



XRSISE: An XR Training System for Interactive Simulation and Ergonomics Assessment

Michail Pavlou, Dimitrios Laskos, Evangelia I. Zacharaki*, Konstantinos Risvas and Konstantinos Moustakas

Visualization and Virtual Reality Group, Department of Electrical and Computer Engineering, University of Patras, Patras, Greece

The use of virtual reality (VR) techniques for industrial training provides a safe and cost effective solution that contributes to increased engagement and knowledge retention levels. However, the process of experiential learning in a virtual world without biophysical constraints might contribute to muscle strain and discomfort, if ergonomic risk factors are not considered in advance. Under this scope, we have developed a digital platform which employs extended reality (XR) technologies for the creation and delivery of industrial training programs, by taking into account the users and workplace specificities through the adaptation of the 3D virtual world to the real environment. Our conceptual framework is composed of several inter-related modules: 1) the *XR tutorial creation module*, for automatic recognition of the sequence of actions composing a complex scenario while this is demonstrated by the educator in VR, 2) the *XR tutorial execution module*, for the delivery of visually guided and personalized XR training experiences, 3) the *digital human model (DHM) based simulation module* for creation and demonstration of job task simulations avoiding the need of an actual user and 4) the *biophysics assessment module* for ergonomics analysis given the input received from the other modules. Three-dimensional reconstruction and aligned projection of the objects situated in the real scene facilitated the imposition of inherent physical constraints, thereby allowed to seamlessly blend the virtual with the real world without losing the sense of presence.

Keywords: virtual reality, extended reality, xr training, ergonomics, digital human model, posture analysis, virtual workplace model, simulation

OPEN ACCESS

Edited by:

Christos Mousas,
Purdue University, United States

Reviewed by:

Kyle John Johnsen,
University of Georgia, United States
Kelly S. Hale,
Draper Laboratory, United States

*Correspondence:

Evangelia I. Zacharaki
ezachar@upatras.gr

Specialty section:

This article was submitted to
Virtual Reality in Industry,
a section of the journal
Frontiers in Virtual Reality

Received: 26 December 2020

Accepted: 09 February 2021

Published: 06 April 2021

Citation:

Pavlou M, Laskos D, Zacharaki EI,
Risvas K and Moustakas K (2021)
XRSISE: An XR Training System for
Interactive Simulation and
Ergonomics Assessment.
Front. Virtual Real. 2:646415.
doi: 10.3389/frvir.2021.646415

1 INTRODUCTION

Over the past years, extended reality (XR) technologies gained much importance in different types of industries, such as the manufacturing, architecture, automotive, health care, entertainment etc. This is due to various factors including the significant contribution of the research community alongside with big investments from the technology industry. Together with the big advances in the hardware supporting this technology, the end-users started to become more familiar with immersive reality experiences being supported on low-end mobile devices and affordable prices in the head mount displays existing in the market. In the field of software, the development of extended reality applications became more intuitive with the use of game engines, such as Unity3D and Unreal, even for professionals that want to make simple implementations, such as artists and architects.

Likewise, the deployment of this technology in the field of manufacturing is evolving rapidly due the new possibilities provided by AR/VR/MR applications in industrial processes. The realistic

representation of 3D models inside a virtual environment can offer the sense of embodiment, helping the design and preview of work spaces, adjusted on ergonomics, efficiency and other measurable factors. Spatial augmented reality visualizations can reduce the errors during industrial processes (Sreekanta et al., 2020), while the use of advanced user interactions within the virtual environment facilitates substantially the simulation of real world training scenarios (Azizi et al., 2019). Thereby, simulations of work tasks in virtual environments allows to avoid onsite learning on real devices with possibly costly materials, or to practice on potentially dangerous actions (e.g., to train inexperienced health care professionals without the risk of harming patients), while it also provides the possibility for unlimited remote training experience.

Several applications introduced maintenance task training with VR (Dias Barkokebas et al., 2019) and MR (Gonzalez-Franco et al., 2017) setups but lacked authoring tools for the creation of the tutorial inside the virtual environment, or required the physical presence of the trainer accordingly. (Wang et al., 2019) proposed a different MR collaborative setup, with the expert streaming his AR environment and supervising the trainee interacting with his own VR environment. A platform for the design of generic virtual training procedures was proposed in (Gerbaud et al., 2008). It accommodated an authoring tool for the design of interactive tutorials and included a generic model to describe reusable behaviors of 3D objects and reusable interactions between those objects.

In addition to operational costs and time optimization, XR training in work environments may allow to study human-machine interactions and optimize the operational design. Simulations-based assessment of ergonomics allows to identify parameters that may reduce the workers's productivity and workability, or may introduce or worsen adverse symptoms affecting the overall quality of life of the person (e.g., back pain worsening due to wrong body posture). Human-centered design can be promoted through the use of surrogates of physical prototypes, the well known digital human models (DHMs) (Duffy, 2008), which enable the incorporation of ergonomics science and human factors engineering principles in the product design process (Chaffin and Nelson, 2001; Caputo et al., 2017). DHMs have been incorporated in relevant software, such as Jack, Santos Pro, Ramsis, UM 3DSSP, to perform proactive ergonomics analysis (Duffy, 2008). Already more than a decade ago it was suggested to include in DHMs valid posture and motion prediction models based on real motion data in order to assure validity for complex dynamic task simulations (Chaffin, 2007).

Nevertheless, such computational approaches are still sparse, while the aim has mainly been on the development of proactive approaches to infuse human-factors principles earlier in the design process (Ahmed et al., 2018). A necessary step in such human-centered design approaches is to decompose the job assignments into subtasks (Stanton et al., 2013) in order to enable a fine-grained analysis of individual activities, and also to identify the objects with which the worker interacts. Most workflows employing DHMs (Stanton, 2006) for this purpose typically lack modularity to modeling new objects and actions as

well as automation (Rott et al., 2018). The presented framework is built to address the lack of linkage between the real-time execution of training scenarios in VR and the offline optimization of ergonomics through a DHM-based approach. This linkage allows to increase automation in the simulation process, as virtual objects designed in one of the modules are seamlessly transferred and utilized by the other.

1.1 Contributions of Current Work

In this paper we present an XR System for Interactive Simulation and Ergonomics (XRSISE). Its main contributions include the provision of (non-commercial) authoring tools that allow to 1) increase modularity in modeling interactable objects and workflows, 2) automate the creation of training tutorials through a "recording" mechanism, and 3) optimize the workplace design through offline DHM-based simulation and ergonomic analysis of selected tutorial parameters. The functionalities of XRSISE are contained within four distinct modules, namely the:

1. *XR tutorial creation module*, for the development of new training material based on interactive simulations of the work task,
2. *XR tutorial execution module*, for getting trained in VR through a step-by-step (previously recorded) training tutorial in XR, supported by visual hints and feedback on performance,
3. *DHM-based simulation module*, that maps the real user to his/her digital twin acting in the same simulated environment in order to test the developed training scenario without the need of an actual human user, thus to allow performing offline optimization of all parameters of interest,
4. *Biophysics assessment module*, that contains all necessary techniques to perform DHM-based postural analysis thereby allowing to fine-tune the most critical parameters according to personal preferences.

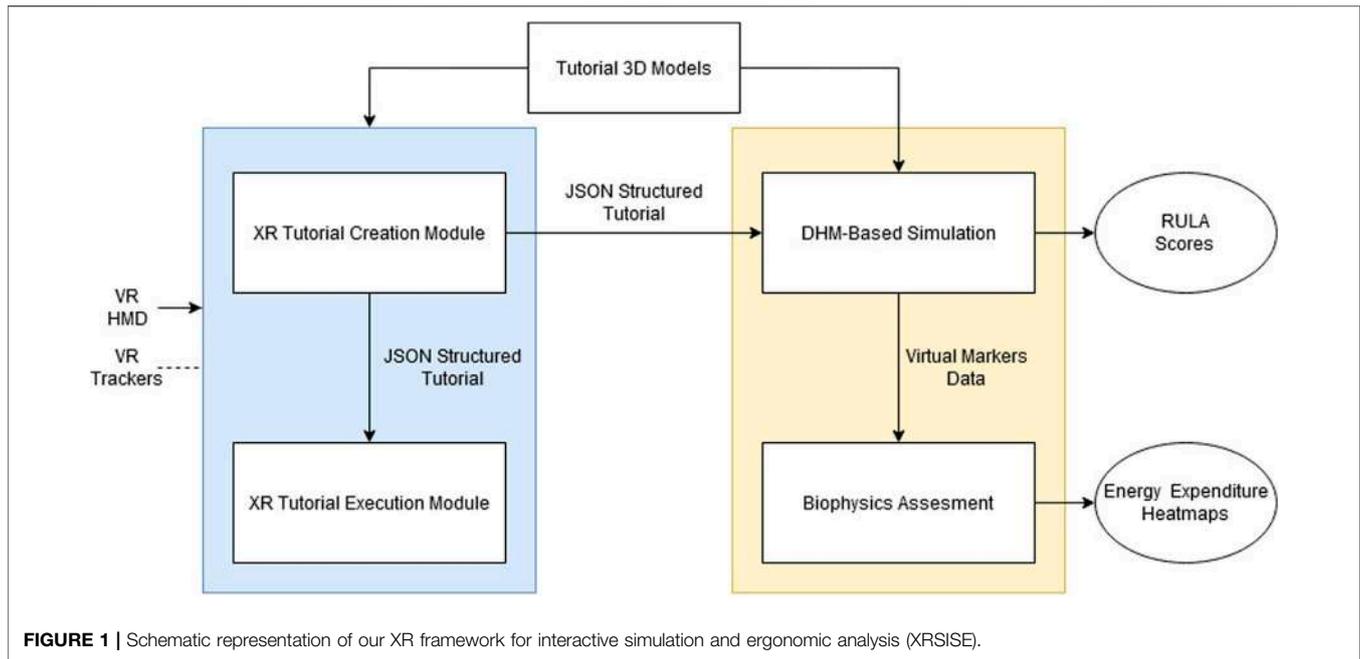
Details on the individual components are presented in the next sections, followed by an industrial training application example (i.e., a drilling operation) and a short user evaluation study assessing user perception (e.g., presence) of the modeled environment.

