TutoriVR: A Video-Based Tutorial System for Design Applications in Virtual Reality

Balasaravanan Thoravi Kumaravel
UC Berkeley, California
tkbala@berkeley.edu

Cuong Nguyen
Adobe Research
San Francisco, California
cunguyen@adobe.com

Stephen DiVerdi
Adobe Research
San Francisco, California
diverdi@adobe.com

Björn Hartmann
UC Berkeley, California
bjoern@eecs.berkeley.edu

ABSTRACT

Virtual Reality painting is a form of 3D-painting done in a Virtual Reality (VR) space. Being a relatively new kind of art form, there is a growing interest within the creative practices community to learn it. Currently, most users learn using community posted 2D-videos on the internet, which are a screencast recording of the painting process by an instructor. While such an approach may suffice for teaching 2D-software tools, these videos by themselves fail in delivering crucial details that required by the user to understand actions in a VR space. We conduct a formative study to identify challenges faced by users in learning to VR-paint using such video-based tutorials. Informed by results of this study, we develop a VR-embedded tutorial system that supplements video tutorials with 3D and contextual aids directly in the user's VR environment. An exploratory evaluation showed users were positive about the system and were able to use the proposed system to recreate painting tasks in VR.

CCS CONCEPTS

• **Human-centered computing → Virtual reality**.

KEYWORDS

Virtual Reality; Software tutorials; Design applications

Figure 1: An overview of our VR-embedded tutorial system. Our video widgets (on the left and below the user) can be rendered directly inside the user's active VR painting application (TiltBrush in this example). The user can access useful tutorial content while painting and can interact with the video widgets to explore and learn at their own pace.

1 INTRODUCTION

Painting, sketching, and sculpting in Virtual Reality (VR), henceforth referred to as "VR design", are emerging forms of artistic expression. The key characteristic of VR design is the ability to move a hand in mid-air to directly manipulate and create 3D shapes. This WYSIWYG paradigm was foreseen by early research [13] and yet has only become widely available recently thanks to advancements in VR optics and spatial tracking technologies. The freedom of expression in VR is perhaps one of the main reasons VR design has received significant interest from creative communities [11].

VR design is a creative process and requires mastery of skills. Through initial conversations with two professional VR artists and observations of online user groups, we found two apparent requirements for designing in VR: VR UI skills

(e.g., how to use interface elements such as 3D menus/tools) and VR artistic skills (e.g., what are the creative techniques available). Currently, learning VR design happens mainly through trial and error. Support for VR design tutorials has been limited. Most VR applications only provide onboarding experiences that focus on basic functions. More importantly, users, while in VR, do not have a convenient way to browse and consume tutorials that were created by other users. This is in stark contrast to the ongoing focus of the HCI community on supporting tutorials for desktop-based design software [15].

The lack of learning aids for VR design motivates our main research question: "How do we design a tutorial system for design tasks in VR?". In this paper, we focus our investigation on the feasibility of using user-generated screencast videos of a VR workflow, as tutorials in VR. Video is prevalent in online VR user communities and on VR applications' learning pages [2]. Its apparent benefit is the low overhead cost of authoring: a user only needs to hit the "record" button once to capture. Other users can then view this capture and learn about the author's design process in VR. This type of learning is self-paced, holistic, and may encourage metacognitive processes [31]. Additionally, screencast video is also application-agnostic—the user does not need to rely on the VR application to provide dedicated APIs for recording. This means that any user can create and share tutorials for any VR applications, techniques, and features.

However, there are inherent limitations to using 2D videos as tutorials in VR. First, traditional video players are not well-integrated into VR design applications. To view video in VR, a user needs to pause the current painting process to switch to another application. Or worse, they may have to take off the Head-Mounted Device (HMD) to view the video on the desktop computer. Second, it is not clear to what extent users can interpret the spatial nature of 3D VR tasks from a 2D video. Compared to standard desktop software, carrying out tasks in VR spaces involves interaction in a much richer 3D input space and with dramatically novel interface concepts and interaction designs (e.g., teleportation, 3D color picking, and so on[1]).

To address these challenges, we develop *VR-embedded widgets*. These widgets can show video content and interfaces directly in the user's VR painting environment, so that a user can simultaneously access the tutorial and the painting. To do this, our widgets inject the VR application's rendering and event system, so that they can be rendered on top of the application in VR and can be programmed to respond to a user's input in VR. Using these widgets as building blocks, we develop a basic timeline-based video player and examine whether users can learn from a screencast tutorial video,

while painting in VR. We find that the presence of a video tutorial system in VR is valuable to users. However, improvements to the interface are needed to increase the utility of screencast video as a VR tutorial. We distill our findings into a set of design implications focusing on 1) UI accessibility 2) Spatio-temporal exploration 3) Depth perception and 4) Instruction awareness.

Based on these findings, we introduce a new tutorial system for VR painting called TutoriVR. Our system takes as input a screencast video captured from both eyes of the instructor and a data log consisting of activity traces of the instructor's VR device such as controller poses, HMD poses and button presses. Both of these are trivially accessible in most VR systems. Using these materials, TutoriVR delivers a more optimal learning experience. First, we enhance the tutorial user's depth perception through rendering the screencast video stereoscopically, and supplementing it with 3D rendering of controllers and active painting strokes. Then, we allow for increased perspective perception by enabling the users to view the stroke creation process from different perspectives. Next, we enable the users to better understand controller interactions through highlighting them for an extended duration allowing for enough time for users to understand any quick actions. Finally, we allow for an increased instructional awareness of users through a widget that follows the user's field of view. We conducted a preliminary evaluation to assess TutoriVR and we found that, users found these improvements useful and were able to complete more critical steps in a tutorial than when using the baseline system.

In summary, the main contributions of this paper are:
- Exploring challenges associated with learning VR design through 2D video-based tutorials
- A VR-embedded widget system that supplements and enhances the understanding of video-based tutorials
- Preliminary evaluations of the developed system indicating that users were able to use the proposed tutorial system to replicate 3D painting tasks in VR

2 RELATED WORK
Designing and sketching in VR
Early research in VR explores the potential of immersive displays in 3D design tasks such as modeling [35] and sketching [39]. More recent work focuses on examining the mechanics of sketching in mid-air, focusing on understanding the quality of the stroke [6, 41] or improving it via correction techniques [8] and advanced workflow [5]. Providing effective learning materials is also important to help develop artistic skills. Our work complements this research by characterizing users' needs in learning design applications in VR and introducing a tutorial system for VR painting.

