Processing of 18th International Conference on Systems, Signals and Image Processing (IWSSIP2011), Sarajevo, v.1, p.131-134, 2011

Robust Background Subtraction on Traffic Videos

E. Nunes, A. Conci
Instituto de Computação
Universidade Federal Fluminense
São Domingos, 24210-240 Niterói-RJ, Brazil
{aconci, enunes}@ic.uff.br

A. Sánchez
Departamento de Ciencias de la Computación
Universidad Rey Juan Carlos
28933 Móstoles (Madrid), Spain
angel.sanchez@urjc.es

***Abstract*—Background subtraction involves processing of a video sequence from a static camera to detect the foreground objects in all frames. This paper introduces a robust background subtraction technique, the Adaptive Local Threshold (ALT) algorithm, which it is based on the Approximate Median Filter (AMF). It has been applied to accurately extract the moving vehicles on complex weather traffic videos (i.e. fog and snow scenes). Experimental results have shown that the proposed algorithm produces similar qualitative detection results (based on the Jaccard coefficient) than AMF for the tested videos. Additionally, our method has the advantage of not needing any threshold parameter to detect the foreground targets.**

***Keywords-** adaptive threshold; background subtraction; segmentation; video-based traffic analysis.*

I. Introduction

Automatic traffic analysis based on video sequences consists on detecting and tracking the involved vehicles of a traffic scene with the goal to detect some types of high level events such us the density of vehicles in a road region, some types of traffic infractions or possible accidents [1]. The increasing social demands of mobility and safety in road transportation need automatic, economic and real-time solutions for reliable traffic flow analysis. Due to the importance of such automatic traffic video-based analysis and monitoring systems, these should also work on more complex illumination (i.e. night videos) and difficult weather conditions (i.e. scenes with rainfall, fog or snow) [2].

In general, these traffic videos are sequences of frames where the involved patterns (i.e. the moving vehicles) are highly correlated in time. Most of existing works for this problem use an approach based on scene segmentation followed by vehicle tracking [3][1]. Consequently, the vehicles in the analyzed scene are first detected using adaptive-background subtraction techniques [4][5]. Later, these vehicles can be tracked using different techniques like optical flow [6], Kalman filters [7] or particle filters [8]. Segmentation and tracking tasks become more difficult on realistic traffic scenes like possible vehicle congestions, variability of weather or illumination conditions. As a consequence, the achieved vehicle tracking results and further traffic event analysis along time will be highly dependent on robust scene segmentation.

Background subtraction techniques are used to extract the interest objects from the scene. In general, the potential applications of such techniques are related with Human-Computer Interfaces, real-time tracking or video-based surveillance [5]. These background subtraction algorithms should be computationally efficient, accurate in detection results and robust under different realistic conditions. Two common approaches are considered in the literature [4][5]. On one hand, the static and simplest one in which the background model is a unique and fixed image of the scene (i.e. without foreground targets) which is subtracted to all the video frames. On the other hand, the adaptive background approaches where the background image model varies along the frames and this model will adapt it to illumination changes and other scene variations. As the adaptive background techniques are much more robust than the use of a static background, many different adaptive approaches have been proposed [9]. Some examples of them are median filters, non-parametric models, Kalman filters or mixture of Gaussian models, among others. This paper introduces a new adaptive background subtraction algorithm, the Adaptive Local Threshold (ALT), which is compared to the Approximated Mean Filter (AMF) algorithm [10] on complex traffic video scenes representing adverse weather conditions.

The paper is organized as follows. Section 2 presents a high-level description of the proposed ALT method and explains each of its involved stages. Section 3 presents the achieved experimental results when comparing our approach to the ALT algorithm. Finally, Section 4 concludes the paper.

II. Adaptive Local Threshold Algorithm for Background Subtraction

A. High-level description of proposed Adaptive Local Thresholding (ALT).

Figure 1 outlines the stages of the proposed background subtraction method. Each function corresponds to a stage of the background detection method. These are detailed in the next subsection. The notation [.] in the algorithm means that the corresponding inputs or outputs of the functions are arrays of values.

