TimeTunnel: Integrating Spatial and Temporal Motion Editing for Character Animation in Virtual Reality

Qian Zhou
qian.zhou@autodesk.com
Autodesk Research
Toronto, Ontario, Canada

David Ledo
david.ledo@autodesk.com
Autodesk Research
Toronto, Ontario, Canada

George Fitzmaurice
george.fitzmaurice@autodesk.com
Autodesk Research
Toronto, Ontario, Canada

Fraser Anderson
fraser.anderson@autodesk.com
Autodesk Research
Toronto, Ontario, Canada

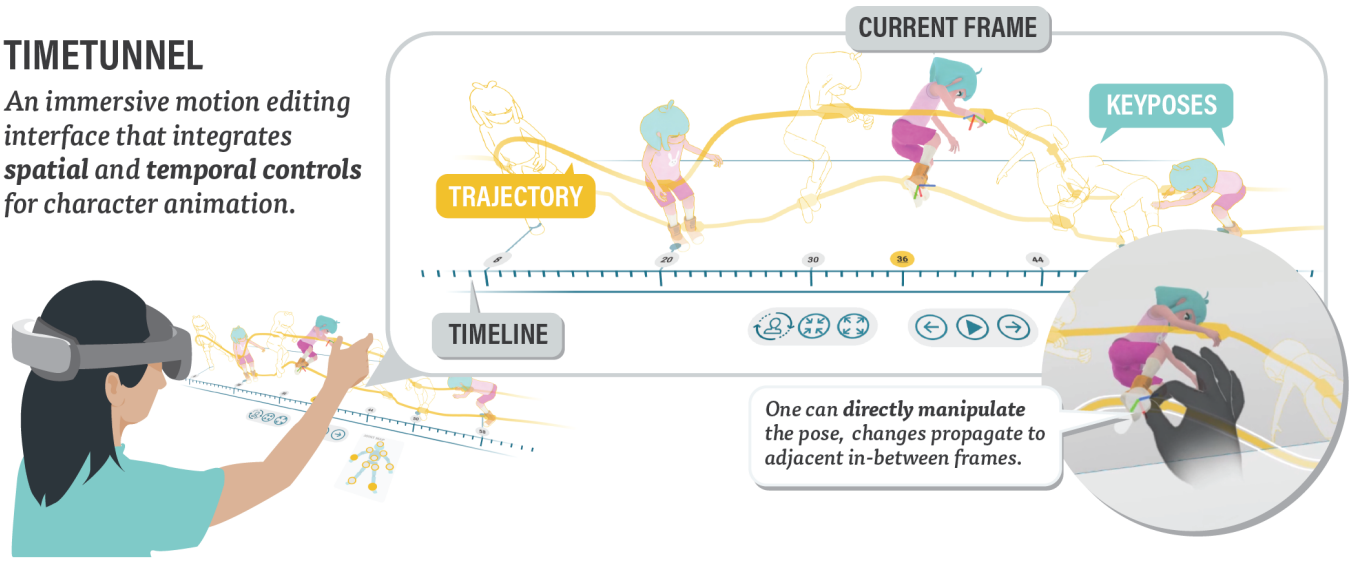


Figure 1: We propose TimeTunnel, an immersive motion editing interface that integrates spatial and temporal controls for character animation. TimeTunnel provides an approachable editing experience by representing motion using KeyPoses and Trajectories in a way compatible with motion-captured data.

ABSTRACT

Editing character motion in Virtual Reality is challenging as it requires working with both spatial and temporal data using controls with multiple degrees-of-freedom. The spatial and temporal controls are separated, making it difficult to adjust poses over time and predict the effects across adjacent frames. To address this challenge, we propose TimeTunnel, an immersive motion editing interface that integrates spatial and temporal control for 3D character animation in VR. TimeTunnel provides an approachable editing experience via KeyPoses and Trajectories. KeyPoses are a set of representative poses automatically computed to concisely depict motion. Trajectories are 3D animation curves that pass through the joints of KeyPoses to represent in-betweens. TimeTunnel integrates spatial and temporal control by superimposing Trajectories and KeyPoses onto a 3D character. We conducted two studies to evaluate TimeTunnel. In our quantitative study, TimeTunnel reduced the amount of time required for editing motion, and saved effort in locating target poses. Our qualitative study with domain experts demonstrated how TimeTunnel is an approachable interface that can simplify motion editing, while still preserving a direct representation of motion.

CHI '24, May 11–16, 2024, Honolulu, HI, USA

ACM ISBN 979-8-4007-0330-0/24/05
https://doi.org/10.1145/3613904.3641927

CCS CONCEPTS

• **Human-centered computing** → **Interaction techniques**; **Interactive systems and tools**; **Virtual reality**.

KEYWORDS

motion editing, immersive animation authoring, 3D interface, motion path, keypose

ACM Reference Format:

Qian Zhou, David Ledo, George Fitzmaurice, and Fraser Anderson. 2024. TimeTunnel: Integrating Spatial and Temporal Motion Editing for Character Animation in Virtual Reality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3613904.3641927

1 INTRODUCTION

Three-dimensional animation is becoming increasingly democratized for non-experts as new technologies and authoring tools become inexpensive and commonplace. Animated avatars are becoming increasingly prevalent through their use by VTubers or in online video-games and virtual worlds [4, 15]. Motion capture techniques, which were once expensive and difficult to set up, have become much more accessible for people to create avatar animations using off-the-shelf cameras [51], monocular videos [55], or directly from the sensors on Virtual Reality headsets [4, 26]. Instead of creating an animation from scratch, content creators can record their motion and edit the captured motion to create a desired clip. However, while capturing motion has become approachable, editing the captured data is still a challenging task.

Beyond the Mocap data, authoring 3D animation in general is difficult for individuals who lack expertise in the field. Professional animation software typically relies on the use of keyframes to define an object's attributes at specific points in time [45]. While this keyframe-based approach is flexible and widely used by professionals, it can be difficult to learn for beginners [40]. Each dimension of an object's position or rotation is represented as an animation curve, with keyframes serving as control points for each curve. Typically, a humanoid model that has a minimum of 15 joints will have at least three animation curves for each joint, representing its x, y, and z position, with each curve containing a different set of control keys. Given the complexity of keyframes and animation curves, desktop animation software such as Maya [2] usually have a complex interface with icons, views, and tools that can be challenging for beginners to navigate. Moreover, curves and keyframes are abstract and require extensive training for novices to understand how changes made to 2D curves will affect the 3D model. While techniques such as inverse kinematics (IK) allow users to directly work on the joints as effectors, the 3D model that contains the spatial information is separated from the timeline, with curves and keyframes containing the temporal information. As a result, the keyframe-based approach divides the spatial and temporal control, making it difficult to coordinate the pose over time.

Virtual Reality (VR) approaches have enabled users to edit 3D poses by directly manipulating the 3D joints with controllers [44, 54] and hand-tracking [37], bringing opportunities for novices to engage in 3D animation authoring. Various immersive animation tools have been developed to support animation authoring. Recent applications such as Flipside [4] also support motion capturing in VR to create animations, making it easy to edit motion immediately after being captured in VR. Although these immersive animation tools show promise, many still leverage keyframes and animation curves, which are inherited from the desktop interface. For example, Quill [8] uses a stop-motion approach that allows users to record 3D poses as keyframes, while VR Blender [33] uses animation curves to create and edit keyframes. These holdovers from the desktop interface take up additional screen real-estate and prevent immersive animation tools from fully benefiting from the 3D interface. Novices still have to work through the difficulties of understanding keyframes and animation curves. Additionally, to edit multiple frames, they need to alternate between the 3D model and timeline editor to apply the changes. More importantly, the separation of the timeline and 3D model requires users to mentally keep track of the temporal context and understand how the changes made in the current frame will affect adjacent frames, in order to create a smooth and desired animation.

To address this challenge, we propose TimeTunnel, an immersive motion editing interface for novices that integrates spatial and temporal control for character animations in VR. Different from existing immersive animation interfaces, TimeTunnel renders Trajectories and KeyPoses to represent the motion in space to provide an approachable editing experience. KeyPoses are a set of representative poses that are carefully computed to depict the motion in a concise view. Trajectories are 3D animation curves that pass through the joints of KeyPoses to help users visualize the motion that occurs in between KeyPoses. TimeTunnel integrates spatial and temporal control by overlaying Trajectories and KeyPoses onto the 3D character. It represents time on an axis with the path of Trajectories indicating the passage of time. As motion-captured data does not contain high-level controls such as keyframes, KeyPoses are automatically extracted from the motion by analyzing Trajectories. Users can go to a frame by scrubbing the 3D character along the time axis and edit the pose by directly manipulating joints with the resultant changes being propagated to adjacent frames.

This work has three main contributions: 1) a motion editing interface using Trajectories and KeyPoses to integrate spatial and temporal editing for character animations in VR along with demonstrations to show its effectiveness. 2) an adapted algorithm to extract KeyPoses from motion-captured data by detecting and combining the extreme points on multiple Trajectories. 3) findings from two user studies with both novice users (n=12) and domain experts (n=5). Results from the novice study showed that the combination of KeyPoses and Trajectories reduced the amount of time required for editing motion and saved the effort to locate target poses compared to a baseline without them. Expert feedback served to critically reflect on TimeTunnel as to how it situates with animation workflows, what strengths could be carried over, as well as the weaknesses of the approach. Through our studies and the demonstrations of use cases, we found that TimeTunnel could simplify motion editing for novices, highlighting the direct representation of motion. We envision TimeTunnel as a complementary technique that can co-exist with other approaches such as immersive motion capturing.

2 RELATED WORK

TimeTunnel is motivated by systems that support animation creation in VR, while the concepts of KeyPose and Trajectory build on prior work that supported motion capturing and the approach of animating with motion paths.

2.1 Immersive Animation Interfaces

As mixed reality technologies have become more accessible, past researchers have explored using AR and VR to improve the productivity of authoring and editing animations. VR animation tools leverage the 6 DoF input of controllers [33, 44, 54], and hand-tracking [12, 29, 37] to make it easier to manipulate 3D poses. Coupled with the Inverse Kinematics (IK), users can directly manipulate the 3D joints to edit poses rather than use the 2D translation and rotation widgets in desktop software. AR-based tools allow users to visualize and manipulate the pose of characters situated in the real environment [53, 58], and commercial VR applications such as AnimVR [1], Mindshow [5], and Quill [8] allow users to create animated cartoons using 3D strokes. Besides the research prototypes and applications, various plugins in animation software such as Maya have been proposed to support motion editing in VR [10, 33].

Similar to desktop animation software, existing immersive tools use the notion of keyframes and animation curves and usually have a timeline editor completely decoupled from the 3D character. For instance, VR Blender uses the animation curves from Blender to edit keyframes in VR [33]. PoseMMR streamed the timeline window from Unity into VR to support keyframe and curve editing [44]. AnimationVR [54] and Quill [8] use a timeline panel to show the motion per joint. Users would manipulate the 3D character for spatial editing and modify the timeline for temporal editing. As a result, the 3D character is still separated from the timeline view so users still have to alternate between the 3D character and the timeline editor to edit the motion. Additionally, the separation of spatial and temporal control makes it difficult for beginners to edit the motion, as they lack in experience knowing how the changes made in the current frame will affect adjacent frames to create a desired motion. This paper aims to fill this separation between the timeline and the character. Building on existing immersive tools that support direct manipulation of joints, this work provides an approach that integrates temporal and spatial motion editing by introducing KeyPoses and Trajectories in VR.

