2011 Seventh International Conference on Natural Computation

Spatio-temporal Video Object Segmentation Using Moving Detection and Graph Cut Methods

Dingming Liu
Research Institute of Computer Science and Technology
Ningbo University, Ningbo, China, 315211

Jieyu Zhao
Research Institute of Computer Science and Technology
Ningbo University, Ningbo, China, 315211

Abstract—**Segmentation of video foreground objects from background has many important applications, such as human computer interaction, video compression, multimedia content editing and manipulation. From a single video sequence with a moving foreground object and stationary background, this paper propose a novel algorithm to extract video object using graph cut and moving detection methods. The key idea in our paper is to obtain the moving object region which can be set as the possibility foreground, and the other region set as background, then this prior can be used by means of graph cut, video segmentation is then transformed to static image segmentation which can be achieved by binary min-cut.**

Keywords-video segmentation; moving detection; frame difference;Gibbs Random Field; graph cut

I. INTRODUCTION

Foreground extraction from image sequences has long been an active area of research. Though many segmentation algorithms have been proposed for special cases [1, 2, 3, 4, 5], there are still no algorithms can solve all the segmentation problems, which let to the fact that it has long been a topic of research in computer vision. Video is just image sequences that have a strict order in times and has a strong correlation between adjacent frames. Video segmentation is split every frame into lots of consistent region, and the purpose is usually to extract the interesting region which is called foreground and the other region was defined as background.

For a long time research by researchers, many video segmentation algorithms have been proposed [6, 7, 8, 9, 10, 11]. All the algorithms can be classified into two types: segmentation based on intra-frame and segmentation based on neighboring frames. Segmentation based on intra-frame takes the traditional image segmentation techniques: selecting some key frames and segment it into many consistency region with the features of image such as color, gray scale, edge, texture; then tracking the object in the frames [6, 9].The most widely used algorithms is based on contour tracking. The morphological watershed algorithm has been widely used for it can quickly obtain the contour of the object [3, 6]. However, segmentation based on intra-frame usually leads to excessive segmentation because it does not use inter-frame information and it is sensitive to noise. Segmentation based on neighboring frames is not only used the space information of each frame but also used the moving information that obtain from neighboring frames. The most efficient approach is background subtraction [4], but it based on a known background image, which limits its application. And the most common approach is frame difference, this approach is simple and effective, but it is usually not accurate enough.

II. OVERVIEW

In this paper, we introduce a new approach based on moving detection and graph cut, a high quality foreground layer extraction algorithm. Our approach will use both the intra-frame and neighboring frame information to segment the video which can be called spatio-temporal video segmentation [10]. Time-domain segmentation use motion information between continuous frames to determine which pixels belong to moving object. It can roughly separate the image into foreground and background. Space-domain segmentation uses color feature to extract object, which could have a more accurate segmentation. In this paper, Firstly, we get the moving information of the object in each frame with frame difference, and detect the object region which can be roughly set as the possibility foreground T_U , and the other region set as background T_B. In this way, we can get a trimap T with known background. Then, we can extract the object in each frame with the graph cut algorithm [12, 13, 14, 15]. The flow of our system is shown in figure 1.

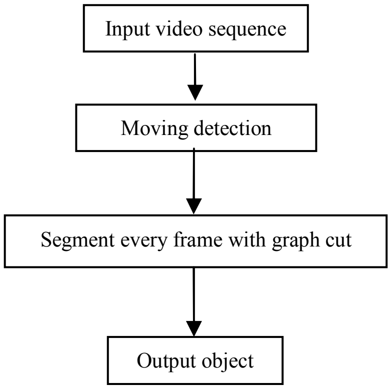

Figure 1. The flow of our system.

III. MOVING DETECTION

Considering the background is relatively static in most of the video and the foreground is always moving region, we can quickly and effectively get the changed region by frame difference. In fact, the border getting from frame difference is part of the object contour. In this way, we can detect the region

978-1-4244-9953-3/11/$26.00 ©2011 IEEE

which contains moving object in each frame and set it as the possibility foreground. However, the object region obtaining from two continuous frames is not accurate enough because it contains the exposure background, w is shown in the figure 2.a. In this paper, we will do frame difference in three continuous frames. It can effectively eliminate the effect of background change and can get more accurate contour information of the moving object, it is illustrated in figure 2.b. Assuming the ellipse is the moving object in video and using f_k means the kth frame. The continuous three frames indicate by f_{k-1}, f_k and f_{k+1}.

