A Novel Adaptive Spatio-temporal Thresholding Policy for Background Subtraction Algorithms

Manish Kakhani
manishkakhani@gmail.com
Faculty of Engineering and Technology,
MITS University, Laxmangarh, Sikar, Rajasthan, INDIA

Abstract

Automatic segmentation of foreground from background in video sequences has attracted lots of attention in computer vision.A commonly used technique is background subtraction. In background subtraction technique, each frame is compared against the background model or reference frame. If the difference is greater than given threshold, than it is considered to be foreground otherwise it is background. A thresholding policy plays a critical role in identifying foreground object in a video sequence. In this paper, we propose a new thresholding policy that uses spatial and temporal information of input video sequence for foreground detection. Our thresholding policy embedded with the existing algorithms and results shows significant improvement in foreground detection by existing algorithms.

Key words- background subtraction, thresholding, foreground detection

I. INTRODUCTION

The capability of extracting moving objects from a video sequence is a fundamental and crucial problem of many vision systems that include video surveillance [1, 2], traffic monitoring [3], human detection and tracking for video teleconferencing or human-machine interface [4, 5, 6], video editing, among other applications. A common approach to identifying the moving objects is background subtraction, where each video frame is compared against a reference or background model. Pixels in the current frame that deviate significantly from the background are considered to be moving objects. These foreground pixels are further processed for object localization and tracking. [6] There is a huge amount of research is already done in the area of background subtraction but most of the existing algorithms lack good thresholding policy. These algorithms use fixed value of threshold and it is set manually. [8] The major drawback of such approach is the value has to be change for each input video. It produces poor results when it mismatched the pixel intensity value of moving object. If it is too high or too low it produces erroneous results containing either noise or no object at all. The value of threshold must be matched with pixel value of foreground object for effective detection. Quality of foreground detection depends mostly on thresholding policy used. The better the thresholding policy, the more good results can be achieved.

Some of the development in the area of thresholding algorithm we can found in [8, 9, and 10]. There is a huge amount of work on iterative thresholding algorithms, and we cannot mention all of them here; the interested reader is referred to [8, 9].Arian Maleki et al. describes coherence Analysis of iterative thresholding algorithms in [8]. In their paper, they analyzed iterative hard and soft thresholding, and proved that under certain conditions they work properly. Their algorithms are very simple to implement and much faster than both convex relaxation and greedy methods and they are much more desirable for large scale problems. Joshua Migdal et al. used markov thresholds for background subtraction in [10]. They had developed a novel approach to the process of background subtraction that exploits the spatial and temporal dependencies objects in motion impose on their images. They achieved this through the development and use of Markov random fields during the subtraction process. Finding a good thresholding policy is hot research area in the field of background subtraction. To find out a best possible thresholding policy is still a big hurdle in background subtraction algorithms.

In this paper, we are proposing a new thresholding policy which is based on spatial and temporal information of the input video. The new policy is embedded with existing background subtraction algorithms to find out its efficiency. We tested our method with frame differencing algorithm. The results are found to be promising after embedding of our approach with the existing algorithm. The rest of the paper is organized as follows: The proposed method is presented in section 2. Experimental results are presented in section 3.Finally; we conclude our paper and discuss future work in section 4.

II. PROPOSED METHOD

Our method involves a new thresholding policy which is based on spatial and temporal information of the input video frames. This method basically a part of background subtraction algorithms which ultimately improving the performance of the background subtraction algorithms. Our method is compared with the existing algorithms and has better foreground detection.

2.1 Proposed architecture

In fact, our architecture is extension of generic framework of basic background subtraction algorithm. Our thresholding policy is basically applied on the foreground detection step of background subtraction algorithm. Before discussing further, let's look for generic frame work of background subtraction algorithm.

Although there exist myriad of background subtraction algorithms but they all follow a general pattern of processing which is described in [6,7]. The four major steps in a background subtraction algorithm are preprocessing, background modeling, foreground detection, and data validation. Firstly, video frames captured from a camera are input to the background subtractor.Preprocessing stages are used for filtration and to change the raw input video to a processable format so that it can be further processed in subsequent steps. Background modeling then uses the observed video frame to calculate and update the background model that is representative of the scene without any objects of interest. Foreground detection is where the pixels that show a significant difference to those in the background model are flagged as foreground. Data validation is used to examine the found objects of interest and to eliminate any false matches. A foreground mask can then be output in which pixels are assigned as foreground or background.

Our adaptive spatio-temporal thresholding policy is applied at the foreground detection step of above frame work which improves the foreground detection significantly. The proposed architecture is illustrated in figure 1.

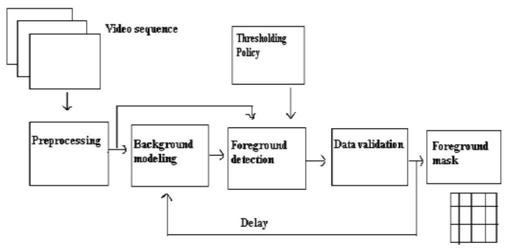


Figure 1. Proposed architecture

2.2 Proposed algorithm

Our proposed algorithm in detail is:

- Initially, we generate sequence of N frames from the input video.
- If it is first frame (f== 1), thresholding value T is equal to median value M of all the pixels values of the current frame.
- For rest of the frames i.e. (f>1), thresholding value T is equal to mean of median value of current frame and thresholding value of last frame.

