

RealityCanvas: Augmented Reality Sketching for Embedded and Responsive Scribble Animation Effects

Zhijie Xia*

University of Calgary
Calgary, Canada
zhijie.xia@ucalgary.ca

Kyzyl Monteiro*

IIIT-Delhi
Delhi, India
University of Calgary
Calgary, Canada
kyzyl17296@iiitd.ac.in

Kevin Van

University of Calgary
Calgary, Canada
kevin.van@ucalgary.ca

Ryo Suzuki

University of Calgary
Calgary, Canada
ryo.suzuki@ucalgary.ca

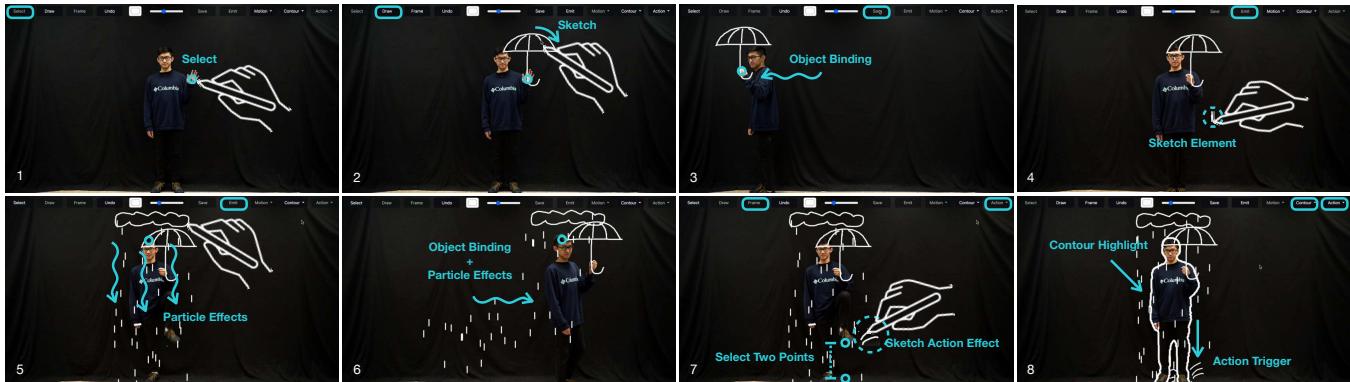


Figure 1: RealityCanvas workflow (from top left to bottom right): 1) select a hand as a tracking point, 2) sketch an umbrella bound to the hand, 3) the umbrella moves when the hand moves, 4) sketch a raindrop, 5) draw a cloud as an emitter line to show particle effects, 6) the cloud also moves with object binding, 7) select the ground and right foot then sketch a water splash, 8) show the water splash and contour highlight based on the stomp action.

ABSTRACT

We introduce RealityCanvas, a mobile AR sketching tool that can easily augment real-world physical motion with responsive hand-drawn animation. Recent research in AR sketching tools has enabled users to not only embed static drawings into the real world but also dynamically animate them with physical motion. However, existing tools often lack the flexibility and expressiveness of possible animations, as they primarily support simple line-based geometry. To address this limitation, we explore both *expressive* and *improvisational* AR sketched animation by introducing a set

of responsive scribble animation techniques that can be directly embedded through sketching interactions: 1) object binding, 2) flip-book animation, 3) action trigger, 4) particle effects, 5) motion trajectory, and 6) contour highlight. These six animation effects were derived from the analysis of 172 existing video-edited scribble animations. We showcase these techniques through various applications, such as video creation, augmented education, storytelling, and AR prototyping. The results of our user study and expert interviews confirm that our tool can lower the barrier to creating AR-based sketched animation, while allowing creative, expressive, and improvisational AR sketching experiences.

*Both authors contributed equally to the paper

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 the author(s) 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.

UIST '23, October 29–November 1, 2023, San Francisco, CA, USA

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0132-0/23/10...\$15.00

<https://doi.org/10.1145/3586183.3606716>

CCS CONCEPTS

• Human-centered computing → Mixed / augmented reality.

KEYWORDS

Sketching Interfaces; Scribble Animation; Augmented Reality; Mixed Reality; Real-Time Authoring

ACM Reference Format:

Zhijie Xia, Kyzyl Monteiro, Kevin Van, and Ryo Suzuki. 2023. RealityCanvas: Augmented Reality Sketching for Embedded and Responsive Scribble Animation Effects. In *The 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23), October 29–November 1, 2023, San Francisco, CA, USA*. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3586183.3606716>

1 INTRODUCTION

In recent years, many augmented reality (AR) sketching tools [1, 12, 45, 66] have been explored in human-computer interaction (HCI) research, thanks to the proliferation of augmented and mixed reality (AR/MR) devices. In contrast to traditional screen-based sketching interfaces such as *SketchPad* [64], *Pegasus* [16], *Draco* [36], and many others [2, 15, 40], AR sketching enables the user to sketch and annotate directly onto a live or recorded real-world scene, opening up untapped opportunities for many applications, such as 3D design and prototyping (e.g., *SymbiosisSketch* [1], *DesignAR* [58]), education (e.g., *Augmented Body* [11], *PaperTrail* [57]), collaboration (e.g., *Vuforia Chalk* [27], *PintAR* [12]), and entertainment (e.g., *Just a Line* [21], *DoodleLens* [71]).

While many existing AR sketching tools focus on *static sketching*, in which sketched elements float in the mid-air without being animated, more recent works have explored *dynamic sketching*, in which the sketched elements can animate and respond to real-world interactions [33, 56, 59, 60, 66]. Such dynamic AR sketching tools allow more interactive and engaging experiences for various applications, however, these existing tools either require both preparation and pre-defined configuration, hindering real-time and improvisational exploration (e.g., *Interactive Body-Driven Graphics* [60], *ChalkTalk AR* [56]) or they do not support freehand drawing except for simple line-based geometries, significantly limiting the flexibility and generalizability of possible animations (e.g., *RealitySketch* [66], *Graphiti* [59], *Sketched Reality* [33]).

To address these limitations, we introduce RealityCanvas, a mobile AR sketching tool that allows the user to embed responsive sketched animation to a live or recorded real-world scene through *freehand* and *real-time* sketching interactions (Figure 1). The goal of RealityCanvas is to enable more expressive yet improvisational AR sketched animations that can respond to and interact with the real world (Figure 2). With both expressive and improvisational capabilities, RealityCanvas allows users to spontaneously blend digital sketches and physical motion in a flexible manner, without the need for prior preparation or planning. This enables interactive exploration and experimentation in real time, unlike prior work [56, 60].

To design our system, we first collect and analyze 172 existing video-edited scribble animation examples to inform us of the possible design space of expressive and comprehensive sketched animations which respond to real-world interactions (Figure 3). Based on the analysis, we identify the following six most common scribble animation effects: 1) **object binding**: dynamically move sketched elements based on the body or object movement, 2) **flip-book animation**: creating an animation based on multiple sketched frames, 3) **action trigger**: one-time animation effect based on the specific action, 4) **particle effects**: spawning many sketched elements around the object, 5) **motion trajectory**: showing the motion

path of objects, and 6) **contour highlight**: line-based animation around the contour of a body or object.

Our main contribution is a set of sketching interaction techniques that allow the user to quickly create all of these expressive scribble animations through direct manipulation with the following three step workflow: 1) *object tracking*: the user specifies a visual entity (e.g., a physical object, a skeletal joint) to track in a live or recorded real-world scene; 2) *sketch elements*: the user adds sketched objects with freehand drawing; 3) *animate sketched elements*: the sketches respond to the world based on the above animation effects. With these range of techniques through a simple workflow, RealityCanvas lets the user sketch responsive AR animation in *real-time* and in *flexible* ways, without prior preparation or planning.

We demonstrate applications of RealityCanvas including social media video creation, augmented classroom education, and storytelling. We evaluate our system through two user studies: 1) a usability study with twenty participants, and 2) expert reviews with seven professional video creators and theatre professionals. The study results confirm that our tool lowers the barrier to creating AR-based sketched animation by enabling the end-user to quickly draw expressive animation in engaging and intuitive ways.

In summary, this paper contributes:

- (1) A taxonomy of existing scribble animation effects that informs us of the possible design space of expressive AR sketched animation.
- (2) A set of techniques that enable authoring of expressive freehand AR sketched animations through improvisational sketching interactions.
- (3) An implementation, applications, and the user evaluation of RealityCanvas.

2 RELATED WORK

2.1 Augmented Reality Sketching Tools

With recent advances of AR technology, there is an increasing number of AR (or VR) sketching tools in both commercial products (e.g., *Just a Line* [21], *TiltBrush* [20], *Gravity Sketch* [22], *Vuforia Chalk AR* [27]) as well as research prototypes (e.g., *SymbiosisSketch* [1], *VRSketchIn* [10], *PintAR* [12]). By leveraging the benefits of *blending* virtual sketches into the physical world, these tools expand the real-world sketching experience for designing (e.g., *SymbiosisSketch* [1], *Mobi3DSketch* [38], *DesignAR* [58]), AR prototyping (e.g., *ProtoAR* [53], *360Proto* [52]), education (e.g., *PaperTrail* [57], *Augmented Body* [11]), collaboration (e.g., *Vuforia Chalk* [27], *PintAR* [12]), and entertainment (e.g., *DoodleLens* [71]).

