

# Vremiere: In-Headset Virtual Reality Video Editing

Cuong Nguyen<sup>1</sup> Stephen DiVerdi<sup>2</sup> Aaron Hertzmann<sup>2</sup> Feng Liu<sup>1</sup>

<sup>1</sup>Portland State University  
Portland, OR, USA  
{cuong3, fliu}@pdx.edu

<sup>2</sup>Adobe Research  
San Francisco, CA, USA  
{diverdi, hertzmann}@adobe.com

## ABSTRACT

Creative professionals are creating Virtual Reality (VR) experiences today by capturing spherical videos, but video editing is still done primarily in traditional 2D desktop GUI applications such as Premiere. These interfaces provide limited capabilities for previewing content in a VR headset or for directly manipulating the spherical video in an intuitive way. As a result, editors must alternate between editing on the desktop and previewing in the headset, which is tedious and interrupts the creative process. We demonstrate an application that enables a user to directly edit spherical video while fully immersed in a VR headset. We first interviewed professional VR filmmakers to understand current practice and derived a suitable workflow for in-headset VR video editing. We then developed a prototype system implementing this new workflow. Our system is built upon a familiar timeline design, but is enhanced with custom widgets to enable intuitive editing of spherical video inside the headset. We conducted an expert review study and found that with our prototype, experts were able to edit videos entirely within the headset. Experts also found our interface and widgets useful, providing intuitive controls for their editing needs.

## ACM Classification Keywords

H.5.1 Information Interfaces and Presentation: Multimedia Information Systems

## Author Keywords

Virtual reality; video editing

## INTRODUCTION

Virtual Reality (VR) video is emerging as a new medium for shared, creative experiences. VR video specifically means spherical panorama (full or partial) video viewed within a head-mounted display. Filmmakers use multi-camera rigs to capture spherical videos and then edit them with standard video production software such as Adobe Premiere or Apple Final Cut Pro. However, these tools only support editing VR video in its flattened equirectangular projection format (Figure 1). One prominent director describes her first experience with editing spherical video:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

CHI 2017, May 06-11, 2017, Denver, CO, USA  
© 2017 ACM. ISBN 978-1-4503-4655-9/17/05\$15.00  
DOI: <http://dx.doi.org/10.1145/3025453.3025675>



(a) View in the VR headset

(b) View on desktop

Figure 1: The same shot, viewed in a headset versus on the screen. VR views look very different than on the desktop screen, and provide a much stronger sense of immersion. © TOYO TIRES JAPAN

The first assembly went as well as expected. Rough in parts. A few nice visual match-cuts. Music with layers that gently guided the visuals. Compositionally sound, wides to close-ups and back again. ... Overall, I felt the edit had a lot of promise, and I was quite pleased with myself.

And then I watched it in a headset.

I don't think I'll ever be able to truly describe how much the edit sucked in a headset... [4]

This anecdote illustrates a major problem of editing VR video: viewing VR video in a headset is a dramatically different experience from viewing it on a desktop display. This difference manifests in many ways. In VR, a viewer sees only a portion of the environment at any time and must decide where to turn their head to view; the filmmaker cannot assume the viewer will look in a particular direction at a given time. VR video is immersive and takes time to absorb, which affects the pacing of cuts. Conventional video edits such as zooming and jump cuts can be jarring and confusing in VR. Editors must consider the relative spherical rotation of adjacent cuts. They must also deal with the potential strain, fatigue, and nausea of VR both for the editor, and for the intended viewer. At present, VR editors must follow an onerous workflow, alternating between editing on the desktop and then putting on a headset to review changes. Being able to perform WYSIWYG editing directly in the headset—and having tools for editing tasks specific to VR—would significantly improve their process.

This paper describes initial exploration, development, and evaluation of an in-headset VR video editing interface. We first interviewed a number of VR video professionals to understand the current practice and difficulties of editing VR video on the desktop. Based on these pilot interviews, we describe the major tasks in VR video editing and the design requirements of an editing tool to carry out these tasks.

Based on our design requirements, we developed a prototype system called *Vremiere* that allows users to edit a video entirely in the VR headset. Our system is built upon a timeline design commonly used in commercial systems, with mouse and keyboard as input. The timeline design provides familiar context for trained video editors, enabling easy transfer of skills. We enhance our design with a collection of interface widgets to support manipulation and navigation of spherical video while inside the VR headset. Specifically, our widgets enable 1) spherical visualization to aid navigation, 2) trimming and alignment of cuts, 3) placement of 2D images (titles) directly on the video, 4) annotating with bookmarks for review, and 5) adaptive vignetting to reduce fatigue.

Finally, we present the results of an expert user evaluation conducted among video editors with significant VR video experience. Experts were asked to use our system in a freeform video editing tasks, and provided us with feedback on both the system and the new workflow.

## RELATED WORK

### Spherical video navigation

Because a spherical video viewer only shows a portion of the scene at once, interfaces are needed to manipulate both the time and viewing direction.

Researchers have developed hand gestures for spherical video navigation, for CAVE-like environments [22] and for VR headsets [20]. These gestures are used for basic controls such as play, pause, fast-forward, rewind, zoom, and pan. We focus on more advanced editing operations. Our intended users are video editing professionals who typically prefer keyboard shortcuts and efficient mouse interactions.

Conventional timeline-based video players have been extended to browse spherical video on the desktop. The *FlyAbout* system [13] provides a map-based interface that lets users browse and explore spherical video similarly to Google StreetView. Neng et al. [19] treat spherical video as “hypermedia” that can be annotated with URLs or text, and support exploration of spherical content with features such as panning, navigation compass, and panoramic thumbnails. Our system also enhances the timeline with various navigation aids, and further extends it into a WYSIWYG in-headset editing interface.

