Video Retargeting with Multi-scale Trajectory Optimization

Yuanning Li[1,2], Yonghong Tian[3], Jingjing Yang[1,2], Ling-Yu Duan[3], and Wen Gao[3]
[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China
[2]Graduate School, Chinese Academy of Sciences, Beijing, 100039, China
[3]The Institute of Digital Media, School of EE & CS, Peking University, Beijing, 100871, China
{ynli, jjyang}@jdl.ac.cn, {yhtian, lingyu, wgao}@pku.edu.cn

ABSTRACT
Mobile devices are increasingly powerful in media storage and rendering. The prevalent request of decent video browsing on mobile devices is demanding. However, one limitation comes from the size and aspect constraints of display. To display a video on a small screen, rendering process probably undergoes a sort of retargeting to fit into the target display and keep the most of original video information. In this paper, we formulate video retargeting as the problem of finding an optimal trajectory for a cropping window to go through the video, capturing the most salient region to scale towards proper display on the target. To measure the visual importance of every pixel, we utilize the local spatial-temporal saliency (ST-saliency) and face detection results. The spatiotemporal movement of the cropping window is modeled in a graph where a smoothed trajectory is resolved by a Max-Flow/Min-Cut method in a global optimization manner. Based on the horizontal/vertical projections and a graph-based method, the trajectory estimation of each shot can be conducted within one second. Also, the process of merging trajectories is employed to capture more saliency in video. Experimental results on diverse video contents have shown that our approach is efficient and subjective evaluation shows that the retargeted video has gained desirable user satisfaction.

Categories and Subject Descriptors
I.4.9 [**Image Processing and Computer Vision**]: Applications

General Terms
Algorithm, Human Factors

Keywords
Video retargeting, Spatio-temporal saliency, Multi-scale trajectory optimization, Max-Flow/Min-Cut.

1. INTRODUCTION
Recently, handheld mobile devices, such as cell phones, PDA and Pocket PCs are widespread for their mobility and portability. With the advent of functionality and storage capacity, browsing video on a small display of handheld mobile devices becomes a common practice. However, the higher resolution of videos may seriously disrupt the user experience of viewing on a small screen. For example, when a video of 720×468 resolution is simply scaled to the display of size 160×120, some important parts would become too small to read. Moreover, the different aspect ratios of a display may introduce visual distortion. Hence, how to adapt a video to fit for decent viewing on a small display, known as video retargeting, has gained much research interest recent years [1-3, 10- 12, 22-24].

Several simple methods have been proposed to tackle the video retargeting problem, especially in industry. For example, cropping the surrounding part of a video but keeping the central part untouched; resizing the whole video by adding back lettering-boxes at the upper and the bottom of a frame; or non-uniform sampling of salient regions. Those methods can be implemented efficiently whereas a small display may cause different problems. For example, directly cropping may break the composition of a frame; resizing usually leads to too small frame to view; and the non-uniform sampling may introduce distortions or wrapping affects.

In this paper, we incorporate an optimization process into the combination of two basic operations of cropping and scaling to retarget video. The local spatial and temporal saliency is employed to measure the pixel-wise importance of video. As shown in Fig.1, our goal is to find an optimal way to chop the less salient regions and keep the most informative parts to browse on a small display of different sizes. Different from previous works, we propose a trajectory-based globally optimized and locally uniform scaling method to retarget a video to gain a viewer experience of smooth and low distortion. The motivation lies in a finding of psycho-physiological activity, known as *visual attention shifting* [4], when browsing the video. Visual attention can guide and shift the gazing point towards the most interesting parts across frames. Hence, we formulate the problem of video retargeting as the process of finding out an optimal trajectory for a cropping window of a proper size to walk through the spatial and temporal volume (ST-volume). Along the trajectory, the moving cropping window is expected to capture the most saliency out of the ST-volume. This procedure is something like the gaze shifting process, where viewer always focuses on the most interesting parts along the video. In our approach, seeking the best one amongst all the trajectories of multi-scale cropping windows is able to model where to crop and how large ratio of areas to scale on the target display.

Our video retargeting approach consists of three stages. Firstly, the pixel-wise local saliency is computed within a spatiotemporal neighborhood, forming a ST-volume of saliency for each video shot. Secondly, a graph is built up on this ST-saliency volume, with its vertex as potential positions for a cropping window and its edge weight as the saliency continuity of two vertexes (i.e., temporally neighboring cropping windows). Finally, a Max-Flow/Min-Cut algorithm is used to find out the optimal trajectory of a cropping window, which captures the maximal saliency with

MIR'10, March 29–31, 2010, Philadelphia, Pennsylvania, USA.

(a) Retargeted by scaling and letter box

(b)Retargeted by our approach on 160×120 target

(c)Retargeted by our approach on 120×120 target

Fig.1 . An example of retargeting sequent frames of a widescreen video to target displays 160×120 and 120×120 respectively. Trajectories of different sizes of cropping window are covered by boxes of red and blue color in (a). Their corresponding retargeted results are shown in (b) and (c). Note that trajectories are optimized smoothly to capture the most salient regions under different retargeting solutions.

the minimal cost. To speed up this process, horizontal and vertical projection are employed to separately deal with x and y variables of a trajectory. Finally, the optimal size of a cropping window is determined by ranking trajectories, which aims to seek a tradeoff between two basic operations cropping and scaling. Our major contributions are summarized as follows:

1) We propose a novel trajectory based video retargeting method to deal with flexible target display of variable sizes. The optimal trajectory of a cropping window is expected to simulate the gaze shifting process across the ST-saliency volume. The cropping window is not limited to either horizontal or vertical shifting only.

2) We propose a fast algorithm to estimate the trajectory. We introduce both vertical and horizontal projections of a moving window through a ST-saliency volume and estimate the optimal trajectory with a Max-Flow/Min-Cut over a 2D-graph. A smooth trajectory of the cropping window can be obtained at the shot level in a global optimization manner, which can satisfy the live retargeting processing.

The rest of this paper is organized as follow: Section 2 review the related works. Section 3 formulates the problem of video retargeting. Our approach will be presented in section 4. Section 5 presents the experimental results. Finally, a conclusion will be made in section 6.

