International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:6, No:1, 2012

Real Time Speed Estimation of Vehicles

Azhar Hussain, Kashif Shahzad, Chunming Tang

***Abstract*—**this paper gives a novel approach towards real-time speed estimation of multiple traffic vehicles using fuzzy logic and image processing techniques with proper arrangement of camera parameters. The described algorithm consists of several important steps. First, the background is estimated by computing median over time window of specific frames. Second, the foreground is extracted using fuzzy similarity approach (FSA) between estimated background pixels and the current frame pixels containing foreground and background. Third, the traffic lanes are divided into two parts for both direction vehicles for parallel processing. Finally, the speeds of vehicles are estimated by Maximum a Posterior Probability (MAP) estimator. True ground speed is determined by utilizing infrared sensors for three different vehicles and the results are compared to the proposed algorithm with an accuracy of ± 0.74 kmph.

***Keywords*—**Defuzzification, Fuzzy similarity approach, lane cropping, Maximum a Posterior Probability (MAP) estimator, Speed estimation

I. Introduction

WITH the immense growth of traffic congestion there is an utmost requirement of traffic monitoring system to manage such issues in an intelligent way. The motive for choosing computer vision lies in the fact that all major metropolitan cities in the world having traffic congestion problems also have setup to install static cameras at front and side view locations. In addition, the number of accidents due to over speeding is widespread and continues to increase. The need to monitor traffic conditions on major roadways or motorways is imperative as it provides urban planners and traffic engineers a great deal of fruitful information. Such as, over speeding detection, when to open new roads, and when to shunt traffic along alternate routes to alleviate traffic congestion. Real-time data applications offer an ability to increase safety and operational efficiency as well. Moreover utilizing existing traffic cameras infrastructure will give benefits like significant cost savings, increase efficiency in predicting the traffic model and quicker surveillance. Underlying induction loops are currently being utilized in most urban traffic systems, but this method to acquire the information of traffic flow includes cost and installation drawbacks.

Moreover vehicle classification and identification is more challenging at reduced speeds, because vehicles are liable to vary their speeds or positions while moving over the sensors.

Azhar Hussain is with Modern Digital Signal Processing research group in Harbin Engineering University, Heilongjiang (H.E.U) , Harbin, China.(phone: 008615146623286; e-mail: engrazr@gmail.com).

Kashif Shahzad is now with the Research and Development group from National University of Science and Technology, Islamabad, 40000 Pakistan, (e-mail:kashifshahzad31@yahoo.com).

Chunming Tang is head of Modern Digital Signal Processing research group in College of Information and Communication Engineering Department, H.E.U, Harbin ,150001 China, (e-mail: tangchunming@hrbeu.edu.cn).

The proposed algorithm can be a vital part in managing urban traffic using widely deployed smart cameras that are located at urban street intersections that are wirelessly networked together and with intermediate and centralized computing resources and that are interfaced with the traffic control network.

Numerous methods have been discussed and investigated for traffic management and monitoring [1-8]. Work by H. A. Rahim[1], using frame differencing technique to acquire foreground and then utilizing start frames and end frames over time to compute estimated speed produced good results. But it suffers accuracy when the vehicle varies its speed. A comparison of robust techniques for background subtraction in urban traffic video by Sen-Ching[3], has shown better performance results in case of adaptive median filter, in which foreground extraction is discussed in a traditional way. Proposed idea by Cristina Madurois[2], was presented to estimate the traffic intensity for each lane. This method does not require background estimation or even the identification and tracking of individual vehicles. It requires only the identification of each lane and the estimation of a bird eye view of the highway using a rectification method. However, this method could not solve individual vehicle speed estimation behavior and also lack accuracy.

Another approach for vehicle speed estimation by Gholam Ali Rezai [4], acquired in frequency domain using motion blur due to speed, suffers accuracy problems moreover it is computationally extensive. Method proposed by Lazaros Grammatikopoulos[5], where a window is defined for normalized cross-correlation among frames to allow vehicle tracking achieved good projection handling issues giving satisfactory estimated accuracy in vehicle speed of about ±3 kmph.

This paper describes a proposed approach by partitioning the traffic lanes into two parts depending on the direction of flow. Real time video is acquired by camera mounted for side view of lanes. The proposed algorithm first estimates the background by computing the median of values at each pixel location from real time stream of frames. The moving vehicles are extracting by using fuzzy similarity approach and α-cut is applied at an optimized threshold. Individual vehicle is tracked and speed is estimated by MAP estimation approach.