2 XRSISE

2.1 Architecture

In more detail, the XRSISE tools were designed as part of the Ageing@Work¹ project, which aims at the development of novel ICT-based, personalized solutions supporting aging workers in flexible management of their evolving needs. Towards this scope, we implemented VR-based context-aware training and knowledge exchange tools following a user-centered design approach, aiming to offer an ambient support into

¹<https://ageingatwork-project.eu/>



workability and wellbeing. A schematic diagram of XRSISE with its different interconnected modules is illustrated in **Figure 1**.

The XR tutorial creation and execution modules share common VR/XR-based technological tools mainly targeting the interaction of a real user with objects in a modeled 3D scene. Their development is based on Unity3D which facilitates the design of interactive XR training scenarios. Specifically, we exploited the Unity XR Plug-in framework (Manual, 2020 accessed November 24, 2020) which provides the ability to integrate cross platform XR applications regardless of the utilized hardware. The user interactions are based on the extension of the XR Interaction Toolkit (Docs, 2020 accessed December 16, 2020), a customizable high level interaction system. The presented framework provides an improved user interface that facilitates the introduction of a semantic layer into the geometric objects and the design of a tutorial by attaching plug-n-play components to the Unity game-objects. The main requirement (that has to be addressed by the designer beforehand) is the segmentation of the machine 3D mesh into individual parts, which will appear as separate game-objects when imported to Unity.

The transition from VR to the interconnected DHM-based simulation environment is presented in **Section 2.4**, whereas details on the ergonomics optimization module are provided in **Section 2.5**.

2.2 XR Tutorial Creation Module

The XR tutorial creation module is used to create a training tutorial in order to educate inexperienced workers on the use of machines or industrial control panels in simple or more complex scenarios, avoiding dangers and risks inherent during the real (physical) job assignments. It utilizes a set of XR tools designed to simplify the process of creating or editing training tutorials. The

module consists of four basic components, which are described next.

- *Interactable component*: An interactable component is a property of any object requiring physical interaction with the user, such as discrete/continuous rotating levers, pressable buttons, squeezable knobs and grabbable objects that can be placed inside specific areas. Interactable objects can hold information about the type and the state of the interaction and the input required from the VR system (e.g., controllers) to activate them. The interactable component acquires and interprets the input of the user and then communicates with its parent *machine component* through a message passing mechanism, resulting in the dissociation of the human interaction and the behaviour of the machine, as followed in (Tanriverdi and Jacob, 2001). The tutorial designer can easily attach the interactable component to a 3D object, requiring physical interaction, through the Unity inspector and tune the settings each interactable provides, such as min/max rotation values, axis of rotation, push offset etc. The interactable component design is based on the core concepts of the XR Interaction Toolkit. The core of the Interaction Toolkit is composed of a set of base Interactor and Interactable components, an Interaction Manager and helper components for improved functionality in drawing visuals and designing custom interaction events. The lack of available interactables (only the grab interactable was supported) led us to exploit the helper classes and design new interactables based on the framework.
- *Machine component*: It stores the set of actual machine functionalities in the form of dynamic properties, thereby allowing the operation modeling of a physical machine.

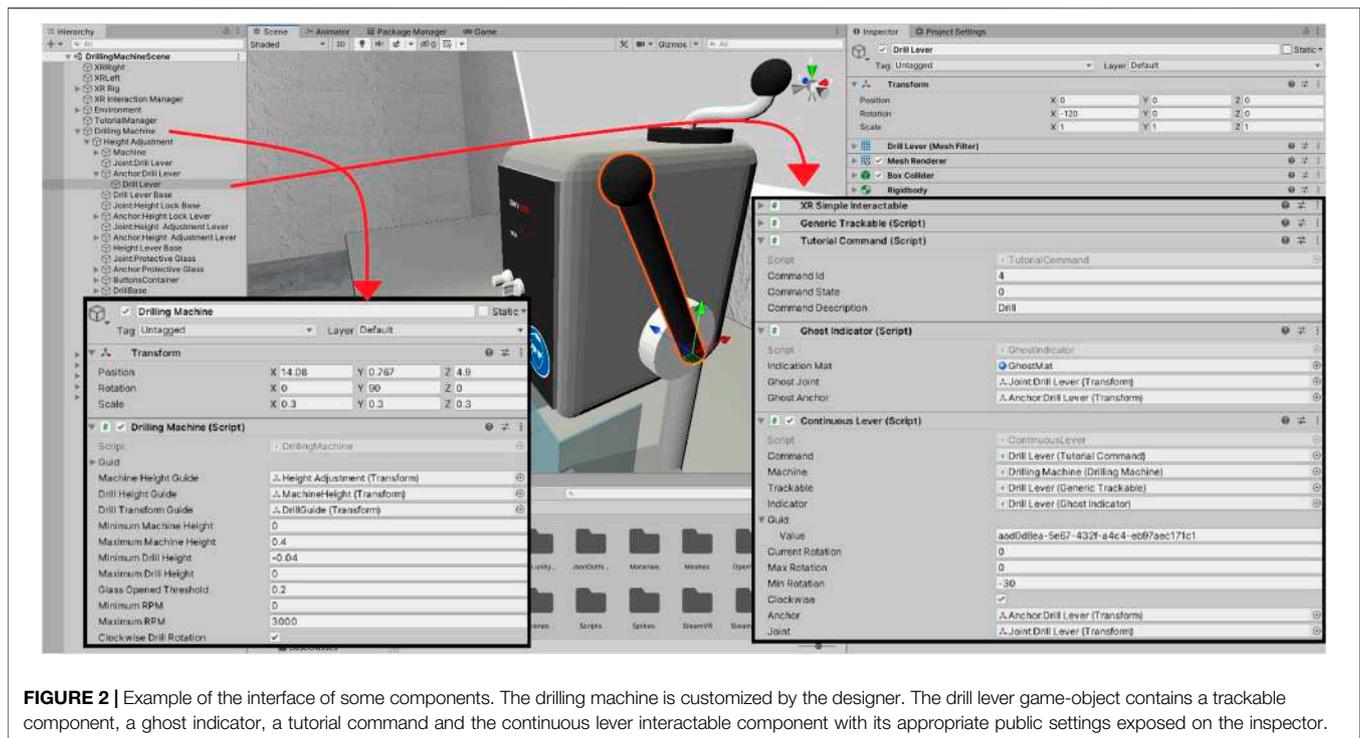


FIGURE 2 | Example of the interface of some components. The drilling machine is customized by the designer. The drill lever game-object contains a trackable component, a ghost indicator, a tutorial command and the continuous lever interactable component with its appropriate public settings exposed on the inspector.

The use of machine components helps in the design of self-aware machine-based behaviour and not just a collection of independent pairs of actions and their resulting effects. The machine through a message passing mechanism, receives inputs from its child interactable components. The operational execution model of this component can determine the reactive behaviour of the machine, through a set of rules that determine the execution hierarchy of the requested actions and a set of activities-actions, using a methodology similar to (Cremer et al., 1995). A simple example describing the aforementioned control mechanism is a machine that does not execute any action requested from any interactable, until it gets a request to get enabled (i.e., an interactable 3D button that notifies the machine component to turn on). It should be noted that, as every machine has different rules of activation of behaviours, the “design” of a machine component is the only part requiring manual effort from the designer, without the need of high expertise in software development. The designer must specify distinct commands for the parts of the machine. The commands are modeled through a simple Unity component class (named “Tutorial command”) containing the ID of the command and the dynamic state which can be then passed through the message mechanism to the machine. Next the designer has to devise the machine component (which inherits the abstract MachineBase class of the framework) with the rules and actions activated when the machine receives interactions from its “children” interactables. The disjunction and abstraction of the tracking procedure

from the interaction itself, facilitates the further extension of the framework with new and more complex interactions in the future.

- *Trackable component*: Depending on the type of the performed interaction this component maintains record of the continuous or discrete state of the interactive objects, or the 3D position of the grabbable objects and the areas they are placed into. Its use is necessary only for interactions that shall be required to be tracked, while the user (instructor) creates the training scenario. The use of a trackable component instance together with the interactable component instance results to the tracking of the interaction by the Tutorial Manager.
- *Tutorial Manager*: Differently from the previous three components the Tutorial Manager class handles the flow of the created tutorial and informs the user on the performed steps through a 2D visualization panel.

An overview of these components being used on the unity interface can be found in **Figure 2**.

The XR tutorial creation module is initiated by first loading a previously designed 3D workplace of interest. Depending on the device the user can interact with objects of the environment using his bare hands or the controllers, as described above. Every 3D interaction is tracked automatically and marked as a new step of the training tutorial. If the user makes a mistake through the process, he/she can delete the tracked step through the 2D panel appearing in front of him/her. When all the steps are completed, the user presses the (virtual) end button, and the Tutorial Manager encodes the tutorial in a JSON formatted file.



FIGURE 3 | Inclusion of a physical box in the 3D training environment, with the use of a VIVE Tracker and SteamVR.

2.3 XR Tutorial Execution Module

The XR tutorial execution module allows to rollout previously developed training programs, i.e., JSON encoded tutorials. The module includes the same components with the XR tutorial creation module, i.e., *interactable components*, *machine components*, *trackable components*, and a *Tutorial Manager*, although their utilization is different. The Tutorial Manager operates as an intermediate control mechanism for both modules. The machine after receiving a request from an interactable component and validating it, communicates with the Tutorial Manager and sends the interaction data instance. While the role of the Tutorial Manager in the tutorial creation module was to hold every new valid interaction data, in this module the Tutorial Manager first loads the tutorial steps from a JSON file, and then, when receiving a new interaction data instance, compares the instance variables and notifies the user if the respective tutorial step was performed correctly.

