Foreground Objects Segmentation in Videos with Improved Codebook Model

[1]Su Su Aung, [2]Nu War
[1]*University of Information Technology, Yangon, Myanmar*
[2]*University of Computer Studies, Mandalay, Myanmar*
[1]*susuaung87@gmail.com,* [2]*nuwar81@gmail.com*

Abstract

*Extraction of foreground objects in real-time is a significant topic for applications in computer vision. Most of the proposed techniques use background subtraction technique to detect moving or static foreground objects in the scene. Despite ongoing lots of research, the domain has not reached mature status and needs more advanced and improved solutions. In this proposed system, background subtraction is done by improved codebook model-based method to get segmented foreground objects. In background modeling, the L*a*b* color space is used instead of RGB color space. This method has been tested with standard datasets and the accuracy of segmentation results are also evaluated. The experimental results demonstrate that the proposed method perform well under difference background subtraction challenges such as dynamic background, shadow, illumination changes and bad weather.*

Key Words- Foreground Segmentation, Background Subtraction, Codebook, Image Processing

1. Introduction

Segmentation of moving regions in video frames sequences is an elemental step of information extraction in several vision systems including human-machine interface, automated visual surveillance, people tracking, traffic monitoring, and semantic annotation of videos. Extracting the objects present in a scene, technically called object segmentation. Motion object segmentation separates foreground images from background images and it is usually followed by object detection, classification and tracking. The detection and tracking of objects are dominated by the accuracy of the foreground object segmentation.

Moving object segmentation is performed primarily by subtracting background. Subtraction of the background is a commonly used method to detect moving objects from static cameras in videos. The basic concept in the approach is that of detecting the moving objects from the difference between the current frame and a reference frame, often called the "background image", or "background model". Frame difference is generally the simplest form of background subtraction. Many background models were introduced to manage various issues. Subtraction of the background is mostly quick and requires small computational requirements. However, sudden changes in illumination and tiny camera movements such as vibrations can be sensitive.

The background scene is statically modeled and is periodically updated for each pixel captured over a period of time to obtain the foreground object. Background modeling is a non-trivial issue related to variations in illumination, incidence of shadow / highlight, entry or removal of objects from the scene. Background modeling methods use previous frame history to model the background. Each pixel of the video frame matches its model of context. If the color value of the pixel is similar to the background model, it is considered as the background pixel otherwise it is a pixel of the object.

In this proposed system, background modeling and foreground extraction is done by using improved codebook algorithm. In background modeling, the L*a*b* color space is used in this approach to calculate the color difference between two pixels using the CIEDE2000 color difference formula to get foreground object segmentation result more closer to the human perception on color differences. The foreground object extracted from the video sequence using this approach is useful for detecting objects in video surveillance applications.

2. Related Work

The Codebook (CB) method is proposed in [8] to build a background model. Codebook background modeling method can be classified as cluster models in which each pixel in the frame can be temporally represented by clusters. It is a quantization technique using long scene observation for each pixel. In order to recognize foreground objects in an image, the method utilizes Maximum Negative Runtime Length (MNRL) based on a color distortion and intensity difference metric relative to pixels of consecutive video frames. Basic codebook model is a pixel-based strategy, so for each pixel there is a codebook. Each codebook contains one or more codewords, and the number of codewords contained in

each codebook differs from each other. Each codeword models a sample cluster that builds a part of the sense.

To be robust to illumination changes, other improvements concerned the color models which can be used instead of the cone cylinder model such as the hybrid-cone cylinder model in [3], and the spherical model in [6]. Other modifications concerned block-based approach, hierarchical approach or multi-scale approach to reach real-time requirements. [12] propose an arbitrary cylinder color model which uses cylinders whose axes need not going through the origin, so that the cylinder color model is extended to much more general cases. In [2], the authors proposed codebook background modeling algorithm based on principal component analysis (PCA). The model overcomes the mistake of gaussian mixture model, sphere model and codebook cylinder model. The method in [3] proposed to convert pixels from RGB to HSL color space, and use L component as brightness value to reduce amount of calculation. There are some problems, however, with which codebook can't deal. For example, if the foreground pixel color is similar to the background pixel color, the foreground will be segmented incorrectly. Although it can adjust the parameters to partly resolve this issue, in other circumstances it concurrently reduces the global performance.

