# A ROBUST TECHNIQUE FOR PERSON-BACKGROUND SEGMENTATION IN VIDEO SEQUENCES BASED ON THE CODEBOOK METHOD OF BACKGROUND SUBTRACTION AND HEAD TRACKING

*Ian Colman, Adel Rhuma, Miao Yu and Jonathon Chambers*

Advanced Signal Processing Group, Electronic and Electrical Engineering Department,
Loughborough University, Loughborough, Leicester, UK
{I.D.Colman-05,a.rhuma,elmy, j.a.chambers}@lboro.ac.uk

## ABSTRACT

In this paper, we introduce a new method to reliably extract humans from a video sequence even when the humans are static for long periods of time. The proposed method addresses a common problem in background subtraction techniques whereby humans that are static are mistaken for new additions to the background scene and are consequently absorbed into the background model. In the proposed method, codebook background subtraction is used to identify foreground regions in the video frame. A motion based particle filter is then used to track one or more human heads in the frame and determine which of these foreground regions represent people. The background model is then selectively updated given this knowledge thus ensuring that people will never be absorbed into the background model once detected, even when indefinitely static. Simulation results confirm that a human body is robustly extracted using this method in a non-static environment.

***Index Terms*—** Codebook, background subtraction, head tracking, motion-based particle filter, human body extraction

## 1. INTRODUCTION

Human-background segmentation in video sequences is an important and frequently used first step in many applications such as video surveillance, human computer interaction, human posture analysis and body tracking [1]. For human body extraction, background subtraction is commonly used. Multiple background subtraction methods exist, each with their own strengths and weaknesses for a given environment. The Mixture of Gaussians (MOG) algorithm is commonly used in highly dynamic multimodal background scenes [2]. Simpler methods such as the approximate median filter (AMF) offer low computationally complexity at the expense of ineffectively modelling the multiple modalities of the pixels such as those on the extremities of a tree blowing in the wind [2]. Kim et al. propose the codebook (CB) method that is computationally efficient yet copes well with shadows, static changes to the background scene and global illumination changes [3]. The CB method described in [3] is treated as a state of the art method for person extraction.

A common problem exists among all background subtraction methods with a dynamic background model and that is that static humans are mistaken for new additions to the background scene and are consequently absorbed into the background model.

We present a technique that addresses this issue by modifying how the codebook background model is updated. By introducing head identification and then tracking by way of a motion based particle filter, we can determine the foreground regions that represent one or more humans and consequently the detected foreground regions that identify new static changes in the true background scene. The background model can then be selectively updated given this knowledge thus ensuring that people will never be absorbed into the background model once detected even when indefinitely static, whilst new inanimate additions to the background scene are absorbed into the background model as desired.

In Section 2, the codebook background subtraction method is introduced. Head tracking is described in Section 3; and Section 4 contains details of experimental evaluations. Conclusions are drawn in Section 5.

## 2. CODEBOOK BACKGROUND SUBTRACTION

The codebook background subtraction algorithm is a quantization technique which constructs a background model from long observation of video sequences [3]. For each pixel, it builds a codebook consisting of one or more codewords. The codewords are determined according to a colour distortion metric together with a brightness bound. Initially, a codebook background model is trained on a pixel-by-pixel basis, and the training process is unconstrained allowing moving foreground objects in the scene during the initial training period.

Detection of whether a pixel in a frame is background involves testing the difference between the current image and the background model with respect to colour and brightness differences. If an incoming pixel meets two conditions–the colour distortion to some codeword is less than a threshold and the pixel's brightness lies within the brightness range of that codeword, it is classified as a background pixel.

In order to cope with the background change in the detection process, an additional model $\hbar$ called a cache [3] is used to update the trained background model, the later changes in the background will be absorbed into the model after a certain time so that associated pixels are not wrongly labeled as background.

