ToolY: An approach for the combination of motion capture and 3D reconstruction to present tool usage in 3D environments

Evropi Stefanidi[1,2], Nikolaos Partarakis[1] Xenophon Zabulis[1], Paul Zikas[1], George Papagiannakis[1,3] and Nadia Magnenat Thalmann[2,4]

[1] Institute of Computer Science, Foundation for Research and Technology – Hellas (FORTH), Heraklion, Greece
[2] MIRALab, University of Geneva, Geneva, Switzerland
[3] Department of Computer Science, University of Crete, Heraklion, Greece
[4] Institute of Media Innovation, Nanyang Technological University, Singapore
evropi@{ics.forth.gr, miralab.ch}, {partarak, zabulis}@ics.forth.gr, papagian@{ics.forth.gr, csd.uoc.gr} zikas@csd.uoc.gr, thalmann@miralab.ch, nadiathalmann@ntu.edu.sg

Abstract. Visualization techniques for usage of tools, handicrafts and assembly operations are employed for demonstrating processes (e.g. assembly instructions). Until today, most commonly used techniques include written information, sketches in manuals, video instructions, etc. The evolution of technology nowadays has generated mature methods for transforming movement to digital information that can be processed and replicated. Motion capture together with 3D reconstruction techniques can provide new ways of digitizing handicrafts. At the same time, Virtual Humans can be used to present craft processes, as well as to demonstrate the usage of tools. For this, the tools utilized in these processes need to be transformed to the digital world. In this paper, we present ToolY, a 3D authoring platform for tool usage presentation in 3D environments, in order to demonstrate simple operations (e.g. usage of hammer, scissors, screwdriver), where the tools are the product of 3D reconstruction. The movement of the Virtual Humans derives from motion capture, while for the one of the tools, a novel approach is used, for inducing the tool motion from the human motion capture. The products of ToolY are Virtual Environments that can be experienced in 3D or through immersion in Virtual Reality.

Keywords: Tool Usage Visualization, Motion Capture, 3D reconstruction, Virtual Humans, Virtual Reality.

1 Introduction

“A picture is worth a thousand words” [1]. It is a known fact that humans like visualization, and that the human mind understands concepts easier when text is accompanied by some sort of visual medium [2]. To date, most handicraft and assembly operation tutorials consist - mainly - of text, as well as demonstrative pictures, with pointing arrows aiming to direct the reader, but how many people dread the thought of having to assemble furniture themselves? Or how many simple handicraft tasks are we afraid to perform, for fear of doing it incorrectly, hurting ourselves, or simply because we think it will be a difficult endeavor? Would it not be helpful to have something more than just text and pictures for help? Naturally, one can argue that countless video tutorials exist on the Internet, for various handicrafts or processes that involves the use of tools. Handicrafts have been namely captured in videos as well as documentaries. But how can we ameliorate, to demonstrate processes by enabling immersion and thus more efficient education of it?

At the same time, Virtual Reality (VR) has been proven an effective learning technique, by immersing the user into the virtual world [3]. Benefits of learning in a Virtual Environment (VE) include: (i) experiencing physical properties such as shape and size directly, (ii) the ability to change point of view to access new or unusual perspectives [4], (iii) it is a low-cost alternative to creating full-scale physical training scenarios, (iv) training scenarios that can be run repeatedly can be simulated, and (v) monitoring of progress during training sessions can be included to evaluate learners' skills [5].

Apart from Virtual Reality, the evolution of technology has led to methods that successfully transform movement to digital information that can then be processed and replicated. For this, technologies such as Motion Capture (MoCap) can be utilized. Furthermore, 3D reconstruction can transform physical objects into their digital representation, which along with Virtual Humans (VHs), can be used to demonstrate processes that involve tools. These two methods together (MoCap and 3D reconstruction) can produce digitization, not only of the human motion while performing the task, but also of the tool being used. The movement of the practitioner can be recorded by MoCap techniques such as IMU based or Visual Tracking methods. Although the objects, i.e. the tools being used in the process, as well as the environment in which the handicraft is performed, can be digitized through 3D reconstruction, MoCap cannot capture the actual movement of the tools easily (e.g. in the case of deformable tools). Adding to this issue is the fact that these tools, as well as the Avatars representing the VHs, and the scenes representing the surrounding environment, may be captured with different technologies. Thus, a platform is needed, to integrate all these technologies and content, in order to create engaging demonstration and training experiences.

