A Vehicle Counting Algorithm Using Foreground Detection in Traffic Videos

Linbo Zhang[1], Feng Wang, Ming Hu, Lei Shi and Long Liang

Abstract. Vehicle counting in traffic videos plays an essential role in traffic surveillance system. It can provide some important traffic information such as congestion level and statistical analysis of traffic flow. In this paper, a novel vehicle counting strategy, which is based on foreground detection algorithm in videos, is proposed. Two advantages are contributed by our method. First, it is cost-saving as no additional hardware devices are required. Second, the influence to performance caused by weather variations is eliminated by the use of foreground detection technique. Finally, We compare our method with the state-of-the-art methods, and the experimental results clearly demonstrate the effectiveness of our method on complex real world traffic videos.

Keywords: Vehicle counting • traffic information • foreground detection.

1 Introduction

Vehicle counting in traffic videos is a challenging research issue in computer vision as unpredicted realistic conditions should be handled, such as uncontrolled illuminations, cast shadows and visual occlusion. Accurate vehicle counting can provide some important traffic information such as information about congestion level and traffic flow et al. It can also be treated as essential input for vehicle monitoring system, which can result in semantic traffic information, such as travel time prediction and driving behaviour analysis. If such information could be used efficiently, the problem about traffic congestion and environmental pollution can be solved more easily. In addition, the information is also useful to road safety [1].

In the past few years, a lot of strategies have been proposed to address this problem. The cameras optical axis are set to be perpendicular to the road plane by some approaches. However, the range and amount of visual information available

[1] L. Zhang (✉)
China Academy of Transportation Sciences, Beijing, China
e-mail: lumbard.zhang@gmail.com

is severely reduced by this configuration, as compared with a perspective camera configuration [2]. An alternative solution is to use more than one cameras (i.e., stereo vision) [3] or use nonvisual sensors, such as inductive loops or laser sensors, in as supplementary means to aid the vision system [4]. However, additional work caused by the installation and signal-processing complexities of nonvisual sensors makes the generalization progress more difficult [1]. As a result, the methods based on computer vision and image processing are attracting more and more attention due to its low cost and ease of use.

In general, foreground detection methods can be categorized into temporal differencing, optical flow and background modeling. Temporal differencing [5] is usually inaccurate and often leads to disintegration in the detected object, such as holes. Optical flow methods often suffer a heavy computation load when adopted in real-time applications. Generally, the background model [6] [7] can overcome the weaknesses above. However, the dominant background modeling methods [10], [11], [12] adopt a single feature (color or texture feature) to solve these problems, while the performances are usually unsatisfactory when handling complex scenes. One of the most popular methods is Gaussian Mixture Models (GMM) [12] which presents pixels with color feature. Despite its success, the method uses only pixel color or intensity information to detect foreground objects, which may fail when foreground objects have similar color or intensity to the background. Due to the advantages of texture feature [13], [14] in many applications, Heikkia et al. [15] developed a novel background modeling method, in which region histograms of LBP were calculated and the background was extracted based on these region histograms. However, background modeling methods using texture feature cannot detect changes in sufficiently large uniform regions if the foreground is also uniform. Thus, we utilize multi-scale fusion of texture and color for background modeling [7], which is robust to noise and illumination variations and can suppress the moving soft shadows.

The rest of this paper is organized as below. Section 2 presents our proposed approach. Section 3 demonstrates the experimental results are more accurate than the state-of-the-art methods on the real world traffic videos. In section 4, we conclude this paper.

2 Approach

In this section, details of the proposed Vehicle Counting Algorithm is presented, which contains traffic videos capture, foreground detection and vehicle counting.

2.1 Capture traffic videos

The traffic videos are captured by CCD TV, saved using computers hard disk and used as the input of our algorithm. The frame rate of the captured videos are 50 frames per second, each frame of which is of the size 640×480..

2.2 Foreground detection

In this section, our approach using multi-scale fusion of texture and color for foreground detection is presented [7].

