Combining Content-based Analysis and Crowdsourcing to Improve User Interaction with Zoomable Video*

Axel Carlier
IRIT-ENSEEIHT
University of Toulouse
carlier.axel@gmail.com

Guntur Ravindra
Dept. of Computer Science
National University of Singapore
ravindra@comp.nus.edu.sg

Vincent Charvillat
IRIT-ENSEEIHT
University of Toulouse
vincent.charvillat@enseeiht.fr

Wei Tsang Ooi
Dept. of Computer Science
National University of Singapore
ooiwt@comp.nus.edu.sg

ABSTRACT

This paper introduces a new paradigm for interacting with zoomable video. Our interaction technique reduces the number of zooms and pans required by providing recommended viewports to the users, and replaces multiple zoom and pan actions with a simple click on the recommended viewport. The usefulness of our technique is visible in the quality of the recommended viewport, which needs to match the user intention, track movement in the scene, and properly frame the scene in the video. To this end, we propose a hybrid method where content analysis is complimented by the implicit feedback of a community of users in order to recommend viewports. We first compute preliminary sets of recommended viewports by analyzing the content of the video. These viewports allow tracking of moving objects in the scene, and are framed without violating basic aesthetic rules. To improve the relevance of the recommended viewports, we collect viewing statistics as users view a video, and use the viewports they select to reinforce the importance of certain recommendations and penalize others. New recommendations that are not previously recognized by content analysis may also emerge. The resulting recommended viewports converge towards the regions in the video that are relevant to users. A user study involving 70 participants shows that an user interface incorporating with our paradigm leads to more number of zooms, into more informative regions with fewer interactions.

Categories and Subject Descriptors: H.5.1 [Multimedia Information Systems]: Video

General Terms: Algorithms, Human Factors, Design

Keywords: Interaction Techniques, Zoomable Video, Content-Analysis, Crowdsourcing

1. INTRODUCTION

Advances in video compression, video processing algorithms, and video sensors have lead to video cameras that are capable of capturing high resolution videos. Ability to capture HD video (at 1920×1080) is commonly available in mobile phones and handheld cameras now. Video sequences, of resolution as high as $7,680 \times 4,320$ (UHDTV) have been recorded and transmitted over the Internet[1]. Video playback, however, is still limited in resolution, due to screen constraints (e.g., on mobile devices) or bandwidth constraints (e.g., when streamed over the Internet). As a result, high resolution videos are typically scaled down before transmission or playback, leading to a loss in information.

Zoomable videos have recently been proposed as a new form of media [14] with zooming and panning as two new ways for users to interact with the video. A zoomable video allows an user to zoom into a selected region in the video, leading to a high resolution version of the region to be displayed. The user essentially views the video through a *viewport* that defines a rectangular region in the high resolution video from which the displayed video is cropped. While zooming in, users can pan (i.e., scroll) around by moving the viewport to view different regions in the video. Such zoom and pan ability is useful for many types of videos, including classroom lectures, stage performance and sports video.

Ngo et. al [14] have proposed a user interface to interact with a zoomable video, based on a design similar to commonly used image zooming tools such as Google Maps. To zoom in and out, users can either scroll up and down with the scroll wheel of the mouse, or use the $+$ and $-$ buttons on the interface/keyboard. To pan, users can either drag the video with the mouse, or use the arrow keys on the interface/keyboard. A study of the interfaces [3] has shown that users interact frequently with the video during playback, with 70% of the interactions occurring within 1.6 seconds of each other.

Such user interaction was originally designed for images, where the content is static. Applying it to video, where the content is dynamic, raises several issues. Users typically zoom in to view one or more objects of interest in higher detail. In a video, these objects of interest can move over time. As a result, it is hard for users to position the viewport around an object that is moving. Furthermore, the user can loose track of the object of interest when the object moves out of the viewport. For instance, a user might zoom into a sports video clip to view the activity of a player in more detail.

*Area chair: Mor Naaman

[1] http://www.bbc.co.uk/news/technology-11436939

MM'11, November 28–December 1, 2011, Scotsdale, Arizona, USA.
Copyright 2011 ACM 978-1-4503-0616-4/11/11 ...$10.00.

If the player moves rapidly, the user would need to quickly pan or zoom out to track the person.

The goal of this research is to propose a new user interaction technique for zooming and panning in a zoomable video, designed with moving content in mind. The new interaction should be simple and intuitive, and users should be able to zoom and pan to view their object of interest easily with few interactions.

Our idea is to recognize the possible objects of interest in the video, and highlight them in a non-intrusive. These highlighted regions are then presented to the users as the *recommended viewports*. When users hover the mouse over any of the recommended viewports, a semi-transparent white overlay will appear, indicating the recommended viewport. Users can click once anywhere in the recommended viewport to zoom and pan into the recommended viewport. Users can still click anywhere outside the recommended viewport to zoom in. At any time, users can drag to move the viewport and right click to zoom out. Note that we replaced mouse wheel scrolling in [14] with clicks, since clicking is a more natural action for selecting a recommended viewport.

The new interaction which we propose is the ability to click on the recommended viewport, there by replacing the action of zooming in and positioning the viewport over the objects of interest (i.e., scrolling and dragging) with a single mouse click.

The recommended viewports serve several purposes. First, it moves along with the detected object of interest, thus making it easy for users to position the viewport over a moving object. If the user's viewport is the recommended viewport, the user's viewport would also move as the recommended viewport moves. As a result, the video pans automatically to track the moving object, further reducing the number of interactions required from the users. Second, the recommended viewport can provide a guide for the user to frame their viewport properly, i.e., in a way that does not violate basic aesthetic principles (e.g., cropping a person's head), leading to more pleasant viewing experience.

The key to the efficacy of our approach is the quality of recommended viewports. Viewports should highlight and track interesting and relevant regions in the video and frame them properly. The research challenge, that we address in this paper, is to automatically compute the recommended viewport given a video. The computer vision research community has studied this domain for many years, and many algorithms exist to detect and track foreground objects, faces, human body, etc, by analyzing the content of the video. While the research community has made significant progress, most of these algorithms work at the "syntax level" of the video content. What the users find interesting, however, depends on the semantic of the video, the purpose of the video, and the intention of the users. It is difficult, therefore, to determine the object of interest just by analyzing the video content using the existing algorithms.

We have recently proposed [2] an approach that relies on implicit feedback of a community of users in order to identify the regions of interest in a video. We refer to the process of collecting user feedback as 'crowdsourcing' as defined by piggyback crowdsourcing systems [6]. By learning from the viewing patterns of the users (which regions of interest are more popular), a re-targeted video highlighting regions of interest is produced. This technique exploits human ability to recognize the semantic of the video and is able to learn the regions of interest from the most common intention of the users. This system, however, collects user traces based on the user interface described by Ngo et al [14] and therefore suffers from the same issues as discussed earlier. In particular, the need for users to pan and track an object of interest introduces delay in identifying a moving object as the region of interest. For instance, in the re-targeted video produced[2], which follows the behavior of most users, an object of interest can move out of frame momentarily, and only moves back into the frame when users react to this event by panning to move the objects back into their viewport. Another drawback of this crowdsourcing system is that, even though the inputs into the crowdsourcing algorithm matches the semantics identified by the users, the process of combining users' intention to produce a region of interest is blind to the content of the video and might violate the basic aesthetic rules.