The paper is organized as follows. Section 2 addresses the test step, which specifies certain procedure and assumptions in order to ease vehicle tracking issues. Section 3 portrays a detailed description of the algorithm to estimate speed of the individual vehicle. Section 4 provides the speed estimation results that are compared to the average speed determined by portable infrared sensors. Section 5 presents conclusion.

II. Test Setup

The test setup consists of a side view camera aligned to the lane axis to nullify the projection errors while transforming real world to image space domain. Live stream from the traffic camera (in our experiment we used Sanyo HD) capturing the side view of (70m) Nantong Highway (near Harbin Engineering University China) is used as input feed. The 30fps video is scaled down to 240x320 resolutions to full fill the faster performance requirement in real time scenario. In the next step color domain is transformed to intensity domain for the same reason of computationally fast performance.

The proportional calibration between the pixels distance and real world lane distance is done to minimize the perspective ambiguities. Multiple vehicles are driven between two infrared sensor points. Time of travel is automatically computed by infrared sensor system to get the accurate average true speed. At the same time these vehicles were monitored in real time by a camera to compare the results.

Fig. 1 Video image of a Van and Taxi under test from a Sanyo HD AX1 camera mounted for surveillance of traffic vehicles

The two lanes are cropped and divided into two 20 x 320. Now each lane is processed in parallel through the same sets of blocks and in the end merged together to display whole image. A Pentium processor (2.5 GHz Dual core CPU) with 2 GB of RAM was utilized with an aiding environment of Simulink. Objects under test are assumed to be visible in whole scenario.

III. Algorithm Description

A. Fuzzy Similarity Approach

Since algorithm has to meet the real time processing requirements, therefore intensity images were utilized to cut the processing time. Following steps describe the performed algorithm to obtain the EF (Extracted Foreground).

1. Estimation of the background pixels B with adaptive median filter to process the issues related to light and shadow so that background updates itself after every adjustable period.
2. Performing similarity approach using (1), among B and current frame pixels I to get the similarity membership degree for each pixel.

$$Sim(B,I)=\frac{1}{(1+\frac{1}{(B-I)^2})} \quad (1)$$

This step produces an image that is low contrast and needs a less value of threshold but it also eliminates the intensity variation effects of the environment such as clouds.

3. In case the background pixel matches the foreground pixel value a set of fuzzy rules is defined, where the pixel belongs to foreground and must not be buried under threshold. Equation 2 captures the moving object by taking the absolute difference of I and I', where I' is one time step delayed frame pixel.

$$ADiff = I - I' \quad (2)$$

Distance_Diff is the distance in pixel units from ADiff pixel (threshold at T2=0.1) location to the arbitrary current pixel location. Fuzzy rule set is defined as show in Fig. 2.

1. If (Sim(B,I) is Low) and (Distance_Diff is Low) then (Output is High) (1)
2. If (Sim(B,I) is Low) and (Distance_Diff is High) then (Output is Low) (1)
3. If (Sim(B,I) is Low) and (Distance_Diff is Medium) then (Output is Medium) (1)
4. If (Sim(B,I) is Good) and (Distance_Diff is High) then (Output is Low) (1)
5. If (Sim(B,I) is Good) and (Distance_Diff is Low) then (Output is High) (1)
6. If (Sim(B,I) is Good) and (Distance_Diff is Medium) then (Output is Medium) (1)
7. If (Sim(B,I) is Excellent) and (Distance_Diff is High) then (Output is Low) (1)
8. If (Sim(B,I) is Excellent) and (Distance_Diff is Low) then (Output is High) (1)
9. If (Sim(B,I) is Excellent) and (Distance_Diff is Medium) then (Output is Medium) (1)