2.2 Capturing Motion and Digital Puppetry

Motion capturing with digital puppetry allows users to create 3D animations using different input methods such their body, hands, or using tracked objects. Body-based digital puppetry tracks the user's body using cameras and maps the motion to the avatar in a 1-to-1 manner [50, 59]. The hand-based approach leverages the dexterity and coordination of human hands and maps the hand joints to the body joints [39] or facial expression [35] of humanoid [28] or non-humanoid avatars [29], and users can generate animations via finger walking [39] or gesturing [52]. In the object-based approach, a physical object can be tracked and act as a puppet to control the avatars [43]. 3D Puppetry [25], SpatialProto [41], and Pronto [36] track physical props and render the corresponding 3D models in a virtual set to perform and record 3D animations. ARAnimator allows users to move a mobile device to directly animate a character moving in a real-work environment [58]. Other work uses modular tangible devices to record 3D motions by mapping physical rotations of tangible input to the joints of the virtual character [22]. Recently there has been more work using the tracking capability of hand-held controllers and the headset to generate 3D motion in VR [11, 26, 49]. Content creators can use applications such as Flipside [4] and Mindshow [5] to generate 3D animations by capturing their motion in VR.

While motion capturing with digital puppetry produces realistic animations, the captured motion can be difficult to modify [23]. Mo-cap systems continuously record the performer's action and typically produce a pose per frame, not just important instants at certain points in time. This method generates large volume of data without a high-level structure that describes the properties of the motion, making it laborious and time-consuming to make a small change across multiple frames [56].

In this work, we aim to support the ease of modification by extracting representative KeyPoses from the captured data. Our work focuses on intuitive editing and is inherently compatible with immersive capturing and digital puppetry interfaces that create and capture motions from hand and body. With the recent immersive approach of capturing motion in VR [4], the proposed interface is promising to provide a seamless workflow that allows users to record and edit motion in VR as suggested in prior work [26].

2.3 Animating with Motion Paths

Animating with motion paths is the process of animating one or more objects moving along a defined path through the scene. The motion path, sometimes also called motion trail [2] or space-time curve [18], displays the movement and orientation and objects and joints over time in the scene and provides more direct spatial-temporal control compared to animation curves. Editing the motion paths is commonly supported in desktop animation tools [40, 46] and applications [2]. Previous work has explored the sketching of motion paths for synthesizing [31] or editing 3D motions [17, 24]. Space-time sketching allowed users to control the motion path and the shape of a 3D character using a single stroke [24]. SketchiMo extended the stroke-based approach and supported natural editing of complex 3D biped models with visualizations that accentuate different aspects of motion [17]. Recent work further improved the motion path approach by optimizing the tangent space to support real-time interpolation control for editing 3D motion [18]. In 2D animations, K-Sketch supported novices to sketch motion paths [19]. Motion Amplifiers further allowed users to inject animation principles when sketching the path [30]. Motion paths have been used as a way to visualize [27] and navigate motion in AR/VR [38, 48] and have been supported in various immersive tools [26, 54]. Recent work also supported recording motion paths to generate animations using VR controllers [16, 21] or hand-tracking [12].

Animating with motion path works well when the character moves from place to place but less if the character stays in one spot with limited root motion, which can be common for motion captured within a limited space. The motion path will overlap with itself and become entangled in space, making it difficult to trace and edit. In this work, we seek to propose an alternative approach to represent the motion using Trajectories which stretch the motion path along an axis to disentangle the overlapping motion. While our work is built in VR, the proposed path representation is conceptually closest to the dynamic path in SketchiMo [17]. We extend beyond the concepts in SketchiMo by complementing the design of Trajectory with KeyPose, which adds explainability to

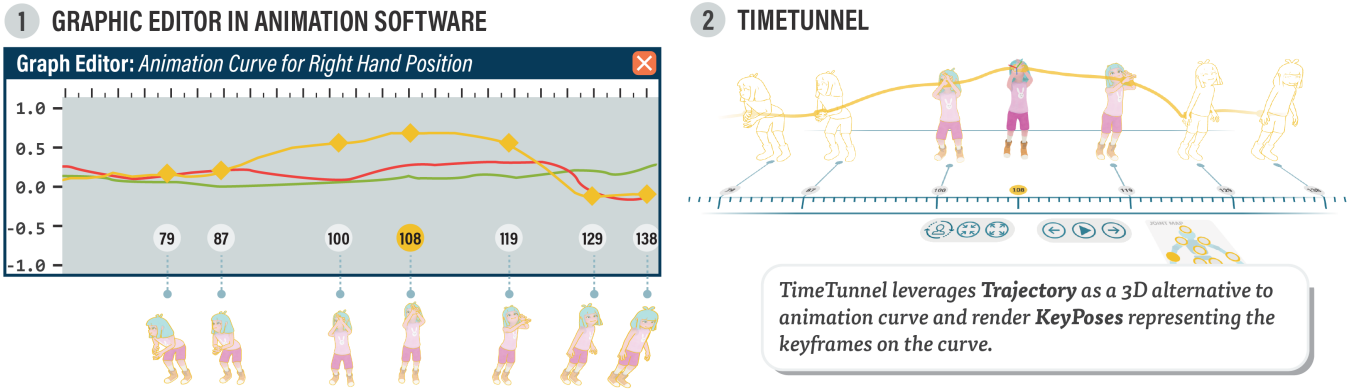


Figure 2: Mapping between (1) traditional animation curve + keyframe representation and (2) Trajectory + KeyPose motion representation. TimeTunnel provides a new way to edit motion by rendering Trajectory as 3D animation curve and showing KeyPoses to represent the keyframes on the curve.

the warped motion path and makes it easy to interpret the motion information. Our study with novices highlights the effectiveness of extending the approach via KeyPoses.

3 TIMETUNNEL

TimeTunnel is a motion editing interface in VR for character animation. The design of TimeTunnel is grounded in three design rationales (described below) informed by prior work on immersive animating systems. While TimeTunnel's interface may be extended to professional animation contexts, it is targeted at, and designed for novices.

R1. Integrating Spatial and Temporal Control: Traditional animation software separates spatial and temporal control [17, 24], requiring animators to switch between a character view for spatial poses and a timeline view for temporal changes in curves and keyframes. This can make it challenging to coordinate body shapes over time and keep track of the editing context. While 3D interfaces and interactions in virtual reality have the potential to unify spatial and temporal control, many existing immersive animation interfaces still separate these two elements [33, 44]. To address this issue, we aim to use motion representations that seamlessly integrate spatial and temporal control.

R2. Providing an Approachable Editing Experience for Character Animation: Human motion data is multivariate with numerous spatial and temporal variations. Editing humanoid animations with keyframes can be challenging, as it involves navigating a complex interface and manipulating multiple control handles in high-dimensional space to achieve the desired change [32]. For beginners, predicting the outcome can be difficult, as keyframe and curve controls are often indirectly applied to the character. We set out to simplify the interface and interactions to make character animation approachable.

R3. Compatible with Motion Captured Data: One notable challenge with motion capture is the editing of the captured data [23, 56]. Mo-cap systems usually record a pose for each frame, rather than just the essential poses at specific time points, resulting in a vast amount of data without a high-level structure that describes the motion's properties. As motion capture technology has become more widely available, we aim to simplify editing of this unstructured data.

3.1 Main Components

Based on the three considerations, we introduce TimeTunnel, an immersive motion editing interface that integrates spatial and temporal control in VR. To provide an approachable editing experience, we combine Trajectory and KeyPose to represent the motion. We use Trajectory as a 3D alternative to animation curve and render KeyPoses representing keyframes on the curve (Figure 2). Unlike prior work that typically separated the animation curves and keyframes from the character [33, 54], Trajectory and KeyPose are overlaid onto the character to integrate spatial and temporal control.

Trajectory is a set of ordered positions of a joint over time. It can be considered as a 3D animation curve that connects one keyframe to another and defines the interpolations in between for a single joint. Different from the motion path that renders the joint's 3D position in the scene, Trajectory maps time over a lateral axis and therefore can be considered as a stretched motion path to disentangle potential overlapping between positions over time. If we use $\vec{x_o}(t)$ to represent the 3D position of a joint at time t, its corresponding position on Trajectory can be computed as:

$$\vec{x_t}(t) = \vec{x_o}(t) + \tau(t - t_c)\vec{v_h}, \tag{1}$$

where t_c is the current frame, τ is a constant representing the degree of stretchiness, and $\vec{v_h}$ is a unit vector of the horizontal axis that aligns with the tunnel direction. Therefore, the position of the current frame t_c on Trajectory is aligned with its original 3D position $\vec{x_o}(t_c)$, while the positions of other frames are exploded out along the direction of $\vec{v_h}$. While Trajectory provides a view that disentangles potential overlapping between positions, it also introduces distortion to the original motion path that artificially stretches the path along $\vec{v_h}$, potentially making it less straightforward to interpret the motion represented by the warped path. Therefore, we introduce KeyPoses to improve its explainability.

KeyPoses are a set of representative poses that are carefully computed to depict the motion in a concise view. Each KeyPose is

a consolidation of a keyframe point on a curve. Rendering multiple KeyPoses with Trajectory presents the motion statically and makes the motion self-explanatory. Additionally, KeyPose enables local control of motion by defining boundaries so that moving one control point only changes a finite range of Trajectory, bounded by adjacent KeyPoses, which is usually considered as an advantage for animation curves [45].

While KeyPoses can be directly generated from keyframes created by animators, motion-captured data intrinsically lacks the support of keyframes that highlight the important instants with representative poses. To make TimeTunnel compatible with motion-captured data, we adapted an algorithm from Action Synopsis [13] to extract KeyPoses from motion-captured data by detecting extreme points on Trajectories. We use the same assumption of Action Synopsis that the high local extrema on the curve may give the best representative poses. The original algorithm of Action Synopsis is designed to find local extremum points on a single curve offline after dimension reduction. We adapted this method to support the extraction of KeyPoses from multiple Trajectories selected by users in real time.

3.2 Extracting KeyPoses From Trajectories

The extraction of KeyPoses is an iterative process to select a sufficient number of representative frames with high local extrema on Trajectories. We use $T_k(t)$ to denote a Trajectory of the joint k ($k = 0, ..., N-1$). We apply a Gaussian smoothing function on $T_k(t)$ to get a smoothed trajectory $\overline{T}_k(t)$. For each joint, we compute the difference between $T_k(t)$ and $\overline{T}_k(t)$ as a function of t by calculating their Euclidean distance. In each iteration, a combined difference for all active joints is computed as a weighted sum of the differences across all Trajectories, where each weight w_k is a ratio of the movement magnitude of the joint k divided by the sum of magnitudes from all joints. By doing so, joints that have greater movement will have a larger impact on the selection of KeyPoses.

$$d(t) = \sum_{k=0}^{N-1} w_k \|T_k(t) - \overline{T}_k(t)\|, \quad w_k = \frac{\|T_k(t)\|}{\sum_{i=0}^{N-1}\|T_i(t)\|} \tag{2}$$

To find a local extreme point, we select the time t_m with the largest difference $d(t_m)$ among all the frames and add it to the list of selected points. The smoothed trajectory is updated by computing a weighted average between $T_k(t)$ and $\overline{T}_k(t)$. The weight $\alpha(t)$ is a function of t that has a higher weight for frames closer to the selected point at t_m within a small constant window δ:

$$\overline{T}_k(t) = \alpha(t)T_k(t) + (1-\alpha(t))\overline{T}_k(t), \quad \alpha(t) = e^{-\frac{(t-t_m)^2}{\delta^2}} \tag{3}$$

When $t = t_m$, the updated smoothed trajectory $\overline{T}_k(t_m)$ is equal to $T_k(t_m)$ to tease out the selected point in the next iteration. The above steps are repeated until the maximum difference $d(t_m)$ is below a threshold or sufficient frames have been selected.