However, the interface described in this paper adopts a hybrid approach that plays on the strength of both content analysis and crowdsourcing (implicit feedback). Not only semantically important regions of interest are identified, but panning and tracking a region of interest is automated.

We bootstrap the recommended viewports as follows. First, we analyze the content of the video and identify salient regions using motion, faces, and saliency features. These salient regions are recommended to users viewing the video using our interface. Then, we learn which viewports are chosen by the users to view the video, and adapt the recommended viewports to match the intention of most users. By combining both content analysis and crowdsourcing, we are able to recommend properly framed regions of interest that match better with user intention.

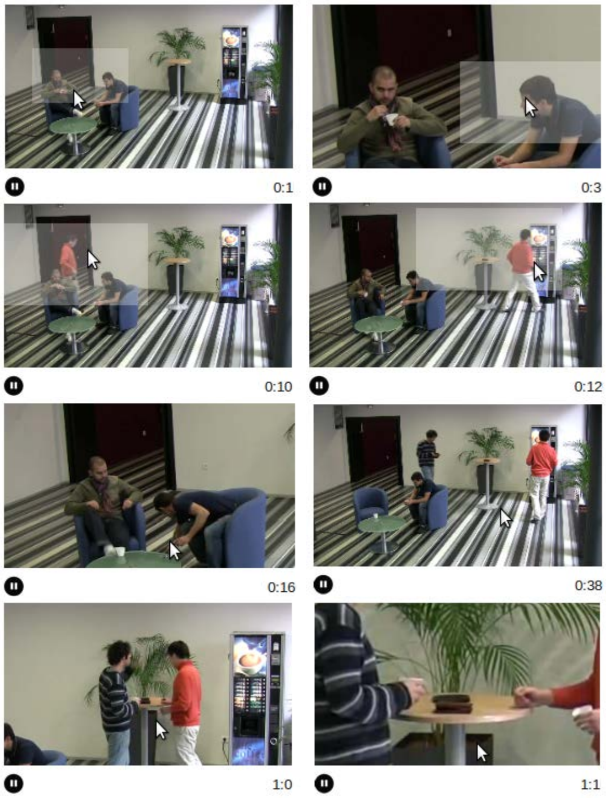


Figure 1: Our Zoomable Interface

In summary, our proposed technique to interact with a zoomable video fuses regions of interest identified from content analysis with regions chosen implicitly by previous users, in order to recommend viewports to new users. These recommended viewports guide user's viewport placement, tracks moving objects of interests, and

[2] http://www.youtube.com/user/autozap

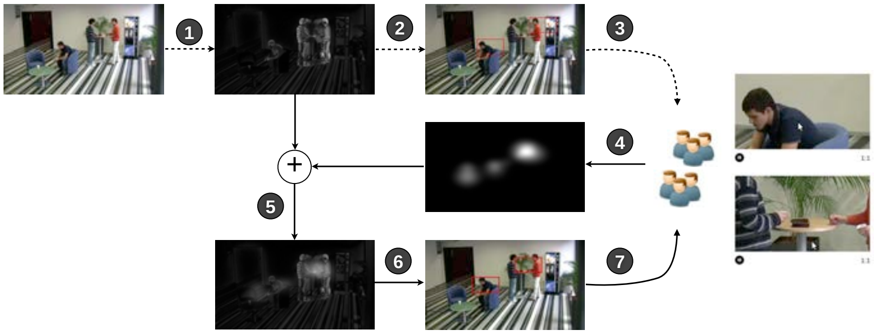


Figure 2: Overview of Approach

thus leads to fewer interactions. Further, the recommended viewports have few violations of basic aesthetic rules and show content that is more relevant to users.

1.1 Example

We illustrate our new interaction technique with an example. Figure 1 shows eight snapshots from our user interface. In the snapshot at time 0:01 (top left), you can see a semi-transparent white rectangle overlayed on the video where the mouse is hovering, indicating the recommended viewport. If the user clicks anywhere within the recommended viewport, the video zooms in, placing the viewport over the recommended ones (time 0:03). Recommendation at further zoom levels becomes available. The user right clicks to zoom out (time 0:10). The new recommended viewport tracks the person in orange sweater as he enters the scene, from time 0:10 to 0:12. The user may also click on the video outside of recommended viewport, as seen in rest of the snapshots.

1.2 Approach Overview

Having briefly described the general idea of recommending viewports, we now give an overview of our approach to compute the recommended viewports.

Figure 2 illustrates the steps taken. We start with a given video. In Step 1, we analyze the content of the video to extract the salient features of each frame. An importance map is built for each frame, indicating the level of saliency in each pixel. The output of Step 1 in Figure 2 shows an example importance map. In Step 2, we cluster the pixels in the importance map. The clusters are analyzed across consecutive frames, and the best viewport trajectory is computed. This viewport trajectory is used as a recommendation, and is integrated into the user interface.

As users use our interface to watch videos (Step 3), we collect information about the viewports chosen by the users. In Step 4, this information is used to generate an interest map, indicating the level of interest users have on each pixel. The interest map is combined with the importance map (Step 5) to generate a new importance map. In Step 6, we repeat what we do in Step 2, with the new importance map. New recommended viewports are shown to the users (Step 7). Steps 4–7 can be repeated by periodically recomputing the new recommended viewports to adapt to new user interests if any.

1.3 Organization

We organize the rest of the paper into six sections. We begin with a review of literature in Section 2, followed by a description of zoomable video in Section 3. In Section 4, we explain how we analyze the content of the video to find the initial recommended viewports (Step 1 and Step 2). Section 5 explains how we crowdsource the interesting regions from users, and combine it to generate new recommended viewports (Step 4 and Step 5). In Section 6, we present our results. Finally, we conclude in Section 7.

2. RELATED WORK

We now discuss related research in the literature and contrast our work with the existing work. We divide existing research into three classes: (i) those that enables new video access and interactions using content analysis, (ii) those that exploit user interactions to learn about the content, and (iii) those that combines both content analysis and user behaviour analysis to learn about content and enable new video access and interactions techniques.

Enabling New Interactions with Content Analysis. Analysis of video content to detect and track objects is highlighted by Goldman et al. [9] as enabler for new interaction paradigms to browse and manipulate video. Examples given include dynamic video annotation and video navigation by direct manipulation. Direct manipulation, also discussed by Dragicevic et al. [7], allows control of video playback and access of nearby frames by directly clicking and dragging on moving objects in the image space. Content nalysis also enables hypervideo links [17], where links associated to moving video objects allows a quick access to additional content.