2. RELATED WORKS

Many works on retargeting image/video over a small display have been reported. For example, Suh et al. [8] crops salient parts of an image to thumbnails based on the low level saliency [5] and high level objects. Xie et al. [9] proposes a window shifting method to browse an image on mobile devices. Although these methods work well in images, the computational complexity and lack of modeling temporal information seriously limits their applications in video domain.

Most of video retargeting approaches can be categorized into two classes: non-uniform sampling methods [2, 3, 22, 24] and cropping/scaling methods [1, 10, 11, 12, 23]. Cheng el al. [3] proposed a content based re-composition method. Their solution recomposes foreground objects and background, Visual attention features (i.e., intensity, color and motion [6]) are used to extract interesting objects. This solution heavily depends on the accuracy of object segmentation. Wolf et al. [2] presents a non-homogenous transforming method where pixel-wise transformation is optimized within a frame based on the saliency map. More recently, Shi et al [22] presents a 3D grid optimization method by taking into account the spatial and temporal saliency. Such works try to keep the aspect ratio of salient objects and squeeze less salient parts by non-uniform sampling. However, visual distortion and wrapping effects would be introduced due to the non-uniform process.

In cropping/scaling methods, a sliding window is used to cover and scale the most salient regions across frames. For example, Fan et al. [10] and Wang et al. [11] retarget video frames to a small display by rendering salient areas with a desirable “display path”. Both methods work on individual frames and optimize retargeting results locally. Liu et al. [1] introduce a set of heuristic penalties to restrict the spatial movements of the cropping window within a video shot. However, their method fixes the cropping window size and restricts the movements to the horizontal pan. It may not be generalized to different video genres. Recently, Deselaers et. al [23] presents a supervised learning method to crop windows based on the manually labeled salient regions. Our approach aims to come up with an automatic and flexible solution that optimizes the retargeting performance at the video shot level.

Most closely related to our work is [12], which also deals with video cropping as a problem of minimizing information loss and solves it as a graph problem. The shifting process of a cropping window in our approach is used to mimic the browsing behavior of a viewer, which has different meaning despite of a similar formula representation in [12]. There are two obvious differences:

1) The method in [12] is proposed to cope with the cropping of surveillance video. Only motion feature is taken into account. It is hard to generalize to other types of video.

2) The initial positions of all potential cropping windows in [12] must be labeled manually throughout the video while our approach is fully automatic. We present a trajectory ranking method to find out the optimal size of a cropping window, while the cropping window size in [12] must be predefined by users.

3. PROBLEM FORMULATION

The human browsing behavior can be regarded as a shifting process to focus on the most interesting parts of the video. To address the retargeting problem, our goal is to find an optimal trajectory of a cropping window, capturing the most salient regions to be scaled and displayed. Ideally, this optimal trajectory is able to approximate the shifting process. However, the independent optimization of a cropping window over individual frames would introduce visible flicker and non-homogeneous zooming which appear like odd camera motions. To produce a smooth trajectory of a cropping window, we carry out the optimization at the shot level. To constrain the zooming effects by retargeting, we fix the size of a cropping window within a shot.

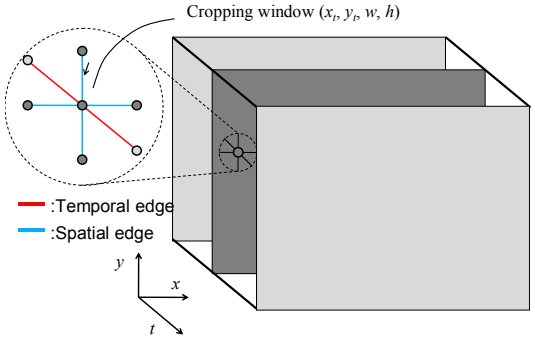


Fig. 2. Searching the optimal cropping window through the ST volume. Each node in the cube corresponds to a potential position of cropping window.

More formally, we deal with video retargeting as a problem of optimizing a smooth trajectory for a cropping window by finding out the optimal size of the cropping window and its trajectory that captures the most of *scaled saliency*. Suppose a source video is of $w_s \times h_s$ resolution and a target display $w_o \times h_o$ resolution. Assume a video shot consists of T frames and each frame t is covered by a cropping window at any position. A cropping window is denoted by (x_t, y_t, w, h) with x_t and y_t being the position in the t frame, w and h specifying the window size. As shown in Fig. 2, the cropping window can move along the spatial and temporal dimensions (i.e. edges in the graph representation) throughout the ST-volume. Two types of edges correspond to the spatial shift and temporal shift, respectively. Now we'd like to solve the problem:

$$\arg\max_{w,h}(\delta_{wo}(w)\delta_{ho}(h)f(w,h)), \quad (1)$$

where

$$\delta_{wo}(w) = (w_o / w)^2, \delta_{ho}(h) = (h_o / h)^2,$$

$$f(w,h) = \max_{\{x_t^*, y_t^*\}_{t=1...T}} \left\{ \sum_{t=1}^{T} [S(x_t, y_t, w, h) + T(x_{t-1}, y_{t-1}, x_t, y_t)] \right\}.$$

In problem (1), δ is the scale factor that penalizes the saliency loss due to scaling. A square lost function is used to alleviate too dramatic downsampling. $f(.)$ measures the amount of saliency that is captured by moving the cropping window throughout the ST-volume. $S(.)$ denotes the saliency captured by the cropping window at a position, $T(.)$ measures the spatiotemporal continuity of two neighboring cropping windows across two successive frames. $\{x_t^*, y_t^*\}_{t=1...T}$ is the optimal trajectory for a cropping window of a fixed size that maximizes the saliency and continuity across the whole video shot. Problem (1) is to maximize the scaled saliency captured by the moving cropping widows of a $w \times h$ size along its optimal trajectory.

