Automatic Vehicle Counting from Video for Traffic Flow Analysis

Conference Paper · July 2007

DOI: 10.1109/IVS.2007.4290146 · Source: IEEE Xplore

3 authors:

Erhan Bas
Howard Hughes Medical Institute
30 PUBLICATIONS **380** CITATIONS
SEE PROFILE

A. Murat Tekalp
Koc University
531 PUBLICATIONS **14,421** CITATIONS
SEE PROFILE

F. Sibel Salman
Koc University
62 PUBLICATIONS **1,204** CITATIONS
SEE PROFILE

Automatic Vehicle Counting from Video for Traffic Flow Analysis

Erhan Baş, A. Murat Tekalp, *Fellow, IEEE,* and F. Sibel Salman

College of Engineering, Koç University, 34450 Sariyer, Istanbul, Turkey

***Abstract*—We propose a new video analysis method for counting vehicles, where we use an adaptive bounding box size to detect and track vehicles according to their estimated distance from the camera given the scene-camera geometry. We employ adaptive background subtraction and Kalman filtering for road/vehicle detection and tracking, respectively. Effectiveness of the proposed method for vehicle counting is demonstrated on several video recordings taken at different time periods in a day at one location in the city of Istanbul.**

I. INTRODUCTION

Traffic congestion is one of the biggest problems in many metropolises, including the city of Istanbul. Istanbul has over 2.35 million motorized vehicles according to July 2006 figures. Any analysis aimed at improving the problems related to congestion and enabling efficient transportation within the city requires collection of reliable data. In order to monitor the traffic flow, Istanbul municipality has installed more than 110 video cameras along the major arteries in the city [1], and this number is growing. Hence, it is of interest to digitally process and analyze these videos in real-time in order to extract reliable data on traffic flow and to detect traffic events. For example, as a result of such video analysis, traffic density in major arteries can be estimated and the least congested routes and travel time estimates can be computed and transmitted to drivers over cell phones. In addition, the videos may be analyzed to automatically detect events such as accidents and traffic violations, as well as snow accumulation and other weather conditions. The data may also be used as input for traffic models and related planning problems.

Several studies exist in the literature on automatic video analysis for vehicle detection and tracking. For example, a double-difference operator with gradient magnitude has been used to detect vehicles [2]; however, it cannot easily handle interframe luminance variations. Adaptive background subtraction algorithms have been used for vehicle detection, which allows changes in lighting and weather conditions [3] [4], but they usually require a priori information about the scene without any moving vehicles and have problems with occlusions. Optical flow techniques have been used to estimate the motion between subsequent frames [5] [6]. Furthermore, 3-D models have been previously implemented such as Sullivan [7], which recover trajectories with high accuracy. However, this approach requires detailed geometric object models for all detected vehicles on the highway. Likewise, 3-D tracking algorithms based on detection of vehicles with probabilistic line feature grouping method such as [8] have been previously implemented. In [4], a feature based algorithm is used for tracking multiple vehicles in congested traffic with occlusion reasoning, where sub-features such as corner locations are tracked. In addition, Kalman filter has been widely used in automatic traffic surveillance systems. For example, Xie *et al*. [9] use position and size information as state variables to track vehicle positions with different set of features. Similarly, [10] uses Kalman filter for tracking vehicles extracted from background models. They implemented a shadow removal algorithm to extract the size and linearity features of vehicles for the purpose of categorizing them.

In this study, we propose a new traffic video analysis method that accounts for the geometry of the scene, where adaptive bounding box size is used to detect and track vehicles according to their estimated distance from the camera. In the rest of the paper, we provide algorithms for vehicle detection and tracking, and report results obtained by implementing these algorithms on several video recordings taken at different time intervals at one location in the city of Istanbul.

II. VIDEO ANALYSIS FOR VEHICLE DETECTION AND TRACKING

Video cameras were first introduced to traffic management for roadway surveillance by transmitting closed-circuit television imagery to a human operator for interpretation. Present-day traffic management systems utilize digital video processing to automatically analyze the scene of interest and extract information for traffic monitoring. A video processor (VP) typically consists of one or more cameras, a microprocessor-based computer for digitizing and analyzing the imagery, and software for interpreting the images and converting them into traffic flow data. The Traffic Control Center of Istanbul Municipality collects real-time images using a VP system consisting of 110 cameras of various characteristics. Currently, all of the images are displayed at a control

room and are monitored by operators to detect any incidents such as accidents or unexpected road conditions.

