A New Background Subtraction Method in Video Sequences Based on Temporal Motion Windows

Hamid Reza Shayegh Broojeni and Nasrollah Moghadam Charkari
Software Engineering Department, Tarbiat Modares University , Iran
shayegh@modares.ac.ir, moghadam@modares.ac.ir

Abstract– **Background detection and removal in video sequences for tracking motion objects is an important part of tracking process. It's clear that removal static background leads to tracking motion object and analyzing their movements more accurately. Older techniques for background removal, only use spatial features of image and remove background by focusing on color, texture …, but newer techniques improve results of background subtraction by using temporal features.**

In this paper we propose a novel method that uses combination of temporal and spatial features for background removal. In this method we reach to some motion windows by comparing and combining each frame with its neighbors, and obviously points out of these windows are parts of background. Then by other comparisons we detect static parts of motion windows and remove them. One advantage of this method is simplicity and the other is the ability of detection motion objects independent of their size and speed. Furthermore it has no noticeable overhead for real-time tracking process.

Keywords– **background subtraction, object tracking, temporal features.**

I. INTRODUCTION

All Automated surveillance systems require some mechanism to detect interesting objects in the field of view of the sensor. Such a mechanism serves as a form of focus of attention. Once objects are detected, the further processing for tracking and activity is limited in the corresponding regions of the image. In vision based systems, such detection is usually carried out by using background subtraction methods. These methods build a model of the scene background, and for each pixel in the image, detect deviations of pixel feature values from the model to classify the pixel as belonging either to background or to foreground.

Totally tracking motion objects in video sequences is a multiple process [1]. This processes show in Fig. 1. Background subtraction is a part of motion detection process that for this, we remove background, and only shadows, motion objects and motion noises (tree leafs for example) remain. After removing background, motion shadows and motion noises, we must track each object in the scene and pursue those. In the last, maybe a motion path analysis is required for example for behavior detection in a sport match.

