Active Window Oriented Dynamic Video Retargeting

Chenjun Tao, Jiaya Jia, Hanqiu Sun

Department of Computer Science and Engineering, CUHK, Hong Kong
{cjtao,leojia,hanqiu}@cse.cuhk.edu.hk

Abstract. When playing high-resolution videos on mobile devices with a limited screen size, a commonly arisen but seldom addressed problem is how to naturally condense the source video and to optimally fit it into the target size with lower spatial resolution. We call this problem *video retargeting*. The retargeted video should contain objects of interest and be perceptually seamless and natural.
In this paper, we propose a systematic approach to address this problem by automatically computing an *active window* set with the predefined size inside input videos. Our method contributes in deriving an optimization process to compute the *active pixels* in videos as well as a density map, which jointly constrains the generation of the retargeted video. To avoid the possible local minima, we employ a robust background subtraction method to eliminate unnecessary pixels and apply clustering in initialization. Our approach is general, and is capable of handling videos with complex foreground motions.

1 Introduction

With the rapid growth of video and image capturing ability, on one hand, it is getting easier and common to capture a video with high resolution. On the other hand, sharing and playing these videos on popular mobile devices are handicapped by a set of factors, one of which is the limited screen size in most of the devices. There are rare methods proposed to address this ubiquitous *video retargeting* problem. Simply resizing the video to fit the small screen will sacrifice most of the details. We show one example in Fig. 1, where the original video in (a) has players running after a football. Directly scaling it down results in the loss of most details as shown in (b).

The problem of video retargeting can be regarded as one kind of the video summarization in terms of using smaller spatio-temporal space to *summarize* the original videos. However, conventional approaches shorten the input videos in order to generate temporal segment abstracts[6, 7] while our approach is to generate a seamless video clip by satisfying the following two requirements: 1) The retargeted video should naturally preserve both the temporal and the spatial distance to be faithful to the original video; 2) The retargeted video should also contain as much useful information as possible in terms of object shapes and motions.

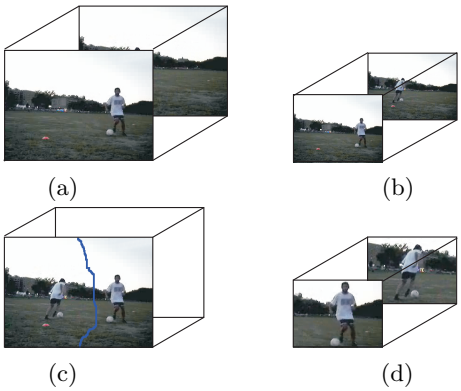

Fig. 1. Video retargeting. (a) The input video. (b) The directly scaled down video. Details of the player and the football are lost. (c) The temporal compression may cause ambiguity when the objects in different frames are placed together. Two footballs appear at the same time. (d) Our method fits a set of windows by tracking most informative pixels in different frames.

Recently, a video retargeting system was proposed in [18] producing a retargeted video to suit the target display with the minimal loss on the important information. Their system was proved to perform well on various kinds of movies with some limitations which can be released by our approach. First, their judgment on the important information is based on the results of low-level feature contrast, face detection, and dominant motion direction. Therefore, if the features and faces cannot be detected well, (e.g., most of the players in sport games always do not face to the camera directly), the locations of the target windows will be ambiguous. Second, "virtual pans" and "virtual cuts" are utilized in [18] to make the optimization, so the orientations of the target windows are restricted to be zero. In order to contain more important information on the target display without producing the ambiguity of the relative positions of the objects, our method allows the target windows change their directions smoothly within a small range.

In this paper, we introduce an automatic approach to solve the general dynamic video retargeting problem by optimizing an *active window* in each of the frames containing most informative pixels. There are 3 steps in our optimization, foreground extraction, the initialization, and dynamic active windows computation. Specifically, we propose to robustly separate the foreground objects from either static or smoothly moving background by minimizing an energy function. To avoid local minima, we introduce a clustering technique to initialize the windows, which is neatly formulated as solving a labeling problem.

The structure of the paper is as follows. An overview of related work is presented in section 2. We describe our approach in section 3, including foreground extraction, dynamic active window optimization and system initialization. Ex-

perimental results are shown and compared in section 4. In section 5, we discuss and conclude our paper.