Initially, a codebook model for the background is constructed, and we have to obtain the codewords for every single pixel from the training sequence. Each codeword $c_i$, i=1...L, of a single pixel, is composed of an RGB vector $v_i = (\bar{R}_i, \bar{G}_i, \bar{B}_i)$ and a 6-tuple $aux_i = \langle \check{I}_i, \hat{I}_i, f_i, \lambda_i, p_i, q_i \rangle$. The meaning of the elements in the 6-tuple $aux_i$ is:

$\check{I}, \hat{I}$ the min and max brightness of all pixels assigned to this codeword

f the frequency with which the codeword has occurred

$\lambda$ the maximum negative run-length (MNRL) defined as the longest interval during the training period that the codeword has not recurred

p,q the first and last access times, respectively, that the codeword has occurred

The procedure of constructing the codebook model for each pixel is shown as follow:

```
I. L←0, $\Im \leftarrow \phi$ (empty set)
II. for t=1 to N do
    (i )$x_t$=(R,G,B), $I \leftarrow \sqrt{R^2 + G^2 + B^2}$
    (ii)Find the codeword $c_m$ in $\Phi = c_i|1 \leq c_i \leq L$ (where $\Phi$ is the codeword set for a particular pixel) matching to $x_t$ based on:
    (a) colordist($x_t$,$v_m$)$\leq \varepsilon_t$
    (b) brightness(I,$\langle \hat{I}_m, \check{I}_m \rangle$)=true
    (iii) If $\Im = \phi$ or there is no match,then L←L+1. Create a new codeword $c_L$ by setting:
    •$v_L \leftarrow (R, G, B)$
    •$aux_L \leftarrow \langle I, I, 1, t-1, t, t \rangle$
    (iv) Otherwise, update the matched codeword $c_m$, consisting of $v_m = (\bar{R}_m, \bar{G}_m, \bar{B}_m)$ and $aux_m = \langle \check{I}_m, \hat{I}_m, f_m, \lambda_m, p_m, q_m \rangle$, by setting:
    •$v_m \leftarrow (\frac{f_m \bar{R}_m + R}{f_m + 1}, \frac{f_m \bar{G}_m + G}{f_m + 1}, \frac{f_m \bar{B}_m + B}{f_m + 1})$
    •$aux_m \leftarrow \langle min\{I, \check{I}_m\}, max\{I, \hat{I}_m\}, f_m + 1, max\{\lambda_m, t - q_m\}, p_m, t \rangle$
end for
III. For each codeword $c_i$, i=1,.......,L, wrap around $\lambda_i$ by setting $\lambda_i \leftarrow max\{\lambda_i, (N - q_i + p_i - 1)\}$.
```

The colordist($x_t$,$c_m$) represents the colour distortion (the details are shown in [3]) between $x_t$ and $c_m$ and the brightness(I,$\langle \hat{I}_m, \check{I}_m \rangle$) is true if I is in the range of $[\hat{I}_m, \check{I}_m]$. And finally, we delete the codewords whose $\lambda s$ are below a threshold to make a more compact codebook, normally, the threshold is set to N/2, where N is the number of training frames.

After obtaining the codebook model, the algorithm for background subtraction is carried out in the following way:

```
Step I. For each pixel P(x,y)=(R,G,B), calculate the intensity from the (R,G,B) value of a colour image by
I← $\sqrt{R^2 + G^2 + B^2}$
Step II. Find the first codeword $c_m$ from the code-book at the position (x,y) matching to P(x,y) based on two conditions:
1. colordist(P(x,y),$c_m$)$\leq \varepsilon_2$
2. brightness(I,$\langle \hat{I}_m, \check{I}_m \rangle$)=true
Update the matched codeword
Step III. If there is no match, then the pixel P(x,y) is categorized as foreground; otherwise, it is regarded as a background pixel.
```

Sometimes, the background will change after the training process and therefore the corresponding codebook model should be updated. The update procedure is shown in details in [3], by updating the background model, any changes in the background will be taken as the new background after certain iterations of the corresponding update steps.