Input: - Color RGB video [v]
Output: Corresponding binary video [v'] where for all frames the foreground pixels are labeled to one and the background pixels to zero
Algorithm:

```
[v'] := ∅;
for i = 1 to NumFrames ([v]) do
  [vg_i] := RGB2Gray ([v_i]);
  for j := 1 to NumPixels ([vg_i]) do
    [b_i] := ComputeApproxMedian ([vg_i]);
    [dif_i] := ComputeDifGraylevel ([vg_i], [b_i]);
    [h_i] := CreateDifHistogram ([dif_i]),
     th_i := FirstMinHist ([h_i]);
    [fg_i] := ExtractForeground ([vg_i], [b_i], th_i)
  end;
  [v']:= AddFrame ([v'], [fg_i])
end;
```

Figure 1. Pseudocode of the proposed ALT algorithm.

B. Involved stages in the ALT method.

This subsection summarizes the stages of the proposed background subtraction method for video segmentation when applied to RGB color videos.

- Transform the original color frames into gray-level frames.

This initial pre-processing is applied to all the frames of each tested video sequence. It consists on averaging the brightness values of the three color channels of the RGB space for each frame to produce a simpler gray-level video.

- Compute the approximate median to model the scene background.

At this stage, we applied an adaptive background model which is based on the approximate median filter (AMF) [10] due to the simplicity and accuracy of this algorithm. In the AMF, if a pixel in the current frame has a brightness value larger than the corresponding background pixel, the background pixel is incremented by one; otherwise, the corresponding background pixel is decremented by one. This is described by eq. (1):

$$b_{t+1}(x,y) = \begin{cases} b_t(x,y)+1, & \textit{if } vg_t(x,y) > b_t(x,y) \\ b_t(x,y)-1, & \textit{if } vg_t(x,y) < b_t(x,y) \end{cases} \quad (1)$$

where $b_{t+1}(x,y)$ and $b_t(x,y)$ represent the respective background model of pixel (x,y) at time $t+1$ and t, and $vg_t(x,y)$ is the brightness value of pixel (x,y) at frame t. Consequently, the scene background image will converge to an estimate value where approximately half the input pixels will be larger than the background, and half will be smaller than the background (i.e. the median value). The convergence time will depend on the frame rate and on the amount of movement in the scene.

- Compute the difference between the actual frame and the updated background model.

After an initial number of video frames k where the scene background is considered as stable, the following difference image is computed for each pixel (x,y) at frame t:

$$|vg_t(x,y) - b_t(x,y)| \quad (2)$$

- Create the corresponding difference histogram for each frame.

This histogram describes the probability differences between the current frame and the corresponding updated background model at each time.

- Compute the first significant minimum at the previous histogram as foreground threshold.

The local significant minimum values of this histogram (i.e. those ones whose absolute difference with their neighbor positions will be higher than a value) are the candidates to grey level threshold used to separate the foreground from the background pixels at each frame. We have chosen as threshold the first significant minimum and this value ensures that only will be classified as foreground pixels those ones whose difference with the computed median background model is high. In consequence, a reduced number of scene pixels will be considered as foreground as it is the case for the analyzed traffic videos where a unique camera is placed at a distance from the road.

- Detected foreground targets at each frame using the previous threshold.

Once computed the segmentation threshold, the intensity of each pixel at every frame is compared to this value and the pixel is set to one (and considered as foreground, as it is the case of the moving vehicles and pedestrians), when its intensity is higher than the computed threshold. Otherwise, the pixel is set to zero (and considered as background, as it is the case of the remaining scene elements).