---

[1]http://www.xr.design/

Tutorial system for design applications

Providing effective instructions for digital content creation has been an active research area within HCI. Previous work focuses on tutorials for desktop applications such as design software [10, 17, 19, 32], sketching [15], and 3D modeling [14]. A major theme in these works is to generate high-quality tutorials based on the raw recordings of a user workflow. This approach can reduce the authoring overhead of the tutorial creator and encourage more sharing of community-generated tutorials. Our work is the first to extend this concept to design applications in VR.

An important question asked in these works is, at what level of the application should the system record the workflow. The recording can either be *application-specific* (e.g., Blender edit operations [14], Photoshop commands [10, 17], Paint.NET document edits and UI events [20]) or *application-agnostic* (e.g., mouse and keyboard inputs [7, 27, 29], OS-level activity tracing [26]). Our work focuses on the latter type of recording. We provide a minimal capturing tool that captures screencast video of a VR workflow together with time-stamped VR system traces. These traces capture button-press events as well as 6DOF interaction data from the HMD and the controllers. They are trivially accessible in most VR systems such as Oculus or HTC Vive. We leverage these data to develop widgets that provide supplemental visualizations to increase the utility of a 2D video in VR.

Alternatively, a VR application may provide APIs for users to capture all the geometric information and the application states of a VR design workflow [1]. A 3D reconstruction of this capture can help users freely explore the workflow from different perspectives. Ponto et al. [33] use tracked HMD data to produce stable videos while preserving the recorder's original intent. Greenhalgh et al.'s [18] work discusses different spatio-temporal mechanisms that allow recording and playback of multi-user VR sessions. Both assume access to the entire VR application in order to facilitate playback. These approaches all depend heavily on the host application to provide proprietary recording and playback mechanisms which are not within the scope of our work.

Tutorial system for VR/AR

Tutorial systems designed for VR and AR environments primarily focus on harnessing spatial tracking and feedback to help users learn motor tasks [3, 24, 36, 40, 42]. Motor skill is only a subset of a VR painting workflow. Our system uses workflow video recording which can convey the full scope of a tutorial including tool usage, task demonstration, and content progression [15].

Video is a lightweight format for capturing workflows. Several AR systems use video to deliver physical assembly instructions [12, 16, 30]. In contrast, our focus is on helping

users create virtual content in VR. Commercial VR design applications like Gravity Sketch, Medium, and Quill provide dedicated 2D video tutorials online and also in VR using standard video widgets such as play, pause, and seek. Our tutorial system also includes a video player, but it is more tightly coupled and is responsive to the user's painting workflow. It can display video interfaces and enable interactive tutorial features directly inside the user's painting environment in an application-agnostic manner. Our system does this by injecting both the rendering and the event system of the host VR application.

3 FORMATIVE STUDY

We carried out a formative study to observe and identify challenges faced by users who learn VR painting using video-based tutorials.

A fair amount of prior work [10, 17, 19, 32] deals with studying how users follow video tutorials to learn to use design software in conventional desktop computing. However, none of these works deal with the usage of video-tutorials to learn the operation of design tools specific to VR. More broadly, none of them deal with the usage of a VR environment embedded video player to learn to do actions within a VR space. Carrying out tasks in VR spaces involves interaction in a richer 3D input space as opposed to the 2D input space of conventional desktops. Hence, learning to paint in VR using 2D video-based tutorials pose additional challenges compared to learning conventional desktop software.

Participants

We recruited six participants (5 males and 1 female) through online forums and mailing lists having community of users with a prior experience in VR painting. Each study lasted for a maximum of 90 minutes and the users were compensated with a $50 gift card.

Baseline tutorial video player

Currently, there exists only limited solutions for users to access a video tutorial directly in VR. The most common way is to switch between VR and watching the video on a desktop computer. A user can also open a video application in VR, but that would require pausing the current VR painting application. Both of these approach disrupt the user's flow of learning and painting.

In order to facilitate an integrated experience of watching and interacting with videos within the same VR painting environment, we built a custom video player that can work in an embedded manner within the environment of any VR application. The video player was built using the Unity engine and SteamVR library, which supports its deployment in a variety of standard VR platforms. Our system was developed for the Oculus Rift device. A preview of this embedded
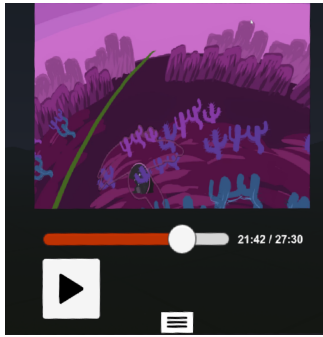
Figure 2: Basic video player used in formative study

video player can be seen in Figure. 2. A user can interact with our system using a standard VR laser pointing technique. We included basic video features such as timeline seeking and play/pause. Additionally, to avoid blocking the user's painting in VR, we enabled users to change the UI position. In pilot testing, we found users frequently needed to check the tutorial by triggering the play/pause button on the UI. This may lead lead to muscle fatigue [9]. Thus, we also added a play/pause shortcut button on the user's controller.

Tasks

The study had three tasks focused on different aspects of learning VR painting. The first task aimed at understanding tool usage in VR. The second task dealt with choosing different tools and creating an object from scratch. The third task required users to follow intricate hand movements and techniques to create a bird-shaped knot for a jewelery art. A short video (2-3 minutes) was used for each task. The videos were screencast video recordings of an expert VR artist's workflow with voice-over. The participants were asked to follow the instructions in the video and create their own version of the painting, while following a think-aloud protocol. Each task had a maximum time limit of 15 minutes.

Measurement

After each task the users had to fill out a questionnaire about the task which asked the user to rate their ability to understand the instructions in the video. They were also asked to rate how easy it was for them to navigate and follow these instructions using the video player. Finally, they were asked to rate whether the video player eased the process of learning compared to how they usually learn VR painting. Then there was a 5-10 minute semi-structured interview about the user's experience of doing the task. Participants' feedback were collected after the study.