3.3 VR User Interface

To provide an approachable editing experience, we developed a VR user interface that allows users to directly edit the motion using controllers or hand-tracking. The interface is composed of a main character, KeyPoses, Trajectories, and supportive UIs.

The main character is positioned in the middle of the tunnel that corresponds to the current frame (Figure 3). Users can select and drag it to scrub the time axis and navigate through the frames. Users can manipulate its pose by selecting and moving the joints. We use a full-body IK approach to support posing with 10 effectors (hands, feet, upper arms, upper legs, head, and spine). To help users select the joint, a spherical widget with local axes is displayed for each activated joint. Users can move and rotate the widget to control the pose in the current frame. The changes made in the current frame are propagated to adjacent frames bounded by a pair of KeyPoses.

KeyPoses are the 3D models placed side-by-side with the main character at important frames to show representative poses (wireframes in Figure 3). KeyPoses can be selected to specify the range of the animation that will be impacted by any alterations made to the main character's active pose. Positional and rotational changes are propagated using weighted linear and spherical interpolation. Weights are determined using a Gaussian distribution, with the peak centred on the frame at which the change occurred. By default, changes are propagated up to the closest KeyPoses on both sides of the main character (highlighted with transparent texture in Figure 3). To expand the range of motion that will be impacted by changes, users can choose and select different KeyPoses as the starting or ending frame.

KeyPoses can also be used for retiming a segment of motion. Users can select a left KeyPose as the start frame and a right KeyPose as the end frame to specify the range of motion that will be impacted by the modification. To modify the timing of the segment, users can adjust the length of Trajectories between the two KeyPoses by either shrinking or expanding them to speed up or down the motion (Figure 4.3). This retiming process includes an ease-in-out function applied to the selected segment for a seamless speed adjustment.

The difference in physical position between the main character and a KeyPose serves as an indicator for their frame duration. Each KeyPose has a frame number displayed on a button beneath it, and pressing the button will take the user to that specific frame. Additionally, users can add or remove a KeyPose by navigating to the frame and pressing the button of the current frame. KeyPoses are not displayed during animation playback to prevent visual cluttering and are only visible when the animation is paused.

Trajectories are 3D lines that pass through the KeyPoses and the main character, designed to help users visualize the motion that occurs between KeyPoses. These lines also indicate the flow of time, with the horizontal direction serving as a reference of time. To provide continuity, we fit a Catmull–Rom spline [45] from the warped 3D positions computed in Equation 1. Trajectories cannot be modified independently, but they are updated automatically when changes are made to the main character's pose and propagated to adjacent frames. Trajectory color conveys information about the retiming operation (Figure 3). Green is used to signify speeding up, while red is used to indicate slowing down, with the shade of the color reflecting the degree of alteration. Trajectories are visible both during animation playback and when the animation is paused.

Supportive UIs include a Joint Map and navigation buttons. Using the Joint Map, users can activate or deactivate joints and view the corresponding Trajectories by pressing the designated buttons

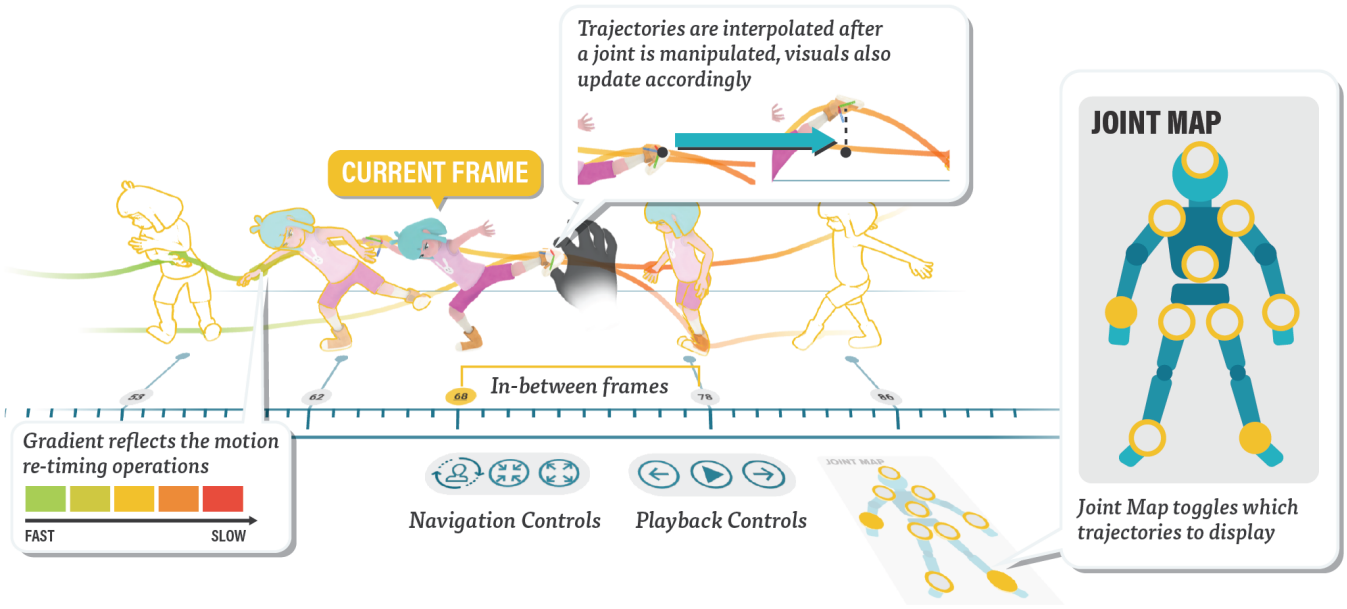


Figure 3: Time-tunnel interface includes the main character, KeyPoses, Trajectories, and supportive UIs. The main character in solid color represents the current frame and can be selected to scrub along the timeline for frame navigation. KeyPoses are a set of representative poses automatically computed to concisely depict motion. Trajectories are 3D animation curves that pass through the joints of KeyPoses to represent in-betweens with the colour representing the retiming operations. Users can directly manipulate the joint with changes propagated to adjacent frames up to the closest KeyPoses (in transparent color) on both sides of the main character.

(Figure 3). The Joint Map determines which KeyPoses are displayed in the TimeTunnel based on the activated joints. This allows users to focus on specific joint movements (such as the feet motion) by activating only the corresponding joints and viewing the KeyPoses that were generated by those movements. When multiple joints are enabled, joints that have greater movement will have a more significant influence on the selection of KeyPoses (Section 3.2).

Alongside the Joint Map, users are provided with standard playback buttons that allow them to play or pause the animation and navigate to the previous or next frame. More buttons are provided for spatial navigation. The rotation button enables users to rotate all KeyPoses and the main character in place, providing them with a view of the movement from various angles. The zoom-in/out buttons allow users to adjust the number of frames that are visible in the TimeTunnel. This enables them to include a broader range of frames or focus on a smaller section of frames.

4 USE CASE EXAMPLES

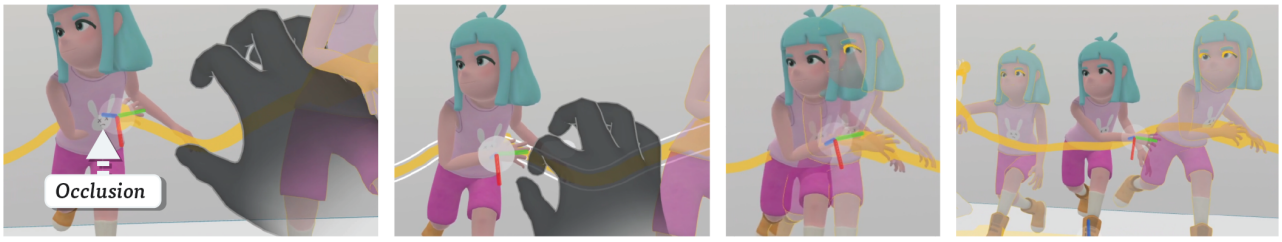
We present three motion sequences edited in TimeTunnel. These three motions represent three different use cases for editing captured motion from the CMU Motion Capture Dataset [3].

Fixing Motion Artefacts. When applying the captured motion to a character, a common issue is body interpenetration, where a character's body parts overlap and intersect with one another. As an example, in a baseball pitching motion, the avatar's hand may overlap with its torso during the follow-through after the ball has been thrown (Figure 4.1). To address this issue, the user can activate the pitching hand in the joint map to display the corresponding Trajectory. When the hand is closest to the torso, the corresponding point on the Trajectory is identified as an extreme point. As a result, the system creates a KeyPose at the point of deepest collision, which the user can locate by scrubbing the Trajectory and pressing its frame button. The user can then simply move the hand away from the body to eliminate the interpenetration. This change is automatically applied to adjacent frames that also have interpenetration, thus resolving the issue in a single edit without requiring the user to address each frame individually.

Supporting Object Contact. TimeTunnel can be used to make sure the body hits specific targets in the scene. For instance, in a soccer kicking motion, the character's foot needs to make contact with a ball (Figure 4.2). To position the foot accurately, the user can activate the foot joint in the joint map to display the corresponding Trajectory. By scrubbing through the Trajectory, the user can locate the point at which the foot is expected to make contact with the ball. The user can then select and move the foot towards the ball to ensure that it hits the target. The change is applied to adjacent frames up to the neighbouring KeyPoses, allowing for a local edit.

Applying Anticipation Principle. While motion-captured data is realistic, it usually lacks the vividness of hand-drawn animations. To stylize a bland jumping motion, the user can apply the anticipation principle (Figure 4.3). The user begins by activating multiple joints and locating the KeyPose right before the jump. The user can then adjust the anticipation pose by pulling back the hands and bending down the body. By selecting the landing KeyPose as the end frame, the changes made to the anticipation pose are applied to the entire jump to ensure a smooth transition. The user can then select the

