Background Subtraction and Foreground Detection based on Codebook Model with Kalman Filter

Su Su Aung, Zin Mar Kyu
University of Computer Studies, Mandalay, Myanmar
susuaung87@gmail.com , *zinmarkyu.pp@gmail.com*

Abstract

Foreground object extraction is an important subject for computer vision applications. The separation of foreground objects form the background is the crucial step in application such as video surveillance. In order to extract foreground object from a video scene, a background model which can represent dynamic changes in the scene is required. A robust, accurate and high performance approach is still a great challenge today. In this paper, the background modeling approach based on Codebook model with Kalman Filter is presented. This approach can be used to extract foreground objects from the video stream. The Lab color space is used in this approach to calculate color difference between two pixels using CIEDE2000 color difference formula. Extracted foreground object from video sequence using this approach is useful for object detection in video surveillance applications.

1. Introduction

In these days, the number of video-based surveillance systems is increasing due to its use in several applications, such as vision based traffic system, video segmentation or human behavior analysis. Most methods for video based surveillance rely on moving object detection [2]. Videos are actually sequences of images, each of which called a frame, displayed in fast enough frequency so that human eyes can percept the continuity of its content. The contents of two consecutive frames are usually closely related. Visual content can be modeled as a hierarchy of abstractions. At the first level are the raw pixels with color or brightness information. Further processing yields features such as edges, corners, lines, curves, and color regions. A higher abstraction layer may combine and interpret these features as objects and their attributes [10]. As a fundamental first step in many computer vision applications such as object tracking, behavior understanding, object or event recognition, and automated video surveillance, various algorithms have been developed ranging from simple approaches to more sophisticated ones. The first step in most of the automated surveillance application and video analysis is background subtraction. Background modeling, subtraction and estimations are the widely used techniques to extract foreground objects from background. However, there is still an issue according to inconsistent performance of method across different scenarios. Video sequences captured by fixed cameras contain moving objects on a fixed background. Real time segmentation of scene into objects and background is really important and represents an initial step of object tracking. The simplest way to model the background is to acquire background image without containing the moving objects in it. But in most of the circumstances, it is difficult to obtain the background because of the changes in the environment such as illumination changes and objects being introduced or removed from the scene. So background model must be more robust and adaptive. There are various background modeling approaches including Basic Background Modeling, Statistical Background Modeling, Fuzzy Background Modeling, Background Estimation and Prediction, Recursion, Adaptation, Modality [11].

In general, it is assumed that pixels belonging to foreground objects have different color values than background. Background modeling techniques model the background using previous frames history. Every image pixel is matched with its background model. If pixel color value is similar to the background model, then it is considered as background model otherwise it is an object pixel.

2. Codebook Background Modeling

The traditional codebook background modeling algorithm was proposed by [4] which were inspired by the algorithm presented in [Kohonen, 1988]. It is a quantization technique using long scene observation for each pixel. Each pixel was represented by one or more codeword and the number of codeword for a pixel is varies due to its background variation. Let χ be training sequences for a single pixel. Each codeword c_L, where L = 1...χ is represented by a RGB vector $v_L =$ $(\bar{R},\bar{G},\bar{B})$ and a six-tuples aux_L = $\langle \widecheck{I_L}, \widehat{I_L}, f_L, \lambda_L, p_L, q_L \rangle$. Where $\widecheck{I_L}$= min$\langle I, \widecheck{I_L} \rangle$ and $\widehat{I_L}$ = max $\langle I, \widehat{I_L} \rangle$are the minimum and the maximum brightness assigned to the codeword respectively. f_L is the frequency or the number of times that codeword is matched. λ_L is the maximum negative run length, meaning the largest time span in which this codeword is not updated or accessed. p_L and q_L are the first and the last access times of the codeword respectively.