The frame difference between current frame and the previous frame:

$$FD_{pre}(x,y,k) = |I(x,y,k) - I(x,y,k-1)| \quad （1）$$

The frame difference between current frame and the next frame:

$$FD_{after}(x,y,k) = |I(x,y,k+1) - I(x,y,k)| \quad （2）$$

$I(x,y,k-1), I(x,y,k)$ and $I(x,y,k+1)$ is the data at (x,y) of three continuous frames. In order to remove noise, we will set a reasonable threshold TH to get binary image. The threshold should not be too small, or the information of frame difference will contain a lot of noise. But it should not be too high, or we could not get enough information from frame difference. In this paper, the TH is 0.1.

$$FDM(x,y,k) = \begin{cases} 1 & FD(x,y,k) > TH \\ 0 & FD(x,y,k) < TH \end{cases} \quad （3）$$

The FDM is a frame difference mask. In this way, we can get the binary image FDM_{pre} and FDM_{after} which denote frame difference of current frame with the previous frame and next frame respectively. In the figure 2.a, the shadow means changed region, but it not all belong to object, some of them caused by background exposure.

The public regions usually are parts of the moving object which can get from frame difference in three continuous frames：

$$C_f(x,y,k) = FDM_{pre}(x,y,k) \cap FDM_{after}(x,y,k) \quad (4)$$

The possibility foreground region was determined by the largest enclosing rectangle of the public parts which is shown by the green rectangle in figure 2.b. In the figure 3, the second row shows the frame difference mask and the third row shows the possibility foreground with red rectangle which should be segmented accurately. Taking into account the situation of moving object suddenly still, the moving detection will be useless to get the possibility foreground region. In order to solve this problem, a more suitable possibility foreground rectangle will be obtain by compare with previous frame with a typically threshold.

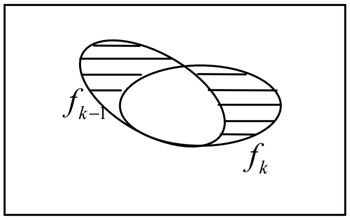

Figure 2.a Frame difference detemine the moving region

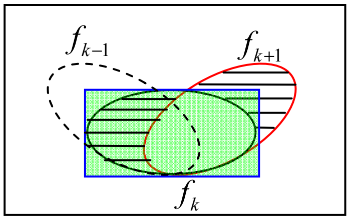

Figure 2.b Three frames difference obtain the object region

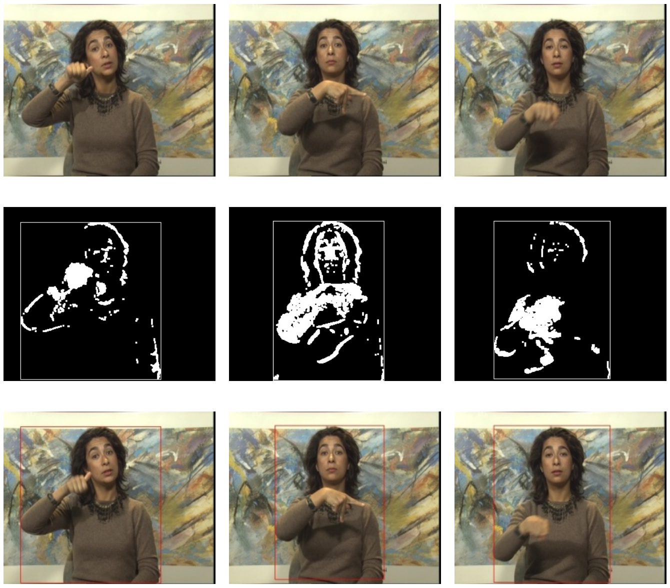

Figure 3. The first row is the original image, the second row is the frame difference mask and the third row determined the background and the possiblity foreground with red rectangle.

