A video-based real-time vehicle counting system using adaptive background method

Manchun Lei, Damien Lefloch, Pierre Gouton, Kadder Madani

HAL Id: hal-02879739
https://hal.archives-ouvertes.fr/hal-02879739

Submitted on 24 Jun 2020

A video-based real-time vehicle counting system using adaptive background method

Manchun LEI[a, b], Damien LEFLOCH[a], Pierre GOUTON[a] and Kadder MADANI[b]

a University of Burgundy, BP 47870, 21078 Dijon Cedex, France;

b C.I.P.(Club Innovant et Pragmatique) Le Creusot Cedex, France;

Abstract

This paper presents a video-based solution for real time vehicle detection and counting system, using a surveillance camera mounted on a relatively high place to acquire the traffic video stream.

The two main methods applied in this system are: the adaptive background estimation and the Gaussian shadow elimination. The former allows a robust moving detection especially in complex scenes. The latter is based on color space HSV, which is able to deal with different size and intensity shadows. After these two operations, it obtains an image with moving vehicle extracted, and then operation counting is effected by a method called virtual detector.

Key Words: Video analysis, Adaptive background estimation, Shadow elimination, Virtual detector

1. Introduction

The vehicle detections are normally based on inductive loop detector, infrared detector, radar detector or video-based solution. Video-based systems offer many advantages compare to other techniques, such as more traffic information obtained, easily installed, scalable with progress in image processing techniques, etc. However, video-based solutions for outdoor environments are easily influenced by falling weather, variant illumination, moving shadows and small motion changes caused by wind or other factors.

A traditional method for real-time segmentation of moving objet in video based system is "background subtraction", or computing the error between a background model and current frame. The essential demand of background subtraction is to estimate a robust background to deal with change of illuminant or object. For this reason, a static background isn't adapted for real time road situations.

As regards the shadow detection, many works have been published. In [1] authors extract shadows using HSV color space, a band-pass filter is applied to the division image on channel "luminosity" between the current image and the background, tow low-pass filters are applied to the difference image on channel "hue" and "saturation" between the current image and the background. In [3] authors suggest that shadow region present same textural characteristics in each frame and in the corresponding background model. The shadow detection approach is improved by evaluating the similarity between little textured patches. In [8] authors perform shadow detection by locating the object-shadow boundary which can be approximated by one or more straight lines when the object is a vehicle or a polyhedron.

We observe that the existing shadow detection methods need for much processing time so they can't be
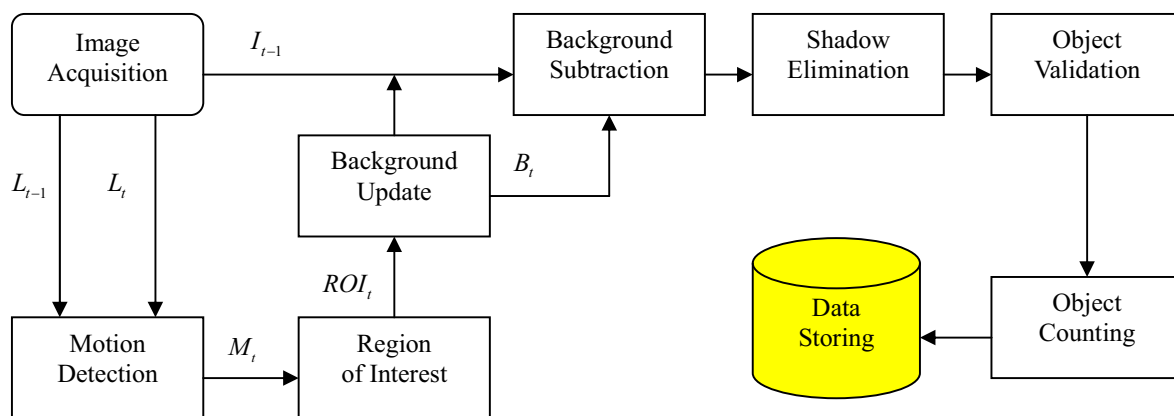

Figure 1 System Overview

IEEE computer society

applied by a low-cost equipment system.

Our goal is to build a scalable vehicle counting system that can achieve the following two demands:

- Robustness: the system can be self-adaptive to variant scenes (daytime, nighttime, overcast, shadow, ghost, wind)
- Performance: the algorithm requires low cost equipment usage so that the processing time can be reduced under a required time.

In this paper, we propose an pixel-oriented method of background subtraction or background learning named "adaptive background estimation[1]," which was successfully applied for a people counting system in indoor situation to detect and segment the moving objects. The method proposed in this paper, "Gaussian shadow elimination", implementing on "saturation" channel, is particularly a less computing method.