Finally, we should remark that for an effective background subtraction, some post-processing steps, such as noise-removal, holes-filling and shadow-removal [4] are used to improve the quality of the background subtraction result.

## 3. HEAD TRACKING

In order to extract effectively the human body even when the body has remained static, the codewords of pixels should be updated selectively. The codewords of the pixels in the human body blob should not be updated and other pixels need to be updated according to the algorithm proposed in [3] so that this human body region will be always taken as foreground and extracted even when the person is static for a very long time while the background model is updated. The region which comprises the head is regarded as a human body region so that in order to identify the human body region, the head is first detected by some head detection algorithms such as [5] and then tracked in the video sequence. For head tracking, we use a more elegant motion based particle filtering method compared to the traditional generic particle filter.

Motion-based particle filtering is superior when compared to the traditional generic particle filter based on the condensation algorithm. The underlying formulation of the motion-based particle filtering algorithm we use is the same as that of the generic one. But its proposal distribution is different and is related to the output of an adaptive block matching (ABM) operation [6], which yields better results in cluttered environments.

### 3.1. Motion-based particle filter

The motion based particle filtering algorithm is based on the Bayesian sequential importance sampling scheme [7]. We assume in 2-D tracking, the head is modeled as an ellipse and its state vector is $\mathbf{S} = [x, y, l, \theta]^T$ [8], where (x,y) is the center of the ellipse representing a head, l is the length of the minor semi-axis (we assume the ratio between the major and minor axis is fixed, 1.2) and $\theta$ is the ellipse's orientation. And at time instant t-1, if we have N particles $\{\mathbf{S}_{t-1}^n\}_{n=1:N}$ and their corresponding weights $\{w_{t-1}^n\}_{n=1:N}$ then $\hat{\mathbf{S}}_{t-1} = \sum_{i=1}^{N} w_{t-1}^i \mathbf{S}_{t-1}^i$, where $\hat{\mathbf{S}}_{t-1}$ is the approximation of the minimum mean square estimation (MMSE) for $\mathbf{S}_{t-1}$ by the sampling method. We can then sample from an importance function $q(\mathbf{S}_t)$ to obtain N new samples for time t, and the corresponding weights are calculated as:

$$w_t^n = \frac{\sum_{j=1}^{N} w_{t-1}^j p(\mathbf{S}_t^n | \mathbf{S}_{t-1}^j)}{q(\mathbf{S}_t^n)} p(\mathbf{Z}_t | \mathbf{S}_t^n) \quad (1)$$

where $\mathbf{Z}_t$ is the measurement vector, $p(\mathbf{S}_t | \mathbf{S}_{t-1})$ is called the proposal distribution of $\mathbf{S}_t$ given $\mathbf{S}_{t-1}$ and $p(\mathbf{Z}_t | \mathbf{S}_t)$ is the distribution for the measurement given the state vector.

The MMSE at time t can be calculated according to the new $\{\mathbf{S}_t^n\}_{n=1:N}$ and $\{w_t^n\}_{n=1:N}$.

In a motion-based particle filtering algorithm, the proposal distribution $q(\mathbf{S}_t)$ has the following form:

$$q(\mathbf{S}_t) = \sum_{i=1}^{N} N_{\mathbf{S}_t}(\mathbf{S}_{t-1}^i + \Delta \mathbf{S}_t, \sum_G) \quad (2)$$

$N_{\mathbf{x}}(\mathbf{u}, \Sigma)$ is a multivariate Gaussian distribution.

Straightforwardly, we can obtain a sampling scheme as follow:

$$\mathbf{S}_t^n = \mathbf{S}_{t-1}^n + \Delta \mathbf{S}_t + \mathbf{v}_t^n \quad (3)$$

where $\mathbf{v}_t \sim N(0, \sum_G)$.

The parameter $\Delta \mathbf{S}_t$ is calculated as: $\Delta \mathbf{S}_t = [x_c(t) - x_c(t-1), y_c(t) - y_c(t-1), 0, 0]$, in which $[x_c(t), y_c(t)]$ is the center of the ellipse which best fits the ABM output at frame t. The details of how to calculate the ABM output are shown in [9].

