



Mitigation of the Microsoft HoloLens' hardware limitations for a controlled product assembly process

Jack Miller¹ · Melynda Hoover¹ · Eliot Winer¹

Received: 18 March 2020 / Accepted: 13 July 2020 / Published online: 21 July 2020
© Springer-Verlag London Ltd., part of Springer Nature 2020

Abstract

Studies have shown that using augmented reality (AR) reduces time and errors by as much as 50% for manufacturing and assembly operations. The use of head-mounted displays (HMDs) is very interesting for these situations as they free up a user's hands. However, with the technology behind these HMDs still under development, limitations in input, field of view, tracking, and occlusion exist. For AR assembly instructions to be effective, these limitations must be mitigated so that a user receives accurate instructions. To investigate this, an application was developed that guides a user through the assembly of a mock aircraft wing using AR on a Microsoft HoloLens 1. The application displays 3D work instructions, user interfaces, and spatially registered content. To ensure accurate instructions were displayed, techniques were developed to mitigate the HoloLens' tracking and display limitations. Image tracking was implemented to augment the HoloLens' limited position determination, stabilize spatially registered virtual content, and account for spatial drift. Also, 3D augmented content was optimized for clarity through the display using specialized computer graphics methods. Lastly, the system provides raw data regarding head position, orientation, and assembly step times. A visualization application was developed that combines this information with wearable sensor data, to examine trends exhibited by a user during the assembly tasks and validate the delivered instructions. The outcomes of this research provided solutions to address limitations of the HoloLens for broad use.

Keywords Augmented reality · Work instructions · Assembly · HoloLens

1 Introduction

Augmented reality (AR) is an emerging technology that could greatly increase training efficiency and effectiveness. In fact, Henderson and Feiner found that AR may reduce time and errors by as much as 50% for maintenance and assembly operations [1]. Essentially, AR consists of computer-generated visual information overlaid on top of a view of the real world. These computer-generated visuals are often delivered via a tablet or head-mounted display (HMD). Traditionally, tablets have been the preferred method of delivering AR content due

to technological constraints and commercial availability. HMDs of the past tended to be expensive custom-built solutions for research purposes and not viable for commercial use [2–4]. However, in recent years, several commercially available AR HMDs have come to market and shown promise for use in maintenance operations. The main advantage of AR HMDs over tablets is that they overlay the computer-generated visuals directly over the user's field of view, allowing hands-free operation. This is ideal for maintenance and assembly operations and training.

AR HMDs show great potential for use in maintenance operations and training as will be shown later in the paper. However, due to the emerging nature of this technology, many of the components that make up these AR HMDs are still under development. Therefore, there are many limitations within the technology that must be mitigated including input methods, navigation, tracking, and occlusion [2–4]. These limitations are especially important to maintenance and assembly tasks as it is essential to ensure the user receives accurate work instructions and effective training. Therefore, augmented 3D content must be accurately spatially registered and displayed;

✉ Jack Miller
Jackm@iastate.edu

Melynda Hoover
mthoover@iastate.edu

Eliot Winer
ewiner@iastate.edu

¹ Iowa State University, 537 Bissell Road, Ames, IA 50011, USA

otherwise, users may misunderstand instructions leading to additional assembly time and errors. Furthermore, as assembly tasks may take a considerable amount of time, it is important to ensure the spatial registration does not degrade via spatial drift of the device tracking system.

This paper discusses the novel techniques employed to mitigate the limitations of the Microsoft HoloLens 1 for use in an AR work instruction assembly application. Specifically, the methods were developed to address the HoloLens' limited position determination, spatial registration of virtual content, and spatial drift as these can be detrimental to using the device for a real-world application. This application was also used to conduct a user study comparing the effectiveness of work instructions delivered via an AR HMD, AR tablet, tablet model-based instructions, and desktop model-based instructions [5]. The focus of this current paper is not the study design and implementation but rather addressing the HoloLens's limitations as stated earlier. Due to all the custom enhancements made to the HMD system, it was necessary to evaluate that the technical parts of the system were functioning properly. To accomplish this, a separate data visualization application was developed, which used the study data for validation. This fusion through the data visualization application allowed an analyst to better understand the training session and identify any issues in the AR application. In addition to validating the work instructions, an analyst may also discover new high-level trends about the user and the training environment through the fusion of these data sources. One of the largest limitations of the HoloLens was to ensure proper tracking when spatial drift was present. To combat tracking limitations, data was collected during the training sessions to measure the drift accumulated and validate that the techniques developed to combat this spatial drift were effective. Through the delivery and validation of these work instructions, this work seeks to enhance the effectiveness of training for maintenance and assembly operations using an emerging commodity HMD.