The identification of a step as correct is based on the common information shared between the machine component and the tutorial manager. This information is practically a struct containing four elements: the machine UUID (Universally Unique Identifier), the interactable UUID, the tutorial command and the command state. The first three elements are unique determining the action. The command state is the result of the interaction which is passed from the interactable to its parent machine. It can be a button press (resulting to a Boolean state) or a lever rotation (which produces a continuous output). The machine component encodes this information, based on the rules specified by the developer, into the state it is programmed to. The interactions currently supported by the framework are simple and can result to categorical states with two or more categories or continuous-valued states (e.g., rotation angle of a handle). The latter are quantized into discrete values to facilitate the state check (by the tutorial manager), which obviously is ensured only within the level of quantization precision.

A 2D visualization panel projected in the users' front view is also utilized here to guide the trainee through the process by illustrating the sequence of performed and required actions and by displaying his/her performance after each execution task. The user gets notified through the panel so that he/she can proceed to the next step. Moreover, to facilitate novice trainees, several types of indications are available in this module; i.e., every interaction is visualized on the (unfamiliar) machine by highlights in non-moving parts and animation in dynamic parts. These ghost

animations are created through a dynamic copy of the game-object's 3D mesh and recreate the motion which is required to reach the target state. In this way the trainee can easily recognize the position of the interactive components and the type of anticipated interaction.

While VR allows to coordinate the spatial and temporal user's interactions, it does not provide any physical constraints that emerge in the real world and that might be critical for the subsequent ergonomic analysis. In order to optimize the design of the workplace according to user specificities and preferences, the XR tutorial creation and execution modules offer the possibility to first adjust the virtual environment to personal and physical objects' characteristics, such as the height of the user. This is achieved by loading a virtual user profile (anthropometrics), or by adapting the head-mounted display (HMD) calibration profile, as provided from the device's corresponding setup software. While this is currently performed through interference of the 3D designer, future work includes the investigation of more automated calibration methods.

Furthermore, in addition to HMD calibration performed to induce the feeling of presence in an workplace that best mimics the real environment, XRSISE supports also the integration of actual (physical) objects inside the virtual scenarios in order to recreate the physical world as realistically as possible, whenever necessary. More specifically, the use of 6-DoF tracking devices gives the designers the possibility to integrate some types of rigid objects that could be important for the training program, such as objects that provide physical support to the human body. If physical objects have to be integrated, the designer adds the tracker objects to the environment, and the calibration is handled through the

TABLE 1 | Adjustable DHM parameters.

Digital human model parameters

Height
Sitting height
Height from floor to top of knee
Height from sitting surface to top shoulder
Shoulder-elbow length
Forearm length
Hip breadth

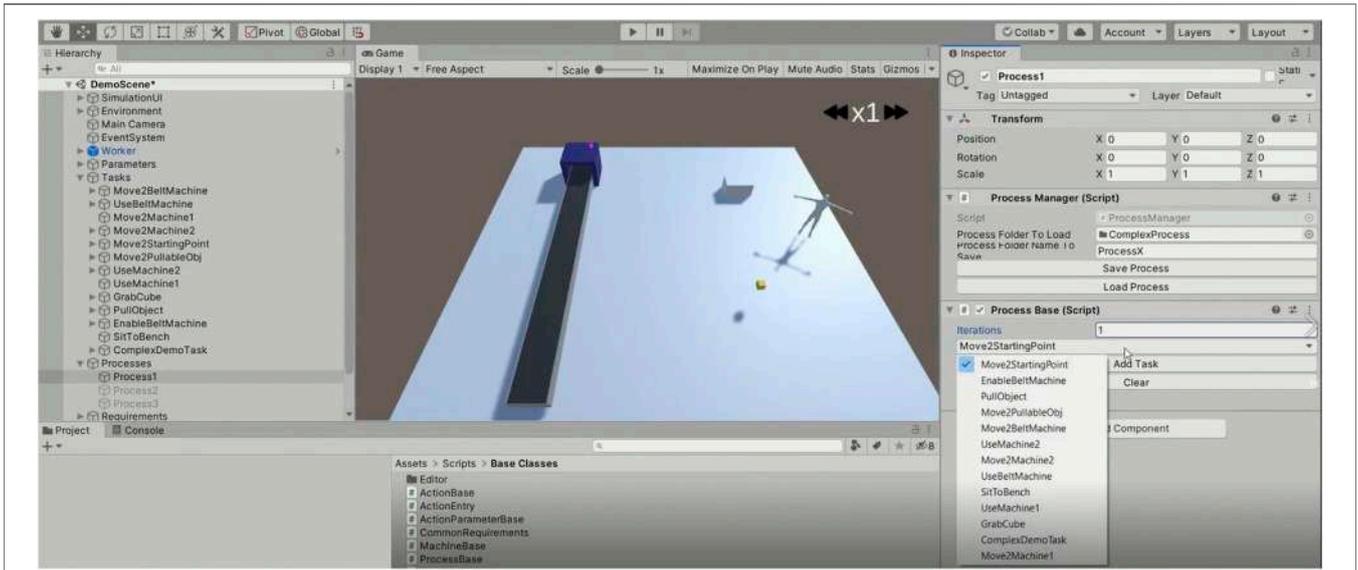


FIGURE 4 | Overall view of the DHM-based simulation module.

corresponding software. In the present work we utilized VIVE trackers and integrated them through SteamVR software (Valve, 2020 accessed November 24, 2020) due to the fact that the current XR plugin framework version does not support 6-Dof tracker devices. An application example of physical objects integration in VR through XRSISE is illustrated in **Figure 3**. For the developed training tutorial we included a real box as supporting element of the right arm during the virtual execution of a manual handling operation (rotation of a lever). Our idea was that the use of a box, with height adjusted to the users’ anthropometrics, could serve as a means to reduce arm fatigue during repetitive task execution. We should note though that in the current version of XRSISE, it is not possible to simply introduce the integrated physical objects within the biophysics assessment module, because the interaction forces (between user and environment) at contact locations are

unknown. Thus the design of such physical objects is intended to be optimized based on users’ preferences after task repetition, and not based on automated ergonomic analysis. Since such evaluation was not part of the protocol of the user study, we did not include physical objects in the application scenario presented in **Section 3**.

2.4 DHM-Based Simulation Module

The operation of industrial machinery has often been related to risk factors for musculoskeletal disorders, especially for repetitive movements and increasing age. The ergonomics optimization module allows to estimate the stress imposed to the human body during the performance of a work task in an industrial environment. By simulating the sequence of body postures obtained during a job execution prior to the

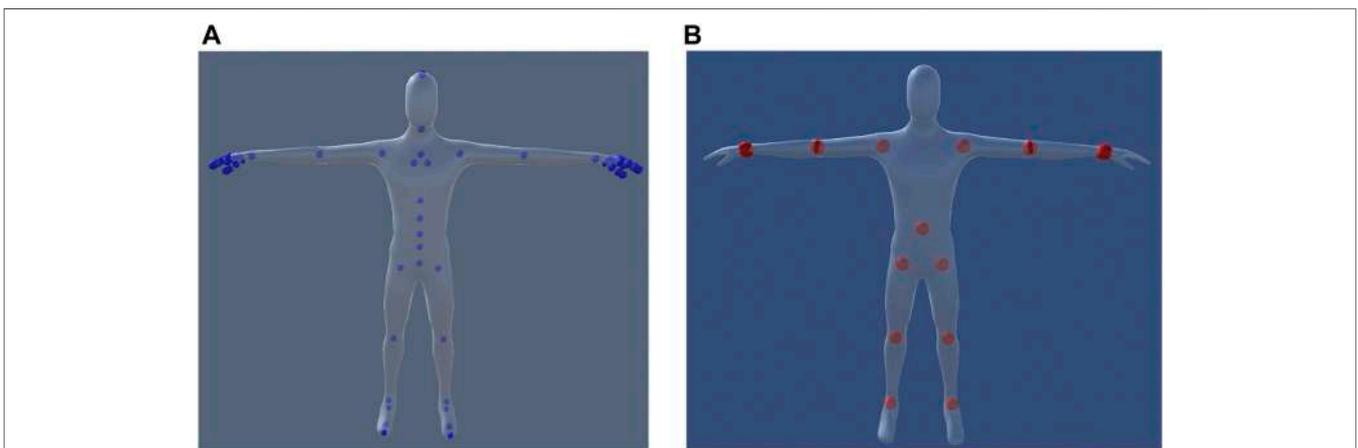


FIGURE 5 | The human model in the DHM-based simulation module. **(A)** The position of each bone **(B)** The 13 joints used for inverse kinematics.