III. Experimental Results

To evaluate the proposed ALT method for adaptive background detection, we used two public traffic video sequences which were created by the *Institut für Algorithmen und Kognitive Systeme* from Karlsruhe University (Germany). These videos can be downloaded from [11]. The first one called *dtneu_nebel* corresponds to a traffic sequence of 290 frames with heavy fog weather and showing the intersection of two streets. The second sequence called *dtneu_winter* contains 300 frames of a snowy road scene and it also corresponds to another intersection of streets. Both color videos were recorded at a spatial frame resolution of 768 x 576 (and initially stored as ppm files) using one stationary camera placed at a certain distance and height from the scene with a perspective.

The developed code was implemented in MATLAB using a standard Intel Pentium (R) Dual CPU T3200 2GHz with 3GB of memory.

Figures 2 and 3 respectively show some qualitative results achieved for the two test videos. The system requires from around 100 frames (or equivalently 4 video seconds) to stabilize the background model result. After that, for each of video we can accurately estimate the background. In these figures, we present an original random frame, its manual segmentation of foreground moving targets (i.e. vehicles) and the corresponding segmentation result produced by the proposed ALT method.

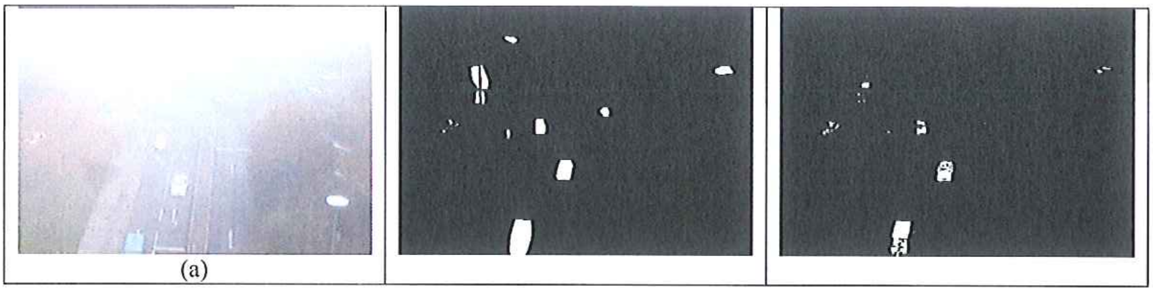

Figure 2. Foreground detection results for *dtneu_nebel* video: (a) original frame, (b) its manual segmentation and (c) result by ALT method.

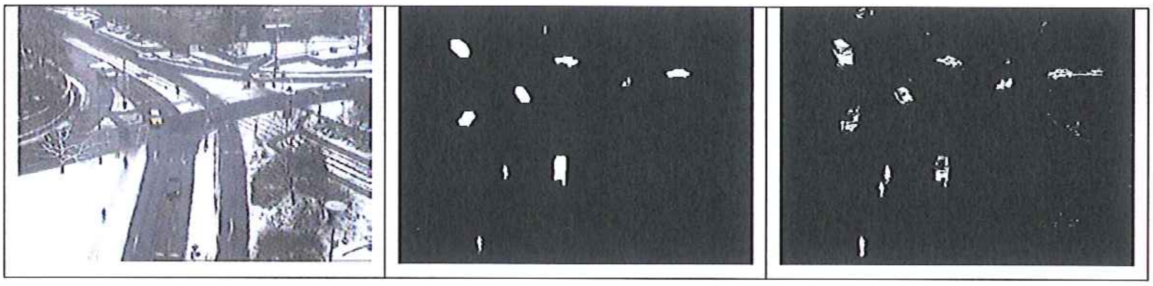

Figure 3. Foreground detection results for *dtneu_schnee* video: (a) original frame, (b) its manual segmentation and (c) result by ALT method.