To go beyond simple static AR sketching, more recent works have started exploring *dynamic and responsive* sketched animation, in which the sketched elements can animate and respond to the corresponding real-world interaction. For example, *Interactive Body-driven Graphics* [60] or *ChalkTalk AR* [56] allows the creation of dynamic sketched animations for augmented storytelling. *RakugakiAR* [30] allows the quick creation of animated sketched characters in AR. Tools like *Rapido* [44] and *Pronto* [45] leverage expressive freehand sketched animations for video-based AR prototyping. Moreover, *RealitySketch* [66], *Reactile* [65], *Graphiti* [59], and *Sketched Reality* [33] let the user bind sketched elements with

a physical object so that the user can create responsive animations with real-time and improvisational sketching interactions.

However, these existing tools often require pre-defined configuration or preparation [45, 56, 60]. For example, prior work [56, 60] distinguishes between a *preparation phase* and *interaction phase*, by requiring the user to prepare assets [60] or pre-programmed animations [56] prior to the performance, which hinders real-time and improvisational sketching exploration. On the other hand, there are some real-time dynamic AR sketching tools [33, 59, 65, 66], but they do not allow freehand drawing, limiting the flexibility and generalizability of possible animations. In contrast, our tool enables both *freehand* and *real-time* AR sketched animation, allowing for more expressive, flexible, and improvisational animation authoring to augment live or recorded real-world interactions through AR.

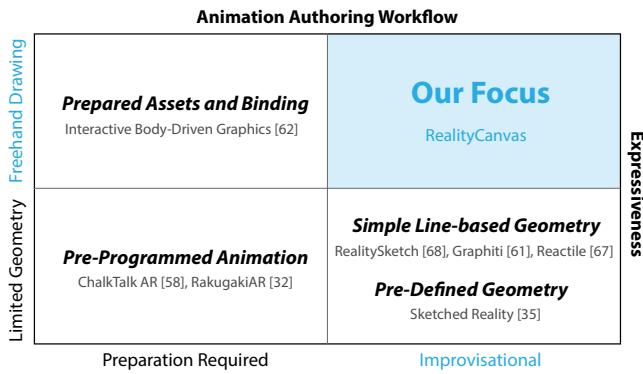


Figure 2: Related work of AR sketched animation tools

2.2 Screen-based Sketched Animation Tools

Apart from AR sketching tools, HCI research also has a long history of screen-based sketching tools (e.g., *SketchPad* [64], *Flatland* [51], *Silk* [39], *Teddy* [17], *Electronic Cocktail Napkin* [14], *ILoveSketch* [2]). In particular, recent works have explored real-time interactive sketched animations, which make static sketches dynamic and animated using the power of computation. For example, *Draco* [36] explores sketched animation through direct sketching interactions based on kinetic textures. Such tools have democratized opportunities for animation authoring, which were previously only accessible to professional animators or video editors. These tools are used for many applications, such as math and science education (e.g., *MathPad2* [41], *PhysInk* [62], *Apparatus* [61]), storytelling (*SketchStory* [43], *Stop Drawing Dead Fish* [69]), data visualization (e.g., *NpakinVis* [5], *Data Illustrator* [48], *Transmogrification* [4], *DataInk* [72]), live music performance (e.g., *Megafauna* [3]), and artistic animation authoring (e.g., *Kitty* [35], *Draco* [36], *Motion Amplifiers* [37]). While these works have expanded possible sketched animation techniques, these interactions are only available on a computer screen. Our goal is to bring these *dynamic sketching* techniques into AR sketching tools by leveraging real-world interactions, as opposed to screen-based interactions.

2.3 Authoring Tools for Augmented Effects

Augmented animation effects — adding visual effects to enhance live or recorded videos — have been used in many videos on YouTube and TikTok. Traditionally, creating these effects often requires extensive skills using professional tools such as Adobe Premiere Pro or Final Cut Pro. For example, tools like *Scribble* [31] also provide a means to create scribble animation effects for recorded videos, but these tools only support *frame-by-frame* authoring without any object tracking or binding, which leads to significant time and effort to create animations, limiting real-time and improvisational exploration. To fill this gap, HCI researchers have explored alternative authoring approaches for such augmented animation effects. For example, *PoseTween* [47] lets the user add virtual effects based on human movement to create augmented action videos. *SnapChat's Lens Studio* [29] also allows face augmentation with a simple authoring workflow. *RealityTalk* [46] also introduces an authoring tool to augment live presentations with embedded graphics. These tools allow for the creation of impressive augmented animation for various applications, including live storytelling [46, 60], entertainment [47], education [13], collaborative discussions [9, 34], data visualization [7], and sports training [6, 26, 63]. In these tools, however, the user needs to prepare *assets* in advance to augment the real world. In contrast, *RealityCanvas* lets the user draw these assets through sketching in real-time, which greatly enhances flexibility and improvisational interactions.

3 DESIGN SPACE OF AUGMENTED AND RESPONSIVE SKETCHED ANIMATION

3.1 A Taxonomy Analysis of Scribble Animation Effects

To better understand the possible design space of augmented and responsive sketched animation, we first collected and analyzed 172 existing video-edited scribble animation examples. *Scribble animation effects* are a common video-editing technique to simply add scribble illustrations over a recorded video for each frame. Because of the flexibility and aesthetics of the scribble effects, such animation techniques are often used in music videos [8, 50], commercial ads [18, 24, 25, 28], trailer promotion videos [19], and presentation videos [23]. Currently, these scribble animation effects are created by manual hand-drawing for each frame, but the resulting videos effectively showcase the potential animation techniques for embedded and responsive sketching.

3.1.1 Corpus and Methodology. We searched for video examples using popular video and image search platforms (e.g., Google Images, YouTube, Pinterest, Behance, Vimeo, TikTok) by using “*scribble animation*”, “*scribble effects*”, or “*scribble animation effects*” as keywords. We first collected 172 scribble animation videos, then filtered these videos by focusing only on those with embedded and responsive animations (i.e., remove those where the animation does not interact with or respond to real-world motion), which gave us a total of 120 videos. Given the collected videos, one of the authors took screenshots for representative examples for each video and conducted open coding to identify a first approximation of the dimensions by categorizing them. After this process, two

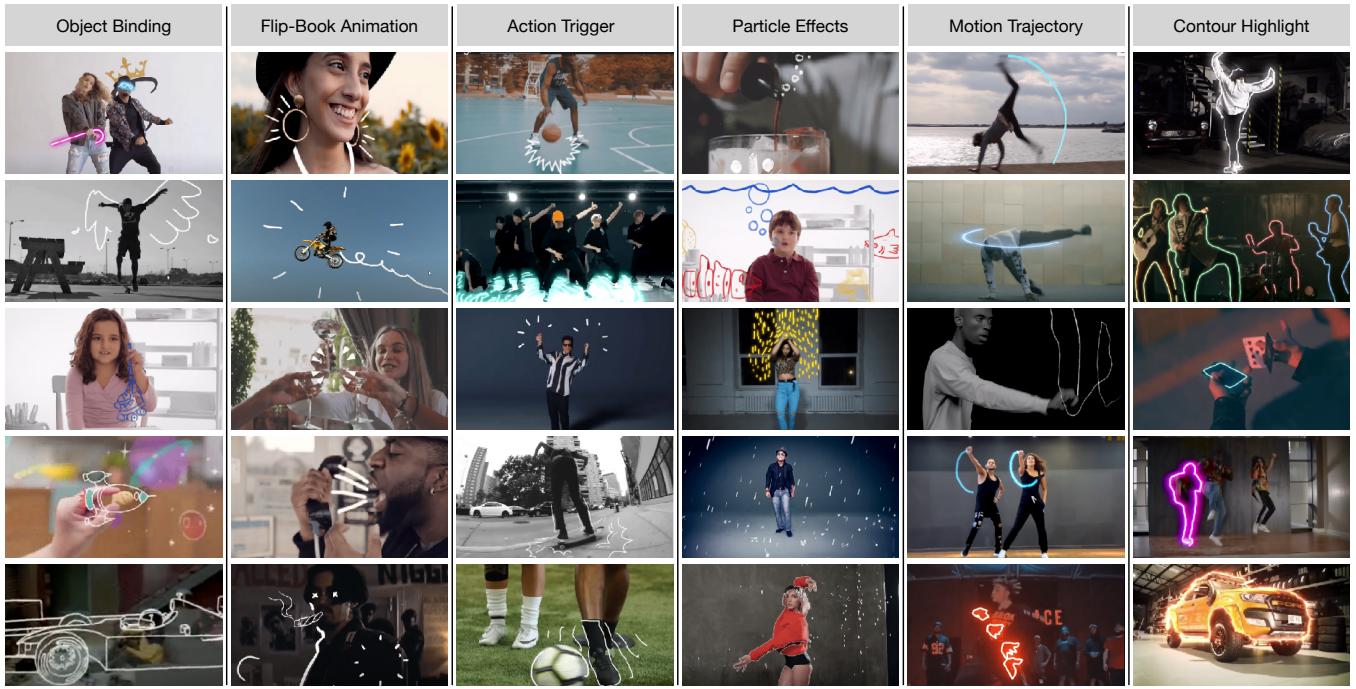


Figure 3: Six common scribble animation techniques based on the taxonomy analysis

authors used an online whiteboard (Miro board) to perform systematic coding with individual tagging. Finally, all authors reflected upon the design space and corresponding examples until all agreed on the consistency and comprehensiveness of the categorization. While we did our best to cover the comprehensive and exhaustive design space, due to the nature of the manual search and taxonomy analysis, we do not argue that our proposed design space is the only way to categorize the existing embedded and responsive sketched animation. Rather, our goal is to identify common animation techniques to get insights for ourselves and the HCI community to explore the possibilities of sketched animation for AR. Our results and example videos will be also available on the website¹.

