STAG: Smart Tools and Applications in Graphics (2018)
M. Livesu, G. Pintore and A. Signoroni (Editors)

A Virtual Character Posing System based on Reconfigurable Tangible User Interfaces and Immersive Virtual Reality

Alberto Cannavò and Fabrizio Lamberti

Politecnico di Torino, Dipartimento di Automatica e Informatica
Corso Duca degli Abruzzi 24, 10129 Torino, Italy
alberto.cannavo@polito.it, fabrizio.lamberti@polito.it

Abstract

Computer animation and, particularly, virtual character animation, are very time consuming and skill-intensive tasks, which require animators to work with sophisticated user interfaces. Tangible user interfaces (TUIs) already proved to be capable of making character animation more intuitive, and possibly more efficient, by leveraging the affordances provided by physical props that mimic the structure of virtual counterparts. The main downside of existing TUI-based animation solutions is the reduced accuracy, which is due partly to the use of mechanical parts, partly to the fact that, despite the adoption of a 3D input, users still have to work with a 2D output (usually represented by one or more views displayed on a screen). However, output methods that are natively 3D, e.g., based on virtual reality (VR), have been already exploited in different ways within computer animation scenarios. By moving from the above considerations and by building upon an existing work, this paper proposes a VR-based character animation system that combines the advantages of TUIs with the improved spatial awareness, enhanced visualization and better control on the observation point in the virtual space ensured by immersive VR. Results of a user study with both skilled and unskilled users showed a marked preference for the devised system, which was judged as more intuitive than that in the reference work, and allowed users to pose a virtual character in a lower time and with a higher accuracy.

CCS Concepts

***•Computing methodologies** → Animation; Graphics systems and interfaces; **•Hardware** → Tactile and hand-based interfaces; **•Human-centered computing** → Virtual reality;*

1. Introduction

Today, a wide range of applications, like videogames, movies, recommender systems and online marketplaces, to name a few, make a massive use of animated virtual characters [WFBSM12,LMBK06]. Despite the incredible use of such content, creating virtual character animations – and computer animations, in general – is still a very labor-intensive and time-consuming task. Reasons can be found in the level of expertise required [VLS18], as well as in the complexity of interfaces used [CIF12]. These facts may represent a problem for skilled users, who may be interested in speeding up the production of quality animations, but also for novice users, who may simply need a "quick-and-dirty" animation tool for sketching an idea and sharing it with other interested parties [DYP03, YSC*11, FHXS14]. For the above reasons, techniques for improving productivity by means of interfaces capable to make specific stages of the animation pipeline more intuitive and effective are becoming of particular interest [Lei12, SMTH13, KZK06].

One the most complex stages of a virtual character animation pipeline is represented by the posing step [KDMH17]. In this step, animators articulate the character's armature (or rig) either in a direct or indirect way by manipulating handles (or bones) which may be characterized by a high number of degrees of freedom (DOFs). The problem is that, although animators are requested to operate on 3D elements, interfaces that are offered by common animation tools are natively 2D [JPG*14, VLS18, Osh13].

Focusing on the issues related to user input, the literature proposes different interaction paradigms that try to face the limitations introduced by 2D interfaces (like mouse, keyboard, touch/multi-touch surfaces, etc.). One of these paradigms, known as motion capture, leverages the hand, body and/or face performance of an animator (an actor), whose movements are tracked and transferred – possibly in real time – to the character to be animated. This technique, often referred to as performance-driven animation, allows for the simultaneously manipulation of multiple DOFs [SMTH13, NQX*05]. Unfortunately, even though some tracking technologies have been developed targeted to consumer users, this technology is mostly adopted in professional animation contexts, due to the complexity of the setup, the high costs and the skills required for the performance [LvdPF00, WLS13, RTIK*14].

Still considering user input, an interaction paradigm that is re-

DOI: 10.2312/stag.20181297

ceiving an increasing attention is represented by tangible user interfaces (TUIs), which are considered as being capable to offer a number of advantages w.r.t. other techniques, especially for non-expert users [IU97]. Benefits mainly come from the higher affordances guaranteed by the handed device in terms, e.g., of native 3D manipulation and direct tactile feedback [HGCA12, YSC*11]. In fact, with TUIs, actual character's pose can be controlled by manipulating a physical prop and feeling changes in its shape.

Often, TUIs are used in special-purpose animation tools, which are either designed to control a specific character, or require sophisticated hardware in order to let the animator manage characters with arbitrary armature topologies [JPG*14, GMP*16]. In this perspective, they may be regarded as sharing the same limitations of technologies like motion capture. However, an affordable TUI-based animation framework exploiting off-the-shelf reconfigurable hardware has recently been proposed in [LPG*17]. Its added value is that it combines a tangible interaction means with performance-driven animation, thus letting animators control the virtual character's armature partly with their body, partly by articulating a handed prop, thus leveraging affordances offered by both the interfaces.

Similarly to what happens with user input, system output is affected by the limited dimensionality of the visualization methods used. In fact, with common animation tools, users are requested to continuously change the position of the virtual camera or to simultaneously look at multiple views of the scene been animated in order to visualize content of interest from the required perspectives [Dee95]. All of the above make the interfaces of existing animation tools rather sophisticated, especially for novice users [VLS18].

In the last decade, the progressive spread of virtual reality (VR) pushed the research community to look for ways to exploit this technology for addressing issues affecting computer animation interaction means (in terms of both input and output) [PST*96]. In particular, several works confirmed already the hypothesis that VR can provide animators with an improved spatial awareness of the interaction environment [VLS18], and offer them an enhanced sense of control and immersion [FBSUM14].

Despite opportunities offered for visualization, VR-based animation systems risk to fall short for what it concerns user input. In fact, even though several technologies for tracking users' hands and body in VR settings have been proposed already, in commercial VR systems the *de facto* standard for user interaction is represented by hand controllers [HTGM17]. Although controllers may bring significant advantages being natively 3D, they still share a number of features with traditional input means. Thus, given the perceived distance between the shape and behavior of the input device and those of manipulated elements (the armature's bones), animators may even loose some of the affordances offered by VR due to a possibly reduced sense of immersion [HTGM17].

By moving from the above considerations, this paper presents an animation system able to combine the benefits offered by reconfigurable TUIs and by VR technology. With the proposed system, animators can manipulate a rigged virtual character by articulating an instrumented prop into an immersive virtual environment that is integrated within a well-know, open source animation suite.

Work reported builds upon the animation system in [LPG*17], which allows animators to pose a virtual character in real time by using both their body, voice and a handed prop, and observing the results on a large screen or a projected display (hence, in the following it will be referred as the Projected System, or PS).

Compared to [LPG*17], the VR-based system presented in this paper (later referred to as VRS) lets the animators immerse themselves in the animation environment, thus helping them to get more aware of the character's pose being set up. Voice-based activation of parts to be animated has been replaced by two different approaches based on gaze and proximity selection, which are expected to guarantee a higher usability. Finally, the system introduces visual feedbacks designed to enhance users' understanding of the mapping between the manipulated interface elements and the controlled DOFs. To evaluate the effectiveness of the devised system, a user study has been carried out by asking users to carry out a character posing task. Objective measurements revealed that, compared to the PS, the proposed VRS allows animators to

- reduce the time required to perform the posing task, and
- achieve a higher accuracy in the final pose.