Video processors can typically classify vehicles by their length and report vehicle density and speed for each class and lane. VPs that track vehicles may also have the capability to register turning movements and lane changes. Vehicle density and link travel time are potential traffic parameters that can be obtained by analyzing data from a series of image processors installed along a section of roadway.

In the following, vehicle detection by background subtraction is addressed in Section II.A and vehicle tracking by Kalman filtering is presented in Section II.B. These algorithms are used for analyzing several video recordings taken at different time intervals at one location in the city of Istanbul. Results are provided in Section IV.

A. Vehicle Detection

In order to distinguish moving vehicles from the static background, we model the background scene with GMM (Gaussian Mixture Modeling) as in [11]. Each pixel color is modeled by a mixture of K Gaussian distributions with specified weight parameters (K is some number from 3 to 5) over a time interval. The weight parameter of a certain mixture is the data proportion that is accounted for by the corresponding mixture. The idea for moving object detection lies in the wider color characteristic of moving objects due to different reflecting surfaces during the movement. Since steady objects form tight color clusters, the rule to decide whether a new pixel belongs to the background or the foreground is based on the variance of this pixel in comparison to the background model. That is, the color value of every pixel is checked to decide whether it matches the GMM or not. A pixel color value that is less than 2.5 standard deviations from the mean of any of the K distributions is decided to belong to the background. If a match occurs, then that mixture (weight parameter, mean and covariance) is updated with the new pixel color value; if no match occurs then a new mixture model is created with the mean at that pixel value and an initially high variance value. The Least probable (smallest weighted) mixture is replaced with the new model with a small weighting.

It is important to adapt the background model to small changes such as brightness variations or new entries to the background. For this purpose, an online update algorithm is implemented. The probability of observing a certain pixel value for a channel (a vector for R-G-B channels, or a scalar value for a single Gray level channel) after t frames is given as

$$P(X_t) = \sum_{i=1}^{K} w_{i,t} \bullet \eta(X_t, \mu_{i,t}, \Sigma_{i,t})$$

where $w_{i,t}$ is the weighting parameter of the i^{th} Gaussian mixture. Here, $\mu_{i,t}$ and $\sum_{it}$ represent the mean value and covariance matrix, respectively, of the i^{th} Gaussian distribution computed from 'n' channel pixel value history. In our implementation we set n=3. Mixtures are ordered in descending probabilities according to their time interval in the scene. For background modeling, the first T mixtures among K Normal distributions are used. The number T is found from

$$T = \arg\min_n (\sum_{i=1}^{n} w_i > thr)$$

where the parameter *thr* represents the minimum portion of the data required to form a background model.

Difference images are formed by subtracting the current frame from the background model in each channel. By thresholding each channel (difference images), three single channel binary foreground images are obtained and by intersecting these foreground images, which belong to different channels, a final single channel binary foreground image is obtained. Then, the foreground objects are detected and labeled using connected component analysis with adaptive blob size, where the blob size varies according to the position of the blob in the picture and imaging as explained below.

Adaptive Blob Size Fitting

For a fixed camera configuration, in imaging geometries where the road is along the z-axis of the camera, vehicles further away from the camera are expected to be smaller in size; hence are modeled by smaller blobs (Fig. I). The adaptation of the blob size depends on the relative position of the camera with respect to the road. First, a mask of the road is extracted from vehicle trajectories to reduce the search space (Fig. II). This mask is also used to fit a cubic road equation that approximates the road curvature (for example the highway in Fig. III).

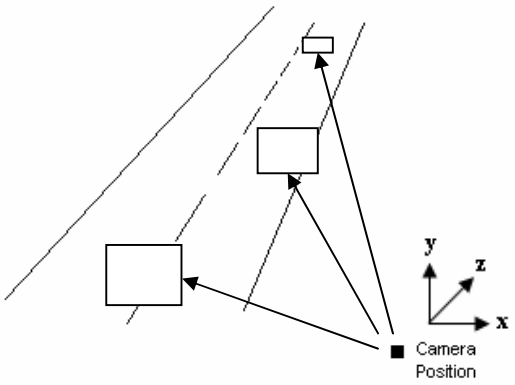

Fig. I. Adaptation of blob sizes. For a fixed camera set-up, blob size of a vehicle is approximated using its distance from the camera position.

The relative road equation is calculated from estimated center strip by fitting a 2nd or higher order curve. Experimentally, we observed that a 3rd order curve equation fits better to road curvature. The relative road equation enables us to approximate the values of

parameters such as the relative pixel speed and vehicle size. These parameters can later be used to estimate traffic flow, and give information about imaging geometry without having camera calibration parameters. Moreover, by setting an upper threshold value for blob sizes gives priori information about anomaly cases such as occlusion and traffic congestion.