1 FIXING MOTION ARTEFACTS

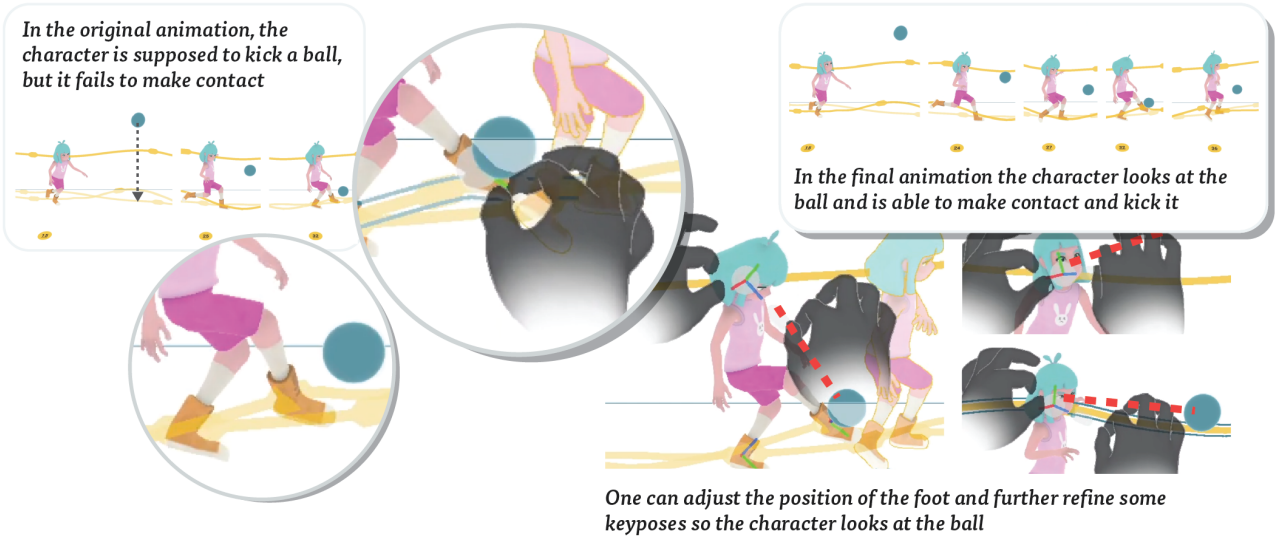


Character arm penetrates body

Adjusting propagates the changes to all in between frames

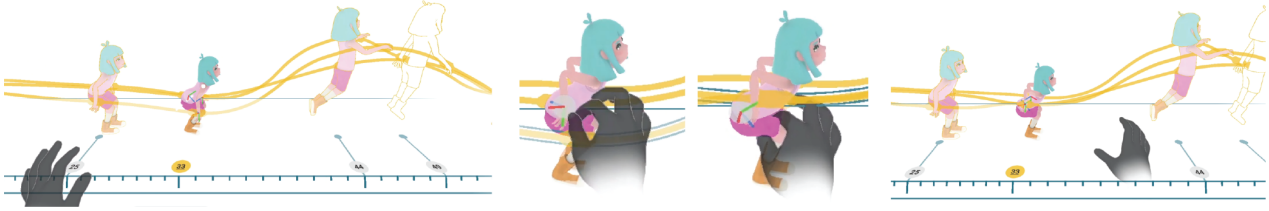
One can navigate in-betweens, visually compare results, adjust where needed, and remove occlusion

2 SUPPORTING OBJECT CONTACT



One can adjust the position of the foot and further refine some keyposes so the character looks at the ball

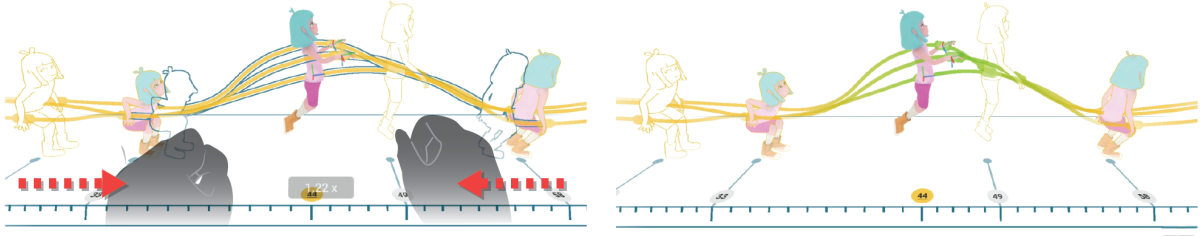
3 APPLYING ANTICIPATION AND EASING PRINCIPLES



Visual inspection shows crouching can be exaggerated to emphasize anticipation for the viewer

Person exaggerates arms and lowers the character's body further

Adjustments can be seen in-context and played back for assessment



Retiming is done via gestures, increasing the speed by bringing keyposes closer together in time

The change in speed with easing can now be visualized via the gradient trajectory, green indicates speeding up

Figure 4: TimeTunnel example use cases in the user's view for (1) fixing motion artefacts, (2) supporting object contact, and (3) applying anticipation and easing principle.

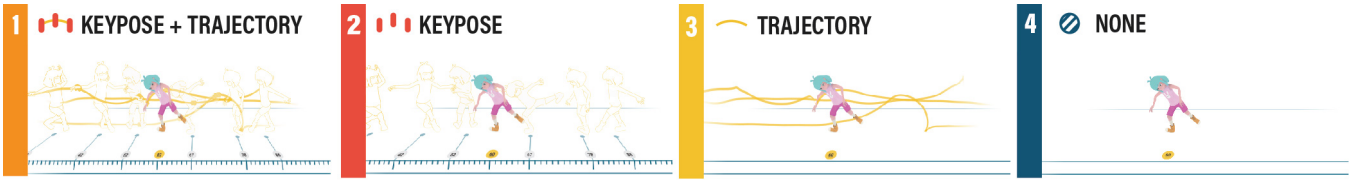


Figure 5: The four conditions in Study 1 include (1) KeyPose + Trajectory, (2) KeyPose only, (3) Trajectory only, and (4) None.

anticipation KeyPose as the start frame and the landing KeyPose as the end frame, and adjust the timing of the jump to be 1.2 times faster. This modification is indicated by the green Trajectories.

5 USER STUDIES

We conducted two user studies to evaluate TimeTunnel. The first study is a quantitative study with two tasks (pose searching and manipulation task) in a controlled environment. The objective of this study is to validate the effectiveness of the key components (Trajectory and KeyPose) in TimeTunnel in supporting motion editing for novice users. We chose to study novices, as they are the target user of the system, and they do not possess preconceptions or expert knowledge that could potentially interfere with the usage of the system such as the conflict that could arise distinguishing KeyPoses from desktop-based keyframes. Therefore the first user study is a study where participants can walk up and use the system under different conditions in a relatively short time [42].

To further define the boundaries of the proposed work, we conducted a second interview-based qualitative study with domain experts in animation. The objective of this study is to understand the strength and weakness of the TimeTunnel interface in a more natural and ecologically valid context. We chose to study animation domain experts because they possess specialized training, established practices, and knowledge of the state-of-the-art in the field. This makes them a strong audience to discuss the strengths and weaknesses of our approach and envision how TimeTunnel might fit into existing practices. Together the two studies help us to form a holistic understanding of the TimeTunnel interface. Ultimately, we envision techniques like TimeTunnel to be complementary to the current suite of tools and co-exist with other authoring approaches.

Apparatus. We implemented TimeTunnel in Unity running on Asus Rog Zephyrus G16 laptop with an Intel i9-13900H CPU and NVIDIA GeForce RTX 4060 Laptop GPU. The laptop is connected to an Oculus Quest Pro via Oculus link cable. 3D interactions of pointing, pinching, tapping, and grabbing gestures were implemented using Oculus Interaction SDK [7]. We used controllers for 3D interactions due to the precision and tracking considerations in the study. Inverse Kinematics control was implemented using RootMotion Final IK [9]. We used a standard 3D character from Mixamo [6] with 20 joints. All motion clips used in the study were from CMU Motion Capture Dataset with a sample rate of 24 fps [3].

5.1 Study 1: Quantitative Study with Novices

We used a pose searching task and manipulation task to evaluate the effectiveness of Trajectory and KeyPose in supporting motion editing for novice users (Figure 6). In the searching task, participants were provided with a target pose and instructed to find the frame that has the pose as fast as possible in an animation clip. In the manipulation task, participants were instructed to edit a jumping animation to avoid the obstacle where the character is hitting an obstacle during the jump.

5.1.1 Experiment Design. We followed a 2x2 within-subject design with two independent variables as KeyPose and Trajectory: each has two levels (with or without keypose or trajectory), resulting in four conditions (Figure 5):

- KeyPose only condition. When changes are made, interpolations are computed within the closest keyposes on both sides of the main character.
- Trajectory only condition. When changes are made, interpolations are computed within a fixed window of frames (± 5 frames).
- KeyPose+Trajectory condition. When changes are made, interpolations are computed within the closest keyposes on both sides of the main character.
- None condition that only has the main character. When changes are made, interpolations are computed within a fixed window of frames (± 5 frames).

The order of the four conditions was counterbalanced using Balanced Latin Squares. In the searching task, we measured the completion time, error count (the number of false submissions), and operation count (the number of playback buttons and slider presses). In the manipulation task, we measured the completion time and frame count (the number of frames that participants edited to produce the final animations).

We did not choose to compare the TimeTunnel interface with desktop-based animation tools for two reasons. First, TimeTunnel is built on existing immersive animation tools that support direct manipulation of joints. The novelty of TimeTunnel is the use of KeyPose and Trajectory to represent the motion in space along with the character, as well as the capability to edit motion segments divided by KeyPoses. Therefore we evaluated KeyPose and Trajectory as individual factors compared to a baseline without them, which is a common representation in existing immersive animation tools [1, 8]. Second, the desktop tools have a lower *expressive match* (i.e., the alignment between the tasks and the interaction) [42] compared to TimeTunnel. TimeTunnel is set up as a walk-up and use system whereas a commercial desktop tool is not, thus potentially leading to an unfair comparison.

5.1.2 Participants and Procedure. We recruited 12 participants (4 females, 8 males) between 21 and 58 years old (averaged 32 years) from within Autodesk. Each was compensated with $75 gift card. All participants were novice users of 3D animations. Most participants had VR experience with a mean self-reported expertise of 2.58 on a 1-5 scale from novice to expert.

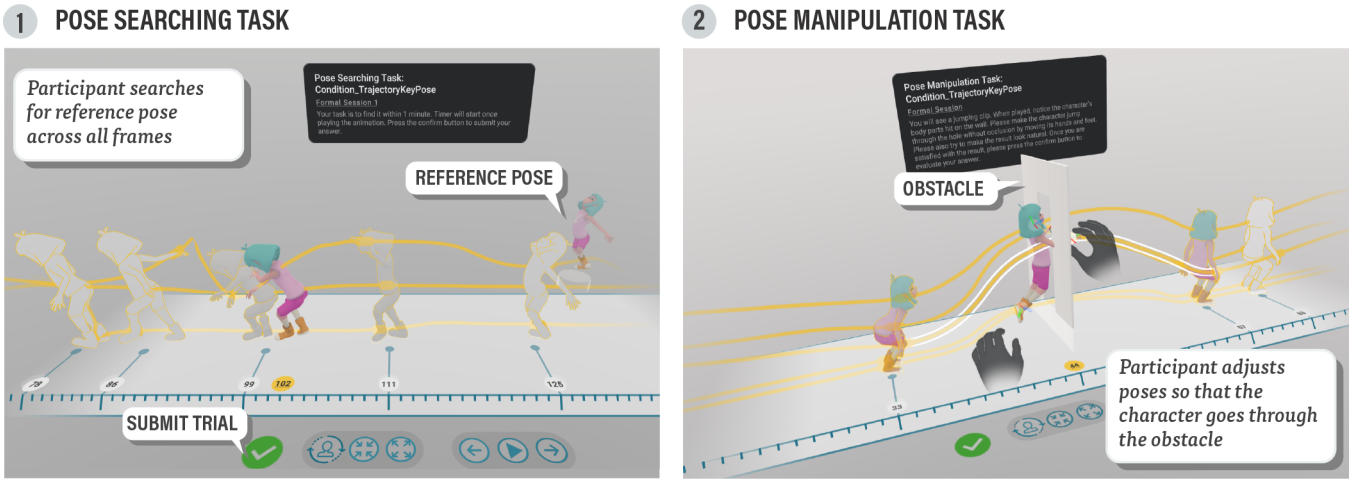


Figure 6: Pose searching task and pose manipulation task in Study 1: (1) In the pose searching task, participants are given a reference pose to find within a motion. (2) In the pose manipulation task, participants are instructed to edit a jumping animation to avoid the obstacle where the character is hitting the wall during the jump.