The paper is organized as follows: Section 2, 3, 4 describe the different steps of our proposed system (see Fig. 1). Experimental results are presented in Section 5. Finally, we conclude this article with conclusions.

2. Adaptive Background Estimation

The approach is improved by combining adaptive background generation with two-frame differencing algorithm.

We assume that the video stream is captured with a RGB 24 bits format. We use the luminance component of the RGB image to estimate the motion for each frame. We compute the luminance values for each pixel $L_t(x, y)$ at time t of image I_t according to the following equation:

$$L_t(x,y) = 0.2989 * R_t(x,y) + 0.5870 * G_t(x,y) + 0.1140 * B_t(x,y) \quad (1)$$

The first frame of the video stream I_0 is used for an initial background B_0. B_0 can be totally erroneous if there are moving objects present in the field of view but it will converge to a robust background after a few frames (period of learning time) thanks to the operation "Background Update".

A binary motion M_t is defined by a binaryzation operation between two frames $L_t(x, y)$ and $L_{t-1}(x, y)$.

We assume that, in most of the time, the pixel variations are due to sensor noise from the camera and light fluctuation, we can model the motion detection as follows:

$$M_t(x, y) = \begin{cases} 1 \text{ if } |L_t(x, y) - L_{t-1}(x, y)| \geq \mu_t + \sigma_t * k \\ 0 \text{ otherwise} \end{cases} \quad (2)$$

where μ_t and σ_t represent the mean and standard deviation of the absolute differences between the frames (L_t, L_{t-1}), k is the gain defined according to the result of experiment.

A pixel (x, y) is considered as moving if and only if its intensity value has changed between the current frame and the previous frame. Therefore, M_t highlights the pixels which contain moving objects or change illumination. However, this algorithm highlights the different edges of moving objects is unable to detect the whole object (see Fig.2-b) because the overlap between the body of the moving object in the two frames. As this binary motion is used in order to update the background model, an under-estimation of the binary motion (the body of the moving object is considered as background (see Fig.2-c)) will create difficulties to extract the moving object in segmentation process. We decided to create a region which contains all the conjoined highlight pixels for each object (see Fig.2-e). This operation should provide an over-estimation of the binary motion, but it is not grave for this application (see Fig.2-f).

The new background B_t is computed as a linear combination of the old background B_{t-1} and the current frame I_t in the following way:

$$B_t(x, y) = \begin{cases} \partial * B_{t-1}(x, y) + (1 - \partial) * I_t(x, y) & \text{if } ROI_t(x, y) = 0 \\ B_{t-1}(x, y) & \text{otherwise} \end{cases} \quad (3)$$

where ∂ is the learning rate and controls the background adaptation speeds.

$$\partial = \frac{\text{Number of all moving pixels}}{\text{Total Frame area in pixels}}$$
$$= \frac{\sum ROI_t}{Area(I_t)} \quad (4)$$

3. SEGMENTATION

3.1 Background subtraction

Once we have a robust background model, we can use a method called background subtraction to segment each frame into foreground and background objects[1],[3],[5].

A pixel would be part of the foreground, when its

value is different enough from its corresponding value in the background model. The main difficulty is to evaluate the distance of each pixel in a color frame (in RGB color space) to the corresponding background pixel. This evaluation allows the classification of all the current image pixels in two categories (foreground and background). In some situations, an oversimplification of the method (e.g. a binaryzation with a static threshold) may cause erroneous segmentation. We propose an automatic approach:

$$BS_t^c(x, y) = \left| B_t^c(x, y) - I_t^c \right|, \forall c \in \{r, g, b\} \quad (5)$$

where $BS_t^c(x, y)$ is result of the background subtraction. For each color channel $c\,(R, G, B)$, an absolute difference is performed between the current frame $I_t(x, y)$ and the background model $B_t(x, y)$.

Then a new binary mask FG_t is performed to extract the foreground regions and is evaluated as follows:

$$FG_t(x, y) = \begin{cases} 1 & \text{if } BS_t^r(x, y) > \tau^r \vee BS_t^g(x, y) > \tau^g \vee BS_t^b(x, y) > \tau^b \\ 0 & \text{otherwise} \end{cases}$$

$$(6)$$

where τ^r, τ^g and τ^b are the automatic thresholds for each channel c and are evaluated by analyzing the background-subtracted image BS_t. We determine τ_t^c by following equation:

$$\tau_t^c = \mu(BS_t^c) + k * \sigma(BS_t^c) \quad (7)$$

In most cases, a simple binaryzation is not sufficient to obtain a clear foreground. We use a morphological closing to fills the missing foreground pixels and a morphological opening to remove the small isolated foreground pixels.