Fig. II. Mask of the road is extracted from the vehicle positions and their trajectories'.

Fig. III. Road equation, shown as the white curve, is extracted from the mask of detection region.

Finally, bounding rectangles are fitted to each detected blob, and the centers of the rectangles are marked as the vehicle position (Fig. III).

B. Vehicle Tracking

In each frame, each detected vehicle is represented with a two-state Kalman filter, based on the constant-velocity motion model

$$x_t = x_{t-1} + v_{x,t-1} \bullet T$$

$$y_t = y_{t-1} + v_{y,t-1} \bullet T$$

where T denotes the frame capture rate of the acquisition system. Velocities in vertical and horizontal directions are represented with v_x and v_y. Here x_t and, y_t denote the center of mass of the rectangles. The state-space model is formulated with state s_t and observation z_t as:

$$s_{t+1} = F \bullet s_t + G \bullet u_t$$

$$z_t = H \bullet s_t + o_t$$

where,

$$s_t = [x_t \; y_t \; v_{x,t} \; v_{y,t}]^T$$

$$z_t = [x_t \; y_t]^T$$

and

$$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The process noise u_t and the measurement noise o_t are assumed to be uncorrelated, with zero-mean white Gaussian distributions and corresponding covariance matrices Q and R as in [12]. Moreover, the rectangle center that is obtained by foreground segmentation is used as the observation, z, for the Kalman filter.

In order to decide if an observed vehicle position belongs to one of i) a previously existing vehicle, ii) a new incoming vehicle, or iii) a missing vehicle that occluded in the previous frames, we use the Euclidean distance between the optical flow estimate of the i^{th} vehicle's center position ($I_{t,i}$) and the observed vehicles' position at time t (θ_t). We denote this distance by $\delta(I_{t,i}, \theta_t)$. Decision varies according to the vehicles' positions in the imaging geometry. Since displacement would be higher for closer pixel locations, a higher threshold value is used for those regions.

The goal of optical flow calculation is to find the location $v = u+d$ in the frame at time $t+1$ for an image point $u = [u_x \; u_y]^T$ at time t, such that the windowed image regions centered at locations u and v, respectively $I_t(u)$ and $I_{t+1}(v)$, are "similar". The vector $d = [d_x \; d_y]^T$ is called the pixel motion or the optical flow vector at u. Similarity is defined in the mean square sense, and d is the vector that minimizes the residual function

$$\varepsilon(d_x, d_y) = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} \left(I_t(x,y) - I_{t+1}\left(x+d_x, y+d_y\right) \right)^2$$

where w_x and w_y are integers that define the search neighborhood for optical flow calculation. For each match of observed vehicle position with optical flow estimate of an existing vehicle, matched vehicles' Kalman filter is updated with the observed vehicle position.

C. Occlusion Reasoning

Different cases are studied by relating observed rectangle centers with optical flow results. Occlusion of multi-vehicles is classified into two categories: occlusion of vehicles and split of occluded vehicles. In each frame, a

log is used to keep track of the points, whether they are new observations or observations correspond to existing vehicles, including their occlusion status in terms of occluded frame number. The following steps are used to match an observed rectangle center with an existing vehicle and track them.

In each frame:

1. For each optical flow estimate vehicle center, rectangle centers are matched according to their Euclidean-distance $\delta(I_{t,i}, \theta_t)$ as explained below.
   a. If the distance $\delta(I_{t,i}, \theta_t)$ is below the threshold, an exact match occurs and the current Kalman filter ($K_{t,i}$) is corrected with matched rectangle center (θ_t).
   b. If the mean-square distance is above the threshold, there appears a split of an occluded vehicle. Kalman filter belonging to the split object is corrected with its prediction. For vehicles entering the scene, new Kalman filters are initialized at step 2.
   c. If the distance is above the threshold and two different optical flow estimates match with the same rectangle center, occlusion case is valid for the objects. A new Kalman filter is initialized at step 2 and prediction values are used as observations for the occluded objects' Kalman filter.
   d. If there is no match, i.e., distance is too high for the optical flow of a vehicle ($I_{t,i}$), the vehicle's Kalman filter is corrected with its prediction since there is not enough observation for the existing vehicle.
2. A new Kalman filter is initialized for each unmatched rectangle center.

By keeping a log table, split objects' centers and their corresponding Kalman filters can be removed while tracking for occluded vehicles for a desired frame period. Also for the no match case, vehicles that are no longer in the scene can be detected and it is possible to estimate a traffic flow from the acquired data.

