Interactive Body-Driven Graphics for Augmented Video Performance

Nazmus Saquib, Rubaiat Habib Kazi, Li-Yi Wei, Wilmot Li

To cite this version:
Nazmus Saquib, Rubaiat Habib Kazi, Li-Yi Wei, Wilmot Li. Interactive Body-Driven Graphics for Augmented Video Performance. CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019, May 2019, Glasgow, United Kingdom. 10.1145/3290605. hal-02005318

HAL Id: hal-02005318
https://hal.archives-ouvertes.fr/hal-02005318
Submitted on 4 Feb 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Interactive Body-Driven Graphics for Augmented Video Performance

Namus Saquib
MIT Media Lab

Rubaiat Habib Kazi
Li-Yi Wei
Adobe Research

Wilmot Li
Adobe Research

Figure 1: Our system is a presentation tool that lets the performer interactively control the graphical elements using a wide range of hand gestures and body postures for a real-time storytelling experience. (a) An umbrella is attached to the presenter’s hand, (b) controlling a custom-made slider to change the humidity chart, (c) directly manipulating a virtual sketch (cross arrow) to change the timeline of a data chart, and (d) pointing and flicking gestures (blue arrows) to move the temperature arrow, which manipulates the sun’s opacity.

ABSTRACT
We present a system that augments live presentation videos with interactive graphics to create a powerful and expressive storytelling environment. Using our system, the presenter interacts with the graphical elements in real-time with gestures and postures, thus leveraging our innate, everyday skills to enhance our communication capabilities with the audience. However, crafting such an interactive and expressive performance typically requires programming, or highly-specialized tools tailored for experts. Our core contribution is a flexible, direct manipulation UI which enables amateurs and experts to craft such presentations beforehand by mapping a variety of body movements to a wide range of graphical manipulations. By simplifying the mapping between gestures, postures, and their corresponding output effects, our UI enables users to craft customized, rich interactions with the graphical elements. Our user study demonstrates the potential usage and unique affordance of this mixed-reality medium for storytelling and presentation across a range of application domains.

CCS CONCEPTS
- Human-centered computing → Mixed / augmented reality; Gestural input; User interface design.

ACM Reference Format:

1 INTRODUCTION
“In the coming age, Computer Graphics will become an integral part of our language” - Ken Perlin [41]

Augmented and mixed reality technologies enable user experiences that leverage virtual elements to alter, enhance, and extend our perception of the real world. One simple but powerful form of augmentation is blending animated graphical elements like illustrations, icons, and text with live action footage of performers (Figure 1). Historically, this technique has been used as a special effect for various types of content, including scientific documentaries, instructional material, and music videos. While augmented graphics are typically added as a post-process after the primary footage has been captured, some examples involve performers manipulating graphics in real-time, as in weather forecasts or more recently, social media apps with video overlays.
The expressiveness of augmented videos stems primarily from the range of possible interactions between human performers and animated graphics. To better understand this interaction space, we have analyzed a diverse set of examples and found that augmented graphics are triggered and controlled by a broad spectrum of different gestures and body poses. For instance, graphics are sometimes "attached" to parts of an actor or the scene to highlight, decorate, and enhance the video footage. One specific example of this technique is adding digital clothing or accessories to a character. In other cases, actors directly drive the appearance or motion of graphics by performing a specific pose or gesture, like sweeping their hand to change the scale of an overlaid data visualization or manipulating a virtual, animated object.

While the richness of the interaction space makes it possible to produce a wide range of augmented effects, many of these effects are hard to achieve in practice. Producing real-time augmented videos, like weather forecasts, usually involves specialized professional tools that require a production team to prepare and trigger graphics manually in response to the performer [21, 40]. Video filters on social media apps offer an easier way to achieve real-time effects, but they support a very constrained set of gestures and graphics that limits the expressiveness and potential applications. An alternative is to add augmented effects via video post-processing, using tools like After Effects. However, this approach is restricted to users with the time and expertise to execute advanced video editing and compositing work. Moreover, post-processing is obviously not suitable for real-time augmented videos.