3.2 Shadow elimination

Following the segmentation, we observe that foreground image contains the moving objects and
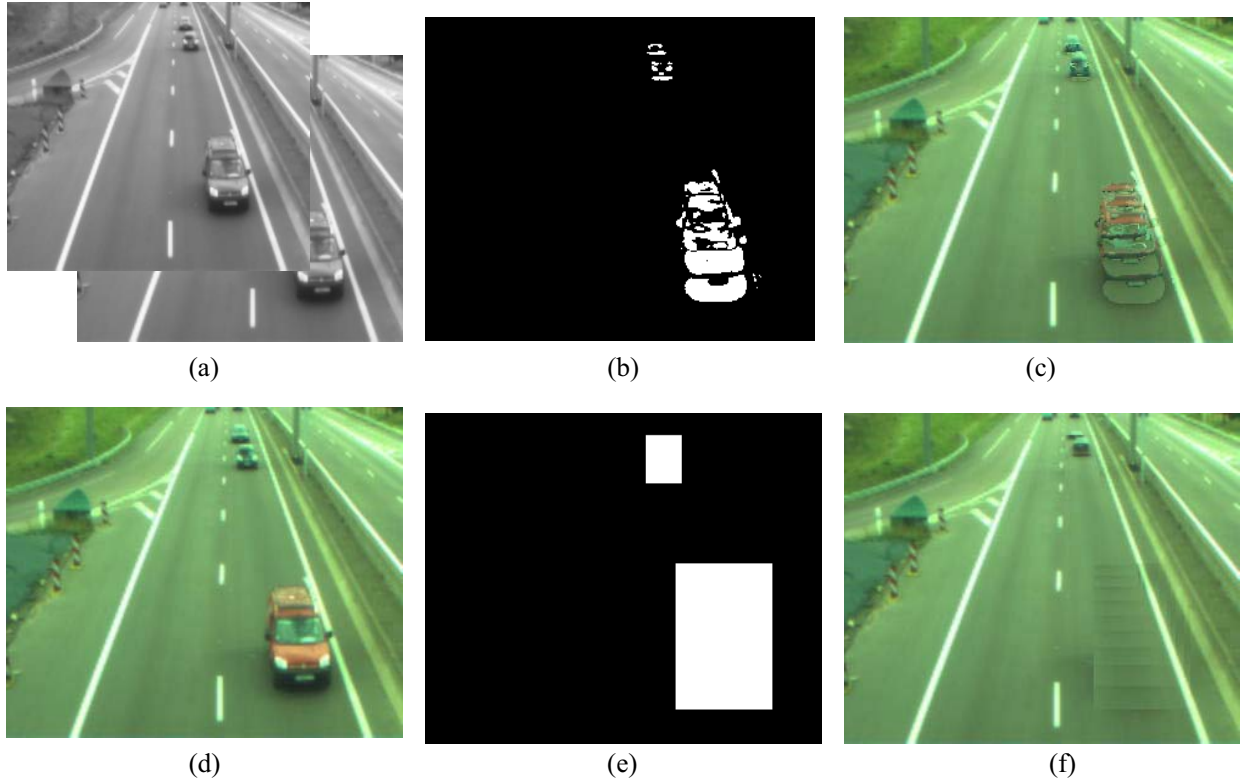

Figure 2:   (a) Two consecutive luminance frame; (b) Binary motion mask computed using the frame differencing algorithm; (c) Background model updated with the binary motion mask; (d) Current image; (e) ROI mask of the binary motion mask; (f) Background model updated with the ROI mask.

shadows. These shadows could sometimes fill between two objects or more ones. Consequently it may cause erroneous vehicle counting.

In the HSV color space, a shadow will not appear in

the saturation channel because the shadows' saturation is nearing roads' (see Fig.3-d). Analyzing the background model's saturation, we note that the saturation of

background model has a Gaussian distribution (see Fig. 3-f). So, we propose a band-stop filter to remove the shadows.

A foreground improved (without shadow) FGI_t is obtained by following equation:

$$FGI_t(x,y) = \begin{cases} 1 & \text{if } I_t^s(x,y) > \partial \vee I_t^s(x,y) < \beta \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where
$$\partial = \mu(B_t^s) - k * \sigma(B_t^s)$$
$$\beta = \mu(B_t^s) + k * \sigma(B_t^s)$$

where B_t^s is saturation of the background model.