3.2 Six Common Animation Techniques

1) Object Binding: Object binding is a technique for binding sketched elements to a body or physical object, so that the sketches dynamically move according to the corresponding body or object movement. Examples of object binding include a scribble wing for a human body, scribble glasses or hats for a human face, or scribble exhaust for a car. The attached sketched objects are mostly static in their shape, so that only the position or orientation changes based on the position of the bound object.

2) Flip-Book Animation: Flip-book animation is a common animation technique that can be used for many different purposes. In scribble animation effects, many videos use flip-book animation for morphing the shape of the sketches or adding a dynamic effect to existing sketches. Flip-book animation is often used to combine

different animation techniques. For example, when combined with object binding, it can create animated object binding effects.

3) Action Trigger: Action trigger is a one-time animation effect played when a certain action happens. For example, when two objects collide, the scribble animation can show a bumping effect to highlight the collision. Alternatively, such actions can also be bound to different triggered actions. Possible actions we have observed are hand-clapping, kicking, touching down, hand-waving, and stomping, among many others.

4) Particle Effects: Particle effects are a technique to create repeated animation with many spawned elements. For example, some videos use scribble particle effects to effectively create raindrops, wind flow, or smoke animations. In contrast to action-trigger, particle effects are intended to be repeated effects.

5) Motion Trajectory: Motion trajectories are used to highlight a certain movement of the body or object. By showing the path trajectory, the video can make physical motion more visible and expressive. Such motion trajectory is often used to make human actions stand out, as in dancing and sports videos. Most of the motion trajectories use a simple path animation that shows the afterimage of the body. Some other examples create more expressive motion effects by using a morphing animation.

6) Contour Highlight: Finally, many scribble animations use an animated contour line to trace a body or object. This allows a selected object or body part to stand out by highlighting its contour. When the body posture changes or the object moves, these contour lines keep following to fit the bound object. Most of the contour highlight lines are static, in which the entire contour is covered by a single line, but some examples leverage an animated line, in which a partial line moves across the contour of the object.

¹hidden for anonymized review

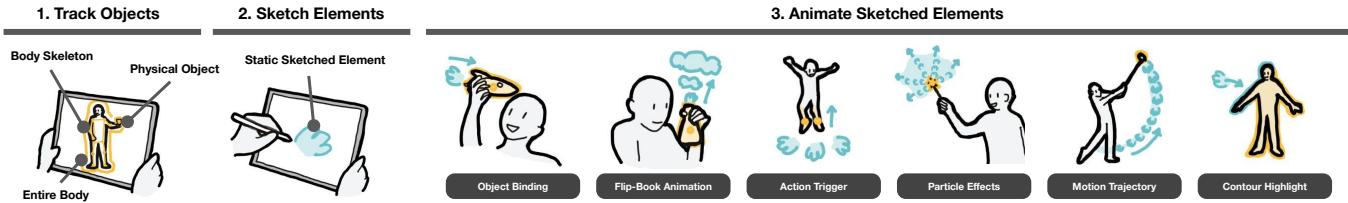


Figure 4: Authoring Workflow of RealityCanvas

4 REALITYCANVAS

4.1 Overview

Based on the above design space, we designed and developed RealityCanvas, an AR sketching tool to augment a captured real-world interaction through responsive scribble animation effects. As a basic setup, RealityCanvas uses a mobile device and a standard RGB camera. Therefore, the user can sketch elements with a finger, a pen, or a mouse on smartphones, tablets, and laptop computers. The RealityCanvas system uses computer vision for object tracking and interaction detection, allowing it to work with either live or recorded videos. For recorded videos, users can pause the video to sketch and define animations, which then start animating when the video is resumed. For live videos, users can embed animations onto a real-time video stream. In our setup, a performer can see the live sketched animation through an external monitor connected to the tablet, similar to [46, 60], and collaborate with the user to create a live performance that can later be saved as a video. All sketched elements are displayed as 2D objects on a screen, without any depth effect based on camera movement. The system is developed using web technology and a live demo is available on our website². To make AR sketched animation accessible to everyone, we will release the source code as an open-source project³.

4.2 Two Key Design Goals

RealityCanvas is designed with two key goals in mind: *expressiveness* and *improvisation*. **Expressiveness** refers to the range of animation effects that the system can provide, and we have designed six types of sketch animation effects (A1-A6 below) to cover a wide range of expressions, informed by the design space analysis of existing scribble animations. This allows for open-ended animation drawing, which was not possible with prior systems [66]. **Improvisation**, on the other hand, enables users to spontaneously create animation *in real-time* without prior preparation or planning, as opposed to prior systems [56, 60]. This is achieved through both freehand drawing and immediate response, which are crucial not only for faster creation but also for unexpected discoveries through playful exploration and experimentation. This design is guided by Victor's principle ("creators need an immediate connection to what they are creating" [68]), and we believe that the support of improvisation enables the users to engage with this new medium to generate new and original ideas they may not have thought of before.

²hidden for anonymous review

³hidden for anonymous review

4.3 Authoring Workflow

At a high-level, RealityCanvas provides the following three-step workflow: 1) object selection and tracking, 2) sketch elements, and 3) animate sketched elements.

Step 1. Object Selection and Tracking: To create embedded and responsive animations, the sketched elements need to be tightly coupled with the body, objects, and/or environments. Thus, capturing and tracking an object is the first step to making the sketches responsive to physical motion. To track an object, the user can simply pause the video input or ask the performer to hold still. Then, the user can select a point of an object (based on color tracking) or a part of a body (based on skeleton tracking). Once selected, the system overlays a green dot on the selected point to indicate the tracked point on the body or object.

Step 2. Sketch Elements: Once the user selects a tracking point, they can start drawing a sketched element. The user can simply sketch any shape by freehand drawing in the scene. The user can also change the color, thickness, and opacity of the stroke when drawing.

Step 3. Animate Sketched Elements: The key feature of RealityCanvas is the animation process. Informed by the design space exploration, the system provides six different ways to animate the sketched elements through direct manipulation.

- A1. Object Binding:** Bind the sketched element to a tracked point on an object or body part
- A2. Flip-book Animation:** Add multiple frames of sketches to create a flip-book animation effect
- A3. Action Trigger:** Define the trigger to specify when the sketched elements appear
- A4. Particle Effects:** Draw a line to spawn many cloned elements
- A5. Motion Trajectory:** Specify a tracked point to show the path by cloning elements
- A6. Contour Highlight:** Select the object to highlight its contour with a sketched line

In the following, we describe each sketched animation technique in more detail.

A1. Object Binding

Once the user selects an object to be tracked, the user can start sketching an element. By default, the sketched elements are automatically bound to the selected tracking point, so that the sketched

element starts moving when the tracked object moves. For example, in Figure 5, the user selects a shoulder as a tracking point, then draws a violin around the tracked point, so that the sketched violin starts moving when the shoulder moves. In the same way, the user tracks a right hand and binds a sketched violin bow; then, the bow also moves based on the user's hand movement.

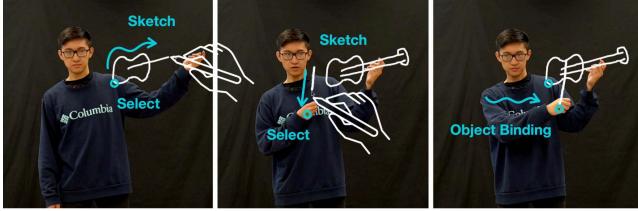


Figure 5: Object Binding

This animation technique is useful for **object binding**, in which the sketch is bound to a body or object to augment the real-world scene. In this simple object binding, the sketched element itself is static and only its position moves and its orientation and scale do not change.