Algorithm for codebook construction [5].

```
Input: Stream of pixel values (R, G, B)
Output: $M$ (codebook)
Initialize $L \leftarrow 0$ and $M \leftarrow \emptyset$
for $t = 1$ to $N$ do
    $X_t \leftarrow (R, G, B)$, $I \leftarrow \sqrt{R^2 + G^2 + B^2}$.
    Find the codeword $c_m$ in $C$ matching to $X_t$
    using Eq. (1), (3) and (4)
    if colordist($X_t$, $V_i \leq \epsilon_1$  AND
        brightness ($I$, $\langle \check{I}, \hat{I} \rangle$) = true then
        Update the codeword cm as follows:
        $V_m \leftarrow \left( \frac{f_m \overline{R_m} + R}{f_m + 1}, \frac{f_m \overline{G_m} + G}{f_m + 1}, \frac{f_m \overline{B_m} + B}{f_m + 1} \right)$
        $aux_m \leftarrow \langle \min\{I, \check{I}_m\}, \max\{I, \hat{I}_m\}, f_m + 1, \max\{\lambda_m, t - q_m\}, p_m, t \rangle$
    end
    else if $M = \emptyset$ or there is no match
    then Increment $L$ and create a new codeword
        $c_L = (V_L, aux_L)$ by setting,
        $V_L \leftarrow (R, G, B)$ and $aux_L \leftarrow \langle I, I, 1, t - 1, t, t \rangle$
        Add $c_L$ in $M$.
    end
end
foreach Codeword $c_i$ in $M$ do
    $\lambda_i \leftarrow max \{\lambda i, (N - q_i + p_i - 1)\}$
end
```

2.1. Color and brightness distortion

When we have an input pixel X_t = (R, G, B) and a codeword c_i where $v_i = (\bar{R_i}, \bar{G_i}, \bar{B_i})$,

$\|X_t\|^2 = R^2 + G^2 + B^2$,

$\|v_i\|^2 = \bar{R_i}^2 + \bar{G_i}^2 + \bar{B_i}^2$,

$\langle X_t, v_i \rangle^2 = (\bar{R_i}R + \bar{G_i}G + \bar{B_i}B)^2$

The color distortion is computed as follows:

$$colorDistortion(X_t, v_i) = \delta = \sqrt{\|X_t\|^2 - P^2} \qquad (1)$$

where P^2 is calculated as follows :

$$P^2 = \|X_t\|^2 \, cos^2\, \theta = \frac{\langle X_t, v_i \rangle^2}{\|v_i\|^2} \qquad (2)$$

On the other hand, the brightness range (I_{low} , I_{hi}) is calculated using the min ($\check{I}$) and the max ($\hat{I}$) as follows:

$$I_{low} = \alpha \check{I}\ ,\ I_{hi} = min\ (\beta \hat{I}, \frac{I}{\alpha}), \qquad (3)$$

Where α and β are the factors used to extend the brightness bound to changes due to the illumination changes. By maintaining I_{low} and I_{hi} values in the codebook, local illumination changes such as shadow and highlight can be detected. I_{low} and I_{hi} values can be updated throughout the training period to cover a certain range of brightness variation.

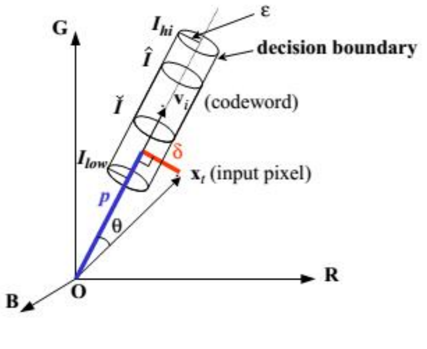

Figure 1. Cylinder color model of Codebook algorithm

The technique alone however cannot be sufficiently used to filter a wide range of code-words in codebooks. Moreover, moving a codeword into a codebook merely on the basis of minimum and maximum intensity comparison is not sufficient to identify foreground pixels from background pixels especially when both assume similar color information in addition to similar intensity values [9].