### Authoring & editing VR video

Recently, commercial tools have been updated to support VR video editing. The 2015.3 release of Adobe Premiere included a “virtual headset” view for spherical video. The user can click and drag on this view to change the viewpoint. The *Skybox VR Player* by Mettle is a Premiere plug-in that streams video to the Oculus Rift headset. Several 360° cameras also come with software that helps with light-weight editing and export such as the THETA+ Video app, Nikon KeyMission 360/170 Utility, and the Samsung Gear 360 Action Director. However, the user still cannot edit directly in the headset and so must switch between the headset and the desktop interface in order to edit. *Skybox* also provides a mode to show the entire Premiere desktop interface within the headset, but this does not provide WYSIWYG editing: the user must switch

back to the viewer mode for review. Moreover, placing such a complex interface within a headset may be quite difficult for a user to interact with, especially since current headsets are relatively low resolution. In contrast, we provide direct viewing of the video together with editing, and we provide the minimal set of controls for the most important in-headset editing tasks.

## VR PROFESSIONAL INTERVIEWS

We interviewed four professional VR video editors to understand VR video and how they currently edit. These professionals include a VR editor/filmmaker, a VR technical art lead, a creative director of a VR studio, and a VR video editor. Since VR video editing is still fairly unexplored, this broad set of discussions allowed us to understand current practice, identify limitations, and also find design requirements for our system from multiple perspectives.

Video editors currently follow the traditional video workflow to produce VR video. As described by Freeman et al. [9], this workflow includes three stages: pre-production (planning), production (capturing), and post-production (navigating and sorting footage, editing, compositing). Spherical video capture has recently received significant interest from the camera industry, leading to cheaper and more robust cameras. However, editors face unique challenges in the post-production stage, since the available tools are not designed for spherical video.

VR editors’ difficulties with current workflows can be summarized by two main themes. First, because spherical video is best experienced in the VR headset, editors must repeatedly switch between editing on the desktop and reviewing changes in the headset. Second, VR video editing creates new interface needs (such as spherical rotation) that are not handled well by current professional tools. As a result, editors resort to ad hoc approaches to editing spherical video on the desktop. According to the model of Hutchins et al. [10], the gulfs of execution (editing spherical video using 2D tools) and evaluation (breaking from the desktop to preview in the VR headset) are significant problems with this workflow, and there is a clear benefit to more streamlined and direct editing of VR video.

We assembled the interview feedback into a common set of workflow issues, which we detail now.

### Navigating footage

Editors frequently browse footage to find key moments and to review the available shots. This task is particularly important for VR video since the spherical environment offers alternative ways to tell a story within a single video. Navigating spherical video in the headset, however, is currently difficult because of the absence of video control interfaces such as the timeline. One editor mentioned difficulty understanding the full 360° scene in the headset, since only a single view is visible at a time. She also noted that choppy raw footage can cause nausea [14], forcing her to remove the headset.

### Assembling cuts

Professional editors currently try to rotate the scenes across a jump cut to align interesting points in two clips [3]. This technique is often done to make cuts less jarring to watch

in VR. Otherwise the viewer may get thrown off, or start to wander around and lose the narrative planned by the editor. One editor explained that in order to align effectively, she needs to view the video clips inside the headset and take notes. She will then look for the corresponding scenes on the desktop and use Premiere's offset tool to align the video. This process is even more tedious when there are more clips in the project, or when she wants to change the offset of an earlier clip and has to propagate the edit to all subsequent clips.

### Previewing edits

All editors noted that being able to preview the changes they made in the headset during editing is a very important feature but is not well supported. Because the spherical video is shown in desktop tools as a flattened projection, any adjustment to the video such as trimming, color grading, or placement of text can look very different when viewed in the VR headset.

### Titling

To add titles, one editor uses an external image editing application to pre-distort the title image so that it looks right in the equirectangular projection. To make any changes, he repeats the entire process. He also said that the title usually looks much smaller on the desktop compared to viewing it in the headset. The editor explained that the process of adding titles requires him to repeatedly export his project to view the video with the title in a Google Cardboard headset.

### Review and annotation

One editor explained that she reviews all raw footage after capture, and then reviews the entire final video after editing. During reviews, she watches carefully and takes notes. Since she is wearing the headset during review, and current viewers do not support annotation, she writes blindly on a piece of paper while wearing the headset.

Spherical video shows the content in all directions, so she checks to make sure that the final edit does not contain inappropriate action that might have been missed during editing. This requires watching all footage at least twice, once each in opposite view directions.

### IN-HEADSET EDITING WORKFLOW

We propose a new workflow to address the above limitations by allowing a user to edit and preview directly in the headset. Our workflow integrates editing and previewing in a single experience, creating a direct manipulation interface that bridges the gulfs of execution and evaluation [10]. Furthermore, it enables WYSIWYG editing of spherical video, since the user is viewing the video in its final output form.

One strategy we did not employ is to implement all steps of the editing process in the headset. There are tasks that we believe are easier to perform on the desktop, such as managing files and performing initial assembly of clips. Current VR displays are much lower resolution than desktop monitors and are not well-suited for complex interfaces. Furthermore, long-term usage of VR headsets is not well-studied and may cause problems such as fatigue and nausea. There may be other limitations as well, such as the user's inability to perceive their surroundings in a work environment.

Instead, our in-headset editing workflow is complementary to desktop editing. We envision editors working with a conventional desktop interface but performing certain steps of the process in a headset. The editor may decide which tasks to do in headset versus on the desktop. One advantage of this approach is that we can reduce the complexity of the headset interface. We do not need to show every feature to the editor, only those that are most important to use in the headset.

Based on our professional interviews, we identified the following tasks as most important for in-headset interaction: **time-line editing**, **video browsing**, **rotation alignment**, **titling**, and **bookmarking and reviewing**. Some more advanced tasks such as *spatial audio authoring* or *color correction* are also interesting to explore in future work.

Throughout our workflow, we emphasize WYSIWYG direct manipulation. We use the timeline metaphor that users are already familiar with from desktop software. Our interface is simplified, containing only the controls that are relevant for our in-headset editing tasks. We also provide tools to make the process more comfortable for the editor, including tools to avoid nausea [11] when navigating shaky video, and techniques to allow 360° navigation without requiring constant head rotations.

For input devices, we rely on the mouse and keyboard. Our interface is orthogonal to the choice of input device. Expert professional editors typically prefer fast interaction over exotic input techniques, often memorizing most keyboard shortcuts in order to be able to work more quickly. Sitting on a swivel chair with a wireless keyboard and mouse, an editor can have a relatively comfortable and unencumbered editing experience.

