Progressive Background Image Generation of Surveillance Traffic Videos Based on a Temporal Histogram Ruled by a Reward/Penalty Function

Rensso V. H. Mora Colque
Departamento de Computação
Universidade Federal de Ouro Preto
Ouro Preto-Brazil
Email: soulchicano@gmail.com

Guillermo Cámara-Chávez
Departamento de Computação
Universidade Federal de Ouro Preto
Ouro Preto-Brazil
Email: gcamarac@gmail.com

***Abstract*—In most surveillance traffic systems the background image is used for segmenting the moving objects. In this paper, we propose a method for progressive and constant renewal of the background image of a video sequence. Our method is divided in three parts: frame difference to get an actual information of the video sequence; reward and penalty function that modifies a temporal histogram; and the background generation uses the information found in the histogram. The principal characteristic of the method is the constant regeneration of the histogram using neighborhood information of intensity levels, penalizing the occurrence of foreign objects and rewarding the values of static objects. For that, the histogram values are modified regionally and its variation correspond to a normal distribution. The principal advantages of our proposed method lies in the progressive increment of the background pixel intensity occurrences, the fast reaction to changes between foreground and background segments and the simplicity of the method that can be used in real time applications. At the end of this paper, experimental results demonstrated the feasibility of the proposed technique in different weather conditions and traffic situations.**

***Keywords*-Background image; frame difference; temporal intensity histogram; reward and penalty function**

I. Introduction

Vehicle detection is an important problem in many related applications such as: driver assistance systems, parking identification system or measurement of traffic parameters like vehicle count, speed, and flow [1]. Traffic management and information systems must rely on a system of sensors for estimating traffic parameters in real-time. Currently, the dominant technology for this purpose is that of magnetic loop detectors, which are buried underneath highways to count vehicles passing over them. New approaches [2] introduce the use of simple cameras, instead of the traditional sensors.

Video monitoring systems present advantages over traditional sensors, for example: simple cameras offer pretty much information, these include vehicle classifications, link travel times, lane changes, rapid accelerations or decelerations, queue lengths. Also cameras are less disruptive and less costly to install than loop detectors, which require digging up the road surface, and obviously the maintenance is cheaper [3] even for uncalibrated cameras [4] [5].

There are two important branches that study automatic vehicle detection systems, one is based on moving cameras [6] and the other one on static cameras [2]. The first branch studies everything related to cameras inside cars and are usually used on automatic parking systems and on computer assisted driving technology [7]. The second boarders systems that get information about the traffic scenes [8]. This work focuses on this second branch.

Automatic vehicle detection for static camera applications needs to segment the moving objects, and recognize the vehicle as distinct from the other motion objects. The most widely used approach for detection of moving vehicles captured with a static camera is based on background substraction where a reference frame of the stationary components in the scene is available (background model). There are many factors that influence the background model such as: time, weather conditions, number of vehicles, speed of the vehicles, shadows cast by buildings and clouds and changes in lighting [9]. Due to these changing environmental conditions, the background frame is required to be updated regularly.

There are several background updating techniques. Simple methods such as standard average [10], median [11] and simple difference [12] can provide acceptable accuracy but only in specific applications and controlled environment. Gaussian models [13], Meanshift [8], Kernel Density Estimation [14], Eigen-backgrounds [15], self-adaptive background substraction [16] and other complex techniques like virtual reality [17], optical flow [18], and energy values [19] offer good accuracy as well, but they require more memory and computational time. In real time applications the resources are limited. Histogram based approaches [20], [21] are methods with simple computation and small time-consuming.

In histogram approaches, the intensity with the maximum frequency in the histogram is treated as background intensity, because each intensity frequency in the histogram is proportional to its appearance probability. Intensities of temporary stop foreground will not be considered as the background intensity because its frequency is smaller than the background pixel. However, it is very common that more than one intensity level have the same maximum frequency. Thus, it is difficult to determine which is the intensity that belongs to the background. To overcome this problem, Wang *et al.* [22] proposed a method that considers the frequency of its neighboring levels instead of the frequency of each intensity level and K.-T. Song and J.-C Tai [23] proposed a reward/penalty voting function in order to reward a background intensity level and penalize a foreground intensity level. Although both methods try to overcome the problem of more than one maximum value, the problem persists in very slow moving and almost stationary vehicles.