2 Related Work

The problem of *video retargeting* is addressed by a few papers [20, 19, 18].All of them need to extract important partitions from less important content using an important model. In [20, 19], the background movement is not taken as a factor and their cropping methods may produce the ambiguity of the relative positions of the extracted objects. Unlike [18] utilizing face detection in the important model, our system focus on more general videos and the orientations of the active windows can be adapted to the content of the videos. Recently, a novel method [8] is presented to solve the similar problem on images, which produces excellent retargeted images fast.

Our method utilizes motion separation to acquire necessarily extracted foreground. So, we review most related previous work on multiple motion layers separation in videos. [2, 3] estimate the static background by modeling it with the Gaussian mixture model. In [15], stereo video sequences are required to build the background model in real time to robustly subtract background. In [14] and [16], assuming no large background motion, the foreground moving objects are segmented out from the monocular video sequences considering the difference of gradient and color. [4] employed the optical flow to estimate the camera motion, and used a polynomial equation to model the motion. Assuming that the movements of foreground objects are independent and rigid, the presented method has difficulties to tackle the problem on sports videos where players do not always move rigidly.

For the purpose of multiple-target-tracking in hokey games, [1] builds a standard hockey rink to locate the shot and eliminate the camera motion. This method can estimate the players if the stadium map is given precisely. However, this method doesn't work well if there are few cues that can be extracted from the dynamic scene to locate the shot to the prefabricated map. The algorithm in [5] is a two-step approach. First, motions of objects are estimated using feature points. Then, labels of objects are assigned and refined based on their motions and positions. This method produces good results for motion segmentation when sufficient feature points are obtained.

3 Our Approach

Aiming at generating spatially resized videos containing the most informative regions naturally and seamlessly, our approach consists of the following three steps. The overview of our method is shown in Fig. 2. In the rest of this paper, we represent the color as a vector $\mathbf{I}_p^t$ in RGB color channels for each pixel p in frame t.

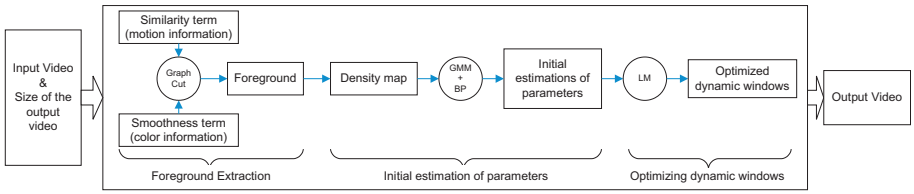

Fig. 2. Overview of our approach.

3.1 Foreground extraction

In extracting the foreground objects, applying methods to detect and track separated object motions considering the object overlapping, occlusion, and dynamic background movement can hardly get satisfactory results. Fortunately, when watching a video, the user is always more interested in the actions or movements of the foreground objects, and is less sensitive to the background scene. Only these pixels from the foreground objects, so called *active pixels*, need to be detected. In other words, we are interested in background subtraction in each frame.

In our approach, we propose to minimize an energy function to automatically separate foreground and background pixels in each frame. We use label $x_p = 0$ to represent that the pixel p is a background pixel while $x_p = 1$ means that the pixel is in the foreground. To analyze motions, initially, we apply the optical flow method presented in [17] to obtain the estimated motion vectors for all pixels in the video. Let $\mathbf{f}_p^t$ be the 2-D optical flow vector of each pixel p in frame t. We group all the $\mathbf{f}$ in each frame into k bivariate normal distributions. Each distribution N_i^t, where $0 \leq i < k$, has the mean μ_i^t, variance Σ_i^t, and the prior weight ω_i^t.

Our energy function $E_b(X)$ is defined as

$$E_b(X) = \sum_p E_{b1}(x_p) + \lambda_1 \sum_{p,q \in N(p)} E_{b2}(x_p, x_q), \tag{1}$$

where $X = \{x_p\}$ is the labeling variables, E_{b2} is a smoothness term and E_{b1} is the probability of a pixel being foreground or background, $N(\cdot)$ denotes the set of neighborhood.

Similarity term. It is noted that the previous work on background subtraction or interactive image and video segmentation [10–12] also models the background and/or foreground colors using a set of clusters. However, in our approach, the Gaussian distributions model the motions of all pixels, without knowing which are in the foreground or background.

