MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Content Aware Video Presentation on High-Resolution Displays

Clifton Forlines

TR2008-020 June 2008

Abstract

We describe a prototype video presentation system that presents a video in a manner consistent with the video's content. Our prototype takes advantage of the physically large display and pixel space that current high-definition displays and multi-monitor systems offer by rendering the frames of the video into various regions of the display surface. The structure of the video informs the animation, size, and the position of these regions. Additionally, previously displayed frames are often allowed to remain on-screen and are filtered over time. Our prototype presents a video in a manner that not only preserves the continuity of the story, but also supports the structure of the video; thus, the content of the video is reflected in its presentation, arguably enhancing the viewing experience.

AVI 2008

Content Aware Video Presentation on High-Resolution Displays

Clifton Forlines

Mitsubishi Electric Research Labs

Cambridge, MA 02139 USA

forlines@merl.com

ABSTRACT

We describe a prototype video presentation system that presents a video in a manner consistent with the video's content. Our prototype takes advantage of the physically large display and pixel space that current high-definition displays and multi-monitor systems offer by rendering the frames of the video into various regions of the display surface. The structure of the video informs the animation, size, and the position of these regions. Additionally, previously displayed frames are often allowed to remain on-screen and are filtered over time. Our prototype presents a video in a manner that not only preserves the continuity of the story, but also supports the structure of the video; thus, the content of the video is reflected in its presentation, arguably enhancing the viewing experience.

Author Keywords

Video playback, digital video, entertainment technology.

ACM Classification Keywords

H5.1. Information interfaces and presentation (e.g., HCI): Multimedia Information Systems - *video*.

1. INTRODUCTION

Despite the large number of hours that a typical person spends watching television and videos each year, little research exists within the CHI literature on improving and understanding video consumption, with some notable exceptions [1][2]. In recent years, personal video recorders, peer-to-peer file sharing, and portable video devices have begun to change the way that consumers interact with digital video. While televisions, projectors, and computer monitors have become physically larger and capable of displaying an increased number of pixels, the manner in which videos are displayed on these surfaces has remained the same. When creating new content for these devices, creators can choose to take advantage of these high-resolution displays; however, videos originally produced for smaller displays are simply scaled up to fill larger displays. Little is done to take advantage of a large display surface or a multi-display device. For example, a high-definition computer monitor, with a resolution of 1600 x 1200 pixels, displays a standard definition television signal, with a resolution of 640 x 480 pixels, by simply scaling the low-resolution video to fill the high-resolution display.

AVI'08, 28-30 May , 2008, Napoli, Italy

Typically, each frame of a video is displayed in place of and covers the entirety of the previous frame. One assumption that conventional video players make is that they should never display more than one frame from the same video at any one time. A similar assumption is that they never display the same frame from a video in multiple locations on the screen during playback. Finally, they never move the presented content around the large display space.

Our proof of concept prototype is an example of *Content Aware Video Presentation*. It converts an input video to an output video with the aim of challenging the above assumptions about video playback for the purpose of improving the experience. We take advantage of the increased pixel and physical size of large displays that modern computers and high-definition televisions have to offer. The input video can be thought of as a series of frames that are normally displayed sequentially. The output video is the same series of frames that have been scaled, rotated, filtered, and displayed in parallel on different regions of the display(s) in a manner that not only preserves the continuity of the story, but also supports the structure of the video.

The manner in which the frames are selected, the length of the frames, and the treatment of previously displayed frames are based on the structure of the input video. We determine the structure by using a variety of known techniques from the fields of video and image processing in conjunction with a new method for scene detection to find the relationship between shots, the content of individual shots, and camera motion. By displaying the frames of a video in this manner, the context of the video is reflected in its presentation, and the viewing experience is arguably enhanced.

We look toward other media, such as music and visual arts, which have both accepted the presentation of another's work in alternative forms, as a justification for our prototype. The techniques described in this paper are needed to explore this new style of video presentation and to determine if such alternative video presentation methods are desirable for high-resolution displays and non-traditional consumption of video content.

Figure 1. A frame from a Content Aware Video. The current shot is displayed in the foreground while the final frames from previous shots remain in the background (courtesy of NASA).

2. RELATED WORK

Several attempts at improving the consumer's viewing experience through understanding the characteristics of the video have been explored.

Boreczky et al. [1] presented a technique for summarizing video that extracted keyframes from the video and scaled these images according to their importance. The differently-sized images were then packed together in a comic-book like layout. Viewers were presented with a graphical overview of the video and could navigate to an interesting part by clicking on any keyframe in the layout. While participants in a study were not able to find specific parts in a video faster using this layout than when using other summarization methods, participants did express a preference for the comic-book like technique.

Philips recently introduced "Ambient Light Technology" (Ambilight) for televisions. Ambilight illuminates the wall behind the television with backlight, and adjusts the brightness and color of this light based on the qualities of the frame currently being displayed on the television. Philips claims that this backlighting aids the visual perception system and enables the human eye to perceive more picture detail, contrast, and color. By filling the periphery of the viewer's vision with content, the designers of Ambilight hope to create a more immersive viewing experience.