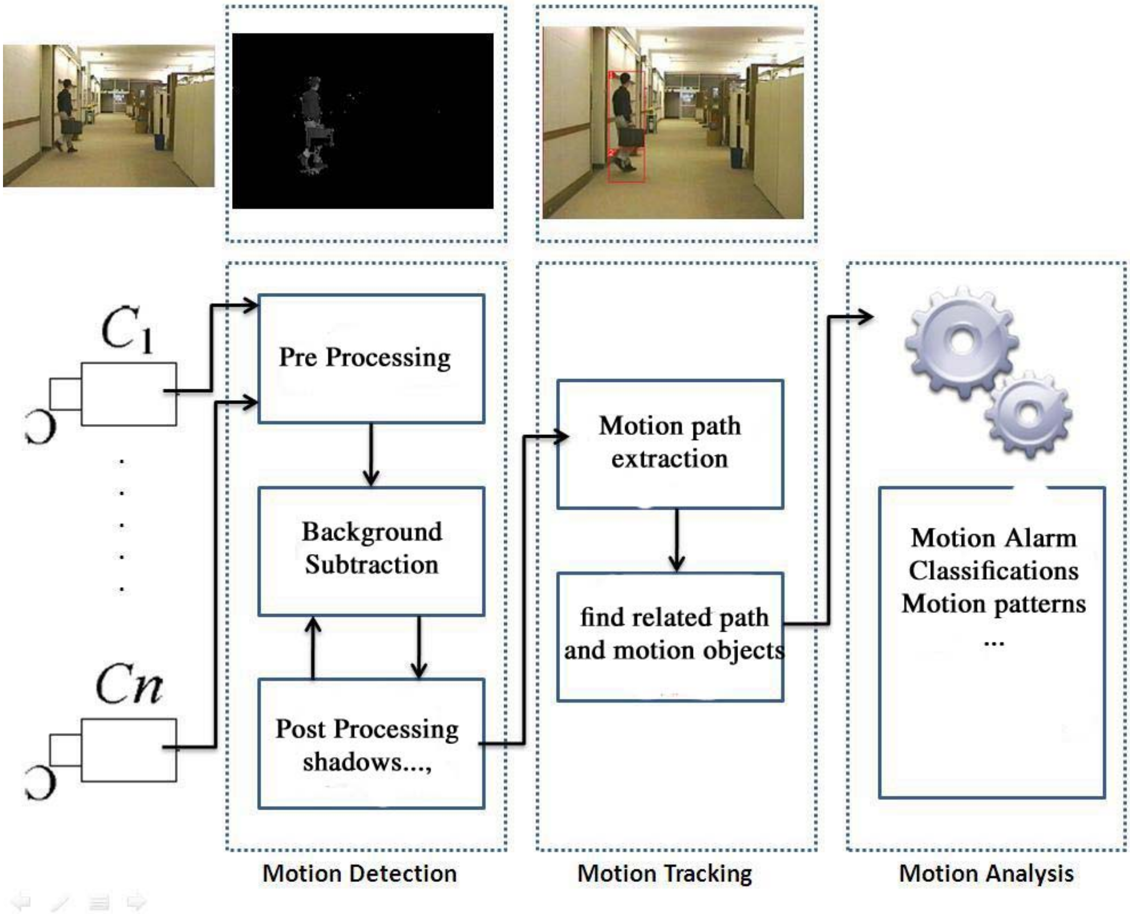

Fig. 1 Main tracking process

There are many methods in the field of background subtraction. Older methods are based on object features such as color [2, 3], intensity [4], edges [5, 6], texture [7]… These methods don't consider relationship between neighborhood frames. In the other hand, motion features don't use for classification of pixels in such methods. Motion features are considered in newer researches with more time complexity and better detection rate [1, 8, and 9].

The Background differencing methods have to deal with several problems in realistic environments. These problems have been discussed in detail by Toyama et.al [10]. Here we briefly describe some of the important problems which have not been addressed by most background subtraction algorithms.

- Initialization with moving objects: If moving objects are present during initialization then part of the background is occluded by moving objects. Thus many algorithms require a scene with no moving objects during initialization. This puts serious limitations on systems to be used in high traffic areas.
- Shadows. Objects cast shadows that might also be classified as foreground due to the illumination change in the shadow region. Sudden changes in illumination. Gradients of image are relatively less sensitive to changes in illumination and can be combined with color information effectively and efficiently to perform quasi illumination invariant background subtraction.

In this paper we proposed a new algorithm based on motion windows for background detection and subtraction. We use motion features of video sequences and decide about pixels of each frame that are a background pixel or not. This method solves three first described problems more than other similar. Our goal, entirely, is designing a robust surveillance system for object tracking and thus we consider fixed cameras in scenes and try to have a realtime background subtraction process. For this reason, our method is very simple, realtime and efficient.

In the continous of this paper, we present related works in this field, first. Then propose our method basically and explain it completely. The next section contains experimental results and comparisons of this with other related methods. We test our method on PETS2006 dataset. The conclusion is the final section of paper.

II. RELATED WORKS AND DEFINATIONS

First methods in background subtraction were based on object features. Methods based on geometry of motion objects [7, 8], colors, texture, light histograms [11] … are some examples of these. In all of these methods, we must have some information about motion object beforehand. These methods called pixel level methods. More efficient methods in this section use motion features (temporal features) for detecting objects. Optical flow techniques [12], frame difference [13], mixture of gussians [14] and so on, are in this group. These methods are more complex but, more accurate than simple methods. Frame level is the name of such methods. In many application areas, a combination of pixel and frame level methods is used for more efficiency [15, 16, and 17].