Problem (1) can be rewritten as a dual minimum problem if $S(.)$ and $T(.)$ are replaced by cost functions. The problem (1) is thus reformulated as:

$$\arg\min_{w,h}(\delta'_{wo}(w)\delta'_{ho}(h)g(w,h)) \quad (2)$$

$$g(w,h) = \min_{\{x_t^*, y_t^*\}_{t=1...T}} (\sum_{t=1}^{T} [C_s(x_t, y_t, w, h) + C_m(x_{t-1}, y_{t-1}, x_t, y_t)]$$

$$\delta'_{wo}(w) = (w / w_o)^2, \delta'_{ho}(h) = (h / h_o)^2$$

In problem (2), $g(.)$ measures the transferring cost of saliency by moving a cropping window throughout a video shot. $C_s(.)$ denotes the cost of saliency transferring for a cropping window. Intuitively, a cropping window capturing a high saliency means that it has a high ability to transfer saliency at low cost. So $C_s(.)$ decays with the increase of the captured saliency. $C_m(.)$ represents the cost of spatial and temporal movements of a cropping window between two neighboring frames. Hence, problem of video retargeting is reduced to find a trajectory with minimal cost. Details about the definition of the cost functions will be discussed in Sec.4.2.

4. VIDEO REARGETING APPROACH

Solving problem (2) by brute-force search is prohibitively expensive. Instead, we propose an efficient method to find the globally optimal solution. Firstly, different types of local saliency features are extracted to give a final 2D-map for each video frame, producing a ST-volume for each video shot. Secondly, the ST-volume is separated into x channel and y channel by horizontal and vertical projection to compress the searching space. Thirdly, for each channel, a weighed 2D graph is built on their saliency maps for a fixed size of cropping window, with the potential

position of the cropping windows as its vertices, and edges measuring the cost of spatial and temporal movement. Then we cast the problem of finding the optimal trajectory of the given size of cropping window as a Max Flow/ Min-Cut problem on x-axis and y-axis channel graphs, as depicted in Fig. 5. The minimum cuts correspond to the optimal x and y axis trajectories of the cropping window. The cropping window with the maximal scaled saliency and its optimized trajectory is picked up by ranking over all potential sizes of cropping windows. This procedure can be repeated to capture the remaining saliency by "wiping out" the saliency of the optimal trajectory from original ST-saliency volume. Finally, a merging procedure is used to improve the captured saliency. The whole process is summarized in Fig. 3, and more details are presented in the remainder of this section.

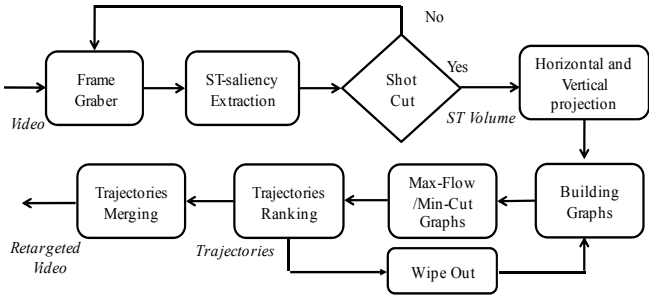


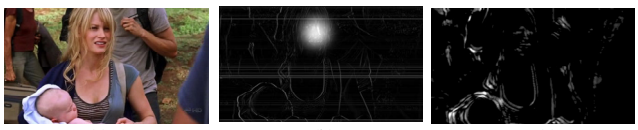
Fig. 3. System Overview.

4.1 Local ST-saliency

In the first parts, local ST-saliency map for each video frame is computed to measure the importance for every pixel. As shown in Fig.4, saliency map consists of three parts, i.e. spatial, temporal and object based saliency. For spatial saliency $\mathbf{S}_{spa}$, a neighborhood in the same frame around a pixel is taken into account to measure how important that pixel to be. For temporal saliency $\mathbf{S}_{tem}$, a fixed number of neighbor frames are used to estimate the motion strength of that pixel. Human faces usually attract attention in video. With the highly developed methods, they can be detected effectively and efficiently. So far, face detection [13] is used for our object based saliency $\mathbf{S}_{obj}$ computation. Other types of saliency can be also integrated into our method. The final map of the local ST-saliency is computed as

$$\mathbf{S} = \min(\mathbf{S}_{spa} + \mathbf{S}_{tem} + \mathbf{S}_{obj}, 1) \tag{3}$$

The value of elements in $\mathbf{S}$ ranges between 0 and 1. The pixel with a big value trends to be important.



(a) (b) (c)

Fig. 4. Local ST-saliency map for a frame. (a) original frame. (b) spatial saliency with face detection results. (c) temporal saliency.

Spatial saliency

As shown in [14], the intensity and contrast of some color pairs may arouse high attention in psychology. Two simple measures are employed for local spatial content, including the intensity of the gradient $\mathbf{S}_{gra}$ and the contrast $\mathbf{S}_{con}$. The spatial saliency is defined as

$$\begin{aligned}
&\mathbf{S}_{spa} = \max(\mathbf{S}_{gra}, \mathbf{S}_{con}) \\
&\mathbf{S}_{gra} = [(\frac{\partial \mathbf{I}}{\partial x})^2 + (\frac{\partial \mathbf{I}}{\partial y})^2]^{\frac{1}{2}} \\
&\mathbf{S}_{con}(x,y) = \mathbf{S}_{rg}(x,y) + \mathbf{S}_{by}(x,y) \\
&\mathbf{S}_{rg}(x,y) = \max_{(x',y')\in N(x,y)} |\mathbf{R}(x',y') - \mathbf{R}(x,y) + \mathbf{G}(x,y) - \mathbf{G}(x',y')| \\
&\mathbf{S}_{by}(x,y) = \max_{(x',y')\in N(x,y)} |\mathbf{B}(x',y') - \mathbf{B}(x,y) + \mathbf{Y}(x,y) - \mathbf{Y}(x',y')|
\end{aligned} \tag{4}$$

In Eq. 4, *N(x, y)* is the 3×3 neighbor of pixel *(x,y)*. $\mathbf{I}$, $\mathbf{R}$, $\mathbf{G}$, $\mathbf{B}$, and $\mathbf{Y}$ denote the intensity, red, green, blue, and yellow component of video frame, respectively. That means, for each frame, the spatial importance value of a pixel is set to the maximum of intensity of the gradient and the contrast with its neighborhood.