Many new interactions are made possible by ROI detection, a fundamental problem in content analysis. ROI are commonly detected using visual attention models. The work by Han et al. [10] and Itti et al. [11] are representative in ROI detection on still images. ROI detection can be extended to the temporal dimension: to look for interesting space-time regions within a video. Detection of interesting or important video segment (or shot) enables new video interactions such as those proposed in the *smart player* [4]. The smart player adjusts the video playback speed based on a predicted level of interest for each video shots. Both inter-frame motion estimations and shot boundary detections are used to support automated fast-forwarding during less interesting shots.

Improving the viewing experience is also the ultimate goal of numerous works on image and video retargeting. Rubinstein et al. have identified video cropping as one of the most basic retargeting operator, despite many recent new proposals [15]. El-Alfy et al. uses motion analysis in a video and crops the region with high motion energe for display on a surveillance wall, resulting in zooming and panning effect over time [8]. Liu and Gleicher [12], in their influencial work, linearly combine multiple saliency features to extract prominent regions from an image for video retargeting.

Our work is similar to these previous work. We all rely on video content analysis to identify interesting regions from the video. As mentioned, however, content analysis alone can fail to identify regions relevant to users' intention, since content analysis is often done using low-level information without user context. Existing work that works well often assume a certain context and predetermined user intention (e.g., El-Alfy et al. works in the context of surveillance video where motion is important). In contrast, our work aims to target a wide range of applications.

Enhancing Content Understanding Through Usage Analysis An alternative approach to content-based ROI detection is to observe what the users are looking at and which video object are they clicking on. By collection usage pattern from the users, we can infer the regions in the video that users are most interested in.

In the context of ROI detection in images, Xie et al. [22] found that on small mobile displays, users tend to zoom and scroll more often to view interesting regions in detail. These user actions yield user interest maps, which are used to extract ROIs from the image.

In the context of ROI detection in video, Ukita et. al uses an eye-mark recorder system to track user gaze. Important objects, including moving objects, can be detected and tracked by analyzing the gaze of the users [19]. Shamma et. al proposed a similar approach: to gather information about what is being watched from a user community to understand the content of the video [16]. Syeda-Mahmood and Ponceleon collect implicit feedback from the users by analyzing the playback interaction in the temporal domain (play, fast-forward, pause) and use it to infer the most interesting video segments from the video [18].

Carlier et al. use a zoomable video interface [2] to crowdsource ROIs from users. The detected ROIs are used to create a retargeted version of the video that would automatically pan and zoom.

Our work is similar in spirit with these existing work, especially Syeda-Mahmood and Ponceleon's [18] and Carlier et al.'s [2]. Similar to their approach, we use implicit feedback from users to identify interesting regions. Our work, however, addresses the unique challenge of tracking moving objects using users' feedback, and proper framing of regions around objects.

Combining Content and Usage Analysis Given the strengths and weaknesses of content and usage analysis, it is natural to combine both, as they can complement each other. This technique, however, is not well explored by the research community. Besides our work, presented in this paper, the only other related work we are aware of is the *smart player* [4], which adapts the playback speed from both the visual richness of the scene (estimated from content analysis) and the user preferences (learnt from their video interaction). The player uses implicit feedback and learns how the users change and override automatically recommended fast-forwards.

Unlike the *smart player*, which focuses on interaction in the temporal domain (play and fast forward), our work focuses on interaction in the spatial domain (zoom and pan), detecting of interesting regions to recommend is therefore more challenging.

3. ZOOMABLE VIDEO

This section gives an introduction to zoomable video and the notion of recommended viewport. We describe how a user would interact with the zoomable video using recommended viewport, and defer the discussion of how the recommended viewport is computed to the next section.

Zoomable video is a term coined by Ngo et. al [14] to refer a video that is encoded and stored at multiple resolutions and supports dynamic cropping and random access into the spatial region in the video. Besides normal operation in the temporal dimension such as play, pause, fast forward, a zoomable video supports two new types of operation: zoom and pan. Users view a zoomable video with a display size that is smaller than the maximum resolution of the video. Zooming allows users to magnify a selected region in the video without loss of resolution (up to a maximum zoom level) and panning allows user to move the zoomed in region spatially around the video. Zooming and panning can occur even when the video is playing.

Zoomable videos are very helpful in situations where display devices are computationally constrained by the ability to decode and render high resolution video. Zoomable videos are also useful in cases where devices have access to streamed video over a low bandwidth network interface, resulting in the inability to stream bandwidth intensive high resolution video. Traditionally, computational and bandwidth constraints have been handled by scaling down the video temporally, spatially and in quality. Such an approach would result in loss of information, despite the fact that the capture devices were able to record the video at very high resolution. Zoomable video provides an alternative where users can select regions of interest from a low resolution video and view these regions at higher resolution. Such a scheme can satisfy both bandwidth and computational constraints in a more scalable fashion.

A zoomable video system can be built using a bit-stream switching architecture. The high resolution video is encoded at multiple low resolutions. The lowest resolution video is first accessed and displayed to the user. When a user wishes to view a specific region at a higher resolution, user's intent is represented as a rectangular viewport. This viewport is first mapped to a higher resolution layer. Then a region corresponding to the specified viewport is cropped from the higher resolution layer, and presented to the user. As is evident, the key requirement is the ability for dynamic cropping, which can be a challenge in encoded video. Methods to handle dynamic cropping and other issues related to streaming of zoomable video have been presented in [14].

3.1 Recommended Viewport

A key feature in our proposed interaction is the *recommended viewport*, which corresponds to interesting objects or events in the video, and is a region in the video that the user is likely to zoom into. Hovering the mouse over a recommended viewport reveals a white semi-transparent rectangular box. The user can zoom into the region by left-clicking the recommended viewport with the mouse button. When the mouse cursor hovers over one or more recommended viewports that overlap, only the most important recommended viewport is shown.

Users can left click outside of a recommended viewport to zoom in. In this case, the new viewport of the user is placed such that the center of the viewport is the coordinate of the mouse click. Right-clicking anywhere of the video zooms out.

The recommended viewport automatically moves to track a moving object of interest. If the user's current viewport is one that matches the recommended viewport (by clicking inside a recommended viewport), the user viewport pans automatically along with the recommended viewport.

One design decision we make is to limit the number of recommended viewports per zoom level to three. While using recommended viewport helps users to place their viewport easily with a single mouse click, it also restricts the viewport placement. Presenting too many recommended viewports to the users is not only too inflexible, but can also be confusing. We choose the value of three since we observe that in a typical video there are rarely more than three events of interest at the same time. Of course, this number can be configured to be higher depending on the content.