This method is able to deal with different size and intensity shadows. Although some pixels of the object back to the light will be removed by this operation (see Fig.3-i), it will always keep the main form of the object.
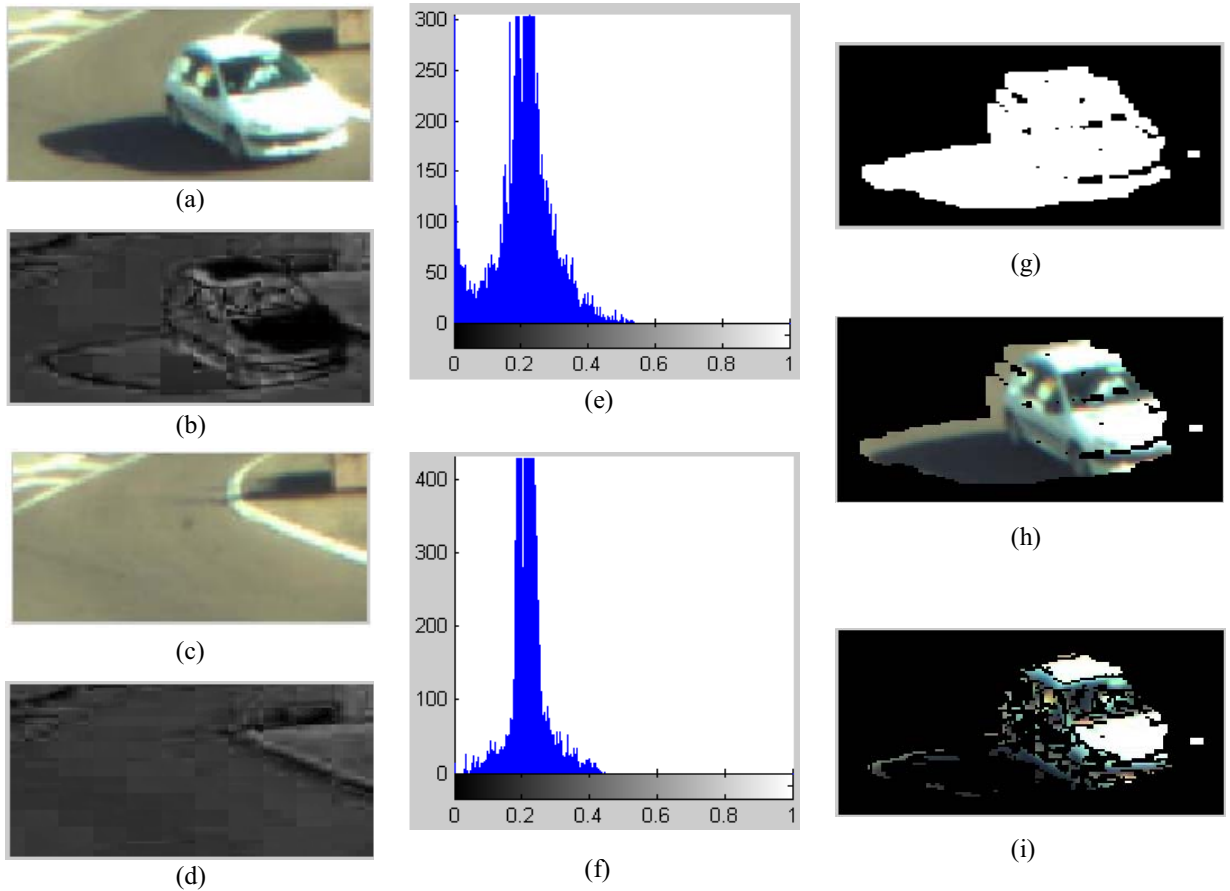

Figure 3: (a) Current frame; (b) Saturation of the current frame; (c) Current background model; (d) Saturation of the current background model; (e) Histogram of the saturation of the current frame; (f) Histogram of the saturation of the current background model; (g) Binary mask FG_t obtained by Background Subtraction; (f) Foreground before shadows elimination; (i) Foreground after shadow elimination.

4. COUNTING

The main object of this part is to count and register the vehicle flow for each lane. We propose a method called "virtual detector" to realize this function. This method is based on the foreground improved.

A foreground improved image include only the moving objects, we can select a rectangle region called virtual detector on every lane and compute the mean of these rectangle regions for each frame. If there is no vehicle cross the rectangle region, its mean should be zero. In the other side, if a vehicle is crossing a rectangle region, we obtain a no-zero mean value for this rectangle region until the vehicle leaves this region (see Fig.4). The variation of the mean value can be transformed into a binary signal. Therefore, we count the flow of vehicle according to the mean of each virtual detector.