Fig. 2 Fuzzy Rule Set

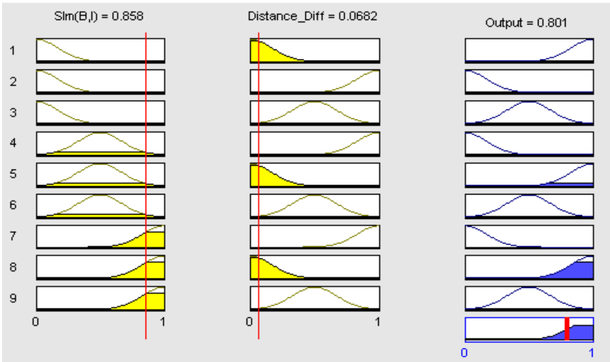

Fig. 3 Rule view for the extraction of foreground

Since the motion is along an axis, (2) provides strong intensity boundaries for a target object. Fuzzy rules set shown in Fig. 2 eliminates the holes created in case foreground pixel match that of current frame pixel. Rule view and surface view of FSA process are shown in Fig. 3 and Fig. 4 respectively.

4. Performing defuzzification of Output with an optimized α cut to obtain the foreground pixels.

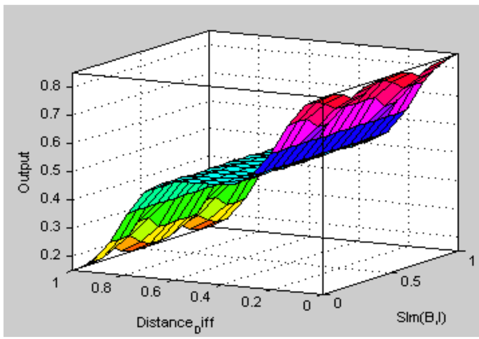

Fig. 4 Surface view of the fuzzy rules defined

The extracted foreground is then passed through the closing morphological operation to fill any non-retrieved foreground pixels or holes in neighborhood. After performing closing operation fine blobs are marked and classified as vehicles depending on their respective areas. The blob extraction process is shown for an arbitrary vehicle in Fig. 5.

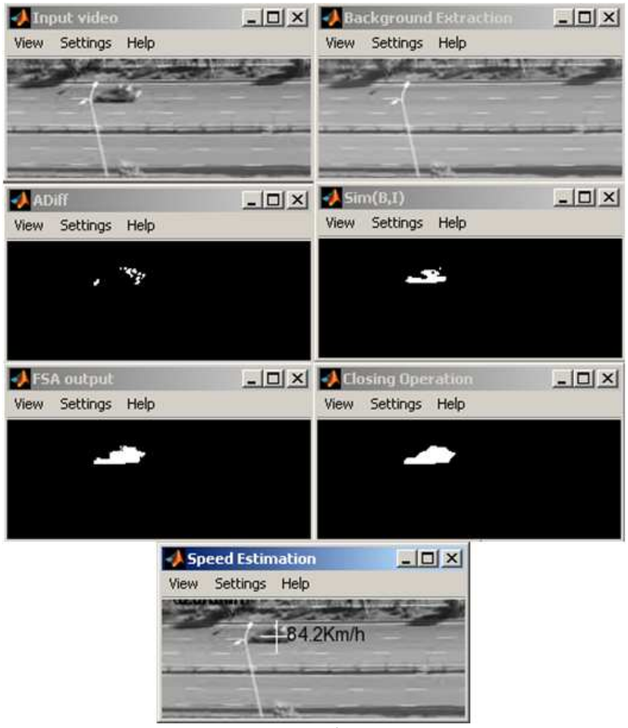

Fig. 5 Processing steps for blob extraction

Centroid of each blob is extracted using central moment mean method. In equation (4), Current frame blob center $C(X_c,Y_c)$ with a gap of every ten frames $G(X_g,Y_g)$ is taken as the unit distance U_d covered by the blob along the spatial domain using relation

$$U_d = \left| \sqrt{X_c^{\,2} + Y_c^{\,2}} - \sqrt{X_g^{\,2} + Y_g^{\,2}} \right| \tag{4}$$

U_d is scaled from pixel domain to the real world projection parameters domain in order to get the real world speed y.

