Modified Codebook Algorithm with Kalman Filter for Foreground Segmentation in Video Sequences

Su Su Aung
University of Computer Studies,
Mandalay, Myanmar
susuaung87@gmail.com

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

Abstract—**Background subtraction method is widely used in most of the video motion detection algorithms especially for video surveillance application. Background subtraction is used to extract moving or static foreground objects from the background scene. The efficiency of foreground-background segmentation heavily relies on background model which must be able to cope with changes in the scene and granularity of the foreground objects. The robust background model can produce good foreground segmentation results and it is still a great challenge to get accurate and high performance result today. In this paper, a video foreground-background segmentation approach is proposed. This approach is based on Codebook (CB) model with Kalman Filter. 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. This approach has been tested with PETS and CDnet2014 datasets and segmentation results accuracy are evaluated compare with ground truth.**

Keywords— Background Modeling; Foreground Detection; Codebook; Kalman Filter.

I. INTRODUCTION

In these days, there are increasing demands for video based automated surveillance system in public place such as airport and stations. These surveillance systems can be used in several applications including traffic monitoring, human action recognition, anomaly detection and semantic indexing of video. Most of these system start with moving object detection in given scene before any higher level object recognition and classification steps can be performed. Background subtraction is most popular choice for the detection of motion in many current visual surveillance systems to extract foreground objects from the current frame because they work reasonably well when the camera is stationary and the illumination changes in the environment is gradual. Scene background modeling plays an important role of background subtraction. In the next step, object classification should be performed which allows distinguishing various object types. Such an operation allows for further analysis depending on the object category.

Video contain a sequence of images call frames and contents of two continuous frames are usually closely related. The visible information of image such as color values of pixel or intensity information can be model at lower level processing. The higher processing can yield more advance feature such as corners, curves, line, edges, and color regions. As a first step, Background initialization using N frames to obtain the first background image without the moving objects. 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. As step two, the moving object detection is made through the foreground detection that consists in classifying pixels as foreground or background by comparing the background image and the current frame. As step three, Background maintenance to update the background image over time. The steps two and three are executed repeatedly as time progresses [R1].

Gaussian Mixture Model is the most popular choice for modeling the background and foreground. GMM can be use for pixel level analysis. The existing methods can be divided into two categories based on the use of one or more background subtraction models. In the first categories, some researcher made frame-by-frame analysis which is based on the typical background subtraction techniques then followed by another type of analysis (e.g., Tracking). Such approach is use in [R2]. Their approach differs in at least two major ways from previously reported work. First, they analyze relationships between objects. Second, they have exploited multiple cameras with overlapping fields of view to cope with occlusions of various types, and have empirically observed this to be essential in realistic situations. The approach in [R3] maintains two backgrounds one for long term and the other for short term, both of them are modeled using GMM. And for every frame, long and short term foregrounds are estimated by comparing the current frame by the background models BL (Long term background) and BS(short term background). The long term foreground mask FL(Long term foreground) shows the color variations in the scene that were not there before including moving objects, temporarily static objects, as well as moving cast shadows and illumination changes that the background models fail to adapt. The short-term foreground mask FS (Short term foreground) contains the moving objects, noise, and so forth. Adaptation of the long- and short term background models by a Bayesian update mechanism.

II. Related Works

Codebook (CB) based background modeling and foreground detection method was proposed by [4]. In this method, the sampled background pixel values are quantized into codebooks array that can represents the compact background model. This method is pixel level analysis and represents each pixel with codebook which contains one or more codewords according to their color and brightness variation during the background training process. It shown that basic CB model using less computational complexity and less memory capacity has better result in foreground-background segmentation with respect to MOG and Kernel [5]. But the original codebook model which use cylindrical color decision boundary cannot represent the color distribution of a pixel. So, in [6] the ellipsoidal CodeWord (CW) model in RGB space was used to represent the distribution of the background pixel. They combine MOG model with the CB model by replacing the cylindrical CW by an ellipsoidal CW. [7] also used arbitrary cylinder color model to cover the weakness of original cylinder color model.

