POLITECNICO DI TORINO
Repository ISTITUZIONALE

Is immersive virtual reality the ultimate interface for 3D animators?

Original
Is immersive virtual reality the ultimate interface for 3D animators? / Lamberti, F., Cannavo', A., Montuschi, P.. - In: COMPUTER. - ISSN 0018-9162. - STAMPA. - 53:4(2020), pp. 36-45. [10.1109/MC.2019.2908871]

Availability:
This version is available at: 11583/2729746 since: 2020-04-13T18:02:35Z

Publisher:
IEEE

Published
DOI:10.1109/MC.2019.2908871

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

19 November 2026

Is Immersive Virtual Reality the Ultimate Interface for 3D Animators?

Fabrizio Lamberti, *Senior Member, IEEE,* Alberto Cannavò, and Paolo Montuschi *Fellow, IEEE*

Abstract—Creating computer animations is a labor-intensive task that requires animators to deal with sophisticated interaction paradigms. Virtual Reality, which is becoming commonplace in many domains, recently started to be considered as being capable to ease also the animators' job, by improving their spatial understanding of the animated scene and providing them with interfaces characterized by higher affordances. However, current VR-based animation solutions only offer heterogeneous subsets of traditional tools' functionalities, and/or are not integrated into them. In this paper, we present an add-on for the well-known Blender animation suite that allows the animators to seamlessly switch between the native and an immersive VR-based interface, and to use the latter to perform a representative set of common animation tasks. A user study including both professional and non-professional users showed that VR can improve usability and reduce animation time while letting animators achieve quality results, though with differences from task to task.

Index Terms—Computer animation, 3D graphics, Virtual Reality, Blender, user interface, evaluation.

1 Introduction

Computer-generated animations are used today in an increasingly growing number of fields, from movies and video games to architectural and industrial design, product advertising, education, etc. [1], [2]. Despite technological advancements, their creation remains a very time-consuming and skill-intensive task [3].

Recently, developments in the area of Virtual Reality (VR) attracted the interest of many digital artists, filmmakers and storytellers, among others, who started to look at this technology not only as a medium for vehiculating their contents, but also as a way for creating them [4].

A key reason to consider VR concerns the sense of presence that it can transmit to animators, which can make them feel as being part of the virtual environment where the story actually takes place, thus boosting creativity and productivity [5]. Furthermore, VR lets animators benefit from navigation and manipulation interfaces that are natively 3D, which can offer higher affordances compared to 2D interaction paradigms based, e.g., on mouse & keyboard (especially when controlling spatial properties) [6]. Working in VR could be beneficial also for the creation of contents for immersive environments, which require, e.g., to consider concepts like peripheral vision and to address challenges such as presenting a story in which users are free move and look in any direction [5].

In this paper we aim to investigate, through both qualitative and quantitative observations, the impact of VR-based animation systems on animators' performance, on interfaces' usability and on the quality of produced contents.

We started by analyzing the features of latest VR-based animation tools, summarized in Section 2.

Then, by taking into account the major elements in the ACM's CS Body of Knowledge for Computer Animation [7], we identified major functionalities that would be appreciated by animators, namely, keyframing and performance animation on diverse objects' properties, forward and inverse kinematics as well as splines manipulation.

Finally, most of the existing solutions were developed as standalone applications preventing access to subtle details of the animations being created, we built a new tool as an add-on of the well-know, open source Blender suite (http://www.blender.org). In this way, core animation functionalities of the underlying software can be made accessible in VR by exploiting some of the effective interaction paradigms exploited in other VR-based solutions, but providing animators with a larger set of features and a deeper level of control onto them.

The add-on, named VR Blender, can be freely downloaded at https://github.com/grainsgroup/VR-Blender.git. Some clips recorded during experimental evaluation are available at https://goo.gl/EBnP5E.

Differently than with other solutions, the tool is not meant as a replacement of traditional animation software but, rather, as a way to possibly improve some (possibly, many) of the stages of the animation pipeline using VR: animators could start their work in Blender, continue it in VR and finalize it again in Blender (or other tools). Although not all the animation functionalities mentioned in [7] are accessible in VR yet, they could still be managed using the native mouse & keyboard-based Blender's interface. Lastly, since the traditional and the VR-based interfaces can be used at the same time, multiple users can observe and possibly work on the same scene in a collaborative way.