Temporal saliency

Motion is a special feature for video that usually attracts attention. Hence, motion is taken into consideration for our temporal saliency. There are many ways to measure the intensity of motion, such as optical flow, the two-dimensional structure tensor [15]. However, errors in motion estimation, object or region based segmentation often add noises to the saliency map. Instead, we use periodic feature [7] to evaluate the temporal saliency for every pixel around its 3D neighborhood. The temporal importance $\mathbf{S}_{tem}$ is defined as

$$\begin{aligned}
&\mathbf{S}_{tem} = (\mathbf{I} * G * H_{ev})^2 + (\mathbf{I} * G * H_{od})^2 \\
&H_{ev}(t,\tau,\omega) = -\cos(2\pi t\omega)e^{-t^2/\tau^2} \\
&H_{od}(t,\tau,\omega) = -\sin(2\pi t\omega)e^{-t^2/\tau^2},
\end{aligned} \tag{5}$$

where G is a 2D Gaussian kernel which is applied along x-axis and y-axis. H_{ev} and H_{od} are a quadrature pair [16] of 1D Gabor filters applied along t-axis. τ, ω and other parameters use the same configurations as [7]. By Eq.5, periodic motion (e.g. mouth talking and bird flapping) and other complex motions (e.g. spatio-temporal corner) can induce a strong response. Although it is a simple measure for motion, periodic feature has shown promising in our work and behavior recognition [7].

4.2 Horizontal and vertical projection

An important step to solve the problem (2) is to find a smooth trajectory of a cropping window with the most saliency. A fixed size of cropping window needs to be located at each frame. According to the center-surround hypothesis of visual saliency [21] and the separableness of the movement, we employ the horizontal and vertical projections of ST-volume to optimize trajectory in x-axis and y-axis separately. Just as text detection [17], projections of a 2D saliency importance map $\mathbf{S}_t$ at the t^{th} frame to 1D array $\mathbf{P}_{t,x}$ and $\mathbf{P}_{t,y}$ are defined as

$$\mathbf{P}_{x,t}(i) = \sum_{j=0}^{y_{\max}} \mathbf{S}_t(i,j) \qquad \mathbf{P}_{y,t}(j) = \sum_{i=0}^{x_{\max}} \mathbf{S}_t(i,j) \tag{6}$$

$0 \le i \le x_{\max}$ and $0 \le j \le y_{\max}$

Horizontal projection of $\mathbf{S}_t$ equals to the sum of the pixels in a same column, and vertical projection corresponds to the sum of pixels in a same row. Accordingly, a 2D map $\mathbf{P}_x$ ($T \times (1 + x_{\max})$) and a 2D map $\mathbf{P}_y$ ($T \times (1 + y_{\max})$) can be obtained from a shot of

T frame. A 3D ST-saliency volume is then reduced to two 2D maps, i.e. $\mathbf{P}_x$ and $\mathbf{P}_y$.

Now, we can separate the problem (2) into x-axis channel and y-axis channel and rewrite it as follow:

$$\arg\min_{w,h}(g(w,h))$$

$$g(w,h)=\delta'_{wo}(w)g_1(w)+\delta'_{ho}(h)g_1(h),$$

$$g_1(w)=\min_{\{x_t^*\}_{t=1...T}}\left\{\sum_{t=1}^{T}[C_s(x_t,w)+C_m(x_{t-1},x_t)]\right\}, \quad (7)$$

$$g_1(h)=\min_{\{y_t^*\}_{t=1...T}}\left\{\sum_{t=1}^{T}[C_s(y_t,h)+C_m(y_{t-1},y_t)]\right\}.$$

In problem (2), $g_1(w)$ represents the cost of transferring saliency along the optimal trajectory in the x-axis channel, and $g_1(h)$ corresponds to the y-axis channel. $\{x_t^*\}_{t=1...T}$ and $\{y_t^*\}_{t=1...T}$ denote the x variables and y variables of the optimal trajectory respectively. $C_s(.)$ measures the cost of transferring saliency for a cropping window that covers on a 1D saliency importance array. $C_m(.)$ measures the cost of the spatial and temporal movement between two cropping windows in neighboring frames. We define $C_s(x_t,w)$ as

$$C_s(w,x_t)=\exp\left\{-\mathbf{Q}_x^2(w,x_t)\right\}, \quad (8)$$

$$\mathbf{Q}_x(w)=\mathbf{P}_x * G(w),$$

$C_s(y_t,h)$ is defined in a same way. In Eq.8, $\mathbf{Q}_x$ is calculated by convolving the 2D map $\mathbf{P}_x$ with a 1D Gaussian Kernel $G(w)$ along x-axis with the size of w. $\mathbf{Q}_x(w,x_t)$, which is calculated as the weighted sum of the neighborhood around x_t in $\mathbf{P}_x$, stands for the saliency that is captured by a w width cropping window at the x_t position. $C_s(x_t,w)$ is inverse proportion to the saliency that the cropping window (x_t,w) covers. Details of the implementation of $C_m(.)$ are presented in the next sub-section.

4.3 Building the graph

In this subsection, we demonstrate how to find the x variables of the optimal trajectory across a video shot of T frames. The estimation of the y variables of the trajectory can be solved in a similar manner. We first denote a graph as $\mathbf{G}=(\mathbf{V},\mathbf{E})$. The vertex set is denoted as $\mathbf{V}$ and the set of edge between vertexes as $\mathbf{E}$. Let each cropping window can be represented by a vertex $v\in\mathbf{V}$ and horizontal saliency importance map of each frame can be sampled by m 1D cropping windows of width w. There are totally $|\mathbf{V}|=mT+2$ vertices in the graph of x-axis channel of the shot. An illustration of the graph is shown in Fig. 5. In the figure, each vertex in the mesh is four-connected to its neighbors. In particular, each vertex of the candidate cropping window is connected with its neighbors by "spatial edge" in x-axis and "temporal edge" in t-axis. An additional "source" node connects to the x_0 nodes at all frames, and a "sink" node connects to $x_{\max}$ nodes. We have

$$\mathbf{E}=\begin{cases}(u,v) & \text{if } (u-v)=(\Delta_x,0) \text{ or } (u-v)=(0,\Delta_t)\\ (\text{source},u) & \text{if } u=(x_0,t)\\ (\text{u},\text{sink}) & \text{if } u=(x_{\max},t)\end{cases} \quad (9)$$

