Journal of Computer Science 8 (7): 1062-1069, 2012
ISSN 1549-3636
© 2012 Science Publications

An Effective History-based Background Extraction System

Seyed Hashem Davarpanah,
Fatimah Khalid and Maryam Golchin
Department of Multimedia,
Faculty of Computer Science and IT,
Universiti Putra Malaysia, Serdang, 43300, Malaysia

Abstract: Problem statement: In many visions-based surveillance systems, the first step is accomplished by detecting moving objects resulted from subtraction of the current captured frame from the extracted background. So, the results of these systems mainly depend on the accuracy of the background image. **Approach:** In this study, a proposed background extraction system is presented to model the background using a simple method, to initialize the model, to extract the moving objects and to construct the final background. Our model saves the history of each pixel separately. It uses the saved information to extract the background using a probability-based method. It updates the history of the pixel consequently and according to the value of that pixel in the current captured image. **Results:** Results of the experiments certify that not only the quality of the final extracted background is the best between four recently re-implemented methods, but also the time consumption of the extraction is acceptable. **Conclusion:** Since History-based methods use temporal information extracted from the several previous frames, they are less sensitive to noise and sudden changes for extracting the background image.

Key words: Adaptive background extraction, background modelling, probability-based method, moving object detection, foreground detection, history-based method, outdoor applications

INTRODUCTION

Many visions-based surveillance systems such as Intelligent Transportation Systems are started by detecting changes in an image sequence. To identify changes, the first group of methods use algorithms which are designed based on inter-frame differences. These methods always detect larger moving object areas than the real. Thus they are not able to detect immobile objects correctly. In addition, they are liable for noise impact and they have the low detection precision and low reliability (Xiaofei, 2009). In the second group of background extraction methods, some researchers use algorithms which are based on the subtraction of two images; the input frame and a reference image. The latter image presents the background image of the scene. The efficiency of this group mainly depends on the accuracy of the extracted background image. In addition, this approach is affected by the adaptability of the background image extracted in various situations. This problem prompted researchers to find a more performance-based approach to model, initialize, adapt and subtract a more accurate background especially in real-time and outdoor environments.

The background objects are a combination of stationary objects like traffic lights and non-stationary objects such as waving bushes. On the other hand, pixels of the background image are a combination of static pixels and dynamic pixels. The static pixels present the stationary objects and the dynamic pixels belong to the non-stationary objects (Li *et al*., 2004). A proper background extraction algorithm should be able to handle both static and dynamic pixels.

The gray scale of a stationary pixel of each frame has a constant value; this constant is the intensity of that pixel. Gray values of the pixels in a motionless background image are not always the same. It is because of disturbances in lighting and intensity, cameras, atmosphere and the existence of moving objects for a short or a long period. So, the constant value allocated to each pixel may change in one or a few time slices. However, it seems that in a certain duration time, the value of each pixel in the background image of a video sequence is likely to be equal to the pixel with highest appearance frequency in corresponding scenes. History-based methods are based on this fact. They save the history of each pixel for a specific period of time and in the form of values allocated to one or a range of different gray scale values.

Corresponding Author: Seyed Hashem Davarpanah, Department of Multimedia, Faculty of Computer Science and IT, Universiti Putra Malaysia, Serdang, Malaysia Tel: (+60-3)89471730 Fax: (+60-3)89466577

In the recent years several studies have been undertaken based on modelling the history of each pixel or a group of them (block or region) as the background model (Xiaofei, 2009; Kong *et al*., 2007; Chiu *et al*., 2010; Cui *et al*., 2009). In these methods, some information of each unit (pixel, block or region) should be saved separately. Finding the Information which is able to model the background more accurate and designing an updating which updates the saved information in the minimum time and with the highest performance is the object for the researchers in all these years.

In this study, we introduce a complete history-based background extraction system. This system is a combination of a simple method to model the background, a new algorithm to initialize the background, a fast algorithm to update the system and adapt the background, a fast technique to extract the binary objects, an enhanced median filter to reduce the existing noise and an effective algorithm to construct the final background.

The proposed method can be categorized as probability-based method. It saves the history of each pixel separately. The method uses the history in order to estimate the current value of the corresponding pixel in the background image. Once that current frame is captured by system, the history is updated.

The rest of this study is organized as follows: The first section reviews a number of more significant related works. The second section gives an overview of the proposed system. The following section provides the experimental results and the study is concluded in the last section.