IV. Segmentation by Graph Cut

In section 2, we have introduced how to roughly obtain the moving region. It can be set as possibility foreground which will segment to extract object, and the other region of the frame can be set as background. So the video segmentation converted into a static image segmentation that have marked background region. It can be solved by graph cut.

An image segmentation problem can be posed as a labeling problem. The image is an array $Z = (z_1, \ldots, z_{n1}, \ldots, z_N)$, indexed by the single index n. The segmentation of the image is expressed as an array of "opacity" values $\alpha = (\alpha_1, \ldots, \alpha_n, \ldots, \alpha_N)$ at each pixel. Generally $0 \leq \alpha_n \leq 1$, but for hard segmentation $\alpha_n \in \{0,1\}$, with 0 for background and 1 for foreground. The foreground is the part we need to preserve and the background is the part we need to remove.

According to the property of the Gibbs random field, the image segmentation can be obtained by minimizing a Gibbs energy [16, 17, 18, 19]:

$$E(\underline{\alpha}, K, \underline{\theta}, Z) = U(\underline{\alpha}, K, \underline{\theta}, Z) + V(\underline{\alpha}, Z) \qquad (5)$$

where the data term U evaluates the fit of the opacity distribution α to the data z, and the smoothness term V is the penalty term when adjacent pixels are assigned with different labels.

The image is taken to consist of pixels $\mathrm{z_n}$ in RGB colour space. Each GMM, one for the background and one for the foreground, is taken to be a full-covariance Gaussian mixture with K components (typically K = 5). In order to deal with the GMM tractably, in the optimization framework, an additional vector $\mathrm{K} = \{\mathrm{k_1}, \ldots, \mathrm{k_n}, \ldots, \mathrm{k_N}\}$ is introduced, with $\mathrm{k_n} \in \{1, \ldots, \mathrm{K}\}$, assigning, to each pixel, a unique GMM component, one component either from the background or the foreground model, according as $\alpha_n = 0\ or\ 1$. The data term U is defined as：

$$U(\underline{\alpha}, K, \underline{\theta}, Z) = \sum_n D(\alpha_n, k_n, \underline{\theta}, z_n) \qquad (6)$$

where k is the GMM component variables, and $D(\alpha_n, k_n, \underline{\theta}, z_n) = -logp(z_n | \alpha_n, k_n, \underline{\theta}) - log\pi(\alpha_n, k_n)$, $p(\cdot)$ is a Gaussian probability distribution, and $\pi(\cdot)$ is a mixture weighting coefficient.

The parameters of the model can denoted as :

$$\underline{\theta} = \left\{\pi(\alpha, k), \mu(\alpha, k), \sum(\alpha, k), \alpha = 0,1, k = 1, \ldots, K\right\} \qquad (7)$$

where the weights π, means μ and covariances $\sum$of the 2K Gaussian components for the background and foreground distributions.

We used the smoothness term V to represent the energy due to the gradient along the object boundary. It can be defined as a function of the color gradient between neighborhood pixels.

$$V(\underline{\alpha}, Z) = \lambda \sum_{(m,n)\epsilon N} |\alpha_m - \alpha_n|\, exp(-\beta(z_m - z_n)^2) \qquad (8)$$

where the parameter λ balances the influences of the data term and smooth term, $|\alpha_m - \alpha_n|$denotes the smoothness term is only have penalty to adjacent pixels are assigned with different labels. $(z_m - z_n)$ is the L2-Norm of the color difference of neighboring pixels n and m, where N is the set of pairs of neighboring pixels. In this paper, we propose 8-connected neighboring structure. β is a robust parameter that weights the color contrast, and can be set to

$$\beta = (2\langle(z_m - z_n)^2\rangle)^{-1} \qquad (9)$$

where $\langle\cdot\rangle$ is the expectation operator. Now that the energy model is fully defined, the segmentation can be estimated as a global minimum:

$$\underline{\alpha} = arg\min_{\underline{\alpha}} E(\underline{\alpha}, K, \underline{\theta}, Z) \qquad (10)$$

Minimization is done using a standard minimum cut algorithm [20].