Participants filled out a consent form and a demographic questionnaire after a verbal explanation of the study. They were guided to setup the equipment by sitting comfortably on a chair, putting on the VR headset, and adjusting its interpupillary distance. Then they were introduced to the TimeTunnel interface with verbal explanations of the UI components. After they were familiarized with the interface, they began the searching task. To avoid memorization from repetition, we chose four similar but different dancing animations for the four conditions from CMU Motion Capture Dataset [3] to avoid memorization from repetition. Each dancing clip has approximately 150 frames (6 seconds) with 15 KeyPoses created by the system. There are three unique target poses determined based on the pose-to-pose principle [34] at significant poses in the clip (one in the beginning, one around the mid-point, and one towards the end). At the beginning of each condition, participants had a practice trial and were instructed to find the target pose as fast as possible. Then they had three formal trials to find the three unique poses. In each trial, they could play, pause, and navigate the time bar to find the target pose. Once they located the frame that matched the pose, they pressed a button to submit the answer. They could submit answers multiple times. They completed the trial when the answer was sufficiently close to the target frame (within ± 2 frames difference).

After completing the searching task, participants conducted the manipulation task following a short break. At the beginning of each condition, participants had a practice session to get familiarized with the joint manipulation and then proceeded to the formal task. The task is adapted from prior work [18] that required motion editing to avoid obstacles. Participants edited a jumping animation (72 frames, 3 seconds, 7 KeyPoses) where the character is jumping forward while its body hits against the wall. The task was to fit the character through a hole in the wall and resolve model penetration by manipulating the poses. Participants could adjust the end effectors (hands, feet, and head) to avoid the obstacle. They could play, pause, and navigate the time bar to edit any frame in the clip. Once they were confident about the result, they pressed a button to submit the animation. They could submit answers multiple times and complete the task when there was no collision between the character and the obstacle. Once they completed the pose-manipulation task, we conducted a short semi-structured interview. They were asked about their preferences for KeyPose and Trajectory in two tasks and how they used KeyPose and Trajectory to complete the tasks. It took about 15 minutes to complete the searching task and 15 minutes to complete the manipulation task, with approximately 50 minutes in total for the study.

5.1.3 Data Analysis. We conducted a repeated-measures ANOVA with significant values reported for $p < .05(*), p < .01(**)$, and $p < .001(***)$ respectively. Effect sizes are reported as partial eta squared (η_p^2). Numbers in brackets indicate mean (M) and standard error (SE) for each respective measurement. When normality assumption is violated, we used the non-parametric Friedman test, followed by Conover post-hoc analysis with Bonferroni correction.

5.1.4 Results.

Pose Searching Task. A repeated measures two-way ANOVA (2 KeyPose × 2 Trajectory) was performed on the completion time and operation count. We found a main effect of KeyPose on the operation count ($F(1, 11) = 17.678$, $p = 0.001$, $\eta_p^2 = 0.616$). The mean operation count for conditions with KeyPose ($M = 7.222$, $SE = 1.014$) was 34.68% lower (**) than without KeyPose ($M = 11.056$, $SE = 1.263$) (Figure 7.2). We did not find a main effect of KeyPose on the completion time ($F(1, 11) = 2.819$, $p = 0.121$), or a main effect of Trajectory on the completion time ($F(1, 11) = 0.025$, $p = 0.854$) or operation count ($F(1, 11) = 0.582$, $p = 0.462$). We did not find interaction effect. Data of the error count did not meet the normality assumption and Friedman's test was performed ($\chi^2 = 12.966$, $p = 0.005$). Post-hoc comparisons with Bonferroni correction showed that the KeyPose condition has a significantly lower error count

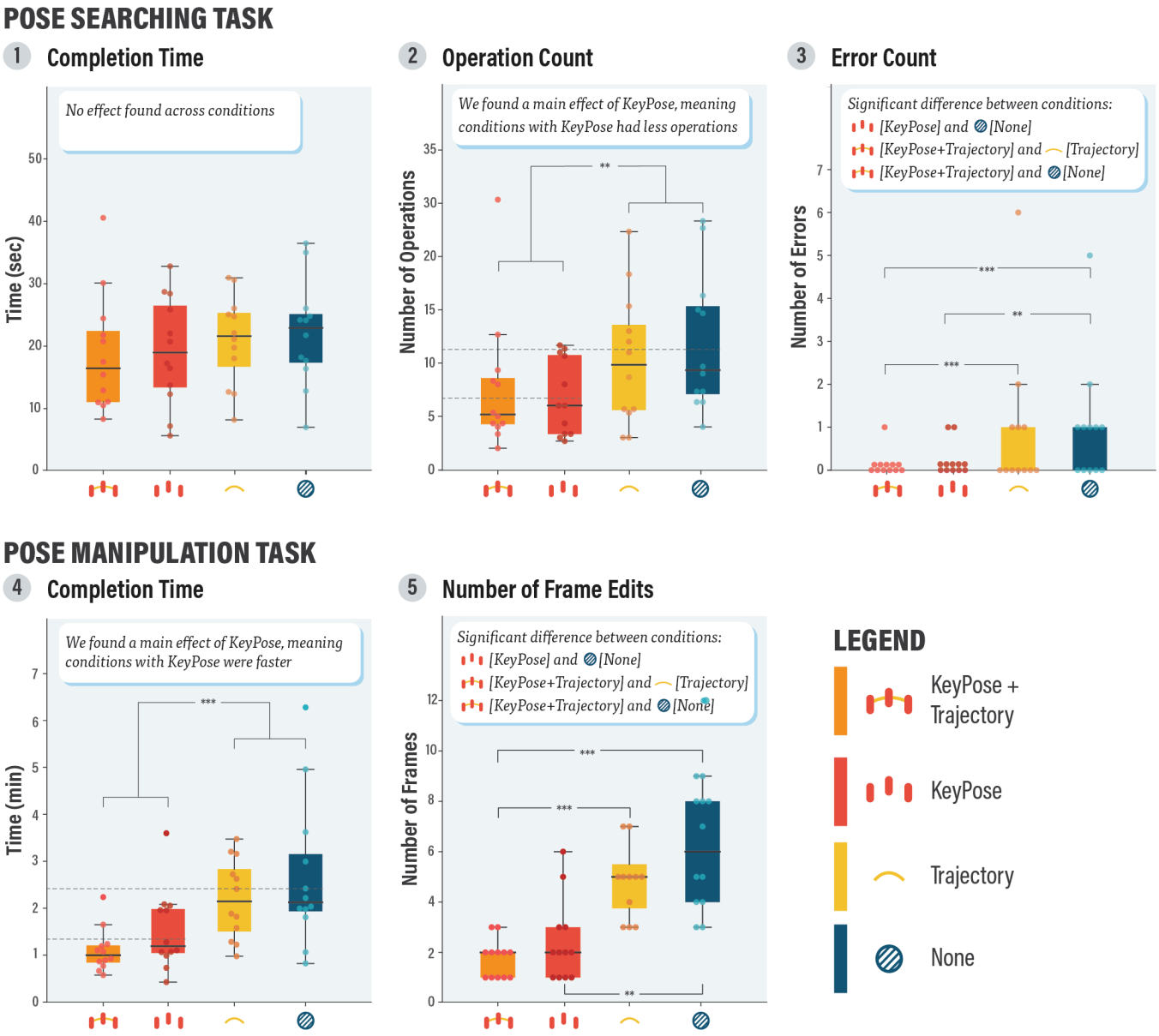


Figure 7: Study 1 results of the pose searching task: (1) no effect found on completion time, (2) the mean operation count (dashed line) for conditions with KeyPose was lower (**) than conditions without KeyPose, and (3) significant differences between conditions on error count. Results of the pose manipulation task: (4) the mean completion time (dashed line) for conditions with KeyPose was lower (***) than conditions without KeyPose, and (5) significant differences between conditions on the number of frame edits.

($t = 3.78, p = 0.004$) than the *None* condition. The KeyPose +Trajectory condition has a significantly lower error count than the Trajectory condition ($t = 6.01, p < .001$) as well as the *None* condition ($t = 4.46, p < .001$) (Figure 7.3).

Pose Manipulation Task. A repeated measures two-way ANOVA (2 KeyPose × 2 Trajectory) was performed on the completion time. We found a main effect of KeyPose on the completion time ($F(1, 11) = 25.802$, $p < .001$, $\eta_p^2 = 0.701$). The mean completion time in minutes for conditions with KeyPose ($M = 1.314$, $SE = 0.120$) was 46.17% lower (***) than without KeyPose ($M = 2.441$, $SE = 0.294$) (Figure 7.4). We did not find a main effect of Trajectory on the completion time ($F(1, 11) = 0.033$, $p = 0.858$). We did not find an interaction effect. Data of the frame count did not meet the normality assumption. Friedman Test was performed on the frame count ($\chi^2 = 23.203$, $p < .001$). Post-hoc comparisons with Bonferroni

correction showed that the KeyPose condition has a significantly lower frame count ($t = 3.66, p = 0.003$) than the *None* condition. The KeyPose +Trajectory condition has a significantly lower frame count than the Trajectory condition ($t = 6.80, p < .001$) as well as the *None* condition ($t = 5.49, p < .001$) (Figure 7.5).

Subjective feedback. All participants found KeyPose useful when searching for a pose. Three participants mentioned using KeyPose as a quick way to filter out information (P7, P10, P11): "*I think the keypose was efficient because like it feels like a menu of a book where you know where is where you could like go to it easily*" (P11). Four participants mentioned KeyPose provided a context of the current frame (P0, P1, P6, P8): "*it's nice cuz then otherwise if you pause you don't really know what is where, what's coming up or what was there. But you can almost determine what the animation will be just from keyposes*" (P1). When manipulating the pose, most participants (10/12) found the KeyPose useful. Three participants further mentioned it was helpful to locate the impactful moments (P1, P3, P10): "*if you want to make the impactful changes, you can go straight to the keypose and make that change versus kind of just changing at some random point*" (P1), which could save them from having to editing more frames: "*I felt when I didn't have the keypose, I have to go like, fix on one, then go out, and go two, three more and then fix them... And when I had the keypose, I just fix one on the keypose and then one in the middle*" (P10).

Most participants (8/12) found Trajectory useful when manipulating the pose, but not as much when searching for a pose (5/12). Participants who found the trajectory useful mentioned the lines helped by "*filling the gap*" (P7) or "*predicting the in-betweens*" (P0, P2, P3, P10, P11), while others might find it distracting as "*too much information*" (P6). Three participants mentioned that they located poses by tracing trajectories (P1, P9, P10): "*trajectory is really helpful to kind of decide like which is the highest point and lowest*" (P1). P1 also mentioned the trajectory could be good for evaluation since "*it helped in knowing if your result was gonna be good or not without having to play it*".

5.1.5 Discussion. Overall, the combination of KeyPose and Trajectory demonstrates effectiveness in both the pose searching task with fewer errors and operations needed, as well as the pose manipulation task requiring less time and fewer frames edited. We attribute this effectiveness primarily to KeyPose while considering Trajectory may play a secondary supportive role.