For each pixel p, its motion has different probabilities g_p^i falling into different Gaussians i constructed above:

$$g_p^i = \omega_i^t N(\mathbf{f}_p^t; \mu_i^t, \Sigma_i^t). \tag{2}$$

To compute the probability that a Gaussian distribution models a background motion, in general, we assume that the background scene has smooth motion mostly due to the camera movement, which implies that the background should consist of a majority of the pixels with smaller motion variances comparing to the foreground objects. Consider all Gaussian distributions above, a Gaussian cluster modeling background distribution should have a large weight ω_i^t and small variance $||\Sigma_i^t||^{\frac{1}{2}}$. Thus, we formulate the probability that the ith Gaussian distribution models the motions of background pixels as

$$D_b^t(i) = \tau_i^t \omega_i^t / ||\Sigma_i^t||^{\frac{1}{2}}, \tag{3}$$

where $\tau_i^t = (\max_i \omega_i^t / ||\Sigma_i^t||^{\frac{1}{2}})^{-1}$ is the normalization term. Similarly, we define the probability that the motions of foreground pixels are modeled by the ith Gaussians as

$$D_f^t(i) = 1 - \tau_i^t \omega_i^t / ||\Sigma_i^t||^{\frac{1}{2}}. \tag{4}$$

Combining the probability distribution that one pixel is in different Gaussians, we compute the sum of background and foreground motion confidence

$$O_b^t(i) = \sum_{0 \le i < k} D_b^t(i) g_p^i, \tag{5}$$

and

$$O_f^t(i) = \sum_{0 \le i < k} D_f^t(i) g_p^i, \tag{6}$$

respectively. Given Eqn. 5 and 6, the similarity energy term for the labeling of each pixel p can be written as

$$\begin{cases} E_{b1}(x_p = 0) = \frac{O_b^t}{O_f^t + O_b^t} \\ E_{b1}(x_p = 1) = \frac{O_f^t}{O_f^t + O_b^t} \end{cases} \tag{7}$$

Smoothness term. Considering the support from the neighboring pixels, we also introduce a smoothness term to impose penalty on discontinuities between neighboring pixels:

$$E_{b2}(x_p, x_q) = |x_p - x_q| f(p, q), \tag{8}$$

where

$$f(p, q) = \frac{1}{\alpha ||\mathbf{f}_p^t - \mathbf{f}_q^t|| + ||\mathbf{I}_p^t - \mathbf{I}_q^t|| + 1}, \tag{9}$$

where α is a weight. Eqn. 9 constrains that if both the color of neighboring two pixels and their optical flow vectors are similar, the penalty of label difference of p and q is large.

Given the above energy definitions, we compute the optimal segmentation using the Graph Cut method [9] where the pixels in result labeling 0 will be considered as the background while the pixels labeling 1 are the active pixels. We use M^t to denote the label map in each frame t.

There are several literatures targeting at the similar problem,e.g. [3] and [5]. However, their approaches cannot handle videos with quite sparse features and dynamic background. Notice that our foreground subtraction is just the first step in our system. Without explicitly estimating the camera motion, our method has large error tolerance than simple combination of background subtraction and video stabilization.

3.2 Optimizing active windows

Given the extracted foreground layer containing *active pixels*, we optimize the *active windows* in the input videos to fit the target size. We describe two optimization terms, i.e., the informative energy E_{f1}, which guarantees that the dense active pixels are included, and the smoothness energy E_{f2}, which encourages temporal continuity, in this section.

E_{f1} requires that in a general video retargeting framework, the active window in each frame should contain most informative pixels. Obviously, it is not computationally feasible to greedily search all possible positions. We estimate it by constructing density maps.

We defined the parameter vector of each active window as $\mathbf{W}^t = [W_x^t, W_y^t, W_\theta^t]^T$, representing the window center in x and y coordinates, and the orientation θ of the window at frame t respectively. The window width w and height h are predefined values.