Many background subtraction algorithms for fixed cameras work by comparing color or intensities of pixels in the incoming video frame to a reference image. Jain et. al. [8] used simple intensity differencing followed by thresholding. Significant differences in intensity from the reference image were attributed to motion of objects. Azerbyjani et. al. [16] used color images and a statistical model of the background instead of a reference image. The color intensity at each pixel was modeled by a single Gaussian. Stauffer and Grimson [14] extended the uni-modal background subtraction approach by using an adaptive multi-modal subtraction method that modeled the pixel color as a mixture of Gaussians (MOG). This method could deal with slow changes in illumination, repeated motion from background clutter and long term scene changes. The model in Haritouglu et. al. [4] is a simplification of the Gaussian models, where the absolute maximum, minimum and largest consecutive difference values are used. All the above mentioned models use only color or intensity information for background differencing and are susceptible to sudden illumination changes. Moreover, these methods do not attempt to resolve the problem of motion of background object. Gao et.al [2] compare the assumption of a single vs. mixture of Gaussians to model the background color. They determine that mixture of Gaussian approach is indeed a better representation of backgrounds even in static scenes. Harville [5] presents a framework to update the mixture of Gaussians at each pixel based on feedback from other modules, for example tracking module, in a surveillance system. Pentland et. al. [12] used an eigen space model for background subtraction. The eigen background model cannot deal with relocation of a background object. Our work is most closely related to Jabri et.al [7]. They have used fusion of color and edge information for background subtraction. However the algorithm uses a pixel based fusion measure, such that either a large change in color or edges will result in foreground regions. Therefore their method cannot deal with sudden changes in illumination. The background edges are not modeled statistically. Moreover, this algorithm doesn’t present a solution to the relocation of background object problem. Horprasert et. al. [6] use brightness distortion and color distortion measures to develop an algorithm invariant to illumination changes. Li and Leung [10] use the fusion of texture and color to perform background subtraction. The texture based decision is taken over a small neighborhood. Ohta [11] defines a test statistic for background subtraction using the ratio of illumination intensities. Greiffenhagen et. al. [3] propose the fusion of color and normalized color information to achieve shadow invariant change detection. All these algorithms don't use regional information to validate local results. Also these algorithms do not attempt to solve the problem of relocation of background object. Toyama et. al. [15] propose a three tiered algorithm to deal with the background subtraction problem. The algorithm uses only color information at the pixel level. The region level deals with the background object relocation problem. Global illumination changes are handled at the frame level. This algorithm is able to handle sudden changes in illumination only if the model describing the scene after the illumination changes is known a priori.