2 Related Studies

In the last few years, a discrete number of VR-based animation solutions have been developed, including both research

• *F. Lamberti, A. Cannavò and P. Montuschi are with the Dipartimento di Automatica e Informatica of Politecnico di Torino, 10129 Torino, Italy. E-mail: http://grains.polito.it/people.php*

Manuscript received XXX XX, XXXX; revised XXX XX, XXXX.

prototypes, like, e.g., AnimationVR [3] or Maestro [8], and commercial tools, such as Tvori (http://tvori.co), MARUI (http://www.marui-plugin.com), Mindshow (http://mindshow.com), etc.

Most of them target casual users and, therefore, (reasonably) lack many of the functionalities offered by professional animation suites. For instance, existing VR-based animation tools generally support only performance animation, a form of motion capture in which virtual objects' properties are controlled in real time using some input devices (often the VR controllers, though animators' head, hands, etc. may be also used) [9]. When keyframing (the other major approach to computer animation) is supported, animators are often allowed to set only the initial and final value of the intended virtual object's property to be animated (the "keyframes"); tools will then interpolate intermediate values, typically without letting animators intervene in this process.

Moreover, solutions developed so far generally lack an integration with common animation suites, making extremely hard to edit, export and re-use created contents.

Most importantly, solutions available today have not undergone extensive validations yet, capable to show which are the animation tasks that could actually benefit from the use of VR (because of, e.g., of the limited set of animation tasks or of user categories considered [4] [10]).

3 Proposed Design and Methodology

In this section, the design of the VR Blender add-on and its components are illustrated.

3.1 VR Blender

The add-on relies on the Virtual Reality Viewport library (http://github.com/dfelinto/virtual_reality_viewport) for visualizing the Blender's viewport in VR on a head-mounted display. New modules have been implemented to create VR-enabled interfaces for many common Blender's functionalities and to map interactions performed with the controllers on the underlying software.

Any element in the Blender's 3D View can be animated. Once modeling has been completed and the scene has been set up in terms of textures, lights, etc., the user can activate the VR modality. This modality can be used both to work on new animations and to edit existing ones. Animations created in VR could then be seamlessly edited using the native Blender's interface.

In this work, the HTC Vive (https://www.vive.com) VR system was used. The system tracks both user's hand and head/gaze movements in a room scale environment. User interaction passes through the following states:

- Idle;
- Selection;
- Interaction;
- Navigation.

In the Idle state, the user does not perform any specific operation on the interface. He or she can only move, e.g., to change his or her point of view on the virtual environment.

Selection is the state in which the user can select/change the element to interact with at a given time (an object, an armature's bone, a camera, a control point of a curve, a material, etc.). Element is selected by moving the controller on it and pressing the Grip button (on the controller's side). Since, selection is a frequent operation, the user can activate it with both the controllers, by selecting up to two elements (one per controller).

In the Interaction state, the user makes changes to the scene and provides inputs to the system with the aim, e.g., to modify the transformations applied to an object, set the value of a specific property, define a keyframe or interact with controls available in the virtual environment. This state is activated by pressing the Trigger (on the controllers' bottom side).

Finally, the Navigation state allows the user to apply transformations to the VR reference system, in order to reach positions in the virtual scene that could not be attained due to the limited size of the HTC Vive's tracked space. This state is activated/deactivated by pressing the Menu button on top of the right controller. More details on the controllers are available on the HTC Vive's website.

Fig. 1(a) reports a screenshot of the virtual environment in the Idle state. The figure shows both the elements of a sample scene and the controls of the devised VR interface that can be used to create the animation.

3.2 Components

3.2.1 Scene Elements

Elements that can be animated include unarticulated and articulated 3D objects, like, e.g., the motorcycle and the rigged robot character in Fig. 1(a) (in the latter case, each bone/end-effector in the object's armature can be individually selected and animated). Besides the main camera, the 3D scene can include also custom cameras that can be manipulated like any other object. For each object, various properties (including transformations, material features, etc.) can be animated separately, creating so called "actions", i.e., collection of frames defined for a given property or set of properties.