3.2 Implementation

We implemented our interface on a Web browser using HTML5. The webpage is minimalist in design, and shows only a video can-

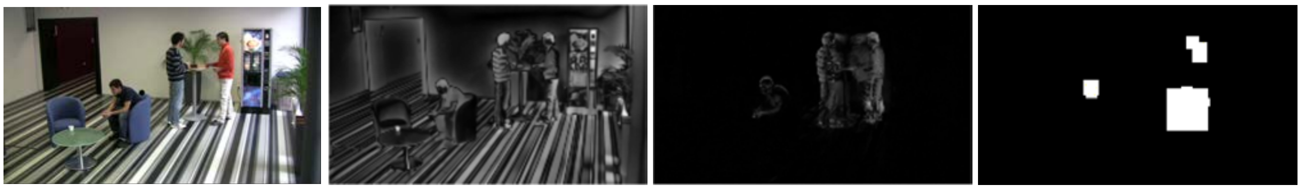
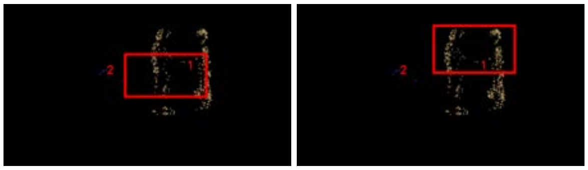

(a) Original Frame (b) Saliency Map (c) Motion Map (d) Face Map

Figure 3: Content Analysis

vas of size 320 × 180 and a play/pause button along with the current playback time. Videos of resolution 1920 × 1080 are loaded along with a JSON file that contains the recommended viewports. A Javascript crops and scale the video for display in the canvas, as well as highlights the recommended viewport according to the JSON file (when hovered by the mouse).

A critical ingredient to the success of our approach is the quality of the recommended viewports. In the next two sections, we detail how we combine content analysis and crowdsourcing to compute the recommended viewports.



(a) Bad Cut (b) Good Cut

Figure 4: Frames Showing a Bad Cut and a Good Cut

4. CONTENT-BASED ANALYSIS

The recommended viewports are determined based on a sequence of content analysis steps. We use saliency, and motion in the frame as the two main criteria to determine the possible regions of interest (ROI), which then serve as the recommended viewports. In cases where a face can be detected, face position is used as an additional criterion. Saliency is determined using the visual saliency features described in [13]. These features have shown good results for human detection, and hence the saliency is biased towards the presence of human or human-like objects in the video. Figure 3(b) shows an example saliency map.

Motion saliency is based on a moving average of frame differences and is similar to the work in [21]. A single channel disparity image of two successive frames is added to a moving average and used in place of the current frame. As a result the long-term motion pattern is available in a single frame. Figure 3(c) shows what a motion saliency map looks like.

Face detection, for both frontal and profile faces, is done using the Viola-Jones face detector [20]. We track faces across frames using a hue histogram of the detected face, with the CAMShift algorithm. Figure 3(d) shows the result of face detection.

4.1 Importance Maps

Each video frame is now represented by three maps, one for each criteria mentioned above. The pixel values are single channel quantities ranging between 0 and 1. The frames representing the three criteria are linearly combined to give an *importance map* I.

The weights assigned for linear combination may be obtained in many ways. In our experiments, we use empirically derived weights 0.7 for motion, 0.2 for saliency, and 0.1 for faces.

4.2 Clustering of Important Regions

The next step is to cluster pixel elements in the importance map to reveal regions that are candidate ROIs. The pixel elements in the importance map of every image are selected based on a threshold. Selected pixels are clustered using Mean-Shift ([5]) clustering algorithm. We chose a 60 pixels bandwidth in our implementation since it allows to detect significantly different viewports in 1920 × 1080 videos. Once clustering is performed, the candidate ROIs are determined at each zoom level. All ROIs at a particular zoom level have the same dimensions, i.e., the dimension of the corresponding viewport at that zoom level.

A ROI is a good candidate if it encloses as many cluster points as possible without leaving out cluster points. Hence we need a measure to quantify the extent to which a viewport cuts a cluster into two or more parts. We define a term called the *cut of a viewport*. If a viewport does not cut a cluster, then its cut is very high. The concept of using a cut has been proposed in earlier work [11, 12, 8], but we use the notion of cut in conjunction with clustering to minimize cuts to the top portion of an object. Such an approach is better suited to aesthetically present human-like objects without cutting body parts. Figure 4(a) shows a case where the viewport cuts the clusters into two parts. There is a cluster region enclosed within the viewport, and the remaining cluster points fall outside the viewport. The region within the viewport encloses the lower part of two human subjects whose presence may be identified by the structure of the clusters shown. Hence, this viewport should be penalized. On the other hand figure 4(b) shows a viewport cutting a cluster such that the upper portion of the human subjects is enclosed, and the lower portions are cut out of the scene. This viewport should be treated favourably in comparison to the previous case. Hence the *cut of a viewport* should also account for how the cut is performed.

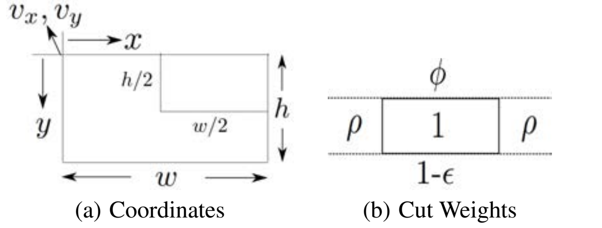
We now formally define how the cut is computed. For a frame f, let C_f represent all the cluster centroids in f. Let $E(c)$, where $c \in C_f$, be the set of pixels clustered around centroid c. Then $CUT(V_f, f)$, the cut of viewport V_f with top left coordinate (v_x, v_y) and height h (See also Fig 5(a)), is a measure for the extent to which $E(c)$ is fully contained within V_f:

$$CUT(V_f, f) = \frac{\sum_{c \in C_f} \sum_{p \in E(c)} W(p, V_f)}{\sum_{c \in C_f} |E(c)|}$$

where $W(p, V_f)$ with p at coordinate (x, y) is given by

$$W(p, V_f) = \begin{cases} 1 & \text{if} p \in V_f \\ 1 - \varepsilon & \text{if} y > v_y + h \\ \phi & \text{if} y < v_y \\ \rho & \text{otherwise} \end{cases}$$

and $1 > 1 - \varepsilon >> \rho > \phi$ (See also Fig 5(b)). $CUT(V_f, f)$ reaches a maximum value of one when V_f contains all cluster points.



(a) Coordinates (b) Cut Weights

Figure 5: Viewport Coordinate System and Weight Assignment for a Cut

Our goal is to find the best viewports that maximize $CUT(V_f, f)$. This step is achieved by evaluating all candidate viewports and selecting a few that have the highest value of cut.

4.3 Tubing: Finding Recommended Viewports over Time

Viewports change in every frame. When users select a recommended viewport in a frame, the same viewport may not be optimal when recommended in the next frame. The system has to switch to a nearest viewport, causing a virtual camera shake. To minimize the irritation caused by frequent and abrupt change in viewport position, we compute an optimal strategy to switch viewports while maintaining a smooth, linear transition of the viewport position. Such a linear change manifests as a virtual camera pan.