3. Background Modeling

Foreground object segmentation in proposed method consists of background codebook modeling and foreground segmentation. Before the background codebook modeling can be carried out, the region of interest (ROI) mask is applied to the incoming video frames. ROI can be defined by creating a binary mask. The main reason for applying ROI is to avoid processing on part of the scene where there is no possibility for detecting stationary object. Another reason is to reduce false positive rate of foreground segmentation process. In modeling the background codebook, L*a*b color difference model is used to calculate the color difference between two pixels. All the input video frames are converted to L*a*b color model from the RGB. The detail process of background codebook modeling and foreground object segmentation by the proposed system is shown in Figure 1.

The color distortion between two pixels values is computed using CIEDE2000 color difference formula. In the background modeling process, pixels are quantized into codeword according to color variation. Frames from the start of the video are used to model the background. The number of training frame can be varied according to the complexity of the scene. If the scene includes complex background movements or foreground object moving at the start, more frames are needed to train the background codebook model to cover all that dynamic scenes in the background. In the foreground detection phase, if the color difference value (ΔE) between current pixel and related codeword is greater than the threshold, the pixel is defined as foreground pixel. If not, the current pixel is defined as background and the background codeword is modified.

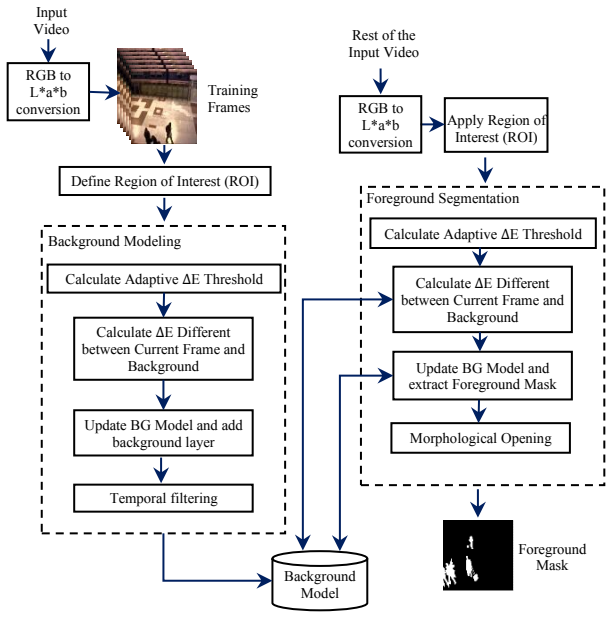

Figure 1. The detail process of Background Modeling and Foreground Segmentation with Codebook

3.1. CIEDE2000 Color Difference (ΔE)

DeltaE (ΔE) is a single number that represents the 'distance' between two colors, one a reference color, the other a sample color. The higher the Delta E, the greater the difference between the two samples that are being compared. The color difference between two L*a*b color values, L*a*b$_1$ and L*a*b$_2$ can be computed by the following equation [10].

$$\Delta E\,(Lab_1, Lab_2) = \sqrt{\left(\frac{\Delta L'}{k_L S_L}\right)^2 + \left(\frac{\Delta C'}{k_C S_C}\right)^2 + \left(\frac{\Delta H'}{k_H S_H}\right)^2 + R_T \frac{\Delta C'}{k_C S_C} \frac{\Delta H'}{k_H S_H}}$$

where,

- $\Delta L'$ = Lightness difference
- $\Delta C'$ = Chroma difference
- $\Delta H'$ = Hue difference
- R_T = Rotation term
- S_L = Lightness weighting function
- S_C = Chroma weighting function
- S_H = Hue weighting function

Default value for parametric weighting factors k_L, k_C and k_H is 1.

3.1. Adaptive Thresholding

Typically, adaptive thresholding requires a gray or color image as input and produces a binary image depicting the segmentation in the simplest implementation. In this work, adaptive threshold is used to separate desirable foreground image objects from the background. Adaptive thresholding chooses for each pixel an individual threshold.