Findings of formative study

Embedded video player. Our embedded video player received mostly positive responses in the formative study. Five out

of the six users rated that our embedded video player eased the process of learning compared to how they usually learn to VR paint. One of the users preferred learning through trial and error of the software and did not prefer using any form of tutorial to learn. The users felt that the embedded player offered an integrated experience for them to learn VR painting using video tutorials while they continue to remain in the creative workflow of their painting process. They appreciated that the video player allowed them make instant visual comparisons of their work with the tutorial and make quick modifications. Users liked the ability to position the video player as per their convenience and that the video player is rendered on top of anything else in the VR environment. Although, in theory, rendering the player on top of VR graphics may introduce depth conflict [28], none of our users reported or showed signs of it. A possible explanation is that the mobility of the video player allowed users to move it around to avoid blocking their content, thereby preventing depth conflicts. However, depth conflict issues could occur in larger and more cluttered VR spaces, in which case the adjustment solutions proposed by Nguyen et. al [28] could be applied. Though it was possible for users to move the video player in *z*-direction to change its size, they preferred to have a more explicit control. A major request from users was to have a clearer step structure on the timeline to help with navigation; this is a well-known issue in previous video tutorial research [32].

Controller interactions. One of the recurring problems that the users faced was understanding the interactions of the VR controllers in the video. Similar to the mouse cursor, the VR controller is the primary input device in most VR applications. Whenever a user gets stuck on trying to perform the low-level actions, it was observed that the user actively looks for the VR controller in the video to figure out what action was done with it. This approach however is not always feasible due to various reasons. Controllers may not be in the field of view of the instructor, in which case the controllers are absent in the video. Button presses on the controllers can be too quick to notice, even when it is presented in the video. This would require users to precisely navigate to the frame the button was pressed in order to identify it. Users also found it difficult to follow actions that required a coordinated combination of consecutive button presses and hand gestures. For instance, in the first task, one needs to move a selected object by touching the object, hold down a button on the controller and then move their hands in order to move the selected object. This multi-step action was not understood by many users through watching videos because it was easy to miss the timing of at least one of those actions.

In summary, we identify three primary sources that lead to the occurrence of these interaction problems while following video-tutorials to learn VR design tools:

- Unfamiliar button mapping - Learning to perform actions in VR environment requires users to learn an unfamiliar button-action mapping. Since interactions in VR are not standardized, there is no preconceived and established notion of what all buttons mean, how they work and what purpose they serve. This is in contrast to controllers like mice where clicking/dragging type interactions are standardized.
- Visiblity problem - There is a lack of ability to visually see the controller interactions followed by an instructor to achieve a particular output.
- Richer input space - We have two 6DoF tracked controllers for interacting with a 3D space. This opens up a richer input space with multiple interactions that need to occur in tandem with button presses, which may not be conveyed effectively in the video depending on the viewpoint of the instructor.

Understanding 3D information. Throughout the tasks, users were unable to figure out relative depth between the controllers and the different elements of the VR painting environment. This prevents users from being able to reason and answer questions like "How far away from the object should I draw something?", "Is the controller touching an artifact or not?", "Is the drawn stroke an ellipse or a circle?". In a particular selection task, where the instructor selects an artifact of the painting by a combination of touching the artifact and pressing a button on the controller, none of the users realized from the video that the controller needed to be in contact with the artifact. All users reported that as a main hurdle in understanding and completion of that task.

In addition to this, all users faced problems in understanding the 3D motion of the instructors' hand gestures. When presented with an action involving the creation of a complex stroke, users felt that they had no way of replicating the stroke by seeing the video alone, and hence chose to create their own version that looked visually similar to that seen from the viewpoint of the instructor. However, there was no means the compare their stroke with that drawn in the video and analyze it from different perspectives. In addition to the problem of viewpoint, the creation of such strokes can be very fast as well as be occluded by itself or other elements in the VR painting environment. This prevents an awareness about the complexity of the stroke which leads users to misunderstand the stroke structure and draw them in an incorrect manner. At least one of these problems were observed in all the users when they attempted our third task which involved tying a bird shaped knot for a jewelery art.

The above problems can be traced back to the fact that the videos are inherently 2D, where information relating to the depth and perspective may not be apparent without explicit instructor elicitation or cues like lighting, shadows and occlusion [34]. But in many paintings as well as painting applications some of these cues could be entirely missing, and explicit elicitation by instructors are generally absent due to the activity focused, screencast nature of these recordings.

Missing out on instructions. Sometimes users missed important changes in the video such as a tool selection or mode changes. From user interviews, it was realized that this happened because of at least one of the three reasons.

- It was very common for users in the VR environment to look away from the video player while carrying out their actions, thereby missing important instructions if the instructor did not explicitly voice out of any important step.
- Painting being a creative practice, often gets users immersed with their own painting process that they missed hearing an important step while the video is constantly played in the background.
- Due to the lack of a good tutorial navigation mechanism, the user skipped an important step while navigating the video.

4 DESIGN IMPLICATIONS

Based on these observations from the formative study, we derive a generic set of design implications for screencast video-based tutorial system for Virtual Reality painting

Accessibility of the video player: The tutorial video player should be accessible within the VR painting environment of the user. This would enable users to have a quick visual reference for the tutorial steps, allow for easy verification of their work and would help retain the user's workflow during the painting process. In addition to this, the player must also address the typical problems faced by any video-based tutorial system such as video navigation by explicitly marking important events and enabling quick access to those sections.

Spatio-temporal exploration: The system should offer the ability for users to explore VR-specific spatial actions including controller poses, button interactions, trajectories as well as the stroke creation process from different perspectives. The system should then retain this information for an extended period of time, so that even if actions are ephemeral, users can still observe and interpret them. This would give users both the spatial as well as temporal freedom to explore, analyze and understand the various complex actions in the VR space.

Relative Depth Perception: User should be aware and instantly be able to perceive the relative depth of an instructor's action. Though the perspective exploration supports understanding the absolute shape and trajectory of actions carried out by the instructor with their controllers, it may not quite convey the awareness as well as information regarding the spatial occurrence of those actions especially in terms of depth, relative to the other elements present in the VR environment. For instance, questions like "Is the controller touching a object or not?", "Did the controller move in z-direction? If yes, by how much?", may be hard to answer by just using 2D videos without additional information on relative depth.

