Supplemental Material for MagicalHands: Mid-Air Hand Gestures for Animating in VR

Rahul Arora, Rubaiat Habib Kazi, Danny Kaufman, Wilmot Li, and Karan Singh

1 Description of Atomic Operations

We briefly describe the atomic operations we observed in the study. In general, we use the term *manipulation* to refer to the editing of an already existing entity, *specification* to the definition of a numeric parameter, and *description* to the definition of a parameter which cannot generally be expressed as a single number.

Abstract parameter manipulation. Manipulate an abstract parameter which does not have a direct spatial or temporal meaning, or a physical analogue in the scene. For example, using a diegetic slider to change the emission lifetime in the *smoke* scene.

Attaching particle to emitter. For a particle system, specify which object serves as the emitted particle geometry. In our study, this typically meant attaching the spherical bubble geometry (*particle*) to the wand (*emitter*) in the *bubbles* scene.

Bringing up a menu. Participants wanted to bring up diegetic or egocentric menus similar to desktop UIs for complex manipulations; e.g., selecting the emitter in the *smoke* scene might show a UI panel with smoke simulation properties.

Characterizing object as emitter. Used in the *smoke* and *bubbles* scenes to specify that the chimney and the wand were to be used as smoke and particle emitters, respectively.

Coupling. Specify that the timelines of two objects are coupled. Typically used to specify the physical coupling in the *hook and spring* scene.

Decoupling. Specify that the timelines of two objects which were previously coupled, are now decoupled.

Duplication. Duplicate an object or group of objects. An example utilization was duplication of the bubble geometry in the *bouncing ball* scene by participants who approached the task from a keyframing perspective.

Emission curve description. Define a 1-D curve which describes the general path that particles emanating from an emitter should take. Note that this does not mean that the particles have to exactly follow this curve, as the **emission spread** described below can change how the particles stray away from the emission curve.

Emission density description. This refers to how many particles are emitted from a particle emitter in some unit of time. The emission density may be a numeric parameter (first clip of *bubbles*), or may be described more elaborately (waning density of smoke in the *smoke* scene).

Emission frequency description. Typically used in the *bubbles* scene, this refers to how often a particle emitter spews out particles. Similar to the **emission density**, this may or may not be a single number.

Emission lifetime specification. Particle systems specify a lifetime on emitted particles. This was either defined as the amount of time after which an emitted particle disappears, or as a spatial location on the emission curve where particles vanish.

Emission speed description. Speed of the emitted particles as they move along the **emission curve**. Similar to the **density** and **frequency**, the emission speed may either be constant, or a spatiotemporal variation scheme may be defined.

Emission spread description. This describes how far particles can stray away from the **emission curve**. One can think of the emission spread and the emission curve together as a "particle cone. Similar terminology is employed in commercial software such as Unity[1] and Unreal Engine[2].

Group path description. Description of the path taken by a group of objects. Our participants typically used this operation to illustrate the motion of the cracked pieces in the fourth clip of the *sheet* scene.

Group shape manipulation. Changing the shape of a group of objects. One common use was the crumpling of the cracked pieces in the final clip of the *sheet* scene.

Grouping. This action was used to mark a set of geometric objects as a logical group. For example, grouping bubbles in the *bubbles* scene to allow future operations to impact the set as a whole.

[1] https://unity3d.com/
[2] https://www.unrealengine.com/

Local region selection. Selecting a region of an object. For example, in the second and third cips of the *sheet* scene, participants would select the regions in the middle of the sheet to allow for a future **local shape manipulation** operations to depict the vibrations. Please carefully observe the relevant clips in the supplementary video.

Local shape manipulation. Manipulating the shape of a portion of an object, while the rest of it is constrained to maintain its shape. See **local region selection** above for an example.

Motion path description. Defining the path of an object in spacetime. For example, the paths taken by the ball and tornado in the *bouncing ball* and *tornado* scenes, respectively.