With KeyPose, participants were able to edit the motion faster with fewer frames to achieve the final results in the pose manipulation task (Figure 7.4). In the pose searching task, KeyPose did not appear to speed up the searching process but it did reduce the number of operations (Figure 7.2) with fewer mistakes (Figure 7.3). In practice, we observed participants often paused the animation and carefully compared each KeyPose with the target pose, while in the conditions without the KeyPoses, they would play and pause the animation when noticing a moment of interest, and exhaustively traverse adjacent frames to locate the target pose. At a given frame, the KeyPose provided more information and therefore saved the effort from having to frequently pause and go back and forth in the timeline. A potential downside of displaying multiple KeyPoses would be visual cluttering with four participants describing the interface as "*a bit overwhelming*" (P2).

Although we did not observe a significant improvement in task performance with Trajectories alone, they may serve a supportive role in motion perception when combined with KeyPoses. Four participants mentioned that Trajectory is most useful with KeyPose (P0, P3, P9, P10): "*I find it harder when it was just the trajectory, but to see it together with the keyframes (keyposes) was a useful guide*" (P3), and "*with the combination, I can go to the key pose directly and then look at the trajectory. It's a mix-and-match*" (P9). When Trajectory is presented alone, participants who knew how to use it would mention it was helpful in "*filling the gap*" (P7), while others might perceive it as "*too much information*"(P6). This could potentially explain Trajectory 's large variation in the operation count compared to KeyPose in the pose searching task (Figure 7.2).

While the quantitative study with novice users showed the promise of the interface, it did not provide a holistic understanding of TimeTunnel such as what might be missing to prevent it from becoming a more effective interface. In practice, we had to hide functionalities unrelated to the task content in order to make the procedure controlled. These functionalities were not evaluated and we have little information as to how to improve them. Therefore, we conducted a second study to evaluate the interface, focusing on qualitative feedback from domain experts.

5.2 Study 2: Qualitative Assessment with Domain Experts

In the second study, we solicited a holistic impression of the TimeTunnel concept and elicited feedback on its current user interface. To do so, we provided five domain experts with the current prototype and conducted a semi-structured interview to explore the benefits and drawbacks of using TimeTunnel for motion editing.

5.2.1 Participants and Procedure. We recruited 5 domain experts (1 female, 3 males, 1 non-binary) between 26 and 53 years old (averaged 37.6 years) from within Autodesk. Domain experts are people who are knowledgeable in animation and have created animations or have worked on animation software. Participants have 2 to 12 years (averaged 4.8 years) of professional experience relevant to animation. All participants had VR experience, with a mean self-reported expertise of 3.4 on a 1-5 scale. Each participant was compensated with $75 gift card.

Participants filled out a consent form and a demographic questionnaire after a verbal explanation of the study. They were guided to setup the equipment by sitting comfortably on a chair, putting on the VR headset, and adjusting its interpupillary distance. Then they were introduced to the TimeTunnel interface with verbal explanations. After they were familiarized with the navigation, they were introduced to motion editing with a baseball pitch motion (128 frames, 5.3 seconds, 12 KeyPoses shown in Figure 4.1). They were encouraged to add details or resolve tracking artefacts in the clip. Once they were satisfied with the results, they were provided with a second jumping clip (72 frames, 3 seconds, 12 KeyPoses shown in Figure 4.3) and encouraged to explore the interface by exaggerating the jump. In total, they spent 20 minutes exploring and working on the animations in the TimeTunnel. Then we conducted a semi-structured interview for 20 minutes. The initial stage of the interview focused on the general impression of TimeTunnel and discussed potential strengths compared to desktop animation

software. Next, participants were asked to critique the concepts of KeyPose and Trajectory, including discussions on what aspects of KeyPose and Trajectory they were excited or skeptical about, how they might use them, and places they could envision TimeTunnel being used. Finally, participants were asked about the limitations of TimeTunnel and suggested improvements to the interface. The study took approximately 50 minutes.

5.2.2 Data Analysis. We collected transcribed audio from the interview and analyzed the results using thematic clustering [14]. The coding process and theme generation was conducted twice by two researchers. Themes were discussed and reviewed collaboratively to create the final coding results.

5.2.3 Results. From the interview data, we identified the **directness** and **simplicity** as the strengths, the **precision** and **visual-cluttering** as the weakness for the future improvement suggested by domain experts.

Directness: All participants mentioned the directness of the interface as a strength compared to desktop animation software. E1 and E3 both mentioned the abstraction of keyframes and curves as a problem in desktop animation software: "*With Maya and desktop animation packages where you have these abstracted editor views, the relationship is harder to understand*" (E1), and "*it would be more difficult to see how it is affecting the animation. Like that's something that I feel is a strength here where you can see you're more directly manipulating the animation rather than working through the abstract of the keyframes and animation curves*" (E3).

There are three aspects that participants expressed regarding the directness of motion representation: the direct manipulation of joints (E0, E2, E3, E4), the directly overlaid Trajectory (E0, E1, E2, E4), and the generated KeyPose (E0, E2, E3, E4). Being able to directly manipulate the joints makes it natural to refine the pose: "*the character immediately felt like a puppet that I could interact with... So it felt quite natural*" (E4), and "*I found that this animation technique was more intuitive in terms of grabbing and adjusting things compared to having to do it on an IK or an FK rig in a 2D screen. From there, I would have to click on something, adjust it, change the gimbal rotation. I'm using shortcuts with my left hand. But in this tool, I was able to just grab it and say, okay, move the foot over here*" (E2).

Domain experts found the Trajectory feature to be more valuable than novices did. E1 commented Trajectory as "*the animation curve overlaid onto the 3D space*". They explained that it provided a direct connection between the movement curves and the character, unlike traditional desktop editing software where the curves are in an abstract form: "*So in desktop editing software, your curve editor is in the abstraction and then you can still manipulate the characters in the viewport of course, but you don't have the visuals of the lines in the view*" (E1). The visual representation of the Trajectories helped participants understand the motion and where specific joints or poses would locate: "*I found that the trajectory paths showing the exact movement was very helpful and it made it easier to understand where a specific joint or pose was going to end up*" (E2). However, this sense of directness was sometimes lost when the character and KeyPoses were rotated to a different direction, making the Trajectory difficult to understand. E2 expressed: "*I found that when I rotated the model, the trajectory became not understandable, but it was totally understandable from a side view*". E2 also suggested rotating the entire tunnel rather than just the KeyPoses: "*I would rather have models behind the one that I'm editing in sequence so that I can move the entire timeline.*".

KeyPose was found to be intuitive to represent the keyframes that can be otherwise dense or abstract to modify: "*I think that's a very good use case for this where you can abstract a lot of those very dense keyframes and just manipulate the broader animation*" (E3). E0 expressed similarity of KeyPose to desktop keyframes which makes it straightforward to edit them: "*So I feel like it's kind of transferred my knowledge in a way where if I want to edit the jump, I should have this keypose and then that keypose*" (E0).

Simplicity: Another strength of TimeTunnel that all participants mentioned is the simplicity of the interface compared to desktop animation software. E4 expressed the complexity of professional animation interface and pointed out the importance of simplicity: "*I've rarely interacted with fully IK-ed characters like that where it's so simplified that you end up with like 12 controllers that are pushing and pulling. More often, You have 40, 50 things in your interface that you're kind of navigating between*"(E4). Participants expressed that this simplicity would make TimeTunnel novice-friendly (E1, E2, E4). For example, E1 expressed: "*I think it would be really fun for a kid or a new animation user like this is, it definitely felt very approachable...when I show them Maya, it's just like there's no interest. It's overwhelming. But something like this would be I think, very appealing with just the limited set of controls and it's very easy to see what's happening*". In spite of the simplicity, E2 mentioned animators may still use it for experienced animations: "*I found that it was easy to work with as if it was a beginner's program, but gave freedom to put in more experienced animation*".

Precision: All participants expressed challenges with precise control of fine movements and identified three potential causes, including 3D mid-air input, a lack of granular controls, and device resolution. E0 expressed that: "*I had a hard time fine-tuning the leg and then the position of the feet*", and further expressed the need for precise control: "*Sometimes it's easier to input that number if you already know what it should be rather than tweaking it to be that number...like I just want like to rotate at like 45 degrees*". Similarly, E1, E2 and E4 expressed the need for more granular controls. For example, E1 expressed: "*it would need to have a lot more flexibility about where you can control, how you can control it*". Finally, E4 noted the challenge of supporting precise control with hardware limitation: "*Precision is probably one of the limits, just the precision of the controllers and the headset and the resolution and just all of those components together*".

Visual clutter: With the Joint Map, participants were able to choose which Trajectories to display in the tunnel. We noticed all participants turned on at least three trajectories to edit the motion. Consequently, the visual space within the tunnel can become cluttered with multiple Trajectories. For example, E2 expressed "*The more bones I had enabled in the joint map, the harder it became to tell which trajectory connected to what*". E1 turned off all trajectories to play and validate the animation itself, suggesting Trajectories may not need to be shown when playing the animation. To address the visual cluttering, E2 suggested visually distinguishing

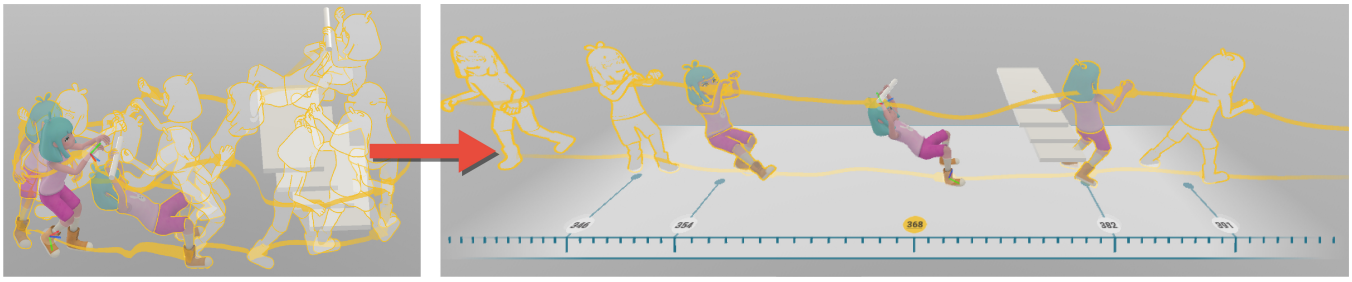


Figure 8: (Left) Detected KeyPoses from the motion paths for a parkour motion (495 frames, 21 seconds) where the character swings from a bar, runs towards the stairs, climbs the stairs, and jumps from the top. While it is possible to directly edit the motion in the scene, (Right) one might zoom in on a selected segment to edit it for a clear view.

the primary trajectory that the user is working on from the rest. E3 and E4 suggested having a sub-tunnel from the main tunnel to reduce irrelevant information: "*Maybe there would be some way to isolate certain parts of the animation and only display those along the time-tunnel*" (E3).

6 DISCUSSION

Through the observation of both novice users and domain experts utilizing TimeTunnel, we revisit our design rationales, discuss the generalizability of our approach, and incorporate the valuable lessons learned for future improvement.

6.1 Revisiting Design Rationales

We revisit our design rationales presented in Section 3.1 and reflect on the degree to which they have been achieved. We also discuss the opportunities and challenges for designing future editing tools.

R1. Integrating Spatial and Temporal Control: TimeTunnel introduces a new approach to integrate the spatial and temporal control by turning keyframes into KeyPoses and superimposing Trajectories onto the character. In Study 1, the combination of KeyPoses and Trajectories demonstrates effectiveness in the two tasks with fewer errors and operations required, or with less time and fewer frames edited. In Study 2, four out of five domain experts expressed the favored use of superimposed Trajectories, as it provided "*a more direct connection between the curves and the character*" (E1), while also finding KeyPoses to be an intuitive method for representing keyframes that may otherwise be difficult to modify. A major challenge encountered during the integration of space and time was visual cluttering, but results from Study 2 suggest these can be mitigated by hiding them during animation playback or visually distinguishing the peripheral and in-focus Trajectories that users are currently working on by adjusting their transparency.

