Moving object Detection Based on Improved Codebook Model

Ruidong Gao
School of Instrumentation Science and Optoelectronics Engineering, Beihang University, Beijing 100191, China
rxdj1990@163.com

***Abstract*—Background modeling is a key technology in video surveillance field. Conventional methods used to detect the moving object have drawbacks in the aspects of accuracy and robustness. In order to address these problems, we change the RGB columnar structure in traditional codebook model and establish codebook background model in the YUV color space. We cluster every pixel on a timeline in an image and extract background templates. Then the current image and the background templates are compared. At the same time, the background templates are renewed. We integrate the frequency information to improve the process of symbol decision, deletion and matching. In the meantime, the spatial information of current pixel is added into the prediction of the foreground, making the results of foreground detection more reliable. The experimental results indicate that the proposed method can detect the moving objects effectively and correctly under various conditions. In addition, the proposed algorithm also has the good noise robustness.**

Keywords-background modeling; codebook; moving object detection; noise robustness

I. INTRODUCTION

Moving object detection is an important issue in the computer vision system [1]. It is widely used in video surveillance, target tracking, and image retrieval [2]-[5]. There are many methods that can extract moving objects, such as frame difference method [6], GVF snake algorithm [7]. Although these methods are effective when the moving objects change slightly, their performance is poor in case of radical change of objects.

Background modeling is a good approach to detect moving objects in the video sequence [8], [9]. The majority of the existing background models are based on the pixels, such as the single Gaussian model, Gaussian mixture model. In [10], the single Gaussian distribution is used to update the background. But in the outdoor scenes, the scenes constructed using single Gaussian model will result in shadows and high time consumption, due to the elapse of time, the variations of illumination, the halting of moving objects and the moving of stationary objects.

The main idea of Gaussian mixture model [11], [12] is to assume that the probability density function of each pixel is either the Gaussian function or the Gaussian mixture function. A reliable scene model can be established by statistically classifying each pixel with a number of frames. But due to the rapid change of the scenes, the model can only create a small number of Gaussian factors and cannot achieve robust forecast of foreground. The learning efficiency is so low that the model can hardly detect the sudden change of the scenes. If the model changes too fast, then the foreground moving slowly will be assimilated into the background, resulting in high misdetection rate.

In this paper, the YUV color model is used in replace of the traditional RGB model. The structures of code cells are simplified because of the independence between the color luminance channels. The processes of predication, deleting and matching code elements in the traditional codebook background model are modified by integrating the frequency information into the traditional codebook background model. Meanwhile, the spatial information of the existing pixels is introduced to the decision procedures of foreground. Because the spatial information contains the dependency relationship between neighboring pixels, the foreground detection can be more reliable. This codebook model is suited for periodically changing background, and makes the traditional codebook background model less sensitive to illumination variations.

II. TRADITIONAL CODEBOOK

For a given pixel in the images of video sequence, its observations in the time axis form a sequence $X = \{x_1, x_2, ..., x_N\}$, which is a sample sequence consisting of N RGB vectors of the pixel at the same position. N denotes the number of training frames. $C = \{c_1, c_2, ...c_L\}$ is the pixel's codebook consisting of L code words. Each code word c is defined as a tuple, $v_i = (\overline{R}_i, \overline{G}_i, \overline{B}_i)$ and $u_i = \langle I_i^{\min}, I_i^{\max}, f_i, \lambda_i, p_i, q_i \rangle$, $i = 1, 2, 3...L$. $I_i^{\min}, I_i^{\max}$ denote the minimum and maximum luminance of the code word's corresponding pixels. λ is the largest time interval between the appearance of the code word during training, namely, the number of negative frames. p and q denote the first and last matching time of the code word after it appears. The variations of background are mostly reflected in variations of luminance. So the luminance of the background pixels increases along the main axle formed with the origin and the code words in the color model. The color model and the similarity determination are shown in Fig. 1.