In order to complete the algorithm description, we should also know $p(\mathbf{Z}_t | \mathbf{S}_t^i)$, in [10], it is shown that

$$p(\mathbf{Z}_t | \mathbf{S}_t^i) = p(\mathbf{Z}_{t,gradient} | \mathbf{S}_t^i) p(\mathbf{Z}_{t,color} | \mathbf{S}_t^i) \quad (4)$$

The calculations of $p(\mathbf{Z}_{t,gradient} | \mathbf{S}_t)$ and $p(\mathbf{Z}_{t,color} | \mathbf{S}_t^i)$ are found in details in [10].

This concludes the motion-based particle filtering algorithm. With the head tracking results, we can identify the particular blob which represents the human body and the corresponding codewords of the pixels in this blob will not be updated.

Figure 1 just shows the procedure of the proposed method in this paper:

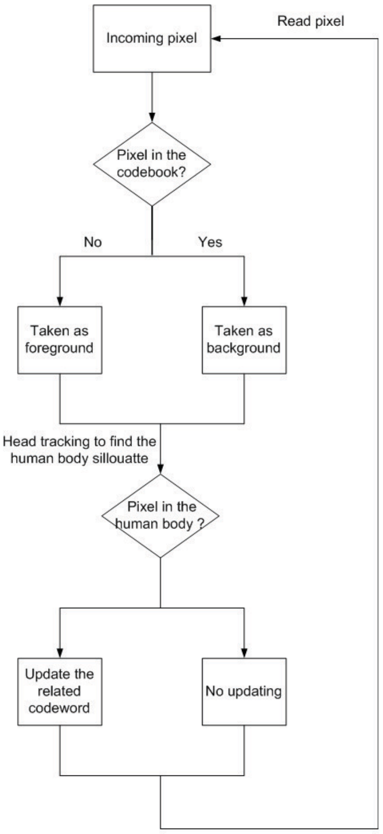

**Fig. 1**: The flow chart of the modified CB method

## 4. EXPERIMENTS AND EVALUATIONS

Figure 2 shows some results for the popular background subtraction methods (Mixture of Gaussian (MoG) [11], kernel [12], codebook (CB)). By comparison with the ground truth images shown in the bottom row, we can see that the codebook retains the most solid representation of the human body, with fewer background pixels being mistaken as foreground than the kernel method.

We evaluate the performance of a background subtraction method from two criteria [13]: true positive ratio (TPR), which is calculated by dividing the number of foreground pixels that are correctly detected by the actual number of foreground pixels; and false positive ratio (FPR), which is calculated by dividing the number of background pixels that are incorrectly detected as foreground by the actual background pixels' number. For a good background subtraction method, its TPR should be large and its FPR should be small. Figure 3 and 4 show the TPR and FPR of the three background subtraction methods for a recorded video, the CB method is most consistent in terms of yielding good results for both performance measures.

Some head tracking results using motion-based particle filtering method are given in Figure 5 to confirm that the head region is tracked consistently.

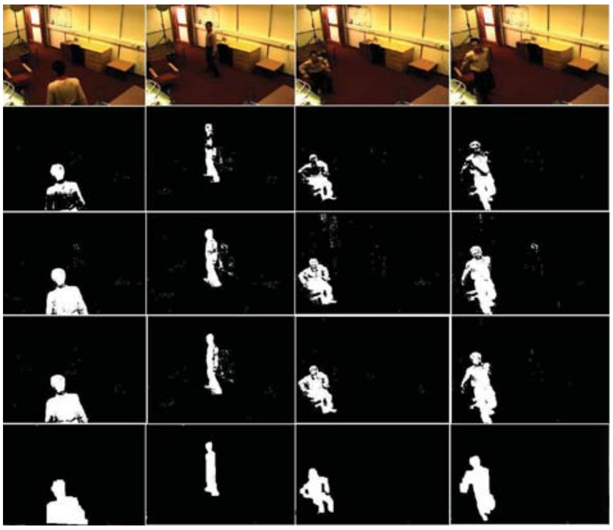

**Fig. 2**: Results of some popular background subtraction methods, first row: original image, the second, third, fourth and fifth rows are the results of MoG, kernel, CB methods and ground truth images respectively

