Graph-Based Segmentation of Moving Objects

Aliakbar Darabi , Amir Hossein Khalili and Shohreh Kasaei, Senior Member, IEEE

Image Processing Lab, Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

***Abstract* — This paper addresses the problem of segmentation of moving objects in image sequences. A graph-based algorithm is introduced to separate the moving layers from the background. By using this method, the problem of contributing spatial and temporal global properties into segmentation of moving objects is handled. In addition, the complexity of segmentation procedure is reduced so that it can be used in real-time applications. The superiority of the proposed algorithm is shown by evaluating the results obtained from our approach and the latest similar methodologies. This assessment is done in both subjective measurement and complexity aspects.**

***Index Terms* — Segmentation, moving objects, graph algorithm, moving layer.**

I. Introduction

Segmentation of moving objects in image sequences is an essential task in many vision applications [1]. As an exemplification, by having precise borders of moving objects, the accuracy of tracking algorithms can be improved efficiently. Moreover, after introduction of object-based video coding [2], the need for a robust and efficient algorithm for distinguishing objects in videos has arisen. The required resolution of segmentation process depends on the undergoing application (*i.e.,* having coarse boarders is sufficient for tracking applications while in compression applications accuracy of segments turns into a crucial criterion).

Because of existence of many artifacts (*i.e.,* lightening disturbance and capturing noise), distinguishing moving objects in image sequences is still an open problem. To alleviate these difficulties, using all temporal and spatial information is an inevitable task. Besides, as objects are mainly recognized by their general properties (*i.e.,* shape of object, texture of object and object motion), algorithms working on global properties are more efficient for segmentation purposes. Although segmentation methods that are based on global features are more desirable, they mostly have more computational cost and thus should be modified for real-time applications.

Previous works mostly are restrictively based on temporal features or spatial features of moving objects [3]. On the other hand, almost all of the fast algorithms employed for discriminating moving objects are using only local behavior of pixels [4,5]. Also, some rare existing approaches that use both temporal and spatial features that are based on global properties as well, suffer from a very high computational cost so that with current processing tools they are impractical for real-time applications.

The proposed algorithm improves the performance of the previously reported works by taking both motion and motion information into account when determining the moving objects. Furthermore, by using graph tools we have included the general behavior of moving objects in our segmentation method.

The rest of the paper is organized as follows. In Section II an overview on related previous works is given. In Section III different steps of the proposed algorithm are introduced in detail. The experimental results are shown in Section IV. Finally, Section V concludes the paper.

II. Related Works

Previous works done in segmentation of moving objects can be categorized in two main groups.

1) Algorithms that work on temporal properties of input signals; such as modeling the background by a mixture of Gaussians [6], techniques that differentiate between consequent frames [7], and optical flow-based methods [8].
2) Algorithms that use the spatial properties of pixels to track and segment objects; like algorithms using color histogram [9], contour-based tracking [10], and mesh-based tracking [11].

Although satisfactory results have been given, but in each group some shortcomings have been encountered. For instance, for spatial-based types, the major drawback is the change of features during time (like color alternation [10]). In temporal-based approaches, the key challenge is facing the noise occurred by environmental changes or by artifacts during video coding [7]. Consequently, many attempts have been made to combine both types together to improve the performance of currently available algorithms. For instance, in [12] the contours and vectors flows are blended together, and in [10] by mixing the estimation algorithms and edge details it has been claimed that acceptable results can be obtained.

The most powerful existing algorithms which use both types of information in moving object

segmentation are the graph-cut-based methods [13]. In this approach two types of energy are defined:

1) The foreground and background energy; which is obtained by probability of relevance of nodes to the foreground or background.
2) Spatial coherence energy; which is gained by spatial resemblance between nodes.