A2. Flip-Book Animation

To make more expressive animation, the system allows the user to create **flip-book animation** effects. In this animation technique, the user can draw an additional sketch as a new frame, and the system shows these sketches one by one to create an animation. For example, in Figure 6, the user first selects a red cup, then draws a circle around the selected point while the performer holds still in live mode. Similarly, the user can pause the recorded video at a specific frame to sketch the circle. Then, the user selects the add a new frame button so that they can draw the next shapes for frame-by-frame animation. Once the user finishes drawing all frames, the user clicks the save button to apply the flip-book animation. In this way, the user can create an animated circle whose size appears to change. The drawn animation is also bound to the tracked object, so that the animated circle moves when the tracked object (red cup) moves.

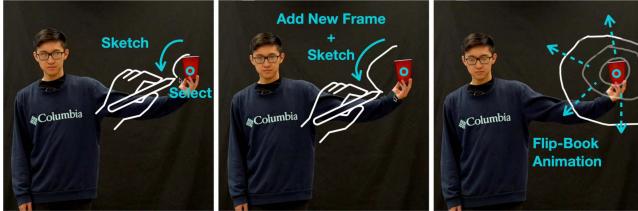


Figure 6: Flip-Book Animation

A3. Action Trigger

The above two animation techniques are always-visible animations. On the other hand, **action trigger** animations allow the user to create animations that appear only when a certain action occurs. To use this feature, the user can simply track multiple points and

specify the user-defined trigger action. In our system, the user can define the trigger based on the distance between two points. For example, in Figure 7, the user first tracks left and right hands, and then the user draws a bumping sketched effect. Once the user clicks the trigger button, then the system continuously tracks the distance between the two points. By default, the system triggers the sketched bumping effect when the distance between the two points decreases below a threshold of a user-defined pixel distance. The user can adjust the trigger parameters using a slider to determine the distance threshold value and a button to change the triggering direction. By toggling the direction, the user can trigger the action when the distance between two points increases rather than decreases. This allows for greater flexibility in defining the trigger conditions.



Figure 7: Action Trigger

By combining this with the above flip-book animation effect, the user can create even more expressive animations based on human action. For example, Figure 1 illustrates how the user can show a water splashing effect when the actor stomps the ground. In this case, the user can select the left foot and the ground as triggering points using both body tracking and object tracking, then draw flip-book animations as trigger action effects. In this way, the user can create a variety of animations based on the user's action.

A4. Particle Effects

The system also lets the user create **particle effects** through improvisational sketching interactions. To create particle effects, the user first sketches an element of the particle, such as a raindrop in Figure 1 or snowflake in Figure 8. Then, the user selects the tracked object and clicks the emit button, which allows the user to draw an *emitting line* to spawn new elements for particle effects. As shown in Figure 8, the same snowflake sketches are randomly emitted from the emitter line for the snow effects. The system offers two options for customizing the particle effect parameters: 1) the user can draw a *motion path* to specify the path each particle follows, and 2) the user can change the *speed* of particle movement using a slider in a menu. These options were inspired by the interactions presented in *Draco* [36].

Our interaction technique is inspired by *Draco* [36]'s kinetic texture and our tool also supports similar interactions, such as allowing the user to draw a trajectory of the emission so that the user can control the path the emitted particles follow. But unlike *Draco*, which focuses on screen-based interaction, the drawn emitter line in RealityCanvas is bound to the selected physical object so that if the object moves, the emitter line follows along. This enables a broader interaction space, such as a magic wand particle effect from a physical stick or showing an air flow bound to a dryer.



Figure 8: Particle Effects

A5. Motion Trajectory

In object binding, the sketched element simply moves and follows the tracked point, but by duplicating the sketched element for each tracked position, the system can also create a ***motion trajectory*** like ghost effect of the sketched elements. For example, in Figure 9, the user selects a left hand as a tracked point and sketches a simple dot. When the user clicks the motion button, the dots become a line to show the trajectory of the hand movement. With a certain time period (by default 30 elements), the cloned elements start disappearing, so that the motion trajectory disappears with a certain length.

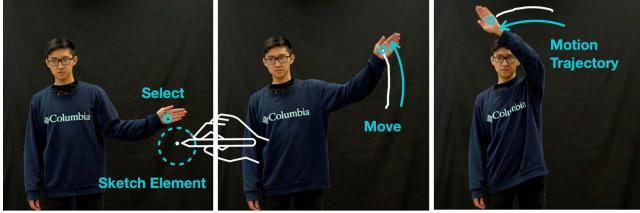


Figure 9: Motion Trajectory

This animation technique is useful when the user wants to visualize the trajectory of a tracked object, such as with a dancing motion or throwing a basketball. The user can also change the parameters of the motion trajectory, such as how fast the elements should disappear, or the opacity or scale of each cloned element. By leveraging these parameters, the user can also create more expressive motion effects.

A6. Contour Highlight

Finally, the system lets the user create ***contour highlight*** effects for a tracked body or object. To do so, the user first selects the contour button and then taps the body or object the user wants to highlight. By default, the enclosed object or body is highlighted with a drawn line. This line can be changed according to the body motion or object movement, as RealityCanvas always tracks and displays the outermost object contour seen in Figure 10. In a similar way, the user can also create animated contour highlights by selecting the appropriate option. For example, in Figure 10, the user can also show an animated contour line around the body.

The user can also fill in the enclosed object. For example, the user can change the fill color of the body by combining the action-trigger effect. The user first selects the foot and defines the trigger based on the foot location, the user can create the highlighting effect along with the animated elements.



Figure 10: Contour Highlight

4.4 Implementation Details

4.4.1 Sketching Interface. RealityCanvas uses HTML Canvas as the main sketching interface. Since most of the sketched elements are 2D, we decided to embed sketches onto the 2D canvas screen. All sketched elements are drawn in SVG format on the canvas. The system also uses Konva.js and Anime.js as supporting graphics and animation libraries, respectively. All elements are calculated and rendered as 2D objects. For example, the distance between two tracked points is based on the pixel value on the screen, rather than the physical distance. Therefore, depth effects and 3-dimensional rotation are currently not supported (although the user can still draw sketches which look 3-dimensional using perspective effects).

4.4.2 Object Tracking. For object tracking, we use a simple color-based tracking method that has been shown to provide fast and reliable real-time tracking for AR animation applications [66]. In our current implementation, the system uses OpenCV JS to track objects based on their RGB values, with a certain threshold to filter out pixels outside of the specified range. The system first obtains the RGB value based on the selected X-Y point by sampling the pixel at the corresponding coordinates in the video frame. The system then filters out all pixels outside the specified RGB range. Then the system identifies the largest continuous contour as the tracked object. . The RGB value range is $(r \pm 10, g \pm 10, b \pm 10)$, where (r, g, b) is the sampled RGB value of the selected point. Given the largest contour, we obtain the center point of the object, which is used to determine the object's location in the video frame. This information is then used to track the object's movements over time and update its position on the screen accordingly. We evaluated the robustness of our color tracking system under four different lighting conditions:

- (1) Dim artificial lighting: Simulated using a single light source in the living room.
- (2) Bright artificial lighting: Achieved by activating four light sources in the living room.
- (3) Dim natural lighting: Simulated by creating subdued sunlight conditions in the living room.
- (4) Bright natural lighting: Achieved by exposing the living room to direct sunshine.

We tracked two types of objects: a reflective object (a fake apple) and a non-reflective object (a tennis ball). The success of the tracking was determined by the accurate movement of a green dot, indicating successful object tracking, while failure was identified when the green dot failed to follow the object. As a result, we found that the system performs more accurately in bright lights than in dim lighting. In dim artificial light, the system successfully tracked the

object 5 times and failed to track it 5 times. However, in bright artificial light, it successfully tracked the object 8 times and failed to track it only 2 times. We also discovered that the mesh of an object can impact the accuracy of color tracking in bright light. Objects with reflective mesh in bright lighting often result in a high false negative rate. Our system performed best when the objects had distinct solid colors.

4.4.3 Body Tracking. In our current implementation, the system tracks body parts based on MediaPipe. MediaPipe provides 33 body points based on the human skeleton. When the user clicks a point on the screen, the system identifies and continuously tracks the closest body point among the 33 body points based on X-Y value. For body segmentation, we used MediaPipe and OpenCV JS to obtain the masked shape of the body. Based on the masked image, we convert the body shape into the simplified SVG path for contour line animation in each frame when the contour highlight is enabled.

4.4.4 Prototype Setup. Since all of the components including the color tracking and body tracking are client-side systems, we do not have any server to run the system. During the prototyping phase, we tested with several devices, including Pixel 3 XL, Pixel 6, Linux Desktop, Windows machine, and Android tablet. (Due to the Safari browser's restriction, the iPhone/iPad did not work, although this problem should be resolved in the future.) For the demo and user evaluation, we used Samsung Tab S8 Ultra Android Tablet (Display: 14.6-inch, CPU: Qualcomm SM8450 Snapdragon 8 Gen 1, GPU: Adreno 730, RAM: 12GB) with a built-in stylus pen. In this setup, the system achieved the targeted 60 FPS at all times with the tablet stock camera during our testing for both color tracking and body tracking. Additionally, it is worth mentioning that we observed a decrease in frame rate when using an external high-resolution camera.