3. Improved Codebook with Lab color model and Kalman Filter

In this proposed method, the background is modeled by codebook algorithm. But the performance of traditional codebook algorithm is highly depend on the cylinder color model which is valid only if the spectrum components of the light source change in the same proportion. In fact, this is not true in many practical cases. [9]. So we use Lab color model instead of RGB color model and compute the color distortion between two pixels values using *CIEDE2000 color difference formula.* Kalman Filter is used to estimate intensity value of each pixel in order to track brightness variation throughout the sequence.

3.1. Lab color model

Lab color space is a color-opponent space with dimension ***L*** for lightness and ***a*** and ***b*** for the color-opponent dimensions, based on nonlinearly compressed coordinates [12]. Unlike the RGB and CMYK color models, Lab color is designed to approximate human vision. The Lab color space encompasses all the colors that human eye can see, which means that its gamut exceeds those of the RGB and CMYK color models. One of the most important attributes of the Lab model is device independence. This means that the colors are defined independent of their nature of creation or the device they are displayed on. Lab is also extremely useful for translating color from one real world condition to another.

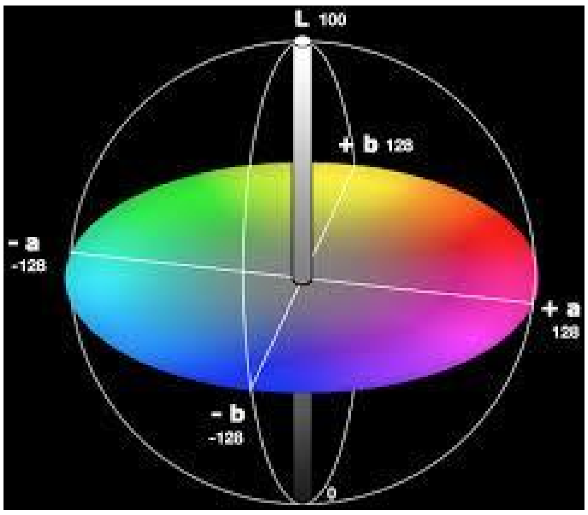

Figure 2. Lab color space[13]

In Lab color space, the vertical ***L*** axis represents Lightness, ranging from 0-100. The other (horizontal) axes are now represented by ***a*** and ***b***. These are at right angles to each other and cross each other in the centre, which is neutral (grey, black or white). They are based on the principal that a color cannot be both red and green, or blue and yellow. The ***a*** axis is green at one extremity (represented by -*a*), and red at the other (+*a*). The ***b*** axis has blue at one end (-*b*), and yellow (+*b*) at the other.

3.1.1. CIEDE2000 Color-difference Formula

The CIEDE2000 color-difference formula [1] was developed members of CIE (Commission Internationale de l'Eclairage which in English is the International Commission on Illumination) Technical Committee. The formula provides an improved procedure for the computation of industrial color differences. The color difference deltaE (ΔE) is generally used for the color difference evaluation with CIE Lab color space. The CIEDE2000 formula is considerably more sophisticated and computationally involved than its predecessor color difference equations for CIE76 and the CIE94 color difference.

The color difference between two Lab color values, Lab$_1$ and Lab$_2$ can be denoted as follow:

$$\Delta E_{00}(Lab_1, Lab_2) = \Delta E_{00}^{12} = \Delta E_{00} \qquad (4)$$

There are three main steps in computing of the color difference given two Lab color values and parametric weighting factors k$_L$, k$_C$ and k$_H$.

$$\Delta E_{00} = \sqrt{\left(\frac{\Delta L'}{k_L S_L}\right)^2 + \left(\frac{\Delta C'}{k_{LC} S_C}\right)^2 + \left(\frac{\Delta H'}{k_H S_H}\right)^2 + R_T \frac{\Delta C'}{k_{LC} S_C}\frac{\Delta H'}{k_H S_H}}$$