R2. Providing an Approachable Editing Experience for Character Animation: Using KeyPose and Trajectory, all novice users were able to edit the jumping animation to avoid obstacles. In Study 2, we identified simplicity as the strength of TimeTunnel. Compared to desktop animation software, which usually has a complex interface overloaded with UIs, TimeTunnel has a simple interface which can be "*very appealing with just the limited set of controls and it's very easy to see what's happening*" (E1), making it approachable for novice users. This is validated by most domain experts (4/5), highlighting the potential casual usage by content creators, amateur animators, or practitioners who create content consumed in VR to save the time and effort of switching between desktop and VR environments. Moreover, TimeTunnel is also approachable for animators. As KeyPoses and Trajectories are developed to represent keyframes and animation curves, the knowledge and experience that animators possess regarding existing concepts are transferable to TimeTunnel. TimeTunnel employs 3D interactions to support intuitive manipulation of 3D joints with controllers or hand-tracking. Although direct manipulation makes editing more approachable, it may lack precision. Study 2 results suggest the need for supporting more precise input devices, such as keyboard and mouse, which are specifically designed for precision and have recently been incorporated into VR [60].

R3. Supporting Ease of Editing Captured Data: TimeTunnel extracts KeyPoses from Trajectories to make it easy to edit motion-captured data. The generated KeyPoses saved effort in finding target poses and made the editing faster in the two tasks of Study 1. In Study 2, E2 commented that the KeyPoses looked natural and expressed surprise when informed that they were generated from captured data: "*It was very natural. To hear that it was captured from motion capture and then keyposes were created out of it. I wouldn't have ever noticed*". For the motion sequences used in the paper, the algorithm generates one KeyPose for every 12 frames on average to represent the motion. The visualization of the generated KeyPoses for these eight clips can be found in the supplementary materials.

6.2 Generalizability

Beyond full-body motions, TimeTunnel can be a versatile tool for other types of animations, including facial animations and finger movements. However, as the complexity of the character increases and the number of control points and curves from finger joints and face muscles grows, the potential issue of visual cluttering arises. To help users deal with this complexity, it is important to provide an efficient way to hide and show the required elements easily. In desktop animation software, this is typically achieved using hotkeys. In an immersive environment, E1 suggested using gestures to quickly enable tools: "*With a hotkey, you can turn on a knife tool. And then if I slice the lines they just disappear...the same*

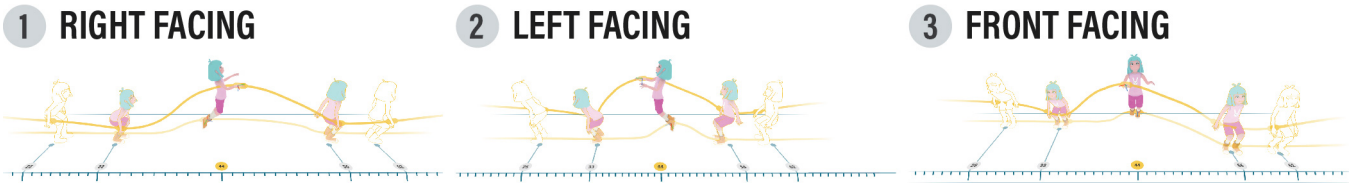


Figure 9: Effect of multiplexing time and space in the time axis when rotating the main character and KeyPoses for a forward jumping animation. Trajectories are mostly comprehensible in the (1) right facing view, when the character jumps along the time axis. Trajectories are compressed in the (2) left facing view, when the character jumps against time. Trajectories become difficult to comprehend in the (3) front facing view, when the jumping direction is perpendicular to the time axis.

tool with a different kind of gesture on a different object would do different things". In addition to enabling and disabling elements quickly, E4 suggested isolating visual elements based on the body part. For example, the user could isolate just the head of a character and then manipulate that facial expression in a sub-tunnel view. This allows the user to focus on specific elements and maintain a clear view of the animation as a whole.

This paper uses example motions from the CMU Motion Capture Dataset [3], which have limited root motion due to the capturing volume. For motion sequences with a large root motion path within the scene, such as running or parkour animation, it is possible to apply the same technique to detect KeyPoses from the motion paths in the scene. Essentially Trajectories are stretched motion paths with the degree of stretchiness defined by τ in Equation 1. Based on that, it would be possible to change between a motion path and a Trajectory by adjusting τ. When τ is zero, the system shows the motion path in the scene. As the user increases τ, the motion path gradually explodes out and becomes Trajectory to reveal overlapped KeyPoses. In this way, TimeTunnel can be applied for editing movements in the scene. Figure 8 (Left) shows all the KeyPoses detected from the motion paths for a parkour animation where the character climbs the stair, jumps from the top, swings from a bar, and runs towards the stairs. While it is possible to directly edit the motion in the scene, KeyPoses can be dense which causes visual cluttering. Users can zoom in on a selected segment and increase τ to stretch the motion path and edit the motion (Figure 8 (Right)).

6.3 VR Interface

The current scope of TimeTunnel lies in a VR interface. Having a VR interface addresses the key challenges of desktop-based paradigms by default. First, with the recent immersive capturing applications [4, 26], users can capture their motion in VR. It is promising to provide a seamless workflow that allows users to record and edit motion in VR. Moreover, users in VR can easily move around to quickly change their perspective. Third, using hands in 3D space enables posing manipulations to be more akin to puppetry, with support for implicit switching between different manipulation *quasimodes* [47]. These three VR features make the experience very different compared to using a desktop or tablet, and introduce a trade-off between rapid experimentation afforded by VR versus a desktop's precision that promotes one-dimensional manipulations. Extending TimeTunnel to desktop and tablet interfaces is a promising area of future work with additional gizmos, widgets, or gestures.

While TimeTunnel uses hand-based 3D interactions such as grabbing as a common way to interact with 3D content, it is possible to extend TimeTunnel with continuous input in existing immersive animation tools. For instance, users can quickly draw a path mid-air using VR controllers [16], depicting a desired trajectory through a continuous stroke gesture instead of manipulating joints.

Ultimately, techniques like TimeTunnel look to complement existing approaches and provide flexibility - different ways to solve problems that are better suited for some tasks over others. For instance, E4 suggested using TimeTunnel in the blocking stage to rough out the motions and setup the overall animation to ensure that the animation is on the right track before moving on to more detailed work with desktop-based precise tools.

6.4 Limitations and Future Work

Despite the promise of TimeTunnel and the associated KeyPose and Trajectory, there are some limitations of the current design. Addressing these issues could enhance the interface's usability and overall experience.

Multiplexing time and space: TimeTunnel integrates spatial and temporal control by representing time on the lateral axis. This means that time and space are multiplexed along the same axis, which could potentially cause confusion, especially when perceiving Trajectories. E2 expressed the confusion when they rotated the main character, as the trajectory became difficult to comprehend. By default, we set the primary moving direction of the motion to align with the time axis. When aligned, there is a distinction between moving against time (180°) and moving along with time (0°). Moving against time will result in compressed Trajectories, while moving along with time will expand them (Figure 9). The extent of compression and extension depends on the scale of stretching τ in Equation 1. In practice, we found that moving along with time provides the best view of the Trajectories and KeyPoses which would be mostly comprehensible for users. Future studies are required to investigate the alignment between the moving direction and the time axis.

Rotations on Trajectory: The current design of Trajectory only shows positions of joints along the time. The orientations of a joint are not visible from Trajectory, which can occasionally cause confusion for users regarding how the orientations are interpolated. While prior work has proposed ways to incorporate orientation on 3D trajectories [57], it is also important to ensure that the interface is not cluttered, as there are already a number of visual

Figure 10: Users using different virtual characters to create the animation of (Left) swimming and (Right) dancing.

elements within the tunnel. Future work is required to explore ways to visualize orientations without causing visual cluttering.

Tunnel shape: We used a constant unit vector ($\vec{v_h}$ in Equation 1) to stretch the Trajectory, resulting in a standard linear representation of the tunnel. While being straightforward, we observed occasionally participants might have problems reaching out to KeyPoses far from the center. Prior work investigated alternative timeline representation in curved, circular, or spiral shapes for visualizing temporal-spatial data [20]. It would be interesting to compare alternatives using varied unit vectors and find out the optimal shape for the TimeTunnel to optimize the space in front of the user. While TimeTunnel can handle longer clips with zoom in/out, the standard flat representation in linear scale is not ideal for prolonged clips. We encourage future work to explore the scale of displaying time as suggested in [20] such as displaying frames in logarithmic scale.

Creating Animations from Scratch: While it is possible to use TimeTunnel to create animations from scratch, additional functionality will be required. For example, tangent lines are commonly provided on animation curves to edit the in-between frames without changing the position of control points. They are not supported in the current implementation with Trajectories, which require users to edit the KeyPoses in order to change the in-betweens. By providing such tools and features, TimeTunnel can help users to author animations more efficiently. Desktop-based animation software supports creating animations from scratch. Future work is required to compared TimeTunnel with desktop-based animation software both for editing and creating animations.

Going beyond rigged characters. We developed a VR user interface on different rigged characters (Figure 10), which are commonly used in 3D animations. However, the design of Trajectories, the extraction of KeyPoses, and the overall interface are independent of the character configuration and can be applied to other types such as articulated objects or non-rigged characters. For instance, some 3D deformable objects may not have joints, but it will be possible to show the Trajectories for feature vertices such as the tail of a rubber duck, and detect KeyPoses based on the Trajectories. We expect that TimeTunnel could go beyond the rigged characters, which opens up interesting directions for future work.

7 CONCLUSION

To further develop the increasing viability of animating with motion-captured data, we presented TimeTunnel, a motion editing interface that integrates spatial and temporal control for character animation in VR. TimeTunnel introduces a new way to represent and edit motions using KeyPoses and Trajectories. KeyPoses are a set of carefully selected poses that are representative to depict the motion in a concise view. Trajectories are 3D animation curves that pass through the joints of KeyPoses to help visualize the tweening frames. TimeTunnel integrates spatial and temporal control by superimposing Trajectories and KeyPoses onto the 3D character and represents time on an axis. To make the approach compatible with motion-captured data which does not usually contain high-level controls such as keyframes, KeyPoses are automatically extracted from the motion by detecting extreme points on Trajectories. We presented findings from two user studies showing that TimeTunnel reduced the amount of time required for editing motion and saved the effort to locate target poses compared to the condition without Trajectories and KeyPoses. Our results demonstrated the effectiveness of TimeTunnel for motion editing and highlighted the strengths and challenges of this approach.