Adaptive ΔE threshold ϵ_t is calculated as follow:
First, standard deviation (σ) of channel 'a' and 'b' is calculated by as, $\sigma = \sqrt{\frac{1}{N}\sum_i^N (x_i - \mu)^2}$

where,

- σ is the standard deviation of 'a' or 'b' channel
- μ is the mean of each channel.
- N is the total number of pixels
- x is 'a' or 'b' value

To get the 2-sigma or 3-sigma ranges, it can be multiplied σ with 2 or 3.

$$\bar{\sigma} = \sigma * 3$$

- Standard L* = μ_L
- Standard a* =$\bar{\sigma}_a$
- Standard b* = $\bar{\sigma}_b$
- *fn* is current frame number.

The ΔE difference between standard L*a*b (Lab_S) and current pixel's L*a*b (Lab_{fn}) is defined as threshold ϵ_t.

4. Modified Codebook Algorithm

For a given pixel in the images of video sequence, $X = \{x_1, x_2, \ldots, x_N\}$, which is a sample sequence consisting of N number of RGB vectors. N denotes the number of training frames. $C = \{c_1, c_2, \ldots, c_k\}$ is the pixel's codebook consisting of k code words. Each codeword c consist of color vector $v_i = (\bar{L}i, \bar{a}i, \bar{b}i)$ with L*a*b values of pixel and a four-tuples $aux_i = \langle f_i, \lambda_i, p_i, q_i \rangle$. f is the frequency or the number of times that codeword is matched. λ is the maximum negative run length (MNRL) which means the largest time span in which this codeword is not updated or accessed. p and q are the first and the last access times of the codeword respectively. The Codebook is a pixel-based model, where each pixel in the video frame is modeled individually. Modified codebook algorithm for background modeling is shown in Table 1.

Codebook which is obtained during the background training represents the training video frames sequence. It may contain foreground objects information also if foreground objects are moving in the scene during training time. Therefore, the variable λ can be used for filtering the codewords which are not updating. It is assumed that codewords that representing the foreground objects colors have higher value of λ. Because codewords representing foreground objects are not updated frequently. The codewords having $\lambda \leq \lambda_{th}$ are deleted from codebook or not used in the process of code matching. This process is called temporal filtering. λ_{th} is a standard Maximum Negative Run Length (MNRL) value. In this proposed system, λ_{th} is defined as λ_{th} = (number of training frame * 0.6). The higher value of λ_{th} becomes the reason of adding many object's pixels color into codebook. Similarly, small value of λ_{th} is unable to model the background movement.

Table 1. Modified Codebook Algorithm for Background Modeling

Input: Stream of pixel values (R, G, B)
Output: *C* (codebook)
1: Initialize the codebook: $k \leftarrow 0$ and $C \leftarrow \emptyset$
2: for t=1 to N do
3: $X_t \leftarrow (R, G, B)$,
4: $Lab_t = rgbToLab(X_t)$
5: $\epsilon_t = findDeltaEThreshold(Lab_t)$
6: Find the codeword c_i in *C* matching to Lab_t by
7: $\Delta E \leftarrow computeDeltaE(Lab_t, V_i)$ using Eq.7
8 if $\Delta E \leq \epsilon_t$ then
9: Update the codeword c_i as follows:
10: $V_i \leftarrow \left(\frac{f_i \bar{L}_i + L}{f_i + 1}, \frac{f_i \bar{a}_i + a}{f_i + 1}, \frac{f_i \bar{b}_i + b}{f_i + 1}\right)$
11: $aux_i \leftarrow \langle f_i + 1, \max\{\lambda_i, t - q_i\}, p_i, t \rangle$
12: else if $C = \emptyset$ or there is no match then
13: Increment *k* by one and create a new codeword $c_k = (V_k, aux_k)$ by assigning,
14: $V_k \leftarrow (Lab_t)$ and $aux_k \leftarrow \langle 1, t - 1, t, t \rangle$
15: Add c_k in *C*.
16: end if
17: end for
18: for each Codeword c_i in C do
19: $\lambda_i \leftarrow max\{\lambda_i, (N - q_i + p_i - 1)\}$
20: end for

5. Foreground Segmentation

To extract foreground mask, each input video frame is compared with the background model like in background modeling process. For the pixel which L*a*b value is match with the codeword, it is regarded as a background pixel. The matching codeword is update according to the step 10 and 11 in the above algorithm. For the pixel which does not match with the codeword, it is regarded as a foreground pixel. In foreground mask, background pixels are valued to 0 and foreground pixels are valued to 1 respectively.