Related works: Several studies have been undertaken over the last three decades to model, extract and adapt the background image. Speed, adaptability to periodic changes such as changes in sunlight, management of fast changes caused by moving objects entering the scene or leaving the scene are the most critical problems for these studies. Previous methods have used nonparametric and parametric methods to model the changing behavior of the scene, moving objects, background or some sections within them. In the following, some of the significant samples of these studies and their properties are highlighted.

As a simple nonparametric model, in order to estimate a new background, researchers construct a combination of the current background and one, two or more previous extracted backgrounds. The technique of combining the frames; the number of combined frames, the composition of these frames and the combination rates have been investigated several times.

The common goal of these previous studies is improving the adaptability of the final extracted background. Namely, Fuzzy logic and Cellular Automata are used to decide how to combine frames together more efficiency (Shakeri *et al*., 2008). Xiaofei (2009) introduced a background extraction system with an update algorithm which is self-adaptive. He also presented an extended method based on histogram statistics combined with multi-frame averages and claimed that this approach is faster and suitable for platforms based on a Digital Signal Processor (DSP). Also, Barnich and Droogenbroeck (2009) presented the ViBe program to combine frames randomly.

Stauffer and Grimson (1999) supposed that a specific pixel may have values following a Gaussian distribution. In the Gaussian distribution model, the background of a scene is assumed as a set of pixels with a Gaussian distribution of between 2 and 2.5 standard deviations for each pixel (Peng and Horng, 2009). Subsequently, a Mixture of Gaussians (MoG) was introduced. According to MoG, an image is modelled in the form of a combination of k separate Gaussian models (usually k is between 3 and 5). Researchers set parameters for each Gaussian (mean, variance, weight and learning rate). They have presented various combinations of a different number of Gaussian models to adapt to the gradual background changes (Sheng and Cui, 2008).

Also, a Local Binary Pattern (LBP) is widely used as a texture descriptor. It has acceptable performance in fabric defect detection, texture classification, face recognition and moving region detection. In this method, local features which are considered as neighboring pixels are assigned to the texture features (Armanfard *et al*., 2009a). LBP has been used by Heikkila and Pietaikinen (2006) to model the history of each pixel in any block. Armanfard *et al*. (2009b) tried to reduce the size of the model.

In our own previous work, a combination of blocking and multi-scale methods was used. Block-based techniques are appropriate to control the movements of non-stationary objects because they are less sensitive to local movements, especially in outdoor applications. They can be useful to reduce the effect of these objects on the extracted background. We also used a blocking method to select the regions in which temporal filtering has to be applied. This method is appropriate to extract moving objects when motion is not predictable. However its results are not stable. They are sensitive to any sudden motion.

MATERIALS AND METHODS

Proposed model: The proposed method is a combination of the following steps:

- A probability-based method to model the background
- An effective algorithm to initiate the background model
- An accurate method to extract moving objects based on the adaptive model resulted from the previous phases
- A post processing technique to reduce the noises in the binary image showing moving objects
- A simple method to extract the final background

To continue, each of these methods is described in detail below. Also, various parts of the proposed model are illustrated in Fig. 1.

Before making a decision concerning the model of the system, it is necessary to know how to transform the visual data for input into the system. Gray scales are usually used by different researchers and it is a key factor to judge the extracted background and moving-objects. Furthermore, to reduce the complexity of the algorithm, we process a color image and convert it to gray scales using a famous Equation, Eq. 1:

$$f(x,y)= 0.299 \times f(x,y)_R + 0.587 \times f(x,y)_G + 0.114 \times f(x,y)_B \quad (1)$$

Where, $f(x, y)_R$, $f(x, y)_G$ and $f(x, y)_B$ refer to the RGB components of a pixel (x,y) of the current frame, respectively. Using a histogram method, we save the history of each pixel separately. To do so, the range of the grayscale values is divided into W slices. The size of each range is Clu_size. Also, the range of the gray scale in each slice is equal to Eq. 2:

$$[I \times \text{Clu\_size} \ldots (i+1) \times \text{Clu\_size-1}] \quad i \in [0..W-1] \quad (2)$$

In the proposed model, each slice is explained by an ordered pair (μ_i, C_i). μ_i is the mean value belongs to the ith slice and C_i is the number of frames which their corresponding pixel has a gray scale located in the i^{th} slice. Whenever a new frame has a pixel in a specific slice, these two parameters of the corresponding slice are updated. The updating process will be explained in the background adoption section. In the proposed method, the number of slices, W, is 64. First, μ_i value of the corresponding slice of each pixel is initialized with the value of that pixel in the first frame. Its C_i is assigned as 1. This value for the other slices is set zero.