REFERENCES

[1] 2023. AnimVR on Oculus Rift. https://www.oculus.com/experiences/rift/1741124389277542/. [Online accessed August-2023].
[2] 2023. Autodesk Maya: Create an editable motion trail. https://help.autodesk.com/view/MAYAUL/2023/ENU/?guid=GUID-A4F827EE-61C2-4F0A-9592-E91AA20F5D2C. [Online accessed August-2023].
[3] 2023. CMU Graphics Lab Motion Capture Database. http://mocap.cs.cmu.edu/. [Online accessed August-2023].
[4] 2023. Flipside XR Real-Time Animation and Motion Capture. https://www.flipsidexr.com/. [Online accessed August-2023].
[5] 2023. Mindshow: The Future of CG Animation Software. https://www.mindshow.com/. [Online accessed August-2023].
[6] 2023. Mixamo: Animate 3D characters for games, film, and more. https://www.mixamo.com/. [Online accessed August-2023].
[7] 2023. Oculus Interaction SDK. https://developer.oculus.com/documentation/unity/unity-isdk-interaction-sdk-overview/. [Online accessed August-2023].
[8] 2023. Quill by Smoothstep. https://quill.art/. [Online accessed August-2023].
[9] 2023. RootMotion Final IK. https://assetstore.unity.com/packages/tools/animation/final-ik-14290. [Online accessed August-2023].
[10] 2023. VR-PlugIns for Maya, 3dsMax, Unity, Unreal and more. https://www.marui-plugin.com/. [Online accessed August-2023].
[11] Karan Ahuja, Eyal Ofek, Mar Gonzalez-Franco, Christian Holz, and Andrew D Wilson. 2021. Coolmoves: User motion accentuation in virtual reality. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 2 (2021), 1–23.

[12] Rahul Arora, Rubaiat Habib Kazi, Danny M Kaufman, Wilmot Li, and Karan Singh. 2019. Magicalhands: Mid-air hand gestures for animating in vr. In *Proceedings of the 32nd annual ACM symposium on user interface software and technology*. 463–477.

[13] Jackie Assa, Yaron Caspi, and Daniel Cohen-Or. 2005. Action synopsis: pose selection and illustration. *ACM Transactions on Graphics (TOG)* 24, 3 (2005), 667–676.

[14] Ann Blandford, Dominic Furniss, and Stephann Makri. 2016. *Qualitative HCI research: Going behind the scenes*. Morgan & Claypool Publishers.

[15] Liudmila Bredikhina, Takayuki Kameoka, Shogo Shimbo, and Akihiko Shirai. 2020. Avatar driven VR society trends in Japan. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 497–503.

[16] Ruizhao Chen, Ye Pan, Zhigang Deng, Lili Wang, and Lizhuang Ma. 2023. Double Doodles: Sketching Animation in Immersive Environment With 3+ 6 DOFs Motion Gestures. In *Proceedings of the 31th ACM International Conference on Multimedia*.

[17] Byungkuk Choi, Roger Blanco i Ribera, John P Lewis, Yeongho Seol, Seokpyo Hong, Haegwang Eom, Sunjin Jung, and Junyong Noh. 2016. SketchiMo: sketch-based motion editing for articulated characters. *ACM Transactions on Graphics (ToG)* 35, 4 (2016), 1–12.

[18] Loïc Ciccone, Cengiz Öztireli, and Robert W Sumner. 2019. Tangent-space optimization for interactive animation control. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–10.

[19] Richard C Davis, Brien Colwell, and James A Landay. 2008. K-sketch: a'kinetic'sketch pad for novice animators. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 413–422.

[20] Gwendal Fouché, Ferran Argelaguet Sanz, Emmanuel Faure, and Charles Kervrann. 2022. Timeline Design Space for Immersive Exploration of Time-Varying Spatial 3D Data. In *Proceedings of the 28th ACM Symposium on Virtual Reality Software and Technology*. 1–11.

[21] Maxime Garcia, Rémi Ronfard, and Marie-Paule Cani. 2019. Spatial Motion Doodles: Sketching Animation in VR Using Hand Gestures and Laban Motion Analysis. In *Proceedings of the 12th ACM SIGGRAPH Conference on Motion, Interaction and Games*. 1–10.

[22] Oliver Glauser, Wan-Chun Ma, Daniele Panozzo, Alec Jacobson, Otmar Hilliges, and Olga Sorkine-Hornung. 2016. Rig animation with a tangible and modular input device. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–11.

[23] Michael Gleicher. 1999. Animation from observation: Motion capture and motion editing. *ACM SIGGRAPH Computer Graphics* 33, 4 (1999), 51–54.

[24] Martin Guay, Rémi Ronfard, Michael Gleicher, and Marie-Paule Cani. 2015. Space-time sketching of character animation. *ACM Transactions on Graphics (ToG)* 34, 4 (2015), 1–10.

[25] Robert Held, Ankit Gupta, Brian Curless, and Maneesh Agrawala. 2012. 3D puppetry: a kinect-based interface for 3D animation.. In *UIST*, Vol. 12. 423–434.

[26] Sarah Hofmann, Cem Özdemir, and Sebastian von Mammen. 2023. Record, Review, Edit, Apply: A Motion Data Pipeline for Virtual Reality Development & Design. In *Proceedings of the 18th International Conference on the Foundations of Digital Games*. 1–4.

[27] Sebastian Hubenschmid, Jonathan Wieland, Daniel Immanuel Fink, Andrea Batch, Johannes Zagermann, Niklas Elmqvist, and Harald Reiterer. 2022. Relive: Bridging in-situ and ex-situ visual analytics for analyzing mixed reality user studies. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–20.

[28] Ching-Wen Hung, Ruei-Che Chang, Hong-Sheng Chen, Chung Han Liang, Liwei Chan, and Bing-Yu Chen. 2022. Puppeteer: Manipulating Human Avatar Actions with Intuitive Hand Gestures and Upper-Body Postures. In *Adjunct Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–3.

[29] Yu Jiang, Zhipeng Li, Mufei He, David Lindlbauer, and Yukang Yan. 2023. HandAvatar: Embodying Non-Humanoid Virtual Avatars through Hands. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 309, 17 pages. https://doi.org/10.1145/3544548.3581027

[30] Rubaiat Habib Kazi, Tovi Grossman, Nobuyuki Umetani, and George Fitzmaurice. 2016. SKUID: sketching dynamic drawings using the principles of 2D animation. In *ACM SIGGRAPH 2016 Talks*. 1–1.

[31] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. 2002. Motion graphs. *ACM Transactions on Graphics* 21, 3 (2002), 473–482.

[32] Yuki Koyama and Masataka Goto. 2018. Optimo: Optimization-guided motion editing for keyframe character animation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–12.

[33] Fabrizio Lamberti, Alberto Cannavo, and Paolo Montuschi. 2020. Is immersive virtual reality the ultimate interface for 3D animators? *Computer* 53, 4 (2020), 36–45.

[34] John Lasseter. 1998. Principles of traditional animation applied to 3D computer animation. In *Seminal graphics: pioneering efforts that shaped the field*. 263–272.

[35] Luis LEite and Veronica Orvalho. 2017. Mani-pull-action: Hand-based digital puppetry. *Proceedings of the ACM on Human-Computer Interaction* 1, EICS (2017), 1–16.

[36] 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.

[37] Bingyu Li, Ines Said, Linda Kirova, Maria Blokhina, and Hyo Jeong Kang. 2021. SpArc: A VR Animating Tool at Your Fingertips. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*. 1–3.

[38] Klemen Lilija, Henning Pohl, and Kasper Hornbæk. 2020. Who put that there? temporal navigation of spatial recordings by direct manipulation. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–11.

[39] Noah Lockwood and Karan Singh. 2012. Fingerwalking: motion editing with contact-based hand performance. In *Proceedings of the 11th ACM SIGGRAPH/Eurographics conference on Computer Animation*. 43–52.

[40] Jiaju Ma, Li-Yi Wei, and Rubaiat Habib Kazi. 2022. A layered authoring tool for stylized 3d animations. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–14.

[41] Leon Müller, Ken Pfeuffer, Jan Gugenheimer, Bastian Pfleging, Sarah Prange, and Florian Alt. 2021. Spatialproto: Exploring real-world motion captures for rapid prototyping of interactive mixed reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–13.

[42] Dan R Olsen Jr. 2007. Evaluating user interface systems research. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*. 251–258.

[43] Sageev Oore, Demetri Terzopoulos, and Geoffrey Hinton. 2002. A desktop input device and interface for interactive 3d character animation. In *Graphics Interface*, Vol. 2. 133–140.

[44] Ye Pan and Kenny Mitchell. 2020. Group-Based Expert Walkthroughs: How Immersive Technologies Can Facilitate the Collaborative Authoring of Character Animation. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 188–195.

[45] Rick Parent. 2007. Computer Animation: Algorithms and Techniques.

[46] Yichen Peng, Chunqi Zhao, Haoran Xie, Tsukasa Fukusato, Kazunori Miyata, and Takeo Igarashi. 2023. DualMotion: Global-to-Local Casual Motion Design for Character Animations. *IEICE TRANSACTIONS on Information and Systems* 106, 4 (2023), 459–468.

[47] Jef Raskin. 2000. *The humane interface: new directions for designing interactive systems*. Addison-Wesley Professional.

[48] Patrick Reipschläger, Frederik Brudy, Raimund Dachselt, Justin Matejka, George Fitzmaurice, and Fraser Anderson. 2022. AvatAR: An immersive analysis environment for human motion data combining interactive 3d avatars and trajectories. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–15.

[49] Mose Sakashita, Tatsuya Minagawa, Amy Koike, Ippei Suzuki, Keisuke Kawahara, and Yoichi Ochiai. 2017. You as a puppet: evaluation of telepresence user interface for puppetry. In *Proceedings of the 30th annual ACM symposium on user Interface software and technology*. 217–228.

[50] Yeongho Seol, Carol O'Sullivan, and Jehee Lee. 2013. Creature features: online motion puppetry for non-human characters. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 213–221.

[51] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Michael J Black, and Tao Mei. 2021. Monocular, one-stage, regression of multiple 3d people. In *Proceedings of the IEEE/CVF international conference on computer vision*. 11179–11188.

[52] Amato Tsuji, Keita Ushida, and Qiu Chen. 2018. Real time animation of 3D models with finger plays and hand shadow. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*. 441–444.

[53] Andreia Valente, Augusto Esteves, and Daniel Lopes. 2021. From A-Pose to AR-Pose: Animating Characters in Mobile AR. In *ACM SIGGRAPH 2021 Appy Hour*. 1–2.

[54] Daniel Vogel, Paul Lubos, and Frank Steinicke. 2018. Animationvr-interactive controller-based animating in virtual reality. In *2018 IEEE 1st Workshop on Animation in Virtual and Augmented Environments (ANIVAE)*. IEEE, 1–6.

[55] Cheng Yao Wang, Qian Zhou, George Fitzmaurice, and Fraser Anderson. 2022. VideoPoseVR: Authoring Virtual Reality Character Animations with Online Videos. *Proceedings of the ACM on Human-Computer Interaction* 6, ISS (2022), 448–467.

[56] Xin Wang, Qiudi Chen, Wanliang Wang, et al. 2014. 3D human motion editing and synthesis: A survey. *Computational and Mathematical methods in medicine* 2014 (2014).

[57] Colin Ware. 2019. *Information visualization: perception for design*. Morgan Kaufmann. 283–284 pages.

[58] Hui Ye, Kin Chung Kwan, Wanchao Su, and Hongbo Fu. 2020. ARAnimator: In-situ character animation in mobile AR with user-defined motion gestures. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 83–1.

[59] Yupeng Zhang, Teng Han, Zhimin Ren, Nobuyuki Umetani, Xin Tong, Yang Liu, Takaaki Shiratori, and Xiang Cao. 2013. BodyAvatar: creating freeform 3D avatars using first-person body gestures. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 387–396.

[60] Qian Zhou, George Fitzmaurice, and Fraser Anderson. 2022. In-depth mouse: Integrating desktop mouse into virtual reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–17.