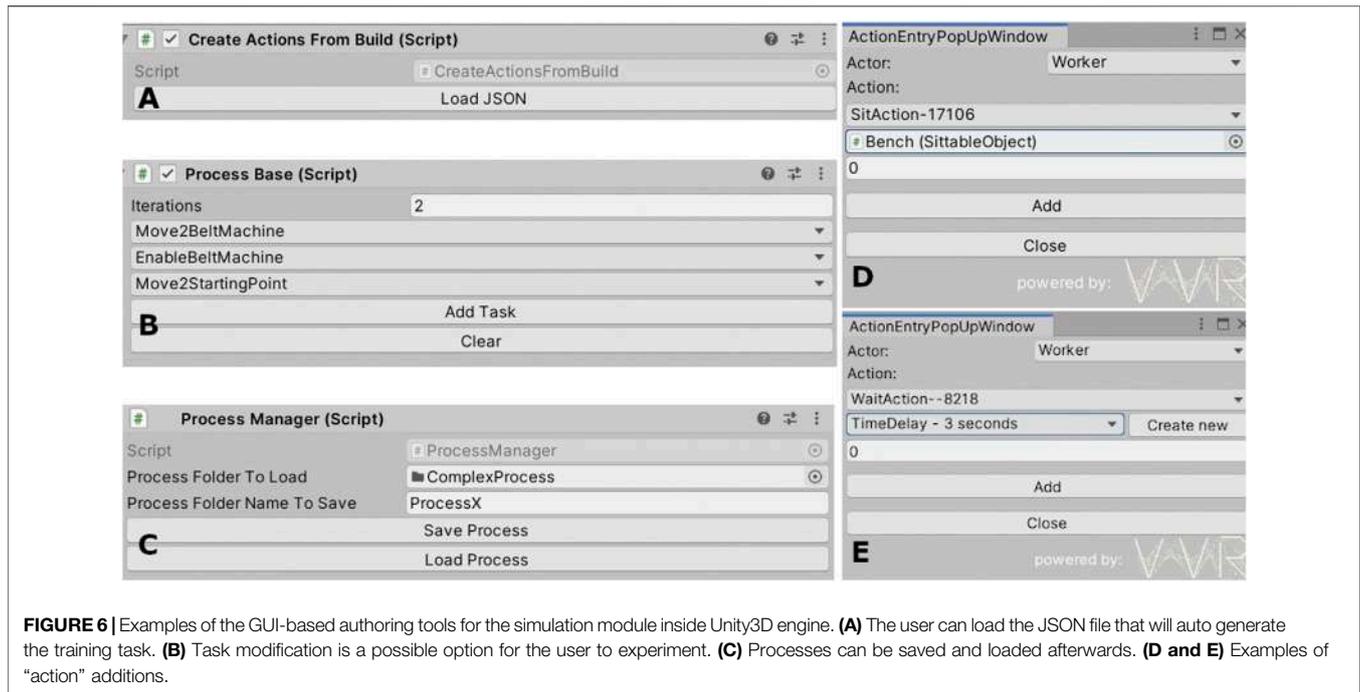


FIGURE 6 | Examples of the GUI-based authoring tools for the simulation module inside Unity3D engine. **(A)** The user can load the JSON file that will auto generate the training task. **(B)** Task modification is a possible option for the user to experiment. **(C)** Processes can be saved and loaded afterwards. **(D and E)** Examples of “action” additions.

physical involvement, harmful actions can be prevented. Personalization in the estimations is achieved through the creation of a DHM acting as a digital twin of the worker. The postural analysis is handled by the biophysics assessment module as described in **Section 2.5**, whereas the current module provides the graphical user interface (GUI) to synthesize a virtual scenario through the selection of sequence of actions, as well as the optional visualization of the simulated environment and the dynamic reproduction of the DHM acting in it. A screenshot of the GUI is illustrated in **Figure 4**. The simulation module is written in C# using the Unity 3D development platform.

The simulation module offers multiple capabilities to the user through its core components:

- **Action:** Action is the simplest simulation element. It is linked to an actor (a DHM), a type and an action parameter. Types of actions can be selected from a predefined set or can be customized based on the simulation needs. Predefined actions include walking, sitting, pressing, pulling, pushing and grabbing an object.
- **Action Parameter:** Depending on the type of action, the parameter can correspond to a simple transformation or to a custom composite data structure with multiple objects and constraints. For example, for a “walking” type of action, the action parameter is the destination point while for a “rotate” type of action the parameters are the rotatable object, the rotation angle and the rotation direction.
- **Requirement:** An action may meet one or more requirements in order to be successfully completed. The simulation will proceed with the execution of the action only if all defined requirements are fulfilled.

- **Task:** A task is a collection of actions executed in sequential order. It can be performed only once or it can be repeatable.
- **Process:** A process is a collection of tasks performed in a sequential order.

The above components are initialized or modified through a GUI and through them the user is capable of creating the scenario to be simulated.

2.4.1 Digital Human Model

The human model that we are using consists of 72 bones, each one of them representing key body parts to simulate in detail every possible movement a worker can do. The model is adjustable according to the worker’s anthropometric characteristics presented in **Table 1**. To determine the joint parameters that provide the desired configuration, an inverse kinematics problem is solved using the FABRIK (Aristidou and Lasenby, 2011) algorithm. For this purpose, a rigid multibody system with fewer degrees of freedom (13 joints and 14 links) is used to allow fast and lightweight execution. The definition of joints and links on the implemented DHM is shown in **Figure 5**. After mapping the human 3D model to the multibody system, we use four iterations for the FABRIK solver to reach the desired target (location of the DHM end effector). Although the ergonomics assessment does not rely on the movement of fingers, for a complete visual simulation we are also using pre-defined finger animations for common actions.

2.4.2 Connection of DHM-Based Simulation Module With XR Training Creation Module

The XRSIZE framework provides also the possibility to (semi-automatically) transfer the output of the XR training creation module to the DHM-based simulation module. This allows to

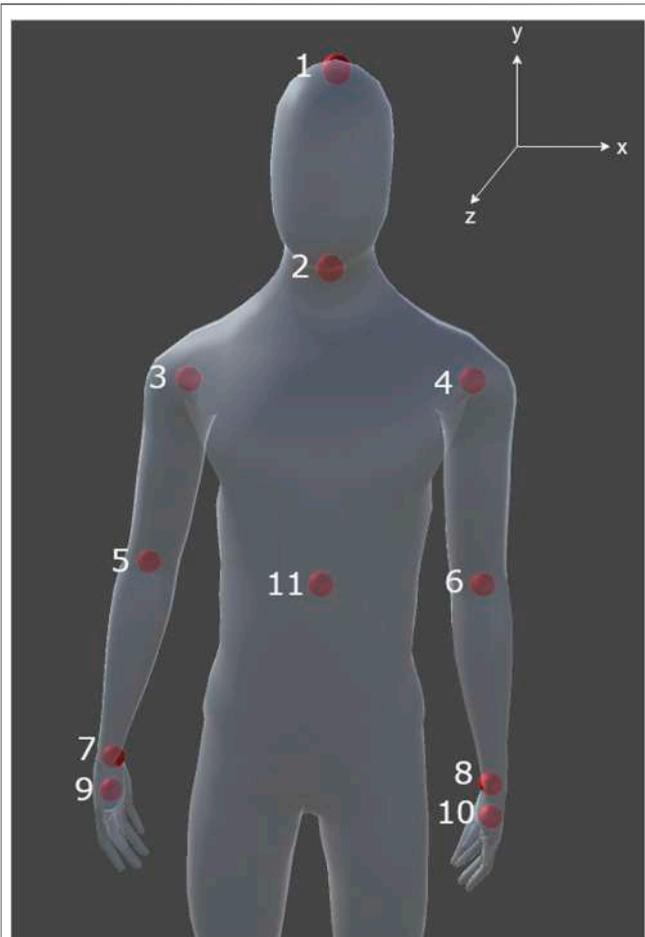


FIGURE 7 | Joint positions and indexing.

simulate offline the tasks that have to be executed by the real user in advance, thereby acting as the front-end of ergonomics analysis. For this purpose it is important to create in the simulation module a virtual scene that includes 3D models identical with the environment the user is interacting with during XR training. This information along with the description of prescribed sequence of actions (such as rotating, pushing, pulling, releasing objects), and the corresponding interactions, are all included in the JSON file. Each action in the JSON file is described by an identifier (i.e., the name of the action) and by the parameters completing the definition of the action as described earlier in the section (“action parameter” and “requirements”). The JSON file also contains information about each “task” and “process” contents. The simulation framework extracts all the required information from that file in order to recreate the complete scene with the exact same procedures the user was following during XR training creation.

After setting up the scene, the simulation module visualizes the sequence of encoded actions in the real world scenario, along with additional information about calculated biophysical

parameters of interest, such as the strain in each body segment (as discussed in the following section). The user can also modify the actions (e.g., change the order or add new actions) through a graphical user interface (Figure 6) or even modify the virtual environment. This allows to examine the effect of workplace specificities for the given anthropometrics (of the DHM) and thereby personalize the design of the workplace environment or identify the best placement for a specific machine. It should be noted that extensive environment modifications might result in corrupted actions from the JSON file and should be performed with caution. For example, altering or deleting an interactable part of a machine structure might cause errors due to the fact that the auto generated action might be incomplete.

2.5 Biophysics Assessment Module

The main functionality of the biophysics assessment module is based on the Inverse Dynamics problem, in which the acceleration of the system is known while the forces and joint torques that caused this observed motion need to be determined (Winter (2009)). The motion of the system (the digital human model in our case) is described by experimental kinematic data acquired by input controllers and restricted by applied constraints and contacts.

Ergonomics assessment in XRSISE can be performed through two distinct approaches: 1) based on the Rapid Upper Limb Assessment (RULA) (McAtamney and Corlett, 1993) that is commonly used across studies and allows to obtain a benchmark baseline metric, and 2) based on biophysics laws that provide more personalized and accurate evaluation of the dynamic movements. While the RULA is simple to apply, calculation of the ergonomic risk score is based on the discretization of posture into ranges of joints’ motion, whereas musculoskeletal modeling approaches provide greater resolution (Mortensen et al., 2018). Moreover, RULA is usually applied to a snap shot of extreme or typical postures, and does not provide insight into the cumulative effect of dynamic postures and temporal energy expenditure. Details on both methodologies are provided in the following subsections. It should be noted that the current version of XRSISE framework supports only the RULA-based approach as a seamless and fully integrated pipeline, whereas the mapping of the DHM interactions into the OpenSim workflow requires manual adjustment and visual verification.

2.5.1 Ergonomics Evaluation Through RULA

RULA (McAtamney and Corlett, 1993) is one of the most adopted approaches in the literature for empirical risk assessment of work-related musculoskeletal disorders. RULA

TABLE 2 | Joints used for angles calculation. L:left; R:right.