### THE VREMIERE SYSTEM

We developed Vremiere, a prototype in-headset VR video editing system. We enable users to perform the target editing tasks while fully immersed in the VR headset. We focus on implementing a complete in-headset workflow and leave desktop software integration as future work. Vremiere runs on the Oculus Rift CV1 headset with rotational tracking, on a Windows 10 desktop with a GeForce GTX 970 graphics card. Our system uses the 80° horizontal field of view of the headset. Currently Vremiere only supports monoscopic video.

Figure 2 presents the main timeline interface for our system, which is overlaid on a spherical video view. This setup allows the user to quickly access editing and navigation tools simultaneously during the playback of the video or reviewing and browsing it.

Tools are accessible via a palette of buttons to the left of the timeline. Video browsing is enhanced by the Little Planet visualization (Figure 2D.3), and discomfort reduced by Adaptive Vignettes (Figure 2D.6). Jump cut alignment is visualized and adjusted using the Rotation Alignment tool (Figure 2D.4). Titles and 2D images can be positioned directly on the video (Figure 2C "Graphics" track), and bookmarks can be placed on the video as well (Figure 2D.5).

In the rest of this section, we describe these features in detail.

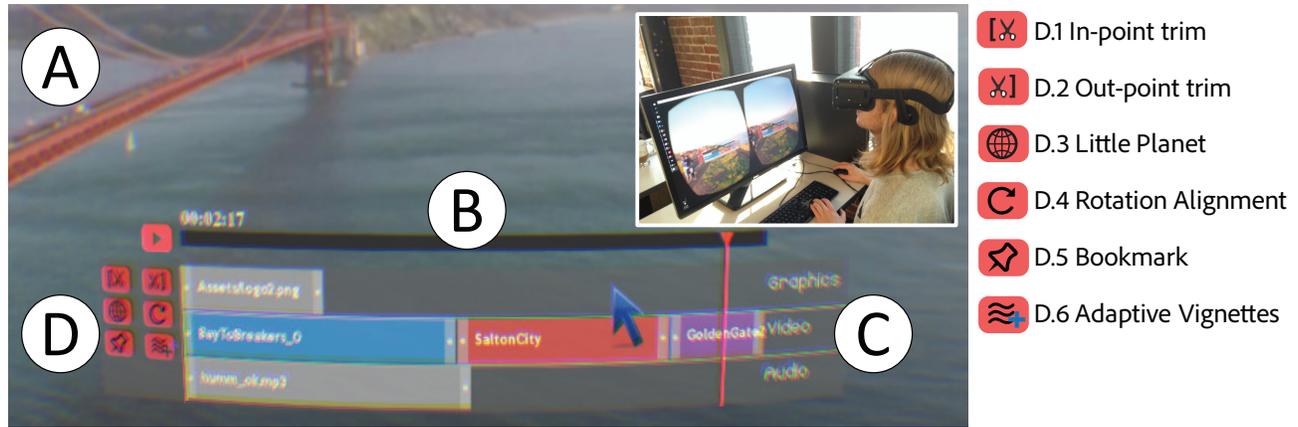


Figure 2: The Vremiere interface contains a video view in the background (A) and floating UI components such as the timeline (B), editing tracks (C) and editing widgets (D). By using familiar mouse and keyboard interaction, an editor can edit and view the spherical video directly inside the VR headset.

### Timeline editing

The editing interface contains the minimal set of conventional interface elements needed to support our target in-headset editing navigation and editing tasks (Figure 2B). It comprises a play/pause button and a timeline, including three main tracks for manipulating assets: graphics, video, and audio (Figure 2C). The interface is view-stabilized, as is the mouse cursor, so everything is attached to the headset view and is always readily accessible. Interface elements are rendered at infinity (monoscopic viewing) and composited on top of the spherical video.

The user can perform basic arrangement and trimming edits on the tracks. To arrange, each block can be dragged along its track. To trim, the user can either drag the left and right handles of each block, or set the in and out trim points using the buttons (Figure 2 D.1 and D.2). Hovering the cursor on top of a block will display its original duration. The interface can be re-positioned by dragging with the right mouse button.

### Video browsing – visualization

Video browsing is typically defined as interactive exploration to search for specific content, or to find inspiration from video content [24]. Our timeline interface supports standard browsing interaction. However, it does not always work well in a VR headset, as we learned from the initial expert interviews.

During VR video viewing, the user sees only a portion of the scene. This can make it difficult to get a sense for all the events in a video, or to find a specific event in some direction. Vremiere allows users to rotate the scene by clicking and dragging on the video to adjust "true north" and make it more comfortable to view other parts of the scene, but still the whole scene cannot be viewed simultaneously.

In order to make browsing and search more effective, Vremiere includes a Little Planet visualization (Figure 3). This visualization is both a minimap and a compass of the scene. In Figure 3, the user quickly maps her current view in the video with the view in the visualization; she can also spot the jumping lady behind her that she might have not noticed before.

The circular, clock-like shape of the visualization enables her to quickly orient herself to that event, or simply drag on the visualization to navigate there by rotating the scene. We also support zooming using the scroll wheel to reveal more details.

The Little Planet is a stereographic projection of a sphere to a plane that has been well-studied in map projection research [16] or used in VR navigation [7]. While it becomes distorted along pitch, it was found to be useful in spatial search [18]. This projection preserves angular properties [16], which has two important implications. First, objects moving around the scene can be followed continuously. Second, Little Planet represents well the spherical characteristic of the video, allowing for better egocentric orientation. It has been reported that such a representation was favored by users when searching for targets along the yaw dimension of a spherical image [18]. Equirectangular projection has also been used to show overview in a desktop-based video player [19]. However, equirectangular projection is often not recommended when observing objects moving around the scene, or when judgement of egocentric directions is important [2].

Following the design of *overview+detail* applications [6], our system displays Little Planet as a secondary window that shows the overview of the scene. This window is toggled by a toolbar button (Figure 2 D.3) or by pressing the Up key on the keyboard. This setup allows the editor to focus on the main content of the scene and trigger the visualization only when necessary. To help users quickly orient and navigate the space (when switching views), the visualization is rotated along yaw so that the up direction always maps to the user's current view. A yellow fan marker is added to help users visualize the field of view (FOV), and a red line marks the true north direction of the video.