Mitsubishi Electric recently released a DVD Recorder [9] that provides a "highlight playback" feature for sporting events. Highlights are extracted from the video during recording by analyzing the audio channel and looking for a characteristic mixture of cheering and the commentator's excited speech. Each second of the program is assigned an importance level, and the interface enables the user to set an importance threshold so only the portions of the program that exceed the threshold are played. The length of the summary corresponding to the choice of threshold is displayed, and the user can choose a desired summary length by moving the threshold up or down as needed.

Whittenburg et al. [15] presented an interface that used rapid serial visual presentation of the individual frames from a recorded program to support fast-forwarding and rewinding through video. Using this technique, the frames from the video are presented in a 3D trail leading away from the viewer, and upcoming shot changes are clearly visible when looking at the trail. By seeing the location of these changes and some of the details from upcoming frames, a viewer is better able to rapidly traverse to a desired location in the video.

Shamma et al describe an interesting use of the closed-captioning included in a television broadcast [11]. In their multi-display environment, while a video is being displayed on the main monitor, a background process is decoding the closed-captioning stream from the input video and using the words that the viewer is listening to as query terms for image searching. The results of these searches are displayed on the surrounding monitors, and the viewer is thus presented with a carousel of auxiliary material related to the video. These images provide context for the main program.

Fan et al [5] describe their approach to viewing video clips on limited resolution small screen devices. In one sense, they are addressing the opposite problem that we are. They detect the saliency of different objects in a frame of the video by measuring the local contrast. Once the interesting objects are identified, the player can zoom into that region of the video, allowing for an optimal use of the limited display space.

Many steps in our technique rely upon related work in video analysis, image comparison, and camera motion reconstruction. We will describe this work in the following sections as we describe the steps in our technique.

3. SYSTEM OVERVIEW

Figure 2 shows a high-level overview of our content aware video presentation prototype. The input to this system is a video and the output is a converted video. The output video has a resolution and aspect ratio that fills the entire high-resolution display space of the computer monitor(s) or television on which the video is to be viewed.

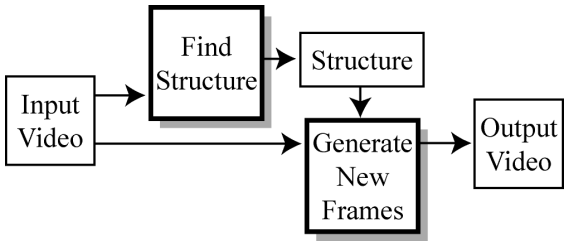

Figure 2. System overview. First, the structure of the video is found. Second, a new video is rendered from the frames of the original video and the structure.

This conversion has two high-level stages. In the first stage, we analyze the video to determine its structure (shot boundaries, related shots, scenes, camera motion, etc.). In the second stage, we use this structure to render a new output frame for each input frame in the original video.

For the purpose of describing our technique, we will commonly refer to a video that includes a scene in which two people are talking to one another (a very common scene in videos). Conventionally, shots in this type of scene alternate between close-ups of the two individuals with occasional overview shots showing both actors. The scene may begin with a foundation shot showing the location in which the conversation between these two characters is taking place. Our prototype converts this alternating sequence of shots into a video in which both actors remain on screen for the entirety of the scene.

This two-person conversation is just one of the many scene structures that occur repeatedly across different videos, and a detailed description of the many other common structures one observes across videos is outside the scope of this paper. We hope that the reader will see how the specific instances described in this paper can generalize to many types of scenes with many different patterns of shots.

4. STAGE 1 – VIDEO STRUCTURE

Video is often described as having a four tier hierarchical structure, as shown in Figure 3. A video is composed of one or more scenes, each of which includes one or more 'shots', each of which includes one or more frames. A 'shot' is a sequence of frames taken by a single camera over a continuous period of time. The shots are separated by shot boundaries.

Figure 4 shows an overview of how we determine the structure of the input video. We use a variety of previously known techniques in our system from the field of video analysis to determine this structure. The video is first segmented into shots by detecting shot boundaries. Each shot is compared to previous shots in the video to detect sets of visually similar shots, for example a series of shots of the same person or object. Similar shots are combined to

form shot 'chains'. Chains that overlap in time are combined to create scenes. In a side step, camera motion, which is present in many but not all shots, is estimated from the motion vectors of the video. In this section, we will describe each of these steps of Stage 1 in detail.

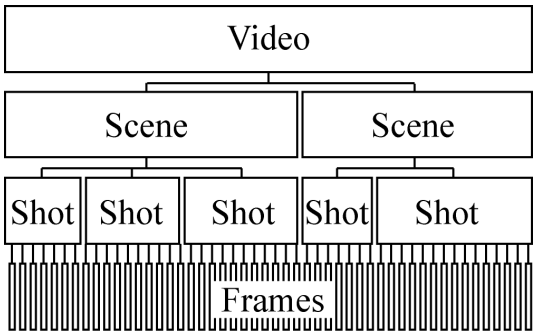

Figure 3. The four tier hierarchical structure of video. The entire video (top) is composed of a series of non-overlapping scenes, each of which is a series of camera shots, which are each composed of a series of individual frames.