V. Proposed Algorithm

The entire algorithmic flowchart can be summarized as follows:

1. Get the moving region of every frame with frame difference, and define the maximum enclosing rectangle of the region.
2. The region out of the rectangle was defined as background T_B, and the region of the rectangle was defined as uncertain T_U. Initialize $\alpha_n = 0$ for $\in T_B$, and $\alpha_n = 1$ for $n \in T_U$.
3. Segment the trimap T by graph cut:
   a) Assign GMM components to pixels which belong to rectangular region：
   $$k_n := \arg\min_{k_n} D_n(\alpha_n, k_n, \theta, z_n)$$
   b) Learn GMM parameters from data Z：
   $$\underline{\theta} := \arg\min_{\underline{\theta}} U(\underline{\alpha}, K, \underline{\theta}, \mathrm{Z})$$
   c) Estimate segmentation: use min cut to solve:
   $$\min_{\{\alpha_n : n \in T_U\}} \min_k E(\underline{\alpha}, k, \underline{\theta}, z)$$
   d) Repeat from step 1), until convergence.
4. Output the segmentation result of current frame and repeat the first step to segment the next frame until have segmented the whole video.

VI. Experimental Results

Our experiments were performed on a desktop PC with Pentium(R) Dual-core CPU E5400, 2.70GHz and 2GB memory. Our system was programmed with MS Visual 2008 based on opencv2.1. We experiment the image sequences in 352*288 and 640*480 at 30 fps. The result is shown in figure 4. The figure 4.a is the test image sequence of 352*288 at 30 fps which is the standard test video of gesture recognition. The upper row shows the 51st, the 81st and the 111st original image in input video, the lower row shows our segmentation result. It is clearly shown that our approach has a good segmentation result and can effectively segment the moving person and gesture. The figure 4.b is the test image sequence of 640*480 at 30 fps which was record in laboratory. The upper row shows the 75th, the 100th and the 125th original image in input video. The lower row shows our segmentation result. As can be seen in the figure, our algorithm does a good job of extracting of the moving object from the complex static background.

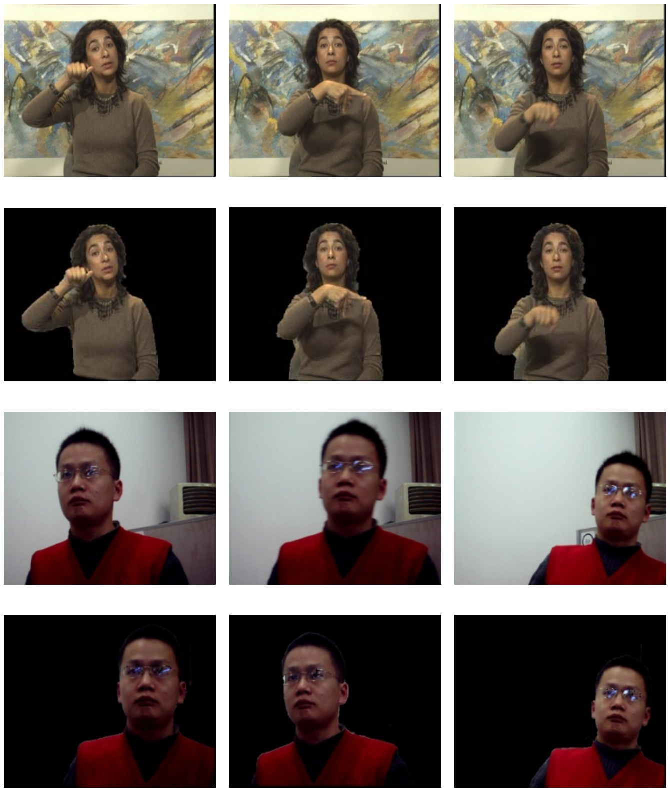

Figure 4. The upper row shows input video and the lower row shows our segmentation result

VII. Conclusion and Future Work

In this paper, we have proposed a high quality and effectively moving object extraction from the complex static background, which based on graph cut combines improved frame difference. First, we obtain the moving region by frame difference. The region was set as the possibility foreground and the other region was set as background. Then we used the graph cut to segment each frame to extract the moving object. Our system is not only effective use frame coherence of time domain, but also effective use feature of space domain.

The current system still has some limitation. First, when the foreground and background colors are very similar, high quality segmentation usually is hard to be obtained with our current algorithm. Enforcing more temporal coherence of the foreground boundary may improve the result to a certain extent. Second, in the current system, assuming the background is stationary, it will limit its application in practice.