Contributions: In this paper, we introduce a novel model based on reward/penalty voting function considering the neighboring intensity levels for background generation of traffic surveillance video. Our model aims to be adaptive to the conditions attached to city traffic, as well as being adaptable to the climatic conditions of the environment. For that, we use a histogram to record and trace the intensity changing of a pixel to determine the best background values. The histogram is updated considering reward/penalty function with a normal distribution that spreads the gain or the penalty over the neighboring intensities. Thus, the competition between maximum values of a same group (background or foreground) is punished and the competition between maximum values of different groups are rewarded. Moreover, our method generates the background image using simple mathematical operations. Also our method combines two techniques for updating the background and improve them.

This paper is organized in the following sequence: in Section II, we present our proposed approach, which is composed by partial background generation, temporal intensity histogram and the background generation steps. In Section III, we introduce the parts that generate a partial background. Temporal intensity histogram and reward/penalty function is described in Section IV. The final background generation is presented in Section V. In Section VI the experimental results are shown. Finally, in Section the conclusion are exposed.

II. Proposed Approach

The intensity of background scene is the most frequently value recorded at each pixel position. Therefore, the background can be determined by analyzing the intensity histogram of each pixel. Many factors influence in the accuracy of the generated background, like sensing variation and noise from image acquisition devices, where a set of intensities that belong to a same group (background or foreground) compete between them for updating the histogram. As a consequence, the maximum intensity that represent the group is low and changing. This also does not allow a fair competition between the groups because the background intensity interval is greater than the foreground intensity interval. We propose a method that overcome this problem by sharing the information between the intensity neighbors.

In Figure 1, we show our method that consists of three steps: partial background generation, temporal intensity histogram update and background generation. In the first step, a partial background Pb_i is generated by the difference of two consecutive frames (f_i and f_{i-1}). The next step updates the values in the temporal histogram (h_i) using a neighboring reward/penalty function. In the last step, a final background B_i is build from the histogram H_i table which uses information of temporal histogram.

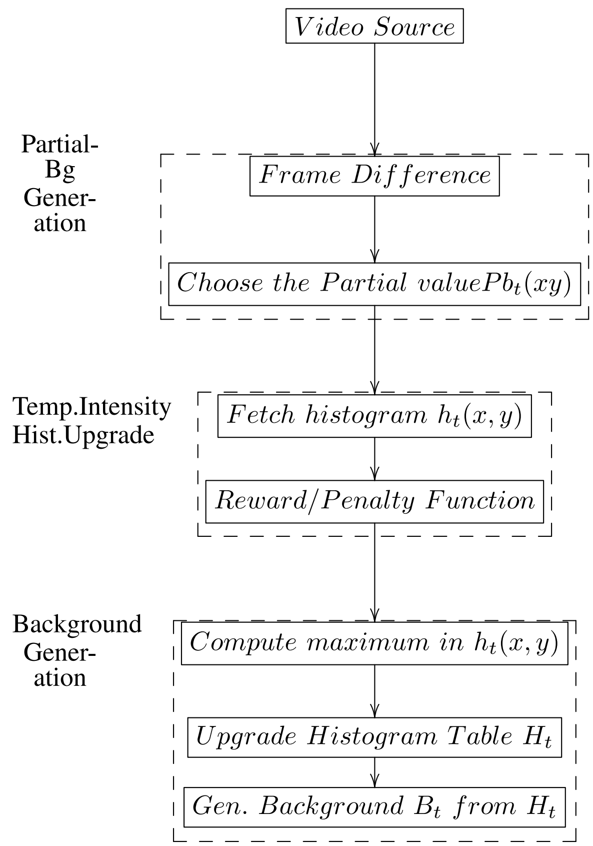


Fig. 1. Our proposed approach.

