Australian Journal of Basic and Applied Sciences, 5(5): 967-974, 2011
ISSN 1991-8178

# Object Tracking in Video Sequence Using Background Modeling

Mostafa Aghaee, Mehdi Amoon, Mohammad hamghalam and Ahmad Ayatollahi

Department of Electrical Engineering, Iran University of Science and Technology

**Abstract:** Modeling of the background is the first step and very important part of visual detection and tracking of objects. In this paper, we propose an efficient and adaptive background model for detecting foreground objects in the difficult scenarios of background scene. In the proposed method wavelet transform (WT) of the input frames is computed, then they are compared with saved wavelet transform samples of the background model. The proposed model can deal with complex background scenarios such as dynamic background scene, moved or new inserted background objects, illumination changes etc. Finally object tracking is done using proposed model. Singular value decomposition (SVD) method has been used for object tracking, which has the good performance in tracking of several moved objects. Experimental results show the efficiency of the proposed algorithm for detection and tracking of the objects in the image sequences.

**Key words**: background subtraction, object tracking, background model, wavelet transform

## INTRODUCTION

Any vision system which related to identification, interpretation and tracking of moving objects, firstly must be able to have good detection and segmentation of moving objects. Detection accuracy of moving blobs reduces the further processing in the later steps (such as tracking and identification objects), since only variant pixels should be tested.

The traditional method for motion detection is background subtraction, which is done by computing difference between current image and background model.

Environmental changes such as illumination changes, non static background, shadow, highlight and etc; make the simple background subtraction to be inefficient. Using background model which updates model in relation to these challenges, reduces false detection in background subtraction. However there is a tradeoff between accuracy in background modeling and processing time.

A good background model should have the following properties,

- Robust to illumination changes,
- Adaptive to changes in background subjects,
- Robust to periodic movement of background,
- Ability to model multi modal background,
- Limited computational complexity,
- Simplicity in implementation.

In the Sections 2 and 3 of this paper an efficient method for background modeling is proposed, which has been used for background subtraction. In the proposed method, wavelet transform (WT) of background samples are saved, then using these samples the statistical model of background is estimated. This model is quite simple but has a good efficiency in modeling of background. Our experimental results show the reliability and effectiveness of this model for various scenarios of background.

In Section 4 people tracking in video has been considered. People tracking is one of the most challenging task in machine vision. People movement is nonrigid, shape and size of people change in the video sequence, also people can group or occlude. An appropriate tracking algorithm must consider these challenges and have solutions for these challenges.

We use proposed model to segment people in the image sequences. Then tracking for each blob in the image is done by using extracted specification.

Finally in Section 5 Conclusions of this paper has been declared.

**Corresponding Author:** Mostafa Aghaee, Department of Electrical Engineering, Iran University of Science and Technology

## 2- Proposed Background Model:

### 2-1 Related Works:

The simplest way for modeling background consider that the value of pixel can be modeled by a single Gaussian distribution (Wren, 1997; Horprasert, 1999). However this simple method cannot model multi modal background pixels.

The well known mixture of Gaussian model (MOG) (Stauffer, 1999) has been used to model more complex background. MOG uses several Gaussian functions and mixes them to model the distribution of pixels. MOG can adapt to the variations of background scene. However, this method has also some problems. For example initialization of its parameters is slow and sometimes inaccurate, background scenes with fast variations are not easily modeled and the computations of model are quite heavy.

Researchers have developed MOG. Several modified MOG models are proposed (Elgammal, 2000; Javed, 2002). Also, MOG has been used in many algorithms such as Bayesian frameworks (Lee, 2003), dense depth data (Harville, 2002), mean shift analysis (Porikli, 2003) and region based information (Cristani, 2002).

Another method that has been proposed to deal with multi modal background pixels is Wallflower (Toyama, 1999). This method uses a linear Wiener filter to learn and predict changes of background. However, this model is less effective when the background scene changes fast.