To compute a virtual camera pan, we first designate some frames in the video as key frames. All frames falling between two key frames constitute a *shot*. The viewport is allowed to linearly change position within the shot. We expect the viewport at the beginning of a shot to smoothly change to a viewport at the end of the shot.

In our implementation, we use one key frame every 20 frames. There is a trade off involved in the choice of inter key frame distance. More key frames would lead to less stable viewport (more shaky) and fewer key frames would lead to less optimal recommended viewports (the reason for which will become clear later).

There are multiple candidate viewports at different zoom levels at the beginning and at the end of a shot. We can choose different combination of the starting viewport and ending viewport. Each of these combinations result in a spatial-temporal trajectory of viewports across the frames in between the two key frames. We refer to this trajectory as a *tube*. Since we linearly interpolate the viewports in a tube, a viewport in an intermediate frame may not be the optimal viewport for that frame (as determined by the cut).

Our goal is therefore to find a good tube that gives good overall viewport quality across all frames in the tube. To this end, we define four metrics to evaluate the quality of the tube. Given a tube $\mathcal{T} = < V_f, V_{f+1}, \ldots V_{f+N} >$ we first consider the cut metric, which is used to prevent violation of aesthetic rules, especially for human body. We define the *cut of a tube* as the sum of all cuts of viewports in the tube. Second, we consider the importance of the tube, and define the *heat* of a tube as the sum of all importance value in every viewport $V_{i,i\in f\ldots f+N}$ in the tube, i.e., the pixel values in the importance map of each frame that falls in the viewport. Third, we consider the *temporal coherence* of a tube. We aim at preserving the motion of foreground objects within a tube. We proceed as in [1] and track our clusters over time by *mode seeking*. A given cluster is tracked across l frames creating a temporal chain of cluster centroids $< c_j, c_{j+1}, \ldots c_{j+l} >_{id}$ that starts at frame j. Each chain is assigned an unique id, and every cluster whose centroid is part of this chain is labelled with that same id. With this simple method clusters might split or merge as we track, creating new chains. This is not an issue as it is not required to precisely track and segment objects to ensure temporal consistency [21]. The temporal coherence of a tube can then be computed. For every pair of viewports $V_{f'}$ and $V_{f'+i}$ in the tube, a score proportional to i is added to the tube's coherence value $COH(\mathcal{T})$ every time clusters with the same id can be found in both $V_{f'}$ and $V_{f'+i}$[3] The coherence is then normalized to range into $[0,1]$. Finally, we consider the spatial displacement of the tube. We define the *regularity* of the tube as the measure for rate of change of the viewport positions within the tube. We compute regularity as

$$e^{-\frac{\|v_f - v_{f+N}\|_2^2}{N^2}},$$

where v_f and v_{f+N} are the initial and final viewport positions respectively, and N is the number of frames in the tube $\mathcal{T}$. This metric penalizes large displacement of viewports with an exponential weighting function.

To find the set of good tubes between two key frames, all possible tubes are computed and assigned a score by simply summing the four metrics. The top k tubes with the highest scores are chosen and form the recommended viewports between the two key frames (we use $k = 3$ in our implementation). An example of the result of creating tubes is shown as viewport sequences in figure 6.

5. COMBINING CONTENT-BASED ANALYSIS AND CROWDSOURCING

Using the approach detailed in the previous section, we have viewports to recommend to users (step 3 of Fig 2). We then start collecting user interaction traces with our zoomable interface. The goal is to find an approach to refine the content based viewport recommendation with the user selected viewports.

While zooming with the interface, each user selects a viewport at a given frame f. This viewport V_f is a rectangle whose top left corner is positioned at (v_x, x_y) and is of size (w, h). The viewport crops the important region the user is interested in. Since the level of interest of each cropped pixel differs with respect to its position within the viewport, as in [2] we assume that users naturally center the viewport on the most interesting area. We then model the interest within a viewport as a gaussian pdf being centered at $\mu = (v_x + w/2, v_y + h/2)$ with a covariance Σ constrained to the dimensions of the viewport: $\Sigma \propto \begin{pmatrix} w^2/4 & 0 \\ 0 & h^2/4 \end{pmatrix}$ and $\int_{V_f} \mathcal{N}(\mu, \Sigma) = 0.99$.

A user interest map UIM_f associated with frame f is then crowdsourced by accumulating the interest levels from multiple users. If we collect K viewport traces from K users who have zoomed on f, a gaussian mixture model can be computed such that $UIM_f^K = 1/K \sum_{k=1}^{K} \mathcal{N}(\mu_k, \Sigma_k)$.

The first image in Figure 7 shows $K = 16$ viewports selected by users while watching a long jump video with our interface. The second image shows the associated user interest map generated using the gaussian mixture model described earlier. In this example, users focus on the sand pit because they were asked to estimate the length of the jump.

With this simple formulation, the user interest map computation stabilizes after only 10 or 15 users. We observed that the KL-divergence $KL(UIM_f^{K+1} || UIM_f^K)$ is negligible for $K \geq 15$. Note

[3] $COH(\mathcal{T}) \propto \sum_{i=f}^{f+N} \sum_{c \in V_i \cap C_i} \sum_{k=i+1}^{f+N} \phi(c,k)$ where $\phi(c,k) = k - f$ if the cluster centroid c has an id that can be found among the ids present in V_k.

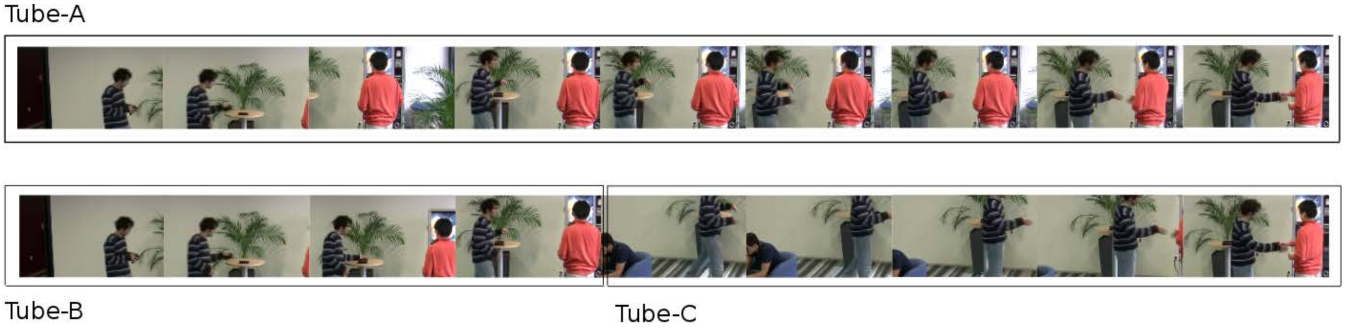


Figure 6: Viewport sequences showing Tubes. Tube-A is a long tube showing a scene where a person in motion is followed. Tube-B and Tube-C are other candidate Tubes rejected by the Tubing algorithm. Tube-B is short and has a low Regularity measure, while Tube-C is not only short but also has a lower cut value that manifests as a not so aesthetic framing

that in this paper we do not consider any sequential estimation of the user interest map where a weight could be used to de-emphasize old traces in a time-varying context.