Objective results were validated by subjective observations. Users largely preferred the VRS to the PS, which was judged as

- being more intuitive and requiring a lower mental effort, and
- being characterized by a higher usability from all the considered perspectives.

2. Related Work

This section provides a review of computer animation methods and tools developed so far that leverage technologies exploited in the proposed system. In particular, approaches that make use of TUIs are first analyzed. Afterwards, solutions based on VR and related technologies will be considered.

2.1. TUIs for Animation

As reported in [IU97], the advantages related to the introduction of TUIs consist in a more intuitive human-machine interaction (HMI). The use of TUIs for the generation of computer animations is not recent, and ranges from leisure to professional applications. An early example is represented by a mechanical device called "waldo", which was used in the early 90's for the animation of a puppet appearing in a TV show [Wal89, Rob93]. Another example, is the sophisticated dinosaur-shaped equipment used to articulate the virtual characters featured in Jurassic Park [KHSW95].

The literature presents a number of devices designed to mimic the shape of the virtual character to be animated, since a high similarity between the input device and the virtual character is expected to increase the animation system's affordances [EPO95, Lei12]. For instance, in [NNSH11], a human-shaped doll embedded with sensors is used to define key poses and retrieve corresponding animations from a repository storing motion capture data. A similar solution is represented by the robot-shaped device developed in [YSC*11]. Here, servomotors are integrated in the tangible device to let it load pre-recorded poses, simulate natural behaviors and provide haptic feedback. A similar solution based on a teddy bear-shaped interface targeted to the animation of non-anthropomorphic characters is illustrated in [SKS*06]. The main drawback of the above solutions is the fixed structure of the interfaces, which make them unsuitable for general-purpose animation tasks [YSC*11].

To cope with the limitations of these specially shaped devices, approaches relying on reconfigurable hardware started to be developed [WIA*04, RPI04]. For instance, in [WDG08], a hub-and-strut construction kit is presented for learning and play purposes. The elements that constitute the interface can be assembled trough a ball-and-socket connection, which makes the interface suitable to articulate simple virtual puppets. Although the topology of the assembled model is automatically captured, the system does not allow users to achieve a fine-grained control on the final pose.

Control accuracy is significantly improved in [JPG*14], where a modular and reconfigurable interface for character articulation is proposed. The interface can be configured by assembling hot-pluggable instrumented components equipped with accurate rotation sensors, which allow the users to easily achieve the desired pose. Given the reduced size of the components, it is possible to reconstruct the whole character's armature. However, compactness could represent an issue, since it makes the interface costly and hard to reproduce. With this system, complex armatures can be animated by possibly controlling a subset of bones at a time. Unfortunately, in this case the intuitiveness of the TUI may be reduced, since the same tangible prop shall be reused to control parts that can be very different. Furthermore, the mapping between tangible prop's and virtual character's elements is based on a manual procedure that needs to be executed using mouse and keyboard, thus slowing down the overall process.

To overcome these limitations, the work in [GMP*16] extends the solution proposed in [JPG*14] by presenting a mapping strategy based on rig simplification that allows to control a complex character with many DOFs by using a TUI made up of few elements. The devised strategy identifies a simplified version of the original armature which can be used to control the entire character by leveraging only tangible elements available. In order to select relevant DOFs for simplification, a number of sample poses should be provided to the system during a preparatory step. The quality and size of the sample set significantly influences the result.

It is worth observing that the focus of the latter works is on virtual character articulation. Hence, should these systems be used to create complete animations, further interfaces for controlling character's position in the 3D space would have to be introduced. A way to cope with the above need is reported in [HGCA12], where a RGB-D camera is used to identify and track an unarticulated tangible prop in a desktop-scale environment for creating performance-based animations. Although the small scale setup and the fixed structure of the tangible prop make the system unsuitable for general-purpose animation scenarios, this work shows the feasibility of transferring both the shape and the position of the TUI in the virtual environment containing the character to be animated.

An animation system capable to combine the advantages of instrumented, reconfigurable TUIs and motion capture in a large-scale, unified animation environment is presented in [LPG*17]. By means of the such a system, users can pose a virtual character with both keyframing and performance-based animation methods by using their body and a tangible device. To assist the user in system configuration, an algorithm is exploited to find the best mapping between the input elements available and the DOFs of the character to be animated, and to create step-by-step instructions for assembling the tangible prop. The system has been integrated in the Blender open source animation suite. The Blender's interface is displayed on a large screen or projected on a wall, where multiple views let the user to see the character's pose from different perspectives. Common animation functionalities available in Blender (e.g., to change the view, set a keyframe, move in the timeline, etc.) are activated through customized voice commands, thus letting the animator keep his or her hands on the prop while articulating the character. When the number of interface elements is not sufficient to control the whole character's armature, additional voice commands allow the user choose the part to manipulate at a given time.

Authors of [LPG*17] proved that their system could allow both skilled and unskilled users to animate a virtual character faster than with mouse and keyboard. However, when using the TUI, users achieved a lower accuracy in the final pose and, depending on the task, experienced a possibly higher physical and mental effort. Results were partly due to the servomotors used, but also to the difficulty of keeping under control the effect of 3D manipulations to the tangible prop on a multi-view 2D screen and to voice interaction, which could get cumbersome when the number of commands to remember was particularly high.

2.2. VR and AR for Animation

The use of VR and related technologies in the production of animations brings a number of benefits related to the improved visual and spatial awareness of the environment and to interaction means made available [FBSUM14].

A number of commercial animation tools based on the above technologies are already available on the main VR stores (for the users of HTC Vive, Oculus Rift and similar equipment). For instance, VIRTUAnimator (http://store.steampowered.com/app/459870/VIRTUAnimator/) is an application that allows users to animate simple humanoid characters in a VR environment by using the hand controllers. Animations created need to be exported to other graphics tools for editing. Another example is Tvori (http://store.steampowered.com/app/517170/Tvori/), which allows users to setup a virtual scene, fill it with simple shapes, characters, props and effects selected from a library, and animate it. The Allright Rig (http://alexallright.com/allrightrig/) is a library developed to automatically generate the rig for a humanoid character in Unreal Engine. A plug-in makes it possible to control the posing in VR. Another plug-in for Unreal Engine, named Marionette VR (https://github.com/pushmatrix/marionettevr), allows users to play with a marionette character in a VR environment by adjusting its body. Thanks to the use of Oculus Rift's controllers, users can see their hands reconstructed in the virtual environment.

For the tools seen so far, flexibility in terms of characters that can be used for animation or operations allowed onto them could represent a serious limitation. Moreover, these solutions are developed as either standalone applications or plug-ins for game engines like Unreal Engine or Unity. Hence, integration with common animation suites like Blender, Autodesk Maya, etc. can take place only in a separate step of the animation workflow, thus making the process more tricky. However, some examples of flexible, integrated VR-based animation tools are already available. For instance, MARUI (https://www.marui-plugin.com/) is a software that allows users to work with Autodesk Maya in VR by using the HTC Vive's or Oculus Rift's controllers. The possibility to create animations is just an example of the wide set of functionalities that (however) are made available to VR users through conventional menus and windows.

