

Evaluating Commodity Hardware and Software for Virtual Reality Assembly Training

Emma R. Dodoo^b, Brittney Hill^a, Austin Garcia^c, Adam Kohf^a, Anastacia MacAllister^c, Jonathan Schlueter^c, Eliot Winer^c; ^aGeorgia State University – Atlanta, GA; ^bPennsylvania State University – State College, PA; ^cIowa State University – Ames, IA

Abstract

Assembling specialized manufactured equipment, like aircraft, requires advanced production skills that can take years of training and experience to master. Training new workers is often labor intensive and expensive for specialized manufacturing companies. Traditionally, product assembly training in the manufacturing industry predominantly focuses on methods such as textbook learning, and more recently, video guidance. Recent technological advances in Virtual Reality (VR) devices, however, have introduced technology with the potential to improve the current training system. Studies show that VR, training can decrease assembly errors, production cost, and time. Unfortunately, in the past these VR devices were too expensive and required extensive programming knowledge to create a training application. The release of commercial virtual reality (VR) head mounted displays (HMD) and easy to use game engines like Unity 3D has taken steps towards solving this issue. However, because of the recentness of virtual reality's commercial availability, research on training interfaces in manufacturing environments is limited. This paper develops a prototype training system to test the viability of using a VR HMD as an assembly training tool. The hope moving forward is that, as this technology matures, these tools and lessons learned can be used to improve the training process.

Introduction

Virtual reality (VR) is a thriving industry with technology that is consistently increasing in popularity. As a result, VR has been explored for use in a variety of applications like rapid prototyping, manufacturing, scientific visualization, engineering, and education [1]. With a projected growth of eight billion USD in revenue by 2025, along with its AR counterpart [2], companies and organizations are now more open to adopting virtual reality practices in their workforce than ever before. VR is described as a 4D simulation of the real world, including the 3D geometry space, 1D time and the immersive or semi-immersive interaction interface [3]. The immersive nature of this 4D simulation is what makes VR so attractive to institutions planning to train workers for manufacturing tasks.

Virtual reality-based training is a more advanced method of teaching manufacturing skills and processes to employees [1]. VR training instructs users by fully immersing them in a virtual world, allowing them to acquire skills in a realistic atmosphere not found in 2D training mechanisms. Studies have concluded that realistic procedural simulations, especially with haptic feedback, lead to better performances, faster performance curves, and a high transfer of operative skill [4].

Though virtual reality has existed for nearly half a century, recent releases of high quality commercially available head mounted display's (HMD's) have made VR affordable for consumers and companies. Due to quality improvements and cost decreases, companies looking to use VR for tasks such as assembly training

can now realistically explore the technology for widespread use. While this new commercial technology is garnering increased attention, little is known about how to fuse emerging software (Unity), hardware (Oculus Rift), and training instruction guides to construct a manufacturing assembly training application. Consequently, research in factory training lacks extensive reviews of plausible user interfaces. This paper's aim is to test the feasibility of using emerging hardware and software to develop a manufacturing training application and to analyze previous research in order to develop an intuitive user interface for VR assembly training.

Background

The benefits associated with employing virtual reality for training purposes have been well documented by academia, specifically in the medical domain. VR-trained surgeons reported 29% less time spent performing operations and were five times less likely to cause errors [5]. Results of the work also show that regardless of the surgeries complexity, surgeons were aided by practicing the task using virtual reality training. This work demonstrates the potential benefits a VR training application could deliver when used to teach complex manufacturing operations.

Another frequently researched topic is the use of virtual reality training for flight simulators. In flight simulations, VR training not only benefits cost-reduction and time consumption, but it also can be used to test new aircraft concepts [6]. However, most papers in academia do not justify the selection of user interface elements in their training simulators, leaving UI research in this field lacking. Though there is research that covers virtual reality UI's in general, most are primarily focused on VR in gaming applications. This lack of research into more manufacturing based UIs is an area that needs to be addressed. Research into VR assembly training UIs will help ensure high quality VR training applications are delivered to the manufacturing industry. For this reason, one of the main focuses of this paper is to explore, create, and justify UI elements included in a training interface for mock airplane wing assembly.

