# Motion Segmentation from Color Video Sequences based on AMF

Alla Kim*, Yoon-Ho Kim**

**ABSTRACT**

A process of identifying moving objects from data is typical task in many computer vision applications. In this paper, we propose a motion segmentation method that generally consists from background subtraction and foreground pixel segmentation. The Approximated Median Filter (AMF) was chosen to perform background modelling. To demonstrate the effectiveness of proposed approach, we tested it gray-scale video data as well as RGB color space.

**Key Word**

background subtraction, background modeling, approximated median filter (AFM), gray-scale, color motion segmentation

## Ⅰ. Introduction

In order to identify moving objects from video sequence we used motion segmentation procedure in static scenes that includes background subtraction and foreground pixel segmentation. The process of separating background and object is a common problem in vision system applications like video surveillance, object tracking, object detection and so on. An important condition of motion segmentation technique is that motion pixels of the moving objects across the sequences were defined as much accurately and carefully as possible. That's for sure very fragile condition. Motion segmentation is based on two assumptions: any static or periodically moving parts of a scene that lies outside the object(s) is considered as background. In the paper technique of background modelling based on the Approximated Median Filter (AFM) was used. Following motion segmentation step, moving objects are detected.

The organization of paper is as follows: in section Ⅱ a description of background research is given. In section Ⅲ detection of moving objects

* 1'st author: Mokwon univ. , Div. of computer eng. , (kimalusha@mokwon.ac.kr)
** Corresponding author: Mokwon univ. , Div. of computer eng. , (yhkim@mokwon.ac.kr)
#논문번호 : KIIECT2009-03-09 #접수일자 : 2009.08.17 #최종논문접수일자 : 2009.09.09

in a video sequence is proposed. In section Ⅳ the result of experiment is presented and the system is evaluated. Finally section Ⅴ is devoted to conclusion and pointing on some further issues.

## Ⅱ. Background

Background subtraction algorithms can be divided on 4 steps: preprocessing, background modelling, background detection and data validation[1][2]. In general, Basic idea of background subtraction is subtracting current image from background and a comparison of all pixels with a threshold for dividing pixels on foreground and background. Background subtraction algorithms can be classified into two large groups; non-recursive and recursive tech. [3]. Such schemes as frame differencing, median filter and linear predictive filter are determined as non-recursive algorithms, approximated median filter, Kalman filter and Mixture of Gaussian are in opposite. In a pre-processing stage raw input video is changed in a form that can be used and conformed to an algorithm's goal and capability of acquisition device. Image processing tasks as frame, color, noise reduction and spatial operations are commonly used in most visual applications on

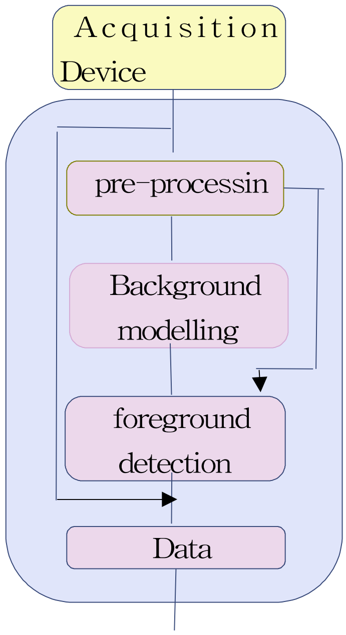


Fig 1. Subtraction algorithm

the first stage. To make a real-time system some size and resolution processed can be done in order to suppress an amount of input information, also conditions of storing data should be considered properly.

Background modeling is a description of the current background scene. The simplest way to model the background is to acquire a background image which doesn't include any moving object. In some environments, the background isn't available and can always be changed under critical situations like illumination changes, objects being introduced or removed from the scene. To take into account these problems of robustness and adaptation, many background modeling methods have been developed. Recently. They can be categorized as follows: Basic Background Modelling, Statistical, Fuzzy and Background Estimation. Also classification can be made in term of recursion, prediction, adaptation, modality and so on[4].