In graph-cut approach, in each frame of sequence a graph is constructed. Each node of graph corresponds to a pixel in frame. Two additional nodes that demonstrate the foreground and background classes are added to the graph; we call them source and sink in respect. Source and sink nodes are connected to each pixel nodes with weighted edges as shown in Fig. 1. Weight of the connecting edges between source/sink and pixel nodes, presents membership power of corresponding pixel to foreground/background segments. This membership power can be calculated from the difference between current frame and background model, at the corresponding pixel. Each pixel vertex is also linked to its neighbors with weighted edges showing how the neighbors are similar together. Graph-cut is a method that segments the pixel vertices to foreground and background classes so that best segmentation is achieved according to weight of edges. By proper setting the source/sink links and neighbor connections the problem of separation of foreground from background is reduced to the problem of finding the maximum flow in graphs which can be easily solved. There are many implementations of these types of algorithms available which vary by their definition of background and foreground energies [14,15]. The most important limitation of this strategy is its high computational cost that reduces its uses especially in applications requiring real-time processes [13].

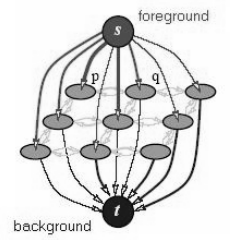

Fig 1. graph-cut structure for 3x3 image.

III. PROPOSED ALGORITHM

Our ultimate goal is reserving the same advantages that graph-cut-based methods have while having much lower computational cost. The proposed algorithm follows the approach presented in [13] but with some modification that makes it feasible for real-time applications. The algorithm first models the behavior of pixels in time and by this for each pixel it obtains the probability of belonging to the background regions. Next, by using the segments obtained from previous frame, the location of the segments are estimated in the current frame. This estimation leads us to find a model for foreground in the current frame. Then by the obtained foreground and background models gained from the previous steps and employing the graph tool, the current frame is segmented. Finally, the segments are merged to produce connected regions. Hence, the proposed algorithm can be divided into four main parts: background modeling, foreground modeling, segmentation, and merging. These are explained in detail in the following subsections.

A. Background modeling

There are many existing algorithms that can be employed in modeling of the background. A popular method for background modeling is using single Gaussian distribution as used in [6]. Improved versions have used more than one Gaussian distribution for background modeling [16]. Not completely static background can be modeled better with a mixture of Gaussians distribution. Many methods have proposed the model parameter estimation approaches. In [17], the number of mixture components is constantly adopted for each pixel. Non-parametric approaches for dealing with limitation of parametric models such as Gaussian assumption for pixels intensity is proposed in [18]. Background modeling is performed using edge features in some methods. For having the best results with a low complexity, we adopt the approach introduced in [16]. As such, the relation that shows the probability of being in background in each turns to:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} \times \eta(X_t, \mu_{i,t}, \Sigma_{i,t}), \qquad (1)$$

where $\omega_{i,t}$ is the weight of the i^{th} Gaussian in the mixture at time t where $\Sigma\omega = 1$, $\mu_{i,t}$ denotes the mean, $\Sigma_{i,t}$ is the covariance matrix, and function η is the Gaussian probability density function. For each pixel, we select the $\mu_{i,t}$ that corresponds to i^{th} Gaussian mixture with maximum weight, $\omega_{i,t}$, as color of background at time t.

B. Foreground modeling

The algorithm performs iteratively and at each iteration it uses the previously obtained segments as the input to find the new segments. In this approach, motions of previous segments are estimated. Next, previous segments are moved by estimated motions to represent preceding segments in current frame. Foreground color of each pixel is considered as the

color of the moved previous segments in that pixel. Consequently, for having foreground model, an algorithm is needed to localize the previous segments in the current frame.

For estimation of motion of previous segments, there exist different kinds of algorithms that operate on different data types. Estimation-based algorithms (like Kalman filters [19]) use previous object locations to estimate the current object state. Recently, new Kalman-based algorithms have been suggested that improve the results of previous ones but with higher computational cost [20]. Another option would be the algorithms using color histograms (like mean-shift tracking algorithm [9]). For having robust tracking with less computations we used the mean-shift algorithm but with some changes. For instance the computation of histogram is performed only on segmented parts in the current frame and the searching correspondence process is performed only on points ensured to be in foreground regions (points having large distance from background).