In light of these challenges, we present an authoring tool that helps novice user produce real-time augmented video presentations with a wide range of gestures, body poses, and graphical effects. Our system provides an offline setup mode where users first map gestural actions to shape and appearance attributes of graphical objects via a direct manipulation interface. Then, in the interactive performance mode, the system interprets user actions in real-time, generates the corresponding animated graphics based on the authored mappings, and overlays the graphics onto the video footage. While our method is designed primarily to support real-time scenarios, the proposed approach also facilitates the creation of traditional videos; in this setting, the live performance mode can be used to generate an initial version that users can refine with additional post-processing if necessary.

Our main high-level contribution is in the design of the direct manipulation mapping interface, which provides a flexible, customizable way to associate hand gestures and body postures with graphical effects. We leverage existing hand gesture taxonomies in HCI to represent common gesture categories [13] as well as static body postures as design elements. Using a relational graph structure [30], the user then maps the variety of gestures and postures to graphical actions — triggering, direct manipulation, indirect parameter tuning, and deformation of the graphical elements. By providing a useful set of predefined and composable primitives, our approach enables users to author a wide range of augmented effects without requiring any explicit programming.

We demonstrate the expressiveness of our system by creating several different styles of augmented video examples, including animated stories, scientific lectures, and explanatory videos. In addition, we have conducted design sessions with a diverse set of users to gain insights about the capabilities and limitations of our tool. We have received positive (albeit preliminary) feedback on the usability, benefits, and potential applications of our approach. The main contributions of this paper include:

- A direct manipulation interface for authoring how input bodies map to output graphical effects.
- An interactive performance interface that applies these mappings in real-time.
- A categorization of gestures and postures based on their different capability and suitability for various mapping scenarios.

2 RELATED WORK

Gestures and HCI

Myron Krueger’s Videoplace [35] is one of the earliest explorations demonstrating a virtual environment that responds to dynamic human gestures for interactions (e.g., painting, pointing, selection). In order to overcome the limitations of WIMP (windows, icons, menus, pointers) interactions, HCI researchers have explored a variety of novel interfaces to leverage the qualities of mid-air gestures for sign-language [44], retrieving and manipulating imaginary objects in 3D [45, 50], and interacting with multi-touch screens or AR (augmented reality) tabletops [16, 32, 46, 48, 49].

In this paper, we consider how to leverage the communicative aspects of gestures to enhance real-time human-to-human communication through dynamic graphics. In this vein, prior systems in HCI and graphics, such as Charade [15], ChalkTalk AR/VR [42], and live multimedia presentation [39], use dynamic hand gestures to control a computer-aided presentation to communicate with the audience. Inspired by this existing work, we propose a direct manipulation interface that enables user to define their own mappings from input actions (gestures and postures) to output graphical effects without programming. Given the idiosyncratic nature of gestures, this flexibility facilitates diverse usage scenarios across many domains. Moreover, our representation and flexible UI accommodates a wide range of input actions (gestures, postures) and output graphical effects, thus enabling an expressive range of interactions with graphical elements.
Interfaces for Dynamic Media and Performance

With the advent of digital technologies, interactive and animated graphics are becoming more popular, prevalent, and a powerful medium for visual art, design, and communication. In general, crafting expressive performance-driven graphical effects requires programming (e.g., openFrameworks [8], processing [9], Unity [10], Flash [7], d3 [6]), highly specialized pre-processing (or rigging) workflows, [12], or timeline-based post-processing tools [4, 5] that require extensive expertise. They are also tailored for application domains (e.g., cartoon characters [21], weather forecasting [40]), with limited interaction capabilities.

In order to make dynamic and animated media accessible, HCI researchers have explored sketch-based and direct manipulation interfaces for animation [23, 27, 28, 31], interface prototyping [36, 38], explanatory illustrations [30, 52], data-storytelling [37], and designing live-performance triggers [47]. Performance based systems have also been explored to map captured human motion into digital characters [24] and arbitrary digital objects (with different topology) [20]. Visual programming tools, such as Scratch [43] and [51], empower users to create their own animations, music, and interactive stories with programmable constructs. Rather than relying on pre-defined models or programming, tools like Kitty [30] and SketchStudio [33] provide an interface where relationships and events can be defined by directly manipulating the underlying relational graph, displayed in the context of the illustration. However, the resulting dynamic artifacts from these tools are designed for pen and touch interactions with limited degrees of freedom. In contrast, our system extends these ideas to produce real-time augmented videos of human performances that leverage the many degrees of freedom of whole body interactions.