Foreground detection supposed to check whether the input pixels are foreground or background. The most commonly approach to find all pixels that match next equitation:

$$|I_t(x,y) - B_t(x,y)| > T \qquad (1)$$

Another popular foreground detection scheme is to threshold based on the normalized statistics:

$$\left| \frac{I_t(x,y) - B_t(x,y) - \mu_d}{\sigma_d} \right| > T_s, \qquad (2)$$

where $\mu_d$ and $\delta_d$ are the mean and the standard deviation of $I_t(x,y) - B_t(x,y)$ for all spatial locations (x, y). Most schemes determine the foreground threshold T or $T_s$ experimentally. Of course in ideal threshold should be a function of the spatial location, that it could be scaled for different types regions. For example the threshold must be higher for regions with high contrast and smaller for low contrast regions.

Another approach to introduce spatial variability is to use two thresholds with hysteresis. The basic idea is that first threshold will identify pixels whose absolute differences with the background estimates exceeded a large threshold. Then, foreground regions are grown from strong foreground pixels by including neighboring pixels with absolute differences larger then a smaller threshold. So algorithm will perform foreground detection by using a two-pass foreground detection scheme[3]. Data validation process aims to improve output. Commonly a lot of different techniques can be used to preserve false-positive and false -negative result.

## III. Proposed method of moving objects detection in a video sequence

Motion segmentation was conducted by using Matlab toolbox. Single static 1.3 mega pixel PC "Wezel" webcxe was used to acquire video sequence [8]. Background modelling (also referred to as background subtraction) was based on AMF that also has been used for classification system and urban traffic monitoring[5][6]. In this paper we focus on the problem of motion segmentation and though many algorithms have been proposed in the literature we will try to extend the AFM scheme for RGB color space[9].

The main disadvantage of median filtering is that the previous frames of video are buffered and background modeled as the median of buffered frames. And though even if median filtering is robust enough it'll require a lot of memory space. Due to the success of non-recursive median filtering, AMF was devised as efficient recursive approximation of the median filter by McFarlane and Schofield[7].

In order to implement AMF, we use next classification if a pixel value in the current frame has a value larger than the corresponding background pixel, the background pixel is incremented by 1. Otherwise the background is decremented by one. In this way input data comes to a state when half of pixels are greater then the background and half are less than the background. The AMF does good job in separating moving object across video sequences. The pseudocode written below gives a brief introduction to AMF.

```
Algorithm for AMF
 1) B and F are sets of RGB describing background and foreground pixels
 Initialization
 2) X is a test sequence
 3) F is output foreground sequence
 4) Foreground detection
 FOR x = 1 to w
   FOR y = 1 to h
        IF all pixels of test sequence for all color spaces (R,G,B respectively) are within the distance T of corresponding background value B(x,y) THEN
          F(x, y) = 0
            ELSE
          F(x, y) = X(x,y)
        ENDIF
5) The background model B={}
      IF a pixel value in the current frame has a value larger than the corresponding background pixel THEN
          B(x, y, z) = B(x,y,z) + 1 //z is color space index
            ELSE
          B(x,y,z) = B(x,y,z) - 1;
      ENDIF
 ENDFOR
```

Foreground pixel are detected by calculating the Euclidean norm at time.

$$\| I_t(x,y) - B_t(x,y) \| > T_e \qquad (3)$$

where $I_t$ is the pixel intensity value, $B_t$ is the background intensity value at time t and $T_e$ is the foreground threshold.

$$I_t = [I_{1,t} \; \cdots \; I_{c,t}]^T \quad (4)$$

$$B_t = [B_{1,t} \cdots B_{c,t}]^T$$

Where c is the number of image channels[5]. The foreground threshold $T_e$ determined experimentally.

## IV. Experimental Result

We acquired 2 video sequences in order to test motion segmentation: a) video sequence with RGB human motion and b) video sequence with pen motion on uniform background.