Our future work will focus on constructing more accurate and fast algorithm to extract foreground from moving background.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (Grant No. 60672071) and the Natural Science Foundation of Zhejiang (Grant No. D1080807).

References

[1] J. Niebles, B. Han, F. Li, Extracting moving people from internet videos[C]. In: Proc. of ECCV2008, pp. 527-540.

[2] J. Wang, P. Bhat, R. Colburn, M. Agrawala, M. Cohen, Interactive Video Cutout[C]. In: ACM Transactions on Graphics. Vol. 24(3), pp. 585-594, 2005.

[3] Y. Li, J. Sun, H. Shum, Video Object Cut and Paste[C]. In: ACM Transactions on Graphics. Vol. 24(3), pp. 595-600, 2005.

[4] J. Sun, W. Zhang, X. Tang, H. Shum, Background Cut[C]. In: Proc. of ECCV2006,pp.628-641.

[5] W. Qiong, P. Boulanger, W. Bischof, Robust Real-Time Bi-Layer Video Segmentation Using Infrared Video[C]. In Proceedings of the 5th Canadian Conference on Computer and Robot Vision. 2008, pp. 87-94.

[6] D. Wang, Unsupervised video segmentation based on watersheds and temporal tracking[J]. IEEE Transactions on Circuits and Systems for Video Technology. Vol. 8 (5), pp. 539-546, 2002.

[7] E. Carmona, J. Martinez-Cantos, J. Mira, A new video segmentation method of moving objects based on blob-level knowledge[J]. In: Proc. of Pattern Recognition Letters. Vol. 29 (3), pp. 272-285, 2008.

[8] S. Natarajan, An Efficient Video Segmentation Algorithm with Real time Adaptive Threshold Technique[J]. In: International Journal of Signal Processing, Image Processing and Pattern Recognition. Vol. 2(4), pp. 154-168, 2009.

[9] J. Niebles, B. Han, F. Li, Efficient Extraction of Human Motion Volumes by Tracking[C]. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2010, 655-662

[10] T.Nagahashi, H.Fujiyoshi, T. Kanade, Video Segmentation Using Iterated Graph Cuts Based on Spatio-temporal Volumes[C]. In: Proc. of ACCV2009, pp. 934-942.

[11] A. Criminisi, G. Cross, A. Blake, V. Kolmogorov, Bilayer Segmentation of Live Video[C]. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2006, pp. 53-60.

[12] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimi - zation via graph cuts[J]. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 23(11), pp. 1222–1239, 2001

[13] Y. Boykov, G. Funka-Lea, Graph Cuts and Efficient N-D Image Segmentation[J]. International Journal of Computer Vision. Vol. 70(2), pp.109–131,2006.

[14] Y. Li, J. Sun, C. Tang, H. Shum. Lazy Snapping[C]. In:ACM Transactions on Graphics. Vol.23(3), pp. 303-308. 2004.

[15] V. Lempitsky, P. Kohli, C. Rother, T. Sharp. Image Segmentation with A Bounding Box Prior[C]. In: Proc. of the IEEE International Conference on Computer Vision. 2009, pp. 277-284.

[16] C. Rother, V. Kolmogorov, A. Blake. GrabCut-Interactive Foreground Extraction using Iterated Graph Cuts[C]. In: ACM Transactions on Graphics. Vol. 23(3), pp. 309-314, 2004.

[17] S. Vicente, V. Kolmogorov, C. Rother, Graph cut based image segmentation with connectivity priors[C]. In: 26th IEEE Conference on Computer Vision and Pattern Recognition . 2008, pp. 1021-1029.

[18] V. Lempitsky, P. Kohli, C, Rother, Toby Sharp. Image Segmentation with A Bounding Box Prior[C]. In: Proceedings of the IEEE International Conference on Computer Vision. 2009, pp. 277-284.

[19] C. Couprie, L. Grady, L. Najman. Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest[C]. In: Proceedings of the IEEE International Conference on Computer Vision. 2009, pp. 731-738.

[20] Y. Boykov, V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision[J] In: IEEE Transactions on PAMI. Vol. 26(9) , pp. 1124-113, 2004.