After obtaining foreground mask, morphological opening operation is done to remove small objects (noise). Morphological opening removes all connected components (objects) that have fewer than P pixels from the binary image.

Table 2. Segmentation results comparison of base line challenge

	Baseline					
	recall	specificity	FPR	FNR	PWC %	Accuracy
Proposed	0.862	0.998	0.002	0.138	0.565	0.994
Original CB	0.675	0.954	0.046	0.325	5.418	0.946
GMM	0.406	0.974	0.026	0.594	4.324	0.957

The basic steps are:

- Determine the connected components
- Compute the area of each component
- Remove small components if component's Area <= P

In this experiment, P is set to 15 pixels.

6. Experimental Results

Foreground segmentation result can be evaluated by comparing it with the ground truth. To evaluate the accuracy of the foreground segmentation, total of 31 video sequences are used in this experiment. These videos are from four different datasets which are:

1. Change Detection (CDNet 2014), [5]
2. Wallflower, [11]
3. I2R [9] and
4. SBM-RGBD. [1]

These videos contain various challenges such as dynamic background, illumination change, bad weather, shadow, bootstrap, camouflage, base line. For each challenge, there are ROI masks which show the spatial region of interest and temporal ROI which containing two frame numbers. Only the frames in this temporal ROI range will be used to calculate the score. An accurate ground-truth segmentation and annotation of change or moving regions for each video frame are provided by the dataset.

Six different performance metrics are used to calculate the foreground segmentation accuracy. Which are recall (Sensitivity, True Positive Rate), Specificity (True negative rate), False Positive Rate (FPR), False Negative Rate (FNR), Error Rate (PWC (Percentage of Wrong Classifications)) and Accuracy.

To compare with the proposed method, the original Codebook Background Modeling (CBM) method [8] is also implemented as well as state of the art foreground segmentation method such as Gaussian Mixture Model (GMM) [7]. Then, foreground segmentation results from above three methods are collected and evaluated with the ground truth. Evaluation results are categorized by the type of the challenge.

Baseline category contains four videos, two indoor (Office and PETS2006) and two outdoor (highway and pedestrians). These videos represent a mixture of mild challenges typical of 4 categories. Highway videos have subtle background motion. Pedestrians video have isolated shadows and have pedestrians that stop for a short while and then move away. Office video have stable foreground object. PETS2006 have an abandoned object. The segmentation evaluation results are given in the Table 2. There are total of 10 videos with dynamic background movements such as tree branches swaying or water surface with moving waves are tested in this experiment. The overall evaluation results are shown in Table 3.

Table 3. Segmentation results comparison of dynamic background challenge

	Dynamic Background					
	recall	specificity	FPR	FNR	PWC %	Accuracy
Proposed	0.878	0.983	0.017	0.122	1.803	0.982
Original CB	0.711	0.901	0.099	0.289	10.153	0.898
GMM	0.684	0.969	0.031	0.316	3.504	0.965

Under the environment with illumination changes, there are 7 indoor videos that have been tested in this experiment. Overall results show that the proposed method managed to get better foreground segmentation than the other two methods. Segmentation evaluation results for videos with illumination changes are shown in Table 4.

Table 4. Segmentation results comparison of Illumination changes challenge

	Illumination Changes					
	recall	specificity	FPR	FNR	PWC %	Accuracy
Proposed	0.751	0.997	0.003	0.249	2.504	0.975
Original CB	0.193	0.961	0.039	0.807	10.752	0.892
GMM	0.459	0.882	0.118	0.541	15.580	0.844

There are four sequences that are captured under bad weather condition. All of the sequences were captured out door. Both GMM and original CB produce high PWC under bad weather condition because there are a lot of FNR pixels in the segmentation. The average scores comparison of sequences with bad weather challenges is shown in Table 5.