2 Background

Augmented reality is a technology that has been heavily researched since the 1990's and shows promise for a variety of applications [6, 7]. AR involves merging computer-generated visuals with a view of the real world. These visuals provide additional information to what is seen in reality. In a seminal paper, Azuma surveyed the various applications of AR and found that it would have a major impact in the fields of medical, robotics, entertainment, visualization, and notably manufacturing and repair [8]. Additionally, Krevelen and Poelman found similar use cases and many more including education, military training, and others [9]. Manufacturing is a large area for AR research including many use cases for

industry and the military. Specifically, AR-guided assembly and disassembly tasks make up 33% of published research for AR manufacturing [10]. Palmarini et al. discuss the extensive promise of AR for manufacturing and assembly operations while noting some key limitations of the technology.

2.1 Augmented reality assembly applications

Numerous papers have explored the use of AR for the delivery of work instructions. One of the first AR work instruction applications was proposed by Caudell and Mizell for an aircraft manufacturing process through a transparent HMD [7]. However, the technology at that time was still in its infancy and suffered from many of the same problems modern HMDs have, specifically tracking and display capabilities [7]. Since this seminal paper, researchers have developed many systems to explore the possibilities of this technology.

Studies have shown that AR-delivered work instructions drastically reduce errors when compared with traditional delivery methods [11–13]. Furthermore, Baird and Barfield found that in addition to reducing errors, AR also decreased the time to complete assembly tasks. In a similar study, Henderson and Feiner found that AR-delivered instructions significantly reduced (1) the amount of time to locate tasks, (2) head and neck movements, and (3) mental workload [1]. Additionally, since the instructions are presented directly over a view of the real world, the user does not have to recall the visual information from 2D instructions [14]. These benefits lead to enhanced assembly performance and therefore reduced costs.

2.2 Augmented reality hardware

AR assembly instructions are often delivered through transparent HMDs, tablets, and smartphones. Due to commercial availability and relatively low cost, tablets and smartphones have been the preferred AR platforms [15]. However, the main drawback of these devices for delivering assembly instructions is that typically a user must hold the device when using it. This is not ideal as the user will likely be working with their hands when receiving work instructions.

AR HMDs may overcome this issue through a hands-free experience. In the past, AR HMDs have been expensive custom-built solutions created for specific use cases. This has been a large pitfall for the use of HMDs in many applications as they are not readily available. However, in recent years, numerous commercially available AR HMDs have come to market including the Microsoft HoloLens, Daqri Smart Glasses, Meta 2, and Magic Leap [16–19]. These devices can track the environment while displaying spatially located 3D content through an optical see-through display. The use of these devices allows the user to remain hands-free and view instructions directly over their field of view.

Evans et al. evaluated the Microsoft HoloLens 1 for use in an assembly application and found the device to be a viable platform [20]. The research presented by Evans et al. showed that the HoloLens had the necessary processing power, display, and tracking capabilities for use in assembly applications. However, more research was needed to further mitigate these issues and evaluate it for use in the field.

2.3 Augmented reality limitations

While there are many benefits for the use of AR, there are many limitations that come along with it. Tang et al. found great success with the use of AR for assembly purposes but noted that the technology was not ready for widespread use due to various limitations in the hardware [21]. Some of the main hardware limitations include tracking, depth perception, and occlusion [9, 22]. While hardware improvements may help overcome some of these limitations, novel development techniques may additionally help mitigate the negative effects.