Methodology

The methods section will discuss the process for hardware selection, virtual world development, UI implementation, and the prototype development process. The hardware section will talk about why the Oculus Rift was chosen along with key features that support its use in manufacturing training. The virtual world development section will justify the creation of the virtual environment using existing research and established practices. In the UI implementation section, elements selected are described and their use justified using previous research. The last section, application development, will describe at a high level the tools and methods used to develop the application.

Hardware Platform Used for Development

For the work, the first task was to select a head-mounted display device used to show the virtual training environment. The Oculus Rift, see Figure 1, was selected from a handful of available devices on the market at the time. This device was selected due to a combination of its consumer price point of under \$500, relatively high field of view, and ergonomic controllers.

To operate the display device a computer requires, at least, an Intel Core i3-6100 and NVIDIA GeForce GTX or AMD Radeon RX 470 and AMD FX4350, 8GB RAM, Windows 7, a compatible HDMI 1.3 video output and two USB 3.0 ports. This computational power allows 1080x1200 resolution per eye at a 90Hz refresh rate and a 110-degree field of view through a Pentile OLED display which generates the 3D content. Other features of the Oculus include spatial sound, built-in mic, accelerometer, gyroscope, magnetometer, and a constellation tracking camera.

The controllers, another feature that aided in the Oculus's selection, have been specifically designed for human hands to experience natural comfort when holding the controllers. The natural feel when gripping the Touch controller allows the user to comfortably manipulate objects in the virtual scene and understand required interaction movements quickly [7, 8]. In addition, the ergonomic Oculus Touch controllers provide tactile force feedback, an advantage the Oculus has over a rival device called the HTC Vive [9]. The feedback reproduces shape, roughness, and rigidity which acts as a substitute of the visual presentations of information [10].



Figure 1. The Oculus Rift

Virtual World Development

In order to begin developing the training application and the virtual world, a training task was required. The training task selected was a 46 step mock aircraft wing assembly task modeled after previous assembly studies using augmented reality [11]. The virtual world for this task was designed to be a replica of a manufacturing work cell used in previous work. An aerial view of the work cell setup is shown in Figure 2. In the real-world environment, work instructions for assemblers were visible from the work area. In the virtual environment, these instructions were placed on a virtual wall, see Figure 3. The wall based assembly instructions allow the user to interact with floating buttons via a pointer using the index trigger on an Oculus Touch controller. The wall based instructions also display the current step number along with the total number of steps.

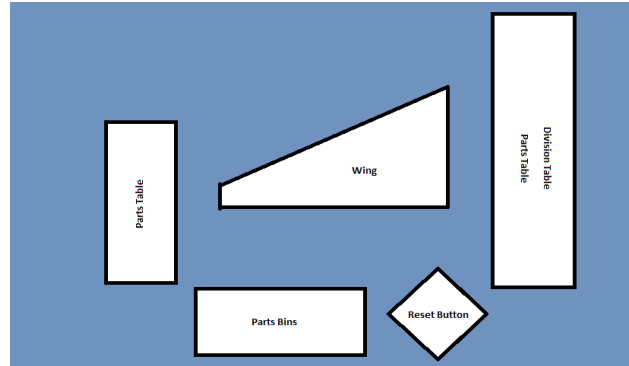


Figure 2. Aerial view of manufacturing cell



Figure 3. VR training area

From the work cell area, the user can choose to enter a tutorial, the general wing assembly, or exit the application. When the wing assembly is selected, the manufacturing environment loads with the completed assembly shown. A control panel is located directly to the right of the assembly, allowing the user to begin the assembly or go back to the menu. Everything within this scene uses the hand trigger on the Oculus Touch, which requires the middle finger. Thus, the grabbing motion is consistent between the assembly parts and the buttons. Once the start button is grabbed, the control panel relocates to a corner and presents options to reset the assembly or return to the menu.

When a user starts the assembly, the first animation becomes visible and indicates where to place the first part. The part to be placed is highlighted on the parts table until picked up. The part will also re-highlight if it is let go of for a certain amount of time. If the part is located in the parts bin with multiple of the same part, the bin is highlighted and a large number of required parts from the bin is displayed above. When a piece is held or dropped into a position that is within a certain tolerance, it will snap into place. After the assembly step is complete, the next step begins automatically.

UI Implementation

In this section, the researchers will explore the different types UI elements the prototype uses. The reasons for animation feedback, choice of color, selection, snapping, and element placement will be discussed and justified because they affect how users interact and learn within a virtual environment.