VR- and AR-based systems for virtual character animation have been experimented also within the research community. For instance, in [VLS18], a plug-in developed for Unity allows users to create performance-based animations in VR. Users can record animations by simply moving and articulating characters and other objects into an immersive virtual environment. System functionalities (e.g., for selecting objects, controlling the timeline, etc.) are accessed via controller-activated menus. The evaluation carried out with expert animators confirmed that the time required for animation creation can be reduced, though precision (in broad terms) is lower than with traditional animation suites.

In [FBSUM14], common VR controllers are replaced by a cube-shaped TUI. AR markers on cube's faces let the user choose the animation to activate (from a pre-defined set) and adjust its speed. Character's position is controlled by grabbing the cube. The user visualizes in real time the resulting animation in AR on a tablet device. The main drawback of this approach is represented by its reduced control possibilities (limited by the use of the cube and its faces as input). Although interface's flexibility could be incremented by considering cube-based gestures (like cube shaking), the use of pre-defined animations could still represent a severe limitation for general-purpose animation scenarios. Furthermore, like in most of previous works, no editing is allowed on recorded animation from within the virtual environment.

The authors of [HTGM17] move from considering that neither special-purpose interfaces (like the cube used in [FBSUM14]), nor general-purpose VR controllers would be capable to create the necessary connection between the real and the virtual worlds. Hence, they propose a custom-designed sensor unit assembled with low-cost off-the-shelf hardware that could be mounted on a variety of common physical objects to track them in a VR environment. Four prototypes (a simple cube, a stuffed animal, a treasure chest, and a wooden boat) are created and exploited in VR narrative animation scenarios to demonstrate that the active and passive feedback information provided by different objects can effectively improve user experience. However, character articulation is not considered, due to the type of sensors used in the tangible assembly.

By moving from the review of the above works, this papers presents an animation system combining both TUI and VR technologies. As said, the proposed system builds upon the PS in [LPG*17], thus exploiting the affordances of a reconfigurable tangible prop that can be used together with a body tracking mechanism to mimic as much as possible the DOFs of the character to be animated and to pose it in the 3D space in a natural way. The novelty consists in the use of VR to let the users immerse themselves in the animation environment, by providing them with an increased awareness of the mapping between physical and virtual elements and with an immediate feedback on the effect of 3D manipulations on the virtual character's pose. A user posing a virtual character with the PS is shown in Fig. 1.a. The functioning of the proposed VRS is illustrated in Fig. 1.b-c. Compared to other TUI- and VR-based solutions above, the proposed system leverages an articulated device, which proved already to be particularly effective to reduce posing time and is expected to be able to guarantee a higher flexibility compared to custom props or common hand controllers.

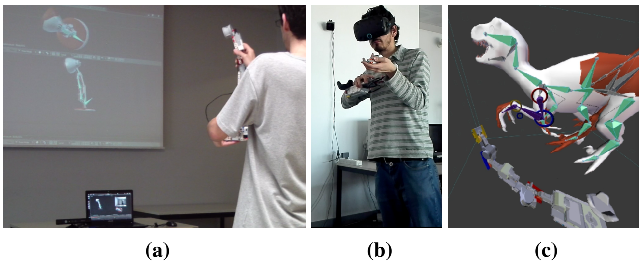


Figure 1: *TUI-based character animation: a) in the PS, user observes the results on a multi-view wall projection, b) in the proposed VRS, user wears a head-mounted display and operates in an immersive environment, c) user's first-person view in the devised VR-based animation environment, including the character and a virtual reconstruction of the TUI used to articulate it.*

3. System Overview

As said, the VR-based animation system proposed in this paper extends the work reported in [LPG*17]. Hence, in the following the proposed system will be presented starting from the high-level architecture of such work (Fig. 2), and showing how it has been modified to accommodate the new features developed (components added or changed are drawn as square boxes in the figure). The behavior of the main modules is discussed in the following subsections. New features will then be analyzed in detail in Section IV. Source code is available for download at https://goo.gl/yKwc2d.

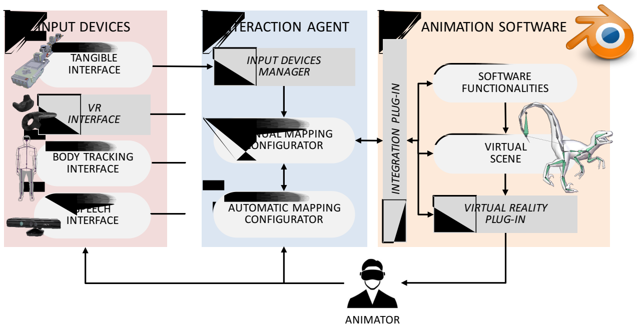


Figure 2: *Architecture of the proposed animation system.*

3.1. Input Devices

The block named *Input Devices* groups the user interfaces that can be exploited to collect information for controlling the position and orientation of bones in the virtual character's armature and for managing the animation system. Like in the PS, several sensing technologies are supported, including color and depth cameras, microphones, positional and rotational sensors, etc. In particular, the *Tangible Interface* is implemented by using the servomotors, sensors and bricks in the core and expansion sets of the Lego Mindstorms EV3's Education Kit. One or more Lego Mindstorms' Intelligent Brick components collect the data measured by the servomotors and the sensors. Data read are sent to the *Interaction Agent* as JSON strings over a Wi-Fi/USB connection thanks to a third-party API (https://github.com/BrianPeek/legoev3). On average, with the Wi-Fi connection, a sampling rate of 10Hz is reached (much higher than over the USB link, due to known issues with the library). The *Body Tracking Interface* is implemented by leveraging the real-time skeleton data gathered by using a Microsoft Kinect device. The *Speech Interface*, which was built upon the Microsoft Speech Platform library (https://msdn.microsoft.com/en-us/library/jj127858.aspx), is in charge of recognizing voice commands used to activate animation functionalities.

With respect to the PS, the proposed VRS includes a *VR Interface* based on the HTC Vive suite. This interface supports as further sources of position and orientation data (and, possibly, of configurable inputs) the elements tracked by the VR suite, i.e., the controllers and the trackers (and the headset, which is used to determine the user's point of view). HTC Vive's trackers are managed with the same technology used to track the controllers, and have been designed to bring any physical object into the VR environment. In this work, only one tracker is used, which is mounted on the tangible prop for tracking its position and orientation in the 3D space. However, in principle the same technology could be used also to implement the *Body Tracking Interface*.

3.2. Interaction Agent

Measurements produced by the *Input Devices* are handled by the *Interaction Agent* component through the *Input Devices Manager*. This component converts received raw data into information suitable for virtual character posing. To this purpose, a strategy to map the available interface elements (servomotors, sensors, body joints, and HTC Vive tracker) onto the virtual character's DOFs is needed.

In the PS, two approaches were developed. The first approach consists in a manual setup that is carried out by the user through a graphics tool named *Manual Mapping Configurator*. The second approach relies on a set of mapping rules that are defined automatically by a so-called *Automatic Mapping Configurator*. In this case, an unsupervised algorithm optimizes the assignment of each individual interface element to a specific DOF of the character's armature by considering several factors, like the similarity between the topology of the virtual character and the possible assemblies of available interface elements, the DOFs (and related ranges) of character's bones and of interface elements, the presence of symmetries in the armature's topology, etc. The automatically generated configuration may be refined by means of the manual tool, e.g., to consider different user's preferences. When the number of DOFs to be controlled is higher than the interface elements available, the character's armature is split into a number of so-called "partitions", in a way that allows the animator to control the subset of bones in a given partition with the same configuration of the interface without the need to reassemble it. Each partition is assigned a voice command, which is used by the animator to activate the control of the corresponding bones.

