Key Object-based Static Video Summarization *

Zhiqiang Tian, Jianru Xue, Xuguang Lan, Ce Li, Nanning Zheng
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
Xi'an, Shaanxi, P.R. China, 710049
{zqtian,jrxue,xglan,celi,nnzheng}@aiar.xjtu.edu.cn

ABSTRACT

In this paper, we present a system for object-based video summarization facilitated by an efficient video object segmentation system. We eliminate the redundancy not only from spatial and temporal domain, but also from content domain. First, we detect shot boundaries and extract video objects by a 3D graph-based algorithm. Once the objects are obtained, the shape of the objects need to be represented. The key objects are extracted in a global manner by K-means clustering of shapes. Experimental results on the proposed object-based scheme combined with efficient video object segmentation show desirable summarization.

Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*Video analysis*

General Terms

Algorithms, Experimentation

Keywords

Video object summarization, graph cuts, video object segmentation, key actions

1. INTRODUCTION

To date, a great demand for video data in applications due to the significant improvement in the processing technology and availability of large storage systems. It is necessary and important to allow the computer to automatically manage the parts of interest form videos. In recent years, video summary for efficient browsing, retrieval, and storage is becoming more and more popular. It is a video content service to make videos easier or faster to understand.

*Area chair: Pal Halvorsen

MM'11, November 28–December 1, 2011, Scottsdale, Arizona, USA.

Generally speaking, a video summary is defined as sequence of still or short dynamic video presenting the content of the video in compact and brief manner. The target summary is rapidly provided with concise information about original content. From the viewpoint of presentation style, the existing methods can be classified into two categories, i.e., static video summarization and dynamic video skimming. Static video summaries are consist of several keyframes, while dynamic video summaries are composed of a set of thumbnail movies (with or without audio) extracted from the original video, our method belongs to the first category. Li [6], Truong [10] and Money [7] have present comprehensive surveys of video abstract.

Since the proposed approach in this paper falls into the static video summarization, we mainly focus the review on this category hereinafter. There has been a rich literature on static video summarization [1, 3], These existing methods provide effective solutions to summary video into a concise representation. However, they focus on low frame-level processing. Object-based or content-based video summarization has been rarely addressed, though content-based videos are widely used in surveillance systems, retrieval systems and second generation video coding systems.

Erol and Kossentini [2] present an automatic key video object plane selection in the MPEG-4 compressed domain using the Hamming and the Hausdorff distance measures. The shape of the objects are approximated using the shape coding modes that can be determined from MPEG-4 bitstream. Nevertheless, most object-based video systems perform processing in the uncompressed domain. Our paper addresses the object-based video summarization in the uncompressed domain which is independent on the video coding standards.

Similarly, Kim [5] presents an integrated scheme for object-based video abstraction. Their object-based keyframes are extracted in a sequential manner through the sequence and first frame in a shot is always chosen as a keyframe. Unlike our system, their work focuses more on static background videos. Our work in contrast focuses on summarizing long films with an global manner, and we allow for both static and moving background. For all frames in one shot are clustered into several segments based on shape similarity without considering the order of frames position. The frames that are closest to the center of each segment are considered as keyframes. This make our scheme more compact than Kim's, and will not yield analogous keyframes, especially for long shots.

In this paper, we proposed key object-based video summarization (KOBVS) scheme, combined with an efficient video

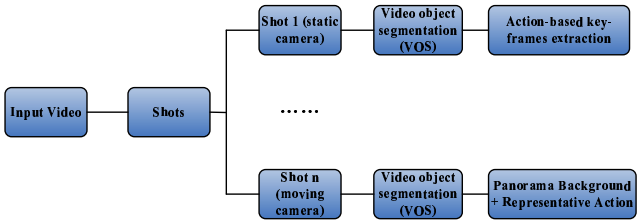

Figure 1: System overview

objects segmentation system, generates desirable summarization of the videos containing semantic representative actions for static background and moving background. For the shot with static background, keyframes are presented based on key objects extraction using 3D graph-based segmentation and shapes comparison. And the shot with moving background is decomposed by 3D graph-based segmentation, and depicted as a combination of background panorama and representative foreground objects. Our proposed framework of video summarization is shown in Figure 1.