First step is to calculate C_i', h_i'

$$C_{i,ab}^{*} = \sqrt{(a_i^*)^2 + (b_i^*)^2} \qquad (5)$$

$$\bar{C}_{ab}^{*} = \frac{C_{l,ab}^{*} + C_{l,ab}^{*}}{2} \qquad (6)$$

$$G = 0.5\left(1 - \sqrt{\frac{\bar{C}_{ab}^{*7}}{\bar{C}_{ab}^{*7} + 25^7}}\right) \qquad (7)$$

$$a_i' = (1 + G)a_i^* \qquad i = 1, 2 \qquad (8)$$

$$C_i' = \sqrt{(a_i')^2 + (b_i^*)^2} \qquad i = 1, 2 \qquad (9)$$

$$h_i' = \begin{cases} 0 & b_i^* = a_i' = 0 \\ tan^{-1}(b_i^*, a_i') & otherwise \end{cases} \qquad i = 1, 2 \qquad (10)$$

Second step is to calculate $\Delta L', \Delta C', \Delta H'$

$$\Delta L' = L_2^* - L_1^* \qquad (11)$$

$$\Delta C' = C_2' - C_1' \qquad (12)$$

$$\Delta h' = \begin{cases} 0 & C_1'C_2' = 0 \\ h_2' - h_1' & C_1'C_2' \neq 0;\ |h_2' - h_1'| \leq 180^\circ \\ (h_2' - h_1')\text{-}360 & C_1'C_2' \neq 0;\ |h_2' - h_1'| > 180^\circ \\ (h_2' - h_1') + 360 & C_1'C_2' \neq 0;\ |h_2' - h_1'| < -180^\circ \end{cases}$$

(13)

$$\Delta H' = 2\sqrt{C_1'C_2'}\,sin\left(\frac{\Delta h'}{2}\right) \tag{14}$$

Third step is the calculation of deltaE (ΔE_{00})

$$\bar{L}' = (L_1^* + L_2^*)/2 \tag{15}$$

$$\bar{C}' = (C_1' + C_2')/2 \tag{16}$$

$$h_1' = \begin{cases} \frac{h_1' + h_2'}{2} & |h_2' - h_1'| \le 180^\circ;\ C_1'C_2' \ne 0 \\ \frac{h_1' + h_2' + 360^\circ}{2} & |h_2' - h_1'| > 180^\circ; (h_2' + h_1') < 360^\circ; \\ & C_1'C_2' \ne 0 \\ \frac{h_1' + h_2' - 360^\circ}{2} & |h_2' - h_1'| > 180^\circ; \\ & (h_2' + h_1') \ge 360^\circ\ ;C_1'C_2' \ne 0 \\ (h_2' + h_1') & C_1'C_2' = 0 \end{cases} \tag{17}$$

$$\mathrm{T} = 1 - 0.17\cos(\bar{h}' - 30^\circ) + 0{,}24\cos(2\bar{h}') + 0.32\cos(3\bar{h}' + 6^\circ) - 0.20\cos(4\bar{h}' - 63^\circ) \tag{18}$$

$$\Delta\theta = 30\,exp\left\{-\left[\frac{\bar{h}' - 275^\circ}{25}\right]\right\} \tag{19}$$

$$R_C = 2\sqrt{\frac{\bar{C}'^7}{\bar{C}'^7 + 25^7}} \tag{20}$$

$$S_L = 1 + \frac{0.015(\bar{L}' - 50)^2}{\sqrt{20 + (\bar{L}' - 50)^2}} \tag{21}$$

$$S_C = 1 + 0.045\bar{C}' \tag{22}$$

$$S_C = 1 + 0.015\bar{C}'T \tag{23}$$

$$R_C = -\sin(2\Delta\theta)R_C \tag{24}$$

If the deltaE (ΔE_{00}) value between current pixel's Lab value and Lab value stored in Codeword is greater than the threshold (ϵ_1), then the current pixel is belong to the foreground object. This method show more accurate foreground-background segmentation result compare with the traditional codebook which use color distance formula.