3.2.2 Controllers

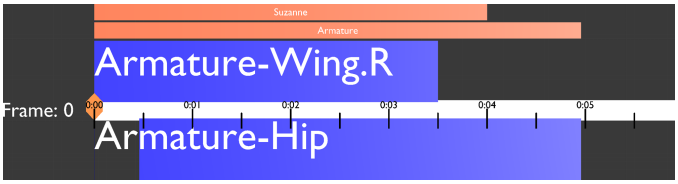
A virtual representation of the controllers allows the user to interact with the elements of the Blender's 3D View and access animation functionalities.

Text displayed close to each virtual controller reports the current state and the name of selected element(s), whereas textures shown on the Trackpads (the four-way interactive elements on the controllers' top side) are dynamically changed during animation to reflect functionalities that can be activated. A user manual extensively describing controller-based interactions is included in the software package (see link in Section 1).

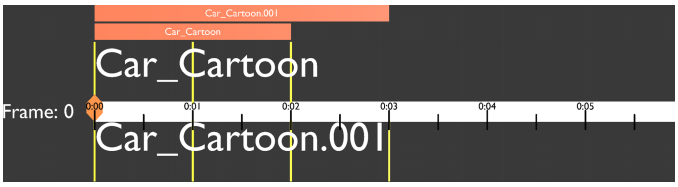
When the user selects a custom camera object, a panel is added below the object name to show the scene as seen from that camera, thus allowing the user to create camera animations from his or her own virtual point of view. With respect to other works where this "camera-in-hand" metaphor is used, here the user is allowed to choose when to switch cameras until rendering is performed.



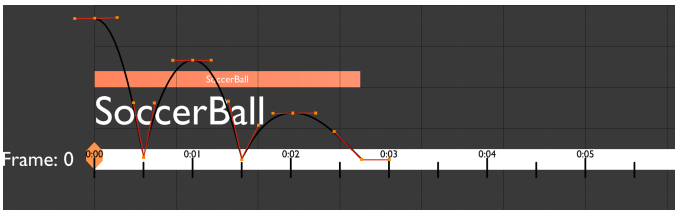
(a)



(b)



(c)



(d)

Fig. 1. Overview a) of the VR animation environment and b–d) of changes in the interface's aspect depending on current configuration; precisely, aspect of the timeline b) when two elements (two bones, in this case) belonging to an object named "Armature" have been selected for animation in Performance mode, c) when operating in Keyframing mode (with vertical bars corresponding to keyframes) and when the F-Curve tool is activated is shown.

3.2.3 Settings Panel

The panel, shown on left side of Fig. 1(a), is displayed when the user presses the Menu button on the left controller. Through this panel, the user can set up the main animation parameters and choose the actual animation mode and/or tool to work with.

So far, two modes have been implemented in VR, named Performance and Keyframing.

Performance mode allows the user to create animations in real time by transferring the movements of the controller(s) to selected objects' or bones' properties by automatically recording keyframes for them (two elements can be animated at a time). Properties to be animated are selected using the buttons under the Keyframe type label. Dedicated buttons allow the user to activate frequently used transformation keyframes (location, rotation and scaling). A separate button allows him or her to record keyframes for other elements' properties. So far, materials' diffuse color and transparency as well as shape keys' deformation intensity are managed, but it would be straightforward to include all the other properties that are supported by Blender by writing more scripts defining the mapping with some interface's widget or controller's button. The speed at which frames are recorded can be set using the Sampling rate control. Frames recorded during the user's performance are stored in a new action.

The Keyframing mode lets the user control the value of a given property at any frame (Blender then performs the interpolation). User can choose the properties to save keyframes for, and animate two elements at a time.

With the Record/Playback speed controls, the user can speed up/slow down the advancement of frames during animation playback/recording.

In both modes, several tools can be activated using dedicated buttons in the panel. The Edit action tool can be used to adjust the timing of the animation, by anticipating/delaying actions as well as shortening/extending their duration.

The Local representation tool implements the known "world in miniature" metaphor, which is particularly helpful in Performance mode. A duplicate of the selected element is automatically created and exploited as a proxy to interactively transfer to the original element the transformations applied by the user independent of his or her position and orientation in the virtual environment.

The Path tool allows the user to create a path curve for animating a given element and record/insert keyframes on the evaluation time.

Finally, the F-Curves tool allows the animator to manually adjust the interpolation curves (named F-Curves, in Blender) for keyframes set or recorded.

Modes and tools described above basically cover the animation techniques listed as topics in [7]. For instance, the Performance mode covers the "motion capture" topic, whereas the F-curves and Path tools cover the broad topic under the "splines" heading.