For the input pixel $x_i = (R, G, B)$ and the codebook c_i , $v_i = (\overline{R_i}, \overline{G_i}, \overline{B_i})$, we can obtain the following equations:

$$\|x_t\|^2 = R^2 + G^2 + B^2 \quad (1)$$

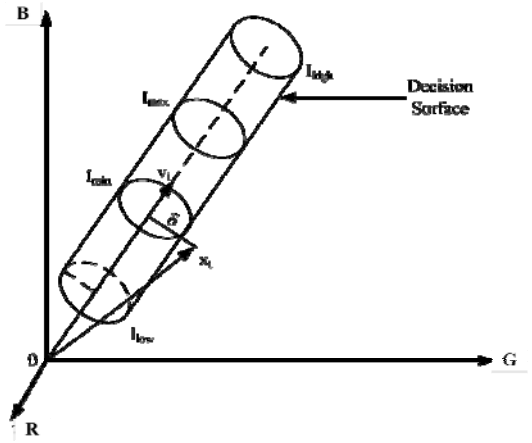


Figure 1. Cylinder model of Codebook algorithm

$$\|v_i\|^2 = \overline{R_i}^2 + \overline{G_i}^2 + \overline{B_i}^2 \tag{2}$$

$$\left(x_i, v_i\right)^2 = \left(\overline{R_i}R + \overline{G_i}G + \overline{B_i}B\right)^2 \tag{3}$$

Color distortion δ can be computed using (5).

$$p^2 = \|x_i\|^2 \cos^2\theta = \frac{\left(x_i, v_i\right)^2}{\|v_i\|^2} \tag{4}$$

$$colordist\left(x_i, v_i\right) = \delta = \sqrt{\|x_t\| - p^2} \tag{5}$$

To accommodating the luminance variations and statistically assign the minimum and maximum luminance variations $I_i^{\max}$ and $I_i^{\min}$, we can empirically set the longest deadline for deleting the un-accessed codebook, achieving background pixel detection approximated to adaptive filtering. Finally, the difference image of the existing foreground targets can be obtained through real-time updates of the background pixels.

The details of the codebook structure algorithm are as follows:

(1) Initialize the codebook, i.e. clear the codebook: $L = 0, C = \Phi$.

(2) For $t = 1$ to N, execute the following operations:

(i) $x_t = \left(R, G, B\right)$, $I = \left(R^2 + G^2 + B^2\right)$

(ii) Find the code word $c_m \in C$ that matches with x_t by satisfying the (a) and (b) conditions:

(a) $colordist\left(x_t, v_m\right) \le \varepsilon$

(b) $brightness\left(I, \left(I_m^{\min}, I_m^{\max}\right)\right)$=true

(iii) If $C = \Phi$ or no match is found, then $L = L + 1$. In this case, add a new code word c_L, and set $x_L = \left(R, G, B\right)$, $u_L = \left(I, I, 1, t-1, t\right)$.

(iv) Otherwise, update the members v_m and u_t of the code word c_m:

$$v_m = \left(\frac{f_m R_m + R}{f_m + 1}, \frac{f_m G_m + G}{f_m + 1}, \frac{f_m B_m + B}{f_m + 1}\right) \tag{6}$$

$$u_t = \left\langle \min\left(I, I_m^{\min}\right), \max\left(I, I_m^{\max}\right), f_m + 1, \max\left(\lambda_m, t - q_m\right), p_m, t \right\rangle \tag{7}$$

End for

After the training, compute the longest time interval between two consecutive appearances of each code word of this pixel, i.e. the number of negative frames. That is, for any c $\left(i = 1, 2, 3...L\right)$, we have $\lambda = \max\left(\lambda_i, \left(N - q_i + p_i - 1\right)\right)$.

(3) Eliminate redundant code words using λ to obtain the refined initial codebook CB that can best represent the actual background.

$$CB = \left\{c_k \mid c_k \in C, \lambda_k \le T_M\right\} \tag{8}$$

Where λ is the threshold of the longest time interval and usually set to half of the number of the training frames, meaning that all representative background words appear for at least half of the time.