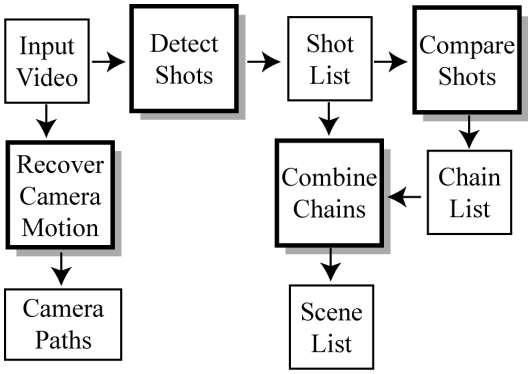

Figure 4. Overview of Stage 1, in which the structure of the video is detected. The shots are detected in the input video, and then these shots are compared to one another to find chains of visually similar shots, which are then combined into scenes. Separately, the camera motion is recovered for each frame of the video.

4.1 Shot Detection

A shot is defined as a continuous series of frames captured by one camera in a single continuous action in time and space. A number of processes are known for segmenting videos into shots by detecting shot boundaries. The methods can be based on color-histogram comparison, pixel differences, encoded macroblocks, and changes in detected edges between consecutive frames.

Lienhart [7] provides an excellent overview and comparison of several shot boundary detection techniques. Cabedo et. al [3] compared several shot detection techniques based on color-histogram comparison. These techniques are similar to one another in that they all create a histogram for two consecutive frames in the video, and then compare these histograms for dissimilarity.

Another promising new method for detecting shot boundaries in a video is presented by Cernekova et. al [4]. Their technique uses the joint entropy between frames to detect cuts, fade-ins and fade-outs. They presented an experiment in which their technique more accurately differentiate fades from cuts, pans, object or camera motion and other types of video scene transitions than previously known techniques.

All of these processes are similar in that they compare adjacent frames to detect when there is a significant difference between the frames that is indicative of a shot boundary. Any technique, or combination of techniques, that produces a list of shots from an input video is compatible with our system.

Our prototype system uses a modified color-histogram comparison algorithm. We first construct a color histogram for each frame of the input video. Each histogram has 256 bins for each RGB color component. We compare the histograms of adjacent frames as follows.

For each of the three color components, we sum the absolute differences between the values for each corresponding pair of bins giving us total differences for red, green, and blue between two frames. Each of the three total differences is compared with the average difference for the respective color for the previous five pairs of frames. If the difference for any of the three colors is greater than a predetermined threshold value times the average difference for that color, then a shot boundary is detected. To handle errors in an encoded video, shots that include fewer than five frames are combined with the following shot. The input of this step is the frames of the input video; the output is a list of shots.

It is worth repeating that our method of shot detection is not presented as a novel contribution to the field, but rather as a means toward an end. Any technique that produces a list of shot boundaries within an input video is compatible with our prototype.

4.2 Scene Detection

While a list of shots is a good first step in understanding the structure of a video, this list does not provide enough understanding for content aware playback. It is as if we have divided all of the words in a book into paragraphs, but have not yet divided these paragraphs into chapters. Further analysis is needed.

A scene, as in our example scene of two characters talking, is typically a contiguous sequence of shots that are logically related according to their content. Scene detection within videos is an active area of research. Yeung et al [16] introduced not only a pioneering piece of scene segmentation work, but also a means to visualize a video's structure. Zhao, et al [17] present an overview of the two major approaches for grouping shots together into scenes. The first approach looks at the boundary of shots and labels a shot boundary as a scene boundary if the visual and aural content change simultaneously. Lu, et al [8] present a scene detection technique that measures the continuity of visual, aural, and textual (closed captioning) elements in a video, and labels a shot boundary as a scene boundary when these continuities drop. The second approach compares the similarity between two shots by looking at the similarity of the shots' frames as a whole. Variations of this approach use different frame similarity measurements similar to the frame comparison metrics described in the previous section.

While many methods for scene detection exist in the literature, the uses of scene detection are less varied. For the most part, the output of a scene detection algorithm is used for indexing and summarization. Because our technique uses scene structure for playback, our needs are different. We need to not only gain an understanding of scene segmentation, but also an understanding of the relationships among the shots within a scene. We need an understanding of the chains of related shots that exist within a single scene.