In order to count passing vehicles, we define a boundary on the image for each outbound and inbound lane to form a region of interest as in Figure IV. When we detect that the position of a tracked vehicle gets out of this region in terms of pixel values, the counting algorithm increases the vehicle count by 1 for the corresponding lane.

III. SPECIAL CONSIDERATIONS

In order to create a robust, adaptive tracking system that can handle environmental and lighting changes, the proposed algorithm is tested under different video scene conditions such as night time and weather condition.

A. Night Time

For night time recordings, a simple modification is applied to the algorithm. To reduce lighting effects, a higher threshold is set for the bounding blob size and a smaller threshold is set for difference images (Fig. V).

Fig. IV. A tracking output: Detected vehicles are shown by rectangles while tracked positions are shown by dots. White line indicates the boundary of inbound and outbound lanes for counting.

Fig. V. The white solid line represents the lower limit of the tracking boundary whereas yellow dots represent tracked vehicles and blue rectangles are detected vehicles.

B. Weather Conditions

The algorithm is further improved with histogram and bounding box size constraints to handle weather conditions such as snow accumulation. To eliminate non-vehicles, histogram of every detected rectangle's background model is calculated. A vehicle or non-vehicle decision is made based on the intensity distribution of the background model for the corresponding rectangle. Moreover, height to width ratio of detected rectangles gives information about the classification of detected objects such as: human, truck, or small vehicle.

An example can be seen in Figures VI and VII, where the moving pedestrians are eliminated with the improvement in the algorithm.

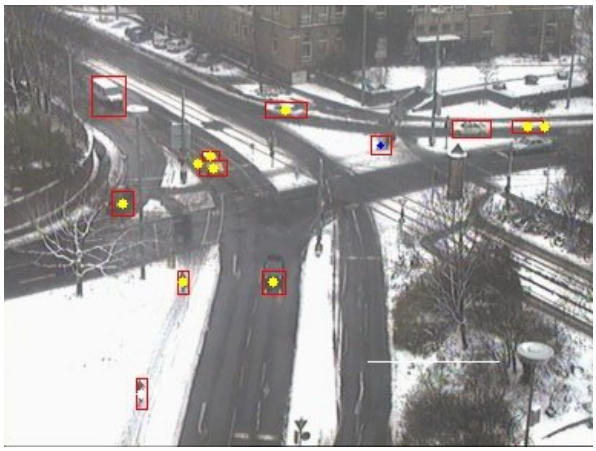

Fig. VI. Tracking result without any improvement.

Fig. VII. Tracking result with histogram information. The moving pedestrians are eliminated.

IV. Results

In this section we present the results from an initial investigation of the proposed algorithm's effectiveness by implementing it on several video recordings taken at different time periods in a day. Images of a two sided highway in Camlica, Istanbul are taken by a stationary camera at 2 different time periods: (CM1) beginning at 12.01 pm and (CM2) at 4.00 pm. The algorithm is implemented with C++ on a Laptop with 1.60 GHz speed, 512MB Ram, and Intel Pentium(R) M processor under Windows OS. In this platform our implementation processes 1000 frames in 357 seconds. The algorithm runs at 3 fps at real time, but it is possible to increase its speed by optimizing the code and by modeling the background only for the extracted foreground mask region. In Table I we report the number of departing vehicles counted by the proposed algorithm, and by inspection of the video scenes in order to measure the effectiveness of the algorithm.

TABLE I
AVERAGE COUNTS OF TRACKED VEHICLES

Duration of video	CM1(A)	CM1(I)	Error	CM2(A)	CM2(I)	Error
1380 (1)	55-44	54-56	1-12	47-31	50-42	3-11
2760 (2)	51-65	54-66	3-1	59-65	60-60	1-5
4140 (3)	43-51	53-56	10-5	37-60	57-70	20-10
5520 (4)	31-56	34-61	3-5	57-53	61-48	4-5
6900 (5)	43-67	50-46	7-21	55-50	65-42	10-8
8280 (6)	53-43	66-52	13-9	58-56	65-46	7-10
9660 (7)	50-65	66-67	16-2	60-62	59-52	1-10
AVG	47-56	54-58	7,6-7,8	53-54	59-51	6,5-8,4
SUM	326-391	377-404	53-55	373-377	417-360	46-59
%Error			14-13			11-16
Std			5,65-6,93			6,75-2,5

Table I gives the number of vehicles departing from the scene in one minute time intervals. During each minute of the video 1380 frames exist. Cameras used in this work operate at 23 frames per second; hence each row of the table corresponds to approximately a one-minute time interval. In the columns, (A) represents the counts calculated by the algorithm and (I) represents the inspection counts. The two numbers in each column are the counts of the number of departing vehicles in both directions of traffic, separately. For example, during the time period CM1, the algorithm counted the number of vehicles departing in both directions during the first minute of the video and found 55 vehicles in the inbound lanes and 44 vehicles in the outbound lanes (Fig. IV).

