Crowd-sourced Automatic Zoom and Scroll for Video Retargeting

ABSTRACT

Screen size and display resolution have limited the viewing experience of videos on mobile devices. The viewing experience can be improved by determining important or interesting regions within the video (called regions of interest, or ROIs) and displaying only the ROIs to the viewer. Previous work focuses on analyzing the video content using visual attention model to infer the ROIs. Such content-based technique, however, has limitations. In this paper, we proposed an alternative paradigm to infer ROIs from a video. The idea is to crowd-source from a large number of users through their viewing behavior and infer the ROIs from their collective wisdom. We let users interact with the video through a zoom and pan interface. What these users choose to view can be used to determine the best ROIs. We thus generate a retargeted version of the video consisting in relevant shots determined from historical users behavior. Such automatically retargeted video can be replayed to subsequent users who would prefer a less interactive viewing experience. This paper presents how we collect the user traces, infer the ROIs and their dynamics, group the ROIs into shots, and automatically reframe those shots to improve the aesthetics of the video. A user study with 48 participants shows that our automatically retargeted video is of comparable quality to one handcrafted by an expert user.

1. INTRODUCTION

In recent years, we have seen a proliferation of consumer video cameras that are capable of recording high-definition video (1920× 1080 pixels or more), capturing fine details in their shots. Consumers, however, are increasingly viewing videos on mobile devices, that are constrained in their screen size and display resolutions due to their form factors. Many of the captured details are lost due to the impedance mismatch between the two resolutions.

Past research have shown that users tend to zoom in and pan on images [18] and videos [1] to view the details on a small screen. Through a user interface, users can select the region in the image or the video that are of interest (called *region of interest*, or ROI for short) to display in finer details.

While letting users specify the zoom and pan actions explicitly put users in control of what they want to see, it requires the users hand for interaction. In certain scenarios, however, minimal user interaction is preferred, for example, watching video during workout in gym and taking notes while watching the video. Automatically identifying and displaying the ROIs to users is useful for such scenarios.

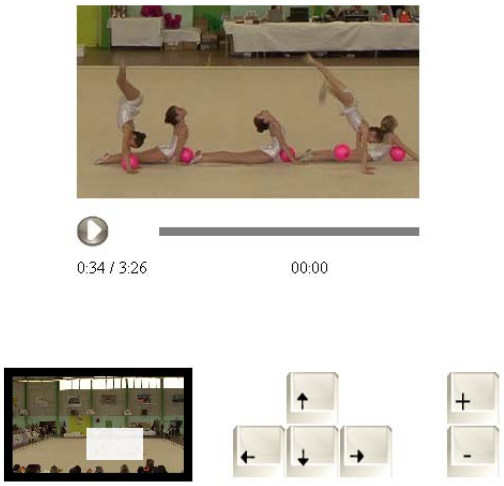

Figure 1: User interface of the zoomable video player

Existing research have investigated various methods to *retarget* an image or a video for display on small screens. The proposed methods generally aims to (i) identify the most informative regions in an image or a video, and (ii) display them to the users in a visually pleasing way. The methods fall into two categories: content-based methods identify the ROI based on low-level features (e.g., color, motion) or semantic (e.g., face, text) of the content; while usage-based methods identify the ROI based on user feedback (e.g., explicit zoom and pan commands).

We found that content-based methods for identifying ROIs do not work well in all cases, especially in a video with complex scenes. Consider the example of the video of a street performer. There could be multiple pedestrians walking around, or audience crowding around the performer, potentially wearing clothes in attention-catching colors. The ROI, however, should remain on the street performer. Content-based methods, which are based on salient features of the video, could easily get confused and extracted pedestrians or audiences as the ROI in addition to the performer.

Despite the tremendous innovation and achievement of the re-

ACM Multimedia Italy

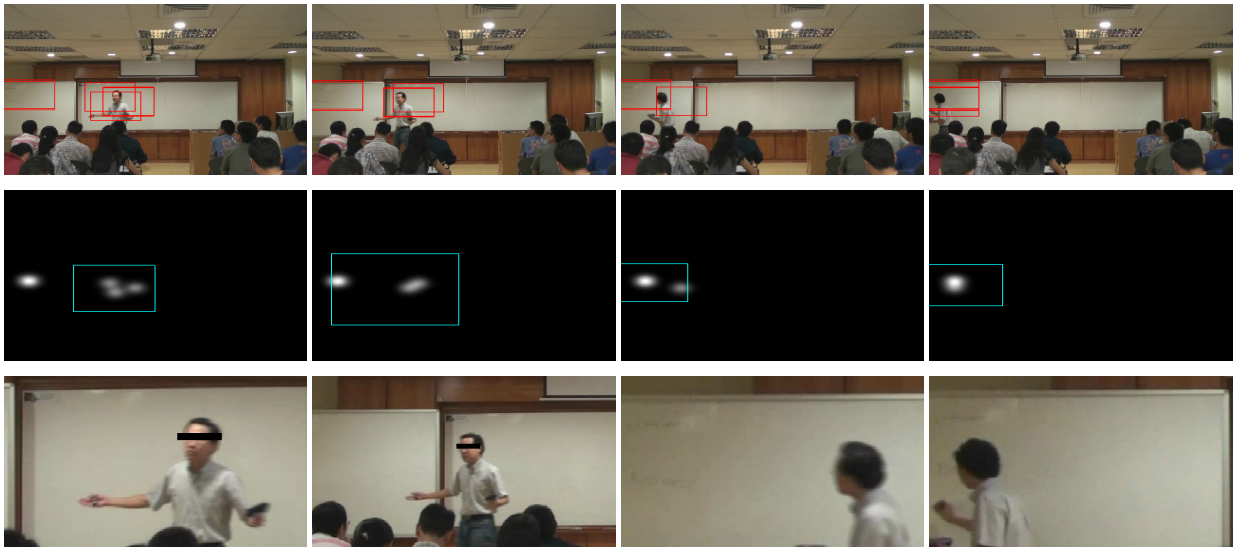

Figure 2: Approach overview: four frames and a few viewports (first row), heatmaps and detected ROIs (second row), retargeted frames including reframing techniques (last row). (Note that the lecturer's eye are redacted for anonymous review).

search community in automatic ROI extraction, we believe that, in complex scenes, extracting the correct ROI becomes difficult. As such, we have turned to an alternative paradigm in this paper.

The main idea behind our method is to identify the ROI through the wisdom of the crowds. Since users are the final consumers of the content, they are naturally the best possible "ROI detectors". To understand our technique, however, we need to first explain how we crowd-source the ROI selections from the users.

We have previously developed a web-based video player that allows users to zoom in and pan on a high definition video sequence captured with a static camera and viewed in a small view window [1]. We call our player a *zoomable video player*. Figure 1 shows the screenshot of the interface to the player, which consists of a video display of size 320×180 (top), a thumbnail display of size 160×90 (lower left corner), and control buttons (lower right corner). Users can zoom in and out of any region within the video to view the chosen region at higher resolution. Zooming is equivalent to viewing a sub-region of the video through a *viewport*. Pixels from the high definition video that fall within the viewport are scaled and shown in the video display. A viewport example is illustrated as the highlighted rectangle in the thumbnail display of Figure 1. Users can pan within the video at the same zoom level to move the viewport to a different region.