3.2.4 Timeline

This component is used to control the timing of the animation. As shown on the top-right of Fig. 1(a), it displays the duration of the whole animation (white bar), together with time information (seconds). Current frame is indicated by a text label, as well as by a cursor on the bar. The names of the elements selected with the left and right controller (if any) are shown above and below the bar (in white).

Scene elements' animations are shown using colored bars. There is one orange bar for every element for which an animation exists; the bar reports the element's name and indicates the frames at which animation starts/ends. For each orange bar there are one or more blue bars, each representing an action. Blue bars' length is proportional to actions' duration. Orange bars are always visible, whereas blue bars are displayed only for selected elements (either

above or below the white bar, depending on the controller used for the selection).

Like for the controllers, the timeline's aspect changes based on user's interactions. For instance, Fig. 1(b) shows how the timeline looks like when the user has selected two elements of the "Armature" object and is animating them in Performance mode. Action are shown, but keyframes that have been automatically recorded are not visible. This visualization is used also when the Edit action tool is activated, letting the user drag and scale the actions. In Keyframing mode, the timeline switches to a more detailed visualization in which keyframes are shown (Fig. 1(c)). When the user activates the F-Curves tool, aspect changes again to let the user adjust the interpolation curves via interactive handles, like in Blender (Fig. 1(d)).

4 Experimental Setup

In order to investigate the added value possibly deriving from the use of VR for executing common animation tasks, we carried out a user study involving students and academic staff of our university.

4.1 Participants

The 27 volunteers (19 males and 8 females), aged between 21 and 43, were categorized into two groups, depending on their experience with computer animation. In particular, 15 participants could be considered as non-professional users (NPUs), since they were students attending a course on computer animation with Blender. The remaining 12 participants could be considered as professional users (PRUs), because of their long-standing expertise in teaching similar courses or because working in the field. PRUs also had, on average, a much higher experience with VR.

4.2 Tasks

Each participant was asked to perform five representative animation tasks using both the native Blender's interface (BNI) and the proposed VR-based interface (VRI) with breaks in between. Tasks, illustrated in Fig. 2, were designed to make the participants experiment with all the implemented functionalities. Tasks and interfaces were tested in a random order, by balancing the coverage of all the configurations to limit learning effects. In all the tasks, participants had to work on a given object (colored green) and make it recreate the target/reference animation (shown in red). In some of the tasks, a spatial offset was set between the given object and the target object, in order to make it easier for the participant to see the scene's contents.

- Posing task (*posing.mp4*): in this task, users had to articulate the armature of a crocodile character including 17 bones and 51 degrees of freedom using both forward and inverse kinematics to make it match as much as possible the reference pose (Fig. 2(a)).
- Keyframing task (*keyframing.mp4*): the goal was to animate the car object in Fig. 2(b) by setting three transformation keyframes recording location and orientation and three keyframes to manage diffuse color's transition between a dark and a light shade and vice-versa.

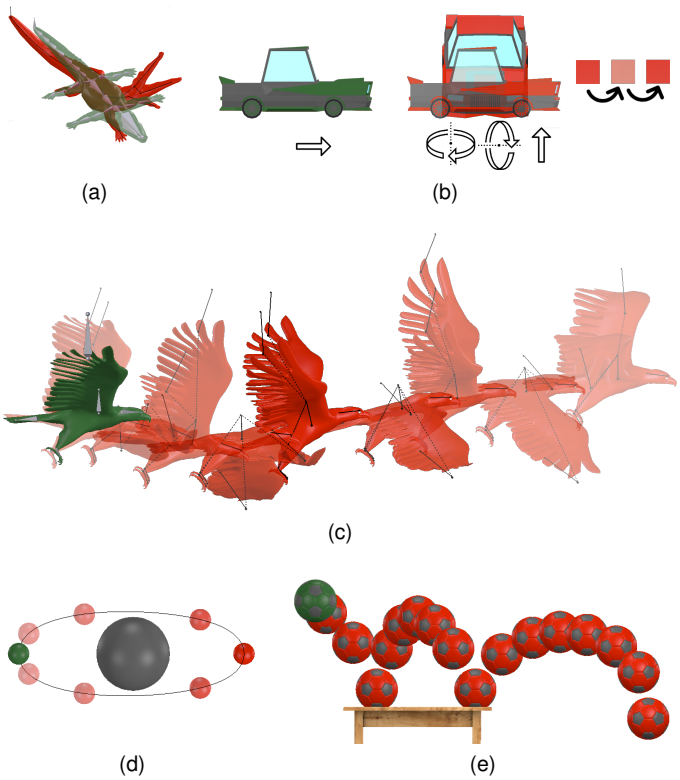


Fig. 2. Tasks considered in the experiments: a) posing, b) keyframing, c) performance, d) path, and e) interpolation curve editing tasks.