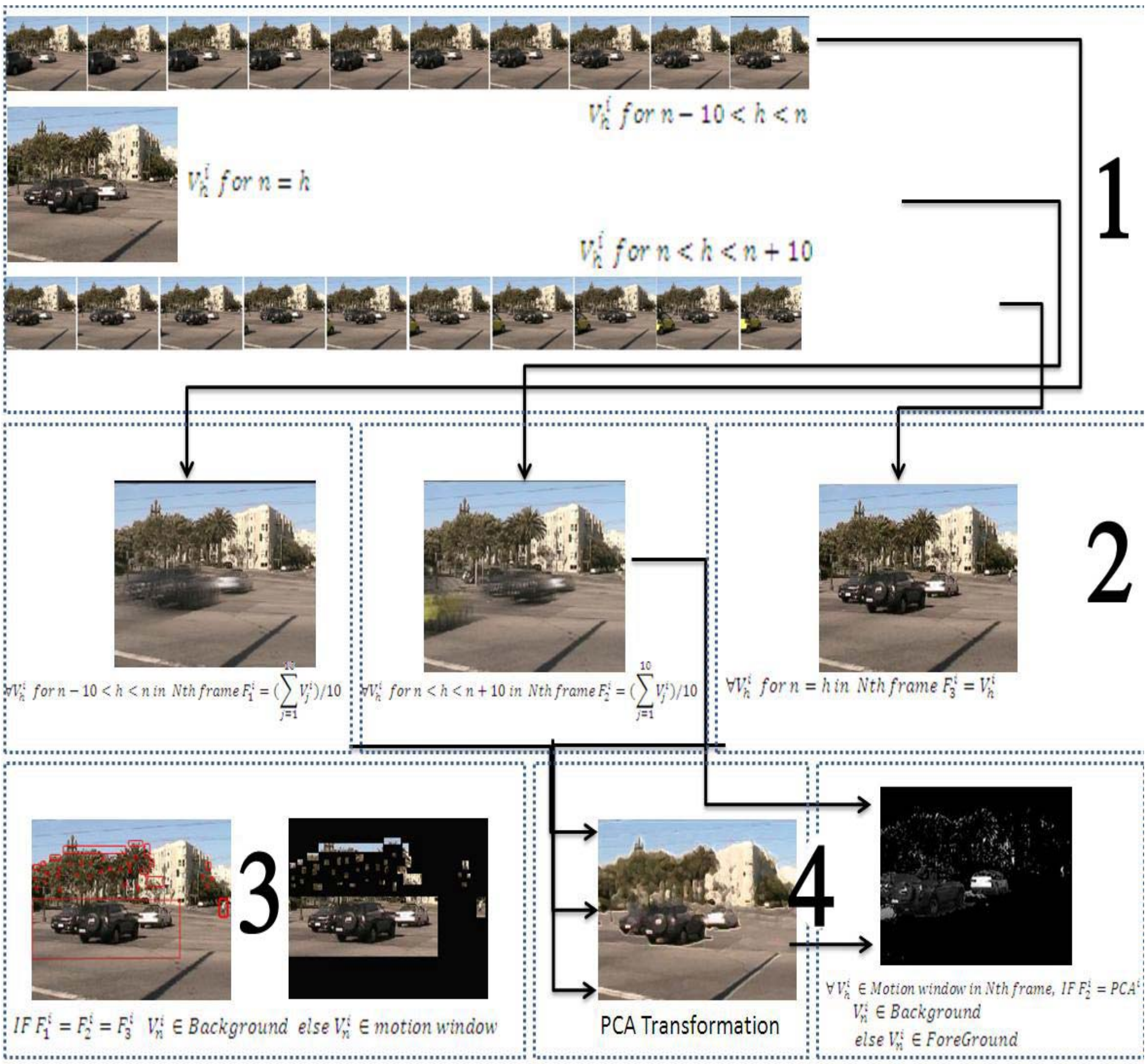

Fig. 2 Propoesd Background Subtraction Method

III. PROPOSED METHOD

Our method for background subtraction is a frame level method that use color information for comparison of pixels. This method is a part of a complete tracking system for surveillance applications and performed in realtime, approximately. Our results in comparison of some same methods, has better error rate.

Basic Idea

In fact, our proposed method is an extension of frame difference method. The basic idea is that by comparison of some neighborhood frames (20 for example), some points in this span, don't have any changes, but some point have changed. It is logical that motion objects are not in fixed points and thus we only must analyze changed pixels.

In this method, we first remove fixed pixels in a time span(2 second span) and then compare remaining pixels by PCA transformation method, for detect real motion pixels.

Finally, we remove some single motion pixels, because of noise reduction. In the last of process, only motion objects and their shadows remain.

This idea is shown in Fig. 2 basically.

Proposed Algorithm

Our proposed algorithm in detail is:

- For each frame, a neighborhood window considers suburbs of 'n' last and 'n' next frame of current frame.
- N last frame added together and make a composition frame. In the other hand these frame, place on top of each other. The same action is done for n next frame. In this phase, we have 3 frames: current frame, composition of n last frame and composition of n next frame. Equations of 2, 3, and 4 show these three composite frames (F1, F2, and F3).

$$\forall V_h^i \ for\ n-10 < h < n\ in\ Nth\ frame\ F_1^i = \frac{\sum_{j=1}^{10} V_j^i}{10} \quad (2)$$

$$\forall V_h^i \ for\ n < h < n+10\ in\ Nth\ frame\ F_2^i = \frac{\sum_{j=1}^{10} V_j^i}{10} \quad (3)$$

$$\forall V_h^i \ for\ n = h\ in\ Nth\ frame\ F_3^i = V_h^i \quad (4)$$

- These three matrices are compared point by point (we consider each 4 square pixel, is a point). Each point that have the same color feature in all 3 matrices, add as a background pixel. Because this point hasn't changed in 2n+1 frame (about 2 second here). Other points are added in motion windows set.

