from the foreground objects. To obtain the SCS, the current image is subtracted from the current background image in the luminance, chrominance and horizontal/vertical gradient density respectively on YCrCb color space. By mapping through various shadow score functions, these subtraction results are transformed into the shadow confidence scores, which provide indication of the likelihood of the pixels belonging to the cast shadow region. The method is implemented via the following steps.

Step 1. To find Rectangular Foreground Regions (*RFR*) which are the rectangles inscribing foreground objects. We use the projection method as detailed in Section 3.5 to build *RFR*.

Step 2. To compute Luminance Score (S_Y) and Chrominance Score (S_{Cr} and S_{Cb}) as given by

$$S_Y(x,y) = \begin{cases} 1 & Y(x,y) \leq T_Y \\ \dfrac{Y(x,y)}{T_Y} & T_Y < Y(x,y) < 0 \\ 0 & Y(x,y) \geq 0 \end{cases} \quad (2)$$

$$Y(x,y) = CI\_Y(x,y) - CB\_Y(x,y), \quad \forall (x,y) \quad where \quad RFR = 1. \quad (3)$$

$Y(x, y)$ is the luminance difference between the current image and the current background image at location (x, y). $CI\_Y(x, y)$ and $CB\_Y(x, y)$ are luminance at pixel (x, y) on the current image and the current background. For every (x, y), *RFR* is "1" if it is located inside *RFR* and "0" otherwise. Since the cast shadow is darker than the background, the luminance threshold T_Y is empirically determined to be –100 in our system.

$$S_{Cr}(x,y) = \begin{cases} 1 & Cr(x,y) \leq T_{Cr1} \\ \dfrac{T_{Cr2} - Cr(x,y)}{T_{Cr2} - T_{Cr1}} & T_{Cr1} < Cr(x,y) < T_{Cr2} \\ 0 & Cr(x,y) \geq T_{Cr2} \end{cases} \quad (4)$$

$$Cr(x,y) = |CI\_Cr(x,y) - CB\_Cr(x,y)|, \quad \forall (x,y) \quad where \quad RFR = 1. \quad (5)$$

$Cr(x, y)$ is the magnitude of chrominance difference between the current image and the current background image at location (x, y). $CI\_Cr(x, y)$ and $CB\_Cr(x, y)$ are the chrominance at pixel (x, y) on current image and the current background. Since the chrominance of the cast shadow should be similar to the background if they are not the same, thus thresholds T_{Cr1} and T_{Cr2} are empirically determined to be 5 and 10 in our system. Also, S_{Cb} is computed similarly as S_{Cr} with T_{Cb1} and T_{Cb2} to be 10 and 20 in our experiment.

Step 3. To compute vertical/horizontal Gradient Density Scores $S_{VG}(x)_k$ and $S_{HG}(y)_k$ as given by

$$G(x,y) = CI\_G(x,y) - CB\_G(x,y), \quad \forall(x,y) \quad where \quad RFR = 1 \qquad (6)$$

$G(x, y)$ is the gradient difference between the current image and the current background image at location (x, y). $CI\_G(x, y)$ and $CB\_G(x, y)$ are the gradients at location (x, y) on current image and the current background.

$$VD(x)_k = \frac{\sum_{i=x-r}^{x+r} VG(i)_k}{Total} \quad \forall(x) \quad where \quad RFR_k = 1 \qquad (7)$$

For each foreground object k embedded in RFR_k. $VD(x)_k$, the vertical density at location (x), is the average gradient difference over a consecutive $2r$+1 columns with (x) located at the center column. r is empirically determined to be 2 in our system. VG is the column sum of vertical gradient difference similar to (6). $Total$ is the total number of pixels being considered in these $2r$+1 columns. The vertical Gradient Density Score $S_{VG}(x)_k$ of the object k at location (x) is calculated as

$$S_{VG}(x)_k = \begin{cases} 1 & VD(x)_k \le T_{G1,k} \\ \dfrac{T_{G2,k} - VD(x)_k}{T_{G2,k} - T_{G1,k}} & T_{G1,k} < VD(x)_k < T_{G2,k} \\ 0 & VD(x)_k \ge T_{G2,k} \end{cases} \qquad (8)$$