We define the cost function $C_m(u,v)$ of moving the cropping window between vertices u and v as

$$C_m(u,v)=\begin{cases}\infty & \text{if } u=\text{source or } v=\text{sink}\\ c_{spa}(u,v) & \text{if } (u-v)=(\Delta_x,0)\\ c_{tem}(u,v) & \text{if } (u-v)=(0,\Delta_t)\end{cases}$$

$$c_{spa}(u,v)=\frac{C_s(u)+C_s(v)}{2} \quad (10)$$

$$c_{tem}(u,v)=kc_{spa}(u,v)\quad(0\le k\le\infty)$$

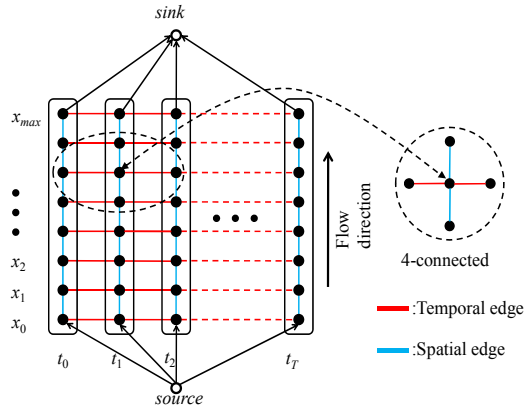


Fig. 5. A graph of trajectory optimization in x-axis. Each vertex corresponds to a potential position of a cropping window (i,e, position in the x-axis and the temporal axis).

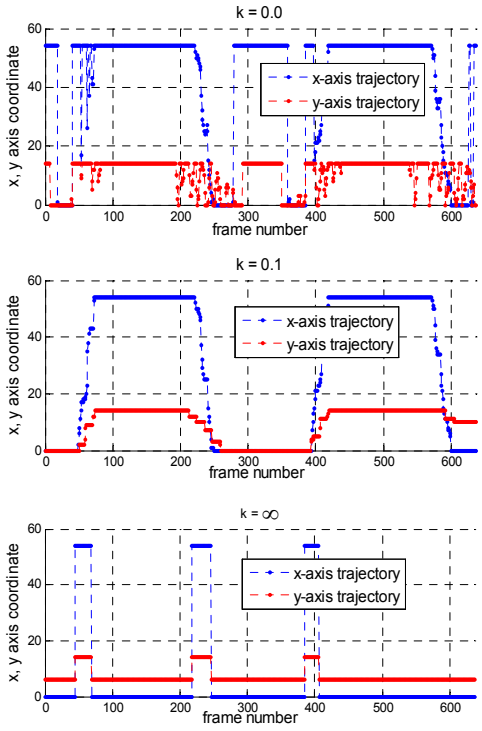


Fig. 6. Toy examples of trajectory estimation for different smoothness factors. $k=0$ corresponds to the maximal independence, $k=0.1$ corresponds to an intermediate smoothness and $k=\infty$ to the maximal smoothness.

In Eq.10, $c_{spa}(.)$ is the cost of moving a cropping window along a spatial edge, while $c_{tem}(.)$ corresponds to the cost along a temporal edge. The cost of an edge is derived from the cost of two vertices that it links. The average of two vertices cost is a heuristic strategy that works well in practice, meaning that a cropping window is likely to move between two nodes with low transferring cost. *k* in Eq.10 is a smoothness parameter for temporal continuous. A higher temporal cost (larger *k*) increases the smoothness of the trajectory, while a lower temporal cost (smaller *k*) increases the discontinuity of the trajectory.

Since dynamic programming is not suitable in this case, we use the Max-Flow/Min-Cut to find the min-cut for the graph. To illustrate the effect of the smoothness parameter *k*, we show a toy example problem in Fig. 6. A min-cut from the left side to the right side of the graph is computed with Max-Flow/Min-Cut algorithm. The min-cut is computed in the graph with the smoothness factors 0, 0.1, and ∞ . The corresponding cuts in x-axis and y-axis are displayed by the blue and red dash lines respectively. Setting *k* to zero, each column of the graph is cut independently, archiving the maximal discontinuity. When $k = \infty$, the minimum cut is flat with the maximal smoothness. A tradeoff can be achieved when *k* = 0.1. We empirically set *k* to 0.1 in our current implementation.

4.4 Trajectory estimation by Max-Flow/Min-Cut

It's been proven in [18] that once the maximum flow is found, a min-cut *C* of the graph minimizes the sum of edge cost separates the source and sink. As demonstrated by property 1, this cut also provides the trajectory with the minimal cost for a cropping window to go through the whole video shot. Hence, the min-cuts of x-axis and y-axis channel provide the optimal estimation of the trajectory of the moving cropping window.

Property 1.

Consider a cut C with some flow from source to sink in graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$. For all *t*, there exist at least one *x* (or *y*) such that the edge $(x,t) - (x+1,t)$ (or $(y,t) - (y+1,t)$) is part of C.

Proof: For any *t*, there is a path from source to sink in the form of $\text{source} -> (0,t) -> (1,t) -> ... -> (x_{\max},t) -> \text{sink}$.

Therefore it contains a set of edges

$$\left\{ \begin{array}{c} \text{source} -> (0,t) \\ (x,t) -> (x+1,t) \quad x \in [0, x_{\max}] \\ (x_{\max}, \text{sink}) \end{array} \right\}$$

Any cut of **G** must break this path and thus contain at least one edge in the form of $(x,t) - (x+1,t)$.

Currently, we solve the Max-Flow/Min-Cut problem with the "augmenting paths" method based on [19]. Two search trees are built from the source and the sink. Flow is pushed along non-saturated paths from the source to the sink until the maximum flow is reach. For a frame of size *l* pixels and a cropping window of size *w*, the number of the vertices *n* is equal to the number of the potential cropping windows throughout the video shot, meaning $n = T(l - w)$. Since the graph is a four-connected, the number of edge is about $e = 4n$. The worst complexities of the algorithm is $O(n^2\sqrt{e})$. However, the topology of the graph trends to make the average running time much smaller than that of the worst case. Fig. 8 and 9 show the typical performances under different conditions.