Codebook model (Kim, 2005) is a method for real time foreground-background segmentation. Sample background values are quantized into codebooks which represent a compressed form of background model for a long image sequence. This method is able to model multi modal background pixels and also is applicable to compressed video such as MPEG.

Sample consensus (SACON) model (Wang, 2007) compute sample consensus of the background samples and estimates a statistical model of the background, per pixel. SACON exploits both color and motion information to detect foreground objects. However, the required memory for saving background samples in SACON model is very much, especially when a lot of frames are needed to be saved.

### 2-2 Construction of the Proposed Model:

This method is inspired from SACON method, which is based on comparing between background samples and new samples.

In the proposed method for background modeling, we use WT. WT is a tool for showing signal characteristics. Using both frequency and time-spatial information is the most important characteristic of this transform, which differs it from Fourier transform.

After applying this transform to an image, four sub images are produced. For this transform first a filtering and down sampling stage in one orientation (for example row) is done, then similar stage apply to another orientation (here column). Remained four sub images are called LL, LH, HL and HH which H points to high pass filtering and L points to low pass filtering. LL also called the approximation coefficient is used in our method and we do not need another sub images.

In our method several frames of background are saved and used for building background model. First we apply third level 2D WT to these samples. This work filters the high frequency noise in the image and decreases the resolution of samples; hence less memory space is needed.

By applying WT to an image, the image is decomposed to 3K+1 sub images, see (1).

$$\left\{LL_J, \left[LH_J, HL_J, HH_J\right]_{J=1,2,\ldots,k}\right\} \tag{1}$$

Which k is the level of decomposition. $LL_J$ is the $J$-th low frequency sub image, represents low frequency component in vertical and horizontal direction. $LH_J$, $HL_J$ and $HH_J$ are another $J$-th sub images which have high frequency filtering in one or two direction. The resolution of the $J$-th sub image in WT is $1/2^{2J}$ resolution of original image. In the proposed method, we adopt three level low frequency sub image, $LL_3$, which is similar to original image and also with decreased noise and resolution.

We saved $N$ number $LL_3$ of sample background for any pixels. So for pixel $m$ at time $t$, which $t>N$, we have:

$$\left\{LL_{[3]_i}(m) \mid i = 1, \ldots, N \quad N < t\right\} \tag{2}$$

As we described before, $LL_{[3]t}(m)$ is the approximation coefficient of WT at third level for pixel $m$, at time $t$. Due to using color images and RGB color space, $LL_{[3]t}(m)$ has three components as,

$$LL_{[3]_i}(m) = \left(LL_{[3]_i}{}^{R}(m), LL_{[3]_i}{}^{G}(m), LL_{[3]_i}{}^{B}(m)\right) \tag{3}$$

For any component of saved samples, we define a binary label which shows amount of similarity between new frame and saved samples. This is done by following function,

$$LL_i{}^{c}(m,t) = \begin{cases} 1 & \left|LL^{c}_{[3]_i}(m) - LL^{c}_{[3]_t}(m)\right| \le T_r \\ 0 & otherwise \end{cases}, \tag{4}$$

In the equation 4, c shows color component R, G and B. $T_r$ is the noise threshold, $LL^{c}_{[3]t}(m)$ and $LL^{c}_{[3]t}(m)$ are C component third level approximation coefficient of pixel m at time $i$ and $t$, respectively. $LL^{c}_{i}(m,t)$ is binary label of C component for $i$-th sample of pixel $m$. Determining of The value of $T_r$ affects the quality of foreground-background detection, hence its value is very important and discuss in section 2-4.