First sequence contains RGB human in-doors motion. The result is presented in figure 3.

Fig 2. Original RGB frames.

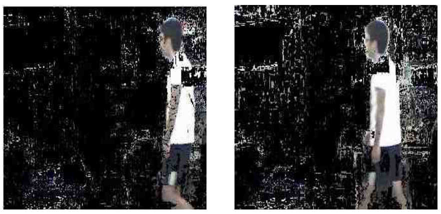

Fig 3. Result image with shadows

As shown above result image (Fig. 3) contains some noise and in some frames even trails behind the object or shadows. Shadows are caused by illumination effects so common remedial is converting an RGB color space to normalized RGB that should remove the effect of any intensity variations.

We normalized RGB as follows:

1) Every frame from video sequence was splitted into three separated red, green and blue component images. For every pixel (5) in component images eq. (6) was calculated.

$$Z = \begin{vmatrix} Z_R \\ Z_G \\ Z_B \end{vmatrix} \quad (5)$$

Where Z is a color pixel, ZR, ZG, ZB are all color components respectively.

$$z = \begin{vmatrix} Z_r \\ Z_g \\ Z_b \end{vmatrix} \Rightarrow \begin{bmatrix} r = \dfrac{R}{\sqrt{R^2+G^2+B^2}} \\ g = \dfrac{G}{\sqrt{R^2+G^2+B^2}} \\ b = \dfrac{B}{\sqrt{R^2+G^2+B^2}} \end{bmatrix} \quad (6)$$

Where z is normalized color pixel, color components defined as follows. Normalized lanes were restacked again to form normalized RGB image as presented in Figure 4.

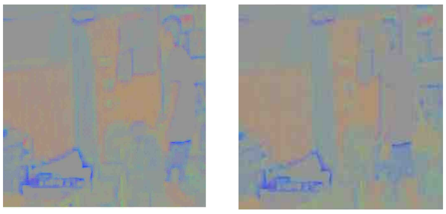

Fig 4 Normalized frames

Figure 5 shows motion segmentation performed on video sequence that was preprocessed as described above, but because some particularly important

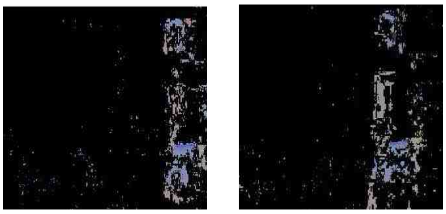

Fig 5 Result image after illumination removing

pixels were removed to and preprocessing step distorted color information (black parts of his shorts became blue), we decided to keep shadows but safe color information as though my application is color based. Video sequence#2 contains pen motion on uniform background: green wall of cube (Figure 6).

This avi-file was converted into gray-scale color space and then the AFM was applied. The result is presented in Figure 7.

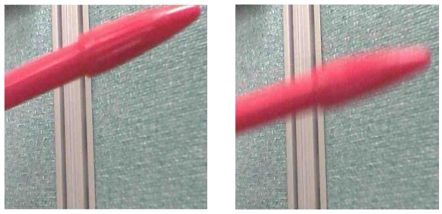

Fig 6. Original frames (test 2)

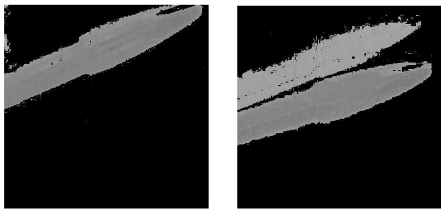

Fig 7. Result gray-scale image

The same sequence was processed in RGB color space without any normalization (for comparison experimental result with first test data input). Result is presented in Figure 8.

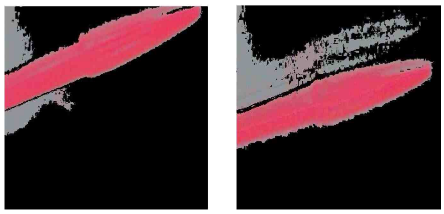

Fig 8. Result RGB-color images

Result images (Figure 7, 8) have some trails cause background updating rate is relatively high: 30 fps (frames per second). Applying to some data validation procedures can significantly improve data output.