### Video browsing – comfort

VR can induce symptoms similar to motion sickness [8, 14], particularly for video with shaky or rotating scene motion. This disturbing motion is accentuated in video editing when the user scrubs along the timeline, speeding through the video.

While shaky video can be stabilized using video stabilization techniques [11, 15], editors will often need to work with raw footage in order to save time and avoid distortions introduced by stabilization, and even smooth video may still be jumpy when scrubbing.



Figure 3: Left: the Little Planet is shown in a small window above the timeline to aid in-headset video navigation. Right: close-up view of the visualization; it can be used both as a minimap and a compass of the 360° image. As noted by the yellow arrow, the user can quickly spot an interesting event outside her field of view (indicated by the yellow fan at the center) that would be otherwise difficult to find.

To make video browsing more comfortable, we provide an adaptive vignetting option, inspired by the method of Fernandes and Feiner [8], who restricted the user’s FOV dynamically, based on gamepad inputs. We use the same vignette setup of [8]<sup>1</sup>, but our system dynamically contracts the vignettes based on the perceived motion of the video from the user’s current viewpoint (Figure 4). When the user is watching in the headset, either normally or through scrubbing the timeline, the vignettes contract faster in shaky scenes and slower in steady scenes. The vignettes’ diameter is measured in the diagonal field of view of the headset. Their default diameter is 120°, which corresponds to no vignetting. They stop contracting at 60° to avoid blocking too much of the view, and expand back to the default size when the user pauses the video or stops scrubbing.

To compute the perceived motion, we use optical flow as a proxy to estimate how shaky the video scene is. This approach is similar to the system by Pongnumkul et al. [21], which also used the number of detected image features to approximate the visual quality of video scenes. We precompute optical flow for all videos using the Lucas-Kanade method [17]. During operation, Vremiere computes the motion magnitude of the user’s current viewpoint in the video and uses it to determine the contraction rate of the vignettes. We followed the approach of Cheng et al. [5] to define the motion magnitude for the user’s current viewpoint as:

$$M_f = \sum_{i=1}^N \|V_i\| \quad (1)$$

where  $N$  is the number of tracked points in the user’s current view and  $V_i$  is the motion vector from the current frame  $f$  to the next frame  $f + 1$  of point  $i$ .

<sup>1</sup>please refer to Fernandes and Feiner [8] for more details on the design of vignettes

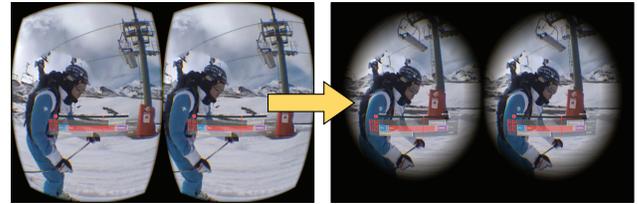


Figure 4: Our system dynamically contracts a pair of vignettes in front of the user’s eyes based on the perceived motion in the video, to help reduce discomfort during video playback or scrubbing. Note, on the left, the default diameter of the vignette is 120°, which corresponds to no vignetting. © Ábaco Digital Zaragoza

When the video plays, the contraction rate is set proportionally to a value between 0 and  $-30^\circ/s$  based on  $M_f$ . These values were defined empirically to make the vignettes contract fast enough in choppy scenes but still subtle in normal scenes. The rate is set to 0 when  $M_f \approx \min(M_f)$  and to  $-30$  when  $M_f \approx \max(M_f)$ , respectively.  $\min(M_f)$  and  $\max(M_f)$  are precomputed over all  $M_f$  of which viewpoints are centered around all tracked points in all the video, before launching the application for the video clips the user is editing. When the user scrubs the video, our system speeds up the contraction rate by multiplying it with the number of frames changed during scrubbing. Therefore, scrubbing the video faster will contract the vignettes faster.

### Rotation alignment

In our initial expert interviews, two editors stressed the importance of being able to rotate the video shots between cuts in order to align the interesting moments between them [3]. We use the term “rotation” loosely to mean offsetting the video pixels uniformly around the yaw axis. We also define a good cut in VR video as a horizontal (yaw) alignment of points of interest before and after a cut. While it is possible to align a spherical video about roll and pitch as well, we focus on yaw because interesting content in spherical video often happens along this dimension. Also, roll and pitch tend to be fixed for spherical video to keep the horizon level.

Vremiere includes a dedicated tool called *Rotation Alignment* (Figure 5B). It shows an equirectangular projection of the clips before and after the cut and supports useful interaction to fine tune the alignment.

The visualization shows two vertically-stacked panoramas: the last frame before the cut, and the first frame after the cut. The equirectangular format was chosen instead of the Little Planet projection because it can better show the horizontal alignment, while still providing a good overview of the scenes. The user’s horizontal FOV is visualized in each panorama as yellow vertical bars. These bars rotate with the user’s viewpoint, so she can quickly see how the views before and after the cut are aligned.

We support several interactions to allow the editor to fine tune the alignment. First, the user can trim the video clips using the timeline and see the frames around the cut update