5 APPLICATIONS

5.1 Social Media Video Creation

First, our system allows users to quickly create videos for social media platforms such as TikTok, YouTube, and Instagram. Currently, video creation and editing with these animation effects require specific skills and non-trivial amounts of time. People can use our system to record their videos with animated effects much more easily. For example, the user can shoot a dancing or music video with real-time scribble animation effects, such as *motion trajectory* of dancing movement, *contour highlight* to make a person stand out, and *action trigger* to highlight the claps or steps.

5.2 Augmented Storytelling

Using our system, the user can also easily create augmented stories [46, 60]. For example, by using animated *object binding*, the user can augment the body with sketched illustrations and animation, like adding a Superman costume or pirate features. In this scenario, the user can sketch on another user's performance, which can be streamed to a large screen in real time so that the performer can see the current performance. Such augmented storytelling would greatly expand, for example, children's storytelling experience with dynamic effects.

5.3 Classroom Education

The user can also add dynamic visual annotations to a physical phenomenon to provide better explanations for science or physics classrooms. In this scenario, similar to the storytelling scenario, the teacher can stream the screen view to a large display to make the animation visible to the entire classroom. For example, by using *particle effects*, the user can visualize the airflow around a levitated ping pong ball. The user can explain how water circulates on Earth by using an animated sketch of rain and clouds with *particle effects* by augmenting a physical model such as terrains. Similar to *RealitySketch* [66] and *HoloBoard* [13], such augmentations can enhance the learning experience for classrooms.

5.4 Prototyping AR Applications

Finally, the RealityCanvas system can also be used to create an interactive and animated prototype for AR applications. Similar to existing AR prototyping tools [44, 45, 52, 53], embedded sketches allow the user to simulate AR experiences for low-fi prototypes quickly. For example, the user can quickly create an animated AR Christmas card using *object binding*. With *action trigger*, the user can create some interactive AR linked to an action, like the opening of the card. The user can also sketch the animation for augmented music visualization like AR Music Visualizer [67] to test how different effects can enhance the music experience.

6 USABILITY STUDY

6.1 Method

We evaluate our system with two user studies: a usability study and expert reviews. The usability study was designed to assess the system's usability, with participants using it individually. On the other hand, the expert study aimed to gather detailed feedback by having participants utilize the tool in pairs or groups.

The first study's goal is to evaluate our system's usability. To do so, we recruited twenty participants (14 male, 6 female, age: 19 - 29) from our local community. The usability study consists of the following two tasks.

6.1.1 Standardized Task. We first demonstrated all of the basic functionalities. We then showed an animation video (Figure 1) we created with our system. The video featured all the functionalities. The animation video had the following elements: 1) *object binding* between hands and umbrella, 2) *flip-book animation* for the triggered water splash, 3) *action trigger* based on the stomp, 4) *particle effects* with rain drops, 5) *motion trajectory* of the hand movement, 6) *contour highlight* of the whole body. We asked the participants to recreate this animation without instructing them how to make it. The goal of this task is to let the user use all of the functionalities to measure each feature's usefulness and understandability.

6.1.2 Exploration Task. Once the user finished the identical task, we moved on to the exploration task. We asked the participants to create their own sketched animation from their imagination. For inspiration, participants were shown existing examples of scribble animations samples from the corpus we collected for our taxonomy analysis as well as many animation videos created by the authors. Each participant was asked to think of a scenario that involves one or more objects driven by a human action or object movement.

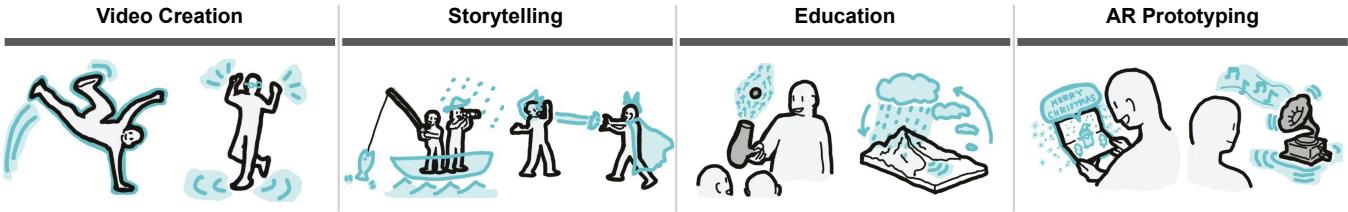


Figure 11: Applications Scenarios

Then one of the authors performed the actions for the participant while the participant added sketches on top of it by themselves. The goal of this task was to measure the expressiveness and flexibility of our system through open-ended animation scenarios.

We designed these tasks as “*usage evaluation*” based on the HCI toolkit evaluation strategy [42]. Instead of a controlled experiment, we chose to conduct the usage evaluation because there was no clear baseline to compare with RealityCanvas. For example, existing video-editing tools have a completely different workflow, so we cannot compare them with our system. Therefore, for the usability study, we focused on the usage evaluation for the end user. On the other hand, we also conducted an expert review to compare our approach with existing video-editing or animation techniques from an expert’s point of view.

Both tasks were conducted in a research lab. All of the participant’s performances were recorded for objective measurements (e.g., task completion time). After the session, we asked the participants to provide feedback through an online questionnaire. In total, the study took approximately 60 minutes, and participants were compensated \$10 CAD.

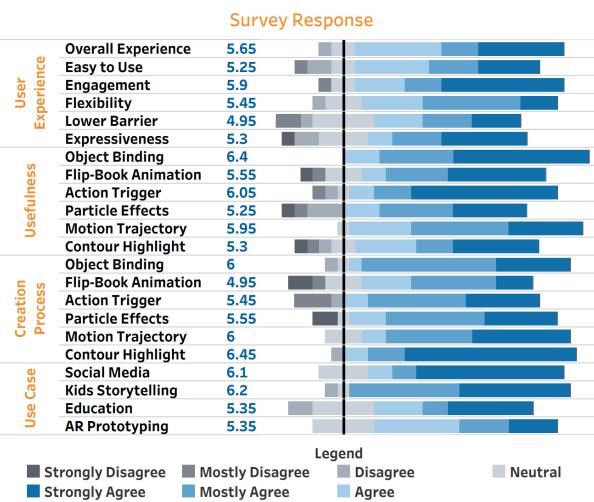


Figure 12: Usability Study Results - A graph summarizing the 7-point Likert scale responses for 20 participants.

6.2 Results

1) Overall Experiences. Figure 12 summarizes the 7-point Likert questionnaire response of the usability study. Overall, participants responded positively to the user experience (5.65/7). “*P7: The entire system was quite neat and fun to use. When watching the examples, I imagined myself re-creating them.*” The participants also responded positively about the interface and workflow. From the questionnaire results, the participants found the system easy to understand (5.25/7), and from the comments, many participants mentioned that the interface was self-explanatory (P2, P5) and easy to learn (P8, P12). “*P5: The interface and the workflow were simple enough to learn that little explanation was required.*”

Most participants did not have animation or video creation experience, but the participants felt that the system could be easy to use even for novice users. The participants generally agreed that the system would lower the barrier to creating AR animation (4.95/7). “*P15: It was very easy to pick up the tools and nuance of the system. Further, I can imagine many ways in which both amateurs and professionals can make interesting things using it.*” For the standardized task, all participants could complete the given task without any explicit assistance from the researcher. For the task completion time, the participants finished the standardized task on average in 3.25 min (Min: 1.48 min, Max: 8.2 min, SD: 1.58 min).

While the participants did not see any issues with the workflow, some complained about the system’s performance. For example, the participants commented that the animation was not smooth (P14) and sometimes lagged (P9, P11). Also, the participants commented that the tracking accuracy could be improved (P18). Additionally, the participants wished for finer-grained body and object tracking (P12, P19).

2) Quality and Expressiveness. One of the goals of our system is to increase the expressiveness and flexibility of sketched animations with freehand drawing. From the questionnaire results, participants found the system flexible (5.45/7) and agreed that the quality and expressiveness of the created result are high (5.3/7). For example, many participants found the system flexible (P2, P10, P18, P20) to create various animations, “*P20: Based off of what I tried to do, it has enough features to support different creative ideas.*” The participants also positively commented on the expressiveness of freehand drawing. “*P7: As mentioned previously, I think once one has a better understanding of the system (more than 20 mins), there are endless possibilities.*” For the exploration task, the participants created various animations with 5.24 min on average (Min: 2.21 min, Max: 18.52 min, SD: 4.87 min) and median 8.87 min. Figure 13 illustrates the example animations created by the participants. Among the