The (a) and (b) conditions in step (ii) will be satisfied if the colors of x_t and v_m are very similar and the luminance of x_i falls in the acceptable range of v_m. Here it suffices to find the first code word that satisfies these two conditions.

Meanwhile, the time interval criterion is introduced due to the redundancy in code words acquired during training. The code words that represent the foreground moving targets and the noises can be statistically eliminated using $\lambda = \max\left(\lambda_i, \left(N - q_i + p_i - 1\right)\right)$, thus allowing the moving targets to exist during the initial learning process.

Foreground detection is similar to background modeling. That is, comparing the current pixel with the existing background model. If it satisfies the color and luminance distortion conditions, then define it as the background, otherwise the foreground, i.e. the moving target.

III. IMPROVED CODEBOOK

We improve the original codebook model and rewrite the model parameters [13], [14]. For sample sequence of the pixels in time-domain $X = \left\{x_1, x_2, ..., x_N\right\}$ and its code word set $C = \left\{c_1, c_2, ...c_L\right\}$, the luminance and color channels are independent from each other in the YUV color space. Processing the three channels separately will simplify the matching equations. The binary combinations of each channel code word is:

$$c_i = \langle learn\min_i, learn\max_i, \min_i, \max_i, f_i, \lambda_i, q_i \rangle \quad (9)$$

Where *learnMin*, *learnMax* denote the lower and upper bound of each channel's learning values, f denotes the number of times that the code element appears, q represents the number of frames when the code element is finally updated, λ denotes the number of negative frames of the code elements.

The updating procedures for training and sustaining the background models are as follows:

(1) Create a codebook.

(2) Traverse each pixel in the codebook, compute the pixel's downward extension value *low* and the upward extension value *high*:

$$\begin{cases} low = \max(p - cbBounds, 0) \\ high = \min(p + cbBounds, 255) \end{cases} \quad (10)$$

Where p is the pixel's value in a channel, *cbBounds* is the offset of each channel's extended boundary.

(3) Traverse each code element of each pixel to find the code element that matches with the current pixel value.

(4) If the code element satisfies $learnMin < p < learnMax$ for each channel, the matching code element can be found. Each channel of the matching code element should be updated using the following equations:

$$\begin{cases} boxMin = \min(boxMin, p) \\ boxMax = \max(boxMax, p) \end{cases} \quad (11)$$

If $learnMin > low$, set *learnMin*=*learnMin*-1. If $learnMax > high$, then set *learnMax*=*learnMax*+1, $f = f + 1$, $q = T$, and $\lambda = 0$. T is the total number of updated frames in the codebook, *boxMin*, *boxMax* denote the lower and upper limits of each channel's pixel values, respectively.

(5) If no matching code element is found, then add a new code element: *learnMin*=*low*, *learnMax*= *high*, $boxMin = boxMax = p$, $q = T$, $\lambda = 0$, $f = 1$. For code elements that do not match, it suffices to set the number of negative frames $\lambda = \lambda + 1$.

By following the above procedures, we will obtain a reliable and adaptable background model, which can be used for foreground detection with the following steps:

Step 1: Compute the pixel's downward extension offset m and the upward extension offset M in each channel:

$m = \mathrm{mod}\, Min$, if the shadow detection needs to be done, then $m = m + BrightnessMin$.

$M = \mathrm{mod}\, Max$, if the shadow detection needs to be done, then $M = M + BrightnessMax$.

Step 2: Traverse each pixel in the codebook, compute the pixel's downward extension l and the upward extension h and the updating frequency threshold f_{ref}.

Step 3: Calculate the following equation:

$$l = p + m \quad (12)$$

Step 4: Calculate the following equation:

$$h = p - M \quad (13)$$

Where p is the current value of the pixel, f_{ref} is the sum of updating frequencies of all code elements corresponding to the current pixel.