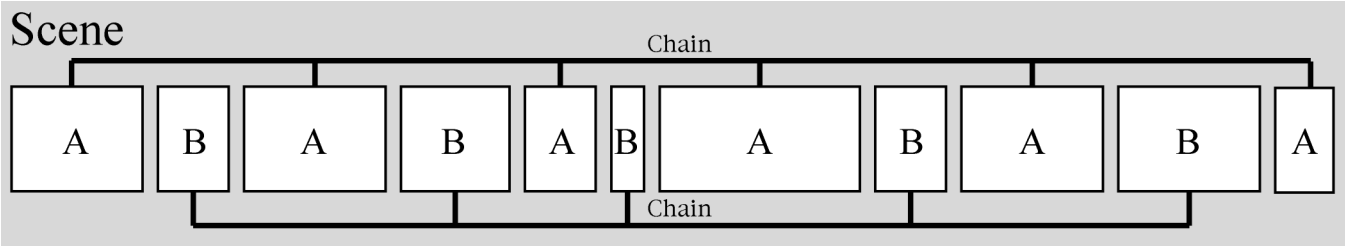

Figure 5. This figure shows a series of shots that have been grouped together into two overlapping chains of shots. The width of a shot is indicative of its length. One chain is all of the similar shots of one character talking, and the other chain is all of the shots of another character talking. Because these chains overlap in time, they are grouped together into a scene for the output video.

Our prototype uses a two step approach to scene detection. In the first step, we find chains of related shots within the video. In the second step, we combine these chains into scenes.

4.2.1 Step 1 - Finding "Chains" of Related Shots

For comparing the similarity of shots, our prototype again uses color histograms. We compare the first frame in a current shot with the last five frames of each of the previous five shots in the manner described in the 'Shot Detection' section of this paper, only using a more relaxed threshold. If a shot begins with a frame that is visually similar to the last five frames of a previous shot, then the shots are likely to be of the same person or object. A chain of shots is created whenever two or more shots are found to be visually similar. Chains can include many shots, and the similar shots in a chain do not need to be contiguous in time.

Any technique or combination of techniques that produce a chain of visually similar shots that are located relatively close together in time is compatible with our technique.

4.2.2 Step 2 - Combining Chains into Scenes

Figure 5 shows a series of shots in a video, which have been grouped into chains as described in the previous section. In this example, there are two chains, A and B. One chain is all the similar shots of one character talking, and the other chain is of all of the similar shots of another character talking. Because these chains overlap in time, we group them together into a scene for the output video.

Figure 6 shows a more complex example. In this figure, we see a series of shots, containing six chains, two and four of which overlap into two scenes.

Of course, not every shot is part of a chain, and we refer to these shots as orphans. Orphans that lie between the first and last shot of a scene and are not included in a chain are added to that scene (Figure 6, *left*). This shot is visually unrelated to its close neighbors, and is often an overview shot of the area surrounding the action taking place or a shot of both the subject of chain A and B. Orphans that are surrounded on either side with a scene are added to the trailing scene (Figure 6, *right*). In our experience, this type of orphan shot is almost always a foundation shot, in which the director tells the audience where the scene is taking place. In this case, the orphan may be a shot of the outside of the building in which the conversation between A and B is about to occur.

4.2.3 Handling Errors in Scene Detection

It is worth noting that errors in scene segmentation are less problematic for our task than for traditional scene segmentation tasks such as indexing and summarization. A summary that cuts off the last shot of a scene will leave the viewer wondering about the resolution of the story; on the other hand, because our technique is intended for playback, the result of a misplaced scene boundary is simply a less-than-ideal layout of the shots within the scene; in fact, when testing our prototype with users, such segmentation errors often passed unnoticed as viewers became engrossed in the story. Our method for scene detection is motivated by our use of scene detection and our need for the chaining of similar shots within a scene. A comparison between the performance of our method of scene detection and other methods is outside of the scope of this paper.

4.3 Estimating Camera Motion

Estimating camera motion with video analysis is another active research area. Videos encoded according the MPEG standard include motion vectors in B-frames and P-frames, and a number of techniques are known for estimating camera motion from the motion vectors.

Jones et al [6] describe a technique for stitching together the frames from a video into a single mosaic image. In the appendix of this work, the authors detail how they reconstruct camera panning and zooming through calculating the average of all motion vectors from the macro blocks within each frame of an MPEG video. Pilu describes a camera motion reconstruction technique [10] in which

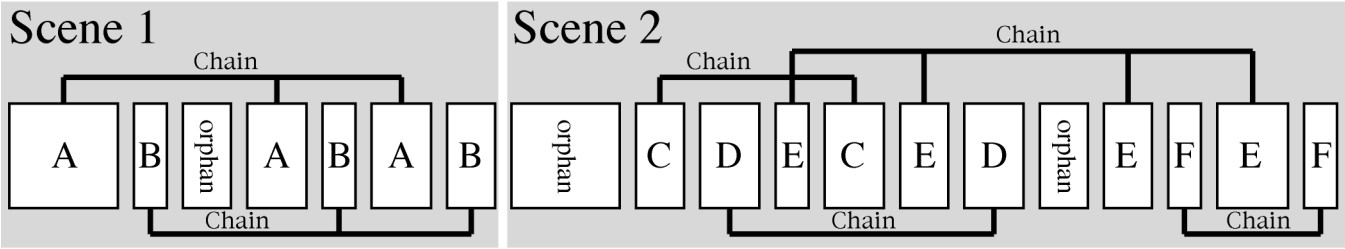

Figure 6. This figure shows the structure of two scenes. The scene on the left contains two overlapping chains of shots. The orphan shot in the middle of these two chains is added to the scene. The scene on the right contains four overlapping chains. The orphan shot in the middle of these chains is added to the scene, and the orphan shot that lies in-between these two scenes (which is most likely a foundation shot) is added to the scene on the right.