Modeling background with codebook can be classified as cluster background model. There are various cluster background model such as K-means model and basic sequential clustering model. [8] proposed an algorithm which assigns each pixel of incoming frame to group of clusters to each pixel in the frame. The clusters are ordered according to the likelihood that they model the background and are adapted to deal with background and lighting variations. Incoming pixels are matched against the corresponding cluster group and are classified according to whether the matching cluster is considered part of the background or not. The proposed method in [9] construct adaptive background model from sequence of images by using Gaussian Mixture Model and K-Means clustering technique. In this technique, the intensity distribution of a pixel is analyzed by GMM over some period. The Gaussian distribution is divided into two cluster, foreground and background by K-Means clustering technique. The method in [10] modified K-means clustering algorithm to choose initial centroids of randomly selected k objects. The aim of the proposed method is to find certain initial threshold which consistent with the distribution of data because the standard k-means algorithm gets easily trapped in a local minimum and different initial threshold lead to different results. In [11] the moving object extraction method based on Genetic K-means algorithm from Gaussian Mixture Model (GMM) is proposed. This method takes the advantages of Genetic K-means (GKA) algorithm over traditional k-means algorithm in order to overcome lack of global optimization problem caused by the randomness. Background modeling based on GKA could establish the background under complex scenes in a higher speed.

III. Codebook Based Background Modeling

The traditional codebook background modeling algorithm was proposed by [12] and it was based on the algorithm proposed by Kohonen in 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 N be training sequences for a single pixel. Each codeword c_L, where L = 1... N is consist of color vector v_L = ($\bar{R}$,$\bar{G}$,$\bar{B}$) with RGB values of pixel and a six-tuples aux_L = $\langle \breve{I_L}, \widehat{I_L}, f_L, \lambda_L, p_L, q_L \rangle$. Where $\breve{I_L}$= $\min\langle I, \breve{I_L} \rangle$ and $\widehat{I_L}$ = max $\langle I, \widehat{I_L} \rangle$are the minimum and the maximum brightness range 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 which means 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.

Algorithm1. Modified Codebook Algorithm

Input: Stream of pixel values (R, G, B)

Output: C (codebook)

Initialize $L \leftarrow 0$ and $M \leftarrow \emptyset$

for each pixel in I_t **do** (t = 1 to N)

$X_t \leftarrow (R, G, B)$,

$Lab_t = rgb\_to\_lab(X_t)$

$\hat{x}_t$, *foreground* $\leftarrow kalman\_estimation(X_t, X_{t-1})$

Find the codeword c_m in C matching to $\hat{x}_t$ using Eq. (5) for delta E

if $\Delta E \leq \epsilon_1$ AND *foreground* = *false* **then**

Update the codeword c_m as follows:

$V_m \leftarrow \left(\frac{f_m \overline{L_m} + L}{f_m + 1}, \frac{f_m \overline{a_m} + a}{f_m + 1}, \frac{f_m \overline{b_m} + b}{f_m + 1} \right)$

$aux_m \leftarrow \langle \min\{\hat{x}_t, \check{I}_m\}, \max\{\hat{x}_t, \hat{I}_m\}, f_m + 1, \max\{\lambda_m, t - q_m\}, p_m, t \rangle$

else if $C = \emptyset$ or there is no match **then**

Increment L by one and create a new codeword $c_L = (V_L, aux_L)$ by assigning,

$V_L \leftarrow (Lab_t)$ and $aux_L \leftarrow \langle X_t, X_t, 1, t-1, t, t \rangle$

Add c_L in C.

end if

end for

for each Codeword c_i in C **do**

$\lambda_i \leftarrow max\ \{\lambda_i, (N - q_i + p_i - 1)\}$

end for

In this proposed approach, the background is modeled by modified codebook algorithm. 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. [13].So, we proposed modified Codebook algorithm to model the background and extract the foreground objects. In the proposed approach, 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.