Data received from the *Input Devices* can be mapped both directly onto virtual character's bones (thus providing the animator with a forward kinematics-based control), or indirectly onto end-effectors (thus letting him or her exploit an inverse kinematic-based approach). Mapping rules, both manual and automatic, are also exploited to generate step-by-step instructions to assemble the Lego Mindstorms bricks that constitute the handed prop in order to simplify the use of the animation system.

The *Interaction Agent* is also responsible for managing the vocabulary that is used to translate the recognized voice commands into *Software Functionalities* (described in the next sub-section).

3.3. Animation Software

Information regarding the scene to be animated (position of the 3D objects, armatures, etc.) is handled by the *Animation Software*, which also provides a number of *Software Functionalities*, e.g., for selecting character's bones to be controlled, defining/removing/copying/pasting keyframes, navigating the timeline, visualizing the animation, enabling/disabling continuous keyframing (for performance-driven animation), selecting the view, etc. In the PS, this component was implemented onto the Blender open source 3D modeling and animation suite. An *Integration Plug-in* tailored to the specific software used was developed in order to interface the *Animation Software* with the *Interaction Agent*.

With respect to the PS, in the proposed VRS the *Animation Software* includes a further component named *VR Plug-in*. The plug-in, written as a Python script for Blender, allows the animators to visualize the Blender's 3D viewport through the HTC Vive headset, and provide them with an actionable representation of the articulated tangible prop into the virtual environment. Based on such visualization, new features easing the interactions requested during animation were implemented, as it will be detailed in Section 4. In order to preserve compatibility with the PS, the newly developed *Interaction Plug-in* lets the user easily switch between the VR-based interaction and the previous visualization modality based on large-screen or wall projection.

4. Proposed VR-based Features

In this section, changes made to the reference PS will be discussed, by first summarizing its drawbacks, and then showing how to cope with them through the new features developed.

4.1. Drawbacks of the Reference System

Experimental results carried out in [LPG*17] demonstrated that, compared to traditional interfaces, the use of an articulated, reconfigurable TUI could make virtual character animation more intuitive and accessible to both skilled and unskilled users. However, although the authors showed that, by using the devised interface, posing time could be reduced in the considered animation tasks, they also found that the accuracy of the final pose was higher when working with mouse and keyboard.

According to subjective feedback and empirical observations, these results were due to multiple factors. First, the system did not allow the animator to precisely decide how to observe the virtual scene, as he or she could only choose among a pre-defined set of views that could be either shown together in a split viewport or activated individually using voice commands. Moreover, even though the system took advantage of 3D input means, the visualization of the Blender's interface was still based on a 2D output. These factors impacted on the understanding of the virtual character's pose.

Furthermore, the lack of a visual feedback representing the mapping between the interface element(s) being manipulated and the controlled DOF(s) reduced the intuitiveness and effectiveness of the paradigm adopted. In fact, the only way for the user to realize which was the effect of a given interaction with the TUI was represented by a trial-and-error process, consisting in adjusting the prop and looking at changes in the character's armature. However, intentional changes to a given sensor/servomotor and controlled DOF could bring unintentional changes to other armature's elements.

A last issue was related to the selection of the armature's partition to be controlled. Although the method based on voice commands allowed the animator to perform a quick selection, it proved to be complicated to use when the virtual character was split in a number of different partitions (each identified by a name and a corresponding voice command). Thus, when a partition that had been already posed was unintentionally activated, changes in the TUI could introduce unwanted modifications to character's DOFs.

To cope with the above issues, this paper proposes a shift to a VR-based interaction. The use of VR was expected to be capable to provide the animator with an improved perception of the 3D space, as he or she could move freely in the virtual environment and choose the best observation point to perform a given manipulation. Notwithstanding, by leveraging the availability of a VR-based visualization, it was also possible to introduce specific features to tackle limitations of the PS in [LPG*17], which concern manipulation feedback and selection of armature's partitions. New features are discussed in the next sub-sections.

4.2. Visual Cues

As said, users of the PS were not provided with a way to visualize the currently active mapping between interface elements and virtual character's DOFs.

In the proposed VRS, the user is immersed in a VR environment, which also displays the position and orientation of the tangible prop (through the attached tracker), as well as the configuration of its elements (as servomotors and sensors continuously transmit their status to the *Animation Software*). Indeed, this feature was essential to preserve the affordances guaranteed by the use of a TUI in the VR environment. However, as shown in Fig. 3, the availability of a virtual representation of the tangible prop was exploited also to address the limitations of the approach used in the PS, by providing the user with suitable visual cues.

Cues are managed by the *VR Plug-in*. When no partition is selected (Fig. 3.a), armature's bones that can be controlled are shown with a green color (remaining bones are assigned a gray color), whereas the TUI is drawn as it appears in the real world. When a partition is selected (see Section 4.3), all the bones belonging to it are displayed with a different color. The same color is used to draw the interface elements in the TUI controlling the selected partition (Fig. 3.b). However, the actual mapping between interface elements and bones' DOFs is not visible yet. To make the mapping rules for a given partition active (and display them), the user needs to wait for a certain amount of time. After activation, for each bone in the partition one or more circles are displayed to show the DOFs that can be controlled by the animator (one circle per DOF). DOFs are drawn with different colors. The same color is used to draw the TUI's element which is mapped onto that DOF (Fig. 3.c).

4.3. Selection Modalities

In the PS, each partition in the character's armature is activated through a voice command. In the proposed system, thanks to VR it was possible to develop two alternative activation modalities, referred to as "gaze" and "proximity" selection (in the future, approaches based, e.g., on head-mounted depth cameras could be investigated [MZ18]).

Gaze selection works as in many VR-based applications, and leverages fixation time, i.e., the time the user keeps looking the same point (or bone, in this case) in the 3D space (to this purpose, a virtual aim is displayed, as shown in Fig. 3). After a certain amount of time, the partition containing the given bone gets activated. The behavior of proximity selection is similar, though in this case the position of the tangible prop is exploited. As said, the prop is tracked using a HTC Vive tracker mounted on it. When the prop (the part representing the tracker, identified by a set of axes, as shown in Fig. 3) is kept close to a bone for a certain time, i.e., it is snapped to it, the partition that contains the bone is activated.

Like for visual cues, selection is managed by the VR Plug-in, which is able to combine bones' data with information concerning the position of the animator's head, the direction of his or her gaze, as well as the position of the tangible prop. It is worth observing that preliminary experiments with the above modalities suggested that gaze selection could be considered as more intuitive than proximity selection (being a common paradigm in VR applications), but it could be hard to use in the given context when bones are small or far from the animator (as it may be difficult to keep the gaze fixed for the fixation time set). Similarly, proximity selection may not work well, e.g., when the bone is hard to reach in the virtual environment (too low, too high, etc.). For these reasons, both the modalities were made available to the animator. To avoid activating a partition while the animator is articulating a different set of bones, selection needs to be initiated through a voice command.