Motion path manipulation. Editing a motion path, such as that of the tornado in the *tornado* scene.

Moving a scaling anchor. While scaling an object, the choice of the origin of the coordinate system in which it is scaled is important. Participants mentioned the use of an "anchor" to indicate the position of the origin for scaling operations. A typical use was in the second and third clips of the *hook and spring* scenes since the spring first scales about its top point, and then about its base.

Object rotation. Rotating an object, such as the tornado in the *tornado* scene.

Object scaling. Uniformly scaling an object.

Object selection. Selecting an object so that further operations can be carried out on it.

Object shape manipulation. Changing the shape of a given object. Our participants utilized this operation to accomplish many shape deformation tasks. For example, to squash or stretch the ball geometry in the *bouncing ball* scene, to bend the tornado object in the *tornado* scene, to non-uniformly scale the spring geometry in the *hook and spring* scene, and to change the shape of the sheet in the first clip of the *sheet* scene. Remarkably, many of these gestures implicitly simulated the three-dimensional widgets designed by Conner et al. [1].

Pause/Play. Pause or play the animation clip. While useful for the animator, note that this is purely navigational and does not change the resulting animation.

Posing (moving without recording). Participants tried keyframing object positions, most notably in the *bouncing ball* and the *tornado* scenes. Given the pervasiveness of both keyframed and performed motion paths in our study, we separate these two categories (the other is **motion path description**).

Randomness description. Adding randomness to a motion path. Typical uses included randomness in the paths of the cracked pieces in the final *sheet* clip, bubble vibrations in the last two *bubbles* clips (when participants interpreted this vibration as random motion), and the turbulence-like behaviour in the *smoke* clips.

Record ON/OFF toggling. In performance-based animation, it is crucial to specify when an animator is posing an object vs. when they want to use their hand motion to actually define a motion path. Many a time, participants utilized an explicit gesture to indicate whether they wanted their actions to pose (record OFF) or to actually move an object (record ON). This operation was typically utilized in the *bouncing ball*, *tornado*, and *hook and spring* scenes.

Scrubbing the timeline. This refers to moving along the animation timeline. Note that this operation in itself does not modify the animated scene.

Secondary emission description. While working with particle emitters, some participants had the notion of a carefully crafted "primary" emission, along with "secondary" emissions which were treated more as random events. This was usually employed for the last two clips of the *smoke* scene to define small wisps of smoke emanating from the main particle cone.

Timing manipulation. This refers to operations—such as locally stretching or compressing an object's timeline—which impact only the temporal properties of an animated effect.

Topological change spatial description. All but the first clip of the *sheet* scene involved topological changes in the animated object. Participants used this operation to define the shape of the tears and shatters.

Topological change timing description. Continuing from above, participants also wanted to indicate when the topological change happens. This involved either setting up different "glue strengths" along the edges where the sheet tore to indicate when each portion of the sheet breaks, or directly indicating when each portion breaks apart by scrubbing the timeline and indicating the portions which break at each instant of time.

UI panel manipulation. This operation refers to moving a UI panel (see **bringing up a menu**) in space in order to focus on a particular panel, relocating a diegetic panel w.r.t its parent object, or to hide a panel to allow for a better view of the animated clip.

Vibrational amplitude specification. Many participants looked at the behaviour of the spring in the last two *hook and spring* clips, of the bubbles in the last two *bubbles* clips, and the middle portion of the cloth in the second and third *sheet* clips as repetitive vibrations. They then proceeded to

define the amplitude of the vibration either by example or by performance.

Vibrational frequency specification. Continuing from above, participants would indicate the frequency of vibration either in the same gesture as the amplitude (performance-based) or indicate the frequency on a timeline (keyframe-based).

World rotation. This operation was utilized to rotate the whole scene to look at it from other viewpoints.

World scaling. Similar to world rotation, participants wanted to scale the world to get a better viewpoint, typically to "zoom into" small details or to "zoom out" and get a bird's eye view of the scene.