$$IF\ F_1^i = F_2^i = F_3^i \text{ then } V_n^i \in Backgroun$$
$$else\ V_n^i \in motion\ window \quad (5)$$

- In this phase, put each set of non-background points, in a rectangle window and name these, motion windows. We are sure that motion objects are in these motion windows and thus, we analyze only points of their. For this action, we calculate PCA transformation of 3 composite frames. The output of this transformation, show the average color of all pixel frames. Then we compare the color of each point in motion windows with the PCA average color. With a threshold, each point that not near the average color, remain in motion window and other points go to background. The remaining points, are the points of motion objects and make foreground set.

$$\forall V_h^i \in Motion\ window\ in\ Nth\ frame,$$
$$IF\ F_3^i = PCA^i \text{ then } V_n^i \in Background$$
$$else\ V_n^i \in ForeGround \quad (6)$$

IV. EXPERIMENTAL RESULTS

For evaluation of proposed algorithm, we test it on PETS2006 video dataset. The various aspects of results, are shown in this section

Error Rate of Background Subtraction

One of the efficient criterions in tracking researches is the error rate of each tracking phase that gets by comparison of results with "ground truth" of video sequences. Error rate for background subtraction calculate by this formula:

((Non-background pixels that detect as background + non-foreground pixels that detect as foreground)/all pixels of frame)*100%

The error rate for our method is about 3.9% in average that is less than other similar methods about 2% for this dataset. Fig.3 show the error rate for proposed method that implement in PETS2006 dataset.

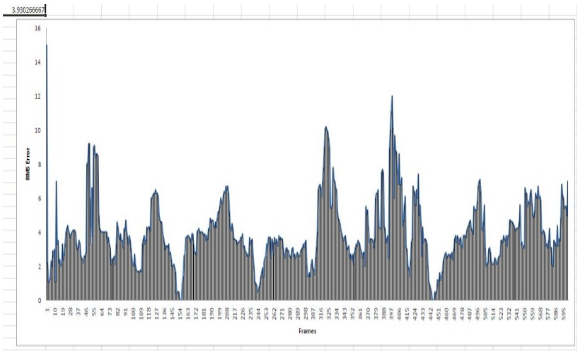

Fig. 3 Error rate for proposed method

The Best fps for Best Result

The fps (frame per second) of video sequences is a major factor in tracking process. We use different fps for videos and get more less error rate in 15 fps for PETS2006. Higher of fewer fps, have higher error rate in bg-subtraction. Fig.6 show error rates of proposed method for different fps on PETS2006.

Best Window Size (Neighborhood Frames)

The number of frames, next and previous of current frame, can effect on background subtraction. We test different size and finally, know that the best number is 10. We test 5 to 25 frame window and the least error rate was for 10.

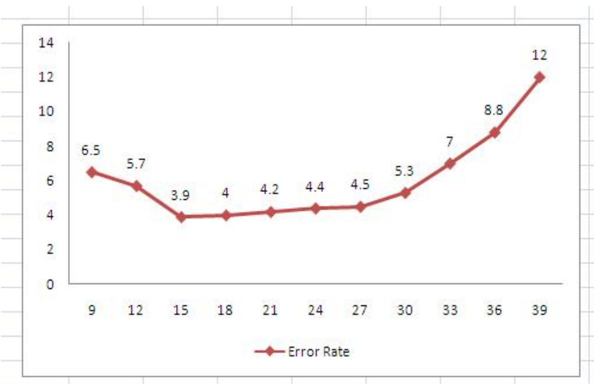

Fig. 4 Error rate for different FPSs

V. COMPARISION

The proposed method in this paper has some advantages and disadvantages too. The main advantages are:

- Simplicity: the algorithm is fully understandable and also it can implement easily
- Real- time: this method is a part of a complete tracking system that we are designing it for surveillance usages. Thus the time complexity is very important aspect. This method only late the time, for 10 frame that is about 0.7 second. (10/15 in real). In fact this method does realtime background subtraction.
- Efficiency: the error rate of method is less than similar available methods, about 2 to 2.5%.