III. Partial Background Generation

Let V be a video source, and $f_i \in V$ an image frame at time i. Then, we define the partial background at time i as

$$Pb_i(p) = \begin{cases} \frac{f_i(p)+f_{i-1}(p)}{2} & if\ |f_i(p) - f_{i-1}(p)| < \gamma \\ -1 & otherwise \end{cases} . \quad (1)$$

where $f_i(p)$ is the pixel intensity at position (x, y) and γ is a minimum difference between corresponding pixels. The partial background is generated by a pairwise difference between frame $f_i(x, y)$ and $f_{i-1}(x, y)$. Chung *et al.* [24] suggest to use the $f_i(x, y)$, $f_{i-1}(x, y)$ or the average for the $Pb_i(x, y)$ computation. In our experiments, the average has a better performance. In Fig. 2, we can see the partial background image generated by the difference of two consecutive frames.

IV. Temporal Intensity Histogram Update

In Temporal Intensity Histogram (TIH) is registered the intensity frequency for each pixel, a maximum frequency is treated as a background intensity. In uninterrupted traffic

(a) Previous Frame (f_{i-1})

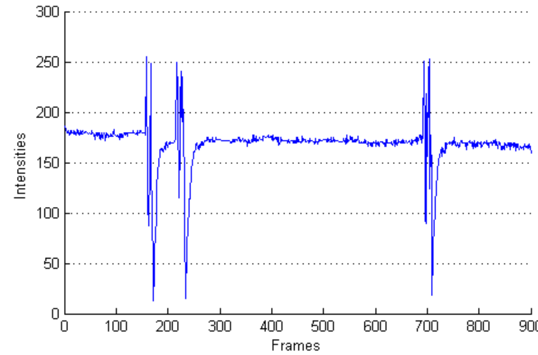
(b) Frame (f_i)

(c) Partial Bg

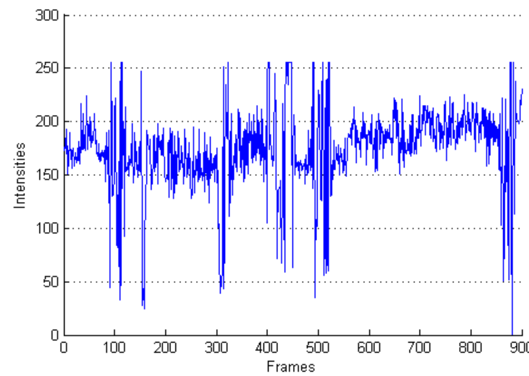
Fig. 2. Partial Background

flow this maximum frequency corresponds to the track, but in interrupted traffic flow the maximum could correspond to a stopped vehicle or in a very low motion. To solve this problem, Chung *et al.* [24] explore this pixel behavior and proposed a reward/penalty function to prevent the growth of values belonging to foreground. Their method registers in a pixel histogram the intensity occurrence in time, rewarding or penalizing the maximum pixel intensity occurrence according to the actual pixel value in the partial background. In Fig. 3(a) and Fig. 3(b), we show intensity variations of a pixel in two different situations: free flowing traffic and heavy traffic, respectively. In the first case, the color of the track remains almost stable with little variations. In the second case, the color of the track oscillate in a wider interval. Another characteristic we can see in Fig. 3(b) is that in heavy traffic the intensities changes a lot and very fast. Therefore, the TIH has to react as fast as possible.

The reward/penalty function is the responsible for variations on the TIH. Chung's method rewards or penalizes an unique



(a) Free Flowing Traffic



(b) Heavy Traffic

Fig. 3. Intensity values of a pixel.

intensity. This means that, when a background pixel changes from one intensity to another (foreground), has to be penalized until the TIH in the corresponding bin reaches a zero value. Then the bin in the TIH that corresponds to the foreground can increase. Thus, this method can not react as fast as the pixel intensities change. As a consequence of this problem, the method needs to start the generation of the background as clean as possible (low traffic).

Our approach improve this method introducing neighborhood information for reward and/or penalize a set of intensities belonging to two groups.

A. Reward/Penalty Function (RPF)

Our Reward/Penalty function is inspired on the group-based histogram proposed by K.-T. Song and J.-C. Tai [23]. Our RPF method generates a weighted accumulative frequency histogram where a weight is added to the maximum level intensity and its neighbors. This means that, if a representant of a group is reward or penalize, a positive/negative value is incremented to it and its neighbors.