User awareness of the tutorial instructions: As with most 3D spatial tasks, user's workspace in VR is much larger than their field of view. This makes it easy for user to be engaged in a task, looking away from the video player. In this case, users can possibly miss the critical steps of a tutorial. The tutorial system must provide awareness aids to the user regarding important events and help user keep track of the tutorial while being minimally disruptive to the user activity.

5 IMPROVED VIDEO TUTORIAL SYSTEM

Using the design implications that were developed in the previous section, Our TutoriVR system seeks to address the problems that were uncovered in our formative studies. TutoriVR achieves this in an application-agnostic as well as platform-agnostic manner through leveraging on data that are available in all the commonly used VR systems.

To generate the necessary metadata for TutoriVR , we provide minimal interfaces that runs in background and records the instructor's VR painting workflow.

- **Screencast video** + **VR system traces**. An active VR application usually mirrors what the user sees to a desktop window. We developed an executable script that one can use to record a screencast video together with a time-stamped log of the VR system traces. This log includes the HMD and controller 3D positions, orientations, and the controller button presses events. This logging mechanism is supported in both of the two standard VR development framework (e.g., Oculus SDK and OpenVR).
- **Author annotations**. The tutorial author can also optionally provide annotations in an XML file to indicate important events or emphasize certain segments in the recording. The important events are of three types - 1.) Critical events that users should pay attention to 2.) Tool/mode changes 3.) Step transitions.

TutoriVR packages these metadata into an embedded tutorial system in the user's VR application environment (Figure 3). It has four main components - Improved video player, Stereo mode, Perspective thumbnail, Awareness widget.
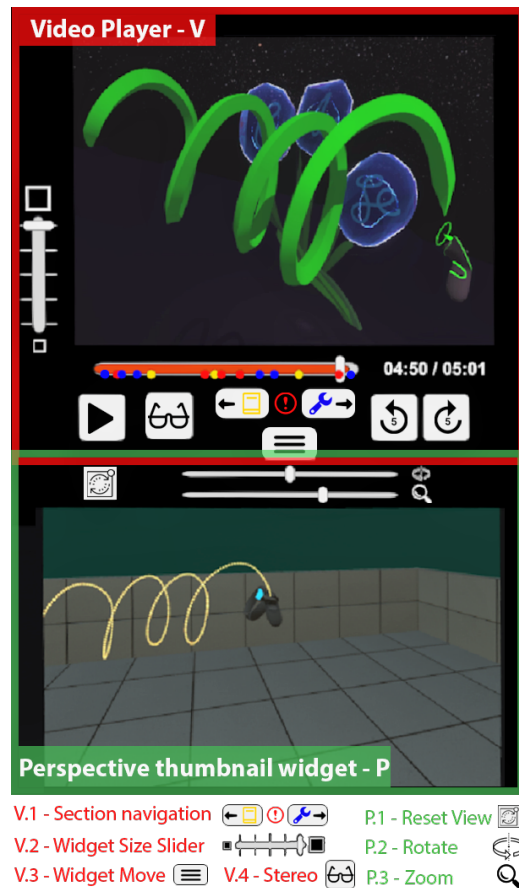

Figure 3: The main panel of the TutoriVR system. Up (V): The improved video interface; Down (P): The Perspective thumbnail view which simulates the actions and interactions of the instructor's controller to help users better understand the video content

Event	Icon	Marker
Tool change	🔧	🔵
Critical event	⚠	🔴
Step change	📄	🟡

Table 1: Tutorial author's annotations, their corresponding icons and timeline markers

Improved video player

Based on user feedback from the formative study, we implemented several improvements to the base video player to make it more accessible in VR. First, the size of the video player needs to be adjustable. A large UI provides more details but may block the user's FOV in VR. A small UI is

less obtrusive but is less visible. We let users resize the video player by adjusting a vertical slider (Figure 3 V.2). Second, important events and temporal structure of the tutorial should be made more explicit. We explicitly visualize author's annotations on the timeline (Figure 3). A detailed explanations of these icons are provided in Table 1. These icons not only provide navigation cues, but also allows for users to be aware of the important events emphasized by the author [32]. Finally, we added various timeline navigation aids. Seeking a video timeline in VR using the touch controller might be slow and tiring because of hand jitters [9, 25]. We added the -5s/+5s seeking buttons and the section navigation buttons (Figure 3 V.1) to aid in video navigation

Stereo mode

As observed in the formative study, a 2D screencast video does not capture the relative depth between the artist's controller and the virtual objects in the scene. To enhance depth perception of the tutorial content in VR, we provide the "Stereo mode" button (Figure 3 V.4). When activated, our system displays a stereo-captured version of the video. Stereoscopy has been found to be the most effective depth cue in VR manipulation tasks [22]. To enable this feature, our recording script captures both the left and right views of the VR system. Then, we use a graphics shader to separate the left and right views from the video image and render them to the viewer's left and right lens, respectively. As a result, the viewer can perceive stereo depth from watching the tutorial video; the effect is similar to viewing a 3DTV in a living room.

However, this approach may incur perceptual problems that are well-known for stereoscopy such as eye straining or window violation [38]. We implemented the standard "black-tape" technique to reduce window violation [4]. We also turn this mode off by default and expect that the user would only use it when necessary.

Perspective thumbnail widget

The Perspective Thumbnail Widget is a major part of our system with the intent of allowing users to better understand stroke structure and controller interactions.

Using the logs generated with our capture-time system, we simulate the movements and interactions of the controller in the video in a separate window below the main video player (Figure 3 P). Here, we render motion trails to the virtual controllers whenever a stroke is drawn and highlights button press events in blue (Figure 4 right). The thumbnail view mirrors the viewpoint and rotation of the tutorial author in the video, but the user can manually change rotation and zoom for further inspection (Figure 3 P.2 and P.3). In this way, this widget offers two advantages in better understanding two key aspects of the painting process:

- **Understanding Stroke structure:** The widget allows users to view and explore the stroke creation process from different perspectives, free of occlusion by other elements. The user can switch to different perspectives using the zoom/rotate sliders present in the widget as well as snap back to the instructor's viewpoint.
- **Controller interactions:** The virtual controller mirrors motions and actions of the instructor's controllers. Whenever a controller button is pressed by the instructor in the tutorial video, the corresponding buttons are highlighted in the virtual controllers. Being able to see what is being pressed and motion of the controller in an separate and isolated space can help users better understand the spatio-temporal relationship of complex controller interactions.