The player logs the viewport, i.e., what the user chooses to watch, on every frame. We assume that most users watch the most interesting and relevant regions within each frame. Thus, these historical viewport traces become good predictors of the most interesting regions in every frame.

This paper presents how crowd-sourcing of users' interaction with a zoomable video can be used to automatically produce a retargeted video on small devices. The overview of our approach is presented in Section 2. We then take the readers through the process in steps. We describe the methodology used to log zoom/pan interactions through our web-based player in Section 3. Section 4 explains how we identify ROIs from the log data in each frame. Section 5 presents how ROIs of consecutive frames are grouped together into shots (a shot is a sequence of frames without cuts). Concatenating the shots together, however, may not lead to a visually pleasing video. We thus apply reframing techniques to improve the visual aesthetic of the video through automated editing. The automated editing is described in Section 6. We evaluate the resulting video through a user study (Section 7) and discuss the related work in Section 8. Finally, we conclude the paper in Section 9.

2. APPROACH OVERVIEW

Figure 2 illustrates our approach to retarget a video using a short lecture video sequence, which contains three subsequences: the speaker talks while written text is visible on the left whiteboard; the speaker walks to the left; the speaker writes more on the board. The top row of Figure 2 shows the source video, which is a high definition video shot with a fixed camera located at the back of the classroom. The video sequence was presented to users through our zoomable video player introduced in Section 1.

We use red rectangles to denote viewports selected by users. We overlay the viewports on top of the source video (first row of Figure 2) to illustrate what the users choose to zoom into when watching this video sequence. One can notice that some users stay focused on the whiteboard, while others follow the speaker's motion. Two users can be clearly distinguished in the beginning of the video sequence (first column). By the end of the sequence, all users concentrate on the board (last column).

Viewports selected by users in each viewing session are stored in a database; viewports selected by many users naturally cluster around ROI within the video. Based on the viewport traces collected, we construct heat maps that express the cumulated user interest for each frame (Section 4.1). On a heat map, interests are represented with brightness of the pixels – the brighter the color is, the more user interest there is. We call these patches of bright spots the *hotspots*. Second row of Figure 2 shows the corresponding heat maps of the lecture video sequence. In this example, some hotspots move along with the speaker, while others do not.

To retarget a video, we need to determine the best region of interest from multiple hotspots that exist in each frame. Note that this cannot be done simply by selecting the hotspots with the most user interest. Multiple hotspots of similar interest level might be located

close by, in which case perhaps the ROI should include all these hotspots. To determine the ROI, we model the hotspots on a frame-by-frame basis using Gaussian Mixture Models and use Minimum Covariance Determinant to find a region with maximum amount of interest with smallest area (Section 4.2). In this paper, we use blue rectangles to denote ROIs. In the second row of Figure 2, we show the ROIs of each frame. Note that a ROI contains multiple hotspots in the first three shots of the figure.

After identifying the ROI of each frame, we model the dynamics of the ROI across time using a graph that links ROIs between consecutive frames (Section 5.1). We segment the graph (section 5.2) to group the ROIs into shots based on the distance between the ROIs. At this point, a ROI may belong to multiple shots. However, the next step decides which shot a ROI belongs to in the final retargeted video. Reframing rules are used to improve the visual aesthetics of the video. ROI movements across frames are dampened to smooth out movements, and reestablishing shots are added to help the viewer in understanding the context of the shots (Section 6). The third row of Figure 2 shows an example of the resulting retargeted video produced automatically using our steps above.

3. CROWD-SOURCING VIA ZOOMABLE VIDEO PLAYER

We now describe in more details how we collected traces of viewports selected by users via our zoomable video player.

Figure 1 shows the screenshot of the player. A video display of size 320 $\times$ 180 can be found on top. The small view display simulates the scenario where a video is watched under resource constraints (low bandwidth, or small screen size, or low decoding capability). The lower left corner of the interface contains a thumbnail window of size 160×90. The thumbnail always display a scale down version of the source video. The viewport is shown in white whenever users zoom in, to provide the users with the context. The lower right corner shows some control buttons where the users can click to zoom and pan, as an alternative to using the mouse.

Video sequences in our system are of a HD resolution (1920 $\times$ 1080). By default (zoom level 0), the video is scaled down for playback in the video display. At this zoom level, the user can view the whole video, although with a lower level of detail. Five levels of zoom are supported (levels 1 to 5). Viewing a region at a higher zoom level is equivalent to viewing a cropped region of size 320 $\times$ 180 from a higher resolution version of the video. For instance, at zoom level 5, users see a 320 $\times$ 180 region from the original source video. At zoom level 4, users see a 640 $\times$ 360 region from the original HD video scaled down to fit the 320 $\times$ 180 window. Users can use either the $+$ or $-$ buttons or the scroll wheel on the mouse to zoom in and out.

Users, after zooming in, can drag the mouse on the video display to pan. They may also use the arrow buttons for finer grain control of the panning. Users can pan anywhere within the content of the original HD video.

Despite using user interfaces that are similar to those used by other zooming interfaces (e.g., Google Earth), we found in our previous deployment that many users tend to play around with the different possibilities of the user interface. To prevent this, we have asked the users that use our system to watch an instructional video, showing them on how to use the interface. After watching the video, we force our users to go through a small practice session, in which they are asked to complete a step-by-step tutorial on how to use the user interface. The practice session must be completed before users can start watching any zoomable video.

The system logs all mouse and keyboard operations performed by the users, along with their timestamp, when they watch the zoomable video (the interactions from practice sessions are not logged). From these logs, one can replay the users' zoom and pan action, as well as determine the viewport of each user on each frame.

4. REGION OF INTEREST MODELING

We now describe how we extract ROIs from the logs of users' interactions with a video. We construct a heat map (or user interest map) by aggregating the selected viewing regions from multiple viewers. The heat map might yield multiple hotspots – regions that are popularly viewed by the users. The challenge now is to extract an ROI based on these hotspots. Note that simply taking an ROI that covers all hotspots might not give the best solution, since it might cause the retargeted video to zoom out, while the interesting regions are in the details. Taking an ROI that covers only the most popular hotspots does not work either. There could be other almost equally popular hotspots nearby that could have been included in the ROI. To produce a good ROI, we thus need to take both the popularity and size of the ROI into consideration, ideally producing a smallest ROI with highest popularity. We elaborate on our approach in the rest of this section.

4.1 Heat map construction

During playback, for any frame, each user chooses a rectangular viewing window (i.e., viewport) via zoom and pan. We observed that users naturally center the window (e.g. the focus) on their preferred regions. Each pixel from a viewing window does not have the same interest level. The closer a pixel is to the window's center, the higher should its contribution to the user interest map be. In line with previous work [8], we model the pixel-wise interest levels using gaussian probability density functions (pdf). Figure 3 illustrates our method. A viewing window centered on $\mu = (u, v)^T$ is shown superposed on a frame. We also plot the gaussian pdf with the same center and a covariance matrix Σ that fits well with the viewing window. Theoretically the matrix Σ is chosen such that the ellipse whose equation is $d_M(\mathbf{x}, \mu)^2 = (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) = 13.8$ is tangent to the viewing window VW[1]. As a consequence, we have a gaussian weight at each pixel and a gaussian elliptical footprint for the associated viewing window (Figure 3). Now we have to aggregate the data from multiple viewers to produce our user interest map. We simply blend the footprints by accumulating gaussian weights and normalize the result by dividing it by the number of windows involved for each pixel.