Next, we compare the proposed ALT algorithm with the AMF for adaptive foreground detection for the two test videos. As a quantitative measure of quality, the Jaccard similarity coefficient J [12] was used. This coefficient determines the similarity between two sets, and is defined as the size of the intersection divided by the size of the union of these sets. This measure can be adapted to compare a pair of digital images i_1 and i_2 with spatial resolution $M\times N$ as follows:

$$J(i_1,i_2)=\frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\begin{cases}=1, & \text{if } i_1(x,y)=i_2(x,y)\\ =0, & \text{if } i_1(x,y)\neq i_2(x,y)\end{cases}}{M\times N} \quad (3)$$

The coefficient varies between zero when the compared images do not have any similarity and one when both images are identical. In this way, we can compare the similarity between the manual and the corresponding foreground detection produced by the compared ALT and AMF algorithms on the same videos. Tables 1 and 2 respectively show the results of these comparisons using this coefficient J on some sample frames of the *dtneu_nebel* and *dtneu_winter* videos. Note that the value of J depends on the threshold value used for the AMF algorithm, while this value is independent of the parameter for the ALT algorithm (since our method has been de-parameterized with respect to the threshold value). When comparing these results with the corresponding manual foreground detections, one can observe that for both algorithms and tested videos the produced Jaccard coefficient values on the analyzed frames were quite similar and very close to one.

In relation with the time required for AMF and ALT methods to extract the foreground pixels in all frames of the tested videos, Table 3 shows the corresponding results in seconds. As it is expected, AMF achieved smaller processing times than ALT. As this last method includes a background subtraction preprocessing similar to AMF, its corresponding processing time will be higher (around 40% in average for both videos).

IV. CONCLUSIONS

This paper presented the ALT algorithm, an adaptive background subtraction method which is developed using the AMF approach. The new method presents as advantages the fact of not requiring any threshold parameter to separate the foreground objects from the background. Moreover, it produces similar high-quality detection results to the AMF method. Our approach has been successfully tested on complex traffic videos that present complex weather conditions (i.e. foggy and snow scenes). As the computation times required to process video sequences by AMF are still higher (i.e. around 40%) than the corresponding ones required by the ALT method; a future work will consist in improving the efficiency of the proposed algorithm. Another interesting improvement is the automatic removal of "ghosts" regions [13].

TABLE 1. JACCARD COEFFICIENT VALUES FOR CERTAIN FRAMES OF DTNEU_NEBEL VIDEO USING AMF AND ALT ALGORITHMS.

Movie	dtneu_nebel.avi											
Method	AMF						ALT					
Threshold	Frame (Nr)					Average	Frame (Nr)					Average
	40	80	120	160	200		40	80	120	160	200	
10	0.9897	0.9875	0.9905	0.9925	0.9957	0.9912						
15	0.9890	0.9863	0.9899	0.9926	0.9961	0.9908						
20	0.9879	0.9851	0.9889	0.9922	0.9960	0.9900						
25	0.9868	0.9840	0.9879	0.9915	0.9960	0.9892						
30	0.9859	0.9831	0.9870	0.9908	0.9958	0.9885	0.9893	0.9861	0.9899	0.9926	0.9961	0.9908
35	0.9850	0.9823	0.9861	0.9903	0.9957	0.9879						
40	0.9845	0.9812	0.9853	0.9899	0.9953	0.9872						
45	0.9844	0.9804	0.9841	0.9895	0.9952	0.9867						
50	0.9843	0.9800	0.9825	0.9890	0.9951	0.9862						

TABLE 2. JACCARD COEFFICIENT VALUES FOR CERTAIN FRAMES OF DTNEU_WINTER VIDEO USING AMF AND ALT ALGORITHMS

Movie	dtneu_winter.avi											
Method	AMF						ALT					
Threshold	Frame (Nr)					Average	Frame (Nr)					Average
	40	80	120	160	200		40	80	120	160	200	
10	0.9882	0.9846	0.9842	0.9862	0.9850	0.9856						
15	0.9902	0.9869	0.9873	0.9893	0.9875	0.9882						
20	0.9909	0.9876	0.9884	0.9897	0.9877	0.9889						
25	0.9911	0.9880	0.9889	0.9895	0.9873	0.9890						
30	0.9910	0.9880	0.9892	0.9892	0.9868	0.9888	0.9901	0.9867	0.9873	0.9897	0.9869	0.9881
35	0.9912	0.9878	0.9895	0.9890	0.9864	0.9888						
40	0.9910	0.9878	0.9895	0.9890	0.9863	0.9887						
45	0.9910	0.9879	0.9893	0.9888	0.9864	0.9887						
50	0.9910	0.9879	0.9891	0.9886	0.9864	0.9886						

TABLE 3. TIME COMPARISON BETWEEN AMF AND ALT BACKGROUND SUBTRACTION METHODS.