We render the Perspective Thumbnail window in stereo to convey stronger depth perception. To achieve this, we applied the same rendering approach used in Stereo Mode. We render the controller simulation scene using a custom stereo camera rig and output the left/right textures to the corresponding rendering eye target in the VR system. The walls and grid lines in the thumbnail window provides additional pictorial and perspective depth cues [34]. It can help users more quickly interpret the motion of the virtual controllers. Additionally, we increase the field-of-view of the stereo camera rig to increase the visibility of the controllers. Thus, the user can observe the controller actions as if they are looking over the shoulder of a tutorial author.
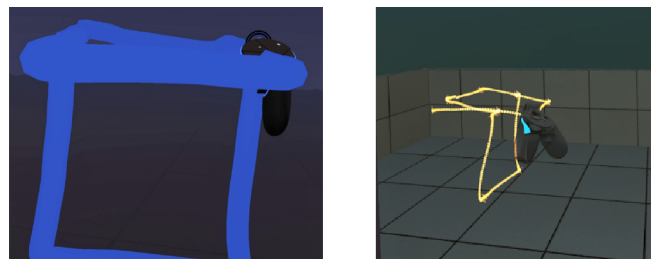

Figure 4: Demonstration of the Perspective Thumbnail Widget. Left: Ambiguous cube-like strokes when viewed on a video; Right: The cube strokes become more interpretable when viewed from a different perspective in the Perspective Thumbnail Widget.

To increase the effectiveness in delivery of these visualizations, we gradually fade the highlights of button presses and the stroke trail, allowing enough time for users to be aware of and understand ephemeral controller actions. This prevents the need to perform a frame-by-frame search of the user to figure out a crucial information required for a task.

Awareness widget

The environment in VR is much larger compared to conventional desktops, it is very easy for users to loose the main video player from their Field of View (FOV) and miss important instructions in a tutorial, especially when they are not explicitly voiced out. For example, when a user is too focused on the painting, they might fail to notice an important action made by the instructor. The user can also choose to place the video player close to the painting, but it may obstruct their workflow. To address this problem, TutoriVR provides a mechanism to non-intrusively provide users with visual information on the progress and key events of the tutorial video (Figure 5). Our Awareness widget is an extension of the main video player. Instead of being positioned in the world coordinate, it is rendered view-fixed—attached to the user's view direction, and becomes visible when they are not viewing the main video player. The widget is transparent for most parts and the user can customize its vertical position so that it does not obstruct their active field of view.
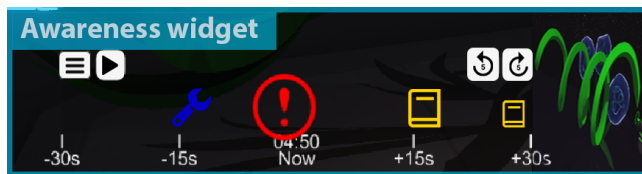

Figure 5: Awareness Widget. This widget is view-fixed in VR. It provides a minimal set of important tutorial content to help users be more aware of the tutorial progress.

We render a minimal set of tutorial content on the Awareness widget including a video thumbnail, current video time and event icons representing events within a +30s/-30s time window of the video play (Figure 5). The icon size increases temporarily as the event approaches the current video time to aid in attracting user attention. User can also quickly navigate to the event by clicking on these icons in this widget.

6 EXPLORATORY USER STUDY

We evaluated our TutoriVR system using an exploratory user study. We wanted to gain initial feedback from users about the improvements we added after the formative study. We also wanted to observe whether users could use these new features in tutorial task in VR.

We compared TutoriVR with the improved baseline video player system from the formative study. Both our baseline and TutoriVR conditions included these improvements (i.e., features V.1, V.2, and V.3 in Fig.3). We chose a within-subjects study design. We designed two tutorial tasks in VR. These tasks are inspired by the *intense task* condition in a previous tutorial research [23]. Specifically, participants were asked to watch a tutorial video in VR and were required to replicate

the final results in the video as quickly and as accurately as possible.

Task 1 (2.5 minutes video) gives users an introduction to basic features of a VR painting application. Task 2 (4.25 minutes video) demonstrates a painting workflow, which involves executing a set of complex strokes and embellishments to create an animated painting. To reduce learning effect, we generated two videos with similar difficulty and length for each task.

Procedure

We recruited 10 participants (2 male, 8 female, age range 19-25) using university and company mailing lists. Participants were compensated with a $50 gift card. 4 participants had regular experience with VR, while all had basic/occasional familiarity. On a scale of 1-5, all but two participants self-rated their art experience as 3 or above and only 2 participants rated their 3D design software experience at 3 or above. The study's total duration was 90 minutes, and required participants to use the TiltBrush application in the Oculus Rift VR system. We selected participants with zero or minimal experience with Tilt Brush. We also selected participants that are right-handed and can perceive stereo vision.

Before the study, participants received a 20-minute training on using the VR system and an introduction of the Tilt-Brush application. The participants then performed Task 1 and Task 2 in order. The order of the systems assigned to a participant were counter-balanced.

Measurement

At the end of each task, the participants completed a custom questionnaire and a NASA-TLX questionnaire[21]. The custom questionnaire asked the participants to rate their perception of the tutorial system and their task performance. The ratings were based on a 7-point Likert scale, with a higher number indicating easier or more useful. Additionally, they completed an open-ended feedback questionnaire after each task, and after the entire study.

In addition to these ratings, we also independently measured the completion rate of each participant in each task. For each task, we created a list of critical steps that a participant must perform. The critical steps in the tasks were selected ahead of time and have at least one of the three attributes: presence of intricate 3D strokes, component of relative 3D depth and complex 6DOF controller interactions. Some examples of critical steps in different tasks are shown in Figure 6. The completion rate is computed by dividing the number of critical steps completed (counted through observation of the participant's performance) by the total number of critical steps.