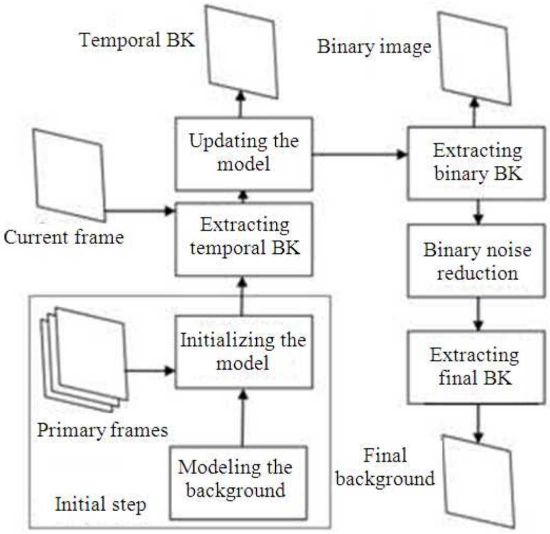


Fig. 1: Proposed flowchart to extract the background

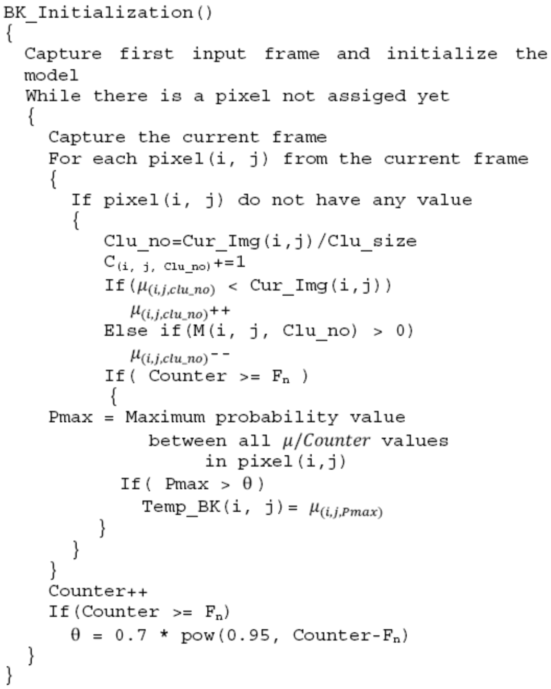
In this study, to initialize the background, a new algorithm is presented. The most similar algorithm to our algorithm was introduced by Chiu *et al*. (2010).

The proposed initialization algorithm uses a distance classifier to classify the input image for each pixel until Fn^{th} frame. Then, the probability of the grayscale of each pixel is calculated and a convergent value is used to extract the gray scale of the background pixel whose probability is maximum and greater than the convergent value. In the following steps, the background initialization algorithm is explained. Also, the pseudo code of the method is also illustrated in Fig. 2:

Step1: Capture the first input frame and initialize the model as explained before.
Step2: While there is any pixel in the background (Temp_BG in the pseudo code) not yet having a value yet, repeat steps 3-12.
Step3: Capture the next input frame. (Counter is the number of grabbing frames).
Step4: If there is any unassigned pixel in the captured frame, do steps 5-7.
Step5: Calculate the corresponding slice of the grayscale for the fetched pixel according to Eq. 3:

$$\text{Clu}_{\text{no}(i,j)} = \left[\frac{\text{Cur}_{\text{img}(i,j)}}{\text{Clu}_{\text{size}}}\right] + 1 \quad (3)$$

where, i and j are the coordinators of the corresponding pixel, Cur_Img refers to the most recently captured image and Clu_size is the size of each slice of grayscale values.



```
BK_Initialization()
{
  Capture first input frame and initialize the
  model
  While there is a pixel not assiged yet
  {
    Capture the current frame
    For each pixel(i, j) from the current frame
    {
      If pixel(i, j) do not have any value
      {
         Clu_no=Cur_Img(i,j)/Clu_size
         $C_{(i, j, Clu\_no)}$+=1
         If($\mu_{(i,j,clu\_no)}$ < Cur_Img(i,j))
           $\mu_{(i,j,clu\_no)}$++
         Else if(M(i, j, Clu_no) > 0)
           $\mu_{(i,j,clu\_no)}$--
         If( Counter >= $F_n$ )
          {
    Pmax = Maximum probability value
             between all $\mu/Counter$ values
                  in pixel(i,j)
           If( Pmax > $\theta$ )
             Temp_BK(i, j)= $\mu_{(i,j,Pmax)}$
         }
      }
    }
    Counter++
    If(Counter >= $F_n$)
      $\theta$ = 0.7 * pow(0.95, Counter-$F_n$)
  }
}
```