Animation Feedback

This application uses a variety of animated visual cues like object selection highlighting, pulsating, and color change (when the user selects and puts down an object), as seen in Figure 4.

Animations are used to grasp the attention of the user through visual cues; they inform a user to perform an action, and when the action has been completed since they are an intuitive representation of assembly instructions [13, 14]. Animation was selected over static part placement guides because research suggest it reduces a user's cognitive and short-term memory load [15]. Research also suggests that animations can improve the speed in which individuals learn tasks that involve physical movements, compared to static pictures [12].

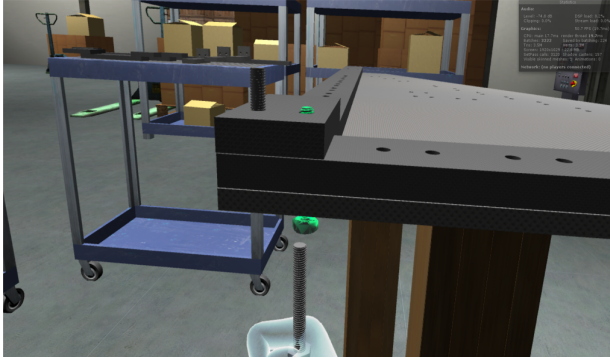


Figure 4. Animation showing user where to place object

Color Choice

Color combinations are another aspect considered during UI implementation. Research suggests that the right selections can improve memory performance and retention [16]. The specific colors selected are meant to ensure that users remember the actions taken throughout their virtual training experience. Pure black-and-white conditions have been proven to be disadvantageous in virtual environments, thus, they were not used [16, 17]. For the user to feel comfortable in the environment while training, a warm color of yellow is used [16]. RGB yellow (255, 248, 0, 122) was used for object selection. Cool colors were used to hint users as to what their next actions would be. RGB green (0, 173, 80, 139) was used for showing users, where to place each selected object. Animations were given a transparent effect as well. This helped inform the user that the animation was instruction, not an interactive part. It is anticipated that using different colors for specific actions will allow users to familiarize themselves with each action type, helping the user remember steps when sent onto the actual manufacturing floor.

Snapping

Snapping alignments allow the user to position virtual objects onto dynamic, real world scenes [20]. Snapping objects into place was selected because some actions, such as turning a wingnut, are challenging to accurately emulate using the Touch controllers. Snapping provided a way to ensure that users had to correctly position parts in the virtual training application. It also provided a solution to interaction challenges associated with bringing the mock aircraft wing assembly from the physical into the virtual world.

Selection: Pointing

A ray like pointer allows users to interact with scene elements outside of their direct physical reach, like the wall based instructions. Ray like pointers have been shown to increase interaction speed and accuracy when manipulating some menu elements [22]. Research also suggests users find a ray pointing tool to be intuitive and easy to use [23]. In the virtual training

application, a pointer appears when the user selects options on the menu.

Selection: Grabbing

The ability to grab objects within a virtual environment (VE) is essential as it is important to make the virtual world feel as real as possible. With the aid of the Oculus Touch, shown in Figure 5, the manufacturing assembly prototype uses a grab function which allows users to reach out and select pieces needed for an assembly operation. Using the grab feature to manipulate objects in the VR assembly training application was selected because the grabbing action is a natural physical motion. According to research this grab to manipulate skill easily transfers to the virtual world, reducing required training time [23]. Grabbing is also used to interact with the control panel when a user is completing an assembly. This is done in order to keep interaction consistent within each scene. Repeated use of the grabbing motion for interacting with scene elements, rather than buttons, makes the application easier to use for those unfamiliar with traditional gamepads or controls.



Figure 5. Oculus Touch Controllers

Prototype Development

In this section, the researchers explain the software tools used and challenges overcome to complete the application prototype. For the work, a VR manufacturing assembly training application prototype was developed using Unity, the Oculus Rift, and the Oculus Touch controller. The prototype allows the user to practice each step of the mock airplane wing assembly by directing users step by step through the assembly process. The application shows users which assembly parts to pick and where to place them. For the purpose of training the user and making the prototype more interactive, a UI is placed around the assembly and parts. The UI has animations that clue the user in on what assembly part to pick up via pulsating, numbering, and highlighting. For every step that requires the grabbing of an assembly part, a yellow pulsating highlight is placed to tell the user the object needs to be picked up. For every placement step, a green highlighted animation is placed to tell the user how and where the selected object should be placed. The program continues to tell the user to grab and place objects until the mock airplane wing assembly is completed.