For modeling foreground, histogram of segment in previous frame is taken. Next, current frame is searched by mean-shift algorithm to find most similar box to prior segment. This similarity is based on histogram. It is assumed that the motion of each segment is the vector connecting box surrounding segment in prior frame to box announced by mean-shift algorithm in current frame (see Fig. 2). After estimation of motion, previous segment is moved by this motion and color of each pixel in moved segment is considered as foreground color in current frame.

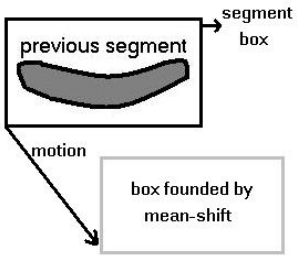

Fig. 2. Motion estimation strategy.

C. Segmentation

At this Step, we have joined the pixels that have similar features. For each pixel two features are selected: 1) distance from background 2) distance from foreground. First feature is gained based on information obtained from the background model and the second one is inferred from foreground estimation as explained in previous step. The desired algorithm must have low complexity and strong power of discrimination. The choices in this regard are quite vast. We adopt the approach introduced in [21] because of three major reasons: 1) in that method it has been tried to act upon global behavior of pixels. 2) It has low computational complexity and can be applicable in nearly linear time. (This property makes the proposed algorithm superior over similar methodologies like graph-cuts.) 3) By using graphs and selecting vectors representing adjacency of two pixels, the information of locality is included. This feature makes the algorithm superior to algorithms that segment upon only color or texture information; like algorithms using K-means clustering on color features [22].

For adapting [21] into segmentation of moving objects we employed a function for combining two types of features one showing the distance from background and the other one showing the distance from foreground. For each pixel, by combination of these two types in the way depicted in Fig. 3, proper features for segmentation are obtained. We have found that a linear combination of these two types make a proper function of blending. As the result, for each pixel its feature relation transfers to

$$F(p) = \alpha\left|d(p, p_b)\right| + (1-\alpha)\left|d(p, p_f)\right|, \quad (2)$$

where $d(p, p_b)$ denotes the distance of color of pixel in the current frame from color of background announced in the background model. $d(p, p_f)$ denotes the distance of color of pixel in the current frame from color of foreground announced in the foreground model.

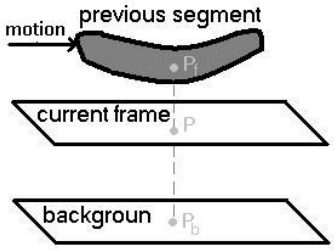

Fig. 3. Strategy of mixing foreground properties by background properties.

After assigning features to each pixel, it is time to segment the pixels based on these features. Like [21] we build a graph having a node for each pixel. Then each two nodes corresponding to two adjacent pixels are connected by a weighted edge. The weight of this edge equals to Euclidian distance of features of those adjacent pixels. After the graph construction step, for each node one set is made and these sets are connected by edges connecting member nodes (see Fig. 4). Next, the algorithm iterates on edges and for each edge that connects two different sets we join those sets if the weight of that edge is below the average weights of edges in both sets. By using the average of weights in each set, it is asserted that general behavior of pixels is reflected in segmentation process. The results of

applying the algorithm on still images can be inferred as a proof for this claiming.

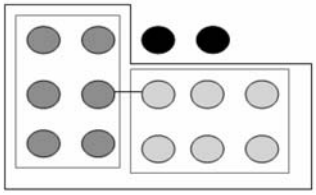

Fig. 4. Graph structure of proposed algorithm. Circles denote graph nodes and rectangles show the sets. The merging step is done by edges connecting two sets together as shown by a line.

As suggested in [21], a disjoint set data structure is used to reduce the computational cost of the algorithm. It is proved that the algorithm complexity is $O(n \log n)$, where n is the number of graph nodes and since it is nearly linear of number of pixels, for videos having regular sizes (lower than 640x480) it can be computed in real-time.

D. Merging

Segments resulted from the previous stage might need post-processing to become more meaningful. Consequently, an algorithm for labeling each segment as foreground or background is required. In this part, segments must be merged together to produce foreground and background regions. Segment combination can be taken upon different criteria such as motion or distance from background/foreground models.