Middle row of Figure 2 shows typical heat maps: the dark areas represent less popular regions in the video frame, whereas brighter areas have cumulated many votes and highlight preferred regions of interest. The clustering of hotspots indicates that users tend to agree on the most interesting regions, with only a slight variations on zoom factor and focus. Less important ROIs may differ from viewer to viewer. Another heat map example is given by Figure 4, extracted from another test video (*dice tricks*). At this stage the reader should ignore the green annotations, to be explained later.

4.2 Modeling ROIs as a GMM

The next step is to extract ROIs. There are many possible techniques to detect and model ROIs of a heat map. Huang et al. [10] cite three main approaches: a simple binary mask, a set of focus points (FOA - focus of attention), an importance map viewed as a probability density function using a mixture distribution. We favor

[1]The threshold comes from the Mahalanobis distance $d_M^2 \sim \chi_2^2$ such that $\forall \mathbf{x} \in \text{VW} \;\; P\{d_M(\mathbf{x}, \mu)^2) \leq 13.8\} \geq 0.999$

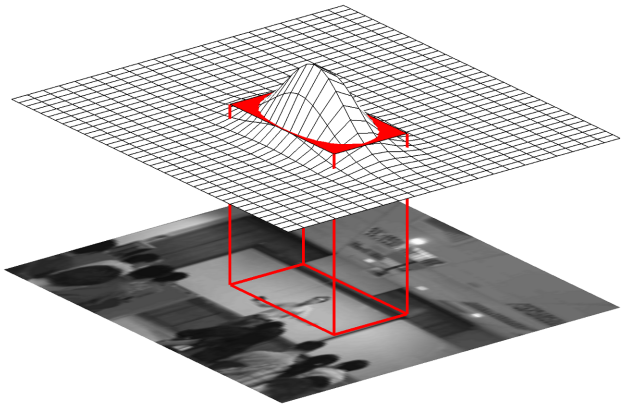

Figure 3: Gaussian weights within a viewing window

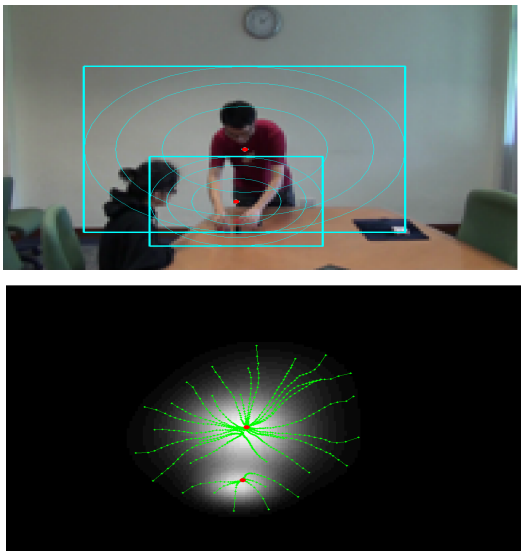

Figure 4: Hotspots for the magic tricks video (top) and multiple starts of Mean Shift, converging to two modes (bottom)

the latter in our video retargeting application: in section 5.1 we will see that modeling the ROI dynamics using a mixture model proves to be rather simple. We basically aim at modeling the heat map at frame t as a Gaussian Mixture Model (GMM) involving K ROIs. We assume that the interest location variable $\mathbf{x} \in \mathbb{R}^2$ leading to a heat map follows a GMM pdf:

$$p(\mathbf{x}|\theta_t) = \sum_{j=1}^{K} \omega_{t,j} \quad p(\mathbf{x}|\mu_{t,j}, \Sigma_{t,j}) \tag{1}$$

where $\omega_{t,j}$ are the relative importance weights of the K ROIs considered at frame t and subject to $\omega_{t,j} > 0$ and $\sum_{j=1}^{K} \omega_{t,j} = 1$. Each ROI's density is a normal probability distribution :

$$p(\mathbf{x}|\mu_{t,j}, \Sigma_{t,j}) = \frac{\exp\left[-\frac{1}{2}(\mathbf{x}-\mu_{t,j})^T {\Sigma_{t,j}}^{-1}(\mathbf{x}-\mu_{t,j})\right]}{(2\pi)\,|\Sigma_{t,j}|^{1/2}} \tag{2}$$

where $\mu_{t,j}$ and $\Sigma_{t,j}$ are the moments of the j^{th} gaussian ROI at frame t. The global set of parameters for frame t is $\theta_t = \{\omega_{t,1} \ldots \omega_{t,K}, \mu_{t,1} \ldots \mu_{t,K}, \Sigma_{t,1} \ldots \Sigma_{t,K}\}$. One may think that it is straightforward to estimate θ_t by EM (Expectation Maximization) but a major question that remains is the correct value for K (the number of ROIs). This can be formally expressed as a model selection problem [10] but we actually show that a non parametric clustering technique and a smart method for the covariance matrix estimation fits better our need.

4.3 Parameters estimation and ROI detection

We use the *Mean Shift Clustering* technique [3] to automatically find the K modes (peaks of the heat map) and to subsequently assign pixel locations to the associated K clusters. The basic *Mean Shift* procedure starts from a pixel point and converges towards a stationary point of the density function (see the green traces corresponding to multiple starts on the figure 4 (bottom)). The set of all pixels that converge to the same mode defines a cluster. The method does not require prior knowledge except one parameter: the kernel bandwidth. A value of $\xi_d = 90$ pixels was selected since it is the right distance to separate significantly different viewing windows in full HD images.

Once we have the modes $\mu_{t,j}$, we also need good covariance matrices that produce ROIs well centered around the modes and small enough (in order to get a good attention window). We estimate each $\Sigma_{t,j}$ with the Minimum Covariance Determinant (MCD) estimator. This estimator is basically a robust estimator that allows to cope with outlying or spread data. Its principle consists in finding a covariance matrix able to capture a maximum amount of interest while having the smallest area. As previously mentioned, the area is proportional to the covariance determinant that is minimized by our MCD estimation for $\Sigma_{t,j}$. Moreover, the weigths $\omega_{t,j}$ are taken proportional to the clusters' sizes. The GMM parameters set θ_t is now fully defined. Figure 4 (top) shows two gaussians centered on the two modes (red points). Each gaussian is represented by a set of iso-values, the biggest one being tangent to the detected rectangular ROI.

5. GENERATING SHOTS

Thanks to our GMM-based models, ROI can be detected for each frame. The next question is how to deal with the temporal dimension. In this section, we first model the ROI dynamics. Then we produce shots highlighting the most preferred areas. Finally, these shots will be enhanced (section 6) using reframing techniques in the last stage of our approach.