5. Experimental Setup

In order to evaluate the effectiveness of the devised approach, a user study was carried out by involving several volunteers, who were asked to perform a posing task in three modalities, i.e., by using the proposed VRS, the reference PS, as well as the Blender's mouse and keyboard interface (later referred to as MK).

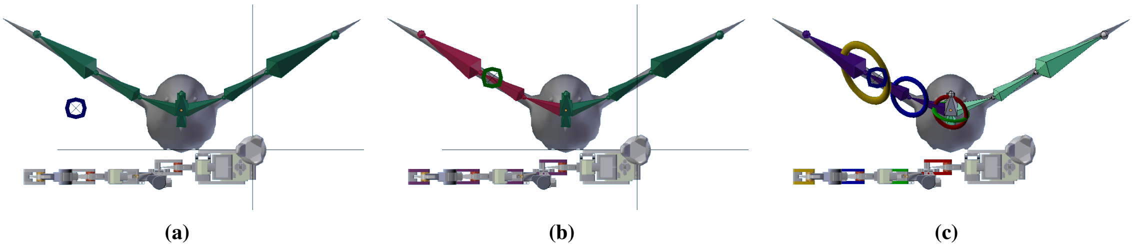

(a) (b) (c)

Figure 3: *Aspect of character's bones and TUI during interaction: a) no partition selected, b) partition selected, bones and interface elements highlighted in red, and c) partition activated, mapping rules shown with with controlled DOFs for each bone displayed using colored circles. Aim for gaze selection is shown, whereas axes in a) and b) identify the point of the TUI used for proximity selection (tracker mount).*

5.1. Case Study

The virtual character considered for the experiments is the dyno object that was exploited in [LPG*17], which was selected given the high number of bones and DOFs to be controlled.

The tangible interface assembled is made up of four large servomotors. The HTC Vive tracker mounted on the Intelligent Brick is used to track it in the 3D space. The resulting configuration is illustrated in Fig. 4. For each armature's bone, DOFs that can be controlled are shown. Names corresponding to voice commands used to activate each of the nine partitions with the PS are reported (corresponding bones are shown with the same color). Finally, arrows are used to indicate an example of mapping between interface elements and controlled DOFs for bones in a selected partition (namely, "body"). Mapping rules are defined to force the animator to use both forward and inverse kinematics. In particular, the "tail" partition is controlled by exploiting inverse kinematics (i.e., the position of the tangible interface is mapped on the position of the end-effector), whereas the remaining partitions are controlled with forward kinematics. In the experiments, users were asked to articulate the armature in Fig. 5 to pass from the initial, or rest pose (in white/gray) to the target pose (in red).

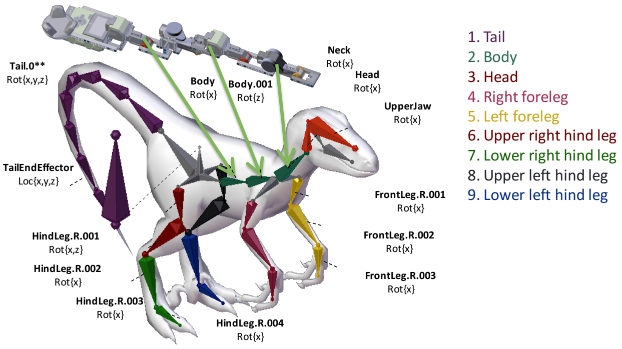


Figure 4: *Dyno character used in the experiments (here, colors are used only to let the reader distinguish the various partitions, not to refer to interactions aimed to partition selection).*

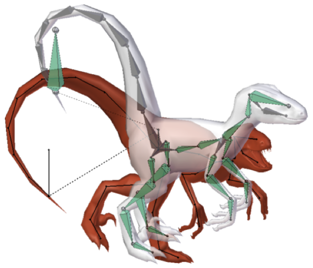


Figure 5: *Armature's rest pose (white) and target pose (red).*

5.2. Procedure

The experiments were performed by involving 20 volunteers (17 males, three females), aged between 22 and 34 years, who were selected among students and academic staff at the authors' university. Half of the participants could be considered as skilled users, because of their expertise with computer animation suites gained by attending and/or teaching 3D modeling and animation courses. The remaining participants were considered as unskilled users.

All the participants were asked to carry out the posing task in the three modalities. Latin Order was used to choose the modality to start and continue with, with the aim to reduce possible learning effects. Each participant was introduced to the main functionalities of the animation system used in a given modality that were needed to complete the task. Afterwards, they were given time to get acquainted with the VR environment and were allowed to familiarize with the interfaces of the considered systems by experimenting with a different posing task.

For the modalities involving the PS and the VRS, participants were then asked to complete the following procedure:

1. issue a voice command to start the task (the command activated the collection of measures exploited in the evaluation);
2. select an armature's partition by issuing the corresponding voice command (when using the PS) or by exploiting the gaze/proximity selection methods (when using the VRS);
3. issue a voice command to reset the mapping between data from the interface elements and the DOFs of the controlled armature;
4. adjust the position or orientation of the bones in the selected

partition by moving and articulating the tangible prop to match the target armature using either forward or inverse kinematics;

5. iterate steps 2 to 4 until the target pose has been reproduced as close as possible (armatures overlapped);
6. issue a voice command to terminate the task.

Two videos showing task execution with the PS and the VRS are available for download at https://goo.gl/6WcP4m.

For the MK modality, participants were asked to adjust the configuration of single bones as needed, by possibly using the simplified interface based on Blender's position and orientation handles.

Like in [LPG*17], no time limit or accuracy goal was set. Volunteers were left free to operate until they considered the task as completed, e.g., because they were not able to improve the pose further. However, an audio signal was used to inform them that accuracy had reached a given threshold (threshold was intentionally set to a high value, namely 20%, so that all the participants kept on adjusting the pose despite the signal).

5.3. Performance Metrics

To assess the effectiveness of the VRS presented in this paper and compare it with the considered alternatives, both objective and subjective measurements were collected.

Objective measurements exploited the three indicators defined in [LPG*17], and were calculated for all the modalities. The first indicator, named *completion time* (T_c), considers the time needed by a volunteer to carry out the whole task, i.e., to finalize the pose of the controlled character.

The second indicator, named *pose distance* (D), measures how much the current pose of the controlled armature differs from the target pose in terms of Euclidean (for the position) and angular (for the orientation) distances. During the task, the value of the pose distance decreases from an initial 100% value to an ideal 0% value when the two armatures perfectly overlap.

The third indicator, named *amount of work* (W), estimates the work required to pose the character. The indicator is calculated as the area under the curve defining the variation of D during animation from time $t = 0$ to time $t = T$, with T corresponding to the maximum time at which the minimum value of D is reached with the considered systems. Area is normalized dividing by the animation time. Briefly, this indicator tells how fast was the animator in making the character's pose converge towards the target one (the lower the amount of work is, the faster the operation was).

Subjective observations were collected by means of an after-test questionnaire, which is available at https://goo.gl/A2fgLK. Though objective evaluation considered also the MK modality (with the goal to gather a numerical ground-through under conditions which were slightly different from those in [LPG*17]), subjective evaluation focused only on the PS and VRS, since a strong preference for PS compared to MK had been already demonstrated. The questionnaire included two sections. The first section was aimed to study the ergonomics factors associated with the interaction means used in the two systems based on the ISO 9241-400 standard. Users' preference for either the PS or the VRS was additionally recorded. The second section was aimed to analyze specific usability factors related to interaction with virtual content according to [Kal99].