1) Texture feature:

Scale-invariant Center-symmetric Local Ternary Pattern, SCLTP for short, is selected as the texture feature. Given any pixel location (*r, c*), it can be encoded as:

$$SCLTP(r,c,R,N,\tau) = \bigoplus_{i=0}^{N/2-1} S_\tau(n_i, n_{i+N/2}) \tag{1}$$

$$S_\tau(n_i,n_j) = \begin{cases} 01, & if\ n_i > (1+\tau)n_j \\ 10, & if\ n_i < (1+\tau)n_j \\ 00, & otherwise \end{cases} \tag{2}$$

where n_i and $n_{i+N/2}$ correspond to the gray values of center-symmetric pairs of N neighboring pixels equally spaced on a circle of radius R, $\oplus$ denotes concatenation operator of binary strings, and τ is a scale factor indicating the comparing range. Each comparison can generate one of the three values in Eq. 2, so the SCLTP operator encodes 2 bits.

There are three advantages for the SCLTP operator in foreground/background segmentation. First, it is compute efficient, which has only one more comparison step than LBP operator. Second, the SCLTP operator is robust to both noise and illumination by introducing a scale factor. Moreover, this scale factor make the operator robust to soft shadow, which is based on the fact that soft shadow is darker than the local background region. Third, the SCLTP operator can represent more texture information with less bits. Concretely, as only center-symmetric pairs of pixels are compared, only 8 bits are required to represent a pixel with all its 8 neighboring pixels, while 16 bits is used in SILTP operator.

2) Color feature:

Photometric invariant color is selected as the color feature. The three components in RGB space are first normalized, and a shadow invariant color distance is utilized to compare an observed color value with a color mode. Through the observation, the pixel values are mostly distributed along the line going toward the

RGB origin point (0; 0; 0), when the illumination changes [7]. Thus, the difference between the observed color pixel and a background color pixel is measured by using their relative angle with respect to the origin in RGB color space.

3) Background modeling by fusing texture and color features:

Several statistical background models are applied, all of which are fusion the texture and color features. The background model $\mathbf{M}(x)$ at each pixel x is represented as:

$$\mathbf{M}(x) = \{K, \mathbf{m}_k(x)_{k=1,\dots,K}\} \tag{3}$$

where at most K models are applied. Each model $\mathbf{m}_k(x)$ consists of 5 components:

$$\mathbf{m}_k(x) = \{\mathrm{I}_k, \hat{I}_k, \tilde{I}_k, \mathrm{SCLTP}_k, \omega_k\} \tag{4}$$

where I_k denotes the average RGB vector $\mathrm{I}_k = (I_k^R, I_k^G, I_k^B)$ of model $\mathbf{m}_k(x)$. $\hat{I}_k$ and $\tilde{I}_k$ indicate the maximal and minimal RGB vectors model $\mathbf{m}_k(x)$ can achieve. SCLTP_k is the average of Scale-invariant Center-symmetric Local Ternary Pattern. $\omega_k \in [0,1]$ denotes the weight which indicates the probability that this pixel belongs to the background. Obviously, the texture and color features are combined into each model.

Given a new pixel x at time t, the SCLTP pattern is calculated and the RGB value are normalized firstly. Then the algorithm calculates the distance between the new pixel and each model at time t-1 which belongs to the background. Details about the distance $Dis(x, \mathbf{m}_k^{t-1})$ will be discussed in the next subsection. If the distance to the closest model is greater than a pre-defined threshold (i.e. $Dis(x, \mathbf{m}_k^{t-1}) > \mathrm{T_A}$), a new model is created with parameters $\{\mathbf{I}^t, \mathbf{I}^t, \mathbf{I}^t, \mathbf{SCLTP}^t, \omega_{init}\}$, where ω_{init} is a low valued initial weight. If the number of the models reaches K, the new model takes place the existing model which has the lowest weight, and otherwise the new model is added to the list of models. On the contrary, if the distance to the closest model is smaller than the threshold (i.e. $Dis(x, \mathbf{m}_k^{t-1}) \le \mathrm{T_A}$), this closest model is updated as follows:

$$\begin{aligned}
\tilde{I}_k^t &= \min(I^t, (1+\alpha)\tilde{I}_k^{t-1}) \\
\hat{I}_k^t &= \max(I^t, (1+\alpha)\hat{I}_k^{t-1}) \\
I_k^t &= (1-\beta)I_k^{t-1} + \beta I^t \\
SCLTP_k^t &= (1-\beta)SCLTP_k^{t-1} + \beta SCLTP^t \\
\omega_k^t &= (1-\gamma)\omega_k^{t-1} + \gamma
\end{aligned} \tag{5}$$