In the next step, we count number of agreement between new frame and saved sample. According to this sum we decide new pixel m is background or not. The following equation shows decision criteria,

$$M_t(m) = \begin{cases} 1 & \sum_{i=1}^{N} L_i^c(m,t) \ \ge \ T_n \ \ \forall \in \{R,G,B\} \\ 0 & otherwise \end{cases} \tag{5}$$

In which $T_n$ is the threshold value and its value is the minimum amount of agreement between new pixel and saved samples to be a background pixel. $M_t(m)$ is the binary label for pixel $m$ at time $t$, when it's value be 1 shows the new pixel is background otherwise this pixel belongs to foreground.

It is clear that $T_n$ is affected by number of saved pixels at time t, $N_t(m)$. The larger $N_t(m)$ should increase $T_n$.

Likewise, $T_r$ influences $T_n$. For example if the amount of $T_r$ increases, then more new pixels have the chance to meet the agreement criteria (see (4)), so we should increase to compensate that increase.

A good decision is to define $T_n$ with respect to $N_t(m)$ and $T_r$ simultaneously, as

$$T_n = \alpha T_r N_t(m), \tag{6}$$

In which is an empirical constant. It is important to note that at the beginning, we set the amount of $N_t(m)$ to fixed and predefined value but because of some reasons this value maybe change and be less than expected value. We describe this subject with more detailed in the section 2-5.

***2-3 Shadow and Illumination Change Removal:***

RGB color space is very sensitive to the noise and the illumination changes. For example in RGB color space, shadows seem to be foreground objects and cause some mistake in object detection. For solving this problem, researchers used normalized RGB color space which is more reliable against noise and illumination changes. Normalized components are achieved with following equation:

$$r = {}^{R}/_{(R+G+B)} \tag{7-a}$$

$$g = {}^{G}/_{(R+G+B)} \tag{7-b}$$

$$b = {}^{B}/_{(R+G+B)} \tag{7-c}$$

However, using RGB normalized color space skips illumination information of color image. Miss of illumination information reduces performance of object detection, so we use (*r, g, I*) components instead of (*r, g, b*) and we apply WT to (*r, g, I*) components.

Assume that ($r_b$, $g_b$, $I_b$) is the color components vector of *b*-th saved sample for a pixel and ($r_t$, $g_t$, $I_t$) are the color components of the pixel at time *t*.

When pixel at time *t* is in the shadow, we expect to be darker or $\beta \leq I_t / I_b \leq 1$. Also when pixel is highlighted we expect $1 \leq I_t / I_b \leq r$. Therefore, we can show total illumination change of pixel with $\beta \leq I_t / I_b \leq r$ . In respect with these modifications, we modify equation 4 to the following equation,

$$L_i^c(m,t) = \begin{cases} 1 & \left|LL^c_{[3]i}(m) - LL^c_{[3]t}(m)\right| \leq T_r \quad \forall c \in \{R,G\} \\ & \text{and} \quad \beta \leq I_t/I_b \leq \gamma \qquad c \in \{I\} \\ 0 & otherwise \end{cases} \tag{8}$$

In which β and γ are empirical coefficient and here we set them β = 0.6 and = 1.5.

***2-4 Choosing the value of $T_r$:***

There are two way for choosing the value of $T_r$. The first one is to define an empirical constant value for all pixels. But in this way, it is difficult to obtain a fit value of $T_r$ for all pixels. In the second way value of $T_r$ is not constant and is defined with respect to standard deviation of each pixel, $\sigma_r$. For example a good choice is $\eta\sigma_r$. However, when distribution of pixel is multi modal, the value of $\sigma_r$ become greater, so we should limit the value of $T_r$ to a maximum value. Finally we set the value of $T_r$ for pixel *i*, $T_{ri}$, to

$$T_{ri} = \min(T_1, \eta\sigma_i) \tag{9}$$

$T_r$ is the constant value which is the maximum value of $T_{ri}$.

***2-5 Updating Background Samples:***

When the background changes, background saved sample should be updated to handle the changing of background. Illumination change, moved or inserted background object and static foreground object for long time are some examples which cause background scene change.