Fig. 2: Pseudo code of the initialization algorithm

Step 6: Set the input pixel as a member of the Clu_No group. It is necessary to update the parameters of this group as Eq. 4 and 5:

$$\begin{aligned}&C(i, j, Clu_{no}) = C(i, j, Clu_{no}) + 1\\&\text{If } (m(i, j, Clu_{no}) < Cur_{img(i, j)}\end{aligned} \quad (4)$$

$$\begin{aligned}&\mu(i, j, Clu_{no}) = C(i, j, Clu_{no}) + 1\\&\text{Else if } (m(i, j, Clu_{no}) > 0)\\&\mu(i, j, Clu_{no}) = C(i, j, Clu_{no}) - 1\end{aligned} \quad (5)$$

Step7: If the Number of frames (Counter) is less than Fn go to the next step (Step8) else calculate probability value of the maximum cluster, Pmax(i,j), for the pixel using Eq. 6 and 7:

$$P(i, j, m) = \frac{C(i, j, m)}{Counter} \forall m \in [0..W-1] \quad (6)$$

$$Pm\,ax(i, j) = \max P(i, j, m) \forall m \in [0..W-1] \quad (7)$$

Now, if (Pmax(i,j)>θ), pixel(i, j) of the background image (Temp_BG) is assigned with the mean value (μ) of the gray scale slice which has the highest probability value (pmax (i, j) for this pixel (Eq. 8). θ is the convergent value:

$$\begin{aligned}&TempBG(i, j) = m(I, j, m)\\&\text{Where } P(i, j, m) = Pmax(i, j)\end{aligned} \quad (8)$$

Step8: To increase the speed of the extraction process of background pixels, the converted value, E, is updated as defined by (Eq. 9). This process is done whenever the counter is equal or more than Fn:

$$\theta = \eta \times \omega(Counter - F_n), counter > F_n \quad (9)$$

Where:

η = The initial convergent value of f

ω = The weighting value of the initial convergent

Here, as in Chiu *et al*. (2010), we set η and ω to 0.7 and 0.95, respectively.

After initializing of the system, by capturing a new frame, its corresponding background should be extracted. First, to extract the adaptive background to the current frame, the moving objects should be obtained. Then, the algorithm updates the history of pixels and applies the effect of the captured frame on the updated history. In the next step, moving pixels are identified to specify the moving objects. After that, steps 5 and 6 of the initializing algorithm are repeated again. For each pixel of the input frame, the cluster of that pixel is calculated and then its history is updated likewise. In addition, the corresponding pixel of the extracted background is assigned by the mean value of the gray scale cluster that has visited most so far. The moving pixels are identified in the form of a binary image by subtracting of each pixel of the background image extracted in the previous step from the corresponding pixel in the current image. Equation 10 calculates the distance between pixel (i, j) from the current frame and its corresponding pixel in the temporal background.

$$Diff(i, j) = |Cur\_img(i, j) - Tem\_BK(i, j)| \quad (10)$$

In the next step, according to Eq. 11, if the computed difference value is not more than two thresholds, ∂ and φ, its corresponding pixel belongs to the static image. Else, it is a pixel of a moving object. ∂ and φ are two thresholds which are calculated by the gray scale difference histograms. We compute the value of these two parameters using the method explained by Chiu *et al*. (2010):

$$\begin{aligned}&\text{If (Diff } (i, j) \le \partial \text{or Diff } (i, j) \le \varphi)\\&\text{Pixel}(i, j) \text{ belongs to the static background}\\&\text{Else}\\&\text{It is a pixel of a moving object}\end{aligned} \quad (11)$$

In both analog and digital signals, noise is an unwanted perturbation to a wanted signal. Here, after constructing the binary image, there are some scattered pixels which the algorithm has detected them as the moving object pixels incorrectly. By applying a simple noise reduction technique, the quality of the binary image is improved.

We use a binary median filter to decrease the effect of the existent noises in the binary image. In this filter, the number of pixels with value equal to 1 in a local area around of each pixel of the binary image is counted. If this amount is less than a static threshold, δ, that pixel is detected as a noise pixel and its value is changed to zero.