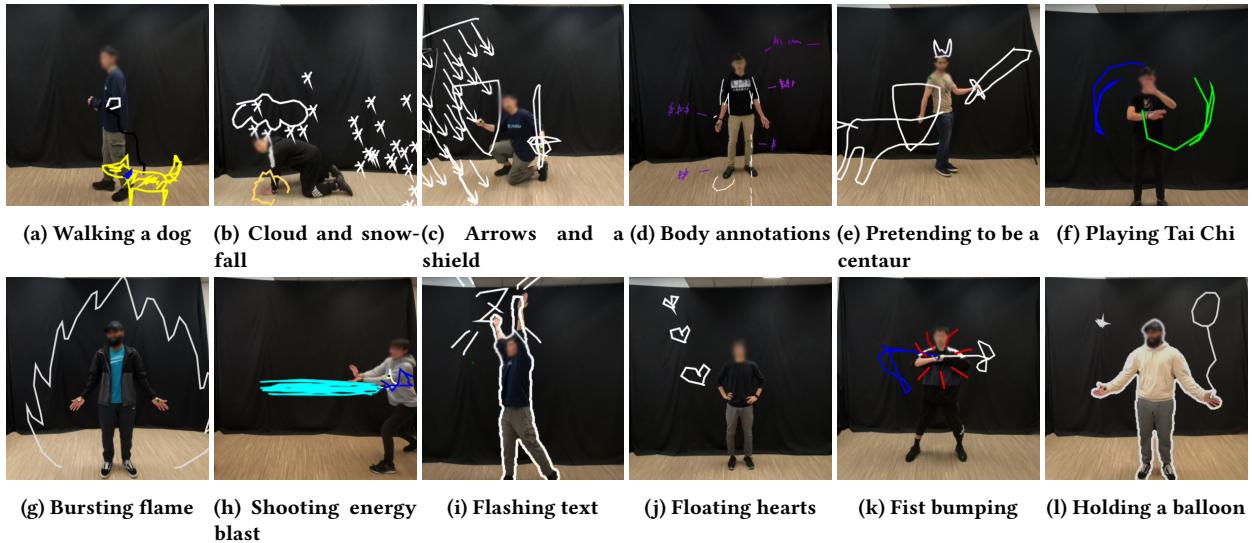


Figure 13: Exploration study scenes created by participants

20 scenarios, we observed some common patterns such as dancing highlights, using superpowers, and equipping weapons.

3) Creation Process and Improvisation. During the creation process for exploration tasks, we observed that most participants did not have a clear idea of what they wanted to create at first. Instead, they developed their narratives by drawing a simple shape and experimenting with supported animation features. For instance, in the case of the arrows and shield example (c), the participant first drew a simple arrow symbol and played with the emitting line. As they did so, the arrow began to look more like arrows from an enemy, which led to the story of a fighting soldier in the Middle Ages. In our study, we observed this type of “unexpected creation” happening with many of the examples, including a simple circle turning into a balloon (l), a centaur inspired by a sword and shield (e), and an energy blast derived from an unexpected action trigger of a simple line (h). While a few participants had a clear vision at the beginning, like body annotation for biology and anatomy education (d), overall, the participants appreciated the ability of improvisational and spontaneous exploration enabled by freehand drawing and interactive experiments.

4) Animation Techniques. We also asked for user feedback for each animation feature. From the questionnaire responses, the participants positively responded to most of the features regarding the usefulness and understandability of the creation process for each feature: object binding (6.4/7, 6/7), flipbook animation (5.55/7, 4.95/7), action trigger (6.05/7, 5.45/7), particle effects (5.25/7, 5.55/7), motion trajectory (5.95/7, 6/7), contour highlight (5.3/7, 6.45/7). We also counted the frequency of each animation technique used in the exploration study: object binding (14/20), flip-book animation (10/20), action trigger (9/20), particle effects (12/20), motion trajectory (7/20), contour lines (8/20). From our observations, most participants began by binding a simple shape and then gradually added more complex animations, such as emitting snowflakes, to

refine their scenarios. Some participants chose not to use our contour highlight, particle effects, and motion trajectory features but instead used flip-book animations to achieve visually similar results by considering the depth effects. From the questionnaire response and usage data, overall, the participants liked all six animation techniques.

The participants also provided comments on individual features. “*P15 on flip-book animation: This feature was particularly useful when paired with other features such as action-trigger. It is relatively simple to create with a few seconds of instruction.*” “*P19 on action trigger: I really liked the action-trigger feature. I see this as being the most useful feature coupled with the frame drawing feature.*” “*P4 on motion trajectory: cool feature, could be used in videos to make movements more fancy and interesting.*” “*P17 on particle effects: Very creative feature for lasting animation like weather and environment effects.*” “*P12 on contour highlight: Easy to implement and useful to extract just the object you want to animate.*”

5) Potential Use Cases. The participants found RealityCanvas useful for social media (6.1/7) and storytelling for children (6.2/7). All participants saw potential benefits of RealityCanvas for classroom teachings (5.35/7). In particular, the participants felt that RealityCanvas provided an easy and engaging way to create educational videos and lab tutorials, which can be helpful for both teachers and students (P3). They also expressed interest in using RealityCanvas for a variety of other applications, including presentations (P8), sports analysis (P15), and safety and training demonstrations (P17).

7 EXPERT REVIEW

7.1 Method

We also conducted an expert review to gain in-depth feedback. We recruited seven experts mainly from two groups - video creator experts and experts with a background in theatre. E1: full-time animation artist who has 18K followers and 1.8M upvotes on TikTok,

E2: professional video creator who has 100K followers on BiliBili, E3: professional video creator who has 97K followers on YouTube, E4: researcher in AR and animation creation fields in a renowned company, E5: theatre performer with 10 years experience, also a Computer Science Ph.D. student with interests in Mixed Reality E6: theatre performance director/creative head with 25+ years of experience E7: theatre director and actor with 5 years of theatre experience

We interviewed video creators to qualitatively compare our system with existing tools. While for experts with a background in theatre, we evaluate our system for improvisation, storytelling, and AR prototyping perspectives. We recruited theatrical performers and directors strongly interested in creating AR-based live performances via university mailing lists. We conducted a workshop to let them use our system to explore and prototype their AR-based performance so that they have a better understanding of the different features our tool offers and the workflow that our tool can support. Designers and technical developers also joined the workshop, thus ensuring an opportunity for collaboration among professionals from different backgrounds and with different skill sets. We were specifically interested in evaluating the qualitative measurements of the improvisational ability of our system. After the workshop, we conducted in-person interviews with each participant to reflect on their prototyping and ideation experiences and gain insights into how they can use our system for augmented live performances and storytelling. The interview lasted approximately one hour for each expert, and we provided \$20 CAD for their participation.

7.2 Results

All experts were excited by the potential of our tool. They found the tool to be intuitive (E1, E3, E4), useful (E3, E5, E7), playful (E1, E6), compelling (E2), general purpose (E1), and versatile (E3). Overall, the experts found the system enabled a creative, improvisational, and expressive range of freehand scribble animations, which they highlight in the strengths and limitations of our system.

1) Comparison to Current Practices and Workflows: Experts found the authoring workflow to be intuitive. They thought the tool could greatly reduce the time and cost of adding embedded animations. “*E2: It takes considerable time to add effects in Adobe AfterEffects than this system’s process of sketching while recording*”. E4, an HCI expert, pointed out that RealityCanvas has a unique proposition of real-time creation and interaction. E4: “*Current tools and research projects are either made for the pre-production or post-production stage of content creation*.” In addition, experts found that the identified six animation techniques were extensive and covered the most commonly seen scribble effects (E1-E4). Theatre experts saw a similarity between the director-performer relationship in theatre and the animator-actor dynamic between tool users (E5-E7). For example, just like the director, the animator directs the actor and makes decisions about the scene, props, and story. Overall, experts thought that our tool could complement their existing workflows.

2) Encourages Discovery and Ideation: Experts shared that RealityCanvas could be used for ideation as you can rapidly test, discard and recreate ideas (E1, E3, E4). E3: “*I can see it as an AR prototyping tool, not for the final product but a quick lo-fi mockup*

which enables idea generation”. Experts also found that our system’s real-time feedback helps ensure that their ideas are technically feasible. E5: “*We spend time coming up with gestures and actions only to find they are not being detected. The ability to interact in real-time allows us to make these beautiful discoveries because now you have some specifics available, instead of making whatever we like now and thinking about the feasibility later*” Other than ideas for augmented performances, experts also found that the tool could be useful in traditional theatre. E7: “*You can also sketch props, costumes, and set designs and even how they interact with each other, which could be especially useful in rehearsals*.”

3) Improves Communication and Collaboration: An important aspect that experts highlighted was that it could help improve communication and collaboration, specifically by reducing miscommunication between teams. E6: “*... and he and I never got to the same visual language. So I think a shared language like this would be really helpful*.” As E6 and E7 had been directors, they remarked that it would benefit directors as they often need to communicate with multiple teams with different skills and technical languages.

4) Lightweight Design: Experts appreciated that RealityCanvas is lightweight with a mobile AR web-based interface and requires minimal setup. Video creators appreciated that the system was quick and did not require expertise. “*E5: I could imagine the world would become more interesting with RealityCanvas on a mobile phone. I could just go out to the streets and shoot a movie with animations and effects*”. The experts with a theatre background also appreciated the low-cost setup (E5-E7) as they pointed out that small theatre groups often have low budgets (E6, E7). E6: “*We don’t want to bring in a ton of projectors or control lighting and other tedious and costly arrangements. So making it available on mobile and a wider audience is precisely why it is so exciting!*”. Experts also appreciated the free-hand sketched modality of embedded animations as it gives more agency over the visuals, adds personality, and makes the video look authentic. E3: “*I have worked with some designers who want to add sketches to their videos as it gives a lot of personality to the video and makes it authentic*.”