3.1.2. Kalman Filter Intensity Estimation

The Kalman filter is essentially a set of mathematical equations that implement a predictor-corrector type estimator that is optimal in the sense that it minimizes the estimated error covariance when some presumed conditions are met. Since the time of its introduction, the Kalman filter has been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation. This is likely due in large part to advances in digital computing that made the use of the filter practical, but also to the relative simplicity and robust nature of the filter itself. Rarely do the conditions necessary for optimality actually exist, and yet the filter apparently works well for many applications in spite of this situation [6].

Kalman filter can be used in any place where there is uncertain information about some dynamic system, and can make an accurate estimation about what the system is going be next. There are five steps in Kalman filter equation [3] including state prediction, error prediction, kalman gain, state correction, and error correction. The following parameters are needed to define in order to perform Kalman filter estimation:

$$\hat{x}_k^- = A\hat{x}_{k-1}^- + Bu_k \tag{25}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{26}$$

$$K_k = P_k^- H^T(HP_k^- H^T + R)^{-1} \tag{27}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{28}$$

$$P_k = (I - K_k H)P_k^- \tag{29}$$

Where,

A matrix relates the state at the previous time step to the state at the current step.
B matrix relates the optional control input to the state.

Q represents the process noise covariance matrix.

H matrix relates the state to the measurement.

R represents the measurement noise covariance matrix.

x is the state variable with k being the current and k-1 being the prior.

u is the control variable with k being the current.

z is the measurement with k being the current.

In the proposed method, we estimate the intensity value of each pixel from its previous value and compare the estimated result with actual measure intensity. Kalman filter is applied on each pixel to track pixel intensity through the video sequence. In the background modeling process, the prediction-correction result from kalman filter is stored in Codeword of each pixel. After background modeling, kalman filter estimate intensity value of each pixel and pixels with high intensity variances are segmented to the foreground.

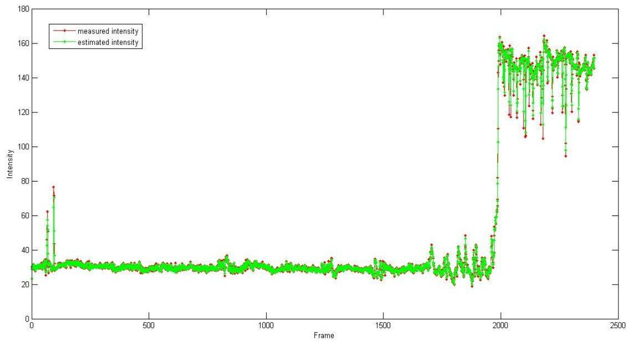

Figure 3. Estimated and measured intensity values of one pixel through video sequences

In the estimation process, the measurement noise is calculated by the Noise Level Estimation method proposed in [7]. Kalman filter can solve the problem of traditional Codebook algorithm which is caused by minimum and maximum brightness range in cylinder color model.