Commonly for background subtraction algorithm and color classifier as well, binary mask is used in order to mark all matched components. However I wanted to keep color information for some future work. So I focused on RGB space and implemented RGB normalization on first video sequence to check whether output is better then not normalized RGB motion segmentation. The test didn't satisfy my color criteria (almost all colors were distorted as you can see in Figure 5) and I preferred not normalized output though it is more corrupted with noise (Figure 3 and 8) compare with normalized one.

Though the AMF has a good performance it adapts very slowly to a large change in background and sensitives to environmental noise. So low frame rate can also improve the result. All others motion segmentation details are described in a section II and III.

## IV. Conclusion and Future Work

In this paper, we have developed a motion-segmentation technique using background segmentation and tested it both for gray-scale and RGB color space. By using AMF, background image can be subtracted without any penalty on segmentation efficiency. To verify the performance of proposed algorithm, some experiments have conducted and result in meaningful outcomes. Future work will include next aims: 1) Elaborating data validation procedures, especially RGB shadow removal techniques and 2) Also need to implement some ideas about color detection system similar that mentioned in reference list [10].

## References

[1] Gary Brodski and Adrian Kaehler, "Learning OpenCV", O'Reilly, 2008.

[2] Thanarat Horprasert, David Harwood, Larry S. Davis, “A statistical approach for real-time robust background subtraction and shadow detection”, 1999.

[3] Sen-Ching S. Cheung, Chandrika Kamath, “Robust techniques for background subtraction in urban traffic video”, proceedings of the SPIE, Vol. 5308, pp. 881-892, 2004.

[4] T. Bouwmans, F. E Baf, B.Vachon, “Background Modeling using Mixture of Gaussians for Foreground Detection - A Survay”, Recent Patents on Computer Science, Vol. 1, no. 3, 219-237, 2008.

[5] Swantje Johnsen and Ashkey Tews, “Real-Time Object Tracking and

Classification Using a Static Camera", Proceedings of the IEEE ICRA 2009,
Workshop on People Detection and Tracking, Kobe, Japan, May, 2009.

[6] P.Remagnino et al., "An integrated traffic and pedestrian model-based vision system", Proceedings of BMVC97, Vol. 2, Colchester, 8-11 th September, University of Essex, UK, pp 380-389, 1997.

[7] N.McFarlane and C. Shofield, "Segmentation and tracking of piglets in images", Machine Vision and Applications, Springer, Vol. 8, no.3, 1995.

[8] Seth Benton, "Background subtraction, Matlab Models", 2008.

[9] Rastislav Lukac, Konstantinos N. Plataniotis, "Color image processing: methods and applications", Published by CRC Press, 2006.

[10] Turgay Celik, Hasan Demirel, Huseyin Ozkaramanli, Mustafa Uyguroglu, "Fire detection using statistical color model in video sequences", Journal of Visual Communication and Image Representation, Vol. 18, Issue 2, 2007.

## 저자약력

알라김(Alla Kim)

2008 B.S. degree in Department of Economics and Management, Tashkent University of Informational Technologies.
2008~ M.D. degrees in Department of Computer Engineering, Mokwon University.

<관심분야> image processing, video processing

김윤호(Yoon-ho Kim)

Yoon-ho Kim received the M.Sc degree in electronic engineering from the Kyeong-Hee University, and the ph. D. degree in electronic engineering from the Cheong-ju University, in 1984 and 1992, respectively. From 2005 to 2006, he was a Research Fellow with the Dept. of computer science, University of Auckland, Auckland. From 1992, he joined a Faculty Member with the Dept. of computer engineering, Mok-won University. He is currently a Professor with the Division of computer engineering, Mok-won university. His research focus for the last five years has been on pattern recognition, vision, and watermarking, and social security. Recently, he has a Korea delegate for ISO/TC223 Societal security.