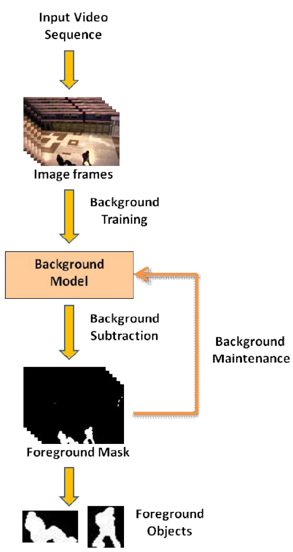


Fig. 1. System Diagram

A. Color and brightness distortion

In order to determine the incoming pixel is foreground or background pixel, we have to calculate pixel's color and brightness distortion. If the distortions are within tolerable range, then we can determine the incoming pixel is background and we must update the background model with new values. In the traditional codebook model, 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:

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

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.

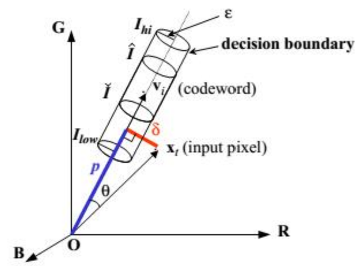


Fig. 2. Cylinder color model of Codebook algorithm [8]

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 [13].

B. CIEDE2000 Color-difference Formula

The CIEDE2000 color-difference formula [14] was developed members of CIE (Commission Internationale de l'Eclairage which in English is the International Commission on Illumination) Technical Committee. In 1976 the CIE published the first internationally endorsed color differencing equation. This formula called ΔEab or ΔE76 specified a difference or ΔE of 1.0 to be the smallest difference perceivable by the human eye [15]. 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 CIELAB color space. But later found out that the sensitivity of human eye is different over regions of color wheel. They fail to measure the visual perception of the color differences sufficiently although they are said to be uniform color difference formulae [16]. In 2001, the CIE recommended the CIEDE2000 formula based on the CIELAB to improve the correlation between measured and human observed color differences.

The color difference between two Lab color values, Lab_1 and Lab_2 can be denoted as follow:

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

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.

$$\Delta E_{00} = \sqrt{\left(\frac{\Delta L'}{k_L S_L}\right)^2 + \left(\frac{\Delta C'}{k_{LC} S_C}\right)^2 + \left(\frac{\Delta H'}{k_H S_H}\right)^2 + R_T \frac{\Delta C'}{k_{LC} S_C}\frac{\Delta H'}{k_H S_H}} \quad (5)$$

First step is to calculate C_i', h_i'

$$C_{i,ab}^* = \sqrt{(a_i^*)^2 + (b_i^*)^2} \quad (6)$$

$$\bar{C}_{ab}^* = \frac{C_{l,ab}^* + C_{l,ab}^*}{2} \quad (7)$$

$$G = 0.5\left(1 - \sqrt{\frac{\bar{C}_{ab}^{*7}}{\bar{C}_{ab}^{*7} + 25^7}}\right) \quad (8)$$

$$a_i' = (1 + G)a_i^* \qquad i = 1, 2 \quad (9)$$

$$C_i' = \sqrt{(a_i')^2 + (b_i^*)^2} \qquad i = 1, 2 \quad (10)$$

$$h_i' = \begin{cases} 0 & b_i^* = a_i' = 0 \\ tan^{-1}(b_i^*, a_i') & otherwise \end{cases} \qquad i = 1, 2 \quad (11)$$

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

$$\Delta L' = L_2^* - L_1^* \quad (12)$$

$$\Delta C' = C_2' - C_1' \quad (13)$$

$$\Delta h' = \begin{cases} 0 & C_1'C_2' = 0 \\ h_2' - h_1' & C_1'C_2' \neq 0;\ |h_2' - h_1'| \leq 180^\circ \\ (h_2' - h_1') - 360 & C_1'C_2' \neq 0;\ |h_2' - h_1'| > 180^\circ \\ (h_2' - h_1') + 360 & C_1'C_2' \neq 0;\ |h_2' - h_1'| < -180^\circ \end{cases} \quad (14)$$

$$\Delta H' = 2\sqrt{C_1'C_2'}\, sin\left(\frac{\Delta h'}{2}\right) \quad (15)$$