Kinect-based Interfaces for Embodied Interaction

With the recent advent of computer vision and real-time pose estimation technologies (Microsoft Kinect, OpenPose [19]), there has been increased interest in exploring how embodied interaction [25] facilitates increased engagement, new learning experiences, and social interactions [29]. In AR mirrors, utilizing the entire body through movement or gesture can support new forms of computer-mediated learning [17, 29] and motor skill improvements [14]. Our work aims to realize these benefits in the context of augmented performance videos for storytelling.

Design Space of Presentation Tools

Since many of the existing systems and techniques discussed above enable users to present ideas with graphics, it is worth examining how our work fits into the overall design space of graphical presentation tools. One way to characterize this space is to consider two related dimensions: 1) what authors must do to prepare the presentation content and 2) what input methods and interactive capabilities are available to perform that content. For example, post-processing tools like Adobe After Effects [4] require authors to prepare all of the content by capturing and editing live action footage of performers and then compositing graphical effects. Since the resulting video is created via an offline process, actors cannot interact with the graphics at all during their performance. Our system is more similar to ChalkTalk [41] and performance-driven animation tools that require some amount of preparation to create graphical assets and map them to allowable interactive behaviors and as a result, support real-time interaction with graphics at performance-time. Table 1 summarizes the design space for various presentation methods based on how users prepare and perform content.

<table>
<thead>
<tr>
<th>Preparation</th>
<th>Performance input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slides</td>
<td>Create and layout slides</td>
</tr>
<tr>
<td>After Effects</td>
<td>Capture and edit footage, composite graphics</td>
</tr>
<tr>
<td>Chalktalk</td>
<td>Create interactive graphics with programming</td>
</tr>
<tr>
<td>Character Animator</td>
<td>Create digital characters, map to voice and facial expressions</td>
</tr>
<tr>
<td>Our system</td>
<td>Create graphics, map to gestures and poses</td>
</tr>
</tbody>
</table>

Table 1: Design space for different performance methods.

3 INTERACTION SPACE: AN INFORMAL ANALYSIS

To better understand the types of whole-body interactions humans perform with virtual graphical elements, we analyzed a set of 19 existing augmented videos. We noted the types of input actions, and the resulting graphical effects. Our set of videos included both live and post-processed examples across a range of application domains (e.g., science documentary [1, 3], commercials [2], music videos, and weather forecasting [11]), where the performer interacts with virtual graphical elements to tell stories.

Observed Gestures, Postures, and Parameterization

Within our analysis, we observed a range of (static and dynamic) input actions with high-degrees of freedom, resulting in a wide and rich set of interactions with the virtual elements. Below we discuss our observations, and present a few representative examples.

Hand Gestures. Consistent with prior literature on gesture taxonomies in HCI [13], within our analysis, we identified
a range of gestures types to interact with the graphical elements. While such categories of gestures are previously defined in HCI literature [13], in this section we report the interaction capabilities afforded by those gestures, and a range of corresponding output graphical effects for storytelling. We refer the readers to Figure 4 in [13] for a nice visual categorization of various gesture types.

**Pantomimic** gestures are used to mimic an interaction with an imaginary, virtual object [13]. We observed several types of pantomimic interaction. First, users often use both hands to manipulate the transformation parameters (translation, rotation, or scale) of the virtual object (e.g., rotating a rigid mug, photo frame). Second, pantomime gestures can also be used to deform the shape of a graphical object (e.g., interacting with a joystick, or an octopus). In such cases, the overall transformation parameters remain the same.

**Iconic** gestures are used to communicate information about objects or entities, such as specific sizes, shapes, and motion paths. We also noted pointing to highlight, and a variety of direct manipulation techniques.

**Semaphoric** gestures are hand movements and postures that convey specific meanings. Semaphoric strokes represent hand flicks which are single, stroke-like movements [13]. We observed that semaphoric strokes are used to manipulate graphical parameters, which are challenging to communicate using direct manipulation or pantomimic gestures, due to proximity, scale, or abstraction. For instance, a science presenter repeatedly flicks one hand to communicate that air is vacuumed out of a jar while a vacuuming animation is displayed. In this case, the semaphoric stroke gesture is used to manipulate a parameter in a more abstract way. Such gestures are also used to point a graphic at a distant location (e.g., a timeline) and manipulate, to counteract the scale and proximity of the graphical element. Since semaphoric gestures are strictly learned, such mappings between the gesture and the parameters should be pre-defined by the user.