This paper is organized as follows. In Section 2, we present an efficient video objects segmentation algorithm that is the preprocess of the content-based video summarization system. The key object-based video summarization for static and moving camera is introduced in Section 3. Experimental results are shown in Section 4. Finally, Section 5 concludes this paper.

2. VIDEO OBJECT SEGMENTATION

2.1 Shot detection

Shot detection is an essential step involved in video summarization. A shot is a continuous strip of motion picture that runs for an uninterrupted period of time. We use Zhang's method [13] to detect shot boundaries. Once shots are extracted it is possible to analyze video content based on color, motion, texture and others features.

2.2 Video object segmentation

Motivation of video segmentation can be classified as the application in retrieval, coding, recognition, scenes understanding, summarization, editing and so on. As widely used, many video object segmentation algorithms have been recently proposed [12, 8]. Due to their high complexity, these methods are not efficient for multimedia application, e.g. video summarization.

We have proposed an efficient video object segmentation method in [9] for static camera. In this work we extend this method to moving camera video sequences. Firstly, we oversegment each frame and take the oversegmented regions as the vertices in the 3D spatio-temporal graph. Then multiple cues are fused together to extract objects accurately. Finally, accurate foreground and background segmentation are efficiently achieved by graph cut.

We segment frames use the mean shift method for its good performance and fast speed compared to other methods. Once obtain the over-segmented frames, we construct spatio-temporal graph in the video volume. Then we define the object segmentation as assigning a label to each region. Our 3D graph cuts algorithm solves the labeling problem by minimizing the following energy function:

$$E(l) = \sum_{r\in R} D_r^c(l_r) + \gamma \sum_{r\in R} D_r^m(l_r) + \alpha_1 \sum_{\{i,j\}\in N_s} V_{i,j}^s(l_i, l_j) + \alpha_2 \sum_{\{i,j\}\in N_t} V_{i,j}^t(l_i, l_j) \quad (1)$$

where l stands for the label of each region, $l = 1$ if region r belongs to the foreground, and $l = 0$ otherwise. R is the set of regions in 3D spatio-temporal video volume. $D_r^c(l_r)$ and $D_r^m(l_r)$ represent the conformity of color and motion of a region to the foreground/background respectively. $V_{i,j}^s$ measures color differences between two neighbor regions in the spatial domain (intra frame), $V_{i,j}^t$ measures color differences between two neighbor regions in the temporal domain (inter frame), both encourage two similarity adjacent regions to be assigned same label. N_s is the spatial neighborhood, N_t is the temporal neighborhood, $\{i, j\}$ represent two neighbor regions. γ is responsible for balancing the importance of color cue and motion cue. We normalized D_r^c and D_r^m to the scale of $[0, 1]$, and set the $\gamma = 0.5$ in all of our experiments. The parameters α_1 and α_2 specify the importance of the spatial smoothness term $V_{i,j}^s$ and temporal smoothness term $V_{i,j}^t$. The default values are fixed to $\alpha_1 = 0.05, \alpha_2 = 0.05$ in all of our experiments.

We model the foreground/background color likelihoods in a non-parameter manner, as histogram in the HSV color space. The color likelihood term $D_r^c(l_r)$ is defined as:

$$D_r^c(l_r) = -\log p(C_k|l_r) \quad (2)$$

where C_k represents the mean color of the kth region. Probabilistic normalization requires that $\sum_C p(C|l_r = 1)$, and similarly for the background likelihood.

The construction of motion likelihood based on the assumption that camera are static. So we need to preprocess the moving background sequence. In this work, we register consecutive frames assuming a geometric transformation to estimate global motion. We choose the affine motion model A defined at point $p(x, y)$ by $A(p) = (a_1 + a_2x + a_3y, a_4 + a_5x + a_6y)$ sine it is a good tradeoff between representative and complexity. We employ a feature-based method to estimate the parameter vector $\Theta = (a_1, a_2, a_3, a_4, a_5, a_6)$ between frame I_t and I_{t+1}. We first detect the feature points of each frame by SIFT, which is reported to perform best among popular feature descriptors. SIFT provides a 128-dimensional local descriptor for each keypoint. Next, for each pair of consecutive frames, we match keypoint descriptors using the approximate nearest neighbors kd-tree initially. We build a kd-tree from the feature descriptors in frame I_{t+1}, and for each feature in frame I_t we find two nearest neighbors in I_{t+1} that are $dist1$ and $dist2$. If the ratio of closest distance to 2nd closest distance less than Th_r then accept this match, e.g. $dist1/dist2 < Th_r$. We used $Th_r = 0.6$ and it works well in most of our experiments. After initial matching for consecutive frames I_t and I_{t+1}, we use RANSAC to robustly extract the feature correspondence between the frames and estimate the restricted model (i.e., eliminating outlier that feature correspondence lies on moving objects, and estimating the scaling and translation parameters). Once we obtain the transformations between each pair of consecutive frames. We can warp I_{t+1} to I_t