Step 5: Traverse each code element of each pixel to find the code element that contains the current pixel values.

Step 6: If the code element satisfies $boxMin \le l$, $h \le boxMax$ and $f > f_{ref}$ in each channel, then the code element that contains the current pixel values is found and the pixel belong to the background.

Step 7: If there is no code element that contains the current pixel values, then consider the pixel as the foreground.

Step 8: Based on the steps (1)-(5), if the pixel is determined to be the foreground, then a two-layer detection is needed to avoid misjudgment, i.e. re-create the codebook on the pixel that has been determined to be the foreground. Update the code element with the input pixel information, and compute the information about the newly created codebook every five frames. If it has $f \ge Tadd$ and $convertScale \ge Taveragestale$ simultaneously, then integrate the pixel into the background pixels again, where

$$convertScale = \frac{(FramNum - T)}{frequency} \quad (14)$$

The above steps (1)-(5) form the one-layer Codebook algorithm and the introduction of the step (6) yields the two-layer Codebook algorithm.

Because λ cannot sufficiently and reliably remove the foreground pixels from the background code elements, the frequency f is empirically integrated into the decision criterion in the proposed algorithm. And the current code element is defined as the background only if $\lambda_k \le T_M$ and $f \ge f_{ref}$. Limiting f_{ref} between 15 and 20 in the proposed algorithm obtains excellent performance.

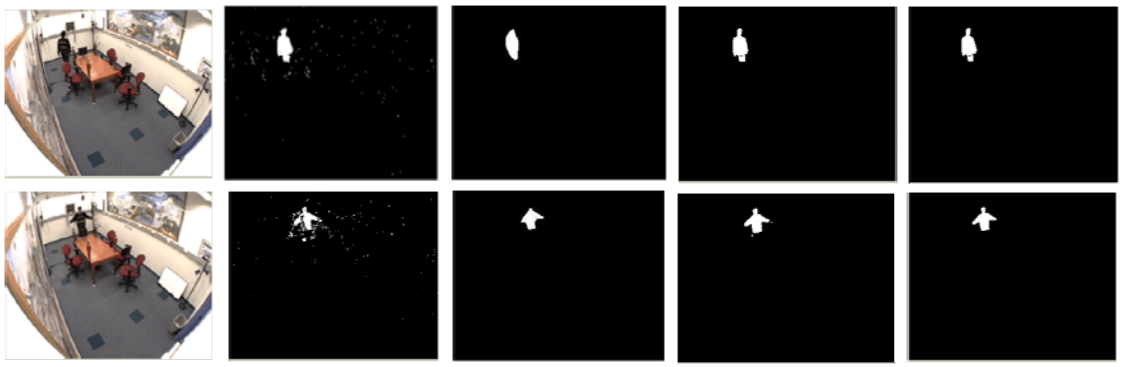
The information from the single pixel is not reliable, so the spatial information is integrated into the background model to make the background detection more reliable. Consider that each background element will oscillate in a small neighborhood. The pixel x that has been determined to be the foreground will be checked again by taking the neighborhood ($y \in N_x$) into account. If it is found that x

matches with the background model of a pixel in the neighborhood, then the current pixel x will be labeled as the background.

The procedures for spatial background detection are as follows:

If x meets the condition in step (4) of the above foreground detection algorithm, then label the current pixel as the background b. Otherwise, temporally label the current pixel as the foreground f. Check the pixel labeled in the previous step again by finding the matching code element in $y \in N_x$. If it is found, reset the pixel that has been labeled as the foreground as the background b. Otherwise, confirm that the pixel is the foreground f.

The created background codebook represents the foreground and background information more reliably and accurately, because the frequency information of the current pixel and the codebook of the neighboring pixels are integrated into the background modeling of the current pixel and the foreground decision criterions.