For labeling the segments, at first each pixel is assigned to be of type of foreground or background. For deciding on type of each pixel, its distance from background model and foreground model is computed. If the pixel color is more similar to the foreground it is assigned as a foreground pixel and otherwise it is considered as a background pixel. Finally, for each segment the number of its foreground pixels and background pixels are counted and if the number of foreground pixels is greater than background ones that segment is announced as foreground segment and other wise it is decided to be background segment.

IV. Experimental Results

In this section the effectiveness of the proposed algorithm is judged. Evaluation is done by comparison of results of proposed algorithm and algorithm presented in [13]. This judgment is based on both subjective measurements and time complexity aspects. We use an indoor video taken from class environment and an outdoor video taken from University location as inputs of both algorithms. Indoor video is captured with 30 frames per second that each frame size is: 640x480. Outdoor video is captured with 30 frames per second and each frame size is: 768x520. Fig. 5 compares result of proposed algorithm and result of graph-cut-based algorithm introduced in [13] on indoor video.

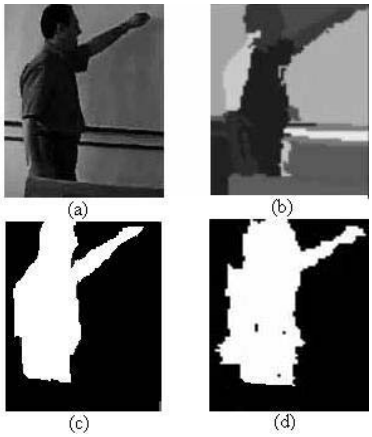

Fig. 5. (a) Original indoor frame, (b) Segmented image produced by our algorithm. (c) Image obtained by merging segments as proposed in D section. (d) Segmented Image obtained by algorithm [13].

It is obvious that results obtained by proposed algorithm are better than results generated by [13]. Superiority of our algorithm is more noticeable when processing outdoor videos as illustrated in Fig. 6. It is apparent that results in [13] are much coarser than ours. It can be observed that small background regions surrender by foreground regions are vanished when using [13].

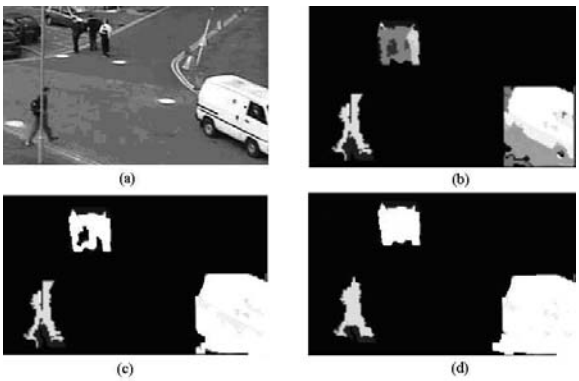

Fig. 6. (a) Original indoor frame (b) Segmented image produced by algorithm. (c) Image obtained by merging segments as proposed in algorithm. (d) Segmented Image obtained by applying algorithm [13].

As it was noticed, our algorithm has lower time complexity than algorithm introduced in [13]. We implement proposed algorithm and algorithm introduced in [13] both with standard C++. Both are run on 3 Giga hertz Pentium Dual Core processor with Windows XP

operating system. Table I compares time taken for each algorithm to process each frame of input video with different sizes.

TABLE I
Comparison of time complexity of proposed algorithm and algorithm introduced in [13].

Frame Size	Proposed Algorithm Cost (ms)	Algorithm in [13] Cost (ms)
160x120	26	98
320x240	32	265
640x480	47	964

V. Conclusion

In this paper an algorithm for segmentation of moving objects is introduced. In suggested approach both temporal and spatial properties of pixels are included. In addition, by using graph it has been tried to segment moving objects by considering general properties of pixels. The obtained results show much improvement in both complexity and subjective measurements in contrast with latest related works. The time needed to process each frame is reduced to 0.04 time needed in graph-cut-based algorithm while having equivalent or better subjective results.

Acknowledgement

This work was supported by a grant from ITRC.