Tracking can be broken down into two main categories, marker based and marker less. The most accurate method is marker-based tracking. This method uses a high-contrast image target to provide a real-world point of reference. This image target can be identified by an RGB camera on the AR device to establish this point of reference. While marker-based tracking provides high accuracy, its main drawback is that it requires an image target for any object it is tracking. Additionally, the image target, the environment, and the camera detecting the image target have many limitations including lighting conditions, gloss, contrast, camera quality, viewing angle, and distance. Marker-less tracking overcomes the issue of needing an image target through various techniques, often combining input from visual sources, depth sensors, and an inertial measurement unit (IMU) [23]. The main issue with marker-less tracking, such as simultaneous localization and mapping (SLAM), is that it lacks the accuracy to register precise points in the real world. While a mesh of the general environment can be mapped, understanding specific points presents many challenges that have yet to be explored [24]. Since it is often difficult to achieve an accurate point of reference, it is not ideal for tasks, such as assembly, that require precise registration.

Depth perception is very important to assembly applications as the user must be able to properly locate and visualize augmented instructions. Many past devices had issues with dim displays and low resolution leading to virtual objects appearing farther away than they should [9]. Even though recent advances in technology have improved resolution and opacity in many devices, there are still many improvements that can be made. Occlusion has been found to effectively give depth perception cues that may overcome limitations in resolution and opacity [25]. Through accurate occlusion, the user may better understand the proper representation of the

instructions given [26, 27]. However, to properly occlude virtual content with real-world objects, the AR system must understand the environment and its geometry. As previously mentioned, tracking the environment is not a simple task, and this limitation must be overcome to ensure instructions are properly displayed.

Additional research is needed to understand and mitigate these limitations for use in AR work instruction applications. While hardware improvements of emerging technology will be very beneficial in the future, many of the current limitations can be mitigated through novel development techniques. The work in this paper explores these limitations and offers solutions to overcome them.

3 Methodology

3.1 Hardware

Several commercially available AR HMDs currently exist on the market. For this research, the Microsoft HoloLens 1 was chosen as the display device for the AR assembly application due to its maturity and popularity compared with the other devices at the time of this research. Furthermore, the HoloLens includes the necessary capabilities for an AR HMD assembly application: environment tracking, sufficient computing power, and a transparent display. Despite these capabilities, the HoloLens has limitations in input, field of view, tracking, and occlusion that must be mitigated for the delivery of AR work instructions.

The HoloLens is a completely self-contained AR HMD and is not required to be tethered to an external computing device [28]. The computing power of the HoloLens consists of a 32-bit Intel processor with a custom-built Microsoft Holographic Processing Unit. The HoloLens takes in sensor data from an inertial measurement unit, four environment understanding cameras, one depth camera, one 2MP photo/HD video camera, four microphones, and one ambient light sensor. Graphics are then displayed through see-through holographic lenses using waveguide technology. An effective resolution of 1268×720 per eye is achieved through two HD 16:9 light engines.

3.2 Development tools

Unity3D is a powerful game engine widely used for the development of 2D, 3D, VR, and AR applications [29]. The Unity3D editor allows for the rapid development of applications through a large toolset including a rendering engine, a physics engine, user interface design tools, and the Unity3D C# scripting API. In addition, support is available for a variety of build platforms including windows, mac, iOS, android, and Universal Windows Platform (UWP). Specifically, Microsoft

provides ample documentation and recommends for the development of AR HoloLens applications using Unity3D [30]. Due to the vast toolset and capabilities of Unity3D, it was chosen as the development platform for the AR assembly application and data visualization tool.

The Vuforia SDK is one of the leading AR SDKs used for a variety of purposes [31]. Using propriety computer vision algorithms, Vuforia provides functionality for image, text, model, and object tracking. Through this tracking, Vuforia is able to deliver position and orientation data which can then be used to spatially register AR content. Like development with Unity3D, Microsoft provides documentation for and highly recommends the use of Vuforia when utilizing image targets in a HoloLens application [32]. Furthermore, as of Unity3D 2017.2, Vuforia is built into the Unity3D engine to increase functionality and ease of use. Vuforia was chosen to augment the tracking capabilities of the HoloLens for the AR system. The functionality, wealth of documentation, and stability of the Vuforia SDK were ideal to provide image tracking capabilities for the AR assembly application; Fig. 1 shows an example image target. The image tracking, accompanied by the HoloLens tracking system, allowed for all AR content to be properly spatially registered.