There are several methods for updating background (Elgammal, 2000; Karmann, 1990). The easiest way is to add each new value of pixel to the background model. However, in this method foreground values are added to background model, too. Another simple and more precise method is to add only background values to the background model. This method is useful but also has some problems. For example, when the background object moves or new background object is inserted to scene, they are considered to be the foreground objects.

For solving these problems and constructing an efficient model, we use a selective updating method. This method considers moved or new inserted background object, as a background object after the specific time.

In the proposed method, we define a counter for each pixel. This counter counts the number of consecutive frames where the pixel is classified as a foreground. The following equation shows the counter for pixel *m* at time *t*,

$$CNT_t(m) = \begin{cases} CNT_{t-1}(m) + 1 & if\ M_t(m) = 0 \\ 0 & otherwise \end{cases} \tag{10}$$

This counter is reset, when the pixel is classified as a background. When the value of this counter reaches to a predefined value, this pixel is classified as a background. In the other hand, after the specific time, if the pixel still is classified as a foreground, assume that this pixel belongs to a moved or new inserted background object. Now, it is necessary to rearrange the background model for this pixel. Since the previous saved background samples of this pixel are not valid any more, we delete them from the model and rebuild the model. The current value of pixel is the first background sample of new model and number of the saved samples for this pixel at this time set to one, $N_t(m) = 1$. We add background samples of later frames to the model and increase the value of $N_t(m)$, until the number of saved background samples reach to *N*. The following equation shows the number of saved samples at time *t* for pixel *m*,

$$N_t(m) = \begin{cases} N_{t-1}(m) + 1 & \text{Pixel m is background and } N_{t-1}(m) < N \\ N_{t-1}(m) & \text{otherwise} \end{cases} \quad (11)$$

***3- Exprimentall Results of Background Model:***

In this section experimental result has been presented for evaluating the proposed method. For testing our background method we apply it to wallflower image sequences. Toyama *et al*. (1999) use this image sequences in their algorithm. In each sequence, there is some difficulty for modeling background scene. The size of each frame is 160×120 pixels. Frame rate is 4 Hz. There is a hand segmented image for each sequence that we can evaluate our results. We test our method with these image sequences.

The Wallflower image sequences consist of seven sequences. We describe each sequence briefly in the following,

- Moved Object, MO: A man enters a room. Makes a phone call and then leave the room.
- Time of Day, TOD: The light of room change gradually. A person comes to the room and sit.
- Light Switch, LS: There is a room with light on. A person enters the room and turn off the light for a long time. Then he walks into the room, turns on the light and moves the chair.
- Waving Trees, WT: A tree is waving because of wind and a person stands in front of the tree for a while then leaves.
- Camouflage, C: A person stands in front of a monitor while the screen of monitor has noisy bars.
- Bootstrapping, B: This sequence shows busy cafeteria.
- Foreground Aperture, FA: A person with similar colored cloths stands up and walks.

In this section we perform background subtraction using our background model and compare our results with Wallflower and MOG in.

For evaluating performance of method, three parameters are defined: false positive (FP), false negative (FN) and total error (TE). FP is the number of background pixels which are wrongly marked as a foreground, FN is the number of foreground pixels which are wrongly marked as a background and TE is the summation of FP and FN.

Figure 1 shows one frame of each seven sequences of Wallflower image sequences which is used for background subtraction. The first row shows the name of each sequence. There is the number of frames in second row. The third row shows the frame under test. The fourth row is the hand segmented ground truth images. Finally, the fifth, sixth and seventh rows show our method, Wallflower and MOG models, respectively. Quantity results for each background model are showed in table 1.