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

$$\bar{L}' = (L_1^* + L_2^*)/2 \quad (16)$$

$$\bar{C}' = (C_1' + C_2')/2 \quad (17)$$

$$h_1' = \begin{cases} \frac{h_1' + h_2'}{2} & |h_2' - h_1'| \leq 180^\circ;\ C_1'C_2' \neq 0 \\ \frac{h_1' + h_2' + 360^\circ}{2} & |h_2' - h_1'| > 180^\circ; (h_2' + h_1') < 360^\circ;\ C_1'C_2' \neq 0 \\ \frac{h_1' + h_2' - 360^\circ}{2} & |h_2' - h_1'| > 180^\circ;\ (h_2' + h_1') \geq 360^\circ;\ C_1'C_2' \neq 0 \\ (h_2' + h_1') & C_1'C_2' = 0 \end{cases} \quad (18)$$

$$T = 1 - 0.17\cos(\bar{h}' - 30^\circ) + 0{,}24\cos(2\bar{h}') + 0.32\cos(3\bar{h}' + 6^\circ) - 0.20\cos(4\bar{h}' - 63^\circ) \quad (19)$$

$$\Delta\theta = 30\ exp\left\{-\left[\frac{\bar{h}' - 275^\circ}{25}\right]\right\} \quad (20)$$

$$R_C = 2\sqrt{\frac{\bar{C}'^7}{\bar{C}'^7 + 25^7}} \quad (21)$$

$$S_L = 1 + \frac{0.015(\bar{L}' - 50)^2}{\sqrt{20 + (\bar{L}' - 50)^2}} \quad (22)$$

$$S_C = 1 + 0.045\bar{C}' \quad (23)$$

$$S_C = 1 + 0.015\bar{C}'T \quad (24)$$

$$R_C = -\sin(2\Delta\theta)R_C \quad (25)$$

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.

TABLE I. DELTA-E COLOR DIFFERENCE THRESHOLDS [17]

Delta E	**Perception**
<= 1.0	Not perceptible by human eyes.
1 - 2	Perceptible through close observation.
2 - 10	Perceptible at a glance.
11 - 49	Colors are more similar than opposite
100	Colors are exact opposite

IV. INTENSITY ESTIMATION WITH KALMAN FILTER

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 [18].

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 [19] including state prediction, error prediction, kalman gain, state correction, and error correction.

In the proposed algorithm, 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.

Algorithm2. Brightness Estimation with Kalman Filter

Input: rgb_{t-1}, rgb_t (t = 1 to N)

Output: $\hat{x}_t$, *foreground*

procedure $kalman\_estimation(rgb_t, rgb_{t-1})$

E_{xt} = *covariance* (rgb_{t-1}, rgb_t) //covariance between two pixel's RGB values

$\hat{P}_t = E_{xt}$ // initial variance

$\hat{x}_k^- = A\hat{x}_{t-1}^-$ // brightness prediction

if $|X_t - \hat{x}_t^-|$ > threshold, **then**

foreground = true

else *foreground = false*

end if

$\hat{P}_t = APA' + E_{xt}$ // predict next covariance

$K_t = \frac{P_t C'}{(C P_t C' + E_z)}$ // compute Kalman gain

$\hat{x}_t = \hat{x}_t^- + K_t(X_t - C\hat{x}_t^-)$ // estimate brightness correction

$P_t = (I - K_t C)\hat{P}_t$ // update covariance estimation

return $\hat{x}_t$, *foreground*

end procedure

Where,

A matrix relates the state at the previous time step to the state at the current step.
$\boldsymbol{E_x}$ represents the process noise covariance matrix.
C matrix relates the state to the measurement.
$\boldsymbol{E_z}$ represents the measurement noise covariance matrix.
$\boldsymbol{\hat{x}_t}$ is the state variable with *t* being the current and *t-1* being the prior.
$\boldsymbol{X_t}$ is the measurement with *t* being the current.