Limitations and Future Suggestions. On the other hand, the experts also shared many limitations of RealityCanvas, which were not identified in the usability study. For example, E2, E3, and E5 raised the concern that RealityCanvas may not be suitable for those lacking sketching expertise. “*E2: the prototype is only usable for artistically inclined users*.” All the experts agreed that freehand drawing provides freedom and flexibility but requires the user to draw a new sketch every time they use it. Experts shared that the system needs a few modifications to replace video editing platforms (E1-E4). “*E4: It does not have a video and audio track and other video editing functionalities*”. Also, although our system allows the user to erase the sketches in the drawing stage, the created effect cannot be removed once applied to the system. “*E1: For a drawing system, an ‘undo’ button is essential*.” and “*E2: I strongly request a fully functional ‘undo’ button*.”

Improvisational vs. Preparation-Based Authoring. Based on the rich insights we gained from the expert interviews, we reflected on the contrast between improvisational and preparation-based authoring systems. Improvisational authoring tools provide users

with more flexibility during the creation process, allowing them to create and modify content in real time without the need for prior planning and preparation. This can be useful for quickly exploring different ideas and possibilities, particularly during the ideation phase. However, improvisational tools can result in lower production quality as a trade-off. Also, improvisational performances can be challenging to control and duplicate, resulting in undesired animation sequences. In contrast, preparation-based tools give users more control over the final product but can be inflexible and time-consuming. Choosing between improvisational and preparation-based tools depends on the user's goals and needs. Future research could explore tools that allow users to switch between improvisational and preparation-based approaches or combine the best features of both in a single authoring tool.

8 LIMITATIONS AND FUTURE WORK

On top of the feedback and suggestions from the usability study and expert interview, we also see many potential future work directions to expand our approach.

3D Sketched Animation with HMDs: Our current implementation uses screen-based mobile AR, but integration with head-mounted displays (HMDs) like Microsoft Hololens or Meta Quest Pro will allow more blended and immersive experiences through mid-air drawing like *Norman* [54]. However, this requires a more complex implementation, as it needs 3D object deformation, as opposed to a simple 2D line animation (e.g., *Motion Amplifier* [37] vs. *Layered 3D Animation* [49]). Future work should explore these 3D animations through either 3D deformation [49], simple frame-by-frame animations [54], or using 2D sketches on an invisible surface embedded in 3D space [33].

Exploring More Complex and Multi-Modal Action Triggers: Currently, the system only supports a simple image-based action trigger (e.g., the distance between two points), which limits complex animation behaviors for many use cases (e.g., storytelling, AR prototyping, etc.). If the system could support more complex and expressive action triggers, like the capability to define multi-modal triggers or a combination of multiple triggers, it would further expand possibilities. For example, if we could use a specific sound as a trigger, like a speech bubble animation when talking or experiences similar to *AR Music Visualizers* [67] and *Music Bottles* [32].

Leveraging Real-time Parameters: In previous works of real-time responsive sketches like *RealitySketch* [66], *Interactive Body-driven Graphics* [60], *GesturAR* [70], or *Reactile* [65], the real-world parameters are often incorporated (e.g., an idea of manipulating sketches based on a distance between hands has been explored in *GesturAR* [70]). In our prototype, we did not explore this aspect except for a simple action trigger to make our interaction and system simple, but the integration of real-time parameters could enhance the expressive animation. For example, based on the distance between two hands, the user could change the scale or sketched objects, rotate the objects, or change the number of particles in dynamic and responsive ways.

Integration with Physics Simulations and Other Pre-defined Animations: The way to create responsive sketched animation can be situated in the spectrum between user-defined and pre-defined animations [66]. Currently, our sketched animation focuses more

on the user-defined aspect, in which the user can define most of the animations on demand. However, the integration with pre-defined animations can enrich the expression. For example, the system could integrate physics simulations to make the sketched animations interact with the real world, similar to *Sketched Reality* [33]. Another pre-defined animation like character motion or pre-programmed behavior like *RakugakiAR* [30] or *ChalkTalk AR* [55, 56] is another possibility to broaden the expression. Future work can investigate this aspect to create more complex and expressive animation.

9 CONCLUSION

In this paper, we introduced RealityCanvas, an augmented reality sketching tool to bring life to AR sketches based on real-world interactions. The design of our system is informed by the taxonomy and design space exploration of existing scribble animations. To support these animation techniques, this paper contributes a set of interaction techniques that enable real-time sketched animation without preparation or pre-defined configurations. Our user study confirms that our tool is easy to use yet enables expressive sketched animation. We discussed limitations and future work to highlight broader challenges to enable more expressive AR sketched animation.

REFERENCES

- [1] 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*. ACM, 185.
- [2] Seok-Hyung Bae, Ravin Balakrishnan, and Karan Singh. 2008. ILoveSketch: as-natural-as-possible sketching system for creating 3d curve models. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*. 151–160.
- [3] Samuelle Bourgault and Jennifer Jacobs. 2021. Preserving Hand-Drawn Qualities in Audiovisual Performance Through Sketch-Based Interaction. In *2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 1–10.
- [4] John Brosz, Miguel A Nacenta, Richard Pusch, Sheelagh Carpendale, and Christophe Hurter. 2013. Transmogrification: causal manipulation of visualizations. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 97–106.
- [5] William O Chao, T Munzner, and MVD Panne. 2010. Napkinvis: Rapid pen-centric authoring of improvisational visualizations. In *IEEE InfoVis*.
- [6] Hua-Tsung Chen, Yu-Zhen He, and Chun-Chieh Hsu. 2018. Computer-assisted yoga training system. *Multimedia Tools and Applications* 77, 18 (2018), 23969–23991.
- [7] Zhutian Chen, Yijia Su, Yifang Wang, Qianwen Wang, Huamin Qu, and Yingcai Wu. 2019. MARVisT: Authoring Glyph-based Visualization in Mobile Augmented Reality. *IEEE transactions on visualization and computer graphics* (2019).
- [8] Bombay Bicycle Club. [n.d.]. Carry Me Official Music Video. <https://youtu.be/OZCfqhRgj0Q>
- [9] Pierre Dillenbourg. 2013. Design for classroom orchestration. *Computers & Education* 69 (2013), 485–492.
- [10] Tobias Drey, Jan Gugenheimer, Julian Karlbauer, Maximilian Milo, and Enrico Rukzio. 2020. VRsketchin: Exploring the Design Space of Pen and Tablet Interaction for 3D Sketching in Virtual Reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.
- [11] Hasan Shahid Ferdous, Thuong Hoang, Zaher Joukhadar, Martin N Reinoso, Frank Vetere, David Kelly, and Louisa Remedios. 2019. "What's Happening at that Hip?" Evaluating an On-body Projection based Augmented Reality System for Physiotherapy Classroom. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.
- [12] Danilo Gasques, Janet G Johnson, Tommy Sharkey, and Nadir Weibel. 2019. Pintar: Sketching spatial experiences in augmented reality. In *Companion Publication of the 2019 on Designing Interactive Systems Conference 2019 Companion*. 17–20.
- [13] Jiangtao Gong, Teng Han, Siling Guo, Jiannan Li, Siyu Zha, Liuxin Zhang, Feng Tian, Qianying Wang, and Yong Rui. 2021. HoloBoard: a Large-format Immersive Teaching Board based on pseudo HoloGraphics. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 441–456.
- [14] Mark D Gross. 1996. The electronic cocktail napkin—a computational environment for working with design diagrams. *Design studies* 17, 1 (1996), 53–69.