B. MAP Estimation Approach

Since the noise related to the blob shape variation is Gaussian in nature, hence the mean and finally the so far acquired speed y, also has additive Gaussian noise n_i with S. In order to estimate the parameter speed S, from parametric space to estimation space S' via observation space $y(S,t)$, MAP is utilized[16]. MAP maximizes the a posteriori probability, which means most likely value of S and is given by (5).

$$\max_{\{S\}} f(S \mid y) = \max_{\{S\}} \left\{ \frac{f(y \mid S) f(S)}{f(y)} \right\} \tag{5}$$

It is quite clear that probability density function (p.d.f) of parameter S needs to be determined in order to maximize the expression (5). It is assumed that n_i is independent and identically distributed with $N(0,\sigma_n^2)$ moreover S is Gaussian random variable, independent of n_i with a *p.d.f.* $N(0, \sigma_S^2)$. We have adjustable K number of frames or samples for this speed estimation procedure as shown here.

$$f(y \mid S) = \prod_{i=0}^{K-1} \frac{1}{\sqrt{2\pi\sigma_n^{\,2}}} e^{-\frac{(y_i - S)^2}{2\sigma_n^{\,2}}}$$

$$f(S) = \frac{1}{\sqrt{2\pi\sigma_S^{\,2}}} e^{-\frac{S^2}{2\sigma_S^{\,2}}}$$

Since

$$f(S \mid y) = \left\{ \frac{f(y \mid S) f(S)}{f(y)} \right\}$$

$$= \left(\prod_{i=0}^{K-1} \frac{1}{\sqrt{2\pi\sigma_n^{\,2}}} \right) e^{-\sum_{i=0}^{K-1} \frac{1}{2\sigma_n^{\,2}} (y_i - S)^2} e^{-S^2/2\sigma_S^{\,2}} \frac{1}{f(y)\sqrt{2\pi}\sigma_S}$$

$$f(S \mid y) = q(y) e^{-(1/2\sigma^2)\left(S - \sigma^2/\sigma_n^{\,2} \sum_{i=0}^{K-1} y_i\right)^2}$$

Where

$$\sigma^2 = \frac{1}{(K\sigma_S^{\,2} + \sigma_n^{\,2})} \sigma_S^{\,2} \sigma_n^{\,2}$$

And $q(y)$ is only function of y.The best estimate of speed is the value where $f(S|y)$ gets the peak, which is obtained when

$$S = \sigma^2 / \sigma_n^{\,2} \sum_{i=0}^{K-1} y_i$$

Hence

$$S'_{MAP} = \sigma^2 / \sigma_n^{\,2} \sum_{i=0}^{K-1} y_i$$

This is solved to get

$$\sigma^2 = \frac{\sigma_S^{\,2}}{(\sigma_S^{\,2} + \sigma_n^{\,2}/K)} \frac{1}{K} \sum_{i=0}^{K-1} y_i$$

If $\sigma_S^{\,2} \gg \sigma_n^{\,2}/K$, then best estimate of speed S is given in 6 for a number of K samples.

$$S'_{MAP} \approx \frac{1}{K} \sum_{i=0}^{K-1} y_i \tag{6}$$

Therefore scaled real world speed is available at each frame count and then passed to the MAP estimation algorithm (6) to estimate the speed in real time. The reset time for running mean block is set at the K^{th} frame counts for more availability of the ongoing speed. Because when a vehicle enters into the region of interest its current center has some value, whereas the delay by

ISSN: 2517-9942
Vol:6, No:1, 2012

K^{th} frames has zero value. Therefore as the vehicle move into the frame forward the estimated value increments. After both reference points are available on the image the estimated speed is the valid one. Same thing happens at the end of frame position or when the object moves closer to the next border the estimated speed decrements.

As Simulink handles multiple elements in an array, and at the vanishing moment of each object, current place of element is decremented to lower order. A control is integrated for this purpose to check on the blob count increase and decrease and shift all element positions accordingly. Hence there should be no confusion for the position of unique vehicle. Both sides of the frame are padded with zeros for resetting the incoming and outgoing vehicles data from memory. Fig. 5 shows estimated background, *ADiff*, *Sim(B,I)*, FSA output and closing operation as described earlier. Closing operation using squared structure of neighborhood 5 is shown to get the fine blobs. Last part reveals tracked and estimated speed of vehicle. Low level processing is done to keep the processing requirements as according to the input frame rate. At the end fractional speed and tracked can be merged with colored input video as in Fig. 6.