5.1 Graph-based dynamics

Regions of interest are different from frame to frame. The viewers usually follow moving attentional objects or focus on particular areas due to their semantic content. Let us consider the magic trick video (figure 4): at a given moment the viewer is orally encouraged to pay attention to a dice or to a card in order to understand the trick. We observed that most of the viewers actually follow such instructions and zoom into the expected visual area. Such a scenario naturally leads to time-varying ROIs both in position and shape. Additionally, split, merge and delete occur within our set of gaussian components as a natural consequence of ROI splitting, merging and disappearing.

In order to model all these variations, we use a graph-based approach. Figure 5 (top row) shows the beginning of a video sequence where the j^{th} detected ROI at frame t is a graph node denoted $\langle t, j \rangle$. Only five frames are shown and there are only two ROIs by frame except for the 4^{th} and a 5^{th} where a third one appears.

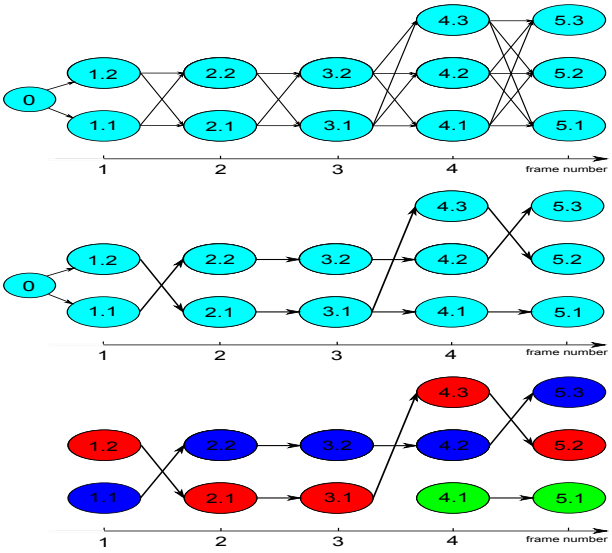

Figure 5: ROI dynamics graphs: full graph (top), MST graph (middle) and MST after cuts (bottom)

Since we will later need a tree structure for the graph, we introduce a virtual frame 0. More precisely, our set of ROIs is represented as a weighted directed graph where an edge is formed between every pair of ROI in subsequent frames (eg. $[\langle t,j\rangle, \langle t+1,i\rangle]$). The edge set is noted E. The weight w_d of an edge between two ROIs is initially set as the euclidean distance between their modes: $w_d(\langle t,j\rangle, \langle t+1,i\rangle)$ is the distance between the ROI $\langle t,j\rangle$ (located on mode $\mu_{t,j}$) and the ROI $\langle t+1,i\rangle$ (located on $\mu_{t+1,i}$):

$$w_d(\langle t,j\rangle, \langle t+1,i\rangle) = ||\mu_{t,j} - \mu_{t+1,i}||. \quad (3)$$

Moreover, we add a few more attributes to label the ROI nodes and to assess the ROI variations. Each ROI node $\langle t,j\rangle$ can be labeled with the *associated weight* $\omega_{t,j}$. The *weight variation* between two subsequent ROIs can also be measured by :

$$w_\omega(\langle t,j\rangle, \langle t+1,i\rangle) = |\omega_{t,j} - \omega_{t+1,i}|. \quad (4)$$

Similarly, each node $\langle t,j\rangle$ can be associated with the *area* associated with the covariance matrix $\pi\sqrt{\det(\Sigma_{t,j})}$. The *area variation* between two subsequent ROIs can be evaluated by :

$$w_a(\langle t,j\rangle, \langle t+1,i\rangle) = \pi.|\sqrt{\det(\Sigma_{t,j})} - \sqrt{\det(\Sigma_{t+1,i})}|. \quad (5)$$

5.2 Shot segmentation

In order to group related and subsequent ROIs into shots[2], we seek a partition of the set of nodes where some consistency measure in a subset is high. Our approach consists in two steps :

- compute a Minimum Spanning Tree according to w_d weights,
- cut some edges to produce consistent candidate shots.

A Minimum Spanning Tree (MST) is an acyclic subset T of edges selected from the edge set E of the initial graph. For now, let us consider that our graph is undirected. The MST edges T connect all the ROI nodes such that their total weight is minimum :

$$w_d(T) = \sum_{[\langle t,j\rangle, \langle t+1,i\rangle]\in T} w_d(\langle t,j\rangle, \langle t+1,i\rangle). \quad (6)$$

[2]A shot is a single stream of images, uninterrupted by editing.

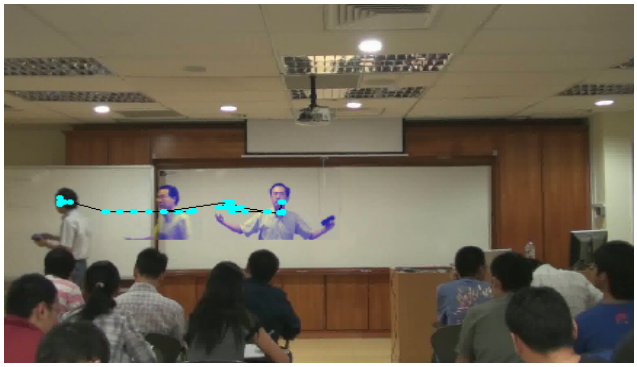

Figure 6: Minimum Spanning Tree (MST)

The second row of figure 5 presents an MST of the full graph. Removed edges are those with the highest weight values. To get it, we apply a simplified version of Prim's algorithm that takes the particular structure of our graph into account (no edge between ROIs at the same frame). Figure 6 also presents an MST on our lecture video. The figure suggests the motion of the speaker walking leftwards in the scene by using ghost effects. All the nodes of the graph of ROIs are plotted in cyan. The MST edges are displayed in black (same color coding as the one used in figure 5). Long edges can be observed on figure 6 when the speaker moves quickly leftwards. Long edges are characterized by an important w_d weight.

The middle graph of figure 5 (MST) can be interpreted as follows. One moving attentional object is present in the frames 1-5. Our ROIs (MST nodes) are able to track it along the tree. For example, the object is tracked by ROIs $\langle 1.2\rangle$, $\langle 2.1\rangle$, $\langle 3.1\rangle$, $\langle 4.3\rangle$ and $\langle 5.2\rangle$ from which we deduce a single stream of cropped images (focused on the object): a tracking *shot*. In order to produce it automatically, the system has to decide to cut the edge between node $\langle 3.1\rangle \rightarrow \langle 4.1\rangle$. In other words, producing shots can be viewed as splitting the MST tree.

A tree can be cut into two disjoint set by simply removing an edge. The third row of figure 5 shows a MST after the cuts. Once again, removed edges are those with the highest weight. Then we form three shots (in red, blue and green). Similarily, figure 7 presents the MST of figure 6 after the cuts. In our application we cut the edges with large w_d weights. The threshold we use is also the bandwidth parameter of Mean-Shift clustering technique that aims at distinguishing separate ROIs:

$$\text{Cut}[\langle t,j\rangle, \langle t+1,i\rangle] \text{ if } w_d(\langle t,j\rangle, \langle t+1,i\rangle) > \xi_d \quad (7)$$

Figure 7 presents the 5 shots we obtained from our lecture video. Shot 1, 3 and 4 present the teacher speaking in the middle of the classroom in the first part of the sequence. Shot 2 is focused on the whiteboard. Shot 5 tracks the speaker motion. Figure 8 shows the shots on the timeline.