Table 5. Segmentation results comparison of bad weather challenge

	Bad Weather					
	recall	specificity	FPR	FNR	PWC %	Accuracy
Proposed	0.664	0.993	0.007	0.336	1.179	0.988
Original CB	0.136	0.996	0.004	0.864	1.637	0.984
GMM	0.530	0.968	0.032	0.470	3.883	0.961

Another challenging video contain moving persons and cars with their shadows to segment. There are total of four

videos, two indoor and two outdoor with shadow challenge. Among them, backdoor and cubicle sequences also contain illumination changes in the background which make them more difficult to get good segmentation result.

Table 6. Segmentation results comparison of Shadow challenge

	Shadow					
	recall	specificity	FPR	FNR	PWC %	Accuracy
Proposed	0.869	0.975	0.025	0.131	3.023	0.970
Original CB	0.748	0.948	0.052	0.252	6.186	0.938
GMM	0.354	0.971	0.029	0.646	6.016	0.940

The video with bootstrap challenge includes people walking around at the very start of the video which make the background modeling process more difficult to get clear background reference. The background model might include foreground object noise so foreground segmentation could become less accurate. All three methods cannot produce good foreground segmentation results. The original CB method wrongly classified most of the background pixels as foreground and thus produce high PWC%. As the original CB segments most of the background regions as foreground, it produces high recall rate than the other two methods. It can be seen that the color model which is used by the original CB cannot produce adequate results.

Table 7. Segmentation results comparison of Bootstrap challenge

	Bootstrap					
	recall	specificity	FPR	FNR	PWC %	Accuracy
Proposed	0.504	0.977	0.023	0.496	6.443	0.936
Original CB	0.763	0.678	0.322	0.237	31.461	0.685
GMM	0.319	0.955	0.045	0.681	10.003	0.900

With the video which include camouflage object, the color of the foreground object is so much similar with the background color. In the camouflage sequence, flickering computer screen in the background in covered by a person during segmentation.

Table 8. Segmentation results comparison of Camouflage challenge

	Camouflage					
	recall	specificity	FPR	FNR	PWC %	Accuracy
Proposed	0.946	0.954	0.046	0.054	5.026	0.960
Original CB	0.768	0.975	0.025	0.232	13.734	0.973
GMM	0.879	0.469	0.531	0.121	30.917	0.662

Proposed method gets good recall rate in all the challenges except in bootstrap sequences. During background modeling state, low resolution of the video and the presence of foreground movements effect the foreground segmentation accuracy. Original codebook gets low recall rates in illumination and bad weather challenges. The modifications on original codebook in proposed method produce better recall rate. Although recall rate of a method is high, if it has high PWC, the segmentation precision and accuracy are still low. Because many background pixels are wrongly classified as foreground. The results of GMM in camouflage challenge and original codebook in bootstrap challenge show how effectiveness of PWC on both precision and accuracy. Figure 2 shows the foreground segmentation of the three methods with the original video frame and the ground truth segmentation.