Fig. 6 The marking and speed of three vehicles

IV. Results

Estimated speed S'_{MAP} for three different vehicles is shown in Fig. (4). Red, green and blue colours represent S'_{MAP} (in Kilometres per Hour) of Taxi, Van and Truck respectively. The straight lines show mean of the estimated speeds of corresponding vehicles over 70 frames. The true speeds are determined by infrared sensors and mentioned on the right hand side. It should be noted that all vehicles covered the same distance and also gradually reduced speeds due to the traffic signal at the road end. Overall error is calculated from the mean speeds and true speeds data.Variation in speed is also obvious in the results mentioned in Fig. 7-9.

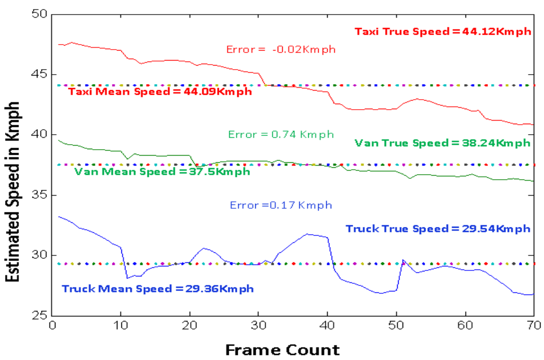

Fig. 7 Estimated speeds without FSA and closing operation with structuring element 5 neighborhoods

Results in Fig. 8 show remarkable improvement in accuracy as compare to Fig. 7, due to the use of 9 neighbourhood structuring element.

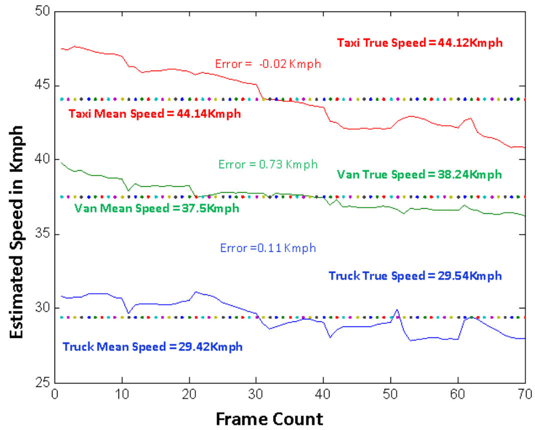

Fig. 8 Estimated speeds without FSA and closing operation with structuring element 9 neighborhoods

As closing operation has filled almost all holes in the blobs, it results in better center point accuracy. But it also has disadvantages, like increase in processing time and abnormal region growing near the edges.

FSA optimized the blob shape for the integrity of centroid position due to less structuring pixels count for filling. Hence Fig. 9 represents better performance in minimizing overall error. Occlusion prevention was applied by detecting blob area variation and hence center variation and mismatching. If the occlusion happens from start to the end of the area of interest then it would be considered as one vehicle. In case of occlusion, past sample of individual vehicles were utilized in order to predict the position as well as speed until the occlusion ends.

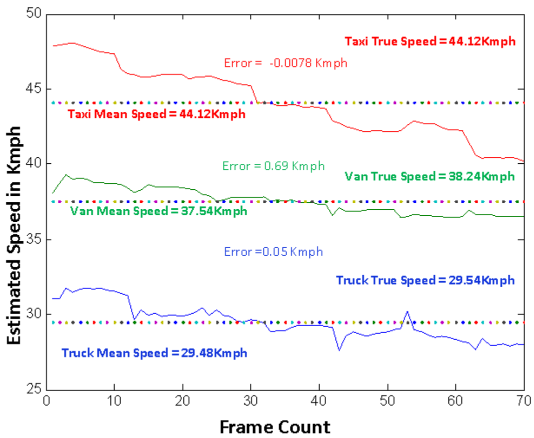

Fig. 9 Estimated speeds with FSA and closing operation with structuring element of 5 neighbourhoods

V. Conclusion

On the basis of so far work done, it is concluded that the speed estimation using computer vision has a lot of potential in real world sense. The proposed algorithm yielded an accuracy of estimated average speed of ± 0.74 kmph, which mainly depends on the blob features and robustness of a particular shape over the coming frames. Moreover, stability of camera frame rates is another optimizing factor. Using FSA in intensity level images has much benefit over traditional foreground extraction techniques because in this approach less filling of the extracted foreground is required. Otherwise estimated speed would become too much random because the centroid position is change due to the irregularity of the blob shape in next frames. Choosing the FSA for the similarity of blob shape in the upcoming frames will further stabilize the running mean variations of the speed estimation accuracy.