Meanwhile, only weight parameter ω is updated for other models. In Eq. 5, α is the learning rate for the update rule of the minimum and maximum color values. This makes the process robust to noise. β is the learning rate that controls the update of the color and texture features. γ manages the updating rate for the model weight.

After the update step, all the weights are normalized and sorted in descending order and the first B models are selected as background model:

$$B = \arg\min_{b}(\sum_{k=1}^{b} \omega_k > T_C) \qquad (6)$$

where T_C is a threshold between 0 and 1 which indicates how much proportion of the data should be included in the background model. Then, the pixel can be classified as foreground or background using these models. Let $\overline{D}$ denote the closest distance between the pixel and B background models. Decisions of foreground and background segmentation can be made by thresholding $\overline{D}$ with a predefined parameter T_F .

4) Texture and color distance: The distance of texture andcolor features is defined as:

$$Dis(x, \mathbf{m}_k^{t-1}) = \varepsilon\, Dis_T(\mathbf{SCLTP}_k^{t-1}(x), \mathbf{SCLTP}^{t}(x)) + (1-\varepsilon)\mathrm{Dis}_{\mathrm{C}}(\mathbf{I}_k^{t-1}(\mathrm{x}), \mathbf{I}^{t}(x)) \qquad (7)$$

where Dis_T and Dis_C denote texture distance and color distance respectively. ε is a parameter balancing texture distance and color distance which is empirically set to 0.7. The smaller distance $Dis(x, \mathbf{m}_k^{t-1})$ is, the better pixel matches the model. The texture distance is defined as:

$$Dis_T(\mathbf{SCLTP}_1, \mathbf{SCLTP}_2) = \frac{1}{P}\sum_{p=1}^{P} D(\mathbf{SCLTP}_1^{p}, \mathbf{SCLTP}_2^{p}) \qquad (8)$$

where P is the total number of involved neighbors, and D(,) is defined as:

$$D(x, y) = \begin{cases} 0, if \mid x - y \mid < T_D \\ 1, \quad otherwise \end{cases} \qquad (9)$$

Here, T_D is a threshold which is empirically set to 0.2. The color distance is defined as:

$$Dis_C(\mathbf{I}_k^{t-1}(x), \mathrm{I}^{t}(x)) = \max(\mathrm{D_A}(\mathbf{I}_k^{t-1}(x), \mathbf{I}^{t}(x)), \mathrm{D_R}(\mathbf{I}_k^{t-1}(x), \mathbf{I}^{t}(x))) \qquad (10)$$

where $\mathrm{D_A}$ and $\mathrm{D_R}$ are two distances based on relative angle and the color range. $\mathrm{D_A}$ is defined as:

$$\mathrm{D_A}(\mathbf{I}_k^{t-1}(x), \mathbf{I}^{t}(x)) = 1 - e^{-\max(0, \theta - \theta_n)} \qquad (11)$$

where θ is the angle between two RGB vectors $\mathbf{I}_k^{t-1}$ and $\mathbf{I}^{t}(x)$. θ_n is the maximum amount of noise that can be tolerated, which is empirically set. $\mathrm{D_R}$ is defined as:

$$\mathrm{D_R}(\mathbf{I}_k^{t-1}, \mathbf{I}^{t}) = \begin{cases} 0, if\ \tilde{I}_{k,l} < I^t < \hat{I}_{k,h} \\ 1, \qquad otherwise \end{cases} \qquad (12)$$

where $\tilde{I}_{k,l}$ =min(λ $\mathbf{I}_k^{t-1}$, $\tilde{I}_k^{t-1}$)(λ $\in$ [0.4, 0.7]), and $\hat{I}_{k,h}$ =max(η $\mathbf{I}_k^{t-1}$, $\hat{I}_k^{t-1}$)($\eta \in$[1, 1.2]). This equation indicates the pixel color values in the range of $\tilde{I}_{k,l}$ and $\hat{I}_{k,h}$.