In this paper we propose a solution, by presenting a platform for authoring tool usage experiences in 3D environments, regarding simple operations, which combines motion capture and 3D reconstruction. These tasks are presented by the VHs, whose movement derives from the MoCap, while for the movement of the tools, a novel approach is used, for inducing the tool motion from the human MoCap. In this chapter we will focus only on handheld tools e.g. hammer, screwdriver, scissors etc., which are free to move in any direction. Some important issues that need to be taken into account regard the grip of the tools by the VHs, the realistic motion of the tool, as well as multiplexing MoCap powered motion with other animations of the VH (e.g. when the VH is moving and hammering at the same time). Our platform is the editor, where the user can create these experiences, and then deploy them, for use by anyone who wants to experience the usage of a tool in a process that concerns handicraft. In this research work the Unity 3D game engine is used as the main development platform, which simplifies the loading of different 3D model formats. This is considered important as MoCap data, 3D digitization, Avatars, 3D models of the environment, etc. are produced by different technologies and tools. In this context it is important to maximize compatibility so as to be able to easily integrate content from multiple sources without increasing the complexity of the produced technology.

2 Related Work

2.1 Key technologies

Motion Capture technologies

Motion Capture (or MoCap) technologies measure the motion of subjects in three dimensions, based on wearable markers whose location in 3D is estimated by corresponding sensors. There are two main technologies that are used, optical based MoCap, and inertial measurement units (IMU) MoCap. Unlike normal video, MoCap directly measures positional and orientational components on human motion. The results accurately encapsulate human motion in 3D and, therefore, provide a comprehensive representation of the recorded motion.

MoCap technologies provide a complete 3D area where only the marked objects are tracked while ignoring the rest of the scenery. In optical-based systems it is possible to have unobstructed view of a large area, if multiple cameras are used in a structured environment without many occlusions (typically a studio). The IMU MoCap systems are a bit different in the sense that they do not directly measure position and displacement, but acceleration. Each IMU is comprised by an accelerometer, a magnetometer, and gyroscope [6] [7] that provide acceleration measurements in 3D with respect to Earth's magnetic field. Although they provide more indirect measurements and are sensitive to magnetic interference, they can provide practical solutions in environments with multiple occlusions, such as in a craft workshop, because the wearable markers are the sensors as well. Thereby no structured environment is required. Visual Tracking technologies use camera-type sensors to record a subject's motion, and for this reason, are quite unobtrusive. These sensors are typically video cameras (RGB sensor), possibly with the addition of a depth camera[1] (RGBD sensor). Motion is estimated in 3D, by processing the visual stream, distinguishing the subject from the rest of the imaged scene, and by fitting in a 3D deformable model into the acquired visual data. The cost of the unobtrusive nature of these methods is the treatment of occlusions, and the inference of subject motion that is not imaged, due to these occlusions. These methods have been the focus of multiple works in Computer Vision and considerable progress has already been achieved in the last decade.

---

[1] Depth cameras are standard RGB cameras that use a separate sensor that emits infrared (IR) light in a specific pattern and reads the deformation as it is being reflected back, producing a depth image that shows the distance of surfaces imaged by the RGB camera from the depth sensor.

Regardless of the technology used to acquire the recordings, the resulting data are always a chain of coordinate frames and the difference in position and orientation between them. In our context, we follow a generic approach, meaning we can support the results of any motion capture technology. This is of great value, as any visualization of movement comes from an abstraction of the movements, which can then be used in various domains, ranging from cartoon movement to instructions on how to assemble furniture.

3D reconstruction for digitization

3D reconstruction is the process of capturing the shape and appearance of real objects. The capabilities of the different technologies vary in terms of several criteria which must be considered and balanced when formulating appropriate campaign strategies. The 3D reconstruction of physical objects has been improved with laser scanning and photogrammetry, digitizing tangible artifacts for cultural heritage and archeological sites. Each methodology proposes a different approach to face the challenge of digitization for visualization or preservation purposes. [8] compared the latest software and hardware techniques for rapid reconstruction of real humans using RGB and RGBD cameras suitable for Virtual, Augmented (AR) or Mixed Reality (MR) experiences. In the context of our approach, we can use any 3D reconstruction that results in a correct, 3D model of a tool. We are namely not bound by any restrictions regarding which 3D reconstruction technique will be used.

2.2 Authoring tools for experiences and tool usage demonstration

An authoring tool encapsulates various functionalities and features for the development of a specific product. The software architecture of such a system empowers users/programmers with the necessary tools for content creation to a) support users with intuitive and easy to use methodologies (visual scripting, editors) and b) provide advanced users with enhanced tools to extend the capabilities of the system.