6. Results

In the following, the results of objective and subjective observations obtained with the experiments described above will be presented.

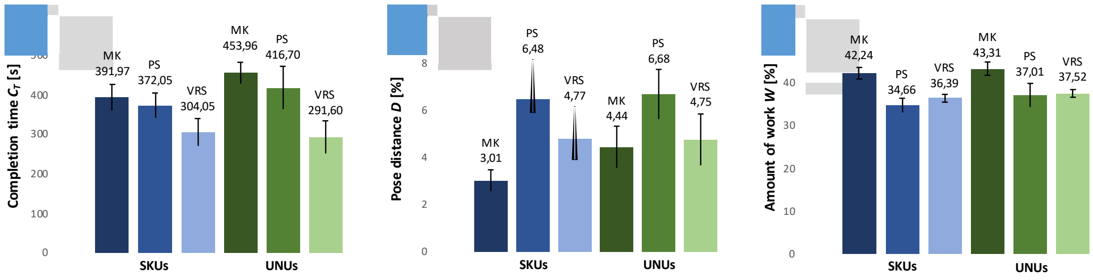
6.1. Objective Observations

Average measurements concerning operation speed (completion time), pose accuracy (pose distance) and amount of work obtained by skilled (SKUs) and unskilled (UNUs) users with MK, the PS and the VRS are reported in Fig. 6. Statistically significance of results was evaluated using ANOVA and paired t-tests ($p = 0.05$).

By focusing first on SKUs, it can be observed that the VRS brought significant benefits in terms of completion time. On average, users were 18% and 22% faster with the VRS than with both the PS ($p = 0.0018$) and MK ($p = 0.0180$). With the VRS, users were also more accurate that with the PS, reaching a pose distance equal to 4.77% w.r.t. 6.48% obtained using the PS ($p = 0.0002$). The VRS did not allow users to reach the accuracy of MK (4.77% vs 3.01%), but the gap was much lower than with the PS (6.48%). For what it concerns the amount of work, no significant difference was found between the VRS and the PS. This result was not surprising. In fact, even though, with the VRS, users may be faster in setting a given DOF because they can see it from a better perspective or can understand how to control it in an easier way thanks to visual cues, they also need to spend part of the time to reposition in the virtual environment, to activate partitions by using either gaze or proximity selection (with their waiting times) and to understand the current mapping rules. These operations are not possible, not requested or implemented in a different way in the PS (e.g., partitions are selected by means of voice commands). Hence, although the new functionalities introduced in the VRS allow the users to complete the animation in a shorter time, this fact is not relevant for the computation of W, since the contribution of T_c is removed by a normalization. Notwithstanding, the amount of work with both the VRS and the PS was lower than with MK, confirming the effectiveness of the TUI as a means for sketching the character's pose.

Results obtained for UNUs confirm the above trends. In particular, saving in terms of time requested to complete the task was even higher, as users were 30% and 36% faster with the VRS than with the PS, $p = 0.0165$ and MK, $p = 0.0001$. For what it concerns pose distance, improvement ensured by the VRS w.r.t. the PS was comparable to that experienced by SKUs (6.68% vs 4.75%, $p = 0.0383$). However, in this case no statistical significance was found between the VRS and MK ($p = 0.7300$), meaning that the VRS allowed UNUs to achieve an accuracy comparable to that of MK. Considering the amount of work, the same considerations made for SKUs hold also for UNUs.

Interesting insights can be obtained also by considering UNUs vs SKUs. In particular, it can be observed that SKUs were faster than UNUs when using both the PS and MK. However, SKUs and UNUs performed almost the same way when using the VRS, being both faster and more accurate than with the PS. These results indicate that the VRS was more effective than the PS in smoothing the



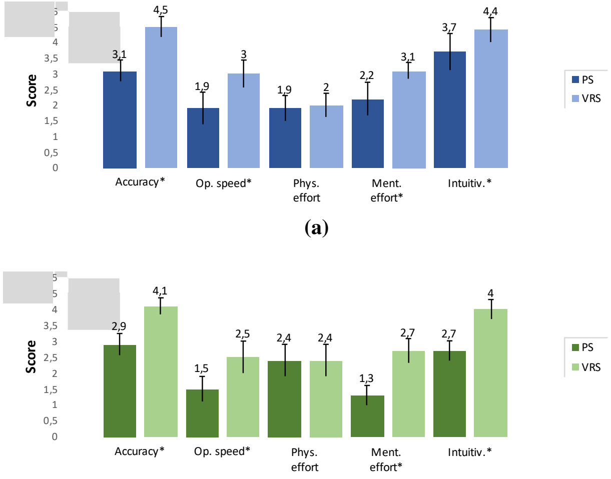
(a) (b) (c)

Figure 6: *Results in terms of a) completion time, b) pose distance and c) amount of work for SKUs and UNUs.*

lack of previous computer animation expertise, confirming its helpfulness for novice users. Concerning accuracy, SKUs performed better than UNUs only with MK, as expected. No difference was found between UNUs with SKUs in terms amount of work.

6.2. Subjective Observations

Questions in the first section of the questionnaire asked users to evaluate (perceived) accuracy and operation speed, physical and mental effort as well as intuitiveness of the interaction with the PS and the VRS on a 1-to-5 scale. Moreover, as said, users were asked to express their preference between the two systems, by providing motivations for their choice. Results for SKUs and UNUs are reported in Fig. 7.a and Fig. 7.b, respectively. For sake of readability, scores concerning physical and mental effort have been mapped on a better-to-worse, 5-to-1 scale (thus, a higher score corresponds to a lower effort). Statistically significant results (based on paired t-tests, $p = 0.05$) are marked with *.



(a)

(b)

Figure 7: *Subjective results concerning interaction means based on ISO 9241-400 factors for a) SKUs and b) UNUs.*

Overall, 19 out of the 20 participants preferred the VRS. Based on motivations provided, choice was mostly driven by the greater awareness of the virtual character to be controlled and by a higher level of control over the animation system.

More insights about this clear preference can be obtained by analyzing answers to specific questions. By focusing on questions that are statistically significant, it can be observed that the VRS was perceived by both the user categories as more accurate ($p = 0.0026$ for SKUs, $p = 0.0025$ for UNUs) and faster ($p = 0.0483$ for SKUs, $p = 0.0319$ for UNUs) than the PS, as confirmed by objective measurements. Both the user categories also found the VRS as more intuitive ($p = 0.0248$ for SKUs, $p = 0.0037$ for UNUs) and requiring a lower mental effort ($p = 0.0099$ for SKUs, $p = 0.0095$ for UNUs) compared to the PS. These results appear to confirm the usefulness of the visual cues as well as the suitability of the alternative methods introduced to select armature's partitions. For what it concerns physical effort, no significant difference was found. For both the systems, scores were not particularly high, probably due to weight of the tangible prop. Notwithstanding, physical effort with the VRS was comparable to that with the PS, meaning that the devised immersive environment did not introduce significant issues, e.g., concerning eye strain or motion sickness. Comparing feedback by the two user categories on the same system, it can be observed that advantages brought by the VRS are more evident for UNUs than for SKUs, especially for what it concerns mental effort and intuitiveness. This is not surprising, since SKUs were already used to work with the multi-view visualization of the PS.