Our function rewards the intensity level that remains unchangeable along the time. If there is a considerable variation between the actual partial background pixel value and the last background value registered, then the value associated to the partial background intensity will increase and the last growing intensity associated value is punished. In order to have a fair distribution, the value associated to the intensity pixel (reward or punish) is shared with its intensity level neighbors using a normal distribution. The Eq. 2 presents the formula to built the normal distribution.

$$m(r) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-\frac{1}{2}\left(\frac{r-\mu}{\sigma}\right)^2} \tag{2}$$

where $r \in [-\tau, \tau]$ is the neighborhood interval and σ is the variance. After some experiments, we suggest the following values: $\tau = 10$ and $\sigma = 0.9$. We use a normal distribution in order to spread the gain or penalty value with its neighbors, where closest neighbors receive more influence. Then $m(r)$ value is normalized between $[0, 1]$. In 4, we show an example of a function with normal distribution.

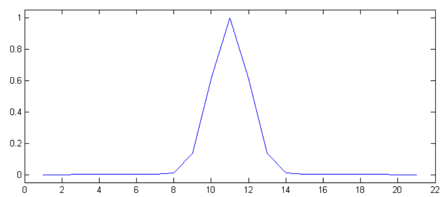


Fig. 4. Gaussian mask normalized between zero and one, $\sigma = 0.9$

We define reward and penalize rules according to:

- If a pixel intensity from the current partial background remains equal or almost equal to the last background generated, then this pixel is becoming stable. Therefore, this intensity occurrence deserves to be reward and the associated value is increased. This occurs because the intensity pixels is becoming stable and probably it belongs to the background.
- If the pixel intensity changes significantly, then the pixel is changing from background pixel to a moving foreground object pixel or vice versa. In this case, the value associated to the entering pixel is rewarded and the last growing value associated to pixel intensity is punished.
  We define these rules based on the frequency of the cars appearing over the track. In heavy traffic (stopped or almost stopped), the cars occlude most parts of the track. The slight increase of the associated value of a pixel intensity that represents a car is immediately punished by the next appearing vehicle in the video sequence. In uninterrupted traffic, the vehicles (foreground objects) occlude the track for a short period of time, thus the intensity that represent the track is punished. This does not greatly influence in the detection of background pixels since the period of time that the track appears is much greater than the vehicle time.
- All reward or punish of a pixel is always shared with its intensity neighbors.

Three constants φ_1, φ_2 and φ_3 are defined to reward or penalize an associated value of an intensity index. The high increment φ_1 is used to reward the intensity index that remains unchangeable, in the same group, in the sequence; the mid-increment φ_2 is used to reward slightly the incoming intensity index; and the penalty φ_3 is used to punish a changeable (unstable) intensity index. There is also an overgrowth bound k that prevents an associated pixel value to increase indefinitely. Next, we present the relationship between these values. For :

- Fluid traffic flow: $\varphi_1 > \varphi_3 > \varphi_2$ and $k = 20 \times v1$
- Slow traffic flow (traffic jam): $\varphi_1 > \varphi_2 > \varphi_3$ and $k = 15 \times v1$

In fluid traffic flow, the background appears more time, so incoming intensities that represent foreground objects (vehicles) are penalized while background intensities are rewarded. In slow traffic condition, there is the uncertainty if the incoming intensity belongs to a foreground object or to the background. Due to this uncertainty, the incoming intensity is penalized with a low value. Thus, in fluid traffic, φ_2 is smaller than φ_3 in order to reward track intensities because they appear more times than vehicle intensities. With this relation, the intensities of the vehicles are penalized with greater emphasis. In the case of heavy traffic, vehicles move with slow motion, φ_2 is greater than φ_3 in order to reward stationary segment. Since vehicles have different intensities, they penalize each other. Then the relation between φ_2 and φ_3 would change according to the scenario. Note that φ_1 is always greater than φ_2 and φ_3, this guarantees the growth of the stationary segments. The products of these parameters and the Gaussian model result in three vectors used to reward or penalize the value associated to the pixel intensity and its neighbors. In Eq. 3 we show the simple formula to compute the vectors.

$$m_j = m \times \varphi_j,\ where\ j\ \in [1, 2, 3]. \tag{3}$$

The histogram table H is defined as a matrix with the same dimension of the video frame, where each value corresponds to the intensity index of maximum counting value.