In the estimation process, the measurement noise (E_z) is calculated by the Noise Level Estimation method proposed in [20]. Kalman filter can solve the problem of traditional Codebook algorithm which is caused by minimum and maximum brightness range in cylinder color model.

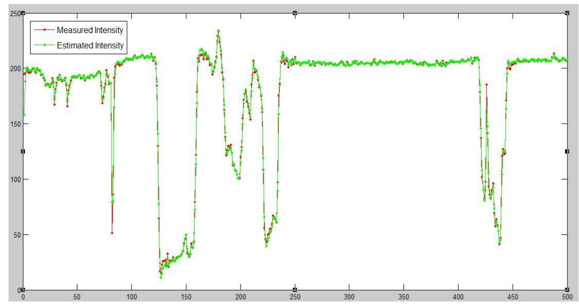
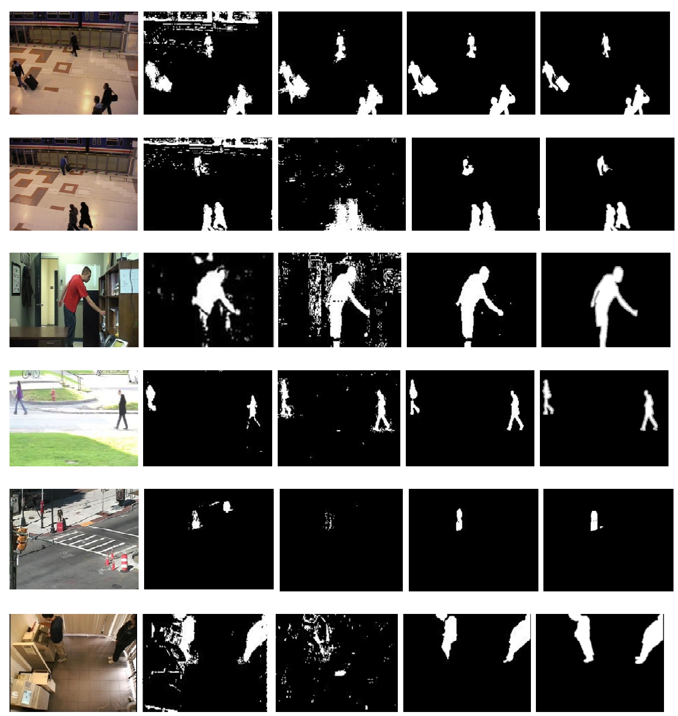


Fig. 3. Estimated and measured intensity values of a particular pixel through 500 frames

V. EXPERIMENTIAL RESULTS

The proposed approach is tested with several video sequences with difference challenges. These video sequences were obtained by using stationary camera. We demonstrated our approach with both indoor and outdoor scenarios. The segmentation results are shown in figure 4 compare with the traditional CB method, GMM method and Ground Truth. According to the experimental results, the proposed approach perform well for difference scenarios while the other two fail to produce good segmentation results under the illumination variation and dynamic background environments.



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

Fig. 4. Background Subtraction results (a) Original Frame (2) Traditional CB Method (c) GMM Model (d) Proposed Method (e) Ground Truth

Each segmented foreground mask is compare to the ground truth and the segmentation accuracy score is computed by using the standard *intersection-over-union* (IUR) ratio metric. The experimental background subtraction and foreground segmentation results of six test video sequences are described in the table below. It shows that the proposed approach can produce more accurate results compare with the traditional codebook algorithm and GMM model.