4.5 Trajectory Ranking

Trajectory ranking is to find out the optimal size of the cropping window. On one hand, video retargeting aims to maintain the original information from each frame as much as possible. On the other hand, information may be lost due to a sharp downsampling. Experimental results in [20] show that 32×32 color image are the minimum size for people to recognize an object. Any lower resolution results in rapid drop of the recognition performance. Hence, retargeting must find a tradeoff between the saliency and visibility. As defined in problem (1), $f(w,h)$ denotes the saliency captured by moving a $w \times h$ cropping window along its optimal trajectory. $\delta_{ho}(h)$ and $\delta_{wo}(w)$ denote the scaling factors that measure the visibility lost caused by scaling. Then trajectories can be ranked according to their scaled saliency.

Since the trajectory estimation can be conducted efficiently, we choose a brute-force search throughout the space of size parameter in trajectory ranking. Specifically, we start with the target display size $w_o \times h_o$, then iteratively enlarge the window width and height respectively with a fixed scale factor until the cropping window grows out of the original frame. The scale factor is currently set to 1.1 in our implementation.

4.6 Merging Trajectories

Activities may happen simultaneously in different areas of the video, especially in the wide-screen video. Retargeting video to small display by one single cropping window may miss some interest parts of the video. Hence, we merge multiple trajectories to capture as much saliency as possible. Saliency captured by the optimized trajectory is firstly wiped out from the ST-saliency volume before the next run of the trajectory estimation. Once we get multiple optimized trajectories, we could view them on different displays at the same time.

But for video retargeting on a single target display, only one trajectory of the cropping window is allowed to be displayed. Hence, we merge several optimized trajectories into one single trajectory. Currently, local greedy method is adopted to merge the best two trajectories. The final trajectory at each frame is decided by the results at its neighboring frames. Fig. 7 illustrates an example of trajectories merging. Fig. 7(a) is the original frames (scaled for display) of equal time interval. The sliding boxes of blue/red color correspond to two trajectories respectively. Fig. 7(b) shows the retargeted frames by merging two trajectories. Fig. 7(c) illustrates the saliency that is captured by the two trajectories at different frames.

5. EXPERIMENTAL RESULTS

Our approach has been implemented on a standard PC (2.22GHz, and 2G RAM). Two common sizes of mobile displays, i.e., 160×120 and 120×120, are used in our experiments. To demonstrate the performance, we conduct three experiments. Firstly, we study the computational performance. Secondly, we evaluate the capability of our approach to capture the salient regions by user study. Finally, we assess the browsing experience through subjective evaluation.

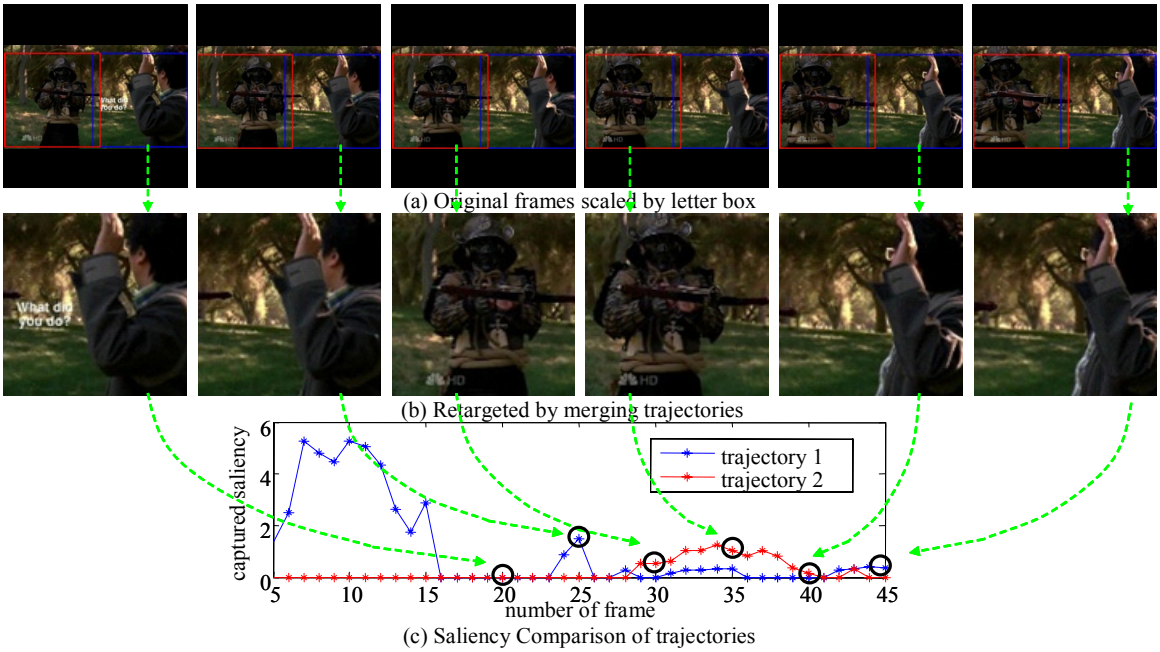


Fig. 7. Illustration of merging trajectories. Two trajectories of cropping window are marked with boxes with different colors in (a). Segments of trajectories with higher saliency are merged together and shown in (b). Saliency distributions of two trajectories are shown in (c).

Participants

Five computer science students (two females and three males) participate in the user study. All participants don't have any technical knowledge on video retargeting work in this paper.

Data set

Two video sets are used in our experiments. The first set consists of "widescreen" movies with an aspect ratio of 16:9, which is widely used in video retargeting research. The second set comes from TRECVID'07 dataset, which is a documentary video corpus at the aspect ratio of 4:3, whose content is more similar to home video. Over two sets of videos, diverse video contents and shooting styles pose challenges for video retargeting in practice. Some example results from two data sets are given in Fig. 11. The prototype system is able to perform as we expect, namely, choosing a proper scale of the cropping window adaptable to video content and generating a smooth trajectory for browsing.

5.1 Speed for Trajectory Estimation

We first report the computational load of our approach. The main computational complexity lies in two steps:

1) Computing the ST-volume of saliency.

2) Estimating trajectory.