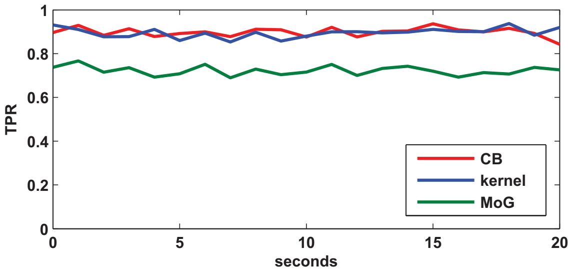

**Fig. 3**: The comparison of TPR for three methods

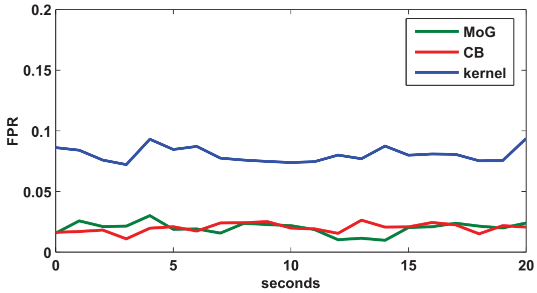

**Fig. 4**: The comparison of FPR for three methods

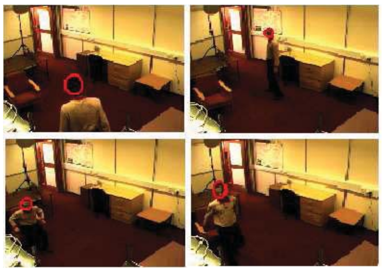

**Fig. 5**: Some head tracking results, the current tracked head position is highlighted by a red ellipse

Here we record a second video which shows two people' movement in a scene, a chair is moved by one person and after moving the chair, the two persons are static in the scene. Three codebook background subtraction versions: codebook background subtraction without updating (CB), codebook background subtraction with updating (ACB), codebook background subtraction with selective updating by using head tracking (SACB) are compared. Figure 6 shows some results of the three versions of codebook background subtraction method, the corresponding ground truth images are shown in the bottom line for comparison purpose.

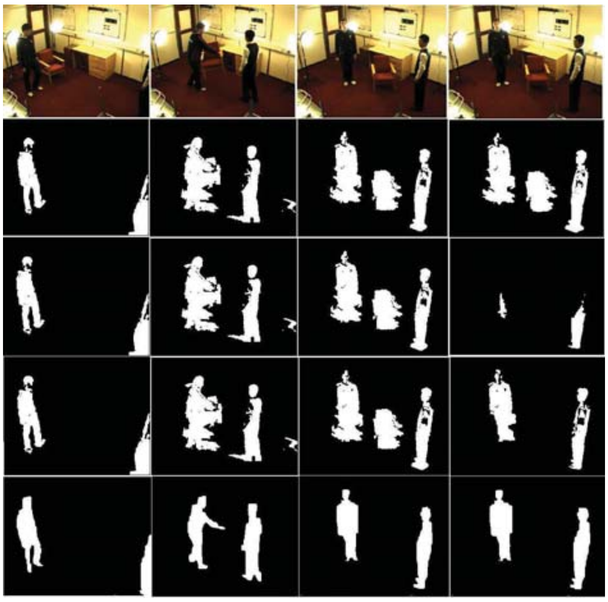

**Fig. 6**: The comparison of three codebook background subtraction methods, first row: original image, the second, third, fourth and fifth rows are the results of CB, ACB, SACB and ground truth images respectively

From the last column of Figure 6, we can see intuitively the SACB method proposed in this paper works best. Not only the chair is absorbed into the background, but also the two static persons are successfully segmented.