Joint points	Direction vector (\vec{V})	Angle description
3–5 (R) and 4–6 (L)	$(0, 1, 0)$	Upper arm angles (L&R)
5–7 (R) and 6–8 (L)	$(0, 1, 0)$	Lower arm angles (L&R)
7–9 (R) and 8–10 (L)	$\vec{P}_{j\beta}P_{j\gamma}$ (R) and $\vec{P}_{j\beta}P_{j\delta}$ (L)	Wrist angles (L&R)
1–2	$(0, 1, 0)$	Neck angle
2–11	$(0, 1, 0)$	Trunk angle

considers only the upper limb and was selected in XRSIZE as it covers the general industrial work tasks in which we are focusing. However, the biophysics assessment module can easily be extended to include methods for whole body assessment, such as the *Ovako Working Posture Analyzing System (OWAS)* (Karhu et al., 1977) and the *Rapid Entire Body Assessment (REBA)* (McAtamney and Hignett, 2004).

The joint angles required by the algorithm were calculated using the position of the joints displayed in **Figure 7** for every simulation frame. Given the position of two joints in the 3D space, P_1 and P_2 we can easily calculate the angle φ between vector $\overrightarrow{P_1P_2}$ and a direction vector \vec{V} using the following equation:

$$\varphi = \arccos \frac{\overrightarrow{P_1P_2} \cdot \vec{V}}{\|\overrightarrow{P_1P_2}\| \cdot \|\vec{V}\|} \quad (1)$$

Table 2 summarizes the joints we used for the angles calculation with the appropriate direction vectors.

After calculating the required joint angles and considering other factors affecting the body strain, such as load or repetitions needed for a specific action, a final score is calculated which varies from 1 to 7. A score of 1 or 2 indicates acceptable posture while a score of 3 or 4 indicates the need of further investigation without the requirement for changes to be implemented in the working environment. Higher scores imply the need for further investigation and changes to be implemented in the near future (score 5 or 6) or urgently (score 7).

2.5.2 Ergonomics Evaluation Through Musculoskeletal Modeling

Building upon our previous work (Risvas et al., 2020), we perform biomechanical analysis using the open source software platform OpenSim (Seth et al., 2018), that is suitable for modeling and simulation of the musculoskeletal system of humans, animals and robots. In the confines of this work, a standard OpenSim workflow was implemented, that includes the Scale, Inverse Kinematics, Inverse Dynamics and Kinematic Analysis tools. The activation and force produced by each muscle is not considered, and we are interested only in the net forces and joint torques that are related to the mechanical energy and power demands of the musculoskeletal system for the evaluated scenario. The exploited OpenSim model of human upper body consists of nine rigid bodies, interconnected by nine joints that offer 20 degrees of freedom (DoF). The key steps of the biomechanical approach are described next.

- *Kinematics Export*: The motion of the digital model in the Unity3D virtual environment was recorded and exported using virtual markers attached in specific locations based on guidelines for collecting experimental data in real motion capture laboratories. Specifically, three non-collinear markers were placed in each body segment and some additional ones were attached close to the anatomical joint centers to record position and orientation at each time frame of the scene during the simulation. The coordinates of each marker were then mapped to the

global coordinate system of OpenSim and stored in YAML format file, which is subsequently converted to TRC format to be compatible with OpenSim.

- *Scale Tool*: The Scale Tool was repeatedly used to calibrate the musculoskeletal model in order to match the anthropometric data of the virtual humanoid model and to properly adjust the positions of the virtual markers on the musculoskeletal model to coincide with the Unity3D “experimental markers”. For this, we used a static trial during which we recorded several frames with the DHM placed in a static pose. Then, the Scale Tool was used to solve a least square minimization problem to fit the virtual markers to the “experimental” markers.
- *Inverse Kinematics (IK)*: The IK Tool was used to compute the generalized coordinate (joint angles and/or translations) trajectories by positioning the model in the appropriate posture at each time step. This posture is determined based on the experimental data by minimizing a weighted least square distance criterion, as performed during musculoskeletal model calibration. The minimization errors were observed and the Scale tool was executed again to fit the marker positions. This process was repeated until the values for the maximum and root-mean-square errors were low enough to not impede the analysis. The Analyze tool was then used to perform kinematic analysis by calculating the generalized coordinates, speeds and accelerations given the motion data estimated by the IK Tool.
- *Inverse Dynamics*: Dynamics refers to the classical mechanics science that deals with the analysis of forces that are related to motion. The motion data that were obtained after the implementation of IK are streamlined to the OpenSim Inverse Dynamics Tool, that calculates the unknown generalized forces and torques by solving the classical equations of motion.
- *Joint Energy*: The results of the Inverse Dynamics (joint torques) and Kinematics (joint angles) Tools were used to calculate the energy for each joint using **Equation 2**:

$$Energy = M \times q \quad (2)$$

where M is the joint net torque and q is the generalized angular displacement.

In order to better comprehend the dynamic posture risks, musculoskeletal modeling results are visualized on the DHM by highlighting the computed energy values through heatmaps on the different body parts.

3 APPLICATION EXAMPLE OF XRSIZE FRAMEWORK

In this section we will describe and evaluate the selected use case for the XRSIZE framework. The training scenario describes the steps required for the operation of an industrial drilling machine. The training tutorial includes actions for safety assurance for the machine and the worker, as

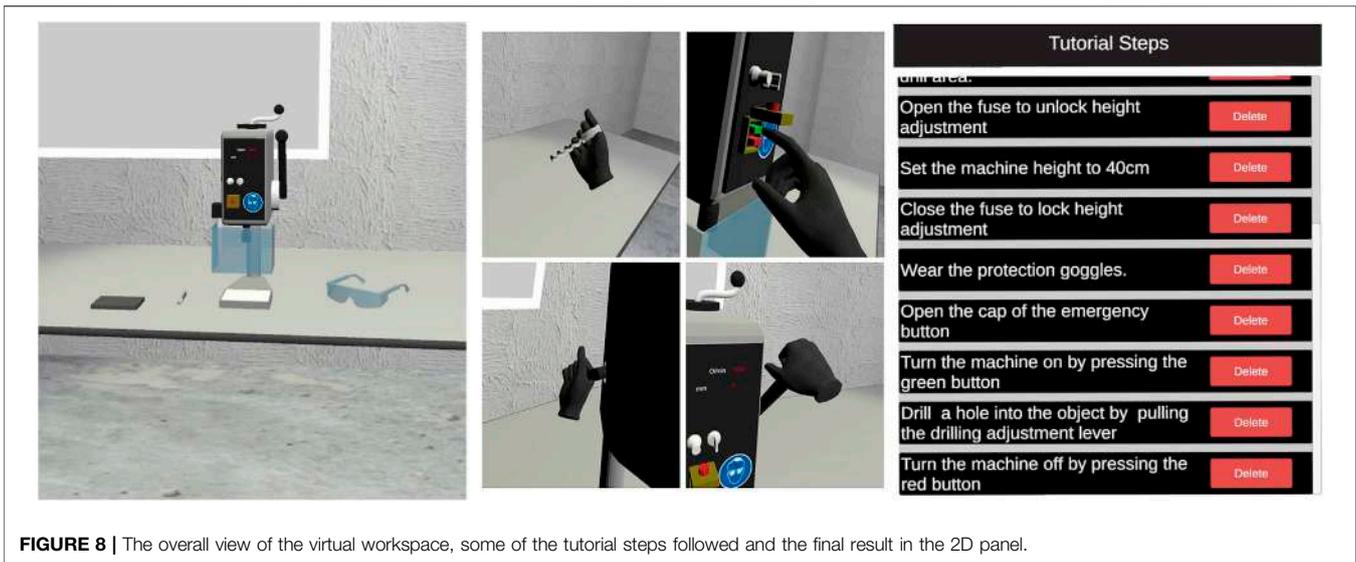


FIGURE 8 | The overall view of the virtual workspace, some of the tutorial steps followed and the final result in the 2D panel.

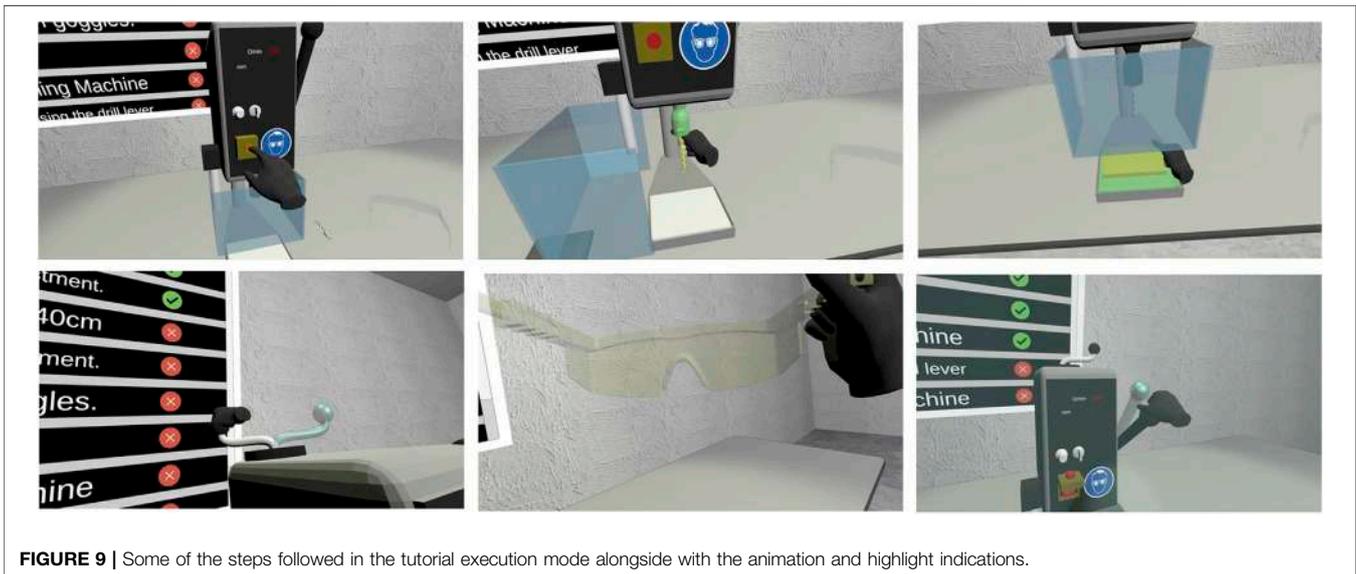


FIGURE 9 | Some of the steps followed in the tutorial execution mode alongside with the animation and highlight indications.

well as operation instructions of drilling a hole in an object of a certain material. It consists of the following 13 specific steps:

1. Press the emergency button to ensure that the machine is off.
2. Open the protection glass.
3. Insert the drill bit.
4. Close the protection glass.
5. Grab the object and insert it under the drill.
6. Open the fuse to unlock height adjustment.
7. Adjust height with the top lever.
8. Close the fuse to lock height adjustment.
9. Wear the protection goggles.
10. Open the cap of the emergency button.
11. Turn the machine on by pressing the green button.