$$with \quad T_{G1,k} = \frac{TG_k}{Total\_RFR_k} * PST. \qquad (9)$$

where $T_{G1,k}$ is the gradient threshold of the RFR_k that is proportional to the ratio of TG_k, total of gradient differences of RFR_K, with $Total\_RFR_k$, total number of pixels in RFR_k. Since the difference in gradient density between the cast shadow and the background is smaller than the difference between the vehicles and the background, a high score is assigned when $VD(x)_k$ is small enough. PST, a percentage, is empirically determined to be 50% and $T_{G2,k} = T_{G1,k} * 2$ in this paper. $S_{HG}(y)_k$ is computed similarly as $S_{VG}(x)_k$.

Step 4. Combining all the scores to a shadow confidence score $S = \max(S_V, S_H)$ where

$$\left.\begin{aligned} S_V(x,y) &= S_Y(x,y) * S_{Cr}(x,y) * S_{Cb}(x,y) * S_{VG}(x)_k \\ S_H(x,y) &= S_Y(x,y) * S_{Cr}(x,y) * S_{Cb}(x,y) * S_{HG}(x)_k \end{aligned}\right\} (10)$$

If $S \ge T_S$ then $CI(x, y)$ is a cast shadow pixel for $\forall$ (x, y) with RFR =1

$S_V(x, y)$ and $S_H(x, y)$ are vertical/horizontal shadow confidence scores at location (x, y). T_S is the threshold of the shadow confidence score and is empirically determined to be 0.7 in our system. Figure 3 shows the result of shadow removing of the Figure 2. Furthermore, since the shadow confidence score within vehicles' shadow zone will be much higher than the scores of vehicles, so our method can also overcome vehicles' shadow occlusion problem.

2.3 Vehicles locating

We use projection method to locate vehicles, shown in Figure 4, as stated in the following steps:

Step 1. Take horizontal projection of the lane. We process one lane at one time.

Step 2. To determine the maximum projection area, the peak of the projection is found and extending the area towards north and south directions as long as the projection counts is more than a given threshold th. The length of a vehicle is thus determined.

Step 3. To determine the dimension of the vehicles as well as whether the maximum projection area has parallel vehicles (e.g. motorcycles), we will process the following steps.

Step a. Do a vertical projection of the maximum projection area obtained from step 2.

Step b. Again find the maximum projection area of the corresponding vertical projection as in step 2. Thus the vehicle is located and both the length and width of the vehicle are determined.

Step c. Discard the maximum projection area of the vertical projection. If there are other peaks with vertical projection counts over the threshold th, then go to step b.

Step 4. Discard the maximum projection area of the horizontal projection. If there are other peaks with the horizontal projection counts over the threshold th then go to step 2.

Furthermore, an extra width towards the separation line is being considered as a lane in order to take care the case of vehicles changing lanes, which cause an overlapping in both lanes. Thus we merged the located vehicles that are overlapped in part of area to be one single vehicle. Notice that we define a dynamic threshold (th) of projections such that they are considered possible maximum projection regions or noise otherwise. Since the lane width and dimension of a vehicle will become larger when it moves from the top line towards the base line of the detection zone, thus the noises will also become larger, so th is determined as following equation.

$$th = th_{\min} + \frac{(th_{\max} - th_{\min}) * (HP - TopLine)}{BaseLine - TopLine} \qquad (11)$$

where th_{Max} and th_{Min} are the maximum and minimum value of th. HP is the position of peak of the horizontal projection. *BaseLine* and *TopLine* are the position of base line and top line. The values of th_{Max} and th_{Min} have been both empirically determined to be 9 and 3.

2.4 Tracking and background updating

The centroid of a vehicle is used for tracking. For two consecutive frames, centroid of every vehicle is located. Centroid (i, j) on the first frame will be matched with centroid (k, l) on next frame if their distance is the shortest and $l \geq j$.

We update the current background for every eight frames. To update, those pixels outside RFR in the current image will become the background pixels. For those background pixels not being updated will remain the previous background color information. The update current background method is fast enough to operate in real time and can overcome lighting and weather conditions changed.