Acknowledgment

Research work was funded by College of Signal and Information Processing H.E.U Harbin, China. My deepest acknowledgement is to Dr. Chunming Tang for kind support and wisdom.

References

[1] H. A. Rahim, U. U. Sheikh, R. B. Ahmad and A. S. M. Zain ” Vehicle Velocity Estimation for Traffic Surveillance System” World Academy of Science Sep, 2010.

[2] Cristina Maduro, Katherine Batista and Jorge Batista “Estimating traffic intensity using profile images on rectified images” in IEEE Proceedings Sep ,2009.

[3] Sen-Ching, S. Cheung and Chandrika Kamath “Robust techniques for background subtraction in urban traffic video “Proceedings of the SPIE, Volume 5308, pp. 881-892 ,Jan 2004.

[4] Gholam ali rezai , Javad mohamadi “Vehicle Speed Estimation Based On The Image” in IEEE proceedings of 4th international conference Sciences of Electronic, Technologies of Information and Telecommunications March, 2007 .

[5] Lazaros Grammatikopoulos, George Karras, Elli Petsa (GR), “Automatic estimation of vehicle speed from uncalibrated video sequences” in Proceedings of international symposium on modern technologies, education and professional practice in geodesy and related fields sofia, 03 – 04 Nov, 2005

[6] P. KaewTraKulPong and R. Bowden, “An improved adaptive background mixture model for real-time tracking with shadow detection,” in Proceedings of the 2nd European Workshop on Advanced Video-Based Surveillance Systems, Sept. 2001.

[7] M. Harville, “A framework for high-level feedback to adaptive, per-pixel, mixture-of-Gaussian background models," in Proceedings of the Seventh European Conference on Computer Vision, Part III, pp. 543{60, (Copenhagen, Denmark), May 2002.

[8] D. R. Magee, “Tracking multiple vehicles using foreground, background, and motion models," in Proceedings of the Statistical Methods in Video Processing Workshop, pp. 12, (Copenhagen, Denmark), June 2002.

[9] R. Cucchiara, M. Piccardi, and A. Prati, “Detecting moving objects, ghosts, and shadows in video streams," IEEE Transactions on Pattern Analysis and Machine Intelligence 25, pp. 1342, Oct 2003.

[10] L. Fuentes and S. Velastin, “From tracking to advanced surveillance," in Proceedings of IEEE International Confer erence on Image Processing, (Barcelona, Spain), Sept 2003.

[11] T. N. Schoepflin and D. J. Dailey, “Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation,” IEEE Trans. on Intelligent Transportation Systems, vol. 4, no. 2,pp. 90-98, June 2003. 13. N. Oliver, B. Rosario, and A. Pentland, “A Bayesian computer vision system for modeling human interactions,"IEEE Transactions on Pattern Analysis and Machine Intelligence 22, pp. 831, Aug 2000.

[12] A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara, “Detecting moving shadows: algorithms and evaluation," IEEE Transactions on Pattern Analysis and Maching Intelligence 25, pp. 923, July 2003.

[13] B. Gloyer, H. Aghajan, K.-Y. Siu, and T. Kailath, “Video-based freeway monitoring system using recursive vehicle tracking," in Proceedings of SPIE, 2421, pp. 173, Feb 1995.

[14] Q. Zhou and J. Aggarwal, “Tracking and classifying moving objects from videos," in Proceedings of IEEE Workshop on Performance Evaluation of Tracking and Surveillance, 2001.

[15] K.-P. Karmann and A. Brandt, “Moving object recognition using and adaptive background memory," in Time-Varying Image Processing and Moving Object Recognition, V. Cappellini, ed., 2, pp. 307, Elsevier Science Publishers B.V., 1990.

[16] Detection theory: applications and digital signal processing, Ralph Hippenstiel, (CRC Press 2002).

[17] P. W. Power and J. A. Schoonees, “Understanding background mixture models for foreground segmentation," in Proceedings Image and Vision Computing New Zealand, pp. 271, (Auckland, New Zealand), Nov 2002.

[18] D.-S. Lee, J. Hull, and B. Erol, “A Bayesian framework for gaussian mixture background modeling," in Proceedings of IEEE International Confererence on Image Processing, (Barcelona, Spain), Sept 2003.