We denote the number of the active pixels included in each possible active window $\mathbf{W}^t$ as $d(x, y, \theta)$ in each frame. So, basically, $d(x, y, \theta)$ is a function regarding all possible variables x, y and θ. If we assume $\theta = 0$, $d(x, y, 0)$ can be computed by constructing a corresponding density map using convolution

$$\begin{aligned} d(x, y, 0) &= \sum_{i=x-\frac{h}{2}}^{x+\frac{h}{2}} \sum_{j=y-\frac{w}{2}}^{y+\frac{w}{2}} M(i, j) \\ &= M \otimes f, \end{aligned} \tag{10}$$

where f is a mean filter with the size exactly the same as the active widow and M is the label map as defined in the smoothness term in Section 3.1. If the $\theta \neq 0$, we sample θ using a scale of $\frac{\pi}{36}$ and constrain $-\pi/6 < \theta < \pi/6$ to avoid large rotation. For each θ, we construct a new M_θ rotated on the original label map M Then the density map $d(x, y, \theta)$ can be computed similarly in each frame as

$$d(x, y, \theta) = \sum_{i=x-\frac{h}{2}}^{x+\frac{h}{2}} \sum_{j=y-\frac{w}{2}}^{y+\frac{w}{2}} M_\theta(i, j).$$

Note that the density of the pixels around the border of each frame will be set to zero. Given the density maps computed in all frames, the energy $E_{f1}^t(x, y, \theta)$ is defined as

$$E_{f1}^t(x, y, \theta) = \frac{1}{d^t(x, y, \theta) + \epsilon}, \tag{11}$$

constraining that all active windows include most dense active pixels in the original video, where ϵ is a small number.

The smoothness constraint E_{f2} requires that the center and orientation of the windows $\mathbf{W}$ crossing the frames should be similar. So we define

$$E_{f2} = |\frac{\partial^2 W_x}{\partial t^2}| + |\frac{\partial^2 W_y}{\partial t^2}| + |\frac{\partial^2 W_\theta}{\partial t^2}| \tag{12}$$

Combining the above two terms, we minimize

$$\begin{aligned} E_f(x,y,\theta) &= \sum_t (E_{f1}^t(x,y,\theta) + \lambda_2 E_{f2}^t(x,y,\theta)) \\ &= \sum_t (\lambda_2 (|\frac{\partial^2 W_x}{\partial t^2}| + |\frac{\partial^2 W_y}{\partial t^2}| + |\frac{\partial^2 W_\theta}{\partial t^2}|) \\ &+ \frac{1}{d^t(x,y,\theta)+\epsilon}), \end{aligned} \tag{13}$$

by using the nonlinear Levenberg-Marquardt minimization method to iteratively optimize the parameters.

3.3 Initialization

Notice that in the optimization process described above, there are a large set of parameters to be estimated, which makes the optimization easily stuck in a local minimum. Thus, a good initialization of parameters $[W_x^{t(0)}, W_y^{t(0)}, W_\theta^{t(0)}]^T$ is necessary in our approach to produce a satisfactory retargeted video.

(a) (b) (c)

Fig. 3. GMM on the density maps. (a) shows a set of input frames in a video. (b) is the density map computed in each frame. The background is subtracted, thus, has low densities. We cluster the density maps into GMMs as shown in (c). Warmer color represents larger probability in GMMs.

In this section, we introduce a robust initialization method by first clustering the density map d using a Gaussian-Mixture model (GMM) in each frame. The corresponding EM is performed in 3-D including the 2-D image plane and an additional 1-D density values for all pixels. The output mean vector $\mu_\mathbf{i}^\mathbf{t} = [\bar{x}_i^t, \bar{y}_i^t, \bar{d}_i^t]^T$ for each Gaussian cluster G_i^t in frame t. $(\bar{x}_i^t, \bar{y}_i^t)$ is the coordinate in the image plane and $\bar{d}_i^t$ is the mean density value. The square root of the principal diagonal of the covariance matrix also consists of the standard deviations $\sigma(x_i^t)$, $\sigma(y_i^t)$, and $\sigma(d_i^t)$. We show one example in Fig. 3 that the density

maps are clustered into Gaussian clusters according to the density of the active pixels.

We then construct a single-chain graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ in the input video by representing each GMM G^t in the video as one node in the vertex set $\mathcal{V}$. The notes in immediately neighboring frames temporally are connected using edges $\mathcal{E}$, as shown in Fig. 4(c).