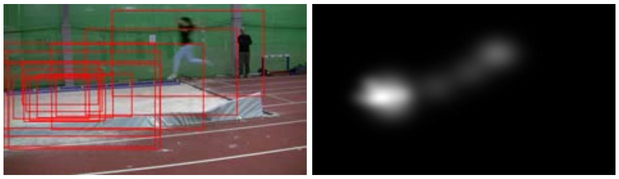


Figure 7: Creating user interest maps

As shown on step 5 of Fig 2, we then use the implicit feedback from users as another modality in the computation of importance map. We merge user interest maps with content-based importance maps by assigning them an equal weight. How to properly weight remains an open question: we plan to study the performances of different (either static or dynamic) weighting strategies in our future work. Yet experiments presented in the next section show that this simple strategy already demonstrates significant success. Indeed by applying the algorithms presented in section 4 to this new importance map, we obtain updated recommended viewports that match intention of the users better.

6. RESULTS

6.1 Experimental Setup

Video Sequences. We use four videos to assess our work. Three of these videos show a long jump, where we can see the end of the runway and the sand pit. The action consists of an athlete running, jumping, and then going back to his coach while the sand is being swept (see Fig 8). We refer to these video clips as longjump0, longjump1 and longjump2, and each last around 30 s.

The fourth video is longer and semantically more complex. Fig 1 summarizes the action taking place in a coffee lounge. At the beginning of the video, two people are sitting in the foreground on blue sofas (times 0:01 and 0:03). A new person in orange sweater is then entering the scene, and loses his keys while removing his wallet from his pocket (time 0:10), before reaching the coffee machine and staying there (time 0:12). At the same time, one of the two seated people picks up the keys (time 0:16) and hands it over to his friend who leaves the scene. After this theft, a fourth person arrives and goes to the coffee machine (time 0:38), after leaving his wallet on the same table as the person in orange (see also Fig 6). At the end of the video, the two people from time 1:00 (Fig 1) take their wallets and leave the scene, leaving the thief alone in the room. We call this 1 minute and 22 seconds video clip coffeelounge.

Figure 8: longjump2 video

Interaction Techniques. We built four variants of user interfaces for zoomable video. The version we proposed in this paper is denoted as RC+U, which stands for *Recommendation based on Content and Usage*.

To study the effect of combining usage analysis with content analysis, we setup a version of the user interface that uses only recommended viewport computed using content analysis, without considering user access pattern. We call this version RC (Recommendation based on Content). This version is equivalent to the output of Step 3, after the process described in Section 4.

The third variant of the user interface we setup is called NR, which stands for No Recommendation. The purpose is to study the effects of presenting recommended viewport to the users. All interaction elements in this user interface remains the same, except that the recommended viewports are removed.

Finally, we setup a variant of the interface which we refer to as NZ, standing for No Zoom. We use NZ in one of our control experiments. NZ does not allow any zooming or panning.

Methodology. We evaluate the three successive versions (NR, RC and RC+U) of our interface by conducting the following user study. In our experiments, the user traces from RC are used to compute the recommended viewports of RC+U.

We assign tasks to users where zooming may be useful. We ask them to estimate the jump length in longjump0, longjump1 and longjump2 and we ask them if there are key thefts and/or wallets thefts in coffeelounge.

First we provide a first group of users with NZ, which plays only the low resolution (320 × 180 px) version of the video sequences without any interaction. We want to evaluate how well users answer the questions without zooming.

The core of the user study involves comparing the three versions of our interface: NR, RC and RC+U using three independent set of 20 users Users0, Users1 and Users2.

Except for NZ where no interaction is available, we always start our user studies with a learning phase. We first demonstrate the features of the interface, and then observe how users interact with our training video (longjump0). We never continue the study without explicitly reminding the user of interactions he/she did not try.

Then we explain to users what task they have to complete while viewing the next video clips. We always present the clips in the same order: longjump0, longjump1, longjump2 and coffeelounge. We collect a user's answers at the end of each clip. We let users watch the video as many times as they want and we record every interaction into a database. In average the test lasted between 8 and 10 minutes for each participant.

Participants. We collected traces from 16 females and 54 males (total 70 participants), with an age ranging between 19 and 55 years old. Among these users, 10 were presented NZ and 20 each were presented other interfaces.

General Statistics. We collected a total amount of 102,470 events in our database, 34,623 being produced during the training phase on longjump0. All those events are needed to compute user interest maps (see Section 5). However in the following discussion, we only consider a subset of those events that allow us to compute the number of zoom in, zoom out and panning. We got 6,694 of such events from a total number of 434 viewings of our 4 videos (6 viewings per person on an average).

We now present the results of our user studies. We discuss only our data on longjump2 and coffeelounge because of the following reasons. First we used longjump0 to train users. Second longjump1 has been used as a control experiment to check that our results are independant from the set of users. Indeed, we observe that Users2 behave the same as Users1 when viewing longjump1 when they were provided with the same set of recommended viewports using the interface RC (see Table 1). We omit the detailed result of this control experiment here due to space constraint.

Video	**Users0**	**Users1**	**Users2**
longjump1	NR	RC	RC
longjump2 and coffeelounge	NR	RC	RC+U

Table 1: Protocol of the Experiment

6.2 Number of Interactions

We analyze the traces to count the number of user interactions in each of NR, RC, and RC+U. Figure 9 shows the results. We first observed that there is no significant difference in the number of zooms, but the number of pans when using NR is on average almost twice as much as RC+U. Since pans are used mostly to position the viewport correctly, this results show that the recommended viewport is useful in reducing the number of interactions. The number of pans for RC+U is also less than RC, indicating that the quality of recommended viewports for RC+U is better, as users pan less using RC+U. This point will be further elaborated in the next result.

Note that coffeelounge lasts 1 minute 22 seconds while longjump2 is only 35 seconds long, an this explains the difference in the number of panning for each video.

Number of Recommended Viewport Selected To further compare the recommended viewports between RC and RC+U, we analyze the trace at one particular event in the coffeelounge video, to further understand how the recommended viewport are clicked.

When Users1 zoom on the coffeelounge video (with RC), the recommended viewports are clicked 45% of the time. The remaining 55% clicks are outside the recommended viewports (Table 2). The first row of Table 3 gives insights into the distribution of zoom levels of those 45% clicks. It shows that users rarely zoom to the maximum level (i.e. close-ups). This result highlights the importance of the relationship between content semantic and participants' tasks. In this task, automatically detected close-ups are not useful enough to successfully complete the task.