**Rigging and Static Body Postures.** Static and dynamic body postures are also used to interact with the graphical elements. Similar to gestures [15, 18, 45], there are a number of advantages of using body postures to interact with virtual graphical elements. A static or dynamic posture can be used to specify both a command and its parameters. For instance, a guitar holding posture would indicate the users intention to trigger the guitar graphics. Further, the posture information (e.g., the position of the hands) also specifies the parameter of the guitar graphics, including the position, orientation, and scale. Often, virtual sketches and graphics (clothes, helmet, other wearables) are rigged and anchored with respect to the human(s) in the video, as well as arbitrary objects in the scene. In such cases, the sketches are deformed dynamically in accordance to the human skeleton.

**Insights and Observations**

This body-driven medium affords a rich interaction space, in terms of input actions (body postures, skeleton, and variety of gestures) and graphical output effects (direct and indirect manipulations, deformation, spatial, visual, temporal, and quantitative parameter tuning).

The types of gestures observed are consistent with gesture taxonomies in HCI. We also found postures and skeletal rigging as a powerful elements to interact with graphics.

Depending on the context and applications, a gesture could be interpreted in multiple ways. For example, the gestures and postures used by a musical band can be very different from an educational setting. Hence, instead of predefining the effects of gestures, we should enable users to map their input actions into output effects.

Despite the potential benefits of this medium, there is a lack of tools that leverage the richness of the interaction space and facilitate flexible mapping between input actions to output effects to a wider range of audience.

**Design Goals.** Based on our observation and analysis, we formulate the following design goals for our system:

1. A direct-manipulation UI that is intuitive to use, and yet comprehensive to achieve diverse mapping of human actions to graphical effects without the complications of programming or traditional video editing.
2. A flexible rigging system that is simple, expressive, and facilitates a wide range of parameterizations to interact with the graphical elements.
3. Providing interactive controls during live performance to address some of the challenges of a gesture based system, such as ambiguity, segmentation, tracking accuracy, and visual feedback.

**4 USER INTERFACE AND INTERACTIONS**

We have designed and implemented a system that enables users to craft augmented, interactive live-streaming presentations. Our system has an offline authoring environment to craft a story, where the presenter prepares the (static and animated) graphical elements, and maps gestures and postures to the parameters of graphical elements. Figure 2 shows our user interface, consisting of a main canvas, a reference skeleton representing the presenter, as well as global and contextual toolbars. In addition to offline authoring, our system also supports a live performance environment that allows the presenter to perform and interact with the graphical elements in real-time.
Figure 2: Our system authoring UI. (a) Reference skeleton, (b) Trigger menu consisting of template postures and iconic gesture, (c) Global toolbar for drawing and system modes, (d) Contextual toolbar for graphical elements, (e) Graphical elements: an imported map and a sketch.

Figure 3: Hardware setup.

Hardware setup
Our current setup consists of a skeleton tracking device (Kinect) facing the user, a computer for authoring and playback, and a wearable clicker for interactive controls during live presentation (Figure 3). An optional large monitor serves as an AR (augmented reality) mirror to facilitate better viewing and feedback for the presenters.

Creating and Importing Graphical Elements
Our system supports a variety of graphical elements to facilitate expressive storytelling: sketches, images, animated GIFs and textures, and 2D scatter plots.

To create a sketch element, the user sketches a few strokes using the pen tool, and clicks the create element tool. This aggregates all the strokes into a single graphical element. The element contextual toolbar allows the user to further configure the graphical element. The user can sketch a translation path [30] to define a parameterized polygon path, which constrains the movement of the graphical element along the path during live presentation mode. The user can also configure the anchor point of the element by direct manipulation. Clicking the animated button in the contextual toolbar turns the sketch element into an animated texture [30]. The user can also directly drag and drop external static images, animated GIFs, and data files (in CSV format for customizable 2D charts) to create graphical elements. Using the selection tool, the user can select, move, edit, and delete a graphical element any time. Some of these effects are shown in Figure 1.