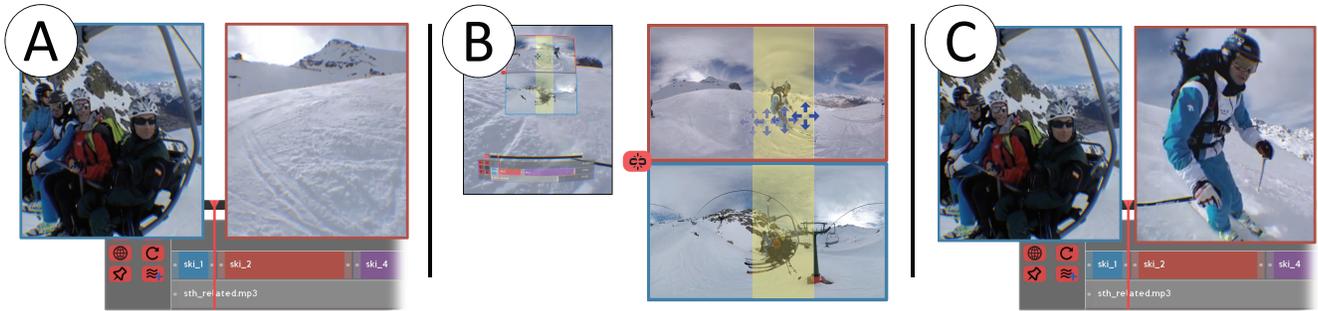


Figure 5: A: an example of a jarring cut in VR video. The cut transitions from the ski lift to an empty view of the mountain in the *ski\_2* video. B: Our Rotation Alignment tool is shown above the timeline. In the close-up view, the user can visualize how the cut transitions from one clip to another given a specific viewpoint in the video (visualized by yellow bars); she can also directly drag on the clips to align events before and after a cut, thereby helping the viewer to follow key elements in the video. Here, the Ripple Rotation mode is shown currently as an *Unlink* toggle. C: after rotation, the skier in clip *ski\_2* is aligned to the previous shot, resulting in a much better cut. © Ábaco Digital Zaragoza

in real-time on the panoramas. Second, hovering the mouse cursor over a panorama will update the main video view to that frame, allowing the user to quickly check the current alignment without having to adjust the timeline. Third, the user can rotate a video by clicking and dragging directly on its corresponding panorama. For example, in Figure 5C, the user has aligned the ski lift before the cut to the skier after the clip, rather than transitioning from the ski lift to an empty view.

In standard video editing tools, rotations are defined with respect to an absolute orientation independently per-video clip. This can be cumbersome in large video projects, as editors generally care about relative orientation between shots. Adjusting the rotation of an early clip can break later cuts. We provide a mode called Ripple Rotation that propagates rotations to subsequent clips. Vremiere displays a *Link/Unlink* toggle in the rotation alignment tool to indicate the relationship between two clips (Figure 5B). Enabling this toggle will set the rotation of the video after the cut to be relative to that of the video before the cut. In Figure 5B, if the toggle is off, the user can change the rotation of each video individually; otherwise, rotating the blue (*ski\_1*) video will also rotate the red video (*ski\_2*) by the same amount.

We experimented with visualizing the rotation alignment by superimposing the frames before and after the cut and allowing the user to directly drag on the video to change its rotation. We discarded this design because our pilot testing revealed two limitations. First, blending two frames makes it harder to see the precise alignment. Second, dragging on the video induced discomfort in some pilot testers. Previous research in simulator sickness advises against rotating the scene in front of the user [14].

### Titling

As discussed in our expert interviews, editors often need to place 2D images of text or logos in VR video. Overlaying 2D images on a video is a basic form of video compositing and is an important operation in making VR videos. On the desktop, editors can use plug-ins to convert 2D images into

equiangular projection before applying it to the video, but the distorted form of the image makes editing and placing it unintuitive. In particular, it is very difficult to tell how big objects will appear in the headset (Figure 6 left).

Vremiere allows the editor to load and view a 2D image in the headset and provides WYSIWYG controls to place it anywhere in the VR video. The image is projected onto the view sphere at infinity (monoscopic viewing), and appears undistorted (straight lines are preserved) when placed at any position in the scene and viewed from any angle. The user can click and drag on the image, which is selected by casting a ray through the 2D mouse position to the view sphere and intersecting it with the image. Because the cursor is view-stabilized, the image position also becomes view-stabilized while dragging. To click on the image when it is not in the current view, the user can locate it quickly using the Little Planet tool discussed earlier. For example, in Figure 6, the editor can easily position the logo image on the parachute (Figure 6 right), which is difficult on the desktop due to distortion (Figure 6 left).

### Bookmarking and reviewing

Video editors use timeline markers for a variety of purposes, including synchronizing with music, reviewing footage, and giving feedback. Wearing a headset makes it difficult to take notes or speak directly with a colleague, so timeline markers are even more important.

Standard marker tools are inadequate for VR video. Because the video content shows in 360°, a marker needs to be placed on the video both at a specific time and at a specific location in the scene. The spatial location of markers presents a challenge when the editor needs to review them in the headset, because in order to find a marked scene element, the video must be adjusted to the bookmark's timestamp, and the view must be adjusted to point to the bookmark's location.

Vremiere allows users to add markers to a video and quickly review them in the headset. When the user activates the Bookmark tool (Figure 7, Figure 2 D.5), the cursor changes to a pin



Figure 6: Left: On desktop, adding 2D text or images to a spherical video is often unintuitive. Right: in the headset, our system provides a natural view of the image (*Sample Logo*) and enables users to directly place it anywhere in the scene. © P J Orravan

icon. The user can click directly on the video to place a marker. Our system casts a ray from the camera center following the cursor direction and places the marker at the intersection between the ray and the video sphere. The markers are shown as pins on both the video and the timeline. The user can browse to each marker by pressing the left or right keys on the keyboard. When a marker is selected, Vremiere navigates to the marker's time and centers the view on the marker. The shift to another marker updates the view instantly, allowing users to browse notes quickly while creating minimal camera motion that can cause disorientation [14]. Alternately, we could show visual indicators on the headset view to help editors search for markers. While indicators are less disrupting, it might be slow when the editor wants to browse through notes quickly.

## EXPERT REVIEW

We conducted an expert review to evaluate the effectiveness of our system in supporting VR video editing. Specifically, we wanted to see 1) if our in-headset editing workflow adds significant value to VR video editing and 2) how our editing interface and widgets aid users in editing spherical video in the headset.

A direct comparison of our system with a professional editing suite would be difficult. Since there are currently no standard tools to edit VR video, editors often have different ad hoc workflows or use third-party plug-ins. Thus, there is no clear baseline for comparison. Alternately, we could use a desktop version of our interface as a baseline. While this comparison would allow us to study the benefit of in-headset editing, it would not measure the utility of our widgets, which were designed specifically for in-headset video interaction.