- Performance task (*performance.mp4*): the goal was to recreate as faithfully as possible the motion of the eagle character in Fig. 2(c). Character was controlled by three bones: two bones for articulating the wings using inverse kinematics, one bone for controlling location and orientation.
- Path task (*path.mp4*): the objective was to create the ellipsoidal path shown in Fig. 2(d) and animate planet's revolution by setting two keyframes for the evaluation time.
- Interpolation curve editing task (*interpolation.mp4*): in this task, a bouncing ball had to be animated by setting six location keyframes and correcting the interpolation curves using the F-Curves tool to sharpen the bounce on the floor (Fig. 2(e)).

Before starting the experiments, participants were introduced to the BNI and the VRI, focusing on functionalities needed for the completion of each task. Afterwards, they were given time to familiarize with the two interfaces for 5–10 minutes by experimenting with the same operations to be carried out in the given task, but with different scene elements. Since all the participants already knew the BNI but were new to the VRI, they were allowed to ask for help during the execution of the tasks with both the interfaces. No time limit was set.

4.3 Evaluation Criteria

Evaluation considered both objective and subjective measurements.

For objective measurements, two metrics exploited in previous works were used [11]. The first metric, named *completion time*, corresponds to the time needed to carry out the task. The second metric, named *animation accuracy*, measures the difference between the reference and the user-generated animations in terms of Euclidean (for the position) and angular (for the orientation) distances averaged on scene elements and frames.

Subjective measurements were collected by asking participants to fill in an after-test questionnaire split in two sections (https://goo.gl/hhFCyi). The first section evaluated the overall usability of the two interfaces based on the System Usability Scale (SUS) [12]. The second section assessed users' satisfaction in using the two interfaces based on questions defined in a previous work on 3D animation with non-traditional interfaces [10]. Participants' preference for the two interfaces in the execution of each task and overall was also collected.

5 Results

Results obtained by applying the above criteria were exploited to compare BNI and VRI performance.

5.1 Objective Results

Measurements concerning completion time and animation accuracy for the two user categories are reported in Fig. 3(a)–(b) and Fig. 3(c)–(d), respectively. Paired sample t-tests with 5% significance were used for the analysis (all the assumptions for t-tests were met). Significant results are marked with *.

By focusing first on NPUs, it can be observed that the VRI brought important benefits in terms of completion time in all the tasks. Compared to the BNI, with the VRI users were, on average, 35% faster ($p = 0.0001$). In particular, speedup was 24% in the posing task ($p = 0.0045$), 20% in the keyframing task ($p = 0.0336$), 51% in the performance task ($p = 0.0259$), 48% in the path task ($p = 0.0002$), and 52% faster in the interpolation curve editing task ($p = 0.0001$). Results were all statistically significant. There were no significant differences in terms of accuracy between VRI and NBI, which could indicate that the two interfaces allowed participants to create animations with comparable quality. Data trended toward a higher accuracy with VRI, suggesting perhaps that with a larger sample statistical significance might be found.

Concerning PRUs, results for completion time confirm the above trends. Compared to the BNI, with the VRI users were, on average, 37% faster ($p = 0.0001$). Speedup was 37% in the posing task ($p = 0.0017$), 29% in the keyframing task ($p = 0,0024$), 54% in the performance task ($p = 0.0001$), 36% in the path task ($p = 0.0005$), and 33% in the last task ($p = 0.0008$). Like for NPUs, differences in accuracy were not statistically significant, which could be regarded as a further indication of a comparable animation quality. This result is particularly relevant, since PRUs already knew the BNI, but they had no expertise with the VRI. The reason could be a higher learnability of the VRI.