In the Algorithm 1, we show how to update the histograms using the Reward/Penalty function. The input values for the algorithm are: the partial background P_b, the histogram table H, the partial background histogram $h(p)$ of pixel p, the maximum difference within pixels of the same group, the last growing intensity pixel L_b and the penalty and reward vectors m_i are the reward and penalty vectors.

The algorithm computes the difference between the partial background $Pb(p)$ of pixel p and its corresponding histogram table $H(p)$. If the difference is less than the threshold δ (maximum acceptable difference) then the pixel remains in the same group, i.e., it remains as foreground or background. As a consequence, the intensity level in $Pb(p)$ and its neighbors are rewarded with m_1, and the intensity index is saved in $Lb(p)$. If the difference is bigger than δ, then the entering pixel index is rewarded and its index neighbors are reward with a lower value m_2 and the last growing intensity index is punished with m_3. In lines 3 and 5 of the Algorithm 1 we use a Matlab notation [25], just to simplify the explanation of the algorithm.

The function IsB (*IsBound*) is resumed in Eq. 4.

$$IsB(p,\varphi) = \begin{cases} 1 & if\ p+\varphi <= k \\ o & otherwise \end{cases} \quad . \tag{4}$$

where p is the intensity pixel, φ is a constant, an k is a threshold that prevents the overgrowth of a group.

Algorithm 1: Reward Penalty Function.

Data: ;
$IsB(p,\alpha)$ binary function;
$Pb(p) \leftarrow$ Partial Background at time i for pixel p;
$H(p) \leftarrow$ Histogram Table at time $i-1$ for pixel p;
$h \leftarrow$ Histogram for each pixel at time $i-1$;
$\delta \leftarrow$ small positive integer;
$Lb(p) \leftarrow$ Last background value for pixel p;
m_1, m_2 and m_3 defined according to scenario ;
Result: ;
h is update at time i

for *Each pixel* $p \in Pb$ *such that* $Pb(p) \geq 0$ **do**
  if $abs(Pb(p) - H_p) < \delta$ **then**
    $h(Pb(p)-\tau : Pb(p)+\tau) = h(Pb(p)-\tau : Pb(p)+\tau) + IsB(h(Pb(p)), \varphi_1) \times m_1$
    $Lb(p) = Pb(p)$
  else
    $h(Pb(p)-\tau : Pb(p)+\tau) = h(Pb(p)-\tau : Pb(p)+\tau) + Isb(h(Pb(p)), \varphi_2) \times m_2$
    $h(Lb(p)-\tau : Lb(p)+\tau) = h(Lb(p)-\tau : Lb(p)+\tau) - m_3$
  end
end

V. Background Generation

In each generation, $H_c(p)$ contains the maximum value associated to a intensity index for each pixel histogram and H contains the corresponding intensity index. In Algorithm 2, the background generation process is presented. For each pixel p in the video frame, a maximum associated value in histogram $h(p)$ is found, then this value is saved in $H_c(p)$ and the intensity index that corresponds to this maximum value is saved in $H(p)$.

The value in $H_c(p)$ represents the behavior of a pixel, a larger value indicates the stability of the pixel intensity while a low value means that the intensity index is unstable, changing for one group to another.

Algorithm 2: Background Generation.

Data: $h \leftarrow$ Histogram for each pixel at time t
Result: H and Hc

for *Each pixel* p **do**
  $Hc(p) \leftarrow maxarg(h(p))$;
  $H(p) \leftarrow$ intensity index of $maxarg(h(p))$;
end

VI. Experimental Results and Discussions

In this section, we present our experiments and a discussion of the results. We compare our proposed method with Chung's method [24]. We conduct many experiments with representative video sequences in order to evaluate the performance of our method in different situations. All experiments are performed using the brightness band of the videos.

After many tests, we define the next values for the parameters of our method: $\tau = 10$ determines the neighborhood size in pixels, $\sigma = 0.9$ the variance of the Gaussian model, $\lambda = \delta = 20$ are the thresholds used in subtraction difference.

