Robust Background Modeling Based on Multiscale Color Description

Kahlil Muchtar[1], Chih-Yang Lin[2], Li-Wei Kang[3] and Chia-Hung Yeh[4]
[1]Asia University, Taichung
E-mail: mr.k.stt@gmail.com
[2]Asia University, Taichung
E-mail: andrewlin@asia.edu.tw
[3]Academia Sinica, Taipei
E-mail: lwkang@iis.sinica.edu.tw
[4]National Sun Yat-sen University, Kaohsiung
E-mail: yeh@mail.ee.nsysu.edu.tw

***Abstract*— Background subtraction is the basis of object detection and tracking for machine vision systems. Traditional background modeling methods often require complicated computations and are sensitive to illumination changes and shadow interference. In this paper, we propose a multiscale background modeling method, which fully utilizes the color characteristics of each incoming frame. The proposed method is quite efficient and is capable of resisting illumination changes and shadow disturbance. Experimental results show that our method is suitable for real-world scenarios and real-time applications.**

I. Introduction

Object detection is imperative for video surveillance. Typically, a background model is harnessed to distinguish between foreground and background. With a robust background model, the objects can then be successfully extracted from the background.

In literature, a number of methods for detecting moving objects, in which many different features are employed for background modeling, have been proposed. The most frequently used features are based on color information. For example, a color statistical approach [10] accomplishes background subtraction without being affected by shadow; furthermore, the algorithm is implemented by a DM270 iMX and DSP subsystem [7] for DV applications. In addition to the running statistics (e.g. average) of neighboring frames, a one-Gaussian adaptive modeling method is a popular approach that can be found in [11].

However, one-Gaussian modeling cannot cope with dynamic background changes. Therefore, the Gaussian mixture modeling (GMM) approach [8, 9] was developed by means of using more than one Gaussian model for each pixel. Pixel values that do not fit the model are recognized as foreground areas. One of the examples using GMM was developed (three Gaussians) for traffic monitoring [2]. Other discussions on implementation using GMM can be found [5], which proposed an adaptive learning rate control scheme for GMM.

Motion-based and edge-based methods are other approaches for background modeling. The motion-based method [10] utilizes optical flow to detect salient motion over frames. This approach usually suffers from complicated computations. The edge-based method [6] considers only edge information in frames and constructs edge histograms as a feature description for background modeling. The histogram-matching process determines the performance of this method.

Recently, Heikkilä and Pietikäinen [3] proposed a texture-based background construction method using local binary patterns (LBPs). LBPs have the property of tolerance for illumination changes. However, LBPs are not robust; when the central pixel value used in LBP is affected by noise or swaying trees, the corresponding LBP histogram would not be stable. This increases the possibilities of false positive and false negative cases, respectively. Furthermore, the overlapping block strategy and histogram-matching process proposed in [3] make their method inefficient.

In this paper, we propose a multiscale structure for background modeling based on color statistics derived from each frame. Instead of applying a single scale in the traditional GMM, we propose a new multiscale color descriptor to enhance the tolerance of illumination changes and shadow interference. Another benefit of our background model is that it can effectively resist noise disturbance. Due to the simplified computations of the proposed method, our background modeling is highly efficient and is suitable for real-time applications.

The remainder of this paper is organized as follows. First, we briefly review Heikkilä and Pietikäinen's method in Section 2. Then, in Section 3, we present our new background-modeling scheme based on the multiscale color descriptor as well as its extension. Then, empirical results and discussions are given in Section 4, and the conclusions are presented in Section 5.

II. HEIKKILÄ AND PIETIKÄINEN'S METHOD

The texture-based method proposed by Heikkilä and Pietikäinen [3] first partitions each image frame into overlapping blocks so that the extracted shape of the moving object can be more accurately described. Then, the pixels in each block produce a histogram according to their LBP values. An example of an LBP value for a pixel is illustrated in Fig. 1. Assume that the pixel value is 6 and its surrounding pixels are 5, 9, 3, and 1 in counterclockwise order. If the central pixel is greater than its neighbor, a bit 0 can be generated; otherwise, bit 1 is produced. Fig. 1(b) shows the result of the binary pattern, which indicates that the value of LBP is 2.

(a) Neighbors of the central pixel (b) Binary pattern

Fig. 1. An example of generating LBP.

The histograms of each block support the background modeling. The history of each block histogram is modeled by K weighted histograms for the purpose of multi-model backgrounds. When a new block histogram comes in, the histogram compares with the K weighted histograms and performs background updating. The update process is similar to Stauffer and Grimson's method [9]. In the updating process, only B ($\leq K$) histograms are selected as the background model. If the incoming histogram is similar to a model histogram, the new block is regarded as a background block; otherwise, it is recognized as a foreground block.