he first weights the motion vectors of each macro block by their reliability in predicting motion and then fits the filtered motion vector field to common velocity fields for common camera movements. Both of these techniques are appealing in that they effectively piggyback on an already occurring process (the presence of motion vectors for the purpose of video compression) to provide a computationally inexpensive means of reconstructing camera motion. Other techniques for estimating camera movement include feature based tracking [10] and optical flow [12].

Our prototype parses motion vector data directly from the input video, which is encoded according to the MPEG-2 standard. For each frame in a shot, the variance for the X-Y motion is determined for all of the motion vectors. If the variance is below a predetermined threshold, then the average motion for all motion vectors is recorded. In other words, if the most of the motion vectors for a single frame are all more or less pointing in the same direction, then we assume that the camera is moving in the opposite direction and we record the motion. If the variance is above the threshold, then we record a vector of length zero. Currently, our prototype only handles camera panning, but others have reconstructed zooming and rotation and the detection of these types of camera movement are left for future work.

In this manner, we produce an average motion vectors for each frame in the video. These camera paths are used when we render the converted video in the second stage.

5. STAGE 2 – RENDERING NEW FRAMES

Figure 7 shows an overview of Stage 2, in which our technique generates the new frames of the output video. The input for this stage is the original input video and the chains, scenes, and camera paths from Stage 1. In this second stage, for every scene in the input video, a new scene of equal length is rendered. Finally, these scenes are combined along with the audio tracks from the input video into the output video.

5.1 Templates for Frame Layout

For each scene in the list of scenes, we compare the structure of that scene to predetermined templates in order to select a most appropriate rendering for the frames of that scene. By structure, we mean the number and pattern of chains in the scene, the presence of shots in the scene not included in a chain, the length of the chains, and the amount of overlap of the chains of a scene. The templates are ranked based on how closely the characteristics of the scene match an ideal scene represented by the template. Our prototype then uses the template that most closely matches the scene to render a new image for each frame in that scene.

As shown in Figure 8, each template initially generates a blank image that is the size and has the aspect ratio of the high-resolution display on which the output video will be viewed. Then, the first frame from the input video is rendered into a region of the blank image, perhaps filling the entire image. This image is then saved as the first frame of the new scene in the converted video. While there are frames remaining in the input scene, the next frame from the input video is rendered into a new region of the image. The region that this next frame is drawn into may or may not overlap the previous region, and the previous image may or may not be cleared of content.

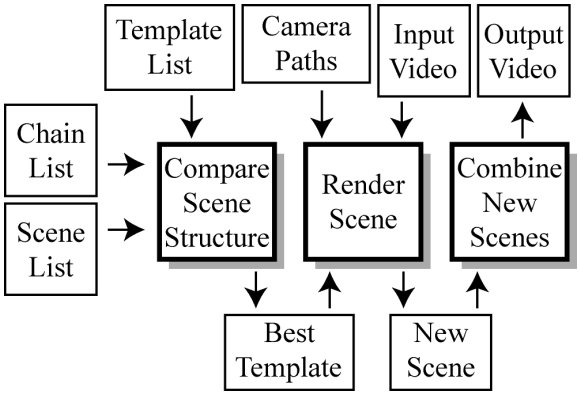

Figure 7. Overview of Stage 2, in which a new frame is rendered for each frame of the input video. For each scene in the input video, the structure of that scene is compared to a list of templates. The best matching template then renders a converted scene using the frames from the original video, and optionally the recovered camera motion. Finally, these scenes are combined into the output video.

As shown in Figure 9, the example scene includes two characters talking to one another. In the original video, the shots alternate sequentially between the two characters as they speak, with no one shot showing both people. The template for rendering this scene renders each frame from the first chain into a region on the left side of the screen, and each frame from the second chain into a region on the right side of the screen.

The result is a sequence of images in which the talking characters appear on the left and right side of the images. During playback, a viewer of this sequence of images alternately sees the actively talking character in either the left or the right region, and the non-speaking character displayed as a still frame in the other region. The still frame corresponds to the last frame of the shot in which that character is talking.

Some templates filter the previous frame from the output video before drawing the current frame. In a variation of the two-person conversation example, the still frame on the right can slowly fade

Figure 9. The top row shows the first frame from five consecutive shots in the original input video. The bottom row shows five rendered frames from the output video. The frames from the five shots above are painted into either the left or the right region of the screen. The final frame of the previous shot remains frozen on screen on the opposite side.