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

Figure 2. The results of "Room" video object detection using different methods. (a) Original video sequence. (b) Gaussian mixture modeling method. (c) The improved GVF Snake method. (d) One layer codebook. (e) Two layer codebook.

IV. Experimental Results and Analysis

To verify the effectiveness of the proposed algorithm, the video sequence "Room", which has 300 frames, is selected. The Gaussian mixture model-based background modeling algorithm, the improved GVF Snake algorithm, the one-layer codebook background modeling algorithm and the two-layer codebook background modeling algorithm, are applied to the video sequence for comparison. The algorithms are implemented with C as the programming language and Visual studio 2008 as the development environment on a platform of Core™ i7-3770 3.40GHz CPU and 4G memory.

The results of the video sequence "Room" are shown in Fig. 2. Fig. 2(a) is the original video sequence images. From the results, it can be observed that the Gaussian mixture model-based background modeling algorithm cannot accurately obtain the foreground images, and the results are heavily polluted with noises, as given in Fig. 2(b). The improved GVF Snake algorithm provides over smoothing performance of boundary convergence, resulting in the loss of the object's important information, as shown in Fig. 2(c). The proposed codebook-based background modeling algorithm can accurately acquire the changing foreground objects, and is highly robust to noises, as shown in Fig. 2(d) and (e). The experiment on the video sequence demonstrate that compared with Gaussian mixture model-based background modeling algorithm and the improved GVF Snake algorithm, our method provides excellent detection performance and is highly robust to noises.

Subjective evaluation of the algorithms is defective. To ensure objective performance evaluation, an objective evaluation approach proposed by Wollborn in the MPEG-4 core experiment [15] is used here.

The resulting errors are either false detection or omission. According to a pre-determined accurate reference template, the probabilities of the two errors can be defined as follows:

$$d\left(A_t^{est}, A_t^{ref}\right) = \frac{\sum_{(x,y)} A_t^{est}(x,y) \oplus A_t^{ref}(x,y)}{\sum_{(x,y)} A_t^{ref}(x,y)} \quad (15)$$

Where $A_t^{ref}(x,y)$ and $A_t^{est}(x,y)$ denote the object template acquired by the reference segmentation and the actual segmentation methods in the t^{th} frame, respectively. $\oplus$ denotes the XOR operation. The SA (Spatial Accuracy) of the algorithm is defined in (16):

$$SA = 1 - d\left(A_t^{est}, A_t^{ref}\right) \quad (16)$$

SA accurately reflects the shape similarity of each frame's extraction to the reference segmentation template. The higher the SA value is, the more accurate the segmentation is. An accurate reference template is needed to ensure right evaluation in the Willbom standard evaluation

system. But in MPEG-4 there is no standard template of relevant video sequences. Hence, the reference template in our proposed algorithm is manually extracted using Photoshop CS5. The quantitative evaluation results are given in Fig. 3. The comparisons show that our algorithm outperforms other methods.

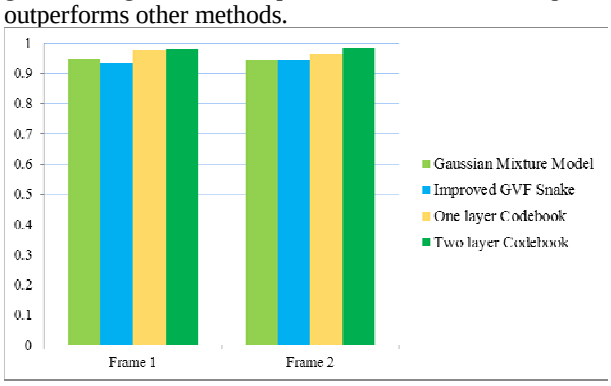


Figure 3. The quantitative evaluation results of different methods on video sequence "Room".