4. Experimental Results

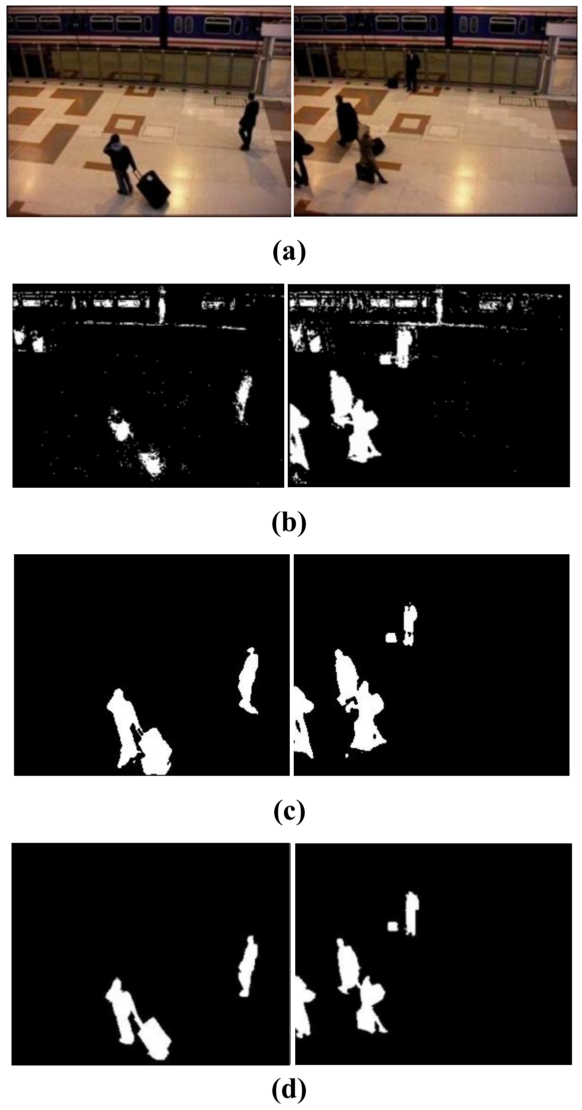

Figure 4. Background Subtraction result on PETS dataset by (b) Traditional Codebook Algorithm (c) Proposed Method (d) Ground Truth

The proposed method is tested on PETS dataset. The experimental background subtraction and foreground segmentation results show that the proposed method can produce more accurate results compare with the traditional codebook algorithm.

5. Conclusion

In this paper, we proposed robust background modeling and foreground detection method using improved Codebook model with Kalman filter. In background modeling, we use CIEDE2000 color difference formula to calculate color deviation between two pixels. Kalman filter estimation is also used to estimate pixel intensity and foreground pixels are extracted by comparing estimated and measured intensity values. This method can provide better foreground segmentation result which can be used in object detection applications.

References

[1] G. Sharma, Wencheng Wu and Edul N. Dalal, "The CIEDE2000 Color-Difference Formula: Implementation Notes, Supplementary Test Data, and Mathematical Observations", 19th IEEE International Conference on Image Processing (ICIP), 2012

[2] J. Kim, A. Ramirez Rivera, B. Ryu and O. Chae, "Simultaneous foreground detection and classification with hybrid features", conference on International Computer Vision (ICCV), December 13-16, 2015.

[3] J. Scott, M. A. Pusateri and D. Cornish, "Kalman Filter Based Video Background Estimation", in the proceeding on Applied Imagery Pattern Recognition Workshop (AIPRW), 2009 IEEE 14-16 Oct. 2009.

[4] K. Kim, T. H. Chalidabhongse, D.Harwood and L. Davis, "Real-time foreground – background segmentation using codebook model", in Conference on Visual Communication and Image Processing 2005.

[5] M. Shah, J. D. Deng, B. J. Woodford, "Self-Adaptive CodeBook (SACB) model for real-time background subtraction", Journal of Image and Vision Computing, October 3, 2013.

[6] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", Journal of Basic Engineering, 1960, pp 35-45.

[7] X. Liu, M. Tanaka and M. Okutomi, "Noise Level Estimation Using Weak Textured Patches of a Single Noisy Image", 19th IEEE International Conference on Image Processing (ICIP), September 2012.

[8] Y. A. Syed, S. Shetty, N. Wilkinson, and D. J. Brown, "A Modified Codebook-based Background Subtraction Technique to improve Activity Classification in Highly Variable Environments", STS Defense Ltd and University of Portsmouth, 2012.

[9] Z. Zeng and J. Jia, "Arbitrary cylinder color model for the codebook based background subtraction", Optical Society of America (OSA), September 2014.

[10] Z. Guo, "Object Detection and Tracking in Video", Journal of Advances in Internet based Systems and Application, November 2001.

[11] T. Bouwmans, "Recent Advanced Statistical Background Modeling for Foreground Detection - A Systematic Survey", Recent Patents on Computer Science, September 2011.

[12] https://en.wikipedia.org/wiki/Lab_color_space

[13]http://t3.gstatic.com/images?q=tbn: ANd9GcSPlYkqiL_XiYV5WcEM1owcFT9A0iYbKAbR7OlutLJjsb6H1W6oZQ