5) Multi-scale strategy: After the above feature fusion process, the multi-scale strategy is also applied. The consideration of multi-scale strategy is based on two folds: 1. Using one scale, holes in foreground sometime appears when the foreground is large; 2. multi-scale analysis can provide more information for background modeling.

Specifically, the video frame is divided equally into size of $v \times v$ blocks which are non-overlapping. Then we calculate the mean value in each block, and down-sample the frame by *v*. In the down-sampled frame, we calculate the closest distance between the pixel and models which belong to the background as Eq. 7. Finally, the resulted closest distances are up-sampled bi-linearly to the original size. Afterwards, we adopt the average of closest distances at each scale as the final closest distance, which is defined as:

$$Dis_{final}(x,\mathbf{m}_k^{t-1})=\frac{1}{Z}\sum_{z=1}^{Z} DisV_z(x,\mathbf{m}_k^{t-1}) \qquad (13)$$

where $DisV_z$ *(x,* $\mathbf{m}_k^{t-1}$*)* is the closest distance of pixel *x* at *z*-th scale. Decisions of foreground and background segmentation can be made by thresholding Dis_{final}(*x,* $\mathbf{m}_k^{t-1}$) with the predefined parameter T_F. It is noticeable that in our multi-scale fusion algorithm, the texture feature SCLTP can be replaced by other unordered features (i.e. other variants of LBP), such as CLBP [8], and DLBP [9].

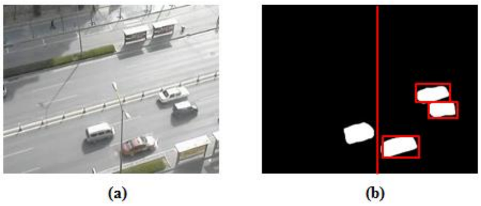

Fig. 1 The process of vehicle count

C. Vehicle count

After the vehicle detection process, the detected vehicles in the traffic videos are counted by setting a virtual line. An example of vehicle detection results and the virtual line are shown in Fig. 1. The conditions of virtual line are 1 or 0, where 1 represents there is a vehicle passing through the virtual line and 0 otherwise. The conditions of virtual line are ensured by analysis the variance of pixels on it. To suppress the noise and high frequency component, a Low-pass filtering (LPF) is used through the virtual line and then the variance of pixels on the virtual line is computed. If the variance is bigger than a pre-defined threshold, there is a vehicle passing through the virtual line.

3 Experiment Results

In this section, the performance of our method is evaluated on real world traffic videos which contain both illumination variation and complex scenes. The size of all the frames are normalized to 240 × 180. The foreground detection technology fusing SCLTP and photometric invariant color is named "MFTC" in this section. For the multi-scale fusion technique, two scales selected and v is set to be 2 and 3. A set of consistent parameters are used for all experiments, that is R= 1, N = 8, and τ = 0.05 for the SCLTP operator; the maximum number of models K=3 for each pixel; T_A = 0.2, T_C = 0.7, T_F = 0.2 and $\alpha = \beta = \gamma$ = 0.005 for learning rate; T_D = 0.2 for the texture distance; λ = 0.5, η = 1.2 in the color distance computation. Our algorithm is carried out on a standard PC with 2.33GHz Intel Core(TM) 2 Duo CPU and it is capable of real-time processing.

Table 1 The experiment results

Methods	Weather	Experimental results	Manual count
Temporal differencing	Fine day	89	104
Our method	Fine day	**96**	
Temporal differencing	Cloudy day	72	97
Our method	Cloudy day	**85**	

The process of vehicle count is shown in Fig. 1. From the figure, we can see that when the vehicle passes through the virtual line, it will be counted. The counted vehicles are marked by red rectangles, which are shown in the right part of Fig. 1. It should be noted that since the foreground detection technique is selected, only the vehicles rather than shadow are detected.