Input Actions: Skeleton, Gestures, and Postures
During the authoring mode, a reference skeleton (Figure 2) represents the performer. In the design of our system, a key challenge is to categorize and represent the high-dimensional input actions into meaningful, concise entities. To this end, we leverage Aigner et al.’s existing gestural taxonomy in HCI [13] to represent the gestural actions - pantomimic, direct manipulation, semaphoric, pointing, and iconic. We also provide a customizable template of static postures to interact with the graphics (Figure 2). These postures can trigger the graphical elements, and set their positions, scales, and orientations. The skeletal joints can also be used drive the position and deformation of graphical elements by anchoring and rigging. Overall, the variety of gestures, postures, and skeletal joints accommodate a range of input actions to interact with the graphical elements.

Output Effects: Parameterization and Deformation
Figure 4 shows the types of parameter manipulations a graphical element can undergo in our system, including visual, spatial, temporal, quantitative parameters, as well as casual relations and freeform deformation. We normalize the range of each parameter value from 0 to 1. As for non-rigid graphical elements, the user can define multiple pins (or constrained handles) within the bounding box. During live presentation, the position of one or more of those pins are driven by the skeletal joints of the presenter for non-rigid deformations. As demonstrated in our results, the range of parameters and flexible riggings facilitates rich interaction possibilities.

Mapping: Input Actions to Output Effects
In this section, we will discuss the interactions that map input actions (gestures, postures, skeletal joints) to output effects (direct and indirect manipulations, deformation, and anchoring). Such mappings are achieved by creating edges, or by associating a gesture or posture directly to a graphical element. Before we dive into individual interaction techniques, we describe the underlying relational graph structure that captures such mappings and causal relationships in the scene.

Relational Graph. Central to all interactions in our system is a graph data structure [30] that treats all graphical elements
Figure 4: Possible parameters for interacting with a graphical object. (a) original graphic, (b) changing opacity (visual parameter), (c) translating along a user defined path, (d) rotation, (e) scaling, (f) non-rigid deformation using control points (pins), (g) speed of animation (temporal), (h) graphical element radial menu, displaying the parameters, (i) skeleton joint radial menu.

(sketches, images, GIFs, animated texture) and the skeleton as nodes. The edges in the graph define the mapping and functional relationship between the source and destination node parameters, thus depicting the coordination and causal relations between them. In addition, edges also capture the rigging mechanism of our tool for deformation, from skeletal joints to graphical element pins. Users can view the graph structure (nodes and edges) by clicking the Graph button in the main toolbar. Figure 5 shows an example.

Creating Edges for Causal Effects. To create a causal relationship between two graphical elements, the user sketches an edge from source to destination node. Similar to Kitty [31], a contextual radial menu and functional relationship widget enable users to specify the source node attribute (Figure 6), destination node attribute, and the corresponding parameterization function (via sketching), where the horizontal and vertical axes represent the driver and driven parameters.

Edges for Direct Manipulation and Pantomimic gestures. Our system supports uni-manual and bi-manual direct manipulation interaction with the virtual graphical elements. We discuss a few direct manipulation techniques using our system. To anchor a graphical element to a skeletal joint (hand, for instance), the user switches to Graph mode, and then sketches an edge from the hand joint to the graphical element pivot. This edge creates a new mapping. During live performance, when the element appears in the scene, the graphical element follows the skeletal joint freely (Figure 5d→e). However, the movement of a graphical element can also be constrained by a parameterized, user-defined translation path (Figure 6). Such constrained manipulations are useful to create custom sliders (e.g., timeline) and other forms of interactions.

As for pantomimic gestures, the user interacts with a single graphical element by grabbing it with both hands. In such cases, during the authoring mode, the user defines multiple pins in the graphical element, and then specifies one edge from each hand joint to the corresponding pin. During live performance, our system computes the optimum rigid transformation parameters (position and rotation) of the element that best matches the hand positions (Figure 7). This setting allows the user to manipulate a rigid graphical element in a way to how pantomimic gestures are used. The transformation parameters of the graphical element can also drive other elements in the scene using causal-and-effect relationships (e.g., by connecting the rotation parameter of the element to a parameter of another element).
Iconic gestures. An iconic gesture specifies the size and position of a graphical element when it appears in the scene (Figure 8). For such mappings, during the authoring mode, when a graphical element is selected, our system displays a trigger menu at the top of the canvas (Figure 2(b)). This menu specifies variety of behaviors that take effect during the triggering of the element. The user then selects iconic gesture as the triggering behavior. Unlike other gesture types, iconic gestures take effect only when the element appears in the scene for the first time. This is why we display this gesture in the trigger menu, as opposed to edges for manipulation. During live presentation, the user performs a freeform gesture, whose position and size sets the position and size of the graphical element. As Aigner et al. reported [13], iconic gestures are spontaneous, and do not rely on commonly known vocabulary. Thus, we do not take the actual shape into account. Only the position and size of the gesture matters for the parameterization.