In this research, the value of δ is set to 2. This value is achieved experimentally and after that we examined different values. In addition, the pixels of a 3×3 square area around of each pixel are assumed as its local neighbors.

At the end, from the combination of the current frame and the temporal extracted background, the final background image is resulted. This combination is done according to the values of the binary image pixels. The binary image is the output of the previous stage explaining the moving pixels. If the value of the binary pixel is one, final background is assigned to the value of correspond pixel in the adapted background. Else, it will be replaced with a weighted combination of the corresponding pixel of the current image and value of the pixel in the temporal extracted background (Eq. 12):

$$\begin{aligned}
&\text{If } (\mathrm{Bin\_BG}(i,j) = 0\\
&\quad \mathrm{Fin\_BG}(i,j) = \mathrm{Temp\_BG}(i,j)\\
&\text{Else}\\
&\quad \mathrm{Fin\_BG}(i,j) = (1-a) \times \mathrm{Temp\_BG}(i,j) \; +\\
&\quad \alpha \times \mathrm{Initial\ \_BG}(i,j)
\end{aligned} \tag{12}$$

In this formula, Bin_BG, Temp_BG, Fin_BG and initial_BG refer to the value of the pixel (i, j) in the Binary image, the Extracted temporal background image, the Final background image and the Initial background image extracted in the initialization phase, respectively. The value of w is also set to 0.1.

RESULTS

In order demonstrated to evaluate the effectiveness of the proposed system and compute its efficiency in comparison to the other related works, we implemented our algorithm and four algorithms from recent studies, namely Xiaofei (2009); Kong *et al*. (2007); Chiu *et al*. (2010) and Cui *et al*. (2009). All the algorithms were implemented using VC++ 2008. They were run on a Pentium 3 GHz PC under the Windows XP Operation System. To have a fair experiment, we tested all of these five systems under the same conditions. We used the standard database ordered by PETS (IEEE International Workshop on Performance Evaluation of Tracking and Surveillance). Actually, the PETS2000 (P0), PETS2001 during the day (PD) and PETS2001at night (PN) were used. These datasets can be found at http://ftp.pets.rdg.ac.uk/. In addition, some other data sets were used as well. We examined the systems using Campus (CM), Fountain (FT), Shopping Mall (SM), Bootstrap (BS), Escalator (ES), Hall Monitor (HM) and Lobby (LB). These data are accessible at http://perception.i2r.a-star.edu.sg/bk_model/bk_index. Html.

In this research, the experiments were repeated several times using various frames a different number of frames. Also, the experimental results were investigated quantitatively and qualitatively.

Figure 3 illustrates the results of the proposed method using various data sets and under different conditions in comparison to the others. It also plots the binary images extracted by these methods. A binary image shows the moving object as the corresponding method expects. The figure illustrates that the performance of the proposed method is acceptable.

To evaluate the efficiency of our algorithm quantitatively compare it with the other algorithms, four different scales were utilized:

- The time consumptions of the algorithms were calculated to compare the time complexity. The averages of the time cost of the methods are shown in the first vertical section of Table 1
- The Root Mean Square error (RMS error) is another scale used. The average values of this factor for all investigated methods are shown in the second vertical section of Table 1
- The False Detection Rate is the third quantitative factor used in this research. Actually, it is the number of pixels which the difference between their values and the values of corresponding pixels in the static background image are more than a user predefined threshold (γ). The meaning of the acquired values in five algorithms is mentioned in the third vertical section of Table 1. We have set (γ) to zero
- Finally, as the last quantitative factor, the Peak Signal to Noise Ratio (PSNR) is calculated. The means of this scale value for five methods are illustrated in the last vertical section of Table 1