Quantitative evaluations of our proposed algorithm is also performed in this paper as shown in **Table** 1 which show the results in fine day and cloudy day respectively. From the table, we can see that our method achieves better results than Temporal differencing method, even in the cloudy day.

4 Conclusion

In this paper, we propose a novel vehicle counting method which is based on foreground detection. There are two advantages for our method. One is cost-saving, as no additional hardware are required. The other is invariance to villainous weather situation, which is realized by utilizing foreground detection technique. Then the method is compared with the Temporal differencing method, and the experimental results demonstrated that it can achieve higher classification accuracy in complex real world traffic videos.

5 References

1. A. H. S. Lai, An effective methodology for visual traffic surveillance, Ph.D. dissertation, Univ. Hong Kong, Hong Kong, 2000.
2. R. Cucchiara, M. Piccardi and P. Mello, Image analysis and rule-based reasoning for a traffic monitoring system, IEEE Trans. Intell. Transp. Syst., vol. 1, no. 2, pp. 119C130, Jun. 2000.
3. K. Otsuka and N. Mukawa, Multiview occlusion analysis for tracking densely populated objects based on 2-D visual angles, in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog., Jul. 2004, vol. 1, pp. 90C97.
4. R. P. Ramachandran, G. Arr, C. Sun and S. G. Ritchie, A pattern recognition and feature fusion formulation for vehicle reidentification in intelligent transportation systems, in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., May 2002, vol. 4, pp. 3840C3843.
5. Z. Zhang, B. Xiao, C. Wang, W. Zhou and S. Liu, Contextual constrained independent component analysis based foreground detection for indoor surveillance, First Asian Conference on Pattern Recognition (ACPR), 2011, pp. 701-705.
6. Z. Zhang, B. Xiao, C. Wang, W. Zhou and S. Liu, Background modelling by exploring multiscale fusion of texture and intensity in complex scenes, First Asian Conference on Pattern Recognition (ACPR), 2011, pp. 402-406.
7. Z. Zhang, C. Wang, B. Xiao, S. Liu and W. Zhou, Multi-scale Fusion of Texture and Color for Background Modeling, in Proc. IEEE Int. Conf. Advanced Video and Signal-Based Surveillance (AVSS), 2012, pp. 154-159.
8. Z. Guo, L. Zhang and D. Zhang, A completed modeling of local binary pattern operator for texture classification, IEEE Trans. IP, vol. 19, no. 6, pp. 1657C1663, 2010.
9. S. Liao, M. Law and A. Chung, A completed modeling of local binary pattern operator for texture classification, IEEE Trans. IP, vol. 18, no. 5, pp. 1107C1118, 2009.
10. A. Elgammal, R. Duraiswami, D. Harwood and L. Davis, Background and foreground modeling using nonparametric kernel density estimation for visual surveillance, In Proceeding IEEE, vol. 90, no. 7, pp. 1151C1163, 2002.
11. K. Kim, T. Chalidabhongse, D. Harwood and L. Davis, Real-time foreground-background segmentation using codebook model, Real-time Imaging, vol. 11, no. 3, pp. 172C185, 2005.
12. C. Stauffer and W. Grimson, Adaptive background mixture models for real-time tracking, In Proceeding CVPR, pp. 637C663, 1999.
13. T. Ahonen, A. Hadid and M. Pietikainen, Face recognition with local binary patterns, In Proceeding ECCV, pp. 469C481, 2004.5. South J, Blass B (2001) The future of modern genomics. Blackwell, London
14. R. Gupta, H. Patil and A. Mittal, Robust order-based methods for feature description, In Proceeding CVPR, pp. 334C341, 2010.
15. M. Heikkila and M. Pietikainen, A texture-based method for modeling the background and detecting moving objects, IEEE Trans. PAMI, vol. 28, no. 4, pp. 657C662, 2006.