Applied Mechanics and Materials Vols 361-363 (2013) pp 2232-2235 *Online: 2013-08-08*
© (2013) Trans Tech Publications, Switzerland
doi:10.4028/www.scientific.net/AMM.361-363.2232

Vehicle Detection And Counting In Traffic Video Based On Opencv

Wen-Jun Wang [1, a], Meng Gao[2, b]

[1] Tianjin University, Tianjin 300072, China

[2] Tianjin University, Tianjin 300072, China

[a] wjwang@tju.edu.cn, [b]bayern8619@126.com

Keywords: digital image processing, ROI, background subtraction algorithm, vehicle detection, vehicle counting

Abstract. With the development of modern social economy, the number of vehicles in China is growing rapidly, so how to get real-time traffic parameters has a very important significance in using the limited road space, vehicle video detection method based on image processing develop rapidly. With the improvement of image processing technology and microprocessor performance, makes video-based traffic parameter detection using universal. This paper deals with the real-time traffic video, gets each frame, uses Gaussian filter denoising, marks the region of interest (ROI), apply background subtraction algorithm based on average method, get the binarization foreground image, set threshold to eliminate the moving objects whose area is too small, check the boundary of ROI to judge the moving vehicle and counting, get the results as parameters of the intelligent transportation.

Introduction

To achieve the traffic flow statistics on traffic video, we have to extract the moving objects, the main moving object detection methods include frame difference method, background subtraction method, optical flow method [1]. Frame difference method has strong adaptability to scene changes, but can't completely extract the entire associated pixel, and generates cavitation in the moving object, easily misses objects that have slow speed. Optical flow method can detect moving objects independently in the premise of moving camera, but its calculation method is quite complex and has poor performance in anti-noise.

Background subtraction method uses the grayscale difference between the corresponding pixel of the current frame image and the background image to detect vehicle. If the pixel difference is very large, we consider this pixel has an object through, contrary, if the pixel difference is very small, in a certain threshold range, we consider this pixel is background pixels. Put into mathematical formula as follows:

$$\mathrm{d} = |F_K(\mathrm{x},\mathrm{y}) - B_K(\mathrm{x},\mathrm{y})| \tag{1}$$

$$D_K(x,y) = \begin{cases} d, if\ d > T \\ 0, if\ d < T \end{cases} \tag{2}$$

In the formula, F_K is background subtraction image, B_K is the grayscale image of Background at the K-th frame, i is the number of frames (i=l,…,N), N is the total number of frames, T is the threshold.

According to the background model, background subtraction method can be divided into histogram method, Average method, Single Gaussian background model, hybrid Gaussian background model, Kalman filter method, HMM model method [2, 3, 4]. In this paper we choose the average method.

The average method stores values of the same pixel in continuous n frames and seek the average gray value, the average value is considered as the background value for the pixel. The update algorithm of average method is using the weighted average of current background and current frame to update background. The current frame may contain the foreground object, so before updating we should divide pixel into foreground and background, and use background pixel of the current frame to revise the corresponding pixels of the current background. In the average method, the formula of update the background model expresses as follows:

$$B_{i+1} = (1-\beta)B_i + \beta F_i \tag{3}$$

B_i is the background of I-th frame, F_i is the background of the current frame, β is the update rate, if the background changes slowly, β should be made smaller, on the contrary, β should be made bigger. Under normal circumstances, select β as 0.005 can achieve a better update effect.

The rest of the paper is organized as follows: Section 2 introduces the method of detect and count vehicle in the traffic video, result of our method is in Section 3. Conclusion concludes the paper.

Methodology

The algorithm process is shown in Fig. 1.

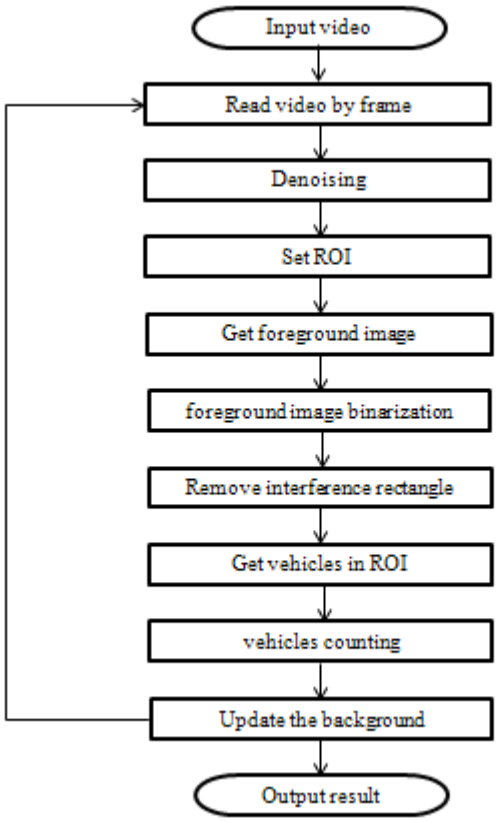

Fig. 1 The process of vehicle counting algorithm

Denoising. The Gaussian filter is a linear smoothing filter according to the shape of the Gaussian function to select the weight value,it has a good result when removes noise followed a normal distribution[5]. One-dimensional zero-mean Gaussian function is

$$g(x) = e^{-\frac{x^2}{2\sigma^2}} \quad (4)$$

σ determines the width of the Gaussian filter. For image process, usually use two-dimensional zero mean discrete Gaussian function to do the smoothing filter,function expressions are as follows:

$$g(i,j) = e^{-\frac{i^2-j^2}{2\sigma^2}} \quad (5)$$

OpenCV achieve this by the function void *cvSmooth*(const CvArr* src, CvArr* dst,int smoothtype,int param1, int param2, double param3)[6]. Src represents the input image,dst represents the output image, smoothtype:we choose CV_GAUSSIAN, for param1 we choose 5, for param2 we choose 0 represent that it's the same as param1, param3 correspond to standard deviation of the Gaussian parameters(sigma),choosing 0 as param3 means that standard deviation is calculated by the following nuclear size: *sigma=(n/2 - 1)*0.3 + 0.8*, in the function n=param1 corresponding to the level of nuclear, n=param2 corresponding to the vertical nuclear. The original image and the image after Gaussian filter is shown in Fig. 2.

(a) (b)

Fig. 2 (a) Original image (b) Image after Gaussian filter

Set ROI. The region of interest (ROI) is a subset of the selected image. ROI types include Polygon, Polyline, Point, Rectangle, and Ellipse Etc. In this paper, based on the characteristics of the vehicle we choose rectangle as ROI, and define two points' pt1 and pt2 as the lower-left corner point and the upper right corner.

OpenCV set the ROI by the function *cvRectangle*, the main parameters are the current frame, pt1, pt2, ROI boundary color, etc. The effect of set ROI in a video is shown in Fig. 3.

Fig. 3 ROI in the video

Get foreground image. OpenCV achieves this by the function void *cvAbsDiff* (const CvArr* src1, const CvArr* src2, CvArr* dst). Src1 represents the array of current frame, src2 represents the array of background image, dst represents the array of foreground image.

Foreground image binarization. An image include the target object, the background and noise, in order to extract the target object from the multi-value digital image, the most commonly used method is to set a threshold value T, the image is divided into two parts by threshold T: pixel values larger than T and smaller than T. Binary image can be obtained by the following method, set a threshold T, suppose that the input image is f (x, y), the output image is f'(x, y), then

$$f'(x,y) = \begin{cases} 1, if\ f(x,y) \geq T \\ 0, if\ f(x,y) < T \end{cases} \quad (6)$$

A frame, current background and the foreground after subtraction are shown in Fig. 4.

(a) (b) (c)

Fig. 4 (a) NO.17 frame (b) current background (c) the foreground image binarization

Get vehicles in ROI and counting. Through the above steps, when the car pass into the ROI , draw a rectangle to represent it, in contrast to the boundary of ROI, we can determine whether that car has passed through.

Update the background. In this paper, after processing each frame we update the average background. OpenCV achieve this by the function void *cvRunningAvg*(const CvArr * image,CvArr* acc,double alpha,const CvArr* mask=NULL). Image represents input image, acc represents accumulator, alpha represents the weights of image when update, mask represents operator mask.

The backgound when NO.6 frame and the backgound when NO.253 frame is shown in Fig. 5.

(a) (b)

Fig. 5 (a) background of NO.6 (b) background of NO.253

Experiment

The algorithm achieves by C++, opencv2.3, CPU frequency 1.86GHz, main memory 2.00GB, video image resolution 640*480, processes 25 images per second. The traffic videos come from Hexi District Government in Tianjin (for road monitor), we get the use rights for this experiment, select the middle of the road as ROI, the result is shown in Table 1:

Table 1: Result of vehicle counting

Vehicle	Count	Cross track	Overlapping	Accuracy (%)
26	22	3	1	84.61%

Conclusion

This paper presents an approach to count traffic flow in ROI of vertical lane, use image processing functions in opencv to handle the traffic video and output statistics, the experiments show that the method can get a good result. However, there is still some work to be done. When exist vehicles in situation of overlapping, the excessive speed and cross track, will affect the output results, we need to improve in the future work.

Acknowledgements

This research is supported by and all the IEMS Laboratory Consortium members. We would like to also express our gratitude to the Hexi District Government for providing traffic videos for our experiments.

References

[1] Zhiqiang Wei, Xiaopeng Ji and Peng Wang, Real-time moving object detection for video monitoring systems [J].Journal of Systems Engineering and Electronics, 2006, 17(4):731-736

[2] Dai Ke-xue, Li Guo-hui,Tu Dan, YUAN Jian, Prospects and Current Studies on Background Subtraction Techniques for Moving Objects Detection from Surveillance Video[J], Journal of Image and Graphics, 2006,11(7):919,928

[3] P. KaewTraKulPong, R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection [A]. In: Proceedings of the 2nd European Workshop on Advanced Video-Based Surveillance Systems[C], Kingston, UK, 2001: 149-158.

[4] Kaweepap Kongkittisan, Object Speed Detection from a Video Scene, Mahidol University, Bangkok Thailand, May 2003.

[5] S.J.Julier, A skewed approach to filtering, in Proc.AeroSense:12th Int.Symp.Aerospace/Defense Sensing, Simulation and Controls, vol.3373, Apr.1998, pp.54-65

[6] Information on http://www.opencv.org.cn/

Sustainable Cities Development and Environment Protection
10.4028/www.scientific.net/AMM.361-363

Vehicle Detection and Counting in Traffic Video Based on OpenCV
10.4028/www.scientific.net/AMM.361-363.2232