As a refinement, for each shot, we also identify a few candidate positions for possible new cuts to be used in the next step. For instance, shot 1 can be further refined at frame 37 and 51. These possible new cuts are identified by high values of $w_a(\langle t,j\rangle, \langle t+1,i\rangle)$ or $w_\omega(\langle t,j\rangle, \langle t+1,i\rangle)$.

5.3 Compositing Shots

The final result we obtain from the graph is a sequence of shots to be edited. For each temporal interval between two possible cuts and among possible shots, we select the best one given a selection criterion. In figure 8, the considered intervals are $[1,6]$, $[6,37]$ etc.

Figure 7: Shots after the MST cuts

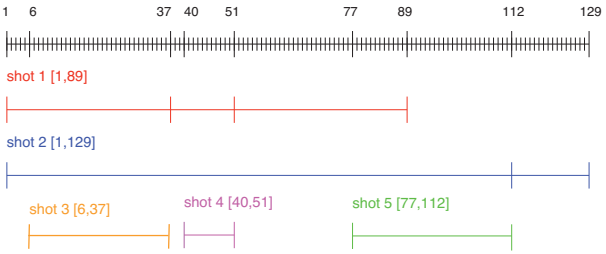

Figure 8: Timeline [1,129] and possible shots

Each interval may be covered by several shots. The simplest selection criterion may be the average weight of the shot divided by its duration. In other words, this strategy selects the most popular shots. Other more complex strategies may be imagined. For example, the selection may favor shots with higher motion.

6. REFRAMING TECHNIQUES

We now have a sequence of shots that represents a reframed version of our input video. The quality of this video is suboptimal. User studies (section 7) show that this automatically produced version is not pleasant to watch since it is too shaky. This is mainly due to annoying visual jumps coming from translational and scale noise.

Therefore we use reframing techniques aiming at (1) stabilizing shots to correct for this noise (lack of aesthetics) and (2) reestablishing shots to establish the scene context before moving to close shots (loss of important context information).

6.1 Simple Guidelines

Before detailing our reframing techniques, let us introduce some good editing practices that we followed. We start by systematically keeping the initial aspect ratio (16/9) for the generated framing. Shots may be produced at different scales. An *extreme long shot* (ELS) is a framing close to the initial full HD format (ranging from 1920×1080 to 1600×900). The *long shot* (LS) is a smaller framing (width from 1600 to 1100) useful to make a rather stable shot that can easily cover movement without reframing. The *medium shot* (MS) is the most popular and its width ranges from 1100 to 700. Finally, *close-up* (CU: width from 700 to 400) and *extreme close-up* (ECU: width < 400) are smaller framings for showing details. Regarding the duration of the shots, we do not impose strong constraints since the decision to extend a shot can be as effective as the decision to cut it. We simply ensure that handled shots include at least 10 frames. Generally speaking we try to produce a retargeted video that maintains a clear and continuous action. Therefore we follow the general style of *continuity editing* [2].

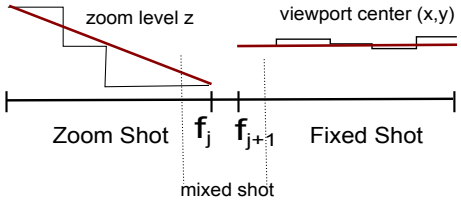

Figure 9: Reframing on two shots

6.2 Bottom-up Reframing

At the *bottom* level (individual shot-level) we stabilize each shot to ensure its spatial continuity. This stabilization is a two-step process. First we classify the shot into *fixed*, *zooming* or *dolly* types of shots. In a fixed shot, the focus does not change. A zooming shot results only from an optical zoom into an object. In a dolly shot, the camera is moving as if it was mounted on a train track, parallel to the scene (pan-like motion). Some *mixed* shots exhibit combinations of these types.

In the second step, each shot is stabilized according to its type. For example, in the zoom shot of figure 9, the zoom level of each frame of a zoom shot is interpolated using the values of the first and last frame. For a detected fixed shot, since there are small variations of the focus, we modify the center of each frame using the average center.

Once we stabilize each shot individually, we work at the *upper* level (inter-shot level) in order to smooth out the transitions between shots. In our example, if there is a discontinuity of the viewport center of frames f_j and f_{j+1}, we create an additional short *mixed* shot that interpolates both the zoom and position of the center.

Additionally, also at the upper level, we apply another technique called *reestablishing shot* (RS). A RS is often a medium or medium-long shot. It usually follows a close-up and is used to help the viewer better understand the context. In other words, it reminds the audience the position of the viewport inside the scene. The figure 2 (third row) shows a RS. The first shot shows a close-up on the speaker and the third is a close-up on a very different ROI (the left board). Our system automatically produced a RS in between, upon the detection of a ROI (the left hotspot of the third shot, representing the left board) that was not visible during the first shot (left hotspot). Therefore the user is presented with a *medium* RS before the second close-up.

7. RESULTS

To evaluate our proposed crowd-sourced video retargeting, we posted several videos for users to watch through our zoomable video player (see figure 1). The videos are recorded in high definition using a fixed camera. The video sequences posted has a range of different content, ranging from lecture, sports, and magic tricks. Within a period of two weeks, we collected traces from 53 user sessions (one session for each viewed video). A total of 11183 interaction events are logged.

Retargeted versions of the video are created using the algorithms described in the previous sections. In this section, we present results for three video sequences. The first two are magic tricks

videos, which we refer to as the *card trick* and the *dice trick* video respectively. The third video sequence is a video of a rhythmic gymnastics movement (introduced by figure 1 and 11), which we refer to as the *gymnastic* video. We first describe the retargeted videos in Section 7.1 and compare our results with saliency-based methods. An analysis of the different shot types, along with their distributions is done in section 7.2. We show that the distribution of shot types in the retargeted video is consistent with those of the collected traces. Then, we carried out a user study to assess the quality of the retargeted shots (7.3).

7.1 Retargeted Video

We now describe two sample videos that our method produced. We qualitatively compare our results with results generated by using visual attention models. Resulting videos are available at `http://www.youtube.com/user/AutoZap`.

Figure 12 shows a selected set of original (HD) video frames and their corresponding retargeted frames from the *dice trick* video. In this trick, the magician is challenged to find the correct dice number. He gives instructions to the girl: put the dice in the black box (2 first rows), show the dice number to the camera (rows 3-5), close the box and put it down on the table (row 6). He returns to the table (row 7) and continues (row 8-9). One should notice the close-up on the dice number (row 5). Another relevant close-up (row 8) emerged and focused on the box: the critical object of interest for understanding the trick. Just after, the reestablishing shot helps to put the user into context for the rest of the trick.

We first highlight some shots of interest region in our video that is hard to be reproduced by existing approaches. The close-up shot on the dice number (row 5) in the retargeted video sequence is synchronized with the instructions of the magician ("show the dice to the audience"). Doing this automatically with content-based approach would require speech recognition to understand what the magician has said and object recognition to identify the dice within the scene. Both are hard problems that are challenging to solve. This example stresses the importance of leveraging user behavior to generate retargeted shots, since users naturally follows the instruction of the magician and zoom in to see the dice.