Therefore, we focus on a qualitative evaluation via expert review on a freeform video editing task to gain insight from experienced users about both our new workflow and our system. We collected feedback on workflow, utility, and usability of Vremiere, as well as subjective preference between Vremiere and the expert's own workflow. We also measured pre- and post- Simulator Sickness Questionnaire (SSQ) scores [12] to observe the comfort level of participants.

We recruited six professional video editors (1 female), aged 19 to 41 (Table 1). The participants have significant experience with VR video production, having either published a VR video



Figure 7: The user can directly click on the video to add markers. These markers are visualized on the timeline and can be reviewed quickly using keyboard shortcuts. © Jacob Phillips

or been a part of a production team that produced a VR video. One editor participated in our initial interviews but was not involved in the design of Vremiere. Each participant was compensated with a \$25 gift card for their time (approximately one hour).

## Procedure

### Introduction and training

Upon arriving at the study site, participants were asked about demographics and then completed a pre-exposure SSQ. They were then introduced to the system. An experimenter demonstrated each feature in the system and explained how it can be used to edit video in the headset. Participants were asked to put on the headset and tried all the features on a test video.

### Main task: freeform video editing

Participants were given four spherical video clips (total time  $\approx$  3.5 minutes), an mp3 soundtrack, and a logo image, and were asked to produce a VR video for the Oculus Rift headset. The clips show scenes of various cities in the world that were suitable for a travelogue video. Before the task, participants were told to take as much time as needed and were encouraged to continue the task until satisfied.

### Questionnaire and interview

After the task, participants completed a post-exposure SSQ questionnaire, and another questionnaire that consists of four 7-point Likert scale questions about various aspects of our workflow and system. Finally, participants were interviewed with open-ended questions about the workflow and specific parts of the system.

## Results

Reported results follow a 7-point Likert scale format, with 1 labelled "strongly disagree" and 7 labelled "strongly agree."

### In-headset editing workflow

All participants completed the task in the VR headset and reported they were satisfied with their result. Participants

Participant	VR video experience	Target platform	Toolset
P1	1 year	Desktop, Mobile VR	Adobe After Effects
P2	0.5 years	Samsung Gear	Autopano, Adobe Premiere
P3	1.5 years	Oculus Rift DK2, HTC Vive	Autopano, Final Cut Pro, Ricoh Theta software
P4	2.5 years	HTC Vive, Google Cardboard	Adobe Premiere, Ricoh Theta software
P5	1 year	Google Cardboard	Adobe Premiere
P6	1 year	Google Cardboard	Autopano, Adobe Premiere

Table 1: Participants' demographics

Edit operations	D			N		A	
	1	2	3	4	5	6	7
Preview the video directly in the headset							6
Timeline navigation of the video					1	1	4
Trimming					1	3	2
Rotation alignment					1	1	4
Titling							6
Bookmark and review						1	5

Table 2: Summary of responses to the question “I feel this editing operation is intuitive and easy to learn” (D: disagree, N: neutral, A: agree).

responded overwhelmingly that our in-headset editing system would be very useful for editing VR video. When asked to rate their agreement to the statement: “My workflow through the entire interface was coherent and fluid”, one rated 7/7, three rated 6/7, and two rated 5/7. Comments from the interviews point to several benefits of in-headset editing: *able to edit and see the changes in real-time* (P3, P5, P6), *able to adjust fine details* (P4), *intuitive* (P1, P2, P3), and *save time* (P6).

#### Comfort level

We were also interested in observing participant's comfort level during the study. Participants spent on average 16.12 minutes to finish the task (STD = 5.26). The SSQ score after the study (M = 3.5, STD = 4.03) rises slightly compared to the score before the study (M = 1.5, STD = 1.37); the difference was not statistically significant (paired-samples t-test,  $t(5) = 1.309$ ,  $p = 0.24$ ). This indicates that participants were mostly comfortable throughout the study. While our system provides the *adaptive vignettes* option, only one participant used it. However, this participant spent the most time in the headset (23.07 minutes) and reported his eyes were more relaxed after the study. While this is promising, we attribute this result mostly to expert's familiarity with viewing video in VR.

#### Interacting with spherical video in the headset

All participants rated their agreement with the statement “I feel this editing operation is intuitive and easy to learn” (Table 2). This indicates that participants could easily edit spherical video in the headset using our interface and widgets.

We also asked participants to rate the usefulness of each widget (Table 3). Overall, *rotation alignment*, *titling*, and *adaptive vignettes* received high ratings by participants. Meanwhile,

Vremiere Widgets	D			N		A	
	1	2	3	4	5	6	7
Little planet		2		1			3
Rotation alignment					1	1	4
Titling						2	4
Bookmark and review	1		2		1	1	1
Adaptive vignettes		1			1	3	1

Table 3: Summary of responses to the question “I feel this widget is useful for my editing task in VR” (D: disagree, N: neutral, A: agree).

*Little Planet* and *bookmark and review* received different opinions. Interview comments show that the usage of them depends on participants' personal preference and editing method. For *Little Planet*, participants who favored this tool used it very frequently throughout the task, based on our observation. They mainly used it to skim the footage before editing, or to find interesting moments to rotate and align the cut; one expert even used it to review the final edit. However, participants that did not find much use of *Little Planet* reported that they preferred to look at the video directly. For *bookmark and review*, we also found the utility of this tool depends on the expert's own experience with markers. Participants who favored this tool frequently used markers in their own workflow, and ended up using this tool more in our study.

#### Subjective preference toward expert's own workflow

When asked to compare each editing operation supported in our system with the expert's own desktop workflow, high ratings were found for certain tasks (Table 4). Specifically, *pre-viewing*, *rotation alignment*, and *titling* were strongly favoured for in-headset editing. One participant described his titling process in Adobe Premiere including extraneous steps such as highlighting the video with a box that simulates the headset view and carefully checking the title to make sure it fits within that box. Any adjustment to the title was very tedious and unintuitive compared to using our system. *Timeline navigation* and *trimming* were also favored, but participants wanted more advanced timeline manipulation and trimming options as in a professional tool. Finally, participants were mostly neutral about the *bookmark and review* operation. Although all participants agreed it is a useful tool for reviewing the video, only one participant currently needs this tool to review footage and communicate with his team.