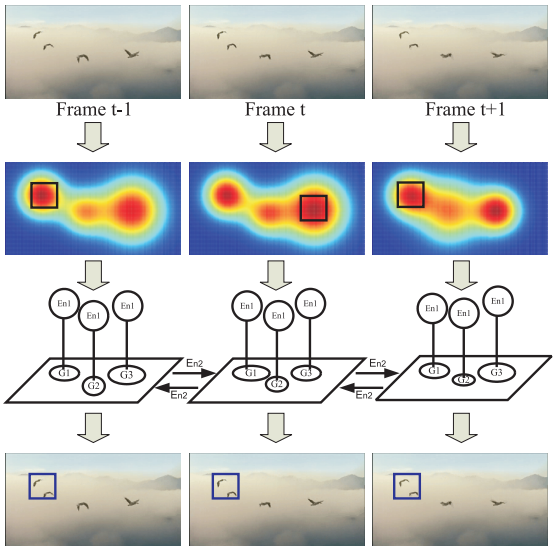

Fig. 4. Initialization in our method. (a) shows a few consecutive frames. (b) are the Gaussian clusters computed on the density maps. The constructed single-chain graph on all GMMs in the video is shown in (c). Each node is a GMM in one frame. (d) shows the initialized windows considering smoothness and active pixel density after the Belief Propagation.

In the initialization, the active windows should be placed inside the clusters with large density means. Meanwhile, the centers of the initial windows in consecutive frames should be close to preserve the temporal smoothness. In our approach, we initialize the window centers as the position of center of the Gaussian clusters in each frame. The orientation $W_\theta^{t(0)}$ is set to be 0 initially.

Suppose that there are K clusters in each frame, the problem to initialize the parameters of active windows is formulated as a labeling problem to search a Gaussian cluster $G_{c_t}^t$, where $c_t \in \{1, 2, ..., K\}$, such that the initial window centers $(W_x^{t(0)}, W_y^{t(0)}) = (x_{c_t}^t, y_{c_t}^t)$. In what follows, we define the Gibbs energy $E_n(C)$ on the graph $\mathcal{G}$

$$E_n(C) = \sum_{c_t \in \mathcal{V}} E_{n1}(c_t) + \lambda_3 \sum_{(c_t, c_{t'}) \in \mathcal{E}} E_{n2}(c_t, c_{t'}), \tag{14}$$

similar to Eqn. 1, where $C = \{c_t\}$,E_{n1} is the likelihood defined on each node, encoding the penalty on each Gaussian cluster individually, and E_{n2} is the prior energy defined on each edge, encoding the cost on the labeling of pairwise nodes.
Likelihood E_{n1}. As described before, to make the initial active windows contain dense active pixels, in cluster level, we assign the labeling cost for each Gaussian cluster $c_t = i$ proportional to its mean density value. Integrating the influence of the Gaussian deviations, we formulate

$$E_{n1}(c_t = i) = \frac{1}{\phi \bar{d}_i^t} \sqrt{\frac{\sigma_n^2(d_i^t)}{\sigma_n^2(x_i^t) + \sigma_n^2(y_i^t) + \varepsilon}}, \tag{15}$$

where ε is a weight, $\sigma_n^2(d_i^t)$ is to impose larger penalty if the Gaussian cluster has large density variance, and $1/(\sigma_n^2(x_i^t)+\sigma_n^2(y_i^t))$ makes region variance large after initialization, leaving sufficient space in active windows optimization in section 3.2. $\phi = \sum_i \frac{1}{\bar{d}_i^t} \sqrt{\frac{\sigma_n^2(d_i^t)}{\sigma_n^2(x_i^t)+\sigma_n^2(y_i^t)+\varepsilon}}$ is a normalization term.
Prior E_{n2}. Considering the temporally connected nodes, E_{n2} encodes the smoothness constraint

$$E_{n2}(c_t, c_{t'}) = \sqrt{(\bar{x}_{c_t}^t - \bar{x}_{c_{t'}}^{t'})^2 + (\bar{y}_{c_t}^t - \bar{y}_{c_{t'}}^{t'})^2}. \tag{16}$$

Given the defined energies, the problem of minimizing $E_n(G)$ is solved using Belief Propagation [13] in an iterative message passing process. We show one example in Fig. 4, where a few consecutive frames are input (a), which are clustered using GMMs in each frame as shown in (b). The corresponding graph constructed in our method is shown in row (c). By solving the optimization problem, we robustly compute the initial active windows as illustrated using the rectangles in (d). It is noted that if we do not consider the temporal smoothness, the initial window parameters will only consider densities, which cause large window jump spatially in the consecutive frames as shown in the rectangle in (b).