Figure 10 shows an example of salient regions in the *dice trick* video, computed using Matlab SaliencyToolbox[17], which is based on visual attention model, particularly, color, intensity, and orientation. The left shows a frame from the video with salient regions enclosed in yellow. The right shows the heatmap indicating the salient regions. This example shows that the most salient region in the video, according to the visual attention model is the back of the magician, followed by the hand, a deck of cards on the table (which is not even used in this trick), and finally the dice. This example illustrate that visual attention model is not sufficient for detecting interest regions in the video sequence.

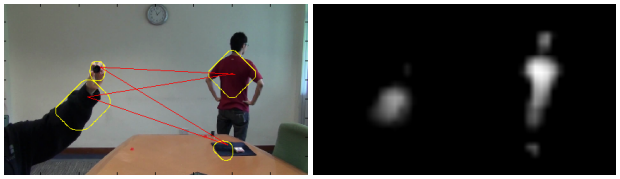

Figure 10: Outputs from the saliency toolbox.

The interest regions detected using visual attention model on the *gymnastic* video is shown in Figure 11 to further illustrate the efficacy of our approach. The *gymnastic* video is a wide angle shot of a gymnastic contest, with five gymnasts performing on the floor, surrounded by audience, judges, and officials. The top image shows the results obtained by the SaliencyToolbox. The salient regions detected include doors, windows, notice boards, posters, and some audience. The most interesting part of this video, however, is the gymnasts. While the SaliencyToolbox managed to detect the gymnasts as salient regions, this detection is only possible after the option which gives skin colors more weight is turned on. On the other hand, our method correctly detect the gymnasts as the ROI (the middle figure). The bottom figure shows another frame from the same sequence, with moving regions highlighted, as detected using the blob tracking algorithm in OpenCV. Visual attention model that assumes motion as a salient feature would have overestimated the number of interesting regions in the frame. In this example, besides the gymnasts, motions of audience and other gymnasts standing-by are also detected.

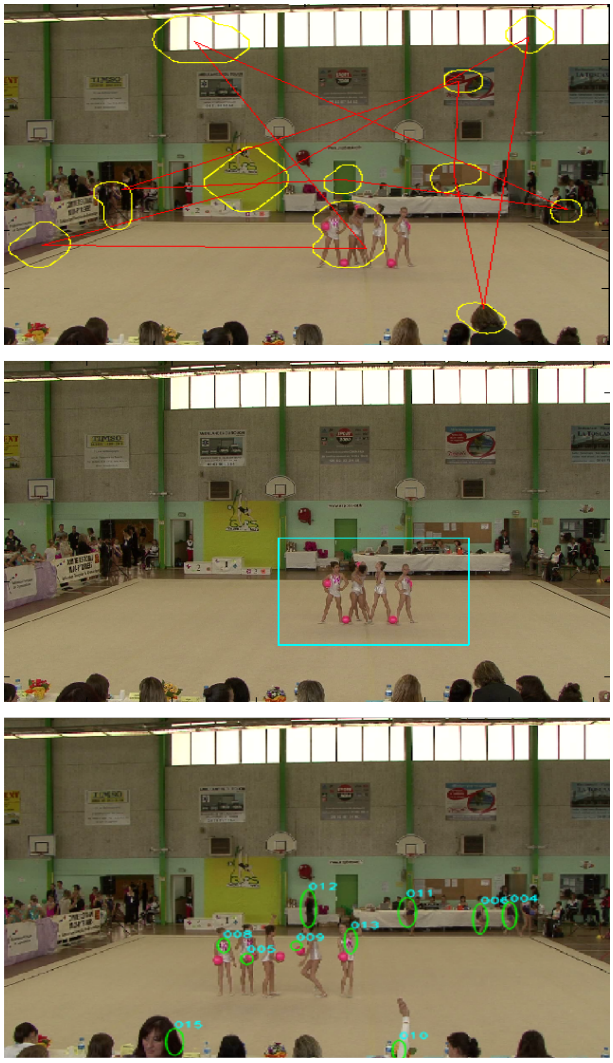

Figure 11: Visual Attention Models vs. our ROI

7.2 Properties of Retargeted Shots

Having described two of our retargeted video sequences, we now present some of their properties. Table 1 summarizes the distribu-

tion of shot types as captured by our logs. Shot types are classified into classes, ranging from *extreme long shot* (ELS) to *extreme close-up* (ECU), as introduced in Section 6.1. The variety of observed shot types is a clear indication that users effectively used the zoom and scroll video interface for a better viewing experience. Shot distributions are not very similar among the three video sequences, since they are strongly related to the content. The *Card Trick* exhibits numerous *close-ups* (CU) and *extreme close-up* (ECU), since the user is explicitly and orally encouraged to gaze at playing cards. These results confirm the importance of the semantic of the contents. This statement is also verified by a detailed analysis of the *gymnastic* video: users who were relatives of one of the gymnasts tended to track her by using close-ups.

shot	ELS	LS	MS	CU	ECU
Dice Trick	17%	26%	29%	19%	9%
Card Trick	13%	25%	13%	24%	25%
Gymnastic	8%	23%	26%	31%	12%

Table 1: Statistics on shot types

Table 2 compares various versions of retargeted shots for the dice video sequence. The first row ("User Traces") shows the previously discussed shot types ratios (in %). The following rows present shot distributions for various versions of our retargeted video. The second row ("Final") is our final retargeted version that includes reframing techniques (Section 6) and the MCD estimator (see Section 4.3). The third row ("No RT") is the retargeted version without reframing techniques produced with MCD. The fourth row ("ML") corresponds to an alternative version without reframing techniques produced with Maximum Likelihood (ML) covariance estimator instead of MCD.

The shot distributions of various versions provide some insights into how our method works. First, we see that the shot distribution for "Final" and "User Traces" are slightly different. Less ELS exist for "Final" due to the retargeting that favors LS and MS. Indeed, the retargeted version shows that for the dice video, all ELS shots can be effectively replaced by smaller shots. We will see in the next section that the user studies proved the effectiveness of this editing, since most of them are satisfied with the quality. The number of ECUs doubles. Second, using ML (commonly used in GMMs models), tend to produce bigger shots. This result justifies our preference of MCD over ML for estimating the covariance matrices. The final retargeted *dice trick* includes more CUs and ECUs since it focuses on a small object of interest (the dice), which are not found if ML is used. A remarkable result is 54% LS, $27\% = 13 + 14$ close-ups of the "No RT" version. We can understand the cause by watching the produced video: the focus changes rapidly from close-ups to the LS. This is smoothed out by the "Final' version around MS thanks to reframing techniques.

Shots from	ELS	LS	MS	CU	ECU
User Traces	17%	26%	29%	19%	9%
Final	0%	32%	38%	12%	18%
No RT	1%	54%	18%	13%	14%
ML	27%	32%	20%	19%	2%

Table 2: Statistics on shot types for different retargetings.