Pointing. Pointing is used to highlight and select a graphical element. The user can activate the "pointing" option using a check box in the element’s contextual menu, so that during the live performance, when the performer points to that element, it is selected and highlighted. By default, this option is de-activated for the elements.

Edges for semaphoric gestures. Unlike direct manipulation techniques described above, semaphoric gestures in our system are used to make indirect manipulation of graphical parameter using flick gestures (Figure 9). To map a parameter of a graphical element to a semaphoric gesture, the user switches to graph mode, and creates an edge from the hand joint to the element. The user then selects the semaphore attribute from the skeletal joint radial menu, and the desired attribute from the element radial menu. This creates a mapping between the semaphoric gesture to the corresponding attribute. Such gesture interactions are suitable to manipulate abstract parameters, as well as interacting with distant elements.

Edges for rigging and deformation. Our system’s flexible rigging and mapping also supports a variety of interaction capabilities with deformable graphics that can be controlled by the user’s body. In order to create such mapping, the user sets the graphical element as non-rigid (via contextual toolbar), and defines a few pins that serve as constrained handles [28]. The user then maps the pins to skeletal control points by sketching edges between them. Figure 10 demonstrates the process of creating deformable interactions.

Body Posture. Our system allows the user to trigger and parameterize graphical elements using pre-defined body poses. For instance, when the performer does a guitar holding
posture during the presentation, it triggers the virtual lighting in the scene, and places the element between the two hand positions (Figure 11). To associate a graphical element to a pre-defined body posture, the user selects the corresponding posture icon from the trigger menu (Figure 2(b)) associated to the element. During the performance, our system detects the pre-defined postures, and trigger the graphics accordingly. Once the performer moves to a different posture, the graphical element disappears from the scene. As noted by Steins et al. [45], such postures enable users to quickly switch between devices to optimally support the current task. They also exploit users’ knowledge and experience using physical devices, which makes such interaction self-revealing. We provide three postures that were widely observed during our informal study for highlighting or triggering objects. However, depending on the use case and applications, the user can train the system for their own posture templates.

Figure 11: Static pose trigger and subsequent parameterization by non-rigid deformation. (a) Selection of posture trigger (inset, guitar pose), placement of control points on a GIF of a thunder, and rigging with both hands for deformation. (b) The thunder graphic element appears when the guitarist holds the guitar (makes a guitar pose), and (c) it deforms based on where the hands are when the guitarist plays (red arrow added for emphasis).

Story and Scenes
The resulting story consists of a sequence of scenes (or slides). A scene consists of a sequence of graphical elements, along with the underlying graph structure.

Other Features
Spatial binding. Our system enables users to set the spatial binding parameter for each element via a contextual toolbar. This parameter allows the users to define the distance at which the elements will interact with their bodies. This customization for each element is important for a highly interactive, gesture-based system, because the users might not intend for all element to interact at once as they move around or perform a gesture.

Interactions during Live Performance
During live performance, the author presents the interactive story using gestures and postures. However, there are a number of practical challenges in the design of such a body-based system, such as accuracy of gesture recognition systems, the problem of segmentation and chunking [15], and ambiguities. During the performance, the user may want to use gestures and postures freely without impacting the graphical elements, even if they are defined as actions and triggers during the authoring mode. The user should have full control over when they want an object to be triggered or interact with existing graphics. This is an ambiguous setting, and defining such nuanced interactions is complex.

To mitigate such problems, we equip the performer with a consumer-grade wearable presenter (known as clicker) to provide controls and interact with the system. We mapped the buttons of the clicker to various system functionalities. The next button loads the graphical elements sequentially, while double tapping loads the next scene. To perform an iconic gesture, or trigger an element with a body posture, the user can press and hold the lower button. They can also press the upper button to perform a semaphoric gesture. Our system only tracks for gestures and postures when the user holds down these buttons. The user can also freeze a scene by disabling spatial binding for all graphical elements in the scene by double clicking the upper button.

Our system also provides visual cues to aid the performer, such as the next graphical element at the top left corner, and a freeze icon (top right) to indicate interaction freeze.