III. NEW BACKGROUND MODELING METHOD

In this section, we describe the proposed multiscale color descriptor and corresponding background modeling.

A. *Multiscale Color Description*

When a camera captures an image, the frame is first divided into non-overlapping blocks with a size of $n \times n$ pixels. For each block, the mean value m is calculated and defined as follows:

$$m = \frac{1}{n \times n} \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij}, \tag{1}$$

where x_{ij} indicates the pixel value in the position (i, j) of the block.

Unlike GMM, which processes each pixel independently, the proposed method uses the mean value of each block to determine whether the corresponding block is a background or foreground block. When a new block comes in, it is checked against existing model components for matching purposes, where a match means that the block mean value lies within 2.5 standard deviations of a distribution in the GMM. If any of the distributions is matched, the matched distribution will be updated. Otherwise (*i.e.*, none of the K Gaussian distributions can be matched to the current block mean value), the distribution of the GMM, which has the minimum probability, is replaced with the distribution associated with the current block mean value, an initially high variance, and low prior weight.

The update and unmatched processes are derived from Stauffer and Grimson's [9]. When an incoming block matches the background model and is considered to be a background block, the weights of the background model are updated by:

$$w'_K = \alpha M_K + (1-\alpha) w_K, \tag{2}$$

where α is the learning rate and M_k is 1 for the best-matched model and 0 for the others.

The learning rate determines the speed of adaptation. That is, larger learning rates result in faster adaptations.

If the incoming block is a foreground block, the unmatched process replaces the model that has the lowest weight in the background model with the incoming block. Then, the weight of the new block is set to a low initial weight of 0.01 in our experiments. Finally, the weights of the background model are renormalized in order to have a sum of one.

In the above description, the incoming block may match the model with a low weight and is regarded as a background block. However, the low weight means that the corresponding model has a low probability of being a background block. To solve this problem, the weights of the background model are sorted in decreasing order, and only the first B distributions are selected as the background model, such that:

$$\sum_{i=1}^{B} w_i > TH_B \text{, where } TH_B \text{ is a predefined threshold.} \tag{3}$$

B. *Extension Of the Proposed Method*

In this section, we depict the extension of the proposed method in order to reduce block effect and enhance the object shape more seamlessly. For simplicity, the extension method is described in a single channel without loss of generality. For each incoming block, it is represented as a multiscale structure, exhibiting the multiscale resolution for dominant colors. The multiscale tree structure for feature representation built in a recursive manner is described as follows:

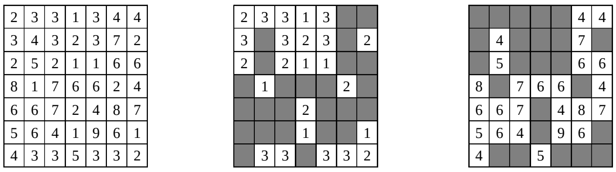
$$R_i = \begin{cases} \{x \mid x \in R_{\lfloor i/2 \rfloor}, where\ x \geq M_{\lfloor i/2 \rfloor}\}, \\ \quad if\ i\ is\ odd\ and\ i \neq 1, \\ \{x \mid x \in R_{\lfloor i/2 \rfloor}, where\ x < M_{\lfloor i/2 \rfloor}\}, \\ \quad if\ i\ is\ even\ and\ i \neq 1, \end{cases} \tag{4}$$

where x denotes the pixel value, M_i is the mean of R_i, and $|R_i|$ is the size of R_i.

Fig. 2 is an example of separating an image into two subsets in the first level of our multiscale structure. Fig. 2(a) is the original image with a size of 7×7 pixels. Based on Eq. (4), we can obtain M_1=3.8 for R_1, and then partition the image R_1 into R_2 and R_3, as shown in Fig. 2(b) and Fig. 2(c), respectively, where M_2=2.17 and M_3=6.84.

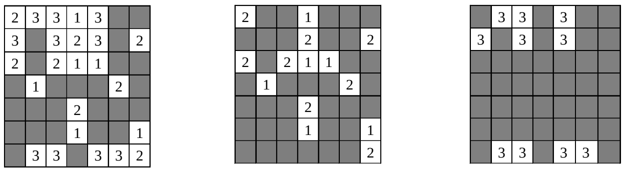
Furthermore, Figs. 3 and 4 present how the second level is derived. From the figures we can observe that, the regions for evaluating the mean values are dynamically determined instead of statically pre-determined. This signifies that these mean values can characterize the block feature more accurately. The corresponding tree structure is shown in Fig. 5.