3.3 Effective user input

For the AR assembly instructions to be usable, the user had to be able to interact with the application in an intuitive manner. To receive input from the user, there were a variety of options available. The HoloLens provides built-in gesture controls, the most popular being the “air tap.” While this input method is relatively hands-free, it is often difficult to teach to a new user and may lead to reduced usability of the system [33, 34]. Another input method often used in AR HMDs is the “gaze and dwell.” While this is completely hands-free, the user must keep their head unusually still and it tends to be very slow. Voice commands were another method considered; however,

this would not be ideal for loud manufacturing environments. To overcome the issues of these input modes and provide an intuitive experience, the authors implemented the HoloLens Bluetooth clicker. The clicker was paired with the HoloLens, and the application received input when the clicker button is pressed. To keep the user hands-free during the assembly process, the clicker was attached to the user’s wrist by a simple strap. While the user gazed at the virtual button, they can press this clicker to deliver input to the HoloLens as shown in Fig. 2. It is an intuitive method that is simple to learn.

3.4 Tracking accuracy

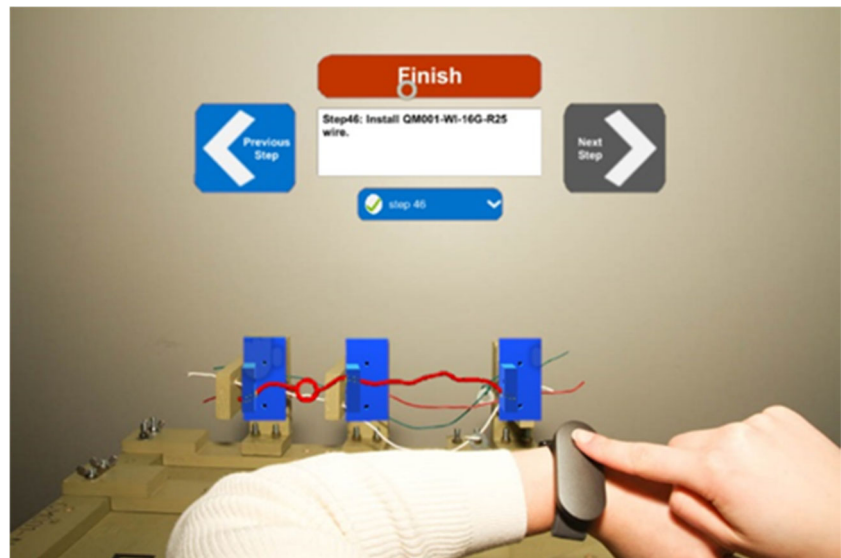
The AR assembly application needed to track the environment well enough to correctly display spatially located assembly instructions. The HoloLens can generate a mesh of the environment using its propriety tracking system in real time. To successfully track the assembly, the generated mesh had to be detailed enough to identify key feature points such as edges and corners. However, Fig. 3 shows that the mesh generated for even a basic assembly is not nearly accurate enough to achieve this. In addition, the HoloLens suffers from spatial drift due to error in the propriety tracking system and IMU. This drift accumulates over time and would be detrimental to any task lasting more than a few minutes [35]. If the AR work instructions were not accurately registered, then the user may misinterpret the instructions leading to safety concerns or errors in the assembly process. To overcome these limitations, the Vuforia SDK was used to provide image tracking capabilities. Image targets were then used to initialize the location of the stations to augment the environment tracking of the HoloLens. The UI was then placed over these image targets so when the user looked at the UI, the application seamlessly reinitialized the position and corrected the drift.