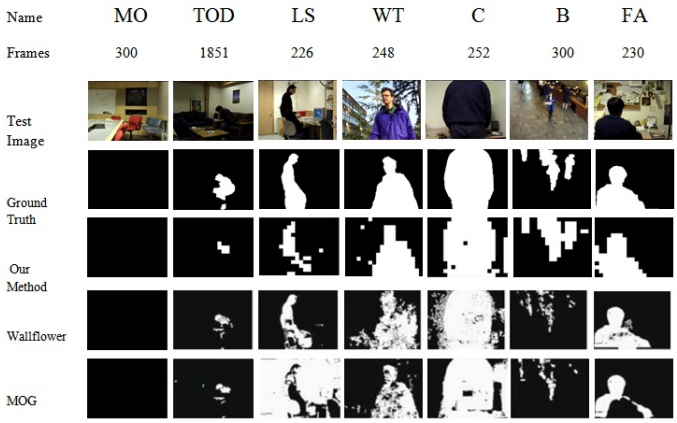


**Fig. 1:** Experimental results for three tested methods using Wallflower image sequences.

***4- Object Tracking in Video Sequence:***

Moving objects in the frame are obtained by background subtraction operation. Then each pixel in the frame is labeled as a background or foreground. After this operation we have a binary frame which should be cleaned. In order to cleaning and segmentation of image, we used morphological operation.

**Table 1:** Quantity results of simulations for the tested methods.

| model | fault | MO | TOD | LS | WT | C | B | FA |
|---|---|---|---|---|---|---|---|---|
| Proposed | FP | 0 | 8 | 528 | 680 | 329 | 917 | 768 |
| model | FN | 0 | 1172 | 1286 | 97 | 975 | 1090 | 286 |
| | TE | 0 | 1180 | 1814 | 777 | 1304 | 2007 | 1054 |
| | FP | 0 | 25 | 375 | 1999 | 2706 | 365 | 649 |
| wallflower | FN | 0 | 961 | 947 | 877 | 229 | 2025 | 320 |
| | TE | 0 | 986 | 1322 | 2876 | 2935 | 2390 | 969 |
| | FP | 0 | 20 | 14196 | 341 | 3098 | 217 | 530 |
| MOG | FN | 0 | 1008 | 1633 | 1323 | 398 | 1874 | 2442 |
| | TE | 0 | 1028 | 15802 | 1664 | 3496 | 2091 | 2972 |

Firstly, for connecting the blobs together, we define a neighborhood region for each pixel. We set the value of this pixel 1when more than half of its neighborhood pixels are 1's. Then we perform opening and closing operations. The resulting image is the binary image which moving objects are detected. Since, sometimes there is more than one moved object in the image, we label each separated blob. So we can track each blob separately.

Now for object tracking, we define a bounding box for each blob. The bounding box is the minimum rectangle around the blob which covers the blob totally.

For tracking of the blobs in consecutive frames we need some features of the blobs. Hence, for each blob a feature vector is defined which includes position of the bounding box centroid, dimension of the bounding box, average of the color components, standard deviation of the color components, area of the blob and orientation of the blob. Now, we should measure the amount of similarity between two feature vectors.

The similarity of two blobs, $I_i$ and $J_j$, can be measured by the Mahalanobis distance with the following equation

$$\boldsymbol{d_{ij}} = (b_i - b_j)^{\mathbf{T}}(\Lambda_I - \Lambda_J)^{-1}(b_i - b_j), \tag{12}$$

$A_I$ and $A_J$ are the covariance matrixes of the feature vectors in image $I$ and $J$, respectively. For matching the blobs we use a method described in (Scott, 1991) and (Pilu, 1997). This method uses singular value decomposition, SVD, to associate feature of two images. Assume $\{I_i\}_{1..n}$ and $\{J_j\}_{1..n}$ are two sets of blobs in image $I$ and $J$ which we want to associate them to each other. Firstly, we build a matrix named proximity matrix, P, using Mahalanobis distance. The element of this matrix is defined: $p_{ij} = e^{-d_{ij}/\lambda}$. The next step is performing the SVD operation on P, $P = USV^T$. U and V are the orthogonal matrices and S is the non negative $m \times n$ diagonal matrix. Eventually, S is converted to a new $m \times n$ matrix D which $S_{ii}$ elements of S are replaced with 1. So a new matrix, $Q = UDV^T$, obtain. The $Q_{ij}$ element of this new matrix shows the amount of correspondence between two blobs $I_i$ and $J_j$. If $Q_{ij}$ is the maximum element in its row and column and their Mahalanobis distance is below the predefined threshold, then $I_i$ and $J_i$ are corresponding blobs in image I and J. We use this technique for tracking multi objects in the image sequences. Experimental results show the effectiveness of this method for multi object tracking.