Unity was used to create the VE as it allows for both swift integration with the Oculus Rift and has an established asset store that supports the ease of development [17]. When constructing the application initial versions of the prototype were hampered by slow framerates. After some investigation, the issue was identified as being due to the importing of highly detailed 3D models from SolidWorks®, specifically the fastener parts. When a bin full of

highly detailed screws was placed in the scene, the number of vertices in frame passed thirty million. This high number of polygons slowed down the application considerably, producing a frame rate below thirty. Since a frame rate below thirty is disruptive to the immersive level of VR the authors used SAP Visual Enterprise Author 9.0, to decimate models with an excessive number of vertices. The amount of decimation, or polygon reduction, was dependent on the different models and the level of detail required. Decimation was mostly applied to smaller models with a high level of detail, such as the fasteners. This is because a small decimated object is barely noticeable and does not detract from the users understanding of the object's function. Overall, this strategy greatly increased the frame rate.

A proximity snapping technique was used for part placement as it gives a natural feel of assembling to the user [24]. Snapping also informed the user that the step has been completed and they can move on. Each piece required an individual tolerance set based on trial and error due to varying part shapes and sizing. Small symmetrical parts presented a challenge as the tolerances caused frustration when attempting to place them in the correct position. Because of this, a separate method of snapping had to be developed. Thus, it was challenging to keep consistency between large object snaps and small object snaps. However, the hand tuning of the snapping tolerance helped mitigate any negative impacts on the user.

Results and Discussion

In order to gauge the performance of the application, the authors first tested the frame rate to ensure sufficient visual quality and responsiveness for users. The resulting assembly training application, shown in Figure 6, performed at an average of 46 frames per second (FPS) while containing a total of 6 million vertices within the environment. While this is slightly below the ideal frame rate of 60 FPS, it is still above the minimum 30 FPS required for real time interaction in a VE. Frame rate results, high-quality visuals, and ease of HMD integration show that Unity is useful tool for quickly generating a high-quality proof of concept VE, although with some frame rate limitations. Future work will look at ways to use Unity's inbuilt options to maximize the prototype applications FPS when using geometry from highly detailed computer-aided drafting systems.

When designing a VR application for assembly training, high fidelity models (i.e., models with a high vertex count) employed in the VE can hinder the application's performance. Hence, the use of high fidelity models will present a challenge to those attempting to import unaltered models into Unity. To combat this issue, high fidelity models of insignificant components (i.e., bolts, washers, and wingnuts) were decimated to reduce the number of vertices. Each model must be decimated individually, but once they have been decimated they can be reused indefinitely. A trial and error process was used to produce each decimation percentage by viewing how the resulting model looked once integrated into the VE. Incrementally, the decimated models were saved until the models contained the least number of vertices that maintained sufficient visual fidelity for users. The resulting lower fidelity models provided a substantial increase in frame rate while maintaining identifiable geometry unique to each component. The use of 3D models with a high vertex count will present a challenge to anyone attempting to put unaltered models straight into Unity. Therefore, model decimation will prove vital in the development of VR training applications where models need to only pass a visual inspection.



Figure 6. Depiction of the wing assembly training application

The second portion of the results focuses on the lessons learned from building the application that can be applied to future assembly training prototypes. Many lessons were learned while developing the application for assembly training. Most of the difficulties faced were encountered when designing the animations, and piece snapping features. Creating accurate guiding animations for representing the correct part location to users consumed more time than anticipated. Each animation required a unique solution because of differences in part geometry and desired location. As a result, moving forward with more complex assemblies the authors anticipate as the overall assembly size and complexity increases a significant amount of time could be expended authoring the instruction animations. This could potentially become a barrier to adoption for the technology. Additionally, compounding the animation issue it's imperative the designer understands the desired component placement. This understanding ensures the animation properly represents to the user the correct location for a piece within an assembly.