Further insights can be obtained by comparing the two user categories (using unpaired t-tests). Concerning completion time, it can be observed that with the BNI, PRUs were significantly slower than NPUs in the posing task ($p = 0.0042$), since they spent a lot of time in refining all the armature's DOFs. The opposite result was obtained in the interpolation curve editing task ($p = 0.0001$), which is probably characterized by a far less intuitive interface. No other statistical differences were found. W.r.t. accuracy, by aggregating results for the two interfaces PRUs were found to be more accurate than NPUs, overall ($p = 0.0001$). Similar considerations hold also for individual tasks. When analyzing the interfaces separately, PRUs proved to be much more accurate than NPUs with the BNI, probably thanks to the higher experience with that interface. Interestingly, no statistical differences were found for the VRI, and average accuracy obtained by NPUs with the VRI was close to that achieved by PRUs with the NBI. These findings may indicate that VR-based systems like this one could be particularly effective in leveling animation skills among different user groups.

5.2 Subjective Results

Questions in the first section of the questionnaire asked users to evaluate usability based on the SUS scale [12]. Normalizing scores on a 0–100 scale, the VRI was judged as characterized by a higher usability than the BNI by both the user categories (mean 75.91, SD 1.28 vs 58.81, SD 2.33 for NPUs, mean 75.00, SD 1.10 vs 53.43, SD 2.03 for PRUs). As expected, there was no marked difference between the scores of the two user categories for the VRI, whereas the greater experience with Blender made the PRUs assign to the BNI a usability higher than that assigned by NPUs.

More insights about these results were obtained by analyzing scores assigned to individual SUS statements (not reported for brevity). Focusing on statistically significant results, the VRI was perceived as easier to learn than the BNI by both the user categories. The two categories also found the VRI as characterized by a more appropriate level of complexity than the BNI, making it easier to use. Overall, the introduction of VR for animation was appreciated by all the users, since they would like to (frequently) use the VRI more than the BNI. Lastly, NPUs felt more confident in using the VRI than the BNI; difference was significant for PRUs, though data trended toward VRI also in this case. The second section of the questionnaire requested participants to express their satisfaction in carrying out the animation tasks with the two interfaces by rating a set of items derived from [10]. Detailed ratings are reported in Fig. 3(e) and Fig. 3(f). Results confirm findings obtained with the first section. Focusing on statistically significant results (*), the VRI was perceived as easier to learn ($p = 0.0001$ for NPUs and $p = 0.0001$ for PRUs) and to use ($p = 0.0001$ for NPUs and $p = 0.0027$ for PRUs). This result suggests that the devised VR-based interface is suitable for users with different levels of expertise in the field of computer animation. Both the user categories expressed a higher appreciation for the VRI than for the BNI: the VRI was perceived as more wonderful ($p = 0.0001$ for NPUs and $p = 0.0022$ for PRUs), satisfying ($p = 0.0001$ for NPUs and $p = 0.0011$ for PRUs) and stimulating ($p = 0.0001$ for NPUs and $p = 0.0014$ for PRUs). No significant differences were found between the two interfaces concerning flexibility and perceived opera-