Table 1. Average processing time (sec)

Step	Average Time
Initial Background	8.320
Find a Foreground	0.028
Shadow Separation	0.070
Vehicles Locating and Tracking	0.010
Background Updating	0.005

3 Experimental results

The system was implemented on Intel Celeron 2 GHz PC. We tested the system on image sequences of town traffic scenes. The system successfully achieved the tasks of vehicles detection, shadow separation, and tracking for most vehicles. Our test result shows that 90% of the vehicles were correctly detected and tracked. Most detecting and tracking errors occur near the start point of the detection zone due to occlusion of vehicles. But once vehicles move towards the camera after one or two frames, they can be correctly detected and tracked, so the vehicles still have correct information recorded. As in Figure 4, although vehicles have look-alike colors to the background's, with help of YCrCb color system, all vehicles were correctly detected and tracked with shadow removed.

The system has worked in real time. The processing was done at a frame rate of 8 frames/s. The image size of sequence scenes is 640*480 pixels. Velocity of vehicles is about 20-50 km/hr in the town. The system's average processing time is shown in Table 1. The time for initial background is more than 8 seconds due to processing consecutive eighteen frames. Initial background only do once at the beginning of the system setup stage, so the time will not be concerned for the later process. Background updating is processed for every eight frames, so the time it takes is not too much of our concern.

4 Conclusions and future work

In this paper, we proposed a mode method to find an initial background. The method can obtain a nearly perfect background image. We also proposed a projection method to locate vehicles. Projection method doesn't need to remove noises, and it can be processed very efficiently with high correct detection rate. Moreover, a shadow confidence score method is also proposed to remove shadows from foreground and can overcome vehicles' shadow occlusion problem. A system is built according to our proposed algorithms. It is able to work robustly under most circumstances. In our future work, we will extend the system to extreme weather conditions such as rainy day, night, etc, and we will also overcome vehicles' overlap and occluding problems.

References

[1] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving target classification and tracking form real-time video," in Proc. IEEE Workshop Applications of Computer Vision, 1998, pp. 8-14.

[2] N. Friedman and S. Russell, "Image segmentation in video sequences," in Proc. 13th Conf. Uncertainty in Artificial Intelligence, Providence, RI, 1997.

[3] S. Gupte, O. Masoud, R. F. K. Martin, and N. P. Papanikolopoulos, "Detection and classification of vehicles," in Proc. IEEE Intelligent Transportation Systems, Vol.3, No.1, March 2002, pp. 37-47.

[4] A. Blake, M. Isard, and D. Reynard, "Learning to track the visual motion of contours," Artificial Intelligence, 1995, pp. 101-131.

[5] J. B. Kim, C. W. Lee, K. M. Lee, T. S. Yun, and H. J. Kim, "Wavelet-based vehicle tracking for automatic traffic surveillance," in Proc. IEEE Catalogue No. 01CH37239, 2001, pp. 313-316.

[6] P. Kumar, K. Sengupta, and A. Lee, "A comparative study of different color spaces for foreground and shadow detection for traffic monitoring system," in Proc. IEEE 5th Conf. Intelligent Transportation Systems, Sep. 2002, pp. 100-105.

[7] G.S.K. Fung, N.H.C. Yung, G.K.H. Pang, and A.H.S. Lai, "Towards detection of moving cast shadows for visual traffic surveillance," in Proc. IEEE 2001 Conf. pp. 2505-2510.

[8] WU Yi-Ming, YE Xiu-Qing, and GU Wei-Kang, "A shadow handler in traffic monitoring system," in Proc. IEEE 2002 Conf. pp. 303-307.

[9] O. Masoud and N. P. Papanikolopoulos, "Robust pedestrian tracking using a model-based approach," in Proc. IEEE Conf. Intelligent Transportation Systems, Nov. 1997, pp. 338–343.

[10] O. Masoud, N. P. Papanikolopoulos, and E. Kwon, "Vision-based monitoring of weaving sections," in Proc. IEEE Conf. Intelligent Transportation Systems, Oct. 1999, pp. 770–775.

[11] S. Smith, and J. Brady, "Asset-2: real-time motion segmentation and shape tracking," IEEE Trans. PAMI, 17(8), 1995, pp. 814-820.