using transformations to align background (global motion compensation).

The optical flow method of Werlberger et al. [11] is adopted as it runs close to real-time and has very good results on the Middlebury Optical Flow Data set. Then we can use the magnitude of optical flow to describe the motion likelihood of a oversegmented region. The larger magnitude of optical flow of a region, the more likely the region is foreground objects. The motion term $D_p^m(l)$ is defined as:

$$D_r^m(l_r) = -\log p(O_k|l_r) \tag{3}$$

where O_k represents the mean magnitude of optical flow of the kth region. Probabilistic normalization requires that $\sum_O p(O|l_r = 1)$, and similarly for the background likelihood.

The prior term $V_{i,j}^s$ and $V_{i,j}^t$ are well-known Ising prior and can be defined as follow.

$$V_{i,j}^s(l_i, l_j) = \sum_{\{i,j\}\in N_s} (1 - \delta(l_i - l_j))e^{-\beta_1 \|C_i - C_j\|^2} \tag{4}$$

$$V_{i,j}^t(l_i, l_j) = \sum_{\{i,j\}\in N_t} (1 - \delta(l_i - l_j))e^{-\beta_2 \|O_i - O_j\|^2} \tag{5}$$

where δ is Kronecker delta function $\delta((l_i - l_j) \neq 0) = 0$. C_i and O_i are the color and motion expectation of all pixels in the region i repectively, and $\|\cdot\|$ is the L_2 norm. The contrast parameters are set to $\beta_1 = (2 <\| C_i - C_j \|>)^{-1}$, $\beta_2 = (2 <\| O_i - O_j \|>)^{-1}$, where $<\cdot>$ denotes expectation over all pairs of neighbors in video volume.

3. KEY OBJECTS EXTRACTION

The object is detected in a sequence of frames, and each action is represented by a sequence of object masks in those frames. The number of video objects obtained by 3D graph cuts is too big to represent the video content efficient. For example, a people walk on the square, there are many object masks obtained from VOS phase. All of these shapes (actions) are very similar, we can use few representative shapes to depict the video content.

We reduce the number of representative actions to a small fixed number N_r by K-means clustering, N_r is set to 3 in our experiments. This idea is motivated by the fact that any significant actions of video objects are very likely to result in changes in the object's shape and variance of this change can potentially describe the type of object action.

For grouping the actions, we need a shape descriptor or feature vector to represent each object mask. Seven Hu moments [4] is selected which are known to be invariant to translation, rotation and scale change. Mahalanobis distance as distance metric is calculated between the moment description of two shapes. Figure 6 and Figure 9 shows the representative actions of two video Longjump and Stefan.

4. EXPERIMENTAL RESULTS

In this section, we present experimental results on the test sequence 'Hall' (352x288x200) for static background, while 'Longjump' (360x288x100) and 'Stefan' (352x240x250) for moving background. They are shown in Figure 2, Figure 4 and Figure 7, respectively. The most time-consuming part of our method is extract shapes of video objects. Given a video of resolution 352x288 and length 200 frames, it took about 120 seconds to process using a AMD 2.1GHz PC.

Figure 2: Hall sequence

Figure 3: Further summarization of Hall into 3 representative actions by K-means clustering

The experimental results are shown in Figure 3, 5, 6, 8, 9. From the results, we can see that important events are shown. For Hall sequence, the people walk with briefcase, bending to put and walk without briefcase are shown (only use one object for distinct show).