7.3 User Evaluations

Recall that the goal of video retargeting is to produce a visually pleasant, yet informative video for a small display. To assess if our results have achieved the two goals of video retargeting, we conducted a user study, comparing retargeted video from our method with two other video sequences produced from the same video.

Video Sequences. We choose the *dice trick* video as the clip for user study. As a ground truth, we compare our retargeted video (denoted crowdsourced) to a version of retargeted video produced by an expert user (denoted expert). This ground truth retargeted video is produced by having the expert, who is well aware of the video content and user interface, to watch the video using our zoomable video player. His choices of viewports are used as input to produce the ground truth. On the opposite, we use either one of unretargeted video (i.e., always in zoom level 0, denoted nozoom), a retargeted video generated from a chosen user trace (user), and a retargeted video produced after Section 5 without reframing technique (noRT). The chosen user trace is a typical trace (zooming and panning to regions similar to our retargeted video), except for a short time when the user is not able to find the dice and has to zoom out and in again to refocus on the dice in the video.

Methodology. We show three videos in random order to each participant. The expert and crowdsourced versions are always shown, whereas the third video is randomly chosen from one of the remaining: nozoom, user or noRT. Users are asked to watch the video clips and to rate the editing quality of the video. A user may watch the video as many times as he wants. We assess the quality of the retargeted video by asking our participants the following questions: (i) Is the editing quality of each video reasonable? (ii) Does the video manage to convey the important information to understand the events in the video? The participants are asked to rate each video in a rating of 1 (poorest) to 5 (best), as well as to make qualitative comments on the video.

Participants. The user study involves 48 participants (28 Male, 20 Female). 16% of the participants have a prior experience in video editing or cinematography. 18 participants are presented noRT and user versions, 12 are presented nozoom.

Results. Respectively 87% and 79% of the participants find the editing quality of expert and crowdsourced reasonable. Only 21% of users find the third video reasonable. Among the possible third video, nozoom is found to be most reasonable (41%), while noRT is the least reasonable (5%). 22% of users find the user video reasonable.

The expert and crowdsourced versions of the retargeted video have an average rating of 3.6 each. The rating for the third videos averaged to be only 2.35 (2.11 for noRT, 2.33 for user, 2.75 for nozoom). There is no significant differences between the ratings made by users with experience in video editing and cinematography. These users rated expert 3.5, crowdsourced 3.3, and the third sets of video 2 on average.

When asked if the video managed to convey useful information, expert scores the highest with affirmative answers from 79% of users, followed closely by crowdsourced. Interestingly, 66% of the participants who watched user thinks that it still manages to convey useful information. Surprisingly, none of the 12 users who watch nozoom found it useful. The last two results validate the importance of retargeting, and validates the usefulness of user traces in choosing important regions to watch, even when it is done by only one single user. Only 38% of users found that noRT managed to be useful.

The results above shows that our crowdsourced version is only slightly worse than the video retargeted manually by an expert user.

8. RELATED WORK

ROI Detection. ROI detection has many useful applications for multimedia content, including ROI coding, progressive content

transmission, and image/video retargeting. Many ROI extraction models have been proposed to automatically detect ROIs. They fall into two main categories: content-based and usage-based. Content-based approaches infer ROIs from the intrinsic properties of the content, such as color, motion, and semantic. Usage-based approaches compute ROI from user feedback, whether implicit or explicit.

In the first case, visual attention models can be used to measure the salience of important regions. In their influential work, Itti et al. [11] detect ROIs with a saliency map computed from pyramid maps for color, luminance, and orientation contrasts. Multimedia content analysis may also help to estimate the ROI. In image retargeting, face and text detection naturally leads to potential ROI [13]. More generally, object-based or event-based segmentation can be used to infer ROIs by decomposing a visual content [14, 15]. Our approach in this paper does not rely on computational model of attention, but rather, we rely on traces from user behavior, assuming that the collective behavior of users naturally guide us to the ROI. Our approach is thus an implicit usage-based approach.

Other usage-based approaches have been proposed in the literature. The first basic idea is to extract ROIs by directly analyzing regions gazed by a user. In their work, [16] Ukita et al. use an eye-mark recorder system to collect gaze points before extracting static or moving regions of interest. Inspired by recent success of collaborative games, Huang et al [10] developed an online game called Photoshoot for extracting image ROIs from shoot points playing the role of gaze points. User interactions while browsing videos have also been used to produce video previews. These previews are composed of remarkable segments of the video automatically inferred from the analysis of the browsing logs. Xie et al [18] have studied mobile image browsers. They found that on small displays, users tend to use more zooming and scrolling actions in order to view interesting regions in detail. From this fact, they designed a specific ROI extraction model from still images and so-called user interest maps. Our work also extracts ROI from browsing logs but we additionally tackle *video* retargeting and model both the important regions and their dynamics.

Video Retargeting. To automatically retarget a video, Liu et al. [13] analyzes a candidate set of cropping windows and chooses an optimal cropping window that minimizes a distortion function to crop the video before it is scaled. The cropping window can change with motion in the video and is therefore adaptive. In the context of ROI detection and tracking for stored video playback, there have been attempts to model the ROI [6] based on the amount of motion. Such models could help determine the ROI, without human interaction with a display device. Multi-scale cropping [5] dealt with automated tracking of multiple ROIs defined by motion change. The aim was to minimize the number of ROI trajectories within the video while covering all the ROIs.

To retarget general movies, careful handling is required for camera motion: the initial cinematography must be preserved and motion artifacts must be avoided [13]. With our videos captured from a fixed HD camera, aesthetic reframing and transitions are however required. Cinematography is an art and only informal rules are described to film various scenes [2]. These informal rules rather led to more heuristic than formal models in computer science. For instance, researchers in virtual reality and game design employed cinematographic rules for real-time positioning of the virtual camera [9]. Automating the film-making process for computer animation with a virtual cinematography system [12] is also a challenge. Declarative language like Declarative Camera Control Language have been devised for that purpose. Formal attempts were made by MPEG7 standard (ISO/IEC 15938-5:2003) to specify tools for describing video editing segment, shots and different types of transition. In a computer vision context, Doubek et al. [4] also explore the use of some basic cinematographic rules for selecting the best view available in a camera network. They then crop the selected views with a virtual zoom and interpolate novel views to finally produce one attractive video stream from many coming simultaneously from the network. Automatic video retargeting can benefit from cinematography. Gleicher et al. [7] propose to improve apparent camera movements in order to better follow cinematographic conventions. For example, they use virtual pans to better show moving objects.

9. CONCLUSION

This paper proposes an automated approach to retarget a high-resolution video for display on small screens by exploiting collective users wisdom through a zoomable video player. Our approach relies on historical user traces to select regions that effectively convey the message of the video and cinematography rules to improve the visual quality of the retargeted video. Our usage-based approach is a natural complement to previous retargeting method based on content analysis. Our approach is able to correctly identify good ROIs in scenarios that is difficult for content-based approach, such as understanding voice instructions or identifying person-of-interest in a crowded video. A user study with 48 participants shows that the retargeted video produced by hand by an expert is only slightly better than those produced automatically using our approach.