| (a) Relative 3D depth | (b) Controller interactions |

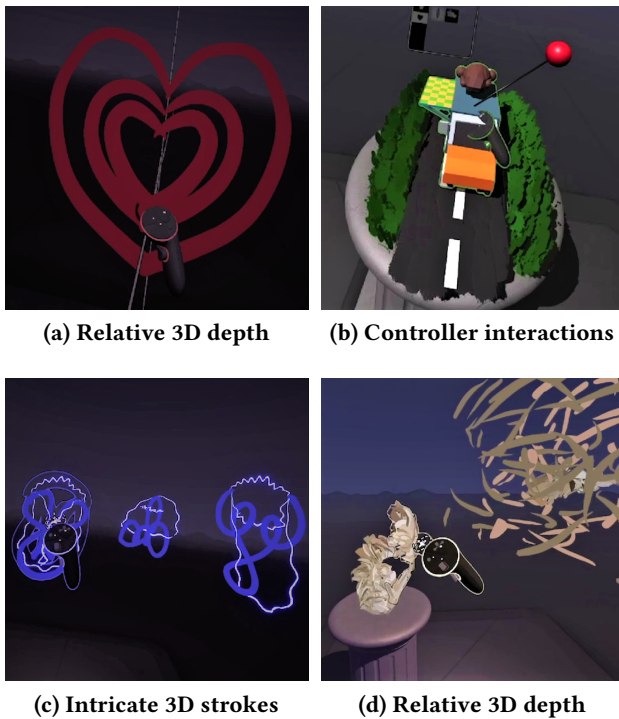| (c) Intricate 3D strokes | (d) Relative 3D depth |

Figure 6: Examples of critical steps in the study tasks: (a) Drawing hearts that are on different planes; (b) An action performed with obstructed controller; (c) Intricate strokes involving 3D loops and coils; (d) assembling a 3D face portrait

7 RESULTS AND DISCUSSION

Figure 7 and 8 give an overview of the ratings collected after the study. Overall, participants achieved more critical steps (M = 7.78 , SD = 3.16) when using our system compared to the baseline system (M = 4.88, SD = 3.31). The difference was found to be statistically significant using a paired-samples t-test (p < 0.02, F = 1.1023 , d = 0.9). Two participants failed the tasks, (i.e., did not complete any critical steps) when using the baseline system. With TutoriVR, on average across the tasks, users were able to complete 49.2% of steps involving intricate 3D strokes, 55.4% involving relative 3D depth, and 63.8% of 6DOF controller interactions, in comparison to the baseline where the corresponding numbers were 28.3%, 31.7%, and 41.3% respectively.

In terms of mental workload, there was a small difference between the mean rating of the NASA-TLX. The rating for our system (M = 52.37, SD = 14.30) was marginally higher than the baseline (M = 51.37 , SD = 18.24). The difference was not statistically significant (p = 0.82 , F = 1.62 , d = 0.06). We observed that there might be two opposing effects at work that led to similar loads. On one hand, the baseline may have reduced cognitive load due to a lack of awareness of the 3D intricacies involved in the VR painting, causing users to create simpler but incorrect paintings. This is in line with the reported study outcome: users achieved more critical steps in the TutoriVR condition relative to baseline. The TutoriVR system helps users be aware of fine details in the painting. This awareness may demand users to put in extra effort during the tasks. On the other hand, cognitive load in the TutoriVR conditions may be lower during controller interaction events. Although users in both conditions can notice on the video the presence of controller events, it is much more difficult to follow these events in baseline. Figuring this out is achieved easily in TutoriVR by using the Perspective Thumbnail widget.

In the questionnaire ratings (Figure 8), participants were positive about both our system and the baseline system. The differences in ratings, however, were not statistically significant. Qualitative evidence for the ability of the system and its features to help users better understand the tutorial was elicited in the open-ended feedback collected from the participants. Out of 10 participants, 8 users felt the Stereo Mode and the Perspective Thumbnail widgets helped them in the tasks. The awareness widget was designed to be of use in free-form exploration tasks with longer task duration, which requires a different kind of study for evaluation. Stereo Mode helped users in getting better task awareness and assessing relative 3D poses of the painting structure. Perspective Thumbnail helped users understand intricate 3D shapes and/or controller interactions. This was also reflected in the written comments by some users:

P2: *"[Stereo mode] was very helpful for understanding the shape and depth of different elements in the video"*

P8: *"[Perspective thumbnail] would have been helpful in this task. Hard to make out the required shape of the loops without the ability to rotate video to see from different perspectives"*
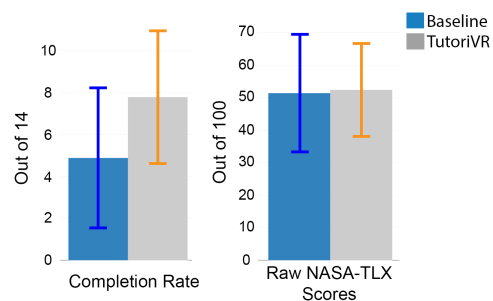

Figure 7: Quantitative Scores and NASA-TLX scores

In summary, our study confirms the value of having an embedded tutorial system in VR. The findings also show
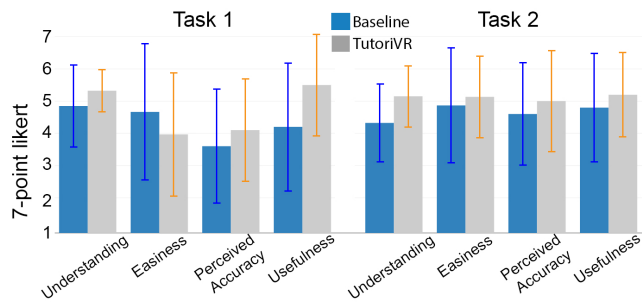
Figure 8: Responses to questionnaire questions: *1) Understanding*: "How easy it to understand the tutorial instructions?", *2) Easiness*: "How easy it was to perform the assigned task?", *3) Perceived Accuracy*: "How accurate do you perceive your result compared to the video?", *4) Usefulness*: "How useful was the assigned tutorial system?"

that our improvements were perceived positively by participants. Most importantly, we found that the improvements helped participants replicate more critical steps in the VR tutorial tasks. Since these tasks were designed to be non-trivial and our participants consist mostly of inexperienced VR painters, these findings provide promising evidences that TutoriVR can aid users in learning to VR paint, while being in the VR environment of the painting application.

8 LIMITATIONS AND FUTURE WORK

Although we found some promising results in our exploratory study, there are several aspects of this new type of tutorial system that we currently cannot answer. First, we do not know the specific utility and usability of individual features. More formal comparisons with dedicated tasks are needed. For example, for Stereo Mode, depth judgment tasks [37] are better at gauging its effectiveness in depth perception. For Awareness Widget, open-ended creative tasks where users tend to focus more on the painting than the tutorial might be more suitable. Second, our study was conducted on a rather small sample size with a specific VR painting application. The findings should be taken with a grain of salt. However, as the first step in evaluating this new type of tutorial, the results are encouraging and we hope will motivate more evaluations with more participants and with other applications in VR (e.g., sculpting, animation, storytelling).