4 Experiments

We show our video examples in this section. In our experiments, the parameters λ_1, λ_2, and λ_3 are fixed and set to be 10, 0.1 and 10 respectively. $\epsilon = 0.001$ in Eqn. 11 and $\varepsilon = 0.001$ in Eqn. 14.

Ice hockey game video. We demonstrate in Fig. 5 an example of the ice hockey game. Several frames from the input video are shown in (a) where the players are scattered in the scene. With the defined small window, it is impossible to include all players. We highlight in (a) and (b) our computed active windows using blue rectangles given two different window sizes. Notice that in both cases, the windows are optimized to include most informative pixels of the moving athletes. (c) shows the output from our foreground extraction where the corresponding density map is computed in (d). (e) is a side-by-side comparison using the window size defined in (a). The first row illustrate the result from

direct resizing the whole video. Most details are lost. Our result is shown in the second row. The most informative pixels are included.

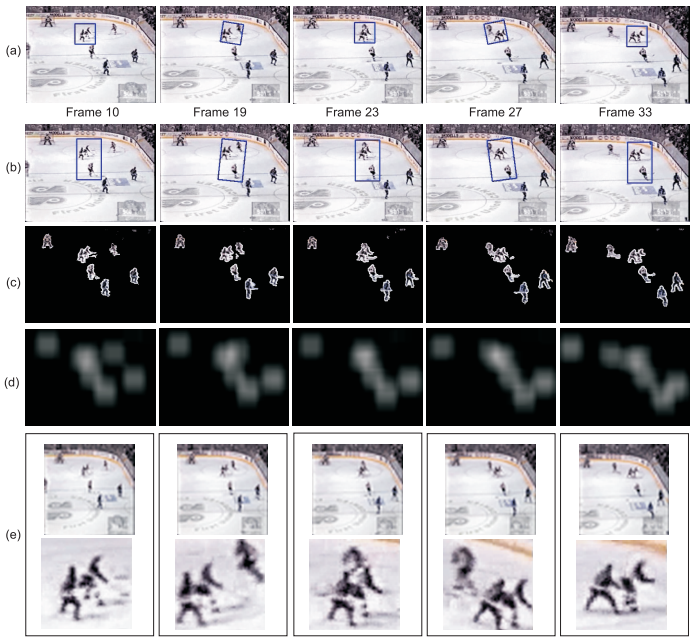

Fig. 5. Ice hockey game example. (a) The input key-frames with the computed active windows highlighted in blue. (b) Using a different size, our method can also produce an optimal output highlighted in blue. (c) The extracted foreground pixels. (d) The corresponding density map. (e) A comparison with the results from directly resized video and our approach.

Football example. This illustration is one of the most complex experiments which is shown in Fig. 6. In this example, the active windows first focus on two players who are running after the ball. From frame 190, a third player run into the focus from the opposite direction. It makes the active window shifting to the center of the three players and rotate to include all of them. In frame 220, the orientation of the active window goes back to zero, since the three players run closely. (b) shows the comparison between our method and the one by directly resizing the video. (c)-left shows one frame (Frame 200) result from the "virtual pan" with the same input video . Since the orientation of the target window is restricted to be zero, the third player cannot be included in this frame.(c)-right shows that our method can reserve more important information.

5 Conclusion and discussion

In this paper, we have proposed a novel video retargeting approach based on active windows aiming to reduce the spatial resolution of an input video without

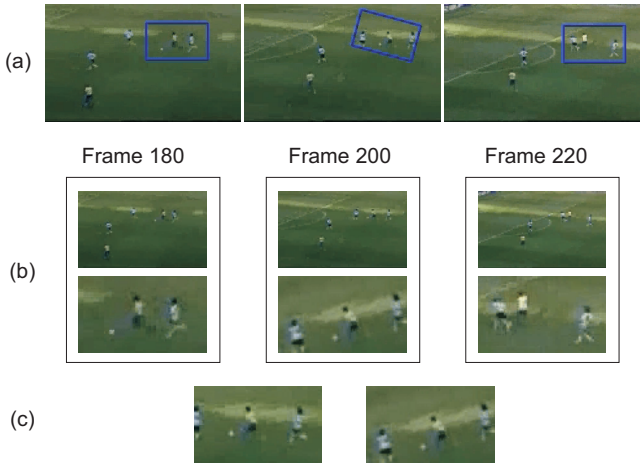

Fig. 6. Football example (shown in color). (a) The input key-frames with the computed active windows. (b) the comparison between the directly resized video and our approach. (c) the comparison with the result of the "virtual pan" (left) and our approach

losing important motion details. Our method consists of the nicely integrated steps: foreground extraction, active window initialization and optimization using three optimization processes.