Figure 10. The top row shows four frames from the first shot in this scene followed by the first frame from the second shot in this scene. The dotted line indicates the shot boundary between these two shots. The bottom row shows the animating region of the screen that the template rendered the original frames into. The effect presented in the converted output video is that the shot begins playing back full screen, and then slowly animates to fill only the left region of the screen. The second shot is then displayed in the right region of the screen, and the final frame from the first shot remains frozen on the left.

to black while the active shots on the left continues, until the still frame on the right becomes an active shot again, and the left region shows a slowly fading still frame.

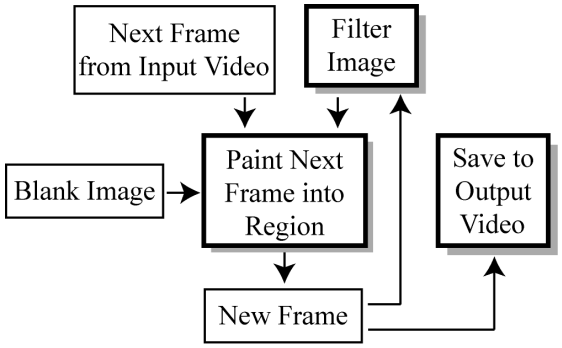

Figure 8. Templates recursively paint frames from the input video into regions of the output video. The background of these new frames is the previous frame from the output video, which may be filtered.

In addition to a simple fade, any number of conventional image filtering techniques can be used. Still frames can reduce their color saturation over time, i.e., change into a black-and-white image, or can be blurred, pixilated, or converted to a sepia tone.

Some templates are designed to animate regions of the output images into which frames from the input video are rendered. Figure 10 (*bottom row*) shows five consecutive output images generated by the template. The template used to render this scene renders each frame from this shot into an animating region. Note that the regions vary in size and location to give the effect of animation. In addition to varying the size and location of the region over time, templates could distort, rotate, and/or reflect the original video frames.

As shown in Figure 11, a template can animate the region into which frames are painted according to the stored camera motion described in the previous section of this paper. In this example, the camera pans from left to right across the scene to reveal a boat that is originally off-camera, right. Therefore, the region into which frames from the input video are rendered moves across the screen, animating according to the camera path. The reader will recall that each frame of the shot has a 2D vector associated with its movement, in pixel units. Each frame in the shot is translated by the summation of all the vectors up to an including that frame, and then all of these translated frames are combined into a single image. A scale factor is then computed by examining the ratio between the size of the composite image and the size of the output video. This scale factor is then applied to the 2D vectors and used to resize the input frames as they are rendered into an animating region of the output video. In this way, as much of the area of the output video is used as possible.

5.2 Creating the Output Video

After a template is matched to each scene in the input video and a new output frame is rendered for each frame in the input video, the rendered images are arranged sequentially and encoded according to the MPEG-2 standard to produce the output video. Our prototype then copies the unchanged audio track from the input video. The output video is now ready for playback on the high-resolution computer monitor, high-definition television, or multi-monitor system using a conventional video playback device.

6. LAYOUT DESIGN GUIDELINES

In building example templates, we have come up with several guidelines to follow when designing new templates for layout.

6.1.1 Time is constant

The first constraint on template design is that the input and output scenes must be equal in length. Cutting pieces out of the original video may drastically affect the story. Speeding up or slowing down portions of a video may be desirable in some situations, but

Figure 11. This row shows five frames from the output video. The original five frames were taken from a shot in which the camera panned across a large room to reveal a boat on the right. The template charged with rendering this scene used the recovered camera motion for this shot to inform the animation of the region into which these frames were painted. The effect in the converted output video is that the content of the shot (the large room) is remaining stationary and that the animating frame is providing a keyhole like view into the room. In the background, we see the final frame of the previous shot fading to black.

we did not see a clear mapping between scene structure and story pacing. Changing the speed of playback also makes the audio track less recognizable. One exception to this guideline that works well in some situations is shot repetition. A template that recognizes an important shot may present it multiple times in succession, perhaps altering the size or scale of the frames.

6.1.2 Current frame is shown in entirety

An early template that we designed animated a shot from an off-screen location, which ended up hiding an important feature of the shot from the viewer. Similarly, another early template presented shots in such as way that they sometime appeared partially occluded by a previous shot shown in another location on the screen. These observations led us to the guideline that, while a template may scale or filter the current frame, the entirety of the current frame is always visible in some region of the screen.

6.1.3 Never show frames ahead of the current time

Many templates leave frames from previous shots on screen, often to create a background content for the currently displayed shot. When we experimented with showing frames from upcoming shots along with the current shot, we began to violate the cause-and-effect relationship between sequential shots. Showing effect before cause was extremely disorienting in many cases (although oddly intriguing in a few cases). This observation led us to the guideline that templates should never show frames from upcoming shots, only from previous ones or the current one.

7. Future Work

A better understanding of the variety of scene structures that occur in commercial programming is needed to generate a more complete list of templates. Our prototype was designed with the adding of templates in mind – and we plan on adding more templates to translate different types of scenes in a content appropriate manner. Analysis of the contents of the frames themselves can be useful in informing frame layout. In this section, we lay out a means by which the frames within a scene can aid in frame layout in the converted video.