While image targets offer additional tracking capabilities, they are not without their own inaccuracies. Lighting conditions, gloss, target image quality, camera quality, viewing

Fig. 1 Image target example



Fig. 2 HoloLens input method and interface



angle, and distance can all lead to errors and/or false positives when detecting the target, examples of this are shown in Figs. 4 and 5 [36]. Any error in the calculated position and orientation of the image target is subject to a lever arm, i.e., rotational error is propagated over distance. This means that even a one-degree error in the calculated rotation of the image target over the distance of 1 m would lead to almost a 2-cm difference in the location of the spatialized content. To overcome this limitation, a calibration stage was implemented to ensure all locations were properly initialized. This stage involved the user gazing at the image targets to initialize the position and then ensuring that the UI was placed flush with the image target. The tracking system then used this initialized position and orientation as a baseline. After this point, anytime the image target was detected, the calculated orientation was compared with the initialized orientation. If the rotations were different, then the image target would not update its position.

This method drastically reduced false positives when detecting the image target and limited error in the tracking system.

3.5 Navigation

Since the user was unfamiliar with the assembly environment, a navigation system was needed to guide them to the correct location. Previous research indicates that a 3D gate system is the most usable form of navigation in large 3D environments [37]. To accomplish this, virtual square yellow gates were placed along a Bezier curve leading to the current step's location, shown in Fig. 6. Since the stations were spatially located, a Bezier curve could be generated between the user's location and the current step's location. One control point was placed in front of the HoloLens to ensure the gates were always in view of the user and a second control point was placed between the previous control point and the end point to smooth

Fig. 3 HoloLens spatial mapping of a basic assembly

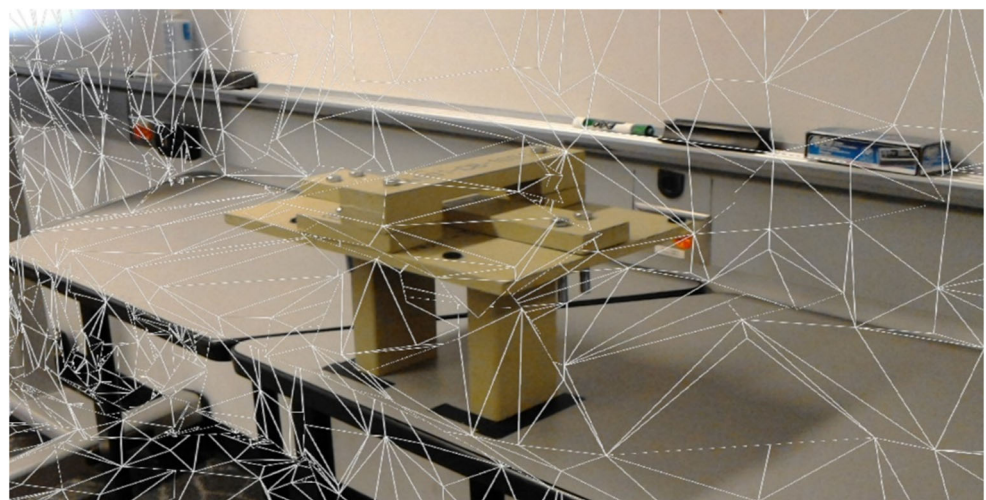


Fig. 4 UI correctly located flush with image target



the curve. Square yellow gates were then placed along the calculated curve at equal distances. The HoloLens' processing power along with the Unity3D C# scripting API was able to perform this operation at a high framerate to ensure a smooth navigation system. The use of this navigation system was able to give the user directional cues in an unfamiliar training environment where many distractions could be prevalent.