In Fig. 5, two video sequences are shown. The first sequence contains a street intersection where the street that is closer to the camera has a lower flow of vehicles while the other has a heavy traffic, most of the time the traffic remains still. In the street cross, the flow is so heavy that vehicles occlude the track for long time, so the time that the track appears is very short compared to the vehicles. Both video sequences present camera vibrations. This video sequence can be found at [26].

In Fig. 6, we show the resulting background using the Chung *et. al.* method and our approach. The red circles show the ghosts (spoor of moving objects) produced by both methods. Chung's method left several ghosts, including the ghost of the truck that was left virtually intact. This occurs because the method is influenced by the first background generation. In addition, the method takes too much time to react to pixel changes from background to foreground or vice versa. In our method, appears the trail of cars due to congestion, but at the same time we can notice that the track begins to be rebuilded.

The second sequence in Fig. 5, shows a continuous flow of vehicles, which pass in both directions. This video sequence can be found at [27]. In Fig. 7, we show the resulting background using Chun *et. al.* method and our method. Notice that the only method that left ghosts in the background image is Chung's method. Also we can see that the ghost of the truck marked with blue circle is almost vanishing while other ghosts remain, this is an example of the low react to changes that Chung's method has. The same occurs with the numbers in black circle, the algorithm delays the change of the numbers.

In Fig. 8, we present the final results of both methods with a sequence with different illumination conditions and foggy weather. The red circles show the ghosts left by Chung's method. In Fig. 9, we present the resulting background produced using our method in a snowing video sequence. Despite the small size of the video, our method reacts very fast, generating the background with a small number of frames [26].

In Table I, we present the values that we use for the φ multiplicands in Reward/Penalty function, we used two variations of the values represented by two classes: first class (Group Seq. 1) is composed of the sequence A in Fig. 5, the foggy and snowing sequence. The second class (Group Seq. 2) is composed by the sequence B in Fig. 5. The setting of these parameters influence in the final result. If $\varphi_2 > \varphi_3$ then the method reacts much faster to changing groups, if $\varphi_2 < \varphi_3$ then the method keeps the already existing group.

(a) Frame 1

(b) Frame 1

(c) Frame 300

(d) Frame 300

(e) Frame 600

(f) Frame 600

(g) Frame 900

(h) Frame 900

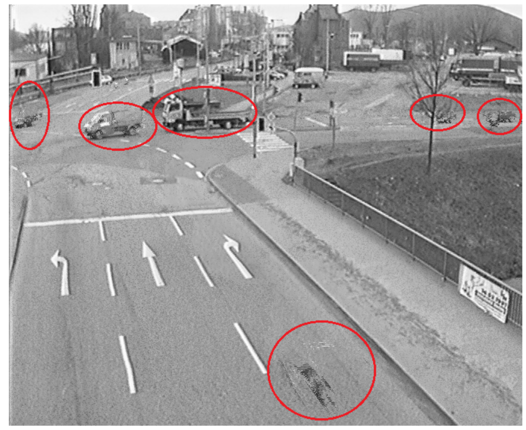
Fig. 5. Video Sequence A and B

	φ_1	φ_2	φ_3
Group Seq. 1	0.9	0.7	0.3
Group Seq. 2	0.9	0.1	0.8

TABLE I
φ VALUES USING IN THE EXPERIMENTS

In most cases, our method success in the generation of the background image, especially in fluid traffic flow. When the background generation begins with heavy traffic, some ghost or trials are left in the final background, this occurs because there is not enough information for rebuilding the background.

Finally, our method improves the Chung's technique adding the Tai's neighboring idea. Our method differs from Tai's method in way the histogram is updated, we use a reward/penalty function.