- [15] Mark D Gross and Ellen Yi-Luen Do. 1996. Ambiguous intentions: a paper-like interface for creative design. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*. 183–192.
- [16] Takeo Igarashi, Sachiko Kawachiya, Hidehiko Tanaka, and Satoshi Matsuoka. 1998. Pegasus: a drawing system for rapid geometric design. In *CHI’98 conference summary on Human factors in computing systems*. 24–25.
- [17] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. 2006. Teddy: a sketching interface for 3D freeform design. In *ACM SIGGRAPH 2006 Courses*. 11–es.
- [18] Budweiser Inc. [n.d.]. Budweiser + NBA Commercial Ads. <https://vimeo.com/577348991>
- [19] FX Networks Inc. [n.d.]. Dave Season 1 Trailer. <https://youtu.be/Qhak9a681zM>
- [20] Google Inc. 2016. TiltBrush. <https://www.tiltbrush.com/>
- [21] Google Inc. 2018. Just a Line. <https://justaline.withgoogle.com/>
- [22] Gravity Sketch Inc. 2017. Gravity Sketch. <https://www.gravitysketch.com/>
- [23] ING Direct Inc. [n.d.]. Credit Immobilier by David Dalidec. <https://vimeo.com/170446327>
- [24] Microsoft Inc. [n.d.]. Your Potential, Our Passion. https://youtu.be/it-Vt2M_-ZU
- [25] Mondelez International Inc. [n.d.]. Oreo: Kid Inside Astronaut / Knight. <https://vimeo.com/213028982>
- [26] NEX Team Inc. 2019. HomeCourt. <https://www.homecourt.ai/>
- [27] PTC Inc. 2017. Vuforia Chalk AR. <https://chalk.vuforia.com/>
- [28] Pepperidge Farm Inc. [n.d.]. Goldfish Grahams TV Spot, Space Explorer. <https://www.ispot.tv/ad/7Cz8/goldfish-grahams-space-explorer>
- [29] Snap Inc. [n.d.]. Snap AR Lens Studio. <https://ar.snap.com/lens-studio>
- [30] Whatever Inc. [n.d.]. RakugakiAR. <https://whatever.co/work/rakugakiar/>
- [31] ZipoApps Inc. [n.d.]. Scribbl. <https://zipoapps.com/scribbl/>
- [32] Hiroshi Ishii, Ali Mazalek, and Jay Lee. 2001. Bottles as a minimal interface to access digital information. In *CHI’01 extended abstracts on Human factors in computing systems*. 187–188.
- [33] Hiroki Kaimoto, Kyzy Monteiro, Mehrad Faridan, Jiatong Li, Samin Farajian, Yasuaki Kakehi, Ken Nakagaki, and Ryo Suzuki. 2022. Sketched Reality: Sketching Bi-Directional Interactions Between Virtual and Physical Worlds with AR and Actuated Tangible UI. *arXiv preprint arXiv:2208.06341* (2022).
- [34] Shunichi Kasahara, Valentin Heun, Austin S Lee, and Hiroshi Ishii. 2012. Second surface: multi-user spatial collaboration system based on augmented reality. In *SIGGRAPH Asia 2012 Emerging Technologies*. 1–4.
- [35] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, and George Fitzmaurice. 2014. Kitty: sketching dynamic and interactive illustrations. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 395–405.
- [36] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, Shengdong Zhao, and George Fitzmaurice. 2014. Draco: bringing life to illustrations with kinetic textures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 351–360.
- [37] Rubaiat Habib Kazi, Tovi Grossman, Nobuyuki Umetani, and George Fitzmaurice. 2016. Motion amplifiers: sketching dynamic illustrations using the principles of 2D animation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 4599–4609.
- [38] Kin Chung Kwan and Hongbo Fu. 2019. Mobi3DSketch: 3D Sketching in Mobile AR. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 176.
- [39] James A Landay. 1996. SILK: sketching interfaces like krazy. In *Conference companion on Human factors in computing systems*. 398–399.
- [40] James A Landay and Brad A Myers. 1995. Interactive sketching for the early stages of user interface design. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 43–50.
- [41] Joseph J LaViola Jr and Robert C Zeleznik. 2004. MathPad 2: a system for the creation and exploration of mathematical sketches. *ACM Transactions on Graphics (TOG)* 23, 3 (2004), 432–440.
- [42] David Ledo, Steven Houben, Jo Vermeulen, Nicolai Marquardt, Lora Oehlberg, and Saul Greenberg. 2018. Evaluation strategies for HCI toolkit research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–17.
- [43] Bongshin Lee, Rubaiat Habib Kazi, and Greg Smith. 2013. SketchStory: Telling more engaging stories with data through freeform sketching. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2416–2425.
- [44] Germán Leiva, Jens Emil Grønbæk, Clemens Nylandsted Klokmosæ, Cuong Nguyen, Rubaiat Habib Kazi, and Paul Asente. 2021. Rapido: Prototyping Interactive AR Experiences through Programming by Demonstration. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 626–637.
- [45] Germán Leiva, Cuong Nguyen, Rubaiat Habib Kazi, and Paul Asente. 2020. Pronto: Rapid Augmented Reality Video Prototyping Using Sketches and Enaction. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.
- [46] Jian Liao, Adnan Karim, Shivesh Jadon, Rubaiat Habib Kazi, and Ryo Suzuki. 2022. RealityTalk: Real-Time Speech-Driven Augmented Presentation for AR Live Storytelling. *arXiv preprint arXiv:2208.06350* (2022).
- [47] Jingyuan Liu, Hongbo Fu, and Chiwei-Lan Tai. 2020. Posetwon: Pose-driven tween animation. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 791–804.
- [48] Zhicheng Liu, John Thompson, Alan Wilson, Mira Dontcheva, James Delorey, Sam Grigg, Bernard Kerr, and John Stasko. 2018. Data Illustrator: Augmenting vector design tools with lazy data binding for expressive visualization authoring. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 123.
- [49] Jiaju Ma, Li-Yi Wei, and Rubaiat Habib Kazi. 2022. A Layered Authoring Tool for Stylized 3D animations. In *CHI Conference on Human Factors in Computing Systems*. 1–14.
- [50] Bruno Mars. [n.d.]. That’s What I Like Official Music Video. <https://youtu.be/PMivT7MJ4IM>
- [51] Elizabeth D Mynatt, Takeo Igarashi, W Keith Edwards, and Anthony LaMarca. 1999. Flatland: new dimensions in office whiteboards. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 346–353.
- [52] Michael Nebeling and Katy Madir. 2019. 360proto: Making Interactive Virtual Reality & Augmented Reality Prototypes from Paper. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.
- [53] Michael Nebeling, Janet Nebeling, Ao Yu, and Rob Rumble. 2018. Protoar: Rapid physical-digital prototyping of mobile augmented reality applications. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.
- [54] James Paterson. [n.d.]. Norman. <https://experiments.withgoogle.com/norman>
- [55] Ken Perlin, Zhenyi He, and Karl Rosenberg. 2018. Chalktalk: A Visualization and Communication Language—As a Tool in the Domain of Computer Science Education. *arXiv preprint arXiv:1809.07166* (2018).
- [56] Ken Perlin, Zhenyi He, and Fengyuan Zhu. 2018. Chalktalk VR/AR. *International SERIES on Information Systems and Management in Creative eMedia (CreMedia)* 2017/2 (2018), 30–31.
- [57] Shwetha Rajaram and Michael Nebeling. 2022. Paper Trail: An Immersive Authoring System for Augmented Reality Instructional Experiences. In *CHI Conference on Human Factors in Computing Systems*. 1–16.
- [58] Patrick Reipschläger and Raimund Dachselt. 2019. Designar: Immersive 3d-modeling combining augmented reality with interactive displays. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*. 29–41.
- [59] Nazmus Saquib, Faria Huq, and Syed Arefinul Haque. 2022. graphiti: Sketch-based Graph Analytics for Images and Videos. In *CHI Conference on Human Factors in Computing Systems*. 1–15.
- [60] Nazmus Saquib, Rubaiat Habib Kazi, Li-Yi Wei, and Wilmot Li. 2019. Interactive Body-Driven Graphics for Augmented Video Performance. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 622.
- [61] Toby Schachman. 2015. Apparatus: a hybrid graphics editor/programming environment for creating interactive diagrams. *Strange Loop* (2015).
- [62] Jeremy Scott and Randall Davis. 2013. Physink: sketching physical behavior. In *Proceedings of the adjunct publication of the 26th annual ACM symposium on User interface software and technology*. ACM, 9–10.
- [63] L Sousa, R Alves, and JMF Rodrigues. 2016. Augmented reality system to assist inexperienced pool players. *Computational Visual Media* 2, 2 (2016), 183–193.
- [64] Ivan E Sutherland. 1964. Sketchpad: man-machine graphical communication system. *Simulation* 2, 5 (1964), R–3.
- [65] Ryo Suzuki, Jun Kato, Mark D Gross, and Tom Yeh. 2018. Reactile: Programming Swarm User Interfaces through Direct Physical Manipulation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 199.
- [66] Ryo Suzuki, Rubaiat Habib Kazi, Li-Yi Wei, Stephen DiVerdi, Wilmot Li, and Daniel Leithinger. 2020. Realitysketch: Embedding responsive graphics and visualizations in AR through dynamic sketching. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 166–181.
- [67] Thitaree Tanprasert and Dongwook Yoon. 2022. AR Music Visualizers: Application Space and Design Guidelines. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. 1–6.
- [68] Bret Victor. 2012. Inventing on principle. In *Invited talk at the Canadian University Software Engineering Conference (CUSEC)*, Vol. 5.
- [69] Bret Victor. 2012. Stop drawing dead fish. In *ACM SIGGRAPH*.
- [70] Tianyi Wang, Xun Qian, Fengming He, Xiyun Hu, Yuanzhi Cao, and Karthik Ramani. 2021. GesturAR: An Authoring System for Creating Freehand Interactive Augmented Reality Applications. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 552–567.
- [71] Aidan Wolf. 2019. DoodleLens. <https://www.doodlēlens.app/>
- [72] Haijun Xia, Nathalie Henry Riche, Fanny Chevalier, Bruno De Araujo, and Daniel Wigdor. 2018. DataInk: Direct and creative data-oriented drawing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.