Questions in the second section of the questionnaire concerned the nine usability categories identified in [Kal99], namely, functionality, user input, system output, user guidance and help, consistency, flexibility, error correction/handling and robustness, sense of immersion/presence and overall system usability. Questions were expressed in the form of statements to be evaluated using a 1-to-5 scale (from strong disagreement to strong agreement). For each category, users also had to provide an overall evaluation again in 1-to-5 scale (from very unsatisfactory to very satisfactory). Scores assigned to overall evaluation questions are reported in Table. 1. It can be easily noticed that both SKUs and UNUs expressed a higher appreciation for the VRS than for the PS, since each category received, on average, a higher score (statistical significance based on paired t-tests is shown). In order to dig into reasons for the above results, scores for questions in each category can be investigated by considering disaggregated data at https://goo.gl/nmHpg2.

Focusing on statistically significant questions, it can be observed that for the first factor, both the user categories found that functionalities of the VRS were easier to access, and their meaning was clearer than with the PS. SKUs also found it easier to remember the functionalities of the VRS, whereas UNUs did not found differences between the two systems from this point of view.

Results concerning user input indicate that both SKUs and UNUs found that it was easier to use the input device as well as to move and reposition themselves in the virtual environment by using the VRS than the PS. UNUs additionally found that with the VRS they made less mistakes, they had the right (higher) level of control over what they wanted to do and it was easier for them to select and move virtual objects.

Considering scores assigned to statements about system output, it comes out that both user categories found the display and the field of view more appropriate with the VRS than with the PS. They also found that feedback shown was more adequate and information was presented in a more meaningful way with the VRS. Moreover, differently than with the PS, with the VRS they did not lack the sense of depth. No statistical significance was found for the question concerning nausea or eye fatigue, meaning that users felt comfortable with using both the wall projection and the immersive head-mounted display.

For what it concerns user guidance and help, no statistical significance was found for questions regarding difficulty to learn how to use the system and the need for further help, i.e., the two system could be considered as comparable in this respect. However, both SKUs and UNUs found the PS more difficult to use than the VRS.

Considering consistency, both user categories found that the sequence of inputs to perform a specific action matched in a better way their understanding of the task when using the VRS than the PS. This finding may be associated with the different set of commands for choosing partitions that is permitted by the use of VR (together with gaze and proximity selection). UNUs also found that they were less confused and more confident that the system performed as they were expecting when using the VRS than the PS.

W.r.t. flexibility, both SKUs and UNUs found it easier to perform the task in the way (order) they chose with the VRS, as they were able to tailor the system to their needd. Users also stated that the VRS allowed them to take shortcuts, probably because of the availability of different selection modalities and the possibility to move and observe the virtual character from the preferred point of view.

Concerning error correction/handling and robustness, both user categories felt that with the PS they were unaware of making mistakes. It is likely that this result was due to the fact that, in the PS, in order to determine whether a DOF was correctly set or not users had to check multiple views. In the VRS it was sufficient for them change a bit their point of view by moving their head or body.

Regarding the sense of immersion/presence, both SKUs and UNUs indicated that they had the feeling of being part of the virtual character's space when using the VRS, and that this fact provided them with a higher sense of scale than in the PS.

Finally, when overall usability is considered, both user categories enjoyed more the VRS than the PS, and said to see a real benefit in the use of VR as a man-machine interface. Both SKUs and UNUs found it more difficult to manage three-dimensionality with the PS than with the VRS. Additionally, UNUs found it more difficult to learn how to use the PS than the VRS, they felt less in control, and sometimes they did not have a clear idea of how to perform a particular operation. These latter findings confirm that benefits brought by the VRS could be even higher for UNUs than for SKUs.

Table 1: *Subjective results concerning usability of the two systems for SKUs and UNUs in terms of the factors defined in [Kal99].*

	SKUs			UNUs		
Factor	**PS**	**VRS**	***p*-value**	**PS**	**VRS**	***p*-value**
Functionality	3.6	4.6	0.0084	2.9	4.3	0.0026
User input	3.4	4.1	0.0248	2.5	3.9	0.0005
System output	3.2	4.3	0.0483	2.7	4.2	0.0011
User guidance and help	3.5	4.5	0.0010	2.9	4.1	0.0025
Consistency	4.0	4.6	0.0238	3.1	4.2	0.0067
Flexibility	3.2	4.2	0.0010	2.5	3.8	0.0037
Error correction	3.7	4.3	0.0238	2.8	4.2	0.0044
Sense of immersion	2.5	4.5	0.0029	2.4	4.9	0.0001
Overall usability	3.0	4.3	0.0094	2.8	4.7	0.0010

7. Conclusions and Future Work

By building upon previous achievements in the field of TUI-based character animation, the aim of this paper was to show how affordances offered by reconfigurable, tangible input devices can be further enhanced by passing from a 2D representation of the virtual scene, possibly based on multiple views, to a 3D visualization leveraging immersive VR. Thanks to VR, users are allowed to easily choose the best point of view in the scene for carrying out a given task. Moreover, they are provided with intuitive ways for understanding spatial constraints, for selecting character's parts to be articulated, and for realizing how their actions on the tangible device translate into modifications of the character's shape.

Experiments showed that, compared with a TUI-based-only solution, the proposed VR-based system allowed both skilled and unskilled users to reduce the time spent to complete the assigned animation task, by also letting them achieve a higher posing accuracy. Moreover, when using the VR-based system, both user categories were faster than with the conventional mouse and keyboard interface. Unskilled users also achieved undistinguishable levels of accuracy. Interestingly, a further added value of the VR-based approach was that it reduce the impact of previous experience, since it allowed unskilled users to perform as skilled users. The proposed system was also judged as being characterized by a higher usability from all the perspectives considered (except physical effort).

Despite the above advantages, accuracy obtained by skilled users was still somewhat lower than with mouse and keyboard. Hence, future work could be aimed to improve performance of technology considered, e.g., by considering different tracking methods. Alternatively, animators could be provided with suitable ways for fine tuning the mapping between TUI's and character's modifications during interaction. Finally, experimental evaluation should be extended to other animation scenarios (like facial animation) and user categories (like professional artists), in order to characterize system performances under different conditions.

Acknowledgements

Work was partially supported by the VR@POLITO initiative.