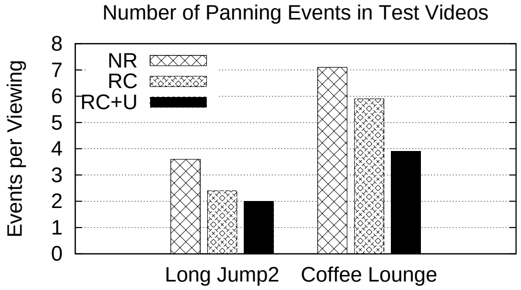


Figure 9: Number of panning events per view session

However, RC+U exhibits better performance regarding the number of recommended viewports clicked like we see on the second row of Table 2. The ratio of clicks on recommended viewports and outside shows that recommended viewports are more relevant. This underlines the interest of combining content analysis with crowdsourcing to learn better ROIs. Moreover, learning better ROIs affects the distribution of zoom levels, summarized by Table 3. As shown below, emergence of task-relevant close-ups encourages users to zoom more when needed.

We observe the same phenomenon on longjump2. Recommended viewports in RC are not really selected by users (only 18%, see first row of Table 2). But once combined with user maps, new recommended viewports are twice as much clicked by users (40%).

Video	**longjump2**	**coffeelounge**
RC	18%	45%
RC+U	40%	55%

Table 2: Percentage of clicks in recommended viewports for RC and RC+U

Interface	960×540 **px**	640×360 **px**	320×180 **px**
RC	73.2%	25.6%	1.2%
RC+U	24.6%	42.5%	32.9%

Table 3: Size of the recommended viewports clicked by users on coffeelounge

In summary, we found that integrating user interest maps to improve the relevance of recommended viewports yields a better recommendation. Users more often used the recommended viewports resulting in lower number of panning events.

6.3 Tracking of Moving Objects

We now demonstrate the importance of using content analysis to recommend the initial set of recommended viewports.

Because content analysis includes motion detection and tubing ensures temporal consistency, some recommended viewports track moving objects on the scene. Fig 11 shows one such example, following a character entering the scene and putting his wallet down a table. This recommendation actually helps answering one of the task questions because it makes users identify which wallet belongs to who. This particular region has been selected by 7 users out of 20 using RC.

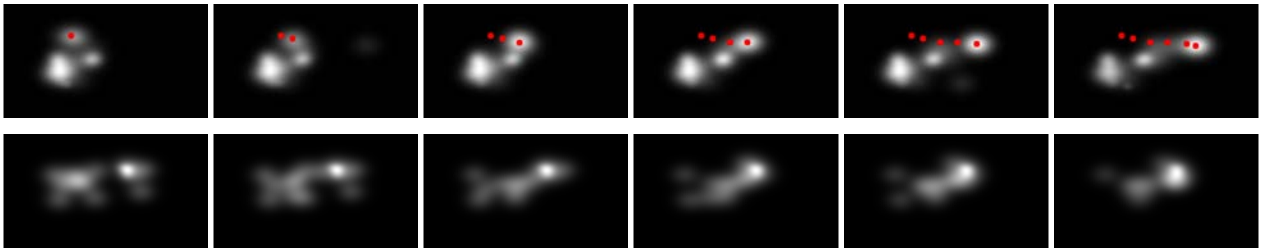


Figure 10: Two sets of user maps during the same time interval: first row comes from use of RC and second row from use of NR

Fig 10 emphasizes the importance of content analysis. The first and second rows show user maps from RC and NR respectively, during the time interval as in Fig 11. We see that the movement of the character clearly emerges as a focused hotspot (materialized by moving red dots) in RC. This hotspot is a result from the previously mentioned 7 viewers who used RC to click on the recommended viewports. With the NR interface, users can not easily follow the moving character even with repeated panning. The user maps from NR (second row of Fig 10) illustrate the absence of a hotspot following the moving person. Therefore, introducing content analysis before explicitly learning from users, reveals more about content semantic than relying only on crowdsourcing.

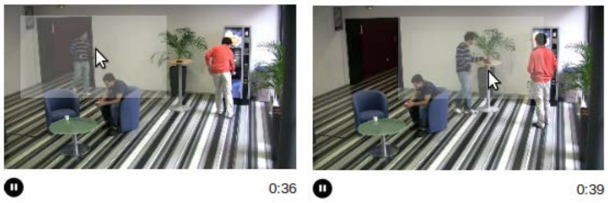


Figure 11: Example of an object producing a moving recommended viewport.

6.4 Better Framing

The other strength of our approach is that, by integrating content analysis into recommendation of viewport, we can eliminate recommended viewports that violate basic aesthetic rules.

Figure 12 presents the output of our framing optimization on the same frame but with different input. In the left part of the figure, we use user interest maps from NR, resulting in an unsatisfied framing. Users of NR have been watching this region at a higher zoom level centered between the two human subjects in the scene. Hence the user interest maps were produced with a blob at this exact place. The framing optimization placed the recommended viewport in the region maximizing the heat (as per algorithms detailed in the previous section) resulting in a case of bad framing.

The right part of the figure shows the same result but with user interest maps from Users1 as an input. In this case, users were biased with the recommended viewports generated, thanks to content analysis. The corresponding user interest map presents a hotspot slightly shifted to the right compared to the one generated from Users0. As a result, the framing optimization produces a recommended viewport that is aesthetically better as it does not cut body parts of the subjects in the scene.

6.5 Understanding Video Content

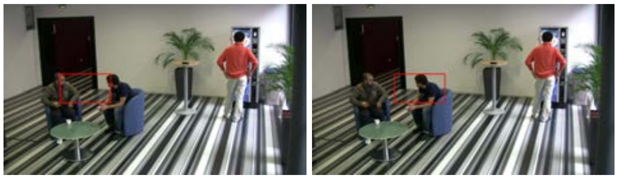


Figure 12: Framing with traces from use of NR (left) and use of RC (right)

Previous results show that recommended viewports from RC+U are selected more often than the ones from RC, but does it mean that it helps them understand the content better? The answer, as shown in this section, is yes.

Table 4 presents users' answers to the questions based on the task specific to coffeelounge. As a reminder, we asked users whether or not a key was stolen (the correct answer is yes), and whether or not a wallet was stolen (the correct answer is no). Whereas it is quite easy to spot the theft of the key even without zooming, there is an ambiguity with regard to the wallets (it may appear as if they were exchanged). Zooming in to the region of the wallets can resolve this ambiguity.

We noticed that 70% users who see the video at a low resolution (NZ) spot the key theft, whereas only 50% identify that no wallets have been stolen. We actually observed during the study that users tried to guess the answer because they could not see accurately, and indeed the answers were equally distributed as yes and no. This result provides us with a lower bound to compare interfaces NR, RC and RC+U.

The percentage of good answers is higher with NR thanks to the zooming functionality, especially for the wallets question (70% of good answers). However results are disappointing for RC. Guiding users with recommendations is potentially double-edged: since they follow the recommendations (Table 2), the quality of their answers is correlated to the relevance of the recommended viewports with respect to the task. About the specific wallets task, it is understandable that content analysis alone fails to detect the region around the wallets as one of the prominent ROIs.