M.A.G.E.S ™ [9] is a novel platform which also facilitates tool usage demonstration. This platform proposed a novel VR SDK to deliver a psychomotor VR surgical training solution. The system generates a fail-safe environment for surgeons, residents and other technical occupations to master their skills and track their abilities. The platform supports a visual scripting editor, scene customization plugins, custom VR software patterns and Unity editor tools for rapid prototyping of VR training scenarios. In addition, the M.A.G.E.S ™ platform offers ToolManager, a unique plugin designed for usage and manipulation of tools in VR environments. Utilizing ToolManager, a developer can transform any 3D model of a tool (pliers, hammer, scalpel, drills etc.) into a fully functional and interactive asset, ready to use in VR applications. After the tool generation, users can interact with it in the VE and use it to complete specific tasks following recorded directions. ToolManager supports additional features, among others the re-initialization of the tool if user drops it accidentally, sound effects and animations for electric tools and finally flashing visual indicators to inform user which tool to select to complete a task among others.

Recent approaches focus on the creation of tools to enhance the development with additional features. In more detail, ExProtoVAR [10] is a lightweight tool to generate interactive experiences in AR, for designers and non-programers who do not have much technical background in AR interfaces. ARTIST [11] is a platform featuring methods and tools for real-time interaction between non-human and human characters to generate reusable, low cost and optimized MR experiences. This project proposes a code-free development environment for the deployment and implementation of MR applications, while using semantically data from heterogeneous resources. RadEd [12] features a new web-based teaching framework with an embedded smart editor to create case-based exercises for image interaction, such as attaching labels and selecting specific parts of the image and taking measurements. It is an assistive tool for complex training courses like radiology.

In the context of this research work, an alternative approach is presented that allows the inference of simple tools usage based on the postures obtained from MoCap data. Using this approach, the 3D model of a tool can be attached to the Avatar representing the VH in the VE, by applying suitable rotation and translation operations.

2.3 Embodied Agents - Virtual Humans

Virtual characters, and in particular VHs, constitute an important aspect of 3D applications, due to our familiarity with a human-like individual. They are mainly used as narrators [13], virtual audiences [14] and in our case demonstrators of tool usage. We can distinguish two different styles in the representation of VHs: a) human-like and b) cartoon style Avatars, each one serving a different purpose and fitting into specific applications.

In the context of VEs, VHs have already been utilized for explaining physical and procedural human tasks [15], simulating dangerous situations [16], group and crowd behavior [17], and assisting users during navigation, both by showing where relevant objects/places are, and by providing users with additional information [18]. Moreover, VHs offer a possible solution to the problem of unstructuredness [19], which requires the user to take the initiative both in exploring the environment and interacting with its parts. This lack of proper assistance clashes with the traditional learning scenario, where a real teacher structures the presentation of material and learning activities [20]. VHs can provide a solution by acting as an embodied teacher. While it is not feasible to provide a human tutor for every learner in the real world, embodied agents can aid anyone with access to a computer, enabling an individualized instruction for a massive number of learners.

Regarding the different styles of VHs, human-like avatars are more closely related to users due to their natural appearance. However, imperfect human-likeness can provoke dislike or strangely familiar feelings of eeriness and revulsion in observers, a phenomenon known as the uncanny valley [21]. The same principle is applied not only in the visual representation of a VH, but also in the way of moving in 3D space. Studies suggested that interaction and animation can overcome the valley in affinity due to matching and common human non-verbal cues [22]. VHs are usually more complex to animate since they consist of humanoid skeletons with more joints than cartoon Avatars. The animation process needs to be precise and with high realism to avoid the uncanny valley effect. In contrast, cartoon style Avatars represent a simplified version of human characters, leading to low complexity in generation and animation processes. Cartoon Avatars are mainly used in applications designed with a similar cartoony style to match their environment. However, this design style reflects limitations on the usage of cartoon Avatars in realistic environments, making realistic VHs ideal for such scenarios.

In our platform we decided to use realistic, human-like VHs for a number of reasons. First of all, tools are designed to be handled by humans, thus making the human hand ideal to interact with them in a natural way. Most cartoon style Avatars do not have five fingers to avoid the uncanny valley, resulting in poor and unrealistic handling capabilities with tools. Additionally, we wanted to develop a realistic representation of tool usage, i.e. a simulation for interacting with tools in 3D environments; thus, utilizing human-like Avatars was the preferred choice.