Unlike in the original proposed method, which represents each block by using only one value, the extended version generates more means as well as background models for each block. In the first level, we classify means into 2 types, namely low mean *lm* and high mean *hm* presented by R_2 and R_3 respectively. Therefore, in Level-1, *lm* and *hm* have their own models, respectively. Similarly, there are four background models for each block in Level-2. The more mean values there are in a block, the more accurate the detection result will be, but more computations are required.



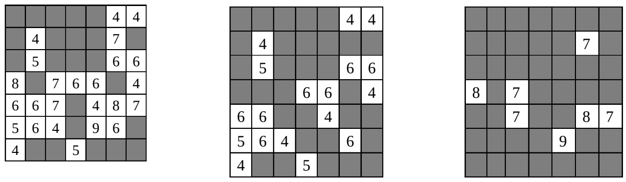
(a) (b) (c)

Fig. 2. An example of the first layer in our multiscale structure.



(a) (b) (c)

Fig. 3. An example of the second layer in our multiscale structure, where (b) is Region R_4 and (c) is Region R_5.



(a) (b) (c)

Fig. 4. An example of the second layer in our multiscale structure, where (b) is Region R_6 and (c) is Region R_7.

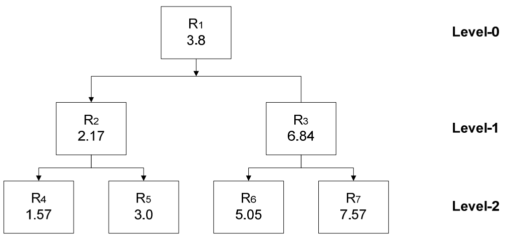


Fig. 5. Result of tree structure of Fig. 2(a).

IV. Experimental Results

The performance of the proposed method is compared with two state-of-the-art approaches, Stauffer and Grimson's method [9] and Heikkilä and Pietikäinen's method [3], using several video sequences. The video sequences were acquired from real indoor and outdoor environments. The simulated environment for the experiments was equipped with a 2.93 GHz Core 2 Duo Intel processor and 2 GB of memory. The image resolution was set to 320×240 pixels. All algorithms were implemented in C++.

For the sake of labeling and segmenting the foreground pixels, the connected components algorithm [1] was applied to each background modeling method. The parameters used in the experiments are listed in Table 1, where α is the learning rate, TH_B is used in (3), K denotes the number of Gaussians, and BS is the block size. An 'X' signifies that the parameter is not required for that method. $LBP_{P,R}$ is only used in Heikkilä and Pietikäinen's method [3, 4] and represents using radius R to find P neighbors such as the example shown in Fig. 1(a).

TABLE I

The parameter values used in the experiments

Parameters	α	TH_B	K	BS	TH_{smooth}	TH_D	$LBP_{P,R}$
Stauffer and Grimson's method	0.005	0.9	3	X	X	X	X
Heikkilä and Pietikäinen's method	0.005	0.9	3	4×4	8	0.65	$LBP_{4,2}$
Proposed method	0.005	0.9	3	4×4	8	0.75	X

The performance comparisons of these three methods are presented in Table 2, where the last row denotes the connected components labeling (CCL) method that is also involved in the background construction. From this table we can observe that the proposed method is much faster than other methods because the proposed multiscale approach requires only mean block operations. Heikkilä and Pietikäinen's method is slowest since they divide each frame into overlapping blocks, and the size of the LBP histogram significantly affects the performance. Furthermore, Table 3 shows the comparison between the proposed method and its extension, where it compares the frame rate when a different number of levels is applied.

TABLE II

Frame rates using different methods

Methods	Stauffer and Grimson's method	Heikkilä and Pietikäinen's method	Proposed method
Frame rate	20.94	3.54	50.75
Frame rate with CCL	20.01	3.38	50.41

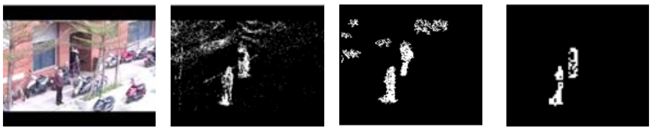
TABLE III

Frame rate comparison of the proposed methods

Proposed Methods	Level-0	Level-1	Level-2
FPS	50.41	44.21	37.15

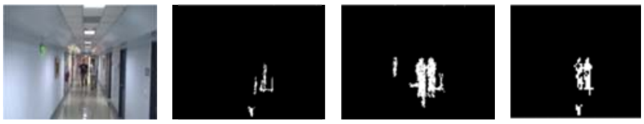
Fig. 6 demonstrates the results from an outdoor scene and the effect of a swaying tree. It shows that Stauffer and Grimson's method is very sensitive to a swaying tree and Heikkilä and Pietikäinen's method is still distracted as well. Since the proposed method uses the mean of each block instead using each pixel strategy to do background modeling, Fig.6.d shows how the proposed method is not susceptible to swaying tree effect.