1.1 Observed Effects to Atomic Operations

Each gestural or non-gestural interaction observed in our study comes in an *action-effect pair*—the participant executes an action which is meant to create an effect in the animation system. Given the variety of tasks in animation, we noticed as many as 70 unique effects in our study. As noted in the main document, many of the observed actions sought to affect numerous objects and parameters in the scene. However, we still observed that we could categorize the desired effects into either *simple* or *compound*. Concretely, if $\mathbf{E}$ is the set of unique effects, an effect $e \in \mathbf{E}$ is called *compound* if and only if it can be expressed as a logical union of other effects in the set. That is, if

$$e = \bigcup_{f \in \mathbf{E} \setminus \{e\}} f.$$

All other effects are called as *simple*. For example, if setting the emission speed of a particle system and setting the emission path (or curve) of that system are both simple effects, then an action implying both the effects is considered to be *compound*.

After limiting ourselves the set of *simple* effects, we noticed that some of the effects were similar in spirit, and could be grouped together. For example, we grouped **bending description** used to bend the tornado and **squashed/stretched shape description** used to deform the bouncing ball into a group called **object shape manipulation**. After finishing this process, we arrived at a set of 39 distinct atomic operations. Note that the atomic operations themselves may still be high-dimensional, but were the most basic operations we observed in our study.

For completeness, we have included the list of *simple* effects in Table 1. Further, all the *simple* effects observed in a single *compound* effect have been marked with a shared superscript. Table 3 in the main document reports the observed frequency of the taxonomic classes for each of the atomic operations.

2 Particle System Implementation

Our particle system implementation builds upon Unity's native implementation, and adds direct control to the particles. The emission curve $\mathbf{c}(s) : [0,1] \to \mathbb{R}^3$ is created by fitting a C^1-continuous piecewise linear Bézier spline to the input points [3]. The particle lifetime $t(s) \in \mathbb{R}^+$ and spread $r(s) \in \mathbb{R}^+$ are stored as piecewise linear functions on the curve. Since, $t(s)$ is monotonic, v and $\mathbf{c}$ can equivalently be thought as functions on t. Particle are generated at a random position on the plane normal to $\mathbf{c}(0)$, given by polar coordinates $(\delta, \theta_0) \in ([0,1] \times [0, 2\pi))$ w.r.t the Frenet frame of the curve at $s = 0$. The emission noise control is realized by modulating the particle's position and orientation with a Perlin noise [2] function, whose scale and frequency are gesturally controlled. Finally, the "spiral velocity" can applied by rotating the particles in the cross-section plane with a speed supplied by the gesture. For a particle born at time t_0, its position at time t is given by

$$\begin{aligned}\mathbf{p}(t) = \mathbf{c}(t) + \delta r(t) \left(\mathbf{N}(t)\cos\left(\theta(t)\right) + \mathbf{B}(t)\sin\left(\theta(t)\right)\right) \\ + \alpha \mathcal{N}(vt).\end{aligned} \tag{1}$$

Here, $\theta(t) = \theta_0 + \omega t$, where ω is the spiral force parameter, and $\mathcal{N}$ is a 2D Perlin noise function modulated by an amplitude parameter α and a frequency (or scale) parameter v. The noise function is applied in the **N**-**B** plane. A similar noise function applied to the particle orientation rotates it about **N** and **B**.