For an objective analysis, Figure 7 and figure 8 show the comparison of the TRP and FPR for the three versions of codebook background subtraction method for this video sequence.

From Figures 7 and 8, we can see that SACB works best when both high TPR and low FPR need to be considered.

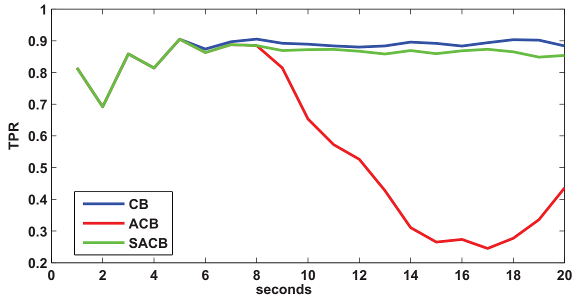

**Fig. 7**: The comparison of TPR for three codebook methods

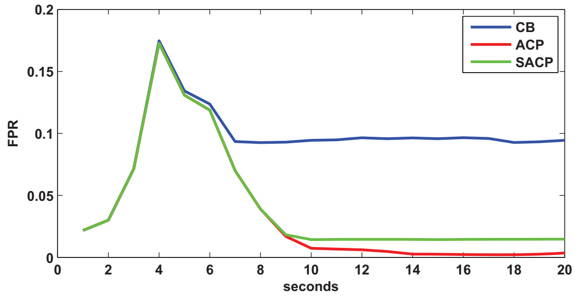

**Fig. 8**: The comparison of FPR for three codebook methods

## 5. CONCLUSION

In this paper, we have proposed a novel human body extraction method from video by combining the codebook background subtraction with head tracking. Codebook background subtraction is verified to be most effective among the popular background subtraction methods. And by the aid of head tracking, we can identify whether a blob is a human body blob by checking whether it contains the head or not, codewords in the background model corresponding to the human body will not be updated. In this way, the human body can be effectively segmented even if a person is static for a longtime, which overcomes a major drawback of current human body segment methods based on background subtraction.

## 6. REFERENCES

[1] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 780–785, 1997.

[2] S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," *in Proc. Video Communications and Image Processing, SPIE Electronic Imaging , San Jose, Calif, USA, January 2004*.

[3] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using code-book model," *Real-Time Imaging*, vol. 11, pp. 172–185, June 2005.

[4] R. Gonzalez and R. Woods, "Digital image processing, second edition," *Prentice Hall*.

[5] Y. Ishii, H. Hongo, K. Yamamoto, and Y. Niwa, "Face and head detection for a real-time surveillance system," *icpr, vol. 3, pp.298-301, 17th International Conference on Pattern Recognition (ICPR'04) - Volume 3, 2004*.

[6] N. Bouaynaya, Q. Wei, and D. Schonfeld, "An online motion-based particle filter for head tracking applications," *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005*.

[7] B. Ristic, S. Arulampalam, and N. Gordon, "Beyond the Kalman filter: particle filters for tracking applications," *In Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*, 2001.

[8] M. Yu, S. Naqvi, and J. Chambers, "Fall detection in the elderly by head tracking," *IEEE Workshop on Statistical Signal Processing, 2009*.

[9] K. Hariharakrishnan and D. Schonfeld, "Fast object tracking using adaptive block matching," *IEEE Transactions on Multimedia*, vol. 7, pp. 853–859, 2005.

[10] X. Xu and B. Li, "Head tracking using particle filter with intensity gradient and colour histogram," *In IEEE International Conference on Multimedia and Exploration*, pp. 888–891, 2005.

[11] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," *IEEE International Conference on Computer Vision and Pattern Recognition, 1999*.

[12] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," *European Conference on Computer Vision, 2000*.

[13] S. Zhang, H. Yao, and S. Liu, "Spatial-temporal non-parametric background subtraction in dynamic scenes," *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2009*.