TABLE II. THE SEGMENTATION ACCURACY SCORE OF TRADITIONAL CODEBOOK, GMM AND PROPOSED METHOD COMPARE WITH GROUNDTRUTH

Dataset	IUR Accuracy Score		
	Traditional CB Method	***GMM Method***	***Proposed Method***
PETS_Abandon	0.256876	0.565213	0.731334
PETS2006	0.149311	0.282596	0.566646
CDnet2014_office	0.467658	0.134342	0.854405
CDnet2014_Pedestrians	0.479504	0.692146	0.613926
CDnet2014_abandonedbox	0.343836	0.161523	0.659212
CDnet2014_CopyMachine	0.528041	0.072467	0.743382

VI. CONCLUSION

In this paper, we proposed robust background modeling and foreground detection approach 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 integration of Codebook model with Kalman filter estimation method can provide more accurate foreground segmentation result which can be used in object detection applications.

REFERENCES

[1] Thierry Bouwmans, "Traditional and Recent Approaches in Background Modeling for Foreground Detection: An Overview, ". Computer Science Review, Elsevier, April 18, 2014, pp.31-66

[2] Michael D. Beynon, Daniel J. Van Hook, Michael Seibert, Alen Peacock, Dan E. Dudgeon, "Detecting Abandoned Packages in a Multi-Camera Video Surveillance System, ". AVSS ,2003, pp.221-228

[3] Chuan-Yu Cho, Wei-Hao Tung and Jia-Shung Wang, "A Crowd-Filter for Detection of Abandoned Objects in Crowded Area, " . 3rd International Conference on Sensing Technology, ICST, 2008, pp.581 – 584

[4] Kyungnam Kim, Thanarat H. Chalidabhongseb, David Harwood, Larry Davis, "Real-time foreground–background segmentation using codebook model, ". Conference on Visual Communication and Image Processing, 2005

[5] Mohamad Hoseyn Sigari, Mahmood Fathy, "Real-time Background Modeling/Subtraction using Two-Layer Codebook Model, ". International MultiConference of Engineers and Computer Scientists, Vol I, Hong Kong, 2008, pp. 19-21

[6] Ting Sun, Shih-Chung Hsu, and Chung-Lin Huang, "A Hybrid Codebook Model for Foreground Object Segmentation and Shadow/Highlight Removal, ".

[7] Zhi Zeng, Jianyuan Jia, "Arbitrary cylinder color model for the codebook based background subtraction, ". Optical Society of America, Vol.22, No.18, 2014.

[8] Darren Butler, Sridha Sridharan, V. Michael Bove Jr, "Real Time Adaptive Background Segmentation, ". S. EURASIP J. Adv. Signal Process, 2005

[9] Theekapun Charoenpong, Ajaree Supasuteekul, Chaiwat Nuthong, "Adaptive Background Modeling from an Image Sequence by Using K-Means Clustering, ". Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010

[10] Kyungnam Kim, Thanarat H. Chalidabhongseb, David Harwood, Larry Davis, "Real-time foreground–background segmentation using codebook model," . Conference on Visual Communication and Image Processing (2005)

[11] Duan Xiuman Sun, Guoxia Yang Tao, "Moving Target Detection Based on Genetic K-means Algorithm,". The 13th International Conference on Communication Technology (ICCT), 2011

[12] Kyungnam Kim, Thanarat H. Chalidabhongse, David Harwood, Larry Davis, "Background Modeling and Subtraction by Codebook Construction,". Internal Conference on Image Processing (ICIP), 2004

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

[14] Gaurav Sharma, Wencheng Wu,Edul N. Dalal, "The CIEDE2000 Color-Difference Formula: Implementation Notes, Supplementary Test Data, and Mathematical Observations,". Vol 30,2005, pp. 21-30

[15] Martin Habekost, "Which color differencing equation should be used,". International Circular of Graphic Education and Research, No. 6, 2013

[16] Dibakar Raj Pant, Ivar Farup, "Riemannian Formulation of the CIEDE2000 Color Difference Foumula,". 18th Color Imaging Conference, 2010

[17] http://zschuessler.github.io/DeltaE/learn/

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

[19] J. Scott, M. A. Pusateri and D. Cornish, "Kalman Filter Based Video Background Estimation, ". Applied Imagery Pattern Recognition Workshop (AIPRW), 2009, pp.14-16

[20] 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), 2012