V. CONCLUSION

Our proposed algorithm changes the columnar structure used to measure the color distance in the traditional codebook model, and simplifies the code element structure by creating the codebook background model with the YUV color model. In the processes of determining, deleting and matching code elements in the traditional codebook background model, the reliability is low because only the number of negative frames is taken into account. So the frequency information is integrated to achieve greater reliability. The foreground detection reliability is improved by introducing the spatial information of the existing pixels that contains the dependency relationship between neighboring pixels to the foreground decision procedures. The sensitivity of the traditional codebook background model to the frequently changing illuminations is reduced. Experimental results show that our proposed algorithm can detect moving target accurately.

ACKNOWLEDGMENT

This research is funded by the National Natural Science Foundation of China (NSFC) under grants No. 61375025.

REFERENCES

[1] Q. J. Zhao, D. B. Zhao and Y. H. Lu, "A novel algorithm for multi-target detection based on spatial-temporal information,". Chinese Journal of Scientific Instrument, vol. 32, no. 4, pp. 877-882, April 2011.

[2] S. P. Zhu, Z. C. Guo and L. Ma, "Shadow removal with background difference method based on shadow position and edges attributes," EURASIP Journal on Image and Video Processing, vol. 22, pp. 1-15, December 2012.

[3] S. P. Zhu, Y. S. Hou, Z. K. Wang and K. Belloulata, "Fractal video sequences coding with region-based functionality," Applied Mathematical Modelling, vol. 36, no. 11, pp. 5633-5641, November 2012.

[4] S. P. Zhu and Y. Gao, "Noncontact 3-D coordinates measurement of cross-cutting feature points on the surface of a large-scale workpiece based on the machine vision method," IEEE Transactions on Instrumentation and Measurement, vol. 59, no. 7, pp. 1874-1887, July 2010.

[5] S. P. Zhu, J. C. Fang, R. Zhou, J. H. Zhao and W. B. Yu, "A new noncontact flatness measuring system of large 2-D flat workpiece," IEEE Transactions on Instrumentation and Measurement, vol. 57, no. 12, pp. 2891-2904, December 2008.

[6] M. G. Gan, J. Chen and J. Liu, "Moving object detection algorithm based on three-frame-differencing and edge information,". Journal of Electronics and Information Technology, vol. 32, no. 4, pp. 0894-0897, April 2010.

[7] S. P. Zhu, J. Gao and Z. Li, "Video object tracking based on improved gradient vector flow snake and intra-frame centroids tracking method," Computers and Electrical Engineering, vol. 40, no. 8, pp. 174-185, November 2014.

[8] B. C. Zhang, Y. S. Gao and S. Q. Zhao, "Kernel similarity modeling of texture pattern flow for motion detection in complex background," IEEE Transactions on Circuits and Systems for Video Technology, vol. 21, no. 1, pp. 29-38. June 2011.

[9] Lee, "Effective Gaussian mixture learning for video background subtraction," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 5, pp. 827-832, May 2005.

[10] C. R. Wren, A. Azarbayejani and T. Darrell, "Real-time tracking of the human body," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 1, pp. 780-785. October 1997.

[11] J. M. Luo, J. Zhu, "Adaptive Gaussian mixture model based on feedback mechanism," International Conference on Computer Design and Applications, vol. 2, pp. 117-181. June 2010.

[12] Q. D. Zhu, K. Li and Z. Zhang, "An improved Gaussian mixture model for an adaptive background model," Journal of Harbin Engineering University, vol. 31, no. 10, pp. 1348-1353, October 2010.

[13] M. J. Wu, X. R. Peng, "Spatio-temporal context for codebook-based dynamic background subtraction," International Journal of Electronics and Communications, vol. 64, no.8, pp. 739-747, May 2010.

[14] A. Ilyas, M. Scuturici and S. Miguet, "Real time foreground-background segmentation using a modified codebook model," IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. 454-459, September 2009.

[15] M. Wollborn, R. Mech, "Refined procedure for objective evaluation of video object segmentation algorithms," ISO/IEC JTC1/SC29/WG11/ M3448, 1998.