Table 1: The meaning of quantitative factors' values in 5ooth Frame

	Time (ms)					RMS error (pixel)			
Proposed system	P0	PD	PN	CM	FT	P0	PD	PN	CM
	69	67	67	13	13	86,686,258	60,200,247	27,493,694	74,767,504
	SM	BS	ES	HM	LB	SM	BS	ES	HM
	51	12	13	16	12	66,742,506	56,086,404	35,809,979	49,782,035
(Xiaofei, 2009)	P0	PD	PN	CM	FT	P0	PD	PN	CM
	57	58	56	11	11	95,533,604	661, 17,002	33,144,562	83,332,336
	SM	BS	ES	HM	LB	SM	BS	ES	HM
	43	10	10	14	11	162,774,840	70,322,391	52,689,689	84,586,145
(Kong *et al*., 2007)	P0	PD	PN	CM	FT	P0	PD	PN	CM
	59	59	84	13	11	250,230,950	113,879,167	31,610,243	79,878,073
	SM	BS	ES	HM	LB	SM	BS	ES	HM
	45	10	10	14	11	223,097,532	7, 366, 0431	66,546,876	116,614,946
(Cui *et al*., 2009)	P0	PD	PN	CM	FT	P0	PD	PN	CM
	5	6	5	1	0.6	153,343,009	60,109,488	34,669,666	81,742,336
	SM	BS	ES	HM	LB	BS	ES	HM	LB
	2.7	1	1.4	1.3	1.2	109,432,166	56,564,428	35,507,989	61,156,282
	P0	PD	PN	CM	FT	P0	PD	PN	CM
	85	84	83	16	15	8, 2425, 922	61,979,850	31,610,243	85,970,819
(Chiu *et al*., 2010)	SM	BS	ES	HM	LB	SM	BS	ES	HM
	63	14	15	19	15	162,913,613	70,592,044	52,577,055	84,492,468

Table 1: Continues

	RMS error (pixel)	FDR (pixel)					PSNR				
Proposed System	FT	P0	PD	PN	CM	FT	P0	PD	PN	CM	FT
	30,159,569	97,846	88,404	93,840	19,775	18,191	23.21	32.59	25.21	15.12	20.15
	LB	SM	BS	ES	HM	LB	SM	BS	ES	HM	LB
	9,035,222	70,052	18,397	18,265	23,132	16,342	22.93	13.09	17.04	21.15	31.89
(Xiaofei, 2009)	FT	P0	PD	PN	CM	FT	P0	PD	PN	CM	FT
	35,032,305	96,564	78,882	96,012	19,835	17,870	25.3	30.43	23.46	14.52	19.3
	LB	SM	BS	ES	HM	LB	SM	BS	ES	HM	LB
	10,372,085	77,274	18580	18,805	23,886	15,865	17.31	11.54	16.64	16.55	30.28
(Kong *et al*., 2007)	FT	P0	PD	PN	CM	FT	P0	PD	PN	CM	FT
	43,030,815	107,945	87,230	95,629	19,901	19,304	20	26.61	23.7	14.55	17.82
	LB	SM	BS	ES	HM	LB	SM	BS	ES	HM	LB
	22,651,359	79,909	18,677	18,921	24,757	19,596	16.25	10.43	15.12	15.43	25.74
(Cui *et al*., 2009)	FT	P0	PD	PN	CM	FT	P0	PD	PN	CM	FT
	33,892,701	105,957	80,483	98,809	19,971	19,204	23.26	31.84	21.85	14.12	19.65
	LB	SM	BS	ES	HM	LB	SM	BS	ES	HM	LB
	29,073,926	78,224	18,354	18,467	23,765	19,502	22.25	12.15	18.18	18.40	25.16
(Chiu *et al*., 2010)	FT	P0	PD	PN	CM	FT	P0	PD	PN	CM	FT
	34,753,926	95,713	78,765	95,629	19,850	17,870	25.44	30.93	23.7	14.3	19.32
	LB	SM	BS	ES	HM	LB	SM	BS	ES	HM	LB
	100, 95, 435	76,362	18,581	18,852	23,847	15,842	15.77	11.4	16.05	16.22	30.53

DISCUSSION

It is clear from Table 1 that the time consumption of Chiu *et al*. (2010) in comparison to the others is less. Actually, the time complexity of their main algorithm (the Extraction Background Algorithm) is O (N^2) versus the others that their time complexity is O (N^3). N is the size of each dimension of the images. This difference is resulted from the fact that in their method just the previous extracted background is participated to update the value of each pixel in the new extracted background image. The others use N previous background images. In spite of this preference, the quality of their extracted backgrounds is not comparable to our results. As it is clear in Fig. 3, our extracted backgrounds using various datasets are more acceptable. They are more adaptable to changes occurring in the existence of moving objects especially in the crowded situations. In addition, all the quantitative factors used to evaluate the quality of output of the methods certify that our results are in the minimum state 20% better than the others.

Future works: In this research we save the history of each pixel separately. Using a blocking technique reduces the time complexity of the algorithm and strengthens it against noise caused by small motions. Also, it seems that it is not necessary to adapt the background in each frame.