We also identify several interesting areas for future work. First, our current system is designed for self-paced learning where the user decides when and where to access the tutorial. More compelling reactive and intelligent tutorials [15] that provide guidance and feedback can be explored. Second, the VR-embedded widgets can have applications beyond tutorial. Being embedded into a VR application, it opens up new possibilities for collaboration and multi-tasking interactions in VR. One example is to allow a non-VR user to communicate and guide an in-VR user on how to use a VR application via desktop commands.

9 CONCLUSION

In this paper, we have taken the first step at exploring the design of a tutorial system for design tasks in VR. Our system was designed based on VR-embedded widgets, which are application interfaces that are observable and interactable directly inside the user's own VR application. These widgets allow us to create a more integrated video learning experience in VR: a user can learn to VR paint while watching tutorial video without ever leaving the application. We designed a new tutorial system based on this concept, and evaluated it in two small user studies. Our findings show that having a tutorial system in VR is valuable for users who want to learn VR painting from video.

REFERENCES

[1] 2018. Oculus Medium. https://www.oculus.com/medium/learn/
[2] 2018. Quill. https://quill.fb.com/tutorials/
[3] Fraser Anderson, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2013. YouMove: Enhancing Movement Training with an Augmented Reality Mirror. In *Proceedings of the 26th annual ACM symposium on User interface software and technology - UIST '13*. ACM Press, New York, New York, USA, 311–320. https://doi.org/10.1145/2501988.2502045
[4] Jerome Ardouin, Anatole Lecuyer, Maud Marchal, and Eric Marchand. 2011. Design and evaluation of methods to prevent frame cancellation in real-time stereoscopic rendering. In *2011 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 95–98. https://doi.org/10.1109/3DUI.2011.5759223
[5] Rahul Arora, Rubaiat Habib Kazi, Tovi Grossman, George Fitzmaurice, and Karan Singh. 2018. SymbiosisSketch: Combining 2D & 3D Sketching for Designing Detailed 3D Objects in Situ. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 185, 15 pages. https://doi.org/10.1145/3173574.3173759
[6] Rahul Arora, Rubaiat Habib Kazi, Fraser Anderson, Tovi Grossman, Karan Singh, and George Fitzmaurice. 2017. Experimental Evaluation of Sketching on Surfaces in VR. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. ACM Press, New York, New York, USA, 5643–5654. https://doi.org/10.1145/3025453.3025474
[7] Nikola Banovic, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2012. Waken: Reverse Engineering Usage Information and Interface Structure from Software Videos. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 83–92. https://doi.org/10.1145/2380116.2380129

[8] Mayra D Barrera Machuca, Paul Asente, Jingwan Lu, Byungmoon Kim, and Wolfgang Stuerzlinger. 2017. Multiplanes: Assisted freehand VR drawing. In *Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology - UIST '17*. ACM Press, New York, New York, USA, 1–3. https://doi.org/10.1145/3131785.3131794

[9] Marcio C. Cabral, Carlos H. Morimoto, and Marcelo K. Zuffo. 2005. On the Usability of Gesture Interfaces in Virtual Reality Environments. In *Proceedings of the 2005 Latin American Conference on Human-computer Interaction (CLIHC '05)*. ACM, New York, NY, USA, 100–108. https://doi.org/10.1145/1111360.1111370

[10] Pei-Yu Chi, Sally Ahn, Amanda Ren, Mira Dontcheva, Wilmot Li, and Björn Hartmann. 2012. MixT: Automatic Generation of Step-by-Step Mixed Media Tutorials. In *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12*. ACM Press, New York, New York, USA, 93. https://doi.org/10.1145/2380116.2380130

[11] Josh Constine. 2016. Oculus "Quill" Turns VR Painting Into Performance Art. https://techcrunch.com/2016/01/26/oculus-quill/

[12] Dima Damen, Teesid Leelasawassuk, Osian Haines, Andrew Calway, and Walterio Mayol-Cuevas. 2014. You-Do, I-Learn: Discovering Task Relevant Objects and their Modes of Interaction from Multi-User Egocentric Video. In *Proceedings of the British Machine Vision Conference 2014*. British Machine Vision Association, 30.1–30.13. https://doi.org/10.5244/C.28.30

[13] Michael F. Deering. 1995. HoloSketch: a virtual reality sketching/animation tool. *ACM Transactions on Computer-Human Interaction* 2, 3 (1995), 220–238. https://doi.org/10.1145/210079.210087

[14] Jonathan D. Denning, William B. Kerr, and Fabio Pellacini. 2011. MeshFlow: Interactive Visualization of Mesh Construction Sequences. *ACM SIGGRAPH 2011 papers on - SIGGRAPH '11* 30, 4 (2011), 1. https://doi.org/10.1145/1964921.1964961

[15] Jennifer Fernquist, Tovi Grossman, and George Fitzmaurice. 2011. Sketch-sketch Revolution: An Engaging Tutorial System for Guided Sketching and Application Learning. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 373–382. https://doi.org/10.1145/2047196.2047245

[16] Michihiko Goto, Yuko Uematsu, Hideo Saito, Shuji Senda, and Akihiko Iketani. 2010. Task support system by displaying instructional video onto AR workspace. In *2010 IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 83–90. https://doi.org/10.1109/ISMAR.2010.5643554

[17] Floraine Grabler, Maneesh Agrawala, Wilmot Li, Mira Dontcheva, and Takeo Igarashi. 2009. Generating photo manipulation tutorials by demonstration. *ACM Transactions on Graphics* 28, 3 (jul 2009), 1. https://doi.org/10.1145/1531326.1531372

[18] Chris Greenhalgh, Martin Flintham, Jim Purbrick, and Steve Benford. 2002. Applications of temporal links: Recording and replaying virtual environments. In *Virtual Reality, 2002. Proceedings. IEEE*. IEEE, 101–108. https://doi.org/10.1109/VR.2002.996512

[19] Tovi Grossman and George Fitzmaurice. 2010. ToolClips: An Investigation of Contextual Video Assistance for Functionality Understanding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 1515–1524. https://doi.org/10.1145/1753326.1753552

[20] Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2010. Chronicle: capture, exploration, and playback of document workflow histories. In *SIGCHI conference on Human factors in computing systems*. 143–152. http://dl.acm.org/citation.cfm?id=1866054

[21] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.