12. Drill a hole into the object by pulling the drilling adjustment lever.
13. Turn the machine off by pressing the red button.

3.1 Training Scenario Design and Tutorial Creation/Execution

The proposed scenario requires the 3D modeling and visualization of the drilling machine, the drill bit, the protective glasses and the 3D work space environment. All the models were created using the open source 3D modeling tool, Blender (Community, 2018). In addition one physical object (a box) was integrated in VR, as described in **Section 2.2**, in order to

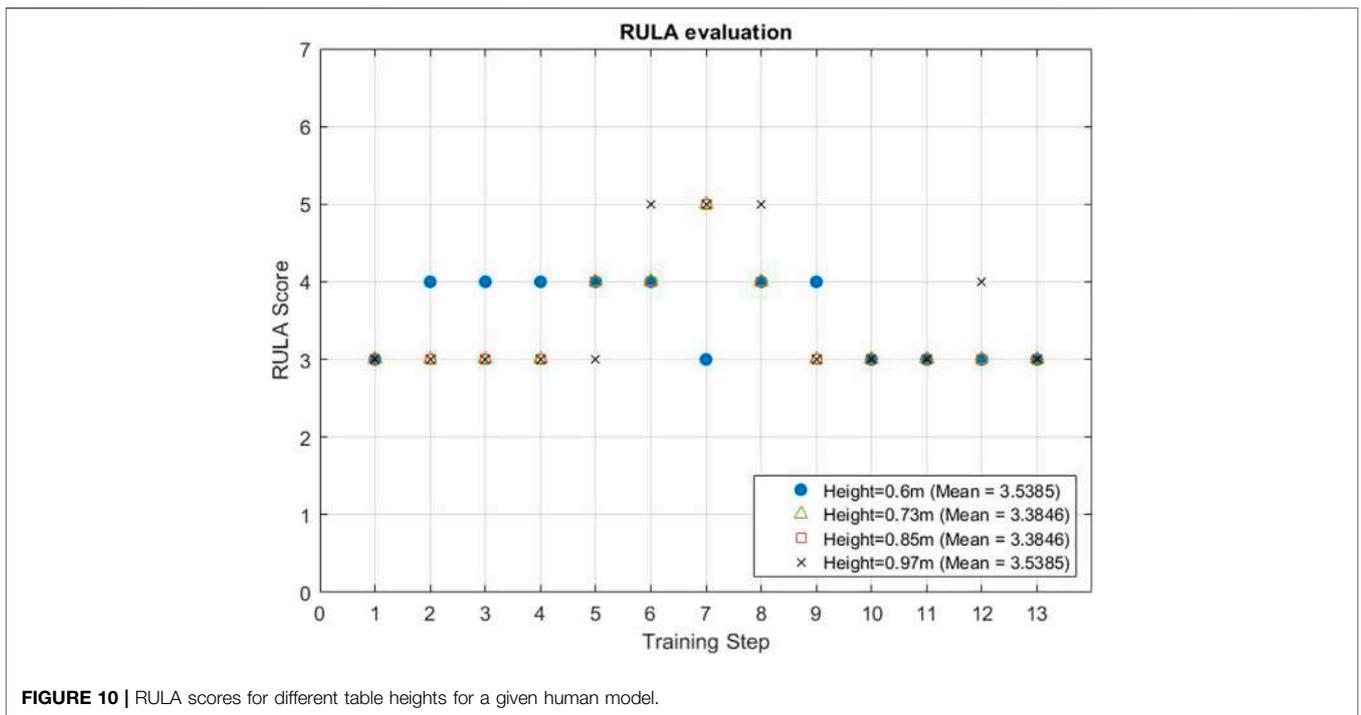
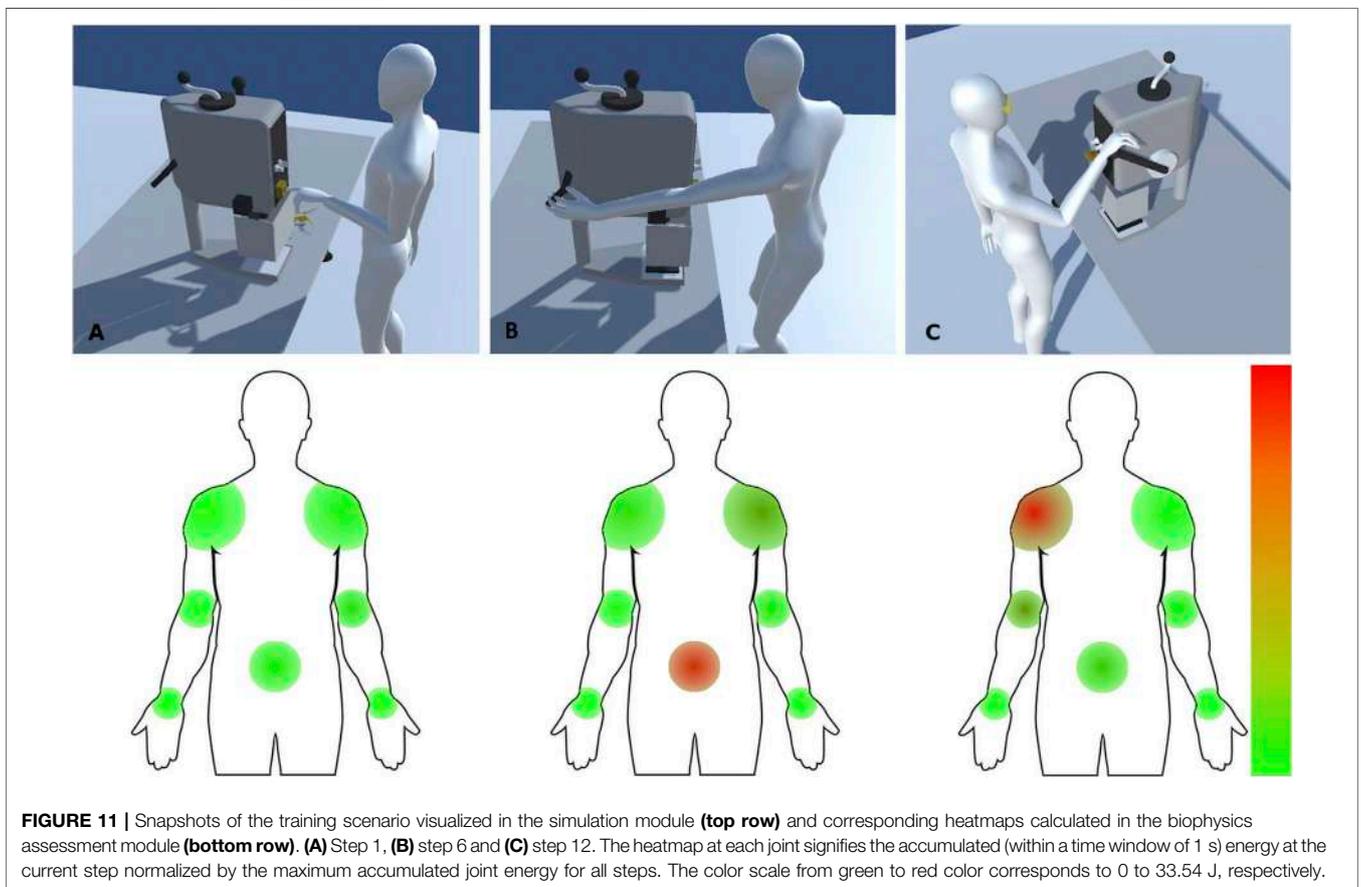


FIGURE 10 | RULA scores for different table heights for a given human model.