3 Our pipeline: from motion capture to motion visualization

This section presents our pipeline, from the MoCap of the human movement, to an authored 3D scene which includes the Avatar(s), the tools, and various 3D objects, as well as the motion of the Avatar and the tool they use. At the beginning, this scene is empty, and subsequently the Avatar, the tool, and finally the surrounding environment (room and 3D objects) are added. Furthermore, the Avatar may use different tools throughout an execution of a Motion Vocabulary Item (MVI), or interchange between tools. A MVI is used in the context of this research work to represent an instance of a movement that is encoded in a BVH file and can be used to represent a specific action or part of an action. MVIs can be combined and interleaved to represent entire procedures and are considered building blocks of a Motion Vocabulary (MV). The MV in turn can be used to create “sentences” that encode different actions and procedures. As a result, the MV can be used to encode a wider variation of actions and combinations of actions than the initial MoCap data used for its implementation. An overview of our pipeline is visible in Figure 1 below.

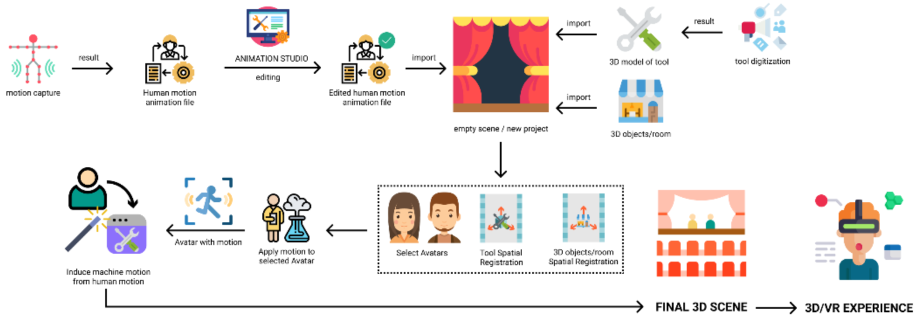

Fig. 1. Overview of our pipeline.

Step 1: Motion Capture

The first requirement is a MoCap file, which represents the human movement. As was analyzed in the Related Work section, there are various ways to procure a MoCap, and all of them can work in our context. For the purposes of testing TooltY, we used MoCap files that were the product of motion capturing with NANSENSE©[2] R2 motion capture suit in the context of motion capture sessions that took place during the second plenary meeting of the Mingei H2020 EU project at Haus der Seidenkultur in Krefeld Germany[3]. For the hammering activity presented in the ToolY demo we used an open BVH dataset from outworldz[4] and specifically package 60-75[5] file 62_08.bvh. The product is a BVH animation file, which is used in our editor.

Step 2: Tool Digitization

Another prerequisite is the digitized form of the tool(s) that will be used. For this, various approaches for 3D reconstruction for digitization can be used, as described in the corresponding section in the Related Work presentation. The result of the digitization is a 3D model that represents the surface geometry and appearance of the object and is, typically, encoded as a textured mesh of triangles (e.g., in VRML or FXB file format). While we can use the model of any digitized tool, some cases require post-processing. For instance, in the case of scissors, post processing is needed to make the tool deformable, in order to introduce joints that describe its deformation. In the context of experiments several 3D models were used coming both from 3D reconstructions of physical objects and from online sources[6]. For complementarity we used 3D models of both non-deformable (i.e. hammer) and deformable (i.e. scissors) tools.

Step 3: Editing of the MoCap (BVH file) in Animation Studio

For editing the MoCap animation files, we use a BHV editor, developed in-house, called Animation Studio[7]. Animation Studio allows visualization, editing, and annotation of 3D animation files, obtained by motion capture or visual tracking. In the case of visual tracking, temporally corresponding video can be also edited. The application allows the user to isolate animation segments and the associated video for further annotation, as well as the synthesis of composite animation files and videos from such segments. Pertinent annotation software exists in the linguistics domain [7], but does not stream for video and motion capture. Using Animation Studio, segments of the BVH input can be isolated and exported to test different "atomic" scenarios (e.g. hand movement when hammering a nail), simplify the input, or to allow the in depth analy-

[2] https://www.nansense.com/
[3] https://seidenkultur.de/
[4] https://www.outworldz.com/Secondlife/Posts/CMU/
[5] https://www.outworldz.com/Secondlife/Posts/CMU/cmuconvert-daz-60-75.zip
[6] www.thingiverse.com
[7] Presented in the public deliverable D1.3 "Scientific protocol for craft representation" of Mingei (European Union's H2020 research and innovation program under grant agreement No. 822336) to be published at Mingei's website www.mingei-project.eu after approval by EC

sis of certain scenarios. For the demo of TooltY, Animation Studio was used to create the MVIs for the case of operating a hammer and scissors.