In addition to the custom crafting of animations for the VE, custom scripts were created to snap parts into a user-specified orientation while completing the assembly. An example of part placement is guide shown in Figure 7. The guide shows the user where to place a unique piece. When the user places the virtual part close to the correct position, it will snap into place. To ensure each piece snapped into the correct location each component type required its own custom script. Custom scripts were required due to the variety of parts as well as the variety in intended location and orientation of the parts. This variation caused an increase in time spent on creating snapping scripts for each unique case. Time required for custom scripting, as with the animations, would increase as the overall assembly size and complexity increases. This could potentially become another barrier to adoption for the technology.

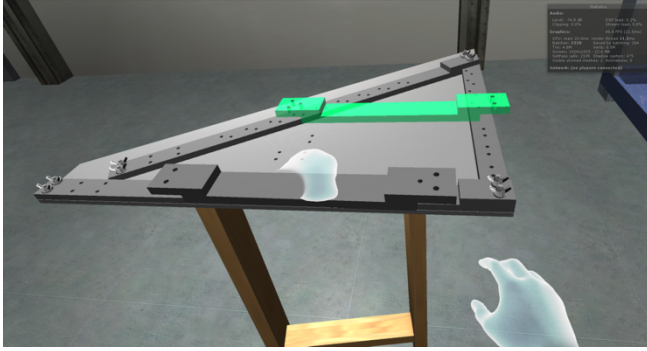


Figure 7. Demonstration of user placing unique part onto the assembly

Snapping parts is relatively easy to implement for larger and simpler components. However, the authors found this more challenging for small and detailed components. This is due to the trial and error process required to determine the tolerance distances needed for snapping a model to a location in the VE. Smaller symmetrical items such as screws and washers required less rotational tolerances. Attention to tolerance levels ensures that users don't experience an unnatural or degraded experience, detracting from the VE training from preventing skill transfer into the physical world. In addition, perceived natural tolerances are important because if users believed they were performing the task correctly and the piece did not snap into place this could cause frustration.

One detraction to using the snapping technique is that the feature limited the ability of users to experience haptic feedback from the parts as assembled. For example, users were not provided an understanding of the amount of torque required to assemble fasteners. Instead, the snapping feature focused on providing the correct locations of assembled parts.

Conclusion

For the work, a VR manufacturing training application was created using Unity 3D and displayed using Oculus Rift. The ergonomic Oculus Touch controllers let users naturally interact with the virtual assembly parts. Assembly instructions and user interface elements in the virtual environment were selected with guidance from previous work on virtual interface design. Methods such as frame rate and ease of interaction with the virtual environment were used to evaluate the performance of the virtual assembly trainer prototype. Results of the work demonstrate the feasibility of using the Oculus Rift and a commercial game engine to create a virtual assembly training application for a mock aircraft wing. After completing the prototype, the authors are convinced the current maturity level of the technology has the potential to improve upon current 2D methods of training. Further research in this field will allow companies to gain a better understanding of how virtual reality can benefit their work performance and decrease their training expenditure. As virtual reality technologies continue to expand, the authors believe that VR training simulators will become more prominent in factory training.

Acknowledgements

This research was performed at Iowa State University as part of a research internship funded by NSF Grant CNS-1461160.