The pseudo code of the above algorithm is illustrated in figure 2.

Thresholding policy (sequence of video frames)

```
Do (f: 1 to n)
{
  If (frame = = 1)
  {
     threshold = median (f)
   }
   Else
  {
   threshold = (median (f) + threshold (f-1))/2
   }
 }
```

Figure 2.Proposed algorithm

The median of all pixels in a frame is calculated by other algorithm. The algorithm for median calculation in a frame works as follows:

- Initialize a linear array (Array1) of length 255 and store '0' at every position.
- Initialize another two dimensional array (Array2) of size equal to frame dimension i.e. MxN.
- Count the different elements in the two dimensional array. (E.g. set [4, 4, 4, 1, and 1] would give three 4's and two 1's) and fill the linear array from backwards: put each element to its x^{th} position. Each time you put in a new element increase its value by one.
- Initialize two elements mid equal to (M*N)/2 and count equal to '0'.
- Run a loop from 0 to 255.Increase the count value by adding array elements to it. When the condition mid value is less than or equal to count value satisfies, ith element in linear array1 become the median value.

The pseudo code for calculation of median value in any frame is illustrated in figure 3

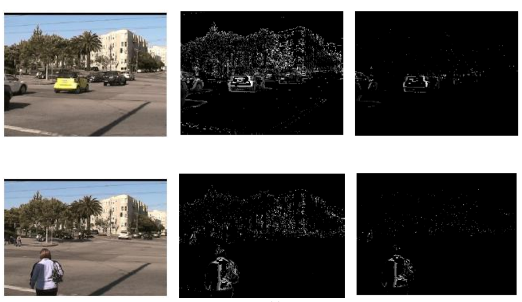
Median calculation (frame (MxN))

```
Array1 [0…255]
Do (i: 0 to 255)
{
  Array1 [i] =0
 }
 Array2 [M][N]
 Do (i: 0 to M)
 {
    Do (j: 0 to N)
    {
      x=Array2 [i][j]
      Array1[x] ++
     }
 }
 mid= (M*N)/2
 count=0
 Do (i: 0 to 255)
 {
   Count=Count+Array1 [i]
   If (mid<=count)
 {
    median=i
 }
 }
```

Figure 3. Algorithm for median calculation in a frame

III.EXPERIMENTAL RESULTS

This section demonstrates the performance of the existing frame differencing algorithm and our proposed algorithm on an image sequence .The proposed method has been tested on two input video files in avi format. The video files are taken from [11]. The sequence shown here are 320x240 dimensions. The two algorithms are compared on the parameter of foreground detection. In our experiment, it was observed that the result produced from our method is better as compare to existing frame differencing algorithm. There is a significant improvement in foreground detection and better segmentation can be seen from our method as shown below. The comparison of results obtained is illustrated in figure 4.



a. Input video b. Frame differencing c. Our method

Figure 4. Comparison of results

IV. CONCLUSION

In this paper, we propose a novel adaptive thresholding policy which is based on spatial and temporal information of video sequence. We embedded our proposed method with existing frame differencing algorithm and a comparison has been made between the two methods. It was found that results have been significantly improved. The experimental results show noteworthy improvement in foreground detection. The main advantages of our method are simplicity and reduced computational burden .Our method solved the problem of manual thresholding in existing algorithm. For future work, this approach can be further embedded with other more complex and sophisticated techniques and can be used for real time systems also.

References

[1] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: Who? When? Where? What? A real-time system for detecting and tracking people. In *Proc. the third IEEE Int'l Conf. Automatic Face and Gesture Recognition (Nara, Japan)*, pages 222–227. IEEE Computer Society Press, Los Alamitos, Calif., 1998.

[2] P.L. Rosin. Thresholding for change detection. In *Proc. IEEE Int'l Conf. on Computer Vision*, 1998.

[3] N. Friedman and S. Russell. Image segmentation in video sequences: A probabilistic approach. In *Proc. 13th Conf. Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1997.

[4] C.R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1998.

[5] J.Ohya and et al. Virtual metamorphosis. *IEEE Multimedia*, 6(2):29–39, 1999.

[6] Sen-Ching, S. Cheung, Chandrika Kamath, "Robust techniques for background subtraction in urban traffic video", Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, 2005

[7] Sarah Deane," A Comparison of Background Subtraction Techniques", 2007 .

[8] Arian Maleki, "Coherence Analysis of Iterative Thresholding Algorithms", Department of Electrical Engineering and Statistics, Stanford University,2009

[9] A. Maleki, D. L. Donoho,"Optimally Tuned Iterative Thresholding Algorithms," submitted to IEEE journal on selected areas in signal processing, 2009

[10] Joshua Migdal And W. Eric L. Grimson, "Background Subtraction Using Markov Thresholds", Computer Science And Artificial Intelligence Laboratory, Massachusetts Institute Of Technology, Cambridge, Massachusetts 02139, Proceedings Of The Ieee Workshop On Motion And Video Computing (Wacv/Motion'05),2005