Step 4: Start TooltY

When starting TooltY, the first thing to be done is to create a new project, which loads an empty scene. After editing the MoCap files, they can be imported to TooltY, along with the 3D models for tools and any desired objects. TooltY has some predefined ready-to-use content that users can browse through in their corresponding collections: Avatars, Human Motions, Tools, Rooms and 3D Objects. Additionally, users can choose to add their own files to any of these collections.

Step 5: Selecting an Avatar

The Avatar representing the VH plays a very important role in our concept, as it is the actor executing the movements representing the tool usage, thus bringing the whole process to life. The user can choose from a selection of available Avatars. TooltY is built using the Unity3D game engine, and thus Avatars created using a plethora of 3D Computer Graphics and Animation Creation editors can be imported (e.g. 3DStudioMax, Fusion 360, etc.). For the purposes of the TooltY demo presented by this research work, Poser Pro 11 [9] as well as Adobe Fuse [23] were employed for the creation of the two (2) Avatars visible in Figure 2. These avatars were exported from Poser and Fuse, and then imported to the developed platform as resources that can be selected and assigned to a simulation scenario. The user can also choose to upload their own Avatar to use as the VH, provided that it is rigged (i.e. it has a humanoid skeleton). After an Avatar for the VH has been selected, it can be added to the scene.

Step 6: Application of motion to the Avatar

After a suitable Avatar has been chosen, the user can apply on it a single or multiple human motion animations. Each animation is mapped through the correspondence of the joints, between the humanoid-type rig (skeleton) of the Avatar, and the joints of the humanoid skeleton resulting from the motion capture. The result of this process can be previewed both using the selected Avatar and in the form of a primitive Avatar animation. The latter is used to help users visualize animation problems (e.g. elbows are getting inside the body when movement is previewed using a certain avatar). This could mean for example, that the avatar's torso is too narrow. This is a known and well-studied problem in Computer Graphics, known as motion retargeting, and a lot of research work has focused on solutions, such as [24] [25] [26]. In the scope of this research work, we decided to solve the problem offline, by utilizing Unity's Avatar Muscle & Settings[8], for configuration of the degrees of freedom of joints in the skeleton of the Avatar. In more detail, it allows tweaking of the character's range of motion to ensure the character deforms in a convincing way, free from visual artifacts or self-overlaps. However, we would like to explore other possibilities for online motion

[8] https://docs.unity3d.com/Manual/MuscleDefinitions.html

retargeting in our future work, to minimize the manual and offline tweaking the users of TooltY are required to perform. An example of an Avatar operating a hammer can be seen below, in Figure 3.

Fig. 2. Screenshot of the 2 VHs inside TooltY, holding (a) a hammer and (b) scissors.

Fig. 3. Screenshots of a VH holding and operating a hammer.

Step 7: Introducing a tool

The user can add to the scene any digitized tool from the available collection of tools, corresponding to the selected human motion animation for the Avatar, as well as their own 3D models of tools.

Tool **o** is represented by a mesh of triangles, which encoded in 2 lists. The first list is the list of vertices, denoted by $\mathbf{l}_v$ and the second is the list of triangles, denoted by $\mathbf{l}_t$.

To add a tool, the following are required:

- A grip point $\mathbf{p_a}$ on the hand of the Avatar.
- A preferred grip on the tool, denoted as **g**.
- Two points on the front and back faces of the desired grip point of the tool, denoted as $\mathbf{p_f}$ and $\mathbf{p_b}$, respectively.
- A grip center $\mathbf{p_c}$ is automatically calculated as the mid-point of $\mathbf{p_f}$ and $\mathbf{p_b}$.
- The projection of $\mathbf{p_c}$ to the top side of the bounding box of the tool **o**, denoted as $\mathbf{p_g}$, also automatically calculated.

Regarding $\mathbf{p_a}$, **g**, $\mathbf{p_f}$ and $\mathbf{p_b}$, they are already calculated when referring to tools in the available collection, otherwise they need to be provided by the user adding their own 3D model.

Below, translations are encoded as 3x1 matrices and rotations as 3x3 rotation matrices.

The coordinate frame of the hand $\mathbf{C}_h$ is the coordinate system on the selected joint of the Avatar (in our case, its hand) at the current configuration of the avatar. This frame is determined at each moment in time, by the animation that the avatar executes.

A grip **g** is a configuration of the Avatar's hand that is represented by a series of rotations of the Avatar's joints. The number of rotations is determined by the skeleton of the Avatar and the values of these rotations by the animation that the avatar executes.