References

[1] Faloutsos C., Barber R., Flickner M., Hafner J., Niblack W., Petkovic D., and Equitz W., "Efficient and Effective Querying by Image Content," Intelligent Information Systems, vol. 3, no. 1, pp. 231-262, 1994.

[2] G. Ahanger and T. Little, "A survey of technologies for parsing and indexing digital video." Journal of Visual Communication and Image Representation,7(1):28–43, March 1996.

[3] P. Salembier, "Region based representation of images and videos: segmentation tools for multimedia services", IEEE Trans. on circuits and systems for video technology, Vol. 9, No. 8, pp. 1147-1167, December 1999

[4] H. Zhang, A. Kankanhalli, and S. Smoliar. Automatic partitioning of fullmotion video. ACM/Springer Multimedia Systems, 1(1):10–28, 2001.

[5] F. Leymarie and M. D. Levine, "Tracking deformable objects in the plane using an active contour model," IEEE Trans. Pattern Anal. Machine Intell., vol. 15, pp. 617–634, 2003.

[6] D. Koller, K. Daniilidis, and H.H. Nagel, "Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes," International Journal of Computer Vision, vol. 10, pp. 257-281, 1993.

[7] Y. Kameda and M. Minoh, "A Human Motion Estimation Method Using 3-Successive Video Frames, " in Proceedings of International Conf. on Virtual Systems and Multimedia, 1996, pp. 135-140

[8] Barron J.L., Fleet D.J., and Beauchemin S.S., "Performance of optical flow techniques," International Journal Computer Vision, volume 12, no. 1, pp. 43-77, 1994.

[9] Fieguth P., "Color-Based Tracking of Heads and Other Mobile Objects at Video Frame Rates," Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition.

[10] F. Leymarie and M. D. Levine,"Tracking Deformable Objects with Unscented Kalman Filtering and Geometric Active Contours", Proceedings of the 2006 American Control Conference, 2006

[11] I. Kompatsiaris, D. Tzovaras, and M. G. Strintzis. "3-D model-based segmentation of videoconference image sequences." IEEE Trans. on Circuit and Systems for Video Technology, 8(5):547–562, September 1998.

[12] P. Doklah, R. Raffih, "Contour-Based Object Tracking with Gradient-Based Contour Attraction Field", ICASSP 2004.

[13].R.Howe, and A.Deschmaps."Better Foreground Segmentation Through Graph Cut", eprint arXiv:cs/0401017 2004.

[14] BLAKE, A., ROTHER, C., BROWN, M., PEREZ, P., AND TORR, P."Interactive image segmentation using an adaptive gmmrf model". European Conf. Computer Vision, 2004.

[15]T.Yu,C. Zhang, M. Cohen, Y. Rui, and Y. Wu," Monocular Video Foreground/Background Segmentation by Tracking Spatial-color Gaussian Mixture Model",IEEE 2006

[16] N. Friedman and S. Russell, "Image Segmentation in Video Sequences: A Probabilistic Approach," Proc. Conf. Uncertainty in Artificial Intelligence, pp. 175–181, 1997.

[17] Z. Zivkovic, "Improved Adaptive Gaussian Mixture Model for Background Subtraction," Proc. Int'l Conf. Pattern Recognition, vol. 2, pp. 28–31, 2004.

[18] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance," Proc. IEEE, vol. 90, no. 7, pp. 1151–1163, 2002.

[19] Y. Satoh, T. Okatani, "A Color-based Tracking by Kalman Particle Filter", IEEE, International Conf. on Pattern ecognition, 2004.

[20] S. Julier and J. Uhlmann ,"Unscented filtering and nonlinear estimation", Proceedings of the IEEE, vol. 92, no. 3, pp. 401–420, 2004.

[21] P. F. Felzenswal, D. T. Hutencheler, "Efficient Graph-Based Image Segmentation", International Journal Computer Vision, volume 32, no. 14, 2004.

[22] A. Abadpour and S. Kasaei. Fast Color FPCA-Based Clustering for Image Segmentation. *Elsevier Science, Image & Vision Computing* (under 2nd review), 2004