5 METHOD AND IMPLEMENTATION
Our system is developed using Openframeworks, a C++ framework for graphical applications. We implemented our system on an Alienware laptop with Intel i9-8950 processor, 32 GB RAM, and an NVIDIA GTX 1080 Ti graphics card.

Skeleton, Gesture, and Pose Tracking. We have used Kinect for Windows V2 for our hardware setup, producing a frame rate of about 60 FPS with the other processes, such as rendering elements, gesture and posture classifiers. While OpenPose [19] produced a lower frame-rate (8 FPS) on average. We project the Kinect skeleton to the 2D screen, and use an adaptive naive Bayes classifier (ANBC) [26] for static pose recognition (with skeleton joint angles as features).

Graphical Transformations. For non-rigid deformation of images, textures, and sketches, we use the as-rigid-as-possible mesh deformation algorithm with control points [28]. For
rigid transformation of objects (pantomime gesture), we calculate the optimum rotation and position of a rigid graphical element based on its control points’ average angle and position difference from skeleton joints that are attached to them in our relational graph structure.

6 DESIGN SESSIONS
We conducted informal, qualitative design sessions with a diverse set of users to gain insights about the potential applications, usability, and limitations of our tool. Users were invited to create a presentation of their own using our tool. The sessions lasted between 2 – 3 hours.

Participants
We invited 6 participants (3 females, age range 23 - 35) with diverse backgrounds and professions (Table 2). Such diverse subjects are better suited to evaluate an emerging mixed medium creation tool, which can be used in many settings. Based on their background survey, we have categorized our participants in four categories in increasing expertise: beginner, intermediate, advanced, and expert.

Methodology
Since there is no existing tool that can provide similar live performance authoring in a flexible setting, we did not compare against a baseline case. The study consists of the following three steps.

Training. We introduced the participants with the overall project, and described the functionalities and usages of our system through a guided training session. This gave the participants an opportunity to systematically learn about each component of the system.

Target task. In this step, participants were given a target story to reproduce with our system. A part of the weather report story (Figure 1) was chosen for this, which includes iconic gestures, a few graphical transformations based on direct manipulation, deformation, a combination of pointing and semaphoric gestures, and a variety of graphical elements (static pictures, textures, and a CSV data file). They were also asked to perform in the live playback mode to get a sense of how they can interact with the elements present in a scene.

Creating Story and Performance. We then let each participant author their own story. Clarifications were provided if they asked questions about the system. After the authoring and performance, we asked them to fill out a questionnaire.

7 RESULTS
Participant feedbacks. Our participants responded positively to the unique affordances, usability, and novel interaction capabilities of our system.

P3: “Each function has a clear goal (like graph, draw) and it is easy to combine them to create different effects.”

However, in contrast to P1-3 (artists), the remaining participants (P4-6) were less experienced with graphical tools (animation, video editing). Not surprisingly, some of our system’s concepts and functionalities (rigging, transformation) were completely new to them, requiring more time to familiarize. However, once they were familiarized, their timing for producing the target task was comparable (P4) or faster (P5,6) than the artists. They also felt more comfortable in the live performance mode.

P6: “There are some quirks in the system and it takes a bit of time to get used to it. But the learning curve relatively flat and I was able to pick it up rather quickly.”

The participants also responded positively to the unique live storytelling aspects of this medium with interactive graphics. In terms of traditional presentation tools, P4 commented: “This system gives me much more control over HOW I want to present and WHERE I want to focus the audience’s attention. Storytelling is often about timing and the right level of emphasis, which is much easier to pull off with this than a slide deck.” As for user engagements and applications, P6 commented - “I engage in STEM outreach quite often and I see potential for software like this to engage with the public. This system allows scientists to transform boring plots to interactive activities and brings the potential to educate the public in a more effective way.”

In contrast to existing video editing tools, participants (P2,5) found our system to be easy, flexible, and more accessible, with the added benefit of live-performance and interactivity. P2 also mentioned the potential of such interfaces and interactions in the context of post-processing video editing tools for greater efficiencies.

Target task. All the participants were able to complete the task without any assistance. The time to reproduce the example in the authoring environment varied from 5 to 14 minutes, with an average of 8.3 minutes. Our program encountered several bugs during P1’s session. Several participants P1,2 were confused about the use of semaphoric gesture to manipulate timeline using a graphical element (at a distant location).
Such confusions could be alleviated by better training and visual feedback during live performances.