Edit operations	D			N	A		
	1	2	3	4	5	6	7
Preview the video directly in the headset				1		1	4
Timeline navigation of the video		1		1	1	2	1
Trimming		1	1		3	1	
Rotation alignment					1	2	3
Titling					2	2	2
Bookmark and review	1			4			1

Table 4: Summary of responses to the question “I prefer this editing operation in this VR system as compared to my current workflow”. (D: disagree, N: neutral, A: agree)

## DISCUSSION

The results from the study demonstrate a clear preference from experts to be able to edit VR video in the headset. Despite the paradigm shift from the desktop to the headset environment and the lack of more advanced features often found in professional editing software, our workflow enabled experts to finish the task entirely in the VR headset. The most prominent benefit of in-headset editing is the ability to see changes and make adjustments directly in real-time. This is particularly important because video editing is a tedious task and editors require a tool that provides a fast feedback loop, so they can focus more on the creative task. As VR video is a recent medium and editors are still exploring new ways to create and tell stories through editing VR video, this finding motivates more explorations of in-headset techniques to support this new creative process.

Our system also provides essential tools needed for editing and interacting with spherical video in the headset. At its core, our system is built upon familiar desktop editing interfaces and enhanced with new widgets to support in-headset interaction. It is clear from the study that this unique design allows expert editors to edit easily in the headset. More importantly, experts were able to use our new widgets to perform tedious tasks that are unique to VR video editing such as aligning cuts, placing 2D images in the video, or adding markers. While some of our widgets received different opinions from experts such as the *Little Planet* or the *bookmark and review* tools, they call for more in-depth exploration in techniques to support video navigation and collaboration in the headset.

### Requested Features and Limitations

We collected feedback from participants on additional features that they want in the headset. All participants wanted more features in a professional suit such as advanced timeline, color grading, and keyframing, so they could get more editing done in the headset. A professional editing suite is a substantial piece of software implemented over years by teams of engineers. Our decision not to implement all features allows us to keep our in-headset interface manageable, but makes comparing our system with a professional suit difficult. Future work will need to explore more in-headset features and ways to evaluate in-headset editing with desktop solutions.

Some participants asked about the possibility of using more expressive controllers in Vremiere rather than mouse and keyboard. While our current setup is straightforward for expert editors, it still requires users to rely on their proprioception because they cannot see the input devices. In our study, none of the participants reported any problems, possibly because they had their hands on the keyboard and/or the mouse most of the time. Nevertheless, we believe using wireless input devices or an untethered headset could make the experience more comfortable. One interesting opportunity is to explore how these devices can support video editing in the headset. The traditional form-filling interaction style that involves heavy usage of mouse and keyboard is often used in creative tools for illustration, computer-aided design, or video editing. In video editing, some examples are tasks that require parameter adjustment or text-entry, which may not be desirable to do in the headset at the moment. While form-filling interaction is fast and precise on the desktop, it has been shown to be tedious and error-prone on other devices such as touch screen, and has been replaced in recent exploration with a more direct, object-oriented paradigm [25].

When asked what parts of VR video editing should be done on the desktop, participants wanted to do tedious organization tasks such as rough assembly or audio alignment before switching to VR entirely to fine tune the editing. Vremiere currently supports only basic desktop integration such as loading assets from the desktop and exporting the edited video. Meanwhile, a fully designed hybrid workflow that allows editor to switch between desktop and the VR headset merits more thorough investigation, similarly to research on hybrid 2D/3D design workflow [1].

Our system currently supports only monoscopic video. While Vremiere can also display stereoscopic content, some interface elements will need to be adjusted carefully to avoid creating contradicting depth cues with the objects in video. Existing guidelines from stereoscopic 3D game interfaces could be used to guide future stereoscopic support of our system [23].

## CONCLUSION

We have presented Vremiere, an application that enables a user to directly edit spherical video while fully immersed in a VR headset. Based on interviews with professional VR video editors, we explored and designed an in-headset editing workflow for spherical video editing tasks that are difficult to do on the desktop. We then developed our system around this new workflow. Our system is based upon familiar desktop editing interfaces, but is enhanced with custom widgets to enable intuitive editing of spherical video inside the headset. This unique design enabled editors to simultaneously view the spherical video in its natural form, and to access tools to edit the video, resulting in a seamless editing workflow inside the VR headset. Our expert review study showed that our workflow and our system allow experts to edit VR video in the headset. Experts also found our interface and widgets useful, providing intuitive controls for their editing needs. These findings showed the potential of this new form of editing to support the growing creative needs around VR video.

## ACKNOWLEDGMENTS

We thank the professionals who participated in our interviews and evaluation for their time and feedback. Figure 1, 4 and 5 use images from YouTube users TOYO TIRES JAPAN and Ábaco Digital Zaragoza under a Creative Commons license. Figure 6 and 7 use images with permission from Youtube users P J Orravan and Jacob Phillips. This work was supported in part by NSF IIS-1321119.