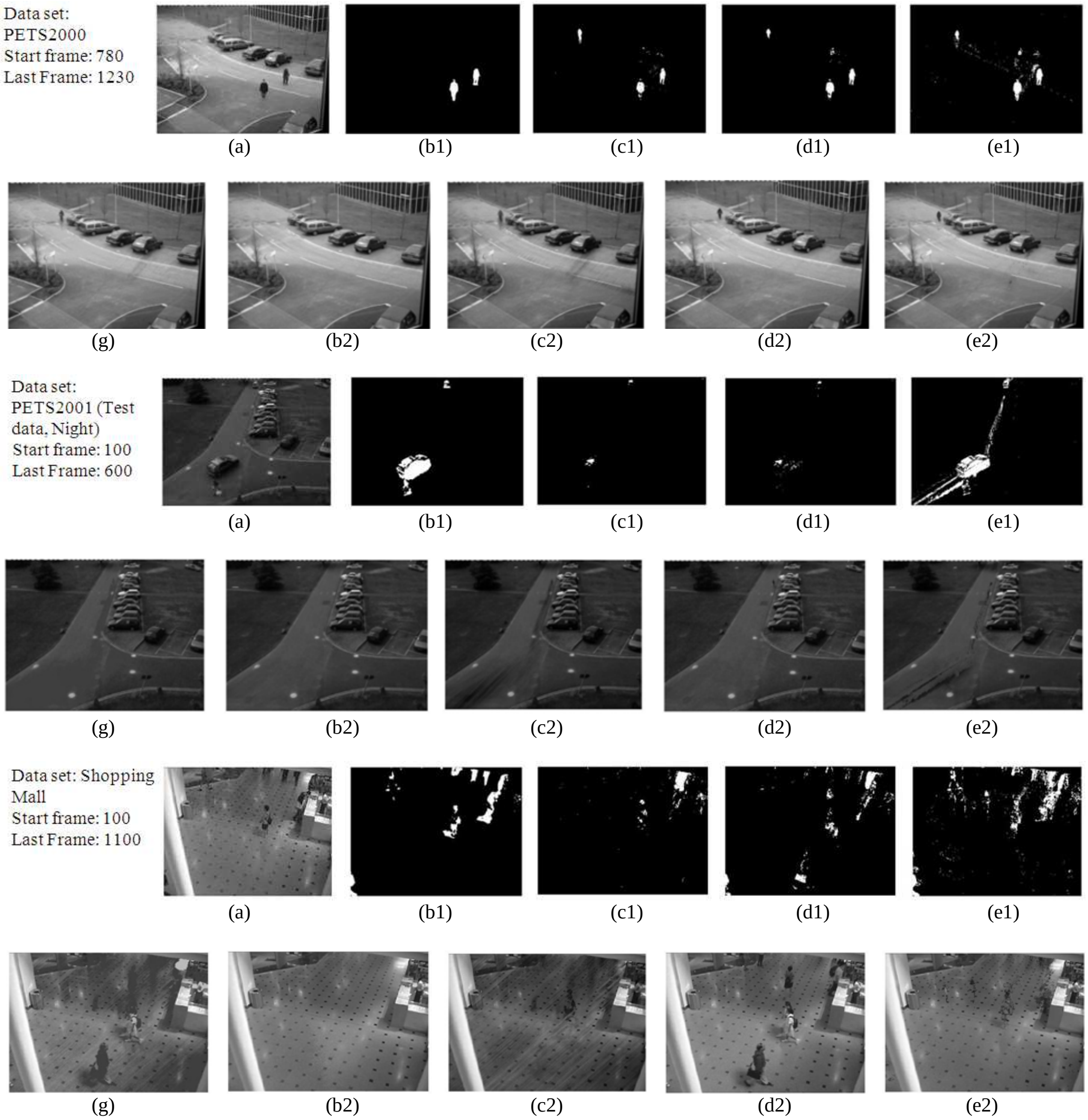


Fig. 3: Experimental Results of five algorithms in different situations. The Current frame (a) The Extracted Binary Image by the proposed method (b1), Kong *et al*. (2007) (c1), Chiu *et al*. (2010) (d1) and Cui *et al*. (2009) (e1) The Extracted Final Background by Xiaofei (2009) (g), the proposed method (b2), Kong *et al*. (2007) (c2), Cui *et al*. (2009) (d2) and Chiu *et al*. (2010) (e2)

Reducing such processing will result in executing the algorithm less time and thus reduce the execution time. As part of our future work, we intend to investigate these problems further.