3.6 Occlusion

Occlusion is crucial to an AR assembly application. The depth perception cues given by occlusion ensure the user sees the proper representation of the work instructions [26, 27]. Therefore, real-world parts must be able to occlude the virtual parts shown in the work instructions. As previously discussed, the HoloLens can create a mesh of the real world but in very poor detail. This is not ideal for occlusion as inaccuracies may misrepresent the instructions given. To overcome this limitation, the authors utilized the augmented tracking capabilities of Vuforia with the static nature of the stations. The station positions and orientations were calculated from the Vuforia image targets. From there, the position and orientation of each assembled part was calculated relative to the station it belonged to using vector math. Finally, a virtual representation of each previously assembled part was placed in the scene with the same position and orientation as its real counterpart. A shader was then used to write this virtual part to the z-buffer

but not render anything. A separate shader was used to render the work instructions a solid opaque color when in front of a real part and a yellow outline where it is occluded; see Fig. 7. This custom solution allowed the user to see virtual work instructions properly occluded by real parts leading to the proper representation of the work instructions.

3.7 Data collection

During the training session, data was collected on the user's experience from the same sensors used to track the environment. This data included position data, orientation data, when input actions were performed, and how long the user spent on steps. The data was then stored locally on the HoloLens as CSV files to be easily parsed later for analysis. In addition, data could be collected from various other sources such as wearable physiological sensors and post-processed assembly error data. These rich data sets were combined to create a greater understanding of the user's experience.

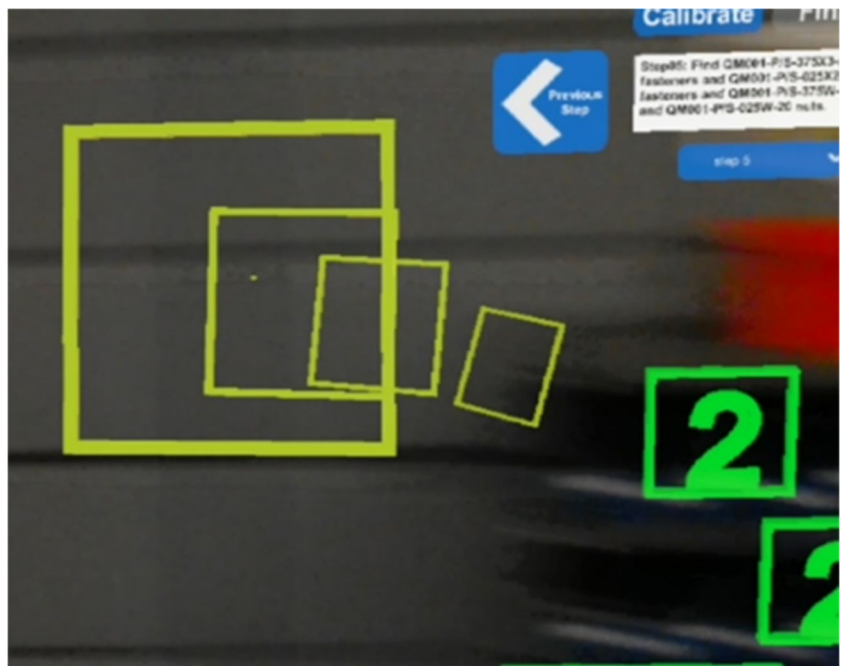
4 Results and evaluation

4.1 Augmented reality assembly application

To assess the hardware mitigation techniques described in this paper, an experimental setup of an assembly of a mock aircraft

Fig. 5 Miscalculated orientation of image target



Fig. 6 Yellow navigation gates

wing was used as shown in Fig. 8. This assembly included a 46-step process of picking, placing, and assembling a variety of parts and fasteners. These steps required the user to navigate through three separate stations: a parts table, fastener bins, and assembly station. The parts table included large wooden parts to be placed and assembled, the bins contained the fasteners used to assemble the parts, and the assembly station was where the final product was put together. Each station was separated by roughly 8 ft as shown in Fig. 9, and the assembly station was positioned at approximately 4 ft high for ergonomic purposes. Physiological data was collected

throughout the assembly training process using a wearable wrist device.

An AR assembly application was developed, with the methods described earlier, to guide a user through the assembly of the mock aircraft wing. The application began with a UI asking for the user's identification number to properly store the session data. Then, a large start button appeared with four large white squares in each corner of the display. These white squares were used to ensure the user was properly wearing the HoloLens device. If the user could see all four white squares, it meant that they were experiencing the full field of view for