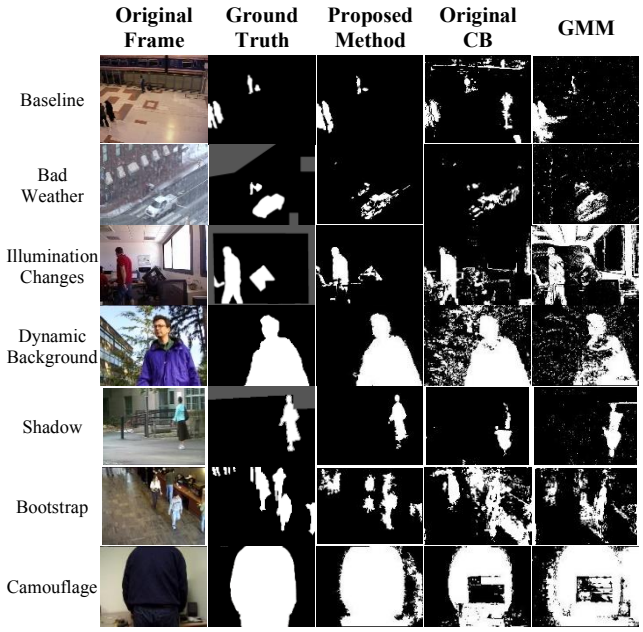

Figure 2. Foreground Segmentation Results Comparison of Three Methods

7. Conclusion

The modified codebook algorithm is proposed in this work for foreground region segmentation. Codebook provides the ability to initialize the background model in the presence of foreground objects. The main difference between the original codebook and the modified codebook is that the use of the color model to calculate the color distortion. L*a*b color space is closer to human perceptive of color difference. It can manage to get more accurate foreground segmentation results even if the gradual illumination changes occur. The experimental results also demonstrate that the proposed method

perform well under difference background subtraction challenges such as dynamic background, shadow and bad weather. Although proposed foreground segmentation method can handle gradual illumination change, it is unable to update the background model when sudden illumination change occurs.

8. References

[1]. Camplani, M., Maddalena, L., Alcover, G.M., Petrosino, A. and Salgado, L. "A benchmarking framework for background subtraction in RGBD videos". *In International Conference on Image Analysis and Processing,* Springer, Cham, 2017, September, pp. 219-229.

[2]. Donghai, H., Dan, Y., Xiaohong, Z. and Mingjian, H., "Principal component analysis-based codebook background modeling algorithm", *Acta Automatica Sinica,* 38(4), 2012, pp.591-600.

[3]. Doshi, A. and Trivedi, M., 2006, November. "Hybrid Cone-Cylinder Codebook Model for Foreground Detection with Shadow and Highlight Suppression", In *IEEE International Conference on Video and Signal Based Surveillance*, 2006. AVSS'06. pp. 19-19. IEEE.

[4]. Fang, X., Liu, C., Gong, S. and Ji, Y., "Object detection in dynamic scenes based on codebook with super-pixels". In *2nd Asian Conference on Pattern Recognition (ACPR)*, IAPR, IEEE, 2013 November (pp. 430-434).

[5]. Goyette, N., Jodoin, P.M., Porikli, F., Konrad, J. and Ishwar, P., "Changedetection. net: A new change detection benchmark dataset". In *CVPR Workshops* 2012, June, pp. 1-8.

[6]. Hu, H., Xu, L. and Zhao, H., "A spherical codebook in YUV color space for moving object detection", Sensor Letters, 10(1-2), 2012 pp.177-189.

[7]. KaewTraKulPong, P. and Bowden, R., "An improved adaptive background mixture model for real-time tracking with shadow detection", In *Video-based surveillance systems,* Springer, Boston, MA. 2002. pp. 135-144.

[8]. Kim, K., Chalidabhongse, T.H., Harwood, D. and Davis, L. "Background modeling and subtraction by codebook construction", In *International Conference on Image Processing*, ICIP'04, IEEE, 2004, October, 2004, Vol. 5, pp. 3061-3064.

[9]. Li, L., Huang, W., Gu, I.Y.H. and Tian, Q., "Statistical modeling of complex backgrounds for foreground object detection", *IEEE Transactions on Image Processing*, 13(11), 2004, pp.1459-1472.

[10]. Sharma, G., Wu, W. and Dalal, E.N.,"The CIEDE2000 color - difference formula: Implementation notes, supplementary test data, and mathematical observations". *Colour Society of Australia, Centre Français de la Couleur*, 30(1), 2005, pp.21-30.

[11]. Toyama, K., Krumm, J., Brumitt, B. and Meyers, B., 1999. "Wallflower: Principles and practice of background maintenance". *In the Proceedings of the Seventh IEEE International Conference on Computer Vision*, IEEE. 1999. Vol. 1, pp. 255-261.

[12]. Zeng, Z. and Jia, J., "Arbitrary cylinder color model for the codebook-based background subtraction". *Optics express*, 22(18), 2014, pp.21577-21588.