Method	dtneu_nebel	dtneu_winter
AMF	16.3 s.	13.7 s.
ALT	22.9 s.	19.0 s.

ACKNOWLEDGMENT

This research has been partially supported by the Spanish project TIN2008-06890-C02-02. We acknowledge the support of the CAPES, as well.

REFERENCES

[1] V. Kastrinaki, M. Zervakis and K Kalaitzakis, "A survey of video processing techniques for traffic applications", Image and Vision Computing 21, pp. 359-381, 2003.

[2] J. Zhou, D. Gao and D. Zhang, "Moving Vehicle Detection for Automatic Traffic Monitoring", IEEE Trans. Vehicular Technology 56(1), pp. 51-59, 2007.

[3] E. Bas, M. Tekalp and F.S. Salman, "Automatic Vehicle Counting from Video for Traffic Flow Analysis", Proc. IEEE Intelligent Vehicles Symposium, v. I, pp. 392-397, 2007.

[4] S.-C. Cheung, and C. Kamath, "Robust techniques for background subtraction in urban traffic video", Proc. SPIE 5308, pp. 881-892, 2004.

[5] H. Wang and D. Suter, "A consensus-based method for tracking: Modelling background scenario and foreground appearance" Pattern recognition 40, pp. 1091.1105, 2007.

[6] B. Li and R Chellappa, "A generic approach to simultaneous tracking and verification in video", IEEE Trans. on Image Processing 11(5), pp. 530-544, 2002.

[7] J.W. Hsieh, S.-H. Yu, Y.-S. Chen and W.-F. Hu, "Automatic traffic surveillance systems for vehicle tracking and classification", Proc. IEEE Conf. on Intelligent Transportation Systems, v. 7, pp. 175-187, 2006.

[8] B. Ristic, S. Arulampalam and N. Gordon, "Beyond the Kalman filter: Particle filters for tracking applications", Artech House, 2004.

[9] M. Piccardi, "Background subtraction techniques: A review", Proc. Intl. Conf. on Systems, Man and Cybernetics, v.4, pp. 3099-3104, 2004.

[10] N. McFarlane and C. Schofield, "Segmentation and tracking of piglets in images", Machine Vision and Applications 8(3), pp. 187-193, 1995.

[11] Image Sequence Server, Universität Karlsruhe (Germany). URL: http://i21www.ira.uka.de/image_sequences. Last accessed: 25-04-2011.

[12] P. Sneath and R. Sokal, Numerical Taxonomy: The Principles and Practice of Numerical Classification, Freeman, San Francisco, 1973.

[13] R. Cucchiara, M. Piccardi and A. Prati, "Detecting Moving Objects, Ghosts, and Shadows inf Video Streams", IEEE Transactions on Pattern Analysis and Machine Intelligence 25, pp. 1337-1342, 2003.

IEEE Region 8

IEEE Bosnia and Herzegovina Section

IEEE Bosnia and Herzegovina Section - Chapter of the Communications Society

IEEE Bosnia and Herzegovina Section - Chapter of the Computer Society

IEEE Croatia Section

EURASIP - European Association for Signal Processing

2011 18th International Conference on Systems, Signals and Image Processing

PROCEEDINGS IWSSIP 2011

Edited by: Branka Zovko-Cihlar
Narcis Behlilović
Mesud Hadžialić

16-18 June 2011, Sarajevo, Bosnia and Herzegovina

ISBN: 978-9958-9966-1-0
IEEE Catalog number: CFP1155E-PRT