Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 11 (2017) 1279 – 1287

Procedia MANUFACTURING

27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy

Human-machine collaboration in virtual reality for adaptive production engineering

Andrea de Giorgio*, Mario Romero, Mauro Onori, Lihui Wang

KTH Royal Institute of Technology, Brinellvägen 68, SE-100 44, Stockholm, Sweden

Abstract

This paper outlines the main steps towards an open and adaptive simulation method for human-robot collaboration (HRC) in production engineering supported by virtual reality (VR). The work is based on the latest software developments in the gaming industry, in addition to the already commercially available hardware that is robust and reliable. This allows to overcome VR limitations of the industrial software provided by manufacturing machine producers and it is based on an open-source community programming approach and also leads to significant advantages such as interfacing with the latest developed hardware for realistic user experience in immersive VR, as well as the possibility to share adaptive algorithms. A practical implementation in Unity is provided as a functional prototype for feasibility tests. However, at the time of this paper, no controlled human-subject studies on the implementation have been noted, in fact, this is solely provided to show preliminary proof of concept. Future work will formally address the questions that are raised in this first run.

Peer-review under responsibility of the scientific committee of the 27th International Conference on Flexible Automation and Intelligent Manufacturing

Keywords: Virtual Reality; Augmented Reality; Unity Game Engine; Human-Robot Collaboration; Industry 4.0; Robotics; Adaptive Production.

* Corresponding author. Tel.: +46-8790-9065.
E-mail address: andreadg@kth.se.

Peer-review under responsibility of the scientific committee of the 27th International Conference on Flexible Automation and Intelligent Manufacturing
doi:10.1016/j.promfg.2017.07.255

1. Introduction

Virtual reality (VR) is not a new concept, but some VR technologies that have recently been introduced for the gaming industry are quite innovative and manufacturing industry is starting to test them, and eventually adopt them. This paper investigates the relevance of an approach that considers both new software and hardware solutions. Using software that has been developed for a different use may be deceptive and lead to unsolved problems. The aim of this paper is, however, not to solve all problems at once, but to address advantages and disadvantages, together with the open questions that arise by proceeding toward one promising direction of research, that is using game engines as simulation software to control AR/VR hardware and industrial machines.

A special focus is placed on human-robot collaboration (HRC) opportunities with VR, which the authors find promising premises for a physical approach of HRC with augmented reality (AR). In fact, when HRC is increasing daily and the industry is moving from mass production to personalized production, looking for new software solutions based on a large community of developers, such as game developers, may prove to be useful not only for industrial production simulation but also for an adaptive improvement of the production process itself for both machines and humans. The advantages given by an open-source and cross-producers development approach are self-evident. Just to give an example, machine learning algorithms could play a main role in adaptations of a given process when shared in form of add-on libraries (which may be further enhanced through cloud-based solutions).

This paper also proposes a simple implementation in Unity of a collaborative robot manipulator as a prototype that allows a human operator to interact with the VR environment through HTC Vive virtual reality tracked headset and hand controllers. Advantages and disadvantages of this approach are discussed and will be fully validated in future work.

1.1. Virtual reality

Virtual reality has been a research topic for a long time. The term VR itself was used for the first time by Jaron Lanier back in 1989 [1]. In 1992 the Commission of European Communities (CEC) recognized VR as a viable technology to be included in future calls for proposals [2]. Since then, hundreds of research projects have explored its potential use in different disciplines, but only in 2012 the Kickstarter project named "Oculus Rift" could bring it to the general public by raising funds and developing an affordable high-quality Head Mounted Display (HMD). The popularity of HMDs has started what has been called a second wave of VR, that is still propagating through academic and industrial researchers and the gaming community [3].

The latest technology has generated HMDs such as the already mentioned Oculus Rift, now at its second version, the HTC Vive, the PlayStation VR, the Open Source Virtual Reality (OSVR) or Zeiss VR One, plus a series of mobile HMDs that enhance mobile phone screens into HMD devices for VR such as Google Cardboard, Samsung GearVR Innovator Edition and Gameface Mark IV. This means that immersive VR has finally become accessible to everyone. Together with HMDs, there exist input/output (I/O) devices that can be adopted for almost any kind of user interactions in VR: from haptic devices to tracking devices, controllers and depth cameras.

Another essential part of VR, together with the hardware described above, is the software that enables such paradigm to be experienced. The main research in this topic has been formulated around the use of Computer Aided Design (CAD) to reproduce real world items as models in a virtual environment or cyber workspace [4]. However, reproductions of real objects as computer models do not qualify yet as VR. The determinant factor is the possibility to exploit the virtual representation for a simulation. In fact, a comprehensive definition of VR is given as "computer-generated simulations of three-dimensional objects or environments with seemingly real, direct, or physical user interaction" [5]. This is where the various roads start diverging from one another: gaming industry has focused on virtual worlds, so that players could interact with them by controlling an avatar, while general purpose industrial and academic research has focused on simulation software, in which the users can experience the VR as a mean to visualize the results of such simulations.

1.2. Human-machine collaboration in virtual reality

Human-Robot Collaboration (HRC) in production engineering is a research topic for which Augmented Reality (AR) and VR have provided interfaces that, respectively, expands the quantities of features that operators can watch in their field of view [6] or replace it completely with a virtual world. Typical industrial production applications span from manufacturing process simulation [7], [8], which are able to provide real time enhanced information used for inspection [9] or with focus on training [10], [11], to collaborative factory planning during product design [12] or re-design [13]. In fact, VR can be used for collaborative (re)designing of production systems when analyzing and evaluating changes prior to implementation. This makes possible to prevent costly design mistakes [13].

Even though some works have considered the human factor as part of the industrial process and adjusted the VR to accurately include the operator movements in the simulation [10], [14], [15], it is often the case that the operator experiences the VR/AR only from a static position, e.g. standing still or seated, where the input sensors are located. Another approach is to immerge the operator in a cave [16], a projected VR wall, a space that is obviously limited by physical constraints. The gaming industry on the other hand has worked on methods to embody the sensors and free the player from physical constraints in the VR/AR environment, providing an enhanced sense of immersion: the ability to move in the virtual world and interact with anything that appears around the player. Even large collaborative teams with one or more operators wearing the VR HMDs are supported. Gaming hence becomes the field in which innovation in AR/VR is being heavily pushed, especially thanks to the large number of end-users, or players, who are also very prone to accept beta versions and become early testers in order to experience it first. A similar innovation in industry requires years of testing, changes of industrial standards, replacement of machines and upgrade of factory design, not to mention costly investments and management decisions.

1.3. A production engineering perspective on virtual reality

The aim of this article is to explore the latest advancements in the gaming industry that can be adopted in production engineering in order to overcome the main limitations of the industrial software provided by manufacturing machine producers, with a particular focus on VR immersive applications based on an open-source community approach.

The remainder of this paper is divided into four sections. In section 2, related work is presented. The ideal steps toward the use of game engines in production engineering are presented in section 3. A practical application is presented and discussed in section 4. Finally, in sections 5, conclusions and future work.

2. Related work

While most of the related work on VR and HRC for production engineering has been cited in the introduction in order to pose the groundwork for this paper, the majority of them share hardware that comes from the gaming industry, but never simulation software. Only few cases include hybrid approaches, so they are mentioned below.

"BeWare of the robot" is a Virtual Reality Training System (VRTS) developed in Unity™ game engine platform [10] that simulates a shop-floor environment accessible through an HMD, also using a Microsoft Kinect™ sensor to capture the operator's movements and virtualize them. An avatar is used to render the operator's body in the virtual environment.

Unity™ has also been used, together with the Robot Operating System (ROS) [17], as middleware for immersive VR teleoperation by driving a mobile robot [18] or as real-time simulator for multi-unmanned aerial vehicle local planning [19], [20], therefore approaching the use industrial robot simulations from the gaming perspective.

3. The contribution of game engines toward human-machine collaboration in virtual reality

3.1. Key elements for immersive VR collaboration

The advantage of an immersive VR is given by the focus and longer attention. For example, a study reveals how low spatial ability learners are more positively affected by VR learning [21]. The key elements to obtain an immersive VR experience are the following [22]:

A virtual world. A collection of objects in a space governed by rules and relationships, where the objects are CAD models of industrial machines and other industrial equipment that completes a factory scene, the rules are defined using classes in Unity, which are called GameObjects, and the relationships are defined by components and scripts that can be attached to any GameObjects. All together these objects form a virtual world.

Immersion. It refers to the status of the industrial operator being involved in an activity within a virtual space, to an extension that their mind is separated from the physical space they are living in. A game engine such as Unity provides full support for HMD devices making the full immersive modality a default setting for the simulation.

Feedback. This element gives the operator the ability to observe the results (outputs) of their activities (input) in the VR. The standard feedback with HMD devices is visual/auditory, but it has also become common the use of haptic devices that can provide a sense of touch [23]. Taste and smell remain difficult to explore.

Interactivity. The ability to interact with the virtual world is fundamental. Sensors and devices allow to capture the operator's body actions and transform them into virtual actions. Navigation and direct manipulation are the two main aspects of an immersive virtual reality. HMDs such as the HTC Vive™ are often sold in bundle with some controllers that operators can hold in their hands. Voice and gesture commands can also be captured by microphones and cameras mounted in the environment.

Participants. Human operators are an essential element of the VR experience. They can be grouped by experience and offered a different VR representation based on their capacity to interact with the virtual objects. A good advantage of the VR is that it allows an operator to be trained simply by using it. This is achieved by structuring the virtual world in different levels that are called, in turn, each time that the operator acquires enough knowledge to perform more complex operations. Exactly as a player advances level by level in a videogame.

3.2. Objects representation with game engines

CAD-VR data exchange is an important issue brought up by the VR community because CAD systems used by the industry to develop their product models are in most cases unsuitable for producing optimal objects representations for VR simulations. In fact, VR graphic engines make use of scene-graphs for visualization, e.g. Openscenegraph, OpenSG or OpenGL Performer, which are hierarchical data structures such as triangulated mesh geometry, spatial transforms, lighting, material properties, etc. and the scene-graphs renderers provide methods to exploit this data structure at interactive frame rates. Converting CAD data into a scene graph consists of producing several polygonal representations of each part and during this translation process, the parametric information of the CAD model and pre-existing texture maps generally do not even get imported into the VR application. In virtual assembly simulations there are generally two representations of the same model: one for visualization and another for constraint modeling algorithms that are used to perform the assembly task; these are unified in the game engine under prefab objects that can be easily instantiated and destroyed using object programming. Similarly, physics modeling applications also use dual model representations: high-fidelity model for visualization and a coarser representation used for interactive physics calculations; the most important and challenging task is the improvement of the physical simulations [24] which can be aided by the use of game engines.

Even if the conversion of CAD models threatens to slow-down processes, advantages are gained by using the game engines as VR software, because they consists of both physics and visualization simulators which are integrated and optimized for a realistic user experience.

3.3. Open community approach

A peculiar aspect of the game engines is that they come with a set of tools that are developed and constantly improved with the help of a community of end-users. All the objects that are collected or created by one user, can be shared in online libraries, e.g. the Unity Assets Store. This allows other users to quickly reuse and eventually redesign existing solutions instead of implementing them from scratch.

Having large libraries of existing solutions available in production engineering has the potential of making the design and development of new processes easier, not to mention facilitating the knowledge sharing.

3.4. Remote collaborative tasks

Another interesting application that can be enhanced by VR experience is remote collaboration. It is known that 3D models have been used to guide remote robotic assemblies [25], and collaborative robot monitoring promises an increased sustainability over the manufacturing process [26], [27]. VR can bring this forward by making the interface for remote control a telepresence immersive application, where all the advantages of the gaming style can be exploited to give the operator a realistic experience and full control of their remote actions.

A promising technology that could contribute to this is the ability to capture and stream a real object representation through a point cloud in the VR. Because the streamed point cloud is lighter than a full video stream it helps to keep the number of transmitted frames per second high, even with a slow connection [25]. A point can be modelled with scripts in the game engine to represent objects that enter the operator's virtual world making the virtual telepresence realistic and giving enough feedback to guide simultaneous responses of the operator.

4. A practical application: collaborative robot manipulator in virtual reality

In order to showcase the possible advantages of using a game engine as simulation software for both an industrial machine and the VR environment, a practical application has been designed as a functional prototype. The application has been used for a preliminary pilot study and will be improved and verified by means of controlled human-subject studies in our future work.

The project aims to reproduce an ABB IRB 120 robot manipulator in the VR that can be moved by the operator wearing an HMD with the goal of performing a simple collaborative assembly task. The choice for the HMD fell on an HTC Vive™ virtual reality tracked headset, together with its hand controllers because of the high tracking accuracy and the state-of-the-art VR display.

The virtual ABB IRB 120 robot manipulator used in the simulations (see fig. 1 left) is controlled through an algorithm imported in Unity as an asset, that is a hybridization of genetic algorithms and particle swarm optimization for inverse kinematics [28]. It allows virtual arms, composed by any number of joints, to be animated with natural human-like movements. The end effector (EE) of the robot manipulator is tied to the spatial position and orientation of the controller that the operator holds in the real world. Once the operator moves their arm, the position of the EE is recomputed and through the algorithm, together with all the other joint positions, so that the movement leading to the new position appears smooth and as natural as possible. If the robot manipulator was a human arm, even though with a different number of degrees of freedom (DOF), its movement would look as close as possible to a human action.

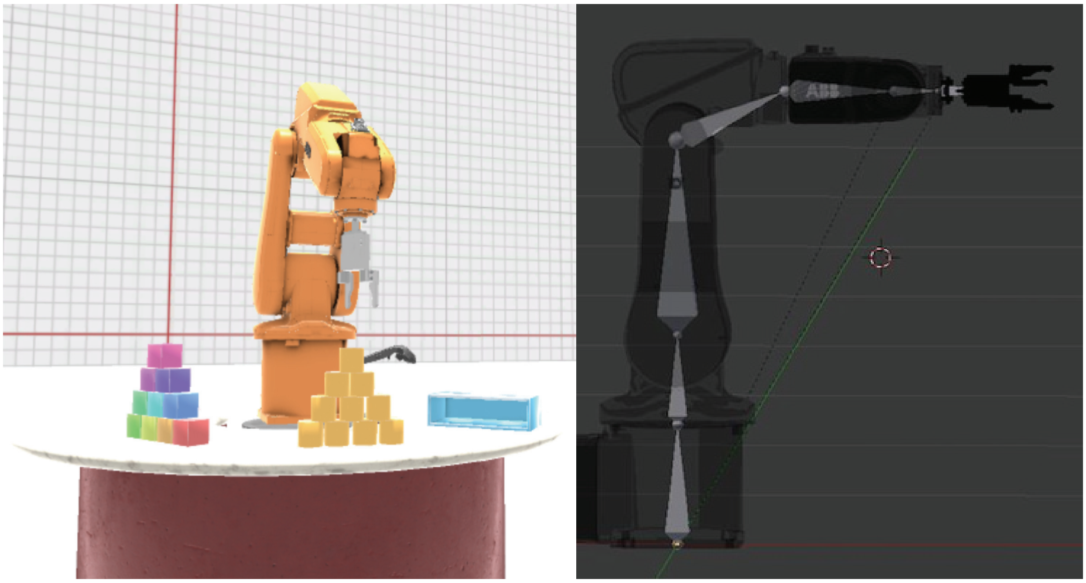

Fig. 1. (left) Simulated robot manipulator. The robot manipulator, specifically an ABB IRB 120, is simulated in the virtual reality using Unity. (right) Joint chain. The joints of the virtual robot manipulator are attached to a joint chain in Unity that allows to define the kinematics.

Unity has a built-in kinematics system, but it is dedicated and limited to the simulation of humanoids, i.e. human-like characters. Anyhow, Unity can do half the job by easily defining a structure for the joints to move, namely a joint chain, as shown in fig. 1 (right). The joint chain poses a structure for the CAD model parts of the robot manipulator and allows the attached script to act on it. Unity also provides some basic settings for each joint that can be used as default when the dynamics are not assigned to an external script. Joint limitations are defined in Unity and maximum speeds are defined at script level. The latter choice is due to the quality of the dynamics that is negatively affected by gaming optimizations that are included in Unity to make the simulation of humanoids more realistic. For this reason, it is recommendable to use external scripts for both the kinematics and the dynamics of robot manipulators or any machines composed of multiple joints with a high number of DOF.

As stated earlier, the operator's arm movements do not correspond to a joint-joint mapping with the robot manipulator movements. Instead, the operator suggests the desired EE position for the robot manipulator by holding the controller in their hand and the computation is left to the kinematics algorithm that solves the following issues:

- Evaluate the relative EE position corresponding to the operator's arm extension from the headset;
- Adapt the robot space to the reachability of the user arm extension, so that the maximum reachable distance by the robot is comparable to the maximum extent of the operator's arm;
- Find a way to let the robot move to positions which the operator's arm cannot easily reach by smoothing the robot movements over a user quick change of pose to increase their spread.

The first problem is solved by delivering tactile feedback to the user in the form of vibration whenever the Inverse Kinematic (IK) system is unable to reach the target position.

In order to solve the second problem, valid trajectories are calculated between the solutions provided by the IK system. This might give the user the impression that the robot makes unnecessary or automatic movements, but the robot simply changes the pose to avoid singularities and so circumvent the joint limitations.

The third and final problem is solved by allowing the user, rather than the robot, to reposition themselves. The chosen interaction mode thus becomes a direct control by relative positions on command: the robot follows the relative movement of the controller, but only when the operator is pressing a trigger button on the controller in their hand.

A tradeoff between speed and accuracy must be chosen. Repositioning of either the robot or the user leads to some difficulty in keeping a stable position of the EE, especially when performing precise actions. Slowing down the robot motion attenuates the effect of subtle operator movements so it increases the robot accuracy. Conversely, using a full speed robot could be interesting for the operator to experience, but makes the control very hard to manage. Different speeds can also be set by using a variable actuator such as a pressure sensor on the controller instead of a binary on/off trigger button. This regulates the speed based on the operator's needs.

Direct control of the robotic arm has been difficult when all the DOF were available, that is because the control is not performed joint to joint, as there is no correspondence between human and robotic arm in terms of DOF. The application became much more usable, in particular for novices, when the wrist (last three joints) was locked to a specified direction, for example to point down where the target for grabbing was located. So, it turns out that one effective trick to improve control consists of assigning a trigger button to lock/unlock the position of the three last joints. This helps to keep the orientation of the EE fixed, allowing the operator to easily perform precise movements such as insertions and grabbing.

The overall VR interaction with the robot manipulator has been divided into three levels: posing, recording and playing. In posing mode, the operator guides the robot manipulator freely toward a certain pose. Once the recording mode is started, all the movements are saved as a trajectory. When the playing mode is active the robot follows its recorded trajectory in loop, independently of the operator movements, allowing them to assist to the process but not interfering with it.

It is interesting to observe the advantage brought by the user perspective in VR. The difference between a first person interaction with the robotic manipulator and the use of a teach pendant is that VR allows for the space coordinates to be aligned with the relative position of the operator instead of the robot manipulator origin in the space. For example, in VR, every time that an operator moves the End Effector (EE) of the robot manipulator with a gesture to his right, "right" is interpreted as the space direction to the right side of the operator's gaze. This means that "right" is a unitary vector constantly updated with the operator's gaze movements and the robotic movement corresponding to the command is perfectly intuitive.

On the other hand, the teach pendant will always interpret a movement of the control pad joystick to the right, as a movement in a specific direction of the robot manipulator axes, independently of the position of the operator. This method requires the operator to be aligned to the robot axes in order to make sense of the commands, or even to perform a mental transformation of the desired movement direction into the corresponding joystick direction that would perform such movement. Either the operator knows a priori the transformation, or it will be necessary to guess it with some test movements, therefore slowing down the operator's job with the machine.

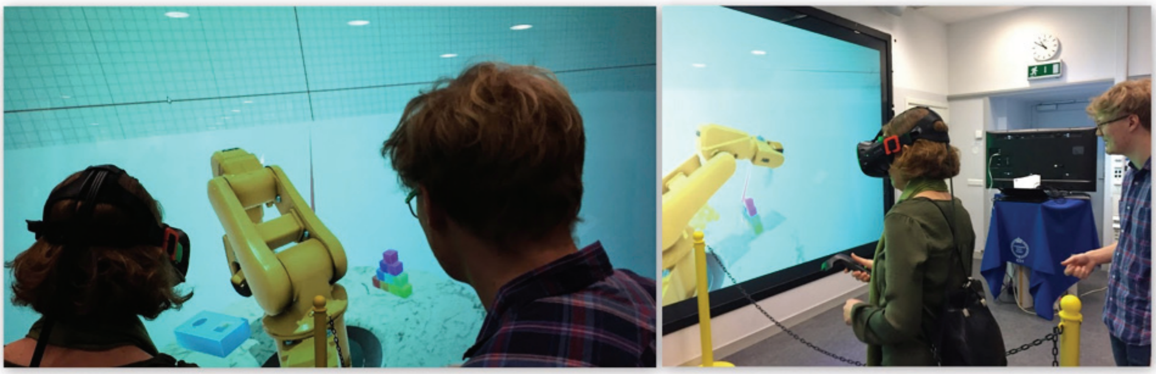

Fig. 2. Assembly task as a game. The virtual ABB IRB 120 robot manipulator is presented as an assembly game where students can test their skills as operators. The student controlling the movement wears the head mounted display for VR and holds a controller in their arm. The robot manipulator end effector follows the movement and direction of the controller, adapting to the closest natural movement allowed by its six DOF. Other students can watch the scene from the operator's perspective on the external monitor.

5. Conclusions and future work

The practical application, although very simple and meant as an informal pilot study, has led to observations that look promising for production engineering simulations based on VR. For example, it helped in understanding how to manage the correspondence of different degrees of freedom when controlling a robotic arm with a human arm. The first user perspective in the control of machines makes it very different from a typical teach pendent activation of the actions. Advantages can be seen on the ability for the operator to embody the machines and learn “on their skin” how to perform the manufacturing process. Therefore, HRC assumes a different and far richer form already in VR, even though only with an AR application the operator can obtain a full physical HRC.

It is worth considering the advantage of using open libraries for assets, including scripts. As seen in this paper, each feature in the game engines can be shared as assets which include CAD models, materials, textures, scripts, renderings, sound effects, animations, etc. If features such as CAD models have been shared for years between production software users, new possibilities arise when sharing scripts in simulation software. This is not a new idea since programmers already share portions of code through specific websites, e.g. GitHub, SourceFourge or BitBucket, but new is the fact that scripts can be shared as plug-and-play items that can be directly attached to a VR environment in a simulation software and they can be shared together with the object that is affected by the script. They will be jointly loaded by the simulator and ready to be used. This corresponds to an open community for industrial applications that uses shared tools with customized behaviors, running in the same simulation software.

The approach presented is destined to face great challenges in order to make the gaming software fully compatible with the software standards needed in production engineering simulation. However, the simple application presented, together with the outlined advantages, encourages further studies in such direction.

The future work that will be carried out includes several open questions presented in this paper, including a whole new set of possibilities. For example, when an industrial machine is modeled as a prefab in a game engine such as Unity, it can be exported as an asset, which includes both the CAD parts and the scripts that regulate the machine complete behavior in the VR. Adaptive production planning could exploit such scripts to simulate a quickly adaptable design to the given manufacturing task.

Fundamental for the remote control of the machines is to ensure that actions in the VR can correspond to real world actions. For example, a mobile robot manipulator could take the place of the operator who is interacting with the VR environment and perform their actions as output in the real world, especially if in a remote location.

More advanced practical cases will be designed and developed to apply and further test the observations and open questions posed by this paper.

Acknowledgements

The authors would like to thank Andreas Linn, Haisheng Yu, Lisa Schmitz, Mathilde Caron and Rodrigo Roa Rodríguez for their contribution through the development of the virtual robot manipulator application [29].

References

[1] H. Rheingold, *Virtual reality: The Revolutionary Technology of Computer-Generated Artificial Worlds - and How It Promises to Transform Society*, First edit. Summit Books, 1991.
[2] J. Encarnacao, M. Gobel, and L. Rosenblum, “European activities in virtual reality,” *IEEE Comput. Graph. Appl.*, vol. 14, no. 1, pp. 66–74, Jan. 1994.
[3] C. Anthes, R. J. Garcia-Hernandez, M. Wiedemann, and D. Kranzlmuller, “State of the art of virtual reality technology,” in *2016 IEEE Aerospace Conference*, 2016, pp. 1–19.
[4] L. Wang, B. Wong, W. Shen, and S. Lang, “A Java 3d-enabled cyber workspace,” *Commun. ACM*, vol. 45, no. 11, pp. 45–49, Nov. 2002.
[5] J. D. N. Dionisio, W. G. B. III, and R. Gilbert, “3D Virtual worlds and the metaverse,” *ACM Comput. Surv.*, vol. 45, no. 3, pp. 1–38, Jun. 2013.
[6] W. Schreiber, T. Alt, and M. Edelmann, “Augmented reality for industrial applications—a new approach to increase productivity,” *Proc.*, 2002.
[7] T. S. Mujber, T. Szecsi, and M. S. J. Hashmi, “Virtual reality applications in manufacturing process simulation,” *J. Mater. Process. Technol.*, vol. 155, pp. 1834–1838, 2004.
[8] D. V. Dorozhkin, J. M. Vance, G. D. Rehn, and M. Lemessi, “Coupling of interactive manufacturing operations simulation and immersive

virtual reality," *Virtual Real.*, vol. 16, no. 1, pp. 15–23, Mar. 2012.

[9] J. Zhou, I. Lee, B. Thomas, R. Menassa, A. Farrant, and A. Sansome, "Applying spatial augmented reality to facilitate in-situ support for automotive spot welding inspection," in *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry - VRCAI '11*, 2011, p. 195.

[10] E. Matsas and G.-C. Vosniakos, "Design of a virtual reality training system for human–robot collaboration in manufacturing tasks," *Int. J. Interact. Des. Manuf.*, pp. 1–15, Feb. 2015.

[11] F. Lin, L. Ye, V. G. Duffy, and C.-J. Su, "Developing virtual environments for industrial training," *Inf. Sci. (Ny).*, vol. 140, no. 1, pp. 153–170, 2002.

[12] N. Menck, X. Yang, C. Weidig, P. Winkes, C. Lauer, H. Hagen, B. Hamann, and J. C. Aurich, "Collaborative Factory Planning in Virtual Reality," *Procedia CIRP*, vol. 3, pp. 317–322, 2012.

[13] E. Lindskog, J. Vallhagen, and B. Johansson, "Production system redesign using realistic visualisation," *Int. J. Prod. Res.*, vol. 55, no. 3, pp. 858–869, 2016.

[14] S. Qiu, X. Fan, D. Wu, Q. He, and D. Zhou, "Virtual human modeling for interactive assembly and disassembly operation in virtual reality environment," *Int. J. Adv. Manuf. Technol.*, vol. 69, no. 9–12, pp. 2355–2372, Dec. 2013.

[15] X. Wang, S. K. Ong, and A. Y. C. Nee, "Real-virtual components interaction for assembly simulation and planning," *Robot. Comput. Integr. Manuf.*, vol. 41, pp. 102–114, 2016.

[16] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, "Surround-screen projection-based virtual reality," in *Proceedings of the 20th annual conference on Computer graphics and interactive techniques - SIGGRAPH '93*, 1993, pp. 135–142.

[17] M. Quigley, K. Conley, B. Gerkey, and J. Faust, "ROS: an open-source Robot Operating System," *ICRA Work.*, 2009.

[18] R. Codd-Downey, P. M. Forooshani, A. Speers, H. Wang, and M. Jenkin, "From ROS to unity: Leveraging robot and virtual environment middleware for immersive teleoperation," in *2014 IEEE International Conference on Information and Automation (ICIA)*, 2014, pp. 932–936.

[19] Y. Hu and W. Meng, "ROSUnitySim: Development and experimentation of a real-time simulator for multi-unmanned aerial vehicle local planning," *Simulation*, vol. 92, no. 10, pp. 931–944, Oct. 2016.

[20] W. Meng, Y. Hu, J. Lin, F. Lin, and R. Teo, "ROS+unity: An efficient high-fidelity 3D multi-UAV navigation and control simulator in GPS-denied environments," in *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, 2015, pp. 002562–002567.

[21] E. A.-L. Lee and K. W. Wong, "Learning with desktop virtual reality: Low spatial ability learners are more positively affected," *Comput. Educ.*, vol. 79, pp. 49–58, 2014.

[22] M. A. Muhanna, "Virtual reality and the CAVE: Taxonomy, interaction challenges and research directions," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 27, no. 3, pp. 344–361, 2015.

[23] D. Wang, Y. Zhang, W. Zhou, H. Zhao, and Z. Chen, "Collocation Accuracy of Visuo-Haptic System: Metrics and Calibration," *IEEE Trans. Haptics*, vol. 4, no. 4, pp. 321–326, Oct. 2011.

[24] B. Frohlich, H. Tramberend, A. Beers, M. Agrawala, and D. Baraff, "Physically-based manipulation on the Responsive Workbench," in *Proceedings IEEE Virtual Reality 2000 (Cat. No.00CB37048)*, pp. 5–11.

[25] L. Wang, A. Mohammed, and M. Onori, "Remote robotic assembly guided by 3D models linking to a real robot," *CIRP Ann. - Manuf. Technol.*, vol. 63, no. 1, pp. 1–4, 2014.

[26] L. Wang, "Collaborative robot monitoring and control for enhanced sustainability.," *Int. J. Adv. Manuf.*, 2015.

[27] L. Wang, P. Orban, A. Cunningham, and S. Lang, "Remote real-time CNC machining for web-based manufacturing," *Robot. Comput. Integr. Manuf.*, vol. 20, no. 6, pp. 563–571, 2004.

[28] S. Starke, N. Hendrich, S. Magg, and J. Zhang, "An Efficient Hybridization of Genetic Algorithms and Particle Swarm Optimization for Inverse Kinematics."

[29] A. Linn, H. Yu, L. Schmitz, M. Caron, and R. Roa Rodríguez, "ViRobot." [Online]. Available: https://rodrigoroarodriguez.github.io/viRobot/.