Disadvantages of this method are:

- This method, can't detect motion non-foreground objects such as tree leafs, shadows, …. In the other hand, in this method, we detect only fixed pixels and remove them. Thus in a complete tracking system we need another process for detecting motion noises, specially motion shadows.(we present a novel motion shadow detection method in another paper)

VI. CONCOLUSION

Background subtraction, remove fixed regions and detect motion objects, is a main part of video object tracking process. Any of proposed methods, couldn't solve all problems in this field such as Quick illumination changes, shadows… in this paper we propose a novel method that simply detect motion objects by using temporal feature of video sequences and spatial feature of each pixel.

The experimental result show 2% to 3% improvement in motion detection for PETS2006 dataset. The main advantages of this method are simplicity and real-time implementation. This method can't remove motion non-foreground objects such as shadows and thus need to use a post-processing algorithm for such noises detection.

REFRENCES

[1] G. Casella and R. Berger. Statistical Inference. Duxbury, 2 edition, 2001.

[2] X. Gao, T.E. Boult, F. Coetzee, and V. Ramesh. “ Error analysis of background subtraction”. In Proceedings of International Conference on Computer Vision and Pattern Recognition, 2000.

[3] M. Greiffenhagen, V. Ramesh, and H. Nieman. “ The systematic design and analysis of a vision system: A case study in video surveillance”. In Proceedings of International Conference on Computer Vision and Pattern Recognition, 2001.

[4] Haritaoglu, D. Harwood, and L.S. Davis. “W4: Realtime surveillance of people and their activities”. IEEE Trans. on PAMI, 22(8):809–830, Aug 2000.

[5] Michael Harville. “ A framework for high-level feedback to adaptive per-pixel mixture of gaussian models”. In Proceedings of European Conference on Computer Vision, 2002.

[6] T. Horprasert, D. Harwood, and L. Davis. “A statistical approach for real time robust background subtraction and shadow detection ”. In IEEE Frame RateWorkshop, 1999.

[7] S. Jabri, Z. Duric, H. Wechsler, and A. Rosenfeld. “ Detection and location of people using adaptive fusion of color and edge information”. In Proceedings of International Conference on Pattern Recognition, 2000.

[8] R. Jain, D. Militzer, and H. Nagel. “Separating nonstationary from stationary scene components in a sequence of real world tv-images”. IJCAI, pages 612– 618, 1977.

[9] O. Javed and M. Shah. “Tracking and object classification for automated surveillance”. In The seventh European Conference on Computer Vision, 2002.

[10] L. Liyuan and L. Maylor. “Integrating intensity and texture differences for robust change detection”. IEEE Trans. on Image Processing, 11(2):105–112, Feb 2002.

[11] N. Ohta. “A statistical approach to background subtraction for surveillance systems ”. In International Conference on Computer Vision, 2001.

[12] N.M. Oliver, B. Rosario, and A.P. Pentland. “A bayesian computer vision system for modeling human interactions”. IEEE Trans. on PAMI, 22(8):831 –843, Aug 2000.

[13] A. Prati, R. Cucchiara, I. Mikic, and M.M. Trivedi. “Analysis and detection of shadows in video streams: a comparative evaluation”. In International Conference on Computer Vision and Pattern Recognition, 2001.

[14] C. Stauffer and W. E. L. Grimson. “Learning patterns of acitivty using real-time tracking”. IEEE Trans. On PAMI, 22(8):747–757, Aug 2000.

[15] K. Toyama, B. Brumitt J. Krumm, and B. Meyers. “Wallflower: Principles and practicle of background maintenance”. In Proceedings of International Conference on Computer Vision, 1999.

[16] C. Wren, A Azarbayejani, T. Darrel, and A. Pentland. “ Pfinder, real time tracking of the human body”. IEEE Trans. on PAMI, 19(7), Aug 1997.

[17] N. M. Oliver, B. Rosario, and A. P. Pentland, “A Bayesian Computer Vision System for Modeling Human Interactions,” IEEE Trans. on Patt. Anal. and Machine Intell., vol. 22, no. 8, pp. 831-843, 2000.