(a) Chung *et. al.* method

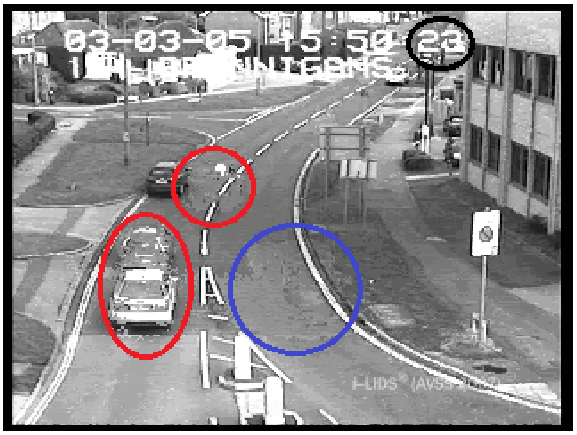
(b) Our approach

Fig. 6. Experimental Results for sequence A

VII. Conclusions and Future Work

In this paper, we present a new technique for background generation based on: frame difference, histogram model and reward and penalty function. The accuracy of the background image quite impacts on output quality of automatic vehicle identification system based in background substraction.

The proposed method introduces a very fast and simple technique for generating and updating a background image from a video sequence. The intensity values in heavy traffic constantly change and traditional background generation methods require a long time to calculate the background image, especially statistic methods. In contrast to [28] and [29], alike histogram models, our method does not need complex mathematical operations, which is an outstanding advantage of our method.



(a) Chung *et. al.* method

(b) Our approach

Fig. 7. Experimental Results for sequence B

(a) Our approach

(b) Chung *et. al.* method

Fig. 8. Results with foggy weather

In all tests, we use the bright band of the images and succeed in generating a background image with good quality. Using color information could better discriminate the groups (background or foreground), including objects with similar color of the track.

Using partial background images is very helpful in camera vibrations, with camera movements (pan, zoom, tilt, track), illuminance changes by providing.

The Reward and Penalty function allows the method to react faster when a pixel changes from one group to another. This basically occurs when the Reward and Penalty function recompenses at the same time an incoming group and punishes an outgoing group. Actually, the values of φ_2 and φ_3 are the parameters that control this phenomenon. So, an adequate adjust of these values can made the method to be adaptive.

Finally, our reward/penalty progressive background generation improves Chung's method using the neighborhood information.

In future work, we are going to make our method adaptive using contextual information of the video sequence, and improving the group discrimination using color information, $i.e.$, better differentiation between the moving objects and the real background.

VIII. Acknowledgments

The authors are grateful to CNPq and CAPES, Brazilian research funding agencies, for the financial support to this work.

References

[1] R. Aarthi, S. Padmavathi, and J. Amudha, "Vehicle detection in static images using color and corner map," in *ITC '10: Proceedings of the 2010 International Conference on Recent Trends in Information, Telecommunication and Computing*. Washington, DC, USA: IEEE Computer Society, 2010, pp. 244–246.

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

[3] D. Beymer, P. McLauchlan, B. C. J., and Malik, "A real-time computer vision system for measuring traffic parameters," *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 495 –501, jun. 1997.

[4] T. Schoepflin and D. Dailey, "Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 4, no. 2, pp. 90 – 98, 2003.

(a) Sample frame

(b) Background

Fig. 9. Results with snowing weather using our Approach

[5] F. Cathey and D. Dailey, “One-parameter camera calibration for traffic management cameras,” in *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, 2004, pp. 865 – 869.

[6] U. Stilla, F. Rottensteiner, and N. Paparoditis, Eds., *Efficient road mapping via interactive image segmentation*. IAPRS, 2009, pp. 1–8.

[7] L.-W. Tsai, J.-W. Hsieh, and K.-C. Fan, “Vehicle detection using normalized color and edge map,” *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 850–864, 2007.

[8] M. Piccardi, “Background subtraction techniques: a review,” in *Proceedings of IEEE SMC 2004 International Conference on Systems, Man and Cybernetics*, vol. 4, oct 2004, pp. 3099–3104.

[9] S. ching S. Cheung and C. Kamath, “Robust techniques for background subtraction in urban traffic video,” in ., S. Panchanathan and B. Vasudev, Eds., vol. 5308. SPIE, 2004, pp. 881–892.

[10] Y. Zhang, P. Shi, E. Jones, and Q. Zhu, “Robust background image generation and vehicle 3d detection and tracking,” in *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, oct. 2004, pp. 12 – 16.

[11] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, “Detecting moving objects, ghosts, and shadows in video streams,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 1337–1342, 2003.