7.1.1 Gaze Direction Detection

Layout templates could use a gaze direction detection process on the frames in each of the chains. A number of techniques are known for estimating gaze direction of faces in images [13]. Such a process would recognizes that the woman in Figure 7 is facing to the right and that the man in Figure 7 is facing to the left. The frames in the chains can then be combined so that the two characters appear to face one another.

The gaze direction of characters can also inform the system as to the "angle" of the shot. By "angle" we mean the relationship between the viewer and the characters, a relationship that tells us something about the intent of the content creators. A "high-angle" shot is one in which the camera is above the eye-level of the subject. The consequence of this point of view is that the character appears small and weak. A "low-angle" shot has the opposite effect. By pointing the camera up at the character, the character appears powerful and large. A template that could classify shots as high, neutral, or low-angle shots could use this information to present shots accordingly – growing low-angle shots to fill the screen, thus enhancing certain characters' power and presence and shrinking high-angle shots to amplify other characters' weakness.

7.1.2 Face Detection and Recognition

Robust face recognition would greatly aid in the finding of chains of related shots. Knowing that a specific character is present in nearby shots would suggest that the shots are part of the same scene. Unfortunately, robust face recognition is an open research area without established techniques that work well in various lighting conditions. Advancements in face recognition should complement our prototype as they arise.

While robustly recognizing specific faces is an unsolved problem, techniques exist that provide robust face detection [14]. These techniques do not provide information about who is present in the frame; however, they do provide information about the presence or absence of faces, as well as the number of faces and the relative size of these faces within the image. Knowing the number of faces in a shot could help in the layout of a scene. For example, a scene containing three chains of shots, two of which have one face and one of which has two faces probably contains a conversation between two people. The chains with one face are close-ups of the two people, and the chain with two faces is an overview shot of the two characters together. A template recognizing this pattern could render the close-up shots of the left and the right side of the screen, and render the overview shots in the middle.

Knowing size of faces within the frames of a shot could be very helpful in informing a template as to the "length" of the shot. By "length" we are not referring to the duration of a shot, but rather the length of camera lens, which relates to the depth of focus. A very small face would indicate a "long" shot, one in which a character is shown in relation to their surroundings. A face that fills are large portion of the screen would indicate a "short" or "close-up" shot. Since a close-up is used to show the physical details of the actor's face, and gain an understanding of how the character feels or to clarify an action, templates would want to render this type of shot into a large region to preserve this detail.

The ordering of shot lengths could also inform the layout. For example, a medium or long shot that is followed by a close-up is probably a two shot sequence meant to first show the context of a person, and then show the details. A template recognizing such a structure would want to render the medium or long shot into a full screen region, and then leave the final frame of this shot on screen as it renders the close-up into an overlapping region of the display.

7.1.3 File Format

A variation of our prototype could generate an XML file rather than a second video file. A modified player application would read from the original video file as well as this XML file, which would include the region on screen into which the current frame would be painted as well as descriptions of any filtering that should take place during each frame of the video. This variation would have the benefit of requiring much less disk space; however, playback would become a more computationally expensive operation.

7.1.4 Audio Structure

All of the structure that our prototype uses to inform the layout of frames comes from examining the video track of the original video. The audio track(s) and closed-captioning are copied unchanged to the converted video and are not used to inform the structure. Certainly, a more sophisticated version of our prototype would examine the audio tracks for content aware presentation.

8. EARLY REACTIONS

We cannot conclude a paper on a new method for video presentation without some discussion of the question of whether or not such a presentation is desirable. To begin answering this question, we have presented several videos generated by our prototype to many coworkers, colleagues, guests of our lab, television manufacturers, and members of the content creation

industry. Reactions have been mixed, but almost everyone seems to have a strong opinion, either enthusiastic or uneasy.

The most common positive adjective has been “fun”. Several people mentioned that this type of presentation might be a way to enjoy previously-viewed programming in a new way. Several viewers have stated that viewing a video in a context aware manner makes video watching a more active experience as they follow the sequence of scenes around the screen. The television manufacturers that we spoke with recognized this increase level of activity, and expressed concern that their customers might not want to maintain a high-level of mental activity when watching videos. There was a concern that context aware video playback “could wear the viewer out” by “over engaging them.”

Not too surprisingly, the members of the content creation industry expressed uneasiness with the idea of presenting another person’s work in an alternative fashion. When questioned about their uneasiness, the most common source was the feeling that the time and effort put into the original by the director and cinematographer were being disregarded by altering the presentation of the video. In our defense, we look to other media and the means in which people derive art from other people’s art. With visual art, we see an analogy between our prototype and the art of collage. A collage draws upon a multitude of previous pieces and combines parts into something new. While the viewer may recognize the source of the pieces within a collage, he never confuses the pieces with the original. Similarly, musical pieces are often covered by other artists, and even sampled, filtered, looped, and remixed to create derivative work.