**Resulting stories in freeform stage.** Participants crafted a range of stories and presentations using our system, ranging from a cooking tutorial to a full research paper presentation (Figure 12). Some participants \( P_{3,6} \) created stories pertaining to their own professional background, while others were more experimental. Before authoring the presentation, all the participants prepared the storyline, and gathered the necessary graphical elements from the web. After the authoring, we recorded the participants live performance presenting the story. Overall, this session took between 1 to 2 hours.

\( P_4 \) prepared a story for his youtube channel, using bi-manual deformation, uni-manual direct manipulation, anchoring, and graphical overlays. \( P_2 \) demonstrated how to cook ramen eggs, with 3 scenes, leveraging direct manipulation, causal relationships, and iconic gestures, demonstrating the steps of activities. \( P_3 \) performed a presentation about how to meditate in 1 minute.

\( P_5 \)’s story about interior planning took the longest time to author (1.5 hours). She explored and tested some interactions before choosing a few for her story. Her performance incorporated a number of interactions, including semaphoric gestures to clean up the window, direct manipulation, and iconic gesture to place paintings. She also leveraged her ankle joint to interact with the graphical elements (moving a sofa). \( P_3 \) produced a story on a social experiment, it took him 30 minutes to author and test his story before performing.

\( P_5 \) had the longest performance among all (13 minutes). He presented an astronomy research paper using our system. The story creation process took an hour overall, and included finding and cleaning a dataset from his computer, and storyboarding to figure out the contents of each slide in our system. He used a pantomimic gesture to rotate a time wheel and control a data chart, and direct manipulation to slide a night sky image to begin and end his story.

Overall, participants used a variety of gestures and parameterization capabilities, indicating that they found those interactions useful in their performances (Figure 13). Based on the study, we believe the main benefit in supporting a diverse set of gestures is that users are free to define mappings that work well with their specific content or style. They can also choose to limit the range of gestures to reduce cognitive load. The distribution of gestures in the study largely aligns with this strategy; overall, participants covered all categories of gestures, but within a presentation, they each used a small set. As for critical gestures, the most popular were pointing and manipulation, but all types were used at least 2 times.

**8 LIMITATIONS AND FUTURE WORK**

Despite the encouraging results, there are also a number of limitations and opportunities for future improvements that warrant discussion. Based on the feedbacks from the participants and our own experiences with the system, we plan to investigate several future works:

In our current prototype, the mappings are defined at authoring time that can be applied interactively in an expressive but predictable way at performance time. A potential future work is to allow interactive mapping of graphical effects during performance similar to ChalkTalk [42].

As pointed out by our participants, a major challenge for live performance is the mental overload of the presenter. Better visualization is required to preview or remind about what they need to perform next, and what the graphics effects look like in real-time. In our current implementation, we provide a large display in front of the presenter in the form of an augmented reality mirror (Figure 3). We believe wearable mixed-reality eyeglasses or headsets, equipped with presenter’s view, can reduce such mental loads. In addition to
live presentation, our interface design and system components can also be recombined for video post processing to reduce manual workload.

Our current design and implementation tracks a single skeleton only. We plan to incorporate multiple users to facilitate collaborative live performance with interactive effects.

The current interaction scope is limited by computer vision and machine learning components, such as accuracy of posture and gesture tracking. In the future, we would like to incorporate additional sensing modalities (e.g., Leap Motion, Project Soli) for more fine-grained hand gesture recognition and interactions. Furthermore, by incorporating scene geometry (e.g. via ARCore/ARKit), we can render the graphical elements in planar surfaces (e.g., walls, floors).

Our system can be extended for multi-modal interaction, such as speech in addition to hand gestures and body postures. Inspired by existing videos such as [22, 34], we intend to add physical objects via object recognition, tracking, and segmentation.

9 CONCLUSIONS

Gestures and postures are an integral part of human to human communications. As mixed and augmented reality are becoming widespread, we explore how such natural, innate human capabilities can augment our live storytelling. We designed and implemented a flexible, intuitive UI that enables users to map a range of input actions to output effects, resulting real-time, rich interaction capabilities with graphical elements. The diverse story and performances by our design session participants demonstrate the potential of this interactive, rich, augmented storytelling medium.

REFERENCES