[12] L. Vibha, M. Venkatesha, P. G. Rao, N. Suhas, P. D. Shenoy, K. R. Venugopal, and L. M. Patnaik, “Moving vehicle identification using background registration technique for traffic surveillance,” in *Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Volume I, IMECS ’08, March 19 - 21, 2008, Hong Kong*, ser. Lecture Notes in Engineering and Computer Science, S. I. Ao, O. Castillo, C. Douglas, D. D. Feng, and J.-A. Lee, Eds., International Association of Engineers. Newswood Limited, 2008, pp. 572–577.

[13] P.-S. Hwang, K.-Y. Eom, J.-Y. Jung, and M. hyun Kim, “A statistical approach to robust background subtraction for urban traffic video,” in *Computer Science and Engineering, 2009. WCSE ’09. Second International Workshop on*, vol. 2, oct. 2009, pp. 177 –181.

[14] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis, “Background and foreground modeling using nonparametric kernel density estimation for visual surveillance,” *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151 – 1163, jul 2002.

[15] J. Rymel, J. Renno, D. Greenhill, J. Orwell, and G. Jones, “Adaptive eigen-backgrounds for object detection,” in *Image Processing, 2004. ICIP ’04. 2004 International Conference on*, vol. 3, oct. 2004, pp. 1847 – 1850 Vol. 3.

[16] S.-C. Chen, M.-L. Shyu, S. Peeta, and C. Zhang, “Learning-based spatio-temporal vehicle tracking and indexing for transportation multimedia database systems,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 3, pp. 154–167, 2003.

[17] A. Dominguez-Caneda, C. Urdiales, and F. Sandoval, “Dynamic background substraction for object extraction using virtual reality based prediction,” in *Electrotechnical Conference, 2006. MELECON 2006. IEEE Mediterranean*, may 2006, pp. 466 –469.

[18] M. Unger, M. Asbach, and P. Hosten, “Enhanced background subtraction using global motion compensation and mosaicing,” in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, oct. 2008, pp. 2708 –2711.

[19] Z. Wei, J. Tian, C. Liu, and S. Wu, “An updating algorithm of self-adaptive background based on energy method,” in *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on*, sept. 2008, pp. 848 –852.

[20] P. Kumar, S. Ranganath, H. Weimin, and K. Sengupta, “Framework for real-time behavior interpretation from traffic video,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 6, no. 1, pp. 43 – 53, march 2005.

[21] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, “Traffic monitoring and accident detection at intersections,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 1, no. 2, pp. 108 –118, jun 2000.

[22] J. M. Wang, Y. C. Chung, S. C. Lin, S. L. Chang, S. Cherng, and S. W. Chen, “Vision-based traffic measurement system,” in *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR’04) Volume 4 - Volume 04*, ser. ICPR ’04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 360–363.

[23] K.-T. Song and J.-C. Tai, “Real-time background estimation of traffic imagery using group-based histogram,” *J. Inf. Sci. Eng.*, vol. 24, no. 2, pp. 411–423, 2008.

[24] Y.-C. Chung, J.-M. Wang, and S.-W. Chen, “Progressive background images generation,” *CVGIP: Graphical Model and Image Processing*, 2002.

[25] I. 1994-2011 The MathWorks. (2011) Oficial site - matlab. For Matlab notation. [Online]. Available: http://www.mathworks.com/

[26] K. Institute. (2011) Video data set. The video data set contains snow, fog and heavy traffic. [Online]. Available: http://i21www.ira.uka.de/image_sequences/

[27] A. Video and S. based Surveillance. (2007) Video data set 2. The video data set contains free flowing traffic. [Online]. Available: http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html

[28] K.-T. Song and J.-C. Tai, “Real-time background estimation of traffic imagery using group-based histogram,” *J. Inf. Sci. Eng.*, vol. 24, no. 2, pp. 411–423, 2008.

[29] A.-N. Lai, H. Yoon, and G. Lee, “Robust background extraction scheme using histogram-wise for real-time tracking in urban traffic video,” in *Computer and Information Technology, 2008. CIT 2008. 8th IEEE International Conference on*, july 2008, pp. 845 –850.