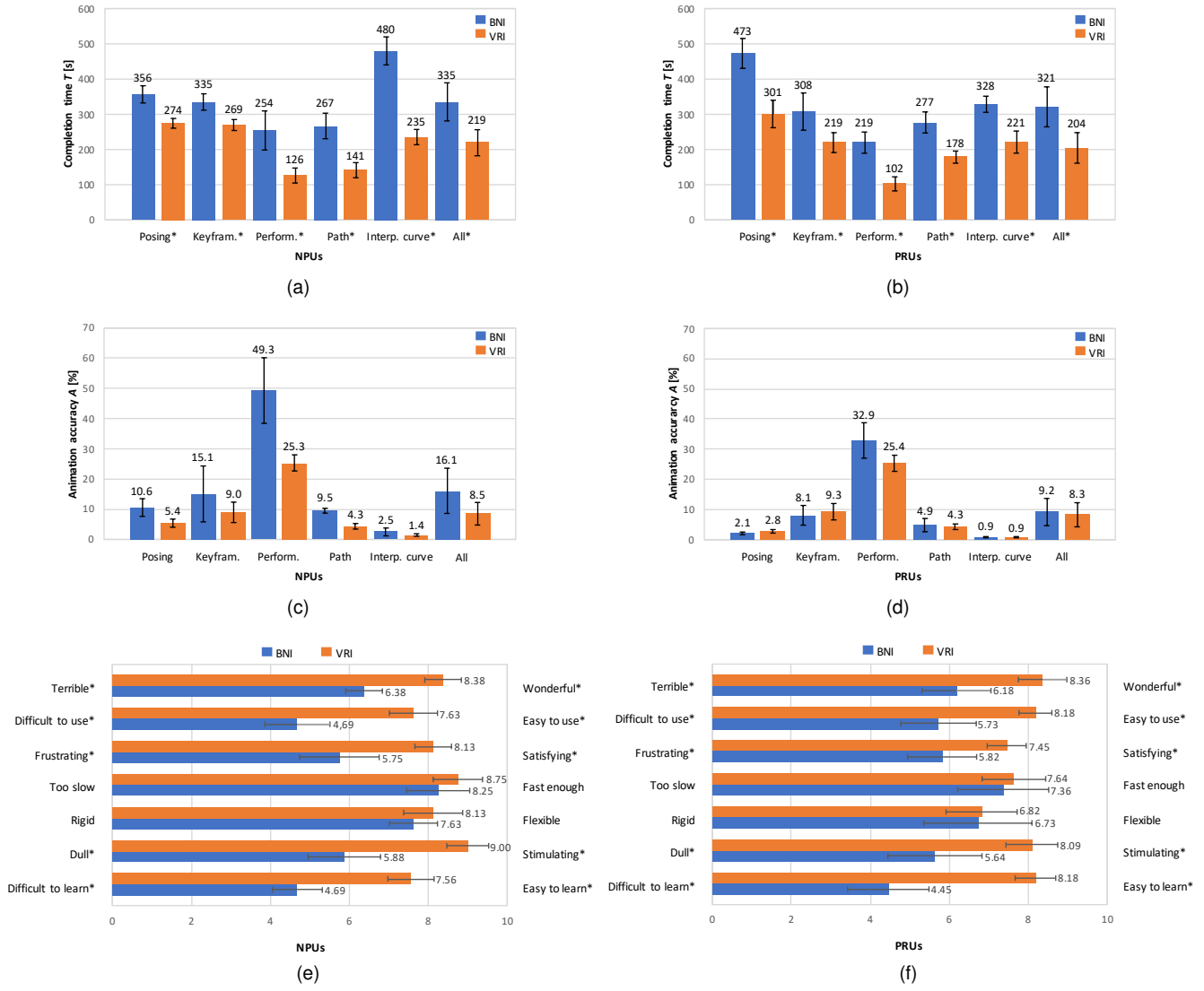


Fig. 3. Objective results in terms of a)–b) completion time and c)–d) animation accuracy, and e)–f) subjective results concerning satisfaction based on criteria (items) in [10] for non-professional and professional users, respectively. Average values (glyphs' height) and SDs (error bars) are shown.

TABLE 1
Users' preference for individual tasks.

Tasks	NPUs		PRUs	
	BNI	VRI	BNI	VRI
Posing	0%	**100%**	19%	**81%**
Performance	25%	**75%**	10%	**90%**
Keyframing	32%	**68%**	45%	**55%**
Path	12%	**88%**	45%	**55%**
Interpolation curve editing	12%	**88%**	36%	**64%**
All	0%	**100%**	45%	**55%**

tion speed (although, based on objective results, VRI was significantly faster than BNI).

Finally, participants were asked to express their preference for the two interfaces in the execution of each task (Table 1). Overall, NPUs preferred the VRI to the NBI. In particular, NPUs strongly preferred the VRI for all the tasks. Concerning PRUs, the VRI was still preferred to the NBI, overall, but the distance between the two interfaces was smaller. VRI was largely preferred for the posing, performance animation and interpolation curve editing tasks, whereas preferences were comparable for the keyframing and path tasks.

Comments gathered during the experiments indicate that findings above are strongly related to design choices made in interaction design, that is, to the way the various features can be accessed/controlled. For instance, participants stated that it would be helpful to let users configure functionalities mapped on the controllers (details on how they were set up in current implementation are provided in the user manual included with the software). They also lamented the lack of mechanisms in VR for quickly activating features that, in Blender, are generally accessed via keyboard shortcuts; several participants suggested to possibly explore the use of voice commands, which are regarded as more suitable than palette- or panel-based approaches used in this and other tools. Lastly, participants asked for the introduction of more effective ways to provide interaction feedback, e.g., to inform the user when specific operations are allowed/not allowed or a particular interaction is re-

quested (at present, changes' in virtual controllers' textures are basically exploited).