## REFERENCES

1. Natalia Bogdan, Tovi Grossman, and George Fitzmaurice. 2014. HybridSpace: Integrating 3D freehand input and stereo viewing into traditional desktop applications. In *2014 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 51–58. DOI: <http://dx.doi.org/10.1109/3DUI.2014.6798842>
2. Wutthigrai Boonsuk, Stephen Gilbert, and Jonathan Kelly. 2012. The impact of three interfaces for 360-degree video on spatial cognition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '12*. ACM Press, New York, New York, USA, 2579. DOI: <http://dx.doi.org/10.1145/2207676.2208647>
3. Jessica Brillhart. 2016a. In the Blink of a Mind—Attention. (2016). <https://medium.com/the-language-of-vr/in-the-blink-of-a-mind-attention-1fdff60fa045>
4. Jessica Brillhart. 2016b. In the Blink of a Mind—Prologue. (2016). <https://medium.com/the-language-of-vr/in-the-blink-of-a-mind-prologue-7864c0474a29>
5. Kai-Yin Cheng, Sheng-Jie Luo, Bing-Yu Chen, and Hao-Hua Chu. 2009. SmartPlayer: User-Centric Video Fast-Forwarding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '09*. ACM Press, New York, New York, USA, 789. DOI: <http://dx.doi.org/10.1145/1518701.1518823>
6. Andy Cockburn, Amy Karlson, and Benjamin B Bederson. 2008. A review of overview+detail, zooming, and focus+context interfaces. *Comput. Surveys* 41, 1 (dec 2008), 1–31. DOI: <http://dx.doi.org/10.1145/1456650.1456652>
7. Kevin Fan, Liwei Chan, Daiya Kato, Kouta Minamizawa, and Masahiko Inami. 2016. VR Planet: Interface for Meta-View and Feet Interaction of VR Contents. In *ACM SIGGRAPH 2016 VR Village on - SIGGRAPH '16*. ACM Press, New York, New York, USA, 1–2. DOI: <http://dx.doi.org/10.1145/2929490.2931001>
8. Ajoy S Fernandes and Steven K Feiner. 2016. Combating VR sickness through subtle dynamic field-of-view modification. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 201–210. DOI: <http://dx.doi.org/10.1109/3DUI.2016.7460053>
9. Dustin E.R. Freeman, Stephanie Santosa, Fanny Chevalier, Ravin Balakrishnan, and Karan Singh. 2014. LACES: Live Authoring through Compositing and Editing of Streaming Video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '14*. ACM Press, New York, New York, USA, 1207–1216. DOI: <http://dx.doi.org/10.1145/2556288.2557304>
10. Edwin L. Hutchins, James D. Hollan, and Donald A. Norman. 1985. Direct Manipulation Interfaces. *Human-Computer Interaction* 1, 4 (dec 1985), 311–338. DOI: [http://dx.doi.org/10.1207/s15327051hci0104\\_2](http://dx.doi.org/10.1207/s15327051hci0104_2)
11. Shunichi Kasahara and Jun Rekimoto. 2015. JackIn Head: Immersive Visual Telepresence System with Omnidirectional Wearable Camera for Remote Collaboration. In *Proceedings of the 21st ACM Symposium on Virtual Reality Software and Technology - VRST '15*. ACM Press, New York, New York, USA, 217–225. DOI: <http://dx.doi.org/10.1145/2821592.2821608>
12. Robert S Kennedy, Norman E Lane, Kevin S Berbaum, and Michael G Lilienthal. 1993. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology* 3, 3 (1993), 203–220.
13. Don Kimber, Jonathan Foote, and Surapong Lertsithichai. 2001. FlyAbout: Spatially Indexed Panoramic Video. In *Proceedings of the ninth ACM international conference on Multimedia - MULTIMEDIA '01*. ACM Press, New York, New York, USA, 339. DOI: <http://dx.doi.org/10.1145/500141.500192>
14. Eugenia M Kolasinski. 1995. *Simulator Sickness in Virtual Environments*. Technical Report. DTIC Document.
15. Johannes Kopf. 2016. 360 video stabilization: A new algorithm for smoother 360 video viewing. (2016). <https://code.facebook.com/posts/697469023742261/360-video-stabilization-a-new-algorithm-for-smoother-360-video-viewing/>
16. Oh-Hyun Kwon, Chris Muelder, Kyungwon Lee, and Kwan-Liu Ma. 2016. A Study of Layout, Rendering, and Interaction Methods for Immersive Graph Visualization. *IEEE Transactions on Visualization and Computer Graphics* 22, 7 (2016), 1802–1815. DOI: <http://dx.doi.org/10.1109/TVCG.2016.2520921>
17. Bruce D. Lucas and Takeo Kanade. 1981. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2 (IJCAI'81)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 674–679. <http://dl.acm.org/citation.cfm?id=1623264.1623280>
18. Alessandro Mulloni, Hartmut Seichter, Andreas Dünser, Patrick Baudisch, and Dieter Schmalstieg. 2012. 360° panoramic overviews for location-based services. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '12*. ACM Press, New York, New York, USA, 2565. DOI: <http://dx.doi.org/10.1145/2207676.2208645>

19. Luís a. R. Neng and Teresa Chambel. 2010. Get around 360° hypervideo. In *Proceedings of the 14th International Academic MindTrek Conference on Envisioning Future Media Environments - MindTrek '10*. ACM Press, New York, New York, USA, 119. DOI : <http://dx.doi.org/10.1145/1930488.1930512>
20. Benjamin Petry and Jochen Huber. 2015. Towards effective interaction with omnidirectional videos using immersive virtual reality headsets. In *Proceedings of the 6th Augmented Human International Conference on - AH '15*. ACM Press, New York, New York, USA, 217–218. DOI : <http://dx.doi.org/10.1145/2735711.2735785>
21. Suporn Pongnumkul, Jue Wang, and Michael Cohen. 2008. Creating map-based storyboards for browsing tour videos. In *Proceedings of the 21st annual ACM symposium on User interface software and technology - UIST '08*. ACM, New York, NY, USA, 13–22. DOI : <http://dx.doi.org/10.1145/1449715.1449720>
22. Gustavo Alberto Rovelo Ruiz, Davy Vanacken, Kris Luyten, Francisco Abad, and Emilio Camahort. 2014. Multi-viewer gesture-based interaction for omni-directional video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '14*. ACM Press, New York, New York, USA, 4077–4086. DOI : <http://dx.doi.org/10.1145/2556288.2557113>
23. Jonas Schild, Liane Bölicke, Joseph J. LaViola Jr., and Maic Masuch. 2013. Creating and analyzing stereoscopic 3D graphical user interfaces in digital games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*. ACM Press, New York, New York, USA, 169. DOI : <http://dx.doi.org/10.1145/2470654.2470678>
24. Klaus Schoeffmann, Marco A Hudelist, and Jochen Huber. 2015. Video Interaction Tools : A Survey of Recent Work. *Comput. Surveys* 48, 1 (sep 2015), 1–34. DOI : <http://dx.doi.org/10.1145/2808796>
25. Haijun Xia, Bruno Araujo, Tovi Grossman, and Daniel Wigdor. 2016. Object-Oriented Drawing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '16*. ACM Press, New York, New York, USA, 4610–4621. DOI : <http://dx.doi.org/10.1145/2858036.2858075>