Our method is different from the conventional video summarization where compressing temporal frames or abruptly reducing relative spatial distance among objects will produce large ambiguities in video understanding when playing the output video alone. The goal of our approach is similar with [18]. However, as illustrated in Fig. 6 (c), our method performs better when new objects need to be included. Our work is readily applicable for sports or surveillant video retargeting on mobile devices.

Limitations: Our method has a couple of limitations. First, our system sometimes produces unsatisfactory results due to the lack of an video understanding scheme. If two or more objects with equal importance are leaving each other, our system may trace the "wrong" one instead of the one which will become more important long time later. Second, when an tiny window is selected, meaningless output video will be produced. However, this problem may be handled by down sample the original frames.

It is difficult to access the quality of our results since the less important information is necessarily to be thrown away. Some of the users believe that the discarded information may indicate the clues of the later events. Therefore, preserving those information without sacrificing the important partitions is still in challenge. Presently, our system preserves the most important information in the target window which satisfies most of the users.

References

1. Yizheng Cai, Nando de Freitas, James J. Little: Robust Visual Tracking for Multiple Targets. ECCV, (2006)
2. J. Sullivan, S. Carlsson: Tracking and Labelling of Interacting Multiple Targets. ECCV, (2006)
3. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. CVPR, **2** (1999) 246–252
4. Rene Vidal, Dheeraj Singaraju: A closed form solution to direct motion segmentation. CVPR, (2005)
5. J. Wills, S. Agarwal, S. Belongie: What went where. CVPR, (2003) 37–44
6. Alex Rav-Acha, Yael Pritch, Shmuel Peleg: Making a Long Video Short: Dynamic Video Synopsis. CVPR, (2006) 435–441
7. Gong, Y., Liu, X.: Video summarization using singular value decomposition. CVPR, **2** (2000) 174–180
8. S.Avidan, A.Shamir: Seam Carving for Content-Based Image Retargeting. ACM SIGGRAPH 2007
9. Yuri Boykov, Olga Veksler, Ramin Zabih: Fast Approximate Energy Minimization via Graph Cuts. ICCV, (1999) 377-384
10. C. Rother, V. Kolmogorov, A. Blake: GrabCut - Interactive foreground extraction using iterated graph cut. ACM Transactions on Graphics (Proc. SIGGRAPH), (2004) 309–314
11. Jue Wang, Pravin Bhat, Alex Colburn, Maneesh Agrawala, Michael Cohen: Interactive Video Cutout. ACM Transactions on Graphics (Proc. SIGGRAPH), (2005)
12. Yin Li, Jian Sun, Chi-Keung Tang, Heung-Yeung Shum: Lazy Snapping. ACM Transactions on Graphics (Proc. SIGGRAPH), **23** (2004) 303-308
13. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, (1988)
14. Jian Sun, Weiwei Zhang, Xiaoou Tang, Heung-Yeung Shum: Background Cut. ECCV, (2006) (628-641)
15. V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, C. Rother: Bi-Layer Segmentation of Binocular Stereo Video. CVPR, (2005) 407–414
16. A. Criminisi, G. Cross, A. Blake, V. Kolmogorov: Bilayer Segmentation of Live Video. CVPR, (2006) 53–60
17. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. ECCV, **3024** (2004) 25–36
18. Feng Liu, Michael Gleicher: Video retargeting: automating pan and scan. MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia, (2006) 241–250
19. Jun Wang, Marcel J.T. Reinders, Reginald L. Lagendijk, Jasper Lindenberg, Mohan S. Kankanhalli: Video content representation on tiny devices. Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on. **3** (2004) 1711-1714
20. Xin Fan, Xing Xie, He-Qin Zhou, Wei-Ying Ma: Looking into video frames on small displays. MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia, (2003) 247–250