2.1 Noise and Spiral Motion Gestures

Since the user uses the same hand pose for spiral motion and noise gestures, we need to automatically distinguish between the two. To achieve this, we utilize the time-sampled sequence of 3D hand positions $\{\mathbf{x}_i\}$ and accelerations $\{\ddot{\mathbf{x}}_i\}$ from the gesture. We first find the nearest point $\mathbf{c}(t_i)$ on $\mathbf{c}$ for each of the sampled positions $\mathbf{x}_i$. We then transform the acceleration $\ddot{\mathbf{x}}_i$ to the Frenet frame at $\mathbf{c}(t_i)$. In the third step, we perform Singular Value Decomposition (SVD) on the transformed acceleration data sequence. The singular values represent the variation in the data along its principal components. For the noise gesture, we expect the acceleration vectors to be clustered along a single axis, while spiral motion gestures should exhibit significant acceleration in two directions. Thus, if the ratio of the first (largest) and second singular values is larger than a threshold (we use 2.0), the data is inferred as a noise gesture. Otherwise, we execute the spiral force command. For the former, we use Fast-Fourier Transform (FFT) to get the main frequency and its associated amplitude, which directly map to the parameters α and v of our procedural noise. For the latter, the ratio of the average speed and position projected to the **N**-**B** planes is enough to define the angular velocity ω.

Index	*Simple* effect	Atomic operation (after grouping)
1	Abstract parameter manipulation[f]	Abstract parameter manipulation
2	Attaching particle to emitter	Attaching particle to emitter
3	Bringing up a menu	Bringing up a menu
4	Bringing up a control panel	
5	Characterizing object as emitter	Characterizing object as emitter
6	Coupling[b]	Coupling
7	Decoupling	Decoupling
8	Duplication	Duplication
9	Emission curve description[fghijl]	Emission curve description
10	Emission density description	Emission density description
11	Emission frequency description	Emission frequency description
12	Emission lifetime specification[i]	Emission lifetime specification
13	Smoke speed description[hij]	Emission speed description
14	Particle speed description[kl]	
15	Emission spread description[fghjkn]	Emission spread description
16	Group path description[d]	Group path description
17	Group shape manipulation	Group shape manipulation
18	Grouping	Grouping
19	Local region selection[t]	Local region selection
20	Local shape manipulation	Local shape manipulation
21	Motion path description[bopq]	Motion path description
22	Motion path manipulation[rs]	Motion path manipulation
23	Moving a scaling anchor	Moving a scaling anchor
24	Object rotation	Object rotation
25	World scaling	World scaling
26	Object selection	Object selection
27	Bending description	Object shape manipulation
28	Boundary description[a]	
29	Shape manipulation	
30	Squashed/stretched shape description[a]	
31	Pause/Play	Pause/Play
32	Posing (moving without recording)[bo]	Posing (moving without recording)
33	(Degree of) randomness specification[m]	Randomness description
34	Random path description[m]	
35	Smoke disturbance description[h]	
36	Record ON/OFF toggling	Record ON/OFF toggling
37	Scrubbing the timeline	Scrubbing the timeline
38	Secondary emission description[nr]	Secondary emission description
39	Timing manipulation[ps]	Timing manipulation
40	Crack location description[cd]	Topological change spatial description
41	Tear location specification[e]	
42	Cracking time/flow description[cd]	Topological change timing description
43	Tearing time/flow description[e]	
44	Manipulating a UI panel	UI panel manipulation
45	Repositioning a UI panel	
46	Vibrational amplitude specification[qtu]	Vibrational amplitude specification
47	Vibrational frequency specification[qtu]	Vibrational frequency specification
48	World rotation	World rotation
49	World scaling	World scaling

Table 1: All the observed *simple* effects along with the atomic operations they were grouped into.

References

[1] Brookshire D. Conner, Scott S. Snibbe, Kenneth P. Herndon, Daniel C. Robbins, Robert C. Zeleznik, and Andries van Dam. 1992. Three-dimensional Widgets. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics (I3D '92)*. ACM, New York, NY, USA. DOI: http://dx.doi.org/10.1145/147156.147199

[2] Ken Perlin. 2002. Improving Noise. *ACM Trans. Graph.* 21, 3 (July 2002), 681–682. DOI:http://dx.doi.org/10.1145/566654.566636

[3] Philip J. Schneider. 1990. Graphics Gems. Academic Press Professional, Inc., San Diego, CA, USA, Chapter An Algorithm for Automatically Fitting Digitized Curves, 612–626. http://dl.acm.org/citation.cfm?id=90767.90941