CONCLUSION

In this study, we have introduced a hybrid background extraction system. In this system, after that background is initialized using a proposed method, the history of each pixel is saved separately. Using a probability-based method based on the saved history, it is detected that each pixel is a background pixel or a foreground pixel. Various quantitative and qualitative experiments show that the accuracy of the extracted background is better than the other methods which are re-implemented here and also the time consumption of the proposed method is acceptable.

REFERENCES

Armanfard, N., M. Komeili, M. Valizade, E. Kabir and S. Jalili, 2009a. A non-parametric pixel-based background modeling for dynamic scenes. Proceedings of the International Conference on Advances in Computational Tools for Engineering Applications, Jul. 15-17, IEEE Xplore Press, Zouk Mosbeh, Lebanon, pp: 369-373. DOI: 10.1109/ACTEA.2009.5227894

Armanfard, N., M. Komeili, M. Valizadeh and E. Kabir, 2009b. Effective Hierarchical background modeling and foreground detection in surveillance systems. Proceedings of the 14th International CSI Computer Conference, Oct. 20-21, IEEE Xplore Press, Tehran, pp: 164-169. DOI: 10.1109/CSICC.2009.5349318

Barnich, O. and M.V. Droogenbroeck, 2009. VIBE: A powerful random technique to estimate the background in video sequences. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Apr. 19-24, IEEE Xplore Press, Taipei, pp: 945-948. DOI: 10.1109/ICASSP.2009.4959741

Chiu, C.C., M.Y. Ku and L.W. Liang, 2010. A robust object segmentation system using a probability-based background extraction algorithm. IEEE Trans. Circ. Syst. Video Technol., 20: 518-528. DOI: 10.1109/TCSVT.2009.2035843

Cui, B.X., S.M. Sun and Y. Duan, 2009. Vehicle detection based on adaptive background. Proceedings of the 2nd International Workshop on Knowledge Discovery and Data Mining, Jan. 23-25, IEEE Xplore Press, Moscow, pp: 821-824. DOI: 10.1109/WKDD.2009.117

Heikkila, M. and M. Pietaikinen, 2006. A texture-based method for modeling the background and detecting moving objects. IEEE Trans. Patt. Anal. Mach. Intell., 28: 657-662. DOI: 10.1109/TPAMI.2006.68

Kong, J., Y. Zheng, Y. Lu and B. Zhang, 2007. A Novel background extraction and updating algorithm for vehicle detection and tracking. Proceedings of the 4th International Conference on Fuzzy Systems and Knowledge Discovery, Aug. 24-27, IEEE Xplore Press, Haikou, pp: 464-468. DOI: 10.1109/FSKD.2007.98

Li, L., W. Huang, I.Y.H. Gu and Q. Tian, 2004. Statistical modeling of complex backgrounds for foreground object detection. IEEE Trans. Image Process., 13: 1459-1472. DOI: 10.1109/TIP.2004.836169

Peng, J.W. and W.B. Horng, 2009. Real-time dynamic background segmentation based on a statistical approach. Proceedings of the IEEE International Conference on Networking, Sensing and Control, Mar. 26-29, IEEE Xplore Press, Okayama, pp: 404-409. DOI: 10.1109/ICNSC.2009.4919310

Shakeri, M., H. Deldari, H. Foroughi, A. Saberi and A. Naseri, 2008. A novel fuzzy background subtraction method based on cellular automata for urban traffic applications. Proceedings of the 9th International Conference on Signal Processing, Oct. 26-29, IEEE Xplore Press, Beijing, pp: 899-902. DOI: 10.1109/ICOSP.2008.4697273

Sheng, Z. and X. Cui, 2008. An adaptive learning rate GMM for background extraction. Proceedings of the International Conference on Computer Science and Software Engineering, Dec. 12-14, IEEE Xplore Press, Wuhan, Hubei, pp: 174-176. DOI: 10.1109/CSSE.2008.963

Stauffer, C. and W.E.L. Grimson, 1999. Adaptive background mixture models for real-time tracking. Proceedings of the IEEE Computer Society Conference on Computer Vision Pattern Recognition, Jun. 23-25, IEEE Xplore Press, Fort Collins, CO. DOI: 10.1109/CVPR.1999.784637

Xiaofei, L., 2009. A new fast hybrid algorithm of background extraction suitable for the DSP platform. Proceedings of the WRI World Congress on Computer Science and Information Engineering, Mar. 31-Apr. 2, IEEE Xplore Press, Los Angeles, CA., pp: 1-4. DOI: 10.1109/CSIE.2009.406