We get the best answers with RC+U, which combines content analysis and crowdsourcing. Indeed we observe in Fig 13 that crowdsourcing complements content analysis to create a new recommended viewport located on the wallets.

Interface	NZ	NR	RC	RC+U
Is there a key stolen ?	70%	75%	70%	**85%**
Is there a wallet stolen ?	50%	70%	50%	**75%**

Table 4: Percentage of right answers to the questions

We do not discuss the answers to the task for longjump2 because the task is not discriminant enough to create differences in

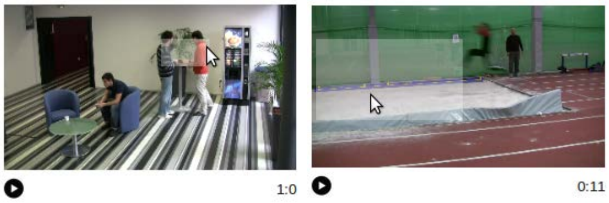


Figure 13: Example of ROIs emerging in RC+U: close-up on the wallets in coffeelounge and sand pit in longjump2.

the quality of answers when different interfaces are used. As the users have a very specific task in the longjump2 video, i.e to identify the length of the jump, the length of the jump is localized to the sand-pit. Hence users directly zoom into the region enclosing the sand-pit irrespective of the interface used. However consistent with coffeelounge task, we observed the emergence of a recommended viewport particularly suited to the task (see Fig 13). Conversely some recommended viewports have a low interest with respect to the assigned task and have not be selected by Users1. As a consequence, they disappeared in RC+U making our video interface adaptive and user-centric.

7. CONCLUSION

The main contributions of the paper are as follows. First, we have proposed the use of recommended viewports as a new way for users to interact with a zoomable video. A recommended viewport highlights interesting and relevant regions in the video and automatically tracks moving objects of interest, making it simple for users to click and zoom on interesting regions. Second, we proposed a hybrid approach to identify suitable recommended viewports automatically, by analyzing the content of the video and the user viewing pattern when watching the video. Our approach plays on the strength of both techniques, using usage analysis to crowdsource user's intention when viewing the video, and using content analysis to help with proper framing of the viewport around objects and with tracking of moving objects. Our approach can be viewed as one that uses implicit feedback from users as another modality to understand the content of the video.

Our work can be extended in many ways. Firstly, more sophisticated content analysis algorithms can be used to identify salient regions, track objects, and frame the viewport aesthetically. Secondly, our current research stops at learning from a set of 20 users. An open question is how to continuously crowdsource from users and adapt the importance map, and thus the recommended viewport. Thirdly, if we have multiple videos of similar types (e.g., multiple lecture videos from the same lecturer), then we can use the user interest map collected on one video to compute proper weights for linear combination of the different saliency features when analyzing the content of other videos. That is, we might be able to perform semi-supervised learning for computer vision algorithms using implicit feedback from users.

Identifying the important and relevance regions in a video has many other applications, beyond improving user interaction with zoomable video. The recommended viewports can be used to produce a retargeted video that automatically zooms and pans, essentially summarizing the content of the video to the users without interaction. Furthermore, we believe that the recommended viewports allows us to better predict what a user would like to watch during a given time in the video. Making such accurate predictions is crucial for prefetching of zoomable video during streaming, and to reduce the response time when users zoom and pan.

Acknowledgement

This research work is supported by National University of Singapore Academic Research Fund under the research grant WBS:R-252-000-368-112, and NExT Search Center, funded by the Singapore National Research Foundation & Interactive Digital Media R&D Program Office, MDA, under the research grant WBS:R-252-300-001-490.

8. REFERENCES

[1] C. Beleznai, B. Fruhstuck, and H. Bischof. Human tracking by fast mean shift mode seeking. *Journal of Multimedia*, 1(1):1–8, 2006.

[2] A. Carlier, V. Charvillat, W. T. Ooi, R. Grigoras, and G. Morin. Crowdsourced automatic zoom and scroll for video retargeting. In *Proceedings of MULTIMEDIA'10*, pages 201–210, Florence, Italy, 2010.

[3] A. Carlier, R. Guntur, and W. T. Ooi. Towards characterizing users' interaction with zoomable video. In *Proceedings of the 2010 ACM workshop on Social, adaptive and personalized multimedia interaction and access*, pages 21–24, Florence, Italy, 2010.

[4] K.-Y. Cheng, S.-J. Luo, B.-Y. Chen, and H.-H. Chu. Smartplayer: User-centric video fast-forwarding. In *Proceedings of CHI'09*, 2009.

[5] 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.

[6] A. Doan, R. Ramakrishnan, and A. Y. Halevy. Crowdsourcing systems on the world-wide web. *Commun. ACM*, 54:86–96, April 2011.

[7] P. Dragicevic, G. Ramos, J. Bibliowitcz, D. Nowrouzezahrai, R. Balakrishnan, and K. Singh. Video browsing by direct manipulation. In *Proceedings of CHI'08*, pages 237–246, Florence, Italy, 2008.

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

[9] D. B. Goldman, C. Gonterman, B. Curless, D. Salesin, and S. M. Seitz. Video object annotation, navigation, and composition. In *Proceedings of UIST'08*, pages 3–12, 2008.

[10] 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.

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

[13] S. Montabone and A. Soto. Human detection using a mobile platform and novel features derived from a visual saliency mechanism. *Image and Vision Computing*, 28(3):391–402, 2010.

[14] K. Q. M. Ngo, R. Guntur, A. Carlier, and W. T. Ooi. Supporting zoomable video streams via dynamic region-of-interest cropping. In *Proceedings of MMSys'10*, pages 259–270, Scottsdale, AZ, USA, 2010.

[15] M. Rubinstein, A. Shamir, and S. Avidan. Multi-operator media retargeting. *ACM Trans. Graph.*, 28(3), 2009.

[16] D. A. Shamma, R. Shaw, P. L. Shafton, and Y. Liu. Watch what i watch: using community activity to understand content. In *Multimedia Information Retrieval*, pages 275–284, 2007.

[17] F. Shipman, A. Girgensohn, and L. Wilcox. Authoring, viewing, and generating hypervideo: An overview of hyper-hitchcock. *ACM Trans. Multimedia Comput. Commun. Appl.*, 5(2):1–19, 2008.

[18] T. Syeda-Mahmood and D. Ponceleon. Learning video browsing behavior and its application in the generation of video previews. In *Proceedings of MULTIMEDIA'01*, pages 119–128, Ottawa, Canada, 2001.

[19] N. Ukita, T. Ono, and M. Kidode. Region extraction of a gaze object using the gaze point and view image sequences. In *Proceedings of ICMI'05*, pages 129–136, Torento, Italy, 2005.

[20] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *CVPR'01*, 1:511–518, 2001.

[21] Y.-S. Wang, H. Fu, O. Sorkine, T.-Y. Lee, and H.-P. Seidel. Motion-aware temporal coherence for video resizing. *ACM Trans. Graph.*, 28:127:1–127:10, December 2009.

[22] 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.