References

[CIF12] Chen J., Izadi S., Fitzgibbon A.: Kinêtre: Animating the world with the human body. In *Proc. 25th Annual ACM Symposium on User Interface Software and Technology* (2012), pp. 435–444. 1

[Dee95] Deering M. F.: Holosketch: A virtual reality sketching/animation tool. *ACM Transactions on Computer-Human Interaction (TOCHI) 2*, 3 (1995), 220–238. 2

[DYP03] Dontcheva M., Yngve G., Popović Z.: Layered acting for character animation. *ACM Transactions on Graphics 22*, 3 (2003), 409–416. 1

[EPO95] Esposito C., Paley W. B., Ong J.: Of mice and monkeys: A specialized input device for virtual body animation. In *Proc. Symposium on Interactive 3D Graphics* (1995), pp. 109–ff. 2

[FBSUM14] Fernández-Baena A., Salle-URL L., Miralles D.: Avatar: Tangible interaction and augmented reality in character animation. *Proc. Works. on Interacting with Smart Objects* (2014). 2, 3, 4

[FHXS14] Feng A., Huang Y., Xu Y., Shapiro A.: Fast, automatic character animation pipelines. *Computer Animation and Virtual Worlds 25*, 1 (2014), 3–16. 1

[GMP*16] Glauser O., Ma W.-C., Panozzo D., Jacobson A., Hilliges O., Sorkine-Hornung O.: Rig animation with a tangible and modular input device. *ACM Transactions on Graphics 35*, 4 (2016), 144. 2, 3

[HGCA12] Held R., Gupta A., Curless B., Agrawala M.: 3D puppetry: A Kinect-based interface for 3D animation. In *Proc. 25th Annual ACM Symposium on User Interface Software and Technology* (2012), pp. 423–434. 2, 3

[HTGM17] Harley D., Tarun A. P., Germinario D., Mazalek A.: Tangible VR: Diegetic tangible objects for virtual reality narratives. In *Proc. Conference on Designing Interactive Systems* (2017), pp. 1253–1263. 2, 4

[IU97] Ishii H., Ullmer B.: Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proc. Conference on Human Factors in Computing Systems* (1997), pp. 234–241. 2

[JPG*14] Jacobson A., Panozzo D., Glauser O., Pradalier C., Hilliges O., Sorkine-Hornung O.: Tangible and modular input device for character articulation. *ACM Transactions on Graphics 33*, 4 (2014), 82. 1, 2, 3

[Kal99] Kalawsky R. S.: VRUSE – A computerised diagnostic tool for usability evaluation of virtual/synthetic environment systems. *Applied Ergonomics 30*, 1 (1999), 11–25. 8, 9, 10

[KDMH17] Kytö M., Dhinakaran K., Martikainen A., Hämäläinen P.: Improving 3D character posing with a gestural interface. *IEEE Computer Graphics and Applications 37*, 1 (2017), 70–78. 1

[KHSW95] Knep B., Hayes C., Sayre R., Williams T.: Dinosaur input device. In *Proc. Conference on Human Factors in Computing Systems* (1995), pp. 304–309. 2

[KZK06] Kim S., Zhang X., Kim Y. J.: Haptic puppetry for interactive games. In *Proc. Conference on Technologies for E-Learning and Digital Entertainment* (2006), pp. 1292–1302. 1

[Lei12] Leite L.: Virtual marionette. In *Proc. International Conference on Intelligent User Interfaces* (2012), pp. 363–366. 1, 2

[LMBK06] Luo J., McGoldrick P., Beatty S., Keeling K. A.: On-screen characters: Their design and influence on consumer trust. *Journal of Services Marketing 20*, 2 (2006), 112–124. 1

[LPG*17] Lamberti F., Paravati G., Gatteschi V., Cannavo A., Montuschi P.: Virtual character animation based on affordable motion capture and reconfigurable tangible interfaces. *IEEE Transactions on Visualization and Computer Graphics* (2017). 2, 3, 4, 5, 6, 7, 8

[LvdPF00] Laszlo J., van de Panne M., Fiume E.: Interactive control for physically-based animation. In *Proc. 27th Annual Conference on Computer Graphics and Interactive Techniques* (2000), pp. 201–208. 1

[MZ18] Memo A., Zanuttigh P.: Head-mounted gesture controlled interface for human-computer interaction. *Multimedia Tools and Applications 77*, 1 (2018), 27–53. 6

[NNSH11] Numaguchi N., Nakazawa A., Shiratori T., Hodgins J. K.: A puppet interface for retrieval of motion capture data. In *Proc. Symposium on Computer Animation* (2011), pp. 157–166. 2

[NQX*05] Nguyen T. H. D., Qui T. C. T., Xu K., Cheok A. D., Teo S. L., Zhou Z., Mallawaarachchi A., Lee S. P., Liu W., Teo H. S., et al.: Real-time 3D human capture system for mixed-reality art and entertainment. *IEEE Transactions on Visualization and Computer Graphics 11*, 6 (2005), 706–721. 1

[Osh13] Oshita M.: Multi-touch interface for character motion control using model-based approach. In *Cyberworlds (CW), 2013 International Conference on* (2013), IEEE, pp. 330–337. 1

[PST*96] Pausch R., Snoddy J., Taylor R., Watson S., Haseltine E.: Disney's Aladdin: First steps toward storytelling in virtual reality. In *Proc. 23rd Annual Conference on Computer Graphics and Interactive Techniques* (1996), ACM, pp. 193–203. 2

[Rob93] Robertson B.: Motion capture meets 3D animation. In *On the Cutting Edge of Technology* (1993), pp. 1–14. 2

[RPI04] Raffle H. S., Parkes A. J., Ishii H.: Topobo: A constructive assembly system with kinetic memory. In *Proc. Conference on Human Factors in Computing Systems* (2004), pp. 647–654. 3

[RTIK*14] Rhodin H., Tompkin J., In Kim K., Varanasi K., Seidel H.-P., Theobalt C.: Interactive motion mapping for real-time character control. *Computer Graphics Forum 33*, 2 (2014), 273–282. 1

[SKS*06] Shimizu N., Koizumi N., Sugimoto M., Nii H., Sekiguchi D., Inami M.: A teddy-bear-based robotic user interface. *Computers in Entertainment 4*, 3 (2006), 8. 3

[SMTH13] Shiratori T., Mahler M., Trezevant W., Hodgins J. K.: Expressing animated performances through puppeteering. In *Proc. Symposium on 3D User Interfaces* (2013), IEEE, pp. 59–66. 1

[VLS18] Vogel D., Lubos P., Steinicke F.: AnimationVR – Interactive controller-based animating in Virtual Reality. In *Proc. 1st Workshop on Animation in Virtual and Augmented Environments* (2018). 1, 2, 4

[Wal89] Walters G.: The story of Waldo. In *SIGGRAPH Course Notes: 3D Character Animation by Computer* (1989). 2

[WDG08] Weller M. P., Do E. Y.-L., Gross M. D.: Posey: Instrumenting a poseable hub and strut construction toy. In *Proc. 2nd International Conference on Tangible and Embedded Interaction* (2008), pp. 39–46. 3

[WFBSM12] Walther-Franks B., Biermann F., Steenbergen N., Malaka R.: The animation loop station: Near real-time animation production. In *Proceedings of the International Conference on Entertainment Computing* (2012), pp. 469–472. 1

[WIA*04] Watanabe R., Itoh Y., Asai M., Kitamura Y., Kishino F., Kikuchi H.: The soul of ActiveCube: Implementing a flexible, multimodal, three-dimensional spatial tangible interface. *Computers in Entertainment 2*, 4 (2004), 15–15. 3

[WLS13] Wu Y., Lu T., Song J.: A real-time mesh animation framework using Kinect. In *Proc. Pacific-Rim Conference on Multimedia* (2013), pp. 245–256. 1

[YSC*11] Yoshizaki W., Sugiura Y., Chiou A. C., Hashimoto S., Inami M., Igarashi T., Akazawa Y., Kawachi K., Kagami S., Mochimaru M.: An actuated physical puppet as an input device for controlling a digital manikin. In *Proc. Conference on Human Factors in Computing Systems* (2011), pp. 637–646. 1, 2, 3