Figure 2 and 3 show the experimental results of one and two people tracking.

***5- Conclusion:***

In this paper a new algorithm of object tracking is proposed using background model. In the proposed we use wavelet transform for modeling background. Wavelet transform reduces the resolution of image and filters the noise of image. Using wavelet transform, reduces the computational complexity and false detection caused by noise. Less required memory needed for saving background samples is another benefit of this method.

Finally, we use singular value decomposition (SVD) method for multi objects tracking. In this method feature vector of moved blobs is defined and used in proximity matrix. Then new matrix is constructed using SVD technique which indicates associated moved blobs in the consecutive frames. Experimental results show effectiveness and efficiency of this method.

**Fig. 2:** One object tracking in an image sequence.

**Fig. 3:** Two object tracking in an image sequence.

## REFERENCES

Cristani, M., M. Bicego, V. Murino, 2002. Integrated region- and pixelbased approach to background modelling. Proceedings of IEEE Workshop on Motion and Video Computing.

Elgammal, A., D. Harwood, L.S. Davis, 2000. Non-parametric model for background subtraction, Sixth European Conference on Computer Vision, pp: 751-767.

Harville, M., 2002. A framework for high-level feedback to adaptive, perpixel, mixture-of-gaussian background models. European Conferenceon Computer Vision, 3: 543-60.

Horprasert, T., D. Harwood, L.S. Davis, 1999. A statistical approach for real-time robust background subtraction and shadow detection. IEEE Frame-Rate Applications Workshop, Kerkyra, Greece.

Javed, O., K. Shafique, M. Shah, 2002. A hierarchical approach to robust background subtraction using color and gradient information, IEEE Workshop Motion Video Comput., 22-27.

Karmann, K.P., A.V. Brandt, 1990. Moving object recognition using and adaptive background memory,Time-Varying Image Processing and Moving Object Recognition, Elsevier, Amsterdam.

Kim, K., *et al*., 2005. Real-time foreground–background segmentation using codebook model, Real-Time Imaging, 11(3): 172-185.

Lee, D.S., J.J. Hull, B. Erol, 2003. A Bayesian framework for Gaussian mixture background modeling. IEEE International Conference on Image Processing.

Pilu, M., 1997. A direct method for stereo correspondence based on singular value decomposition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp: 261-266.

Porikli, F., O. Tuzel, 2003. Human bodytracking by adaptive background models and mean-shift analysis. IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS-ICVS).

Stauffer, C., W.E.L. Grimson, 1999. Adaptive background mixture models for real-time tracking. IEEE International Conference on Computer Vision and Pattern Recognition, 2: 246-52.

Scott, G., H. Longuet-Higgins, 1991. An algorithm for associating the features of two images, Proceedings of the Royal Society of London B, 244: 21-26.

Toyama, K., *et al*., 1999. Wallflower: principles and practice of background maintenance, Seventh International Conference on Computer Vision, pp: 255-261.

Wang, H., D. Suter, 2007. Aconsensus-based method for tracking: Modelling background scenario and foreground appearance, Pattern Recognition, 40: 1091-1105.

Wren, C.R., A. Azarbayejani, T. Darrell, A. Pentland, 1997. Pfinder*: realtime tracking of the human body*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7): 780-5.