A tool posture is the rotation that orients the tool in the hand of the Avatar as intended by this posture. The posture is represented by rotation matrix **R**. Matrix **R** aligns the $\mathbf{C}_h$ with $\mathbf{C}_o$, by rotating the tool in-place, by **R**. The rotation center is $\mathbf{p}_c$, and R needs to be an "in place" rotation to avoid rotation about the world center. R is determined by the current orientation of the hand. It should be noted that we choose to rotate around $\mathbf{p}_c$, since that is the point from which the Avatar holds the tool.

Let $\mathbf{b}_o$ the bounding box of tool **o**. The user names the faces of $\mathbf{b}_o$ as front / back, top / bottom, and left / right. This determines the coordinate frame $\mathbf{C}_o$ for tool **o**. This frame is represented by rotation matrix $\mathbf{R}_o$ that aligns the $\mathbf{C}_o$ to the world coordinate frame. In the case of tools in the available collection, these have already been calculated, i.e. the user does not need to use the User Interface to name these points.

Typically, the Avatar and tool 3D model are not in the same scale, metric unit, and coordinate system. Tool **o** is brought to the correct scale by scaled by factor **s**; the scaled tool is denoted as $\mathbf{o}_s$. Let **s** the scalar that adjusts the scale and metric unit of the model.

To emulate the grip of a tool, grip **g** is applied to the avatar.

The transformation that brings the tool in the hand of the Avatar is as follows:

Let **x** a 3D point of **o**.

Let $\mathbf{T = p_a - p_c}$ the translation that brings $\mathbf{p_c}$ and $\mathbf{p_a}$ to coincidence.

The required (see above) in-place rotation is applied to **o** about $\mathbf{p_c}$. The operation required is $\mathbf{(R*(x\text{-}p_c)) + p_c}$, where '*' denotes matrix multiplication.

In the case of our demo in Unity, where the VH is operating a hammer, it is important to note that the default pivot is the center of the mesh of the object, and it is according to this point that any transformation is applied on the object. Therefore, in order to manipulate the object by $\mathbf{p_c}$, we created an intermediate (invisible) object, as a parent in the hierarchy of the tool, while preserving the world coordinates of the tool. Thus, transformation formulas are applied to the custom pivot point, instead of directly on the object.

Figure 4 below aims at explaining the aforementioned concepts, notations and operations.

Fig. 4. Display of grip points and resulting attachment of hammer to the VH hand.

Step 8: Addition and spatial registration of room/3D objects

The user can add to the scene 3D objects, either from the available collection, or by importing new ones. Such objects can either be craft workspace objects (i.e., pieces of furniture) or the room itself i.e. the 3D environment where the scene is set. As with the 3D models of the tools, the room and the objects imported may derive from heterogeneous sources and models, and thus spatial registration may need to take place, i.e. the 3D models may need to be scaled, rotated or translated.

Step 9: Tool manipulation from human motion

We do not have MoCap for the movement of the tools, so we induce the tool motion, from the human motion. We argue that it is easier and more cost-efficient to induce the tool motion, since the tool may not be rigid (e.g. scissors) or/and might not be usable if we add motion capture instrumentation on it. For example, if sensors for motion capturing are put on a scissor, it might not be possible for the operator to hold it and use it easily or as they would normally do. Our approach aims to solve such issues, by allowing generalization, as we can induce the motion of any simple handheld tool, by the one of the humans, deriving from the MoCap.

Step 10: Play the scene in 3D

After following these steps, the user can choose to play the scene, i.e. trigger the execution of the animations that comprise the MV. Thus, the VH comes to life, enacting the use of the tool, in the desired 3D environment. We argue that the proposed approach provides positive feedback regarding its validity and the possibility to further generalize and extend it. This could ultimately lead to an improved version of TooltY, where any handicraft process involving the manipulation of simple hand tools can be authored and visualized.

Step 11: Experience the scene in 3D or VR

After the 3D scene is ready, it can be deployed for use in 3D or VR, where users can have immersive experiences regarding the handicraft procedure in question. Namely, TooltY produces VEs that can be experienced in 3D or through immersion in VR, where the VHs demonstrates how to execute a handicraft, step by step. In more detail, users experiencing this can see the hands of the Avatar in VR, showing them the tools they need to use and how to operate them, to complete an action correctly.

4 Results and Contribution

The result of this research work is TooltY, an authoring platform which allows the authoring of tool-usage experiences. These scenes can subsequently be exported as 3D or VR immersive experiences, so that simple handicrafts can be demonstrated to the user.