Figure 5 and Figure 8 show the panorama background of long jump scene and tennis court scene. From the panorama view, we can catch the comprehensive surroundings fleetly without browsing every frame of videos. In our experiment, we use the public panorama software to generate background panorama. From these results, we can see that the video content are highly compact in the spatial domain and temporal domain.

Figure 6 reports representative actions of Longjump, such as running up, prancing and dropping. Figure 9 also shows the representative actions which are waiting, hitting and running. These results show that the video information are condensed in content domain effectively.

5. CONCLUSIONS

We have proposed an action-aware scheme for key object-based video summarization. The contribution and characteristics of our scheme are as followings. First, our video object segmentation method is efficient and the KOBVS is easy to implement. Second, it is effective to capture key actions. Third, videos are highly condensed not only in spatial and temporal domain, but also in content domain. In the future, our scheme should be able to discriminate the

Figure 4: Longjump sequence

Figure 5: Panorama background of Longjump

Figure 6: Further summarization of Longjump into 3 representative actions by K-means clustering

Figure 7: Stefan sequence

Figure 8: Panorama background of Stefan

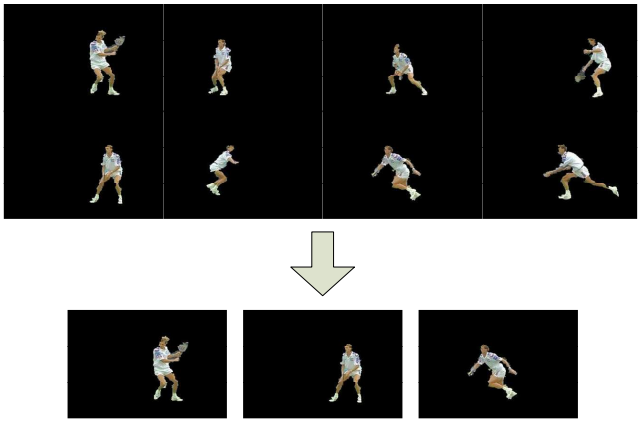

Figure 9: Further summarization of Stenfan into 3 representative actions by K-means clustering

higher level semantic aspects of different activity and events for better summarization of image sequences.

6. ACKNOWLEDGMENTS

This work was supported in part by the National Basic Research Program of China (973 Program) under Grant No. 2010CB327902, the "863" of P.R.China under Grant No. 2009AA011709, and NSFC Nos. 60875008, 90920301 and 60805044.

7. REFERENCES

[1] S. de Avila, A. Lopes, et al. VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32(1):56–68, 2011.

[2] B. Erol and F. Kossentini. Automatic key video object plane selection using the shape information in the MPEG-4 compressed domain. *Multimedia, IEEE Transactions on*, 2(2):129–138, 2000.

[3] M. Furini, F. Geraci, M. Montangero, and M. Pellegrini. STIMO: STIll and MOving video storyboard for the web scenario. *Multimedia Tools and Applications*, 46(1):47–69, 2010.

[4] M. Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, 1962.

[5] C. Kim and J. Hwang. An integrated scheme for object-based video abstraction. In *Proceedings of the eighth ACM international conference on Multimedia*, pages 303–311. ACM, 2000.

[6] Y. Li, S. Lee, C. Yeh, and C. Kuo. Techniques for movie content analysis and skimming: tutorial and overview on video abstraction techniques. *Signal Processing Magazine, IEEE*, 23(2):79–89, 2006.

[7] A. Money and H. Agius. Video summarisation: A conceptual framework and survey of the state of the art. *Journal of Visual Communication and Image Representation*, 19(2):121–143, 2008.

[8] B. L. Price, B. S. Morse, and S. Cohen. Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *ICCV*, 2009.

[9] Z. Tian, J. Xue, N. Zheng, X. Lan, and C. Li. 3d spatio-temporal graph cuts for video objects segmentation. In *International Conference on Image Processing*, 2011.

[10] B. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 3(1):3, 2007.

[11] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic huber-l optical flow. In *The British Machine Vision Conference*, 2009.

[12] B. Xue, W. Jue, S. David, and S. Guillermo. Video snapcut: Robust video object cutout using localized classifiers. In *ACM SIGGRAPH*, 2009.

[13] H. Zhang, A. Kankanhalli, and S. Smoliar. Automatic partitioning of full-motion video. *Readings in multimedia computing and networking*, page 321, 2002.