For the first step, saliency can be extracted in real time due to the simplicity of our used saliency measures. The processing speed for spatial saliency and temporal saliency is over 30fps. Shot boundary is detected by comparing simple color histogram intersection to a predefined threshold. Hence, shot detection does not involve heavy computation either. Face detection is time-consuming. Some 16 frames can be processed in a second for a 720×405 video.

For the second step, though the theoretically worst complexity of Min-Cut is polynomial with the number of graph nodes, our approach can be conducted efficiently because of the structure and topology of the graph. Fig. 8 shows the typical retargeting performance for a shot of 54 frames (720×405) as a function of the number of potential cropping windows c in each frame. The average running time is $O(c)$, which is linear with respect to the difference between the original frame size and the cropping window size.

Then we fix the number of potential cropping windows for each frame (c = 800) and assess the performance over different sizes of the shot. As illustrated in Fig. 9, the running time varies due to the content of the corresponding shot. For the worst case, it is linear with the number of frames within each shot. But the time for most of the runs is far less than that of the worst case. Typically, we were able to process a video clip of 635 frames in about 52 seconds (12fps), including the shot cut, saliency extraction, trajectories estimation and merging.

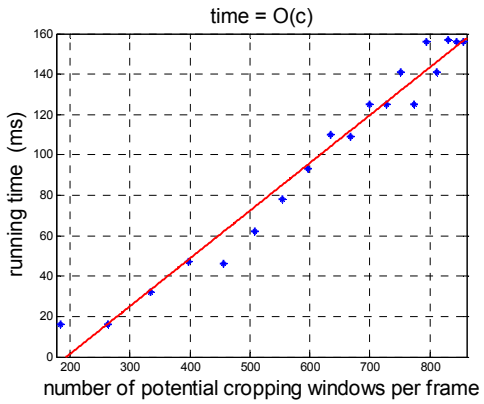


Fig. 8. Performance of trajectory estimation as a function of the potential cropping window number in a frame.

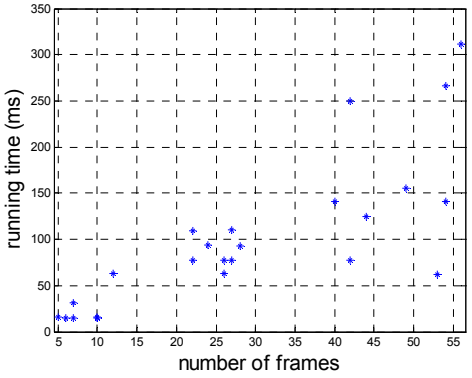


Fig. 9. Performance of trajectory estimation as a function of frame number. The number of potential cropping windows is fixed cropping window for a frame.

5.2 User study experiments

Part1. Effectiveness in capturing saliency

As mentioned, we want to find an optimal trajectory of the cropping window to mimic the gaze shifting process throughout the video. To evaluate the effectiveness in capturing saliency, we ask the participants to label the most interest regions in i-frames of the testing videos, resulting in about 16k labeled frames for videos of about three hours in length. The salient objects in adjacent sampled frames are manually labeled by subjects as we did in [25]. The labeling results are then merged to generate the ground-truth saliency maps. Then our retargeted results are measured by a hit rate h , which is defined to measure the coverage of the retargeted results over the ground-truth. The hit rate of the retargeted trajectory over the labeled frames i frame is defined as:

$$h(i) = \frac{S_{\text{labeled}}(i) \cap S_{\text{retargetd}}(i)}{S_{\text{labeled}}(i)} \quad i = 1...T, \tag{11}$$

where $S_{\text{labeled}}(i)$ denotes the area of the labeled regions in the i frame, and $S_{\text{retargetd}}(i)$ is the area covered by the trajectory of the cropping window in the corresponding frame.

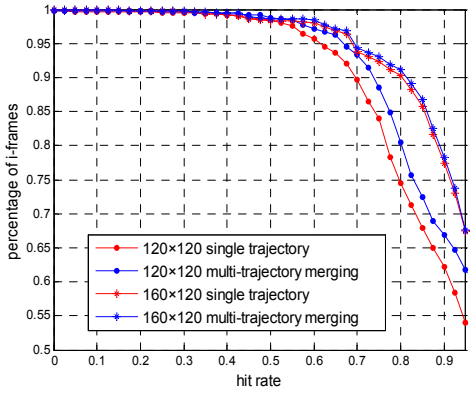


Fig. 10 . Accumulated hit rate distributions for different trajectory based retargeting methods.

The accumulated distribution of average hit rates for different retargeting methods and different display sizes are illustrated in Fig. 10. The x-axis denotes the hit rates while the y-axis the percentage of i-frames whose hit rate is higher than the corresponding hit rate in x-axis. The curve with slower decline implies that the retargeted result is more consist with the actual gaze shifting process. Two observations are revealed:

1) Retargeted results on a bigger display show slower decline. This shows that the size of the target display is still the main factor affecting the viewing experience.

2) Retargeted results by multi-trajectory merging are more close to the human labeling than that of a single trajectory. This phenomenon may be attributed to the fact that a multi-trajectory method captures more ST-saliency than the single one, which is coincident with the results in [4, 12].

Part 2. Subjective assessment

In this part, we ask the participants to evaluate the retargeted results over shots which are obtained by multi-trajectory merging. In table 1, "Good" stands for that the content captured by our method represents what users have expected from the original video and fit for viewing smoothly, "Accept" means that the position of the cropping window is not good enough but still acceptable for viewing, while "Bad" means that the retargeted result misses over 50% of important parts of video or interrupts viewing. Although we only use simple saliency features currently, about 80% of the results are good or acceptable in both data sets. In the most of cases, our approach produces a smooth trajectory that zooms to the salient objects. However, in most of the failure cases, camera motions draw the trajectory of a cropping window to focus on the background regions with fierce variation.

Table 1. Subjective assessment results over two video sets.

DataSet	Shot #	Good	Accept	Bad
Widescreen Movies	751	72.4%	15.7%	11.9%
TREC VID 07	723	68.5%	21.9%	9.6%

.

6. CONCLUSION

A novel approach for video retargeting has been presented, where a trajectory of cropping window is introduced to simulate the attention driven human gaze shifting process. Video retargeting is formulated as a problem to find an optimal cropping window size and an optimal trajectory for the cropping window.