6 Conclusions

With this work we wanted to evaluate, in both objective and subjective terms, possible advantages and disadvantages brought by the introduction of VR in a computer animation pipeline, with the ultimate goal of determining whether this technology could be appropriate to replace mainstream interfaces. By developing the VR-based system that was used in the experimental evaluation as an add-on of an existing animation suite, we were allowed to directly compare the proposed interface with the native one on the execution of five common, representative tasks.

Results obtained showed that, by working with the controllers in an immersive virtual environment rather than with mouse & keyboard in front of a screen, both PRUs and NPUs were able to reduce the time required to complete the tasks, with no statistically significant difference in the produced animations. Overall, users were more satisfied with the VR-based interface rather than with the native one and considered the former as characterized by a higher usability, as they found it as easier to learn and operate but also as more stimulating.

Another important outcome of this work is that it provided hints that could be exploited to help users determine, based on considered metrics, which are the animation tasks that could benefit more from a VR-based interface and those for which VR technology may not be much beneficial. With current implementation, VR-based interface was largely preferred to mouse & keyboard by NPUs in all the tasks. However, for PRUs preference was not as sharp, as there were specific tasks for which interfaces were rated as almost comparable.

Although those above could be regarded already as very relevant results, we believe that identifying ways to improve the user experience for all the tasks is probably even more important. In fact, we feel that the greatest advantage of the proposed approach lays in the potential to let the users carry out most of (if not all) the animation tasks by working within the same, integrated environment. This is also the reason why we decided to release our software as open source. Thanks to the contribution of the active Blender's community, we can expect other features that are usually accessed through the mouse & keyboard interface to be made available in VR in the future by using ever more effective interaction paradigms.

Acknowledgments

This work has been supported by the VR@Polito initiative.

References

[1] M. Zyda, "From visual simulation to virtual reality to games," *Computer*, vol. 38:9, pp. 25–32, 2005.

[2] Y. Usui, K. Sato, S. Watabe, B. S. Santos, and G. Alford, "Computer graphics animation for objective self-evaluation," *IEEE Computer Graphics and Applications*, vol. 37:6, pp 5–9, 2017.

[3] D. Vogel, P. Lubos, and F. Steinicke, "AnimationVR – Interactive controller-based animating in virtual reality," in *Proc. IEEE Conference on Virtual Reality and 3D User Interfaces*, 2018.

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

[5] R. Henrikson, B. Araujo, F. Chevalier, K. Singh, and R. Balakrishnan, "Multi-device storyboards for cinematic narratives in VR," In *Proc. 29th Symposium on User Interface Software and Technology*, pp. 787–796, 2016.

[6] F. Lamberti, G. Paravati, V. Gatteschi, A. Cannavò, and P. Montuschi, "Virtual character animation based on affordable motion capture and reconfigurable tangible interfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24:5, pp. 1742–1755, 2018.

[7] B. M. Lunt, B. M. Ekstrom, S. Gorka, G. Hislop, R. Kamali, E. Lawson, R. LeBlanc, J. Miller, and H. Reichgelt, "Curriculum guidelines for undergraduate degree programs in information technology," ACM Technical Report, 2008.

[8] B. Berford, C. Diaz-Padron, T. Kaleas, I. Oz, and D. Penney, "Building an animation pipeline for VR stories," In *ACM SIGGRAPH 2017 Talks*, pp. 64–64, 2017.

[9] F. Andreas, J. Mller, and D. Lindlbauer, "Creature teacher: A performance-based animation system for creating cyclic movements," In *Proc. 3rd ACM Symposium on Spatial User Interaction*, 2015.

[10] P. Natapon, "A tangible interface for 3D character animation using augmented reality technology," In *Proc. 8th International Conference on Information Technology and Electrical Engineering*, 2016.

[11] A. Cannavò, and F. Lamberti, "A virtual character animation system based on reconfigurable tangible user interfaces and immersive virtual reality," In *Proc. Smart Tools and Applications in Graphics*, 2018.

[12] J. Brooke, "SUS-A quick and dirty usability scale," *Usability Evaluation in Industry* vol. 189:194, pp. 4–7, 1996.