9. CONCLUSION

We have presented a prototype video presentation system that presents a video in a manner consistent with the video’s structure. By reconstructing scene structure through shot detection and shot comparison, we take advantage of the large display and pixel space that current high-definition computer monitors and televisions provide by displaying shots from the video in various locations on the display. By reflecting the content of the video in its presentation, we hope to add to the viewing experience.

REFERENCES

1. Boreczky, J., Girgensohn, A., Golovchinsky, G., and Uchihashi, S. 2000. An interactive comic book presentation for exploring video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands, April 01 - 06, 2000). CHI '00. ACM Press, New York, NY, 185-192.
2. Brown, B. and Barkhuus, L. 2006. The television will be revolutionized: effects of PVRs and filesharing on television watching. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Canada, 2006). CHI '06. ACM Press, New York, NY, 663-666.
3. Cabedo, X.U. and Bhattacharjee, S.K., Shot detection tools in digital video. In *Proceedings of Nonlinear Model Based Image Analysis 1998* (Glasgow, July 1998). Springer Verlag, 121-126.
4. Cernekova, Z., Nikou, C., and Pitas, I. Shot Detection in Video Sequences Using Entropy-Based Metrics. International Conference on Image Processing 2002 (ICIP2002), Vol. 3, 421-424.
5. Fan, X., Xie, X., Zhou, H., and Ma, W. 2003. Looking into video frames on small displays. In *Proceedings of the Eleventh ACM international Conference on Multimedia* (Berkeley, CA, USA, November 02 - 08, 2003). MULTIMEDIA '03. ACM Press, New York, NY, 247-250.
6. Jones, R. C., DeMenthon, D., and Doermann, D. S. 1999. Building mosaics from video using MPEG motion vectors. In *Proceedings of the Seventh ACM international Conference on Multimedia (Part 2)* (Orlando, Florida, United States, October 30 - November 05, 1999). MULTIMEDIA '99. ACM Press, New York, NY, 29-32.
7. Lienhart, R. Comparison of Automatic Shot Boundary Detection Algorithms. In Image and Video Processing VII 1999, Proc. SPIE 3656-29, Jan. 1999.
8. Lu, X., Ma, Y.-F., Zhang, H.-J. and Wu, L., An Integrated Correlation Measure for Semantic Video Segmentation. In *Proceedings of IEEE International Conference on Multimedia and Expo – ICME ‘02*, (Lausanne, Switzerland, 2002), 57- 60.
9. Mitsubishi Electric. RakuReko DVD Recorder. http://www.mitsubishielectric.co.jp/news/2006/0601.htm
10. Pilu, M., On Using Raw MPEG Motion Vectors To Determine Global Camera Motion, Digital Media Department, HP Laboratories, August 1997.
11. Shamma, D. A., Owsley, S., Hammond, K. J., Bradshaw, S., and Budzik, J. 2004. Network arts: exposing cultural reality. In *Proceedings of the 13th international World Wide Web Conference on Alternate Track Papers &Amp; Posters* (New York, NY, USA, May 19 - 21, 2004). WWW Alt. '04. ACM Press, New York, NY, 41-47.
12. Teodosio, L. and Bender, W. 1993. Salient video stills: content and context preserved. In *Proceedings of the First ACM international Conference on Multimedia* (Anaheim, California, United States, August 02 - 06, 1993). MULTIMEDIA '93. ACM Press, New York, NY, 39-46.
13. Varchmin, A. C., Rae, R., and Ritter, H. 1998. Image Based Recognition of Graze Direction Using Adaptive Methods. In *Proceedings of the international Gesture Workshop on Gesture and Sign Language in Human-Computer interaction* (September 17 - 19, 1997). I. Wachsmuth and M. Fröhlich, Eds. Lecture Notes In Computer Science, vol. 1371. Springer-Verlag, London, 245-257.
14. Viola, P. and Jones, M.J. Robust Real-Time Face Detection. International Journal of Computer Vision, 57 (2). 137-154.
15. Wittenburg, K., Forlines, C., Lanning, T., Esenther, A., Harada, S. and Miyachi, T., Rapid serial visual presentation techniques for consumer digital video devices. in *Proceedings of the 16th annual ACM symposium on User interface software and technology*, (Vancouver, Canada, 2003), ACM Press, 115-124.
16. Yeung, M. M. and Yeo, B. 1996. Time-Constrained Clustering for Segmentation of Video into Story Unites. In *Proceedings of the international Conference on Pattern Recognition (ICPR '96) Volume Iii-Volume 7276 - Volume 7276* (August 25 - 29, 1996). ICPR. IEEE Computer Society, Washington, DC, 375.
17. L. Zhao, W. Qi. Y.J. Wang, S.Q. Yang, and H.J. Zhang. Video Shot Grouping Using Best-First Model Merging. Proc. 13th SPIE symposium on Electronic Imaging - Storage and Retrieval for Image and Video Databases, SPIE vol. 4315, pp.262-269. Jan. 2001.