Regarding the implementation of the authoring platform, it was developed in Unity, and presents the following main functionalities:

- Allows users to use existing Avatars available in the platform's collection, or upload new ones to the collection, for use as the VHs enacting the tool-using task.
- Allows users to use existing tools available in the platform's collection, or upload new ones to the collection, for the objects used in the handicraft they wish to demonstrate.
- Allows users to use existing rooms and objects available in the platform's collection, or upload new ones to the collection, for the room used as a background scene and the objects in this room.
- Allows users to use existing MoCap files for the VHs, or upload new ones to the collection, for the human motion.
- When uploading new content, users have to configure specific details:
  - Avatars need to have a humanoid skeleton (if they don't have one the user needs to add it),
  - For the tools, they need to define $\mathbf{p_a}$, $\mathbf{g}$, $\mathbf{p_f}$, $\mathbf{p_b}$, and the faces of the bounding box $\mathbf{b}_o$ as front / back, top / bottom, and left / right, as explained in Step 7 of the TooltY pipeline.
- Users can add Avatars and tools to the scene, link them, and select the suitable human motion for the VH, to enact the tool usage. The desired tool is correctly po-

sitioned in the hand of the Avatar, by performing translation and rotation operations, as described in Step 7 of the ToolTY pipeline.
- Users can add different background scenes (rooms) and objects to the scene.
- Users can export the final scene, for 3D and VR immersive experience of the tool usage.

To achieve high quality rendering for the displayed tools, we utilized global illumination as the main rendering pipeline. Our approach focuses on high quality visualization of used tools, aiming to deliver a realistic interaction with handheld tools, improving the visuals and providing a high-fidelity final product. We applied the same rendering techniques to the VHs, highlighting the interaction with the available tools.

Our methodology follows a modular pipeline, thus transferring the generated 3D application to a VR immersive experience is straightforward. The proposed solution contains a list of well-defined steps to generate an animated human-tool behavior. TooltY facilitates an authoring tool regardless of the deployed medium, since the same principles are applied for both 3D and VR applications. To prepare the generated scene for VR, minimal changes are needed including the initialization of a VR camera into the VE representing the headset. In addition, TooltY generates a fully personalized virtual world, giving users the ability to customize both the human character along with the tool, as well as the surrounding 3D environment. This customization process will improve the virtual scene, offering a realistic representation of the environment, ideal for VR integration.

5 Future Work

This paper has presented an authoring platform for combining motion capture with 3D reconstruction, to present tool usage in 3D environments, in contexts such as handicrafts and assembly operations. The usage of these tools is carried out by Virtual Humans, whose movements are the result of the MoCaps, using digitized tools from the 3D reconstruction. For the motion of the tools, a novel approach is used, for inducing the tool motion from the human motion capture.

Our future work includes (i) the further development of the current VR experience to incorporate a VR training module, so that users can be immersed in a learning environment, where the technique and usage of each tool can be demonstrated and subsequently executed by the user, including feedback (ii) exploring the effects the tool usage could have on the scene (e.g. how the hammer is deforming a nail), and (iii) the addition of a Storytelling module, where a second VH can explain the usage of the tool and its story, and optionally narrate its usage through history during the demonstration. For the orchestration of the narration, as well as the dialogue between the VHs, the CasandRA framework [27] will be used. Finally, we plan to conduct user evaluation experiments, to assess the usability of the authoring platform, as well as of the produced immersive module, and the ability of the users to learn how to perform a handicraft with its help.

6 Acknowledgments

The work reported in this paper is being conducted in the context of the European Union's Horizon 2020 research and innovation program under grant agreement No. 822336 (Mingei). The authors would like to thank Nikolaos Patsiouras for his contribution to the technical discussions and development of the TooltY demo.