We define the presence of the vehicle P_t^c for detector c by the equation (9):

$$P_t^c = \begin{cases} 1 & \text{if } \text{mean}(\text{detector}^c) > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$
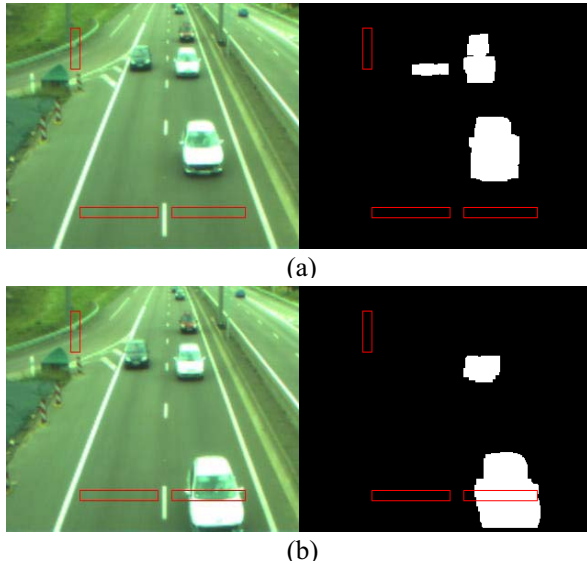

(a)

(b)

Figure 4: (a) No vehicle traverse the virtual detector; (b) A vehicle traverse the virtual detector;

To avoid the erroneous counting caused by vehicle overlapping in image, we propose to mount the video camera not aside the road, but above.

5. EXPERIMENTAL RESULTS

Four different traffic scenes are selected to test the proposed method. The scenes are acquired with different hours in the day. The camera is fixed with the resolution set at 320x240. The four scenes are as follows:

Scene 1: daytime, obvious shadows, camera faces the headlight.
Scene 2: daytime, cloudy day, non obvious shadows, camera faces the tail-light.
Scene 3: daytime, cloudy day, non obvious shadows, camera faces the headlight.
Scene 4: nighttime, non street lamp, camera faces the headlight.

The programs are written in C++ and executed on a standard PC. The results of accuracy rate are shown in following table:

	Scene 1	Scene 2	Scene 3	Scene 4
Accuracy rate (%)	90.41	70.31	98.39	80

We observe that this system will depend on, to certain extent the visual angle and the choice of the position of virtual detector. When camera faces the headlight, we get a lot accuracy rate then others position.

6. CONCLUSIONS

In this article, the authors present a method to detect and count vehicle in complex scenes. Our proposed system resolves relatively well various troublesome situations such as shadows and ghosts. Shadows are automatically removed. This system will totally remove the ghost created by the starting moving object with short period of time. Actually, our version is not able to recognize vehicle types (lorry-driver, car, motorcycle). The second improvement could be implementing vehicle classification for each detector in order to improve the statistic function.

REFERENCES
[1] D. Lefloch, F. A. Cheikh, J.Y. Hardeberg, P. Gouton & R. Pcot-Clemente, "Real-time people counting system using a single video camera", *Real-Time Image Processing,* SPIE, 2008, pp. 681109-681109-12.

[2] X-j. Tan, J. Li & C. Liu, "A video-based real-time vehicle detection method by classified background learning", World Transactions on Engineering and Technology Education Vol.6, No.1, 2007

[3] A. Leone and C. Distante, "Shadow detection for moving objects based on texture analysis", *Pattern Recognition* **40**, pp. 1222–1233, April 2007.

[4] T. Kim, S. Lee, and J. Paik, "Evolutionary algorithm-based background generation for robust object detection", *Lectures Notes in Computer Science,* LNCS, Springer Berlin, 2006, **4113**, pp. 542–552.

[5] S. Cheung and C. Kamath, "Robust Techniques for Background Subtraction in Urban Traffic Video", *Video Communications and Image Processing*, SPIE Electronic Imaging, San Jose, January 2004.

[6] C. Stauffer, and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking", *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Fort Collins, USA, 2, 246-252 (1999).

[7] C. Zhang, S. C. Chen, M. L. Shyu & S. Peeta, "Adaptive background learning for vehicle detection and spatio-temporal tracking", ICICS-PCM, 2003

[8] Angie W. K. So, Kwan-Yee Kenneth Wong, Ronald H. Y. Chung, Francis Y. L, "Shadow detection for vehicles by locating the object-shadow boundary", SIP 2005: 315-319.