[22] H.H. Hu, A.a. Gooch, W.B. Thompson, B.E. Smits, J.J. Rieser, and Peter Shirley. 2000. Visual cues for imminent object contact in realistic virtual environments. In *Proceedings Visualization 2000. VIS 2000*. IEEE, 179–185. https://doi.org/10.1109/VISUAL.2000.885692

[23] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2011. Ambient Help. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. ACM Press, New York, New York, USA, 2751. https://doi.org/10.1145/1978942.1979349

[24] Peter Mohr, David Mandl, Markus Tatzgern, Eduardo Veas, Dieter Schmalstieg, and Denis Kalkofen. 2017. Retargeting Video Tutorials Showing Tools With Surface Contact to Augmented Reality. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. ACM Press, New York, New York, USA, 6547–6558. https://doi.org/10.1145/3025453.3025688

[25] Brad A Myers, Rishi Bhatnagar, Jeffrey Nichols, Choon Hong Peck, Dave Kong, Robert Miller, and A Chris. Long. 2002. Interacting at a Distance: Measuring the Performance of Laser Pointers and Other Devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems Changing our world, changing ourselves - CHI '02*. ACM Press, New York, New York, USA, 33. https://doi.org/10.1145/503376.503383

[26] Alok Mysore and Philip J. Guo. 2017. Torta: Generating Mixed-Media GUI and Command-Line App Tutorials Using Operating-System-Wide Activity Tracing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 703–714. https://doi.org/10.1145/3126594.3126628

[27] Toshio Nakamura and Takeo Igarashi. 2008. An Application-independent System for Visualizing User Operation History. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST '08)*. ACM, New York, NY, USA, 23–32. https://doi.org/10.1145/1449715.1449721

[28] Cuong Nguyen, Stephen DiVerdi, Aaron Hertzmann, and Feng Liu. 2018. Depth Conflict Reduction for Stereo VR Video Interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 64, 9 pages. https://doi.org/10.1145/3173574.3173638

[29] Cuong Nguyen and Feng Liu. 2015. Making Software Tutorial Video Responsive. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*. ACM Press, New York, New York, USA, 1565–1568. https://doi.org/10.1145/2702123.2702209

[30] Nils Petersen and Didier Stricker. 2012. Learning task structure from video examples for workflow tracking and authoring. In *IEEE International Symposium on Mixed and Augmented Reality*. 237–246. https://doi.org/10.1109/ISMAR.2012.6402562

[31] Catherine Plaisant, Anne Rose, Gary Rubloff, Richard Salter, and Ben Shneiderman. 1999. The design of history mechanisms and their use in collaborative educational simulations. In *Proceedings of the 1999 conference on Computer support for collaborative learning - CSCL '99*, Vol. 11. Association for Computational Linguistics, Morristown, NJ, USA, 44–es. https://doi.org/10.3115/1150240.1150284

[32] Suporn Pongnumkul, Mira Dontcheva, Wilmot Li, Jue Wang, Lubomir Bourdev, Shai Avidan, and Michael F. Cohen. 2011. Pause-and-play: Automatically Linking Screencast Video Tutorials with Applications. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 135–144. https://doi.org/10.1145/2047196.2047213

[33] Kevin Ponto, Joe Kohlmann, and Michael Gleicher. 2012. Effective Replays and Summarization of Virtual Experiences. *IEEE Transactions on Visualization and Computer Graphics* 18, 4 (April 2012), 607–616. https://doi.org/10.1109/TVCG.2012.41

[34] Stephan Reichelt, Ralf Häussler, Gerald Fütterer, and Norbert Leister. 2010. Depth cues in human visual perception and their realization

in 3D displays, Bahram Javidi, Jung-Young Son, John T. Thomas, and Daniel D. Desjardins (Eds.). 76900B. https://doi.org/10.1117/12.850094

[35] Steven Schkolne, Michael Pruett, and Peter Schröder. 2001. Surface Drawing: Creating Organic 3D Shapes with the Hand and Tangible Tools. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '01*. ACM Press, New York, New York, USA, 261–268. https://doi.org/10.1145/365024.365114

[36] Rajinder Sodhi, Hrvoje Benko, and Andrew Wilson. 2012. LightGuide: Projected Visualizations for Hand Movement Guidance. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*. ACM Press, New York, New York, USA, 179. https://doi.org/10.1145/2207676.2207702

[37] Geng Sun and Nick Holliman. 2009. Evaluating methods for controlling depth perception in stereoscopic cinematography. In *Proc. SPIE*, Andrew J. Woods, Nicolas S. Holliman, and John O. Merritt (Eds.), Vol. 7237. 72370I. https://doi.org/10.1117/12.807136

[38] Kasim Terzić and Miles Hansard. 2016. Methods for reducing visual discomfort in stereoscopic 3D: A review. *Signal Processing: Image Communication* 47 (2016), 402–416. https://doi.org/10.1016/j.image.2016.08.002

[39] Gerold Wesche and Hans-Peter Seidel. 2001. FreeDrawer - A free-form sketching system on the responsive workbench. In *Proceedings of the ACM symposium on Virtual reality software and technology - VRST '01*. ACM Press, New York, New York, USA, 167. https://doi.org/10.1145/505008.505041

[40] Sean White, Levi Lister, and Steven Feiner. 2007. Visual hints for tangible gestures in augmented reality. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR*. IEEE, 1–4. https://doi.org/10.1109/ISMAR.2007.4538824

[41] E. Wiese, J. H. Israel, A. Meyer, and S. Bongartz. 2010. Investigating the Learnability of Immersive Free-hand Sketching. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium (SBIM '10)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 135–142. http://dl.acm.org/citation.cfm?id=1923363.1923387

[42] Ungyeon Yang and Gerard Jounghyun Kim. 2002. Implementation and Evaluation of "Just Follow Me": An Immersive, VR-Based, Motion-Training System. *Presence: Teleoperators and Virtual Environments* 11, 3 (2002), 304–323. https://doi.org/10.1162/105474602317473240