Several directions can be taken to continue this research. We plan to investigate the possibility of including split-screen effects, possibly showing two or more ROIs at once automatically. An example where this is useful is in our lecture video. One screen could show the lecturer, while another shows the whiteboard. We are also interested in exploiting the interaction traces to profile the users, identifying types of regions that a particular user (or a class of users) are interested in. Or, instead of analyzing traces of many users on one video, we profile users by analyzing traces of one user on many videos. Users profile can then be used to generate personalized retargeted video. For instance, a student who is a visual learner might prefer to focus on slides or whiteboard during the lecture, while a student who prefer aural learning style might want to focus on the lecturer instead.

There are potential synergies between the crowd-sourced approach and content-based approach in the context of video retargeting. On one hand, crowd-sourcing would help content-based approach to understand the semantic of the video. For instance, in the *dice trick* video, the ROIs allow content analysis algorithms to correlate between the word "dice" and the dice object in the video as hint to object recognition. On the other hand, content-based approach could supplement crowd-source techniques, for instance, by helping to track the motion of object-of-interest or to position object of interest according to the rule of thirds. We plan to pursue the collaboration between the two approaches as the next agenda of our research.

10. REFERENCES

[1] Anonymous. Removed for blind review.

[2] D. Arijon. *Grammar of the Film Language.* Silman-James Press, 1991.

[3] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, 2002.

[4] P. Doubek, I. Geys, T. Svoboda, and L. V. Gool. Cinematographic rules applied to a camera network. In *Proc. of Omnivis, The fifth Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras*, pages 17–30, 2004.

[5] H. El-Alfy, D. Jacobs, and L. Davis. Multi-scale video cropping. In *Proceedings of MULTIMEDIA '07*, pages 97–106, Augsburg, Germany, 2007. ACM.

[6] X. Fan, X. Xie, H.-Q. Zhou, and W.-Y. Ma. Looking into video frames on small displays. In *Proceedings of MULTIMEDIA '03*, pages 247–250, Berkeley, CA, USA, 2003. ACM.

[7] M. L. Gleicher and F. Liu. Re-cinematography: Improving the camerawork of casual video. *ACM Trans. Multimedia Comput. Commun. Appl.*, 5(1):1–28, 2008.

[8] J. Han, K. N. Ngan, M. Li, and H. Zhang. Unsupervised extraction of visual attention objects in color images. *IEEE Trans. Circuits Syst. Video Techn.*, 16(1):141–145, 2006.

[9] L.-w. He, M. F. Cohen, and D. H. Salesin. The virtual cinematographer: a paradigm for automatic real-time camera control and directing. In *Proceedings of SIGGRAPH '96*, pages 217–224. ACM, 1996.

[10] T.-H. Huang, K.-Y. Cheng, and Y.-Y. Chuang. A collaborative benchmark for region of interest detection algorithms. In *Proceedings of CVPR '09*, Miami, FL, June 2009.

[11] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(11):1254–1259, 1998.

[12] T.-Y. Li and X.-Y. Xiao. An interactive camera planning system for automatic cinematographer. volume 0, pages 310–315, Los Alamitos, CA, USA, 2005. IEEE Computer Society.

[13] F. Liu and M. Gleicher. Video retargeting: automating pan and scan. In *Proceedings of MULTIMEDIA '06*, pages 241–250, Santa Barbara, CA, USA, 2006. ACM.

[14] Y. Pritch, A. Rav-Acha, and S. Peleg. Nonchronological video synopsis and indexing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(11):1971–1984, 2008.

[15] M. Rubinstein, A. Shamir, and S. Avidan. Improved seam carving for video retargeting. *ACM Trans. Graph.*, 27(3):1–9, 2008.

[16] N. Ukita, T. Ono, and M. Kidode. Region extraction of a gaze object using the gaze point and view image sequences. In *ICMI '05: Proceedings of the 7th international conference on Multimodal interfaces*, pages 129–136, Torento, Italy, 2005. ACM.

[17] D. Walther and C. Koch. Modeling attention to salient proto-objects. *Neural Networks*, 19:1395–1407, 2006.

[18] X. Xie, H. Liu, S. Goumaz, and W.-Y. Ma. Learning user interest for image browsing on small-form-factor devices. In *Proceedings of CHI '05*, pages 671–680, Portland, Oregon, USA, 2005. ACM.

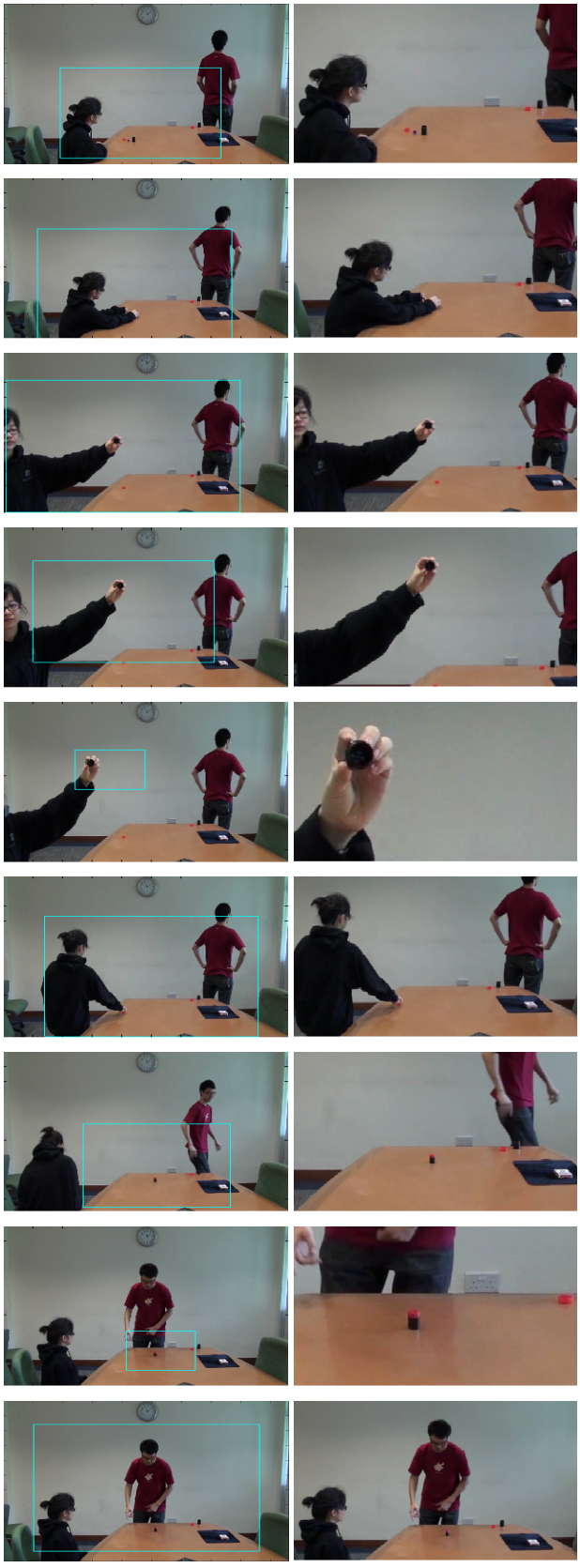

Figure 12: Retargeted video: selected frames (left) and retargeted frames (right)