References

1. A picture is worth a thousand words, https://en.wikipedia.org/w/index.php?title=A_picture_is_worth_a_thousand_words&oldid=909812483, (2019)
2. Dewan, P.: Words versus pictures: leveraging the research on visual communication. Partnership: the Canadian Journal of Library and Information Practice and Research. 10, (2015)
3. Hussein, M., Nätterdal, C.: The benefits of virtual reality in education-A comparision Study. (2015)
4. Ferrington, G., Loge, K.: Virtual reality: A new learning environment. Computing Teacher. 19, 16–19 (1992)
5. Ieronutti, L., Chittaro, L.: Employing virtual humans for education and training in X3D/VRML worlds. Computers & Education. 49, 93–109 (2007)
6. Bachmann, E.R., Yun, X., Peterson, C.W.: An investigation of the effects of magnetic variations on inertial/magnetic orientation sensors. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004. pp. 1115–1122. IEEE (2004)
7. Brigante, C.M., Abbate, N., Basile, A., Faulisi, A.C., Sessa, S.: Towards miniaturization of a MEMS-based wearable motion capture system. IEEE Transactions on industrial electronics. 58, 3234–3241 (2011)
8. Papaefthymiou, M., Kanakis, M.E., Geronikolakis, E., Nochos, A., Zikas, P., Papagiannakis, G.: Rapid Reconstruction and Simulation of Real Characters in Mixed Reality Environments. In: Digital Cultural Heritage. pp. 267–276. Springer (2018)
9. Papagiannakis, G., Lydatakis, N., Kateros, S., Georgiou, S., Zikas, P.: Transforming medical education and training with VR using MAGES. In: SIGGRAPH Asia 2018 Posters. p. 83. ACM (2018)
10. Pfeiffer-Leßmann, N., Pfeiffer, T.: ExProtoVAR: A Lightweight Tool for Experience-Focused Prototyping of Augmented Reality Applications Using Virtual Reality. In: International Conference on Human-Computer Interaction. pp. 311–318. Springer (2018)
11. Kotis, K.: ARTIST-a reAl-time low-effoRt mulTi-entity Interaction System for creaTing reusable and optimized MR experiences. Research Ideas and Outcomes. 5, e36464 (2019)
12. Xiberta, P., Boada, I.: A new e-learning platform for radiology education (RadEd). Computer methods and programs in biomedicine. 126, 63–75 (2016)
13. Zikas, P., Bachlitzanakis, V., Papaefthymiou, M., Kateros, S., Georgiou, S., Lydatakis, N., Papagiannakis, G.: Mixed reality serious games and gamification for smart education. In: European Conference on Games Based Learning. p. 805. Academic Conferences International Limited (2016)
14. Chollet, M., Chandrashekhar, N., Shapiro, A., Morency, L.-P., Scherer, S.: Manipulating the perception of virtual audiences using crowdsourced behaviors. In: International Conference on Intelligent Virtual Agents. pp. 164–174. Springer (2016)

15. Rickel, J., Johnson, W.L.: Animated agents for procedural training in virtual reality: Perception, cognition, and motor control. Applied artificial intelligence. 13, 343–382 (1999)
16. Traum, D., Rickel, J.: Embodied agents for multi-party dialogue in immersive virtual worlds. In: Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2. pp. 766–773. ACM (2002)
17. Paul, Z., Margarita, P., Vasilis, M., George, P.: Life-sized Group and Crowd simulation in Mobile AR. In: Proceedings of the 29th International Conference on Computer Animation and Social Agents. pp. 79–82. ACM (2016)
18. Chittaro, L., Ranon, R., Ieronutti, L.: Guiding visitors of Web3D worlds through automatically generated tours. In: Proceedings of the eighth international conference on 3D Web technology. pp. 27–38. ACM (2003)
19. Economou, D., Mitchell, W.L., Boyle, T.: Requirements elicitation for virtual actors in collaborative learning environments. Computers & Education. 34, 225–239 (2000)
20. Hertz-Lazarowitz, R., Shachar, H.: Teachers' verbal behaviour in cooperative and whole-class instruction. Cooperative learning: Theory and research. 77–94 (1990)
21. Mori, M.: Bukimi no tani [the uncanny valley]. Energy. 7, 33–35 (1970)
22. Seymour, M., Riemer, K., Kay, J.: Interactive Realistic Digital Avatars-Revisiting the Uncanny Valley. (2017)
23. Create 3D models, characters | Download Adobe Fuse (Beta), https://www.adobe.com/gr_en/products/fuse.html
24. Hecker, C., Raabe, B., Enslow, R.W., DeWeese, J., Maynard, J., van Prooijen, K.: Real-time motion retargeting to highly varied user-created morphologies. In: ACM Transactions on Graphics (TOG). p. 27. ACM (2008)
25. Dariush, B., Gienger, M., Arumbakkam, A., Goerick, C., Zhu, Y., Fujimura, K.: Online and markerless motion retargeting with kinematic constraints. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 191–198. IEEE (2008)
26. Monzani, J.-S., Baerlocher, P., Boulic, R., Thalmann, D.: Using an intermediate skeleton and inverse kinematics for motion retargeting. In: Computer Graphics Forum. pp. 11–19. Wiley Online Library (2000)
27. Stefanidi, E., Leonidis, A., Partarakis N., & Zabulis, X. (2019). CasandRA: a screenplay approach to dictate the behavior of Virtual Humans in AmI Environments. In Proceedings of HCI International 2019 (21st International Conference on Human-Computer Interaction), Springer CCIS XXX (in print), Orlando, Florida, USA, 26-31 July