The key idea of our approach is that we turn the trajectory estimation to a graphical problem solved by Max-Flow/Min-Cut method. Firstly, finding the optimal trajectory through the ST-volume of saliency is reduced to a 2D searching problem by horizontal and vertical projection. Then we show that a global optimized trajectory can be efficiently estimated by Min-Cut a graph, which depicts candidate cropping windows as vertexes and their spatial and temporal continuity as edges. In practice, this procedure can be finished within one second and repeated to compute multiple independent trajectories. We also show that retargeted results can be further improved by merging trajectories.

Our approach is fully automatic, globally optimized, without any constraints on the movement of the cropping window. Despite the stage of shot detection, our approach can be implemented efficiently for real time retargeting. At present, our approach relies on three types of saliency measures, including spatial saliency,

"TREC VID 07" "TREC VID 07" "Akiyo" "Chubbchubbs" "Lost"

Fig. 11. Illustration of some retargeted results. The original frames scaled by adding letter box are shown in the first row. Results of the optimized cropping window trajectories for 160×120 and 120×120 target displays are marked with red and blue box respectively. The corresponding retargeted results are shown in the second and third rows. Our retargeting method zooms to the interest region in different scenes.

temporal saliency and face detection. Experimental results demonstrate that those simple saliency features are effective and suitable for the real time process. In the future research, we intend to incorporate more information to enhance the saliency extraction.

7. Acknowledgments

The work is supported by grants from the Chinese National Natural Science Foundation under contract No. 60973055 and No. 90820003, National Hi-Tech R&D Program (863) of China under contract 2006AA010105, and National Basic Research Program of China under contract No. 2009CB320906.

8. REFERENCES

[1] F. Liu and M. Gleicher. Video retargeting: automating pan and scan. In Proceedings of the ACM international conference on Multimedia, pages. 241–250, 2006.

[2] L. Wolf, M. Guttmann, D. Cohen-Or. Non-homogeneous content-driven video-retargeting. In Proceedings of the IEEE 11th International Conference on Computer Vision, pages 1 – 6, 2007.

[3] W. H. Chen, C. W. Wang, J. L. Wu. Video Adaptation for small display based on content recomposition. IEEE Transactions on Circuits and Systems for Video Technology, 17(1), pages 43-58, 2007.

[4] L. Itti and C. Koch. Computational modeling of visual attention. Nature Rev. Neurosci., 2(3), pages 194–203, Mar. 2001.

[5] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(11), pages 1254–1259, Nov. 1998.

[6] L. Itti and P. Baldi. A principled approach to detecting surprising events in video. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 631–637, 2005.

[7] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In Proceedings of International Conference on Computer Communications and Networks, pages 65 – 72, 2005.

[8] B. Suh, H. Ling, B. B. Bederson, and D.W. Jacobs. Automatic thumbnail cropping and its effectiveness. In Proceedings of the 16th annual ACM symposium on User interface software and technology, pages 95 - 104, 2003.

[9] X. Xie, H. Liu, W-Y Ma, and H-J Zhang. Browsing Large Pictures Under Limited Display Sizes. IEEE Transactions on Multimedia, 8(4), pages 707-715, 2006.

[10] X. Fan, X. Xie, H.-Q. Zhou, and W.-Y. Ma. Looking into video frames on small displays. In Proceedings of the ACM international conference on Multimedia, pages 247–250, 2003.

[11] J. Wang, M. J. Reinders, R. L. Lagendijk, J. Lindenberg, and M. S. Kankanhalli. Video content representation on tiny devices. In Proceedings of the IEEE International Conference on Multimedia and Expo, pages 1711– 1714, 2004.

[12] E. A. Hazem, J. David, D. Larry. Multi-scale video cropping. In Proceedings of the ACM international conference on Multimedia, pages 97–106, 2007.

[13] J. Chen, R. Wang, S. Yan, S. Shan, X. Chen, Wen Gao. Face detection based on the example resampling by manifold. IEEE Transactions on System Man, and Cybernetics (Part A), 37(6), pages 1017-1028, Nov. 2007.

[14] S. Engel, X. Zhang, and B. Wandell. Color tuning in human visual cortex measured with functional magnetic resonance imaging. Nature, 388 (6637), pages 68–71, 1997.

[15] C.-W. Ngo, T.-C. Pong, and H.-J. Zhang. Motion analysis and segmentation through spatio-temporal slices processing.

IEEE Transactions on Image Process, 12(3), pages 341–355, 2003.

[16] G. Granlund and H. Knutsson. Signal processing for computer vision. Kluwer Academic Publishers, Dordrecht, Netherlands, 1995.

[17] M. R. Lyu, J, Song, and M. Cai. A comprehensive method for multilingual video text detection, localization, and extraction. IEEE Transactions on Circuits and Systems for Video Technology, 5(2), pages 243-255, 2005.

[18] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. Introduction to Algorithms. McGraw-Hill, New York,1990

[19] Y. Boykov, V. Kolmogorov. An experimental comparison of Min-Cut/Max-Flow algorithm for energe minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(9), pages 1224–1237, Sept. 2004.

[20] A. Torralba and R. Fergus and W. T. Freeman. Tiny image. In MIT-CSAIL-TR-2007-024.

[21] D. Gao, M. Vijay and V. Nuno. Discriminant center-surround hypothesis for bottom-up saliency, In Proceedings of Neural Information Processing Systems (NIPS), 2007.

[22] L. Shi, J. Wang, L. Y. Duan, H. Lu. Consumer Video Retargeting: Context Assisted Spatial-temporal Grid Optimization. In Proceedings of the ACM international conference on Multimedia, pages 301-310, 2009.

[23] D. Thomas, D. Philippe, and N. Hermann. Pan, zoom, scan time-coherent, trained automatic video cropping. In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pages 1-8, 2008.

[24] M. Rubinstein, A. Shamir, and S. Avidan. Improved seam carving for video retargeting. In Proceedings of ACM International Conference on SIGGRAPH, 2008.

[25] J. Li, Y. Tian, T. Huang and W. Gao, A dataset and evaluation methodology for visual saliency in video, in Proceedings of IEEE International Conference on Multimedia and Expo, 2009.