Fig. 7 shows the indoor scene with some illumination changes, where people were walking towards the camera. In the detection result, it can be observed that Stauffer and Grimson's method is very sensitive to illumination changes, and Heikkilä and Pietikäinen's method also suffers from noise interference. On the contrary, the proposed method is much better able to resist illumination changes. Since the non-overlapping block approach is adopted, the contour of the proposed method is coarser than that of the compared methods.



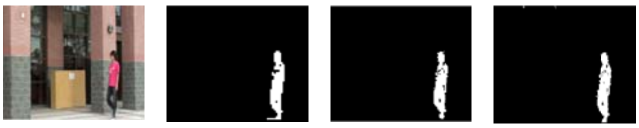
(a) (b) (c) (d)

Fig. 6. Detection results of the first sequence: (a) original image, (b) Stauffer and Grimson's method, (c) Heikkilä and Pietikäinen's method, and (d) proposed method level-0.



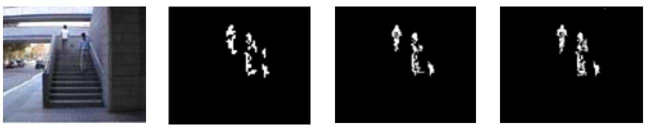
(a) (b) (c) (d)

Fig. 7. Detection results of the second sequence: (a) original image, (b) Stauffer and Grimson's method, (c) Heikkilä and Pietikäinen's method, and (d) proposed method level-0.



(a) (b) (c) (d)

Fig.8. Detection results of the fourth sequence: (a) original image, (b) proposed method level-0 model, (c) proposed method level-1 model, and (d) proposed method level-2 model.



(a) (b) (c) (d)

Fig.9. Other detection results of the fourth sequence: (a) original image, (b) proposed method level-0 model, (c) proposed method level-1 model, and (d) proposed method level-2 model.

With regards to the extension of the proposed method, the comparison and detection results of Level-0, Level-1 and Level-2 modes are shown in Figs.8. In the result, the accuracies of Level-1 and Level-2 modes are greater than that of Level-0. Fig.8(d) shows Level-2 mode could enhance shape of motion object and alleviate block effect.

Fig. 9 shows that the noise around foreground objects can be alleviated by Level-1 and Level-2. In essence, the extension, which utilizes a multiscale color descriptor, can give more a complete object shape without drastically affecting time consumption.

V. Conclusions

In this paper, we proposed a multiscale color feature method for background modeling. The proposed method has the following advantages: (1) alleviating the effect of swaying tree, (2) low computations, and (3) easy implementation. Since the proposed method possesses a very high frame rate, it is quite suitable for real-time applications or low-power computation systems such as cell phones and personal digital assistants (PDAs).

Acknowledgment

This work was supported by National Science Council, Taiwan, under Grants NSC 99-2218-E-468-023.

References

[1] F. Chang, C. J. Chen, and C. J. Lu, "A Linear-Time Component-Labeling Algorithm Using Contour Tracing Technique," *Computer Vision and Image. Understanding*, vol. 93, no. 2, pp. 206-220, 2004.

[2] N. Friedman and S. Russell, "Image Segmentation in Video Sequences: A Probabilistic Approach," *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence*, San Francisco, pp. 175-181, 1997.

[3] M. Heikkilä and M. Pietikäinen, "A Texture-Based Method for Modeling the Background and Detecting Moving Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 657-662, 2006.

[4] M. Heikkilä, M. Pietikäinen, and J. Heikkilä, "A Texture-Based Method for Detecting Moving Objects," *Proceedings of British Machine Vision Conference*, British, pp. 187-196, 2004.

[5] H. H. Lin, J. H. Chuang, and T. L. Liu, "Regularized Background Adaptation: A Novel Learning Rate Control Scheme for Gaussian Mixture Modeling," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 822-836, 2011.

[6] M. Mason and Z. Duric, "Using Histograms to Detect and Track Objects in Color Video," *Proceedings of the 30th on Applied Imagery Pattern Recognition Workshop*, Washington, DC, USA, pp. 154-159, 2001.

[7] T. Matsuyama, T. Ohya, and H. Habe, "Background Subtraction for Non-Stationary Scenes," *Proceedings of the 4th Asian Conference on Computer Vision*, Taipei, Taiwan, pp. 662-667, 2000.

[8] C. Stauffer and W. E. L. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Fort Collins, Colorado, USA, pp. 246-252, 1999.

[9] C. Stauffer and W. E. L. Grimson, "Learning Patterns of Activity Using Real-Time Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747-757, 2000.

[10] L. Wixson, "Detecting Salient Motion by Accumulating Directionally-Consistent Flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 774-780, 2000.

[11] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-Time Tracking of the Human Body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780-785, 1997.