In Table I, it is seen that, the tracking algorithm underestimates the count of the vehicles in general; the intuition behind this fact is that for long durations occluded vehicles are assigned as a single vehicle. By extending the tracking time of vehicles, these results can be further improved. The reason for the high difference in algorithm and inspection results at 5^{th} and 7^{th} intervals of CM1 is due to the environment conditions. The proposed algorithm is implemented for stationary setups but cameras are affected from environmental factors such as wind.

Since our proposed model forces to assign a Kalman filter to every new bounding box (rectangle) and keeps the other objects' Kalman filters, problems with tracking mostly occur from foreground segmentation. For example, in over 4500 frames there are only 4 tracking errors due to the nature of the adaptive bounding box size in CM1; small objects in high threshold regions are omitted while implementing connected-component analysis. On the other hand, 29 vehicles' tracking is lost only in the left lane of the highway in 4500 frames in CM2. This is the result of assigning foreground regions to background due to the high similarity between the objects and the background modeling for low resolution images.

V. Conclusions

Although the obtained results are promising, the algorithm still needs further modifications. To enable detecting and tracking in day and night recordings, background subtraction with shadow elimination techniques can be used. In order to improve results lane based tracking of vehicles can be implemented rather than tracking vehicles all over the highway., Cross-roads can be added as an extension to special considerations since it becomes challenging to track vehicles as they change directions. To differentiate the no-vehicle case from the congested traffic case, bounding-box sizes of detected vehicles will be used.

Acknowledgment

This work has been supported by Istanbul Metropolitan Municipality, Strategic Planning Department. The authors would like to thank the Municipality for providing highway video recordings.

References

[1] http://tkm.ibb.gov.tr/kameraAna.aspx

[2] R. Cucchiara, M. Piccardi, and P. Mello, "Image analysis and rule-based reasoning for a traffic monitoring system," IEEE Trans. on Intelligent Transportation Systems, vol. 1, no. 2, pp. 119-130, Jun. 2000.

[3] K. P. Karmann and A. von Brandt, "Moving object recognition using an adaptive background memory," in Time-Varying Image Processing and Moving Object Recognition, 2 (edited by V. Capellini), Elsevier, Amsterdam, The Netherlands, 1990.

[4] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, "A real-time computer vision system for measuring traffic parameters," in Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 496-501, Puerto Rico, June 1997.

[5] B. Li and R. Chellappa, "A generic approach to simultaneous tracking and verification in video," IEEE Trans. on Image Processing, vol. 11, no. 5, pp. 530-544, May 2002.

[6] H. Tao, H. S. Sawhney, and R. Kumar, "Object tracking with Bayesian estimation of dynamic layer representations", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 24, no. 1, pp. 75-89, Jan. 2002.

[7] K. Baker and G. Sullivan, "Performance assessment of model-based tracking" Proc. of the IEEE Workshop on Applications of Computer Vision, Palm Springs, CA, pp. 28-35, 1992.

[8] Z. Kim and J. Malik, "High-Quality Vehicle Trajectory Generation from Video Data Based on Vehicle Detection and Description", Proc. of IEEE Conf. on Intelligent Transportation Systems, pp. 176-182, 2003.

[9] Lei Xie, Guangxi Zhu, Yuqi Wang, Haixiang Xu, Zhenming Zhang, "Real-time Vehicles Tracking Based on Kalman Filter in a Video-based ITS", Proc. of IEEE Conf. on Communications, Circuits and Systems, vol. 2, p. 886, May 2005.

[10] Jun-Wei Hsieh, Shih-Hao Yung-Sheng Chen, and Wen-Fong Hu, "Automatic Traffic Surveillance System for Vehicle Tracking and Classification", IEEE Trans. on Intelligent Transportation Systems, vol. 7, pp. 175-187, June 2006

[11] C. Stauffer, W.E.L. Grimson. "Adaptive Background Mixture Models for Real-Time Tracking" IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'99), vol. 2 pp. 2246, 1999.

[12] R.K. Mehra, "On the identification of variances and adaptive Kalman filtering", IEEE Transactions on Automatic Control, AC-15, pp. 175-183, 1970