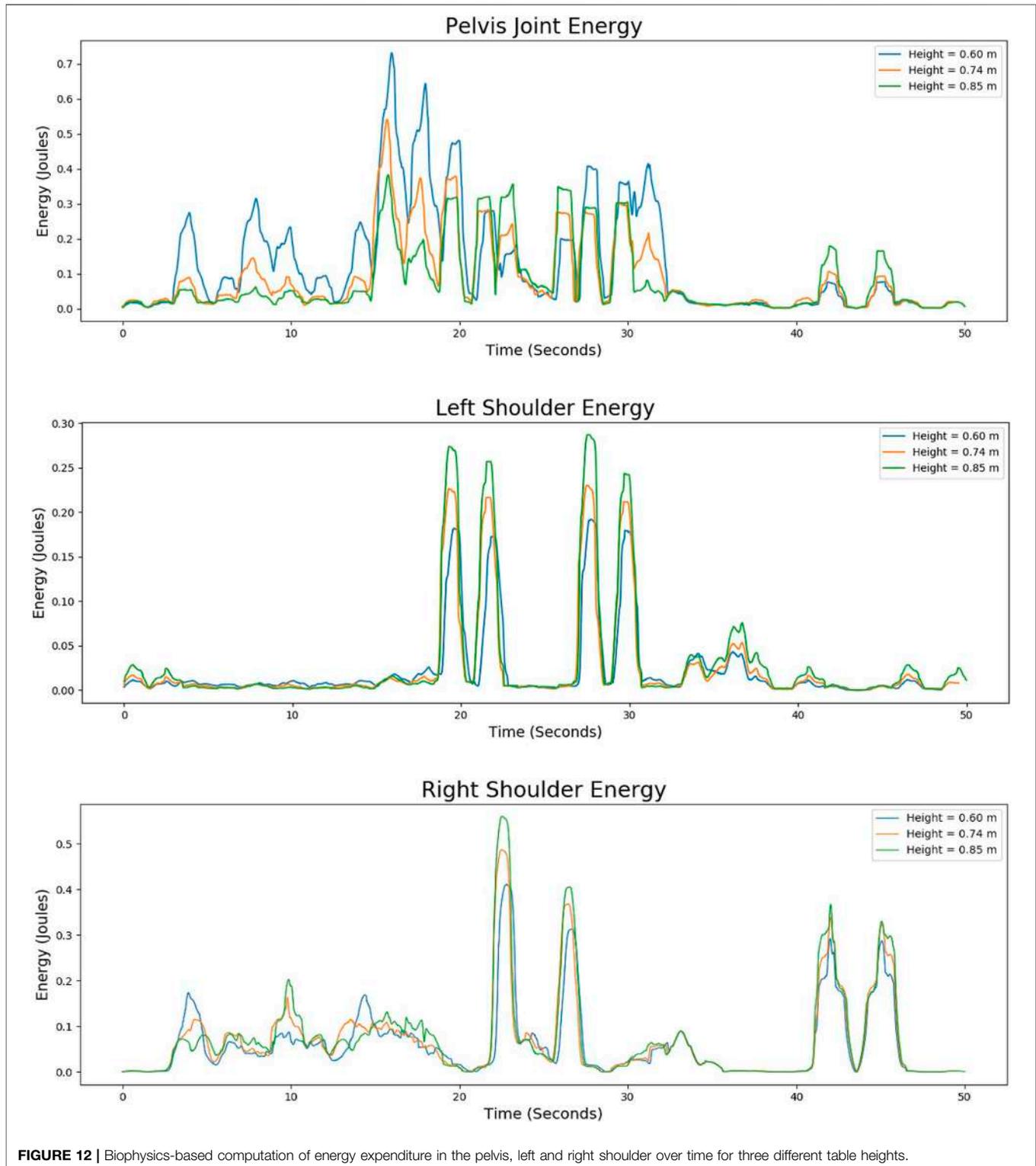


FIGURE 12 | Biophysics-based computation of energy expenditure in the pelvis, left and right shoulder over time for three different table heights.

provide support to the worker's arm during drilling (12th step of the tutorial) and thus reduce fatigue. The parameterization of the physical object (e.g., box height) is important for the biophysics assessment module, as it allows to identify the ergonomically best settings for each user.

After importing the models of the Unity3D scene, and based on the components described in **Section 2.2**, the tutorial designer defines the interactable and trackable components to the interactive parts of the 3D machine and the spare parts (drill bit and protective glasses), and then tunes several real world

TABLE 3 | Mean values and standard deviations of different factors and subscales.

Factors	Items	Mean	± SD
Control factors	1,2,3,7,13,14,25,26,27,29,30	5.59	1.31
Realism factors	11,12,14,16,17,22	5.32	1.33
Sensory factors	4,5,6,10,14,15,16,17,18,19,20	5.21	1.13
Distraction factors	8,9,24,28	5.20	1.70
Subscale			
Involvement/Control	1,2,5,10,13,14,18,23,25,26,27,32	5.60	1.28
Natural	3,7,12,19	5.46	1.28
Resolution	19,20	5.92	1.03
Interface quality	28,29,30	5.21	1.64

parameters, such as the rotation axis and range of rotation. A last step is to specify the drilling machine behaviour. While the interactable/trackable components are plug-and-play components and independent of the 3D object they are attached to, the machine component has to be designed exclusively for each machine. Nevertheless, although manual effort is required to import the machine 3D objects and tune the parameters, the definition of machine components provides sufficient level of abstraction to facilitate the development of XR tutorials for the majority of industry training applications. **Figure 8** demonstrates the overall view, some steps of the tutorial creation process and their illustration in the 2D panel. **Figure 9** demonstrates some steps of the tutorial execution process, indicated in the 2D panel as successfully completed or pending.

3.2 DHM-Based Simulation and Ergonomics Assessment

The records of the 13 steps comprising the drilling operation, along with the 3D models of the working environment, are used to simulate the training scenario and provide visualization of the whole drilling operation and corresponding data for ergonomics assessment. Some of the tutorial steps produced by the simulation module are illustrated in **Figure 11**.

For biophysics assessment of the aforementioned training scenario we evaluated posture of the DHM by varying the height of the table, where the drilling machine is located. Four values were examined: 0.6, 0.73, 0.85 and 0.97 m. RULA scores were calculated in every simulation frame for both the left and the right part of the body. The presented results correspond to the maximum score value during the execution of each training step, indicating the posture causing the most postural fatigue and discomfort. **Figure 10** illustrates the RULA scores for each selected table height. For the selected anthropometrics of the DHM, the heights 0.73 and 0.85 m resulted in the same score, which was on average smaller (i.e. ergonomically better) than for heights 0.6 and 0.97 m, as shown in the same figure. Nevertheless, the maximum score for table height 0.6 m was smaller than for the other three heights, which exceeded their maximum value at training step 7. The height 0.97 m is considered the ergonomically less efficient, since it leads to a relatively high RULA score for three consecutive steps.

It is therefore excluded in the subsequent biomechanical analysis.

In addition to RULA scores we calculated the energy expenditure following the musculoskeletal modeling approach described in **Section 2.5.2**. The results are shown in **Figure 11** and illustrate the energy in the form of heatmaps in regions of the human body that correspond to the shoulders, elbows, wrists and pelvis. Moreover, we illustrate the dynamically changing energy expenditure (averaged within a sliding window of 1 s) for the execution of the drilling operation. The energy is illustrated in **Figure 12** for the main body joints (pelvis and left/right shoulder) using three different values for the table height. The results indicate that for the utilized human model, the smallest energy is consumed at the pelvis when the highest table is utilized, whereas the shortest table leads to reduced energy for each of the two shoulders, with the right shoulder experiencing higher strains. The selection of the optimal table height obviously depends on multiple personal risk factors for each user, such as any prior injuries on specific body parts. The calculation of temporal graphs illustrating the maximum energy values (in addition to average values) is also possible and provides additional information in respect to the estimation of the highest risk.

4 USER EVALUATION

drilling tutorial execution, described in **Section 3.1**, was evaluated by twenty employees in our laboratory, aged between 23 and 50 years, who voluntarily agreed to participate in evaluation tests. The users have been introduced to the context and goal of the tutorial before using the system. Upon XR training, the users completed the Questionnaire of Presence (QoP) (Witmer and Singer, 1998), the most widely used presence questionnaire for virtual environments, which consists of 32 questions with a seven point Likert Scale. The included questions are associated to consequential factors contributing to the sense of presence, such as *Control Factors (CF)*, *Sensory Factors (SF)*, *Distraction Factors (DF)* and *Realism Factors (RF)*. In order to better assess the results of QoP in respect to familiarization with virtual environments, the users also completed the *Video Game Experience* questions of (Maneuvrier et al., 2020). CF are related to the degree of control the user has over the task environment and affect their immersion. RF are related to the environment realism alongside the consistency of information, and affect mostly the user involvement, since users pay more attention in the virtual environment stimuli. SF and DF refer to the isolation of the user, the sensory modality and consistency of the information received by the user which should optionally describe the same objective world in order to affect both immersion and involvement.

With a mean video game experience 3.25 ± 1.92 the users reported a global mean (total 28 questions without haptics and audio items) of 151.95 ± 16.81 , and a maximum possible score of 196. The mean values and standard deviations of item groups, related to the four aforementioned factors and three subscales of the QoP original analysis, are listed in **Table 3**. We should note that items

related to haptics and sound user experience (no. 6,15,16,17) were not taken into account. While our study achieves better scores than the results presented in (Maneuvrier et al., 2020) (presence global mean score of 101.95 ± 10.38), the subjectivity in the questionnaires' responses does not allow to deduce a definite outperformance. In addition every experiment design focuses on different targets and each user rates his experience differently, according to his standards. Focusing on the sensory factors, which are directly linked to our interaction system, we estimate that a satisfying score was achieved. The majority of the users reported the experience as satisfying, indicating that features of the framework, such as visual indications and comprehensive interactions, composed an easy to follow tutorial. Thirty-five percent of the users faced difficulties at the 10th step of the tutorial where the requested interaction (open a cap) did not follow the natural movement of the hand pose. This might be attributed to the size of the interactable 3D object, which in this case was a small rotating cap. Future work includes the design of more complex finger interactions, such as the pinch interaction, that is expected to solve this issue.

DISCUSSION AND CONCLUSION

Extended reality training allows to fully experience and iterate on a virtual task before committing to the physical workplace, thus alleviating resources and ensuring a safe environment, both for trainees and equipment. Moreover, virtual interactions facilitate the granular adjustment of conditions based on personalized preferences and needs and can also become a valuable means of data gathering for ergonomic assessment and user-centered design. However, the development of XR platforms that can support all different aspects of industrial training within a single system is extremely challenging. Two limitations of our system are discussed next.

First of all, the XRSISE system does not provide an automatic way to model the 3D scene, e.g., using input from depth cameras, but rather provides the authoring tools that allow to add and control properties of the objects, and simulate working tasks and scenarios. For most industrial machines, which are modeled with computer-aided design (CAD) tools, the automatic retrieval of the respective 3D models is easy. However, every-day objects that might be part of a scene and have a role in the machinery operation (such as chairs, boxes, natural obstacles) need to be explicitly modeled using 3D modeling software, and this can be time-consuming. Secondly, while the connection of DHM-based simulation module with the XR training creation module is automated for most common actions, the combination of demanding and parallel actions with complex machine structures may require customization to avoid collisions and unrealistic behavior. Moreover, as the authoring tools provided by XRSISE for 3D models creation are based on Unity's existing game-object logic, the *de novo* design of industrial training applications involving large scenes (e.g., for assembly tasks) would require significant manual effort. For such purposes, 3D models may be retrieved from other sources (such as databases with CAD objects for industrial applications), or 3D scanning technologies may be exploited along with point cloud denoising (Nousias et al., 2021) and semantic segmentation techniques.

Concluding, in this paper we presented the XRSISE framework for the development and the subsequent rollout of new training programs in VR and illustrated its efficacy in a drilling scenario. In XRSISE ergonomic analysis encapsulates, along with standard (RULA) ergonomic risk assessment, biophysics-based computations of mechanical loading on the joints of the human body enabling a dynamic and more comprehensive postural assessment. Users evaluation showed satisfactory perception of the XR training process, but further evaluation is required in order to assess the effectiveness of the virtual training on drilling operation with actual workers in industrial environment. Future work includes also the use of a motion capture vest for the connection of XR-related modules with the biophysics assessment module, that will allow to bypass the need of motion estimation through DHM-based simulation. This will result to better accuracy of the biomechanical analysis based on real personalized data.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

ETHICS STATEMENT

The study involving human participants was reviewed and approved by the Research Ethics Committee of the University of Patras. Data were collected in an anonymous way. Written informed consent for participation was not required for this study.

AUTHOR CONTRIBUTIONS

Conceptualization, MP, DL, EZ, and KR; Investigation and formal analysis, DL, MP, KR, and EZ.; Methodology, MP, DL, and KR; Supervision, EZ and KM; Visualization, MP and DL; Writing "original draft, all authors; Writing" revision, all authors.

FUNDING

This work has been supported by the EU Horizon2020 funded project "Smart, Personalized and Adaptive ICT Solutions for Active, Healthy and Productive Aging with enhanced Workability (Ageing@Work)" under Grant Agreement No. 826299.

ACKNOWLEDGMENTS

The authors would like to thank their technical partners from Siemens (from Ageing@Work project) for providing material and instructions on the drilling operation. They also thank their colleague Dimitris Bitzas for the conceptual design and initial prototype of the Unity-based simulation tools.

REFERENCES

- Ahmed, S., Gawand, M. S., Irshad, L., and Demirel, H. O. (2018). "Exploring the design space using a surrogate model approach with digital human modeling simulations," in *International design engineering technical conferences and computers and information in engineering conference*, New York; NY American Society of Mechanical Engineers (ASME), 51739, V01BT02A011.
- Aristidou, A., and Lasenby, J. (2011). Fabrik: a fast, iterative solver for the inverse kinematics problem. *Graph. Models* 73, 243–260. doi:10.1016/j.gmod.2011.05.003
- Azizi, A., Yazdi, P. G., and Hashemipour, M. (2019). Interactive design of storage unit utilizing virtual reality and ergonomic framework for production optimization in manufacturing industry. *Int. J. Interact. Des. Manuf.* 13, 373–381. doi:10.1007/s12008-018-0501-9
- Caputo, F., Greco, A., Egidio, D., Notaro, I., and Spada, S. (2017). "A preventive ergonomic approach based on virtual and immersive reality," in *International conference on applied human factors and ergonomics*. Berlin, Germany: Springer, 3–15.
- Chaffin, D. B. (2007). Human motion simulation for vehicle and workplace design. *Hum. Factors Man.* 17, 475–484. doi:10.1002/hfm.20087
- Chaffin, D. B., and Nelson, C. (2001). *Digital human modeling for vehicle and workplace design*. Warrendale, PA: Society of Automotive Engineers.
- Community, B. O. (2018). *Blender - a 3D modelling and rendering package*. Amsterdam, Netherlands: Blender Foundation, Stichting Blender Foundation.
- Cremer, J., Kearney, J., and Papelis, Y. (1995). Hcsm: a framework for behavior and scenario control in virtual environments. *ACM Trans. Model. Comput. Simul.* 5, 242–267. doi:10.1145/217853.217857
- Dias Barkokebas, R., Ritter, C., Sirbu, V., Li, X., and Al-Hussein, M. (2019). "Application of virtual reality in task training in the construction manufacturing industry," in *Proceedings of the 2019 international symposium on automation and robotics in construction (ISARC)*, Banff, Canada, May 21–24, 2019. doi:10.22260/ISARC2019/0107
- Docs, U. (2020). XR interaction toolkit 1.0.0. Available at: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@1.0/manual/index.html> (Accessed December 16, 2020).
- Duffy, V. G. (2008). *Handbook of digital human modeling: research for applied ergonomics and human factors engineering*. Boca Raton: CRC Press.
- Gerbaud, S., Mollet, N., Ganier, F., Arnaldi, B., and Tisseau, J. (2008). "Gvt: a platform to create virtual environments for procedural training," in *2008 IEEE virtual reality conference*, Reno, NV, March 8–12, 2008. 225–232. doi:10.1109/VR.2008.4480778
- Gonzalez-Franco, M., Pizarro, R., Cermeron, J., Li, K., Thorn, J., Hutabarat, W., et al. (2017). Immersive mixed reality for manufacturing training. *Front. Robot. AI* 4, 3. doi:10.3389/frobt.2017.00003
- Karhu, O., Kansil, P., and Kuorinka, I. (1977). Correcting working postures in industry: a practical method for analysis. *Appl. Ergon.* 8, 199–201. doi:10.1016/0003-6870(77)90164-8
- Maneuverier, A., Decker, L. M., Ceyte, H., Fleury, P., and Renaud, P. (2020). Presence promotes performance on a virtual spatial cognition task: impact of human factors on virtual reality assessment. *Front. Virtual Real.* 1, 16. doi:10.3389/frvir.2020.571713
- Manual, U. L. (2020). XR plug-in framework. Available at: <https://docs.unity3d.com/Manual/XRPluginArchitecture.html> (Accessed November 24, 2020).
- McAtamney, L., and Corlett, E. N. (1993). Rula: a survey method for the investigation of work-related upper limb disorders. *Appl. Ergon.* 24, 91–99. doi:10.1016/0003-6870(93)90080-s
- McAtamney, L., and Hignett, S. (2004). "Rapid entire body assessment," in *Handbook of human factors and ergonomics methods*. (Boca Raton: CRC Press), 97–108.
- Mortensen, J., Trkov, M., and Merryweather, A. (2018). "Improved ergonomic risk factor assessment using opensim and inertial measurement units," in *2018 IEEE/ACM international conference on connected health: applications, systems and engineering technologies (CHASE)*, Washington, DC, September 26–28, 2018, (IEEE), 27–28.
- Nousias, S., Arvanitis, G., Lalos, A. S., and Moustakas, K. (2021). Fast mesh denoising with data driven normal filtering using deep variational autoencoders. *IEEE Trans. Ind. Inf.* 17, 980–990. doi:10.1109/TII.2020.3000491
- Risvas, K., Pavlou, M., Zacharaki, E. I., and Moustakas, K. (2020). "Biophysics-based simulation of virtual human model interactions in 3d virtual scenes," in *IEEE conference on virtual reality and 3D user interfaces abstracts and workshops (VRW)*, Atlanta, GA, March 22–26, 2020. IEEE, 119–124.
- Rott, J., Weixler, J., Rabl, A., Sandl, P., Wei, M., and Vogel-Heuser, B. (2018). "Integrating hierarchical task analysis into model-based system design using airbus xhta and ibm rational rhapsody," in *2018 IEEE international conference on industrial engineering and engineering management (IEEM)*, Bangkok, December 16–19, 2018, IEEE.
- Seth, A., Hicks, J. L., Uchida, T. K., Habib, A., Dembia, C. L., Dunne, J. J., et al. (2018). Opensim: simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. *PLoS Comput. Biol.* 14, e1006223–20. doi:10.1371/journal.pcbi.1006223
- Sreekanta, M. H., Sarode, A., and George, K. (2020). "Error detection using augmented reality in the subtractive manufacturing process," in *2020 10th annual computing and communication workshop and conference*, Las Vegas, NV, January 6–8, 2020 CCWC. doi:10.1109/CCWC47524.2020.9031141
- Stanton, N. A. (2006). Hierarchical task analysis: developments, applications, and extensions. *Appl. Ergon.* 37, 55–79. doi:10.1016/j.apergo.2005.06.003
- Stanton, N., Salmon, P. M., and Rafferty, L. A. (2013). *Human factors methods: a practical guide for engineering and design*. London: CRC Press
- Tanriverdi, V., and Jacob, R. J. (2001). "Vrid: a design model and methodology for developing virtual reality interfaces," in *Proceedings of the ACM symposium on virtual reality software and technology*. (New York, NY: Association for Computing Machinery), VRST '01, 175–182. doi:10.1145/505008.505042
- Valve (2020). Steamvr unity plugin. Available at: https://valvesoftware.github.io/steamvr_unity_plugin/ (Accessed November 24, 2020).
- Wang, P., Bai, X., Billingham, M., Zhang, S., Han, D., Lv, H., et al. (2019). "An mr remote collaborative platform based on 3d cad models for training in industry," in *IEEE international symposium on mixed and augmented reality adjunct*, Beijing, China, October 10–18, 2019, ISMAR-Adjunct, 91–92. doi:10.1109/ISMAR-Adjunct.2019.00038
- Winter, D. (2009). *Biomechanics and motor control of human movement*. New Jersey: Wiley.
- Witmer, B. G., and Singer, M. J. (1998). Measuring presence in virtual environments: a presence questionnaire. presence: teleoper. *Virtual Environ.* 7, 225–240. doi:10.1162/105474698565686

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2021 Pavlou, Laskos, Zacharaki, Risvas and Moustakas. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.