References

[1] T. Mujber, T. Szecsi, and M. Hashmi, "Virtual reality

- applications in manufacturing process simulation," *J. Mater. Process. Technol.*, vol. 155156, pp. 1834–1838, 2004.
- [2] H. Bellini, W. Chen, M. Sugiyama, M. Shin, S. Alam, and D. Takayama, "Virtual & Augmented Reality: Understanding the race for the next computing platform," 2016.
- [3] A. Y. C. Nee, S. K. Ong, G. Chryssolouris, and D. Mourtzis, "Augmented reality applications in design and manufacturing," *CIRP Ann. - Manuf. Technol.*, vol. 61, no. 2, pp. 657–679, 2012.
- [4] O. A. J. Van Der Meijden and M. P. Schijven, "The value of haptic feedback in conventional and robot-assisted minimal invasive surgery and virtual reality training: A current review," *Surgical Endoscopy and Other Interventional Techniques*, vol. 23, no. 6. Springer, pp. 1180–1190, Jun-2009.
- [5] N. E. Seymour *et al.*, "Virtual reality training improves operating room performance: results of a randomized, double-blinded study.," *Ann. Surg.*, vol. 236, no. 4, pp. 458–63–4, 2002.
- [6] T. Aslandere, D. Dreyer, F. Pankratz, and R. Schubotz, "A Generic Virtual Reality Flight Simulator."
- [7] "The difference between Oculus Touch, HTC Vive, PlayStation VR and Windows Mixed Reality controllers | VRheads." [Online]. Available: <https://www.vrheads.com/difference-between-oculus-touch-htc-vive-and-playstation-vr-controllers>. [Accessed: 31-Jul-2017].
- [8] "Oculus Touch vs. HTC Vive - Which Is The Better VR Controller?," 2016. [Online]. Available: <https://uploadvr.com/oculus-touch-vs-htc-vive-better-controller/>. [Accessed: 03-Aug-2017].
- [9] D. W. F. van Krevelen and R. Poelman, "A Survey of Augmented Reality Technologies, Applications and Limitations," *Int. J. Virtual Real.*, vol. 9, no. 2, pp. 1–20, 2010.
- [10] V. Chouvardas, an Miliou, and M. Hatalis, "Tactile display applications: A state of the art survey," *2nd Balk. Conf. Informatics*, pp. 290–303, 2005.
- [11] T. Richardson, S. B. Gilbert, J. Holub, F. Thompson, and A. MacAllister, "Fusing Self-Reported and Sensor Data from Mixed- Reality Training Recommended Citation."
- [12] M. Wong, J. C. Castro-Alonso, P. Ayres, and F. Paas, "Gender effects when learning manipulative tasks from instructional animations and static presentations," *Educ. Technol. Soc.*, vol. 18, no. 4, pp. 37–52, 2015.
- [13] N. N. J. Mitra, Y.-L. Y. Yang, D.-M. D. Yan, W. Li, and M. Agrawala, "Illustrating How Mechanical Assemblies Work," *Commun. ACM*, vol. 56, no. 1, pp. 106–114, 2013.
- [14] M. L. Yuan, S. K. Ong, and A. Y. C. Nee, "Assembly Guidance in Augmented Reality Environments Using a Virtual Interactive Tool," *Int. J. Prod. Res.*, vol. 46, no. 7, pp. 1745–1767, 2005.
- [15] B. Shneiderman, "Visual Analytics: New Tools for Gaining Insight from Your Data," 2000.
- [16] M. A. Dzulkifli and M. F. Mustafar, "The influence of colour on memory performance: a review.," *Malays. J. Med. Sci.*, vol. 20, no. 2, pp. 3–9, Mar. 2013.
- [17] G. Evans, J. Miller, M. Iglesias Pena, A. MacAllister, and E. Winer, "Evaluating the Microsoft HoloLens through an augmented reality assembly application," vol. 10197, p. 101970V, 2017.
- [18] M. Alger, "Visual Design Methods for Virtual Reality," no. September, p. 98, 2015.
- [19] T. Shibata, J. Kim, D. M. Hoffman, and M. S. Banks, "Visual

- discomfort with stereo displays: Effects of viewing distance and direction of vergence-accommodation conflict.," *Proc. SPIE*, vol. 7863, p. 78630P1--78630P9, 2011.
- [20] B. Nuernberger, E. Ofek, H. Benko, and A. D. Wilson, "SnapToReality: Aligning Augmented Reality to the Real World," *Proc. 2016 CHI Conf. Hum. Factors Comput. Syst. - CHI '16*, pp. 1233–1244, 2016.
- [21] C. Ware and S. Osborn, "Exploration and Virtual Camera Control in Virtual Three Dimensional Environment s," pp. 175–183, 1990.
- [22] D. A. Bowman, D. Koller, and L. F. Hodges, "Travel in Immersive Virtual Environments : An Evaluation of Viewpoint Motion Control Techniques Georgia Institute of Technology," *Proc. 1997 Virtual Real. Annu. Int. Symp.*, p. 45-, 1997.
- [23] K. M. Fairchild, B. H. Lee, J. Loo, H. Ng, and L. Serra, "The heaven and earth virtual reality: Designing applications for novice users," *Proc. IEEE Virtual Real. Annu. Int. Symp.*, no. July 2015, pp. 47–53, 1993.
- [24] J. M. Ritchie, R. G. Dewar, and J. E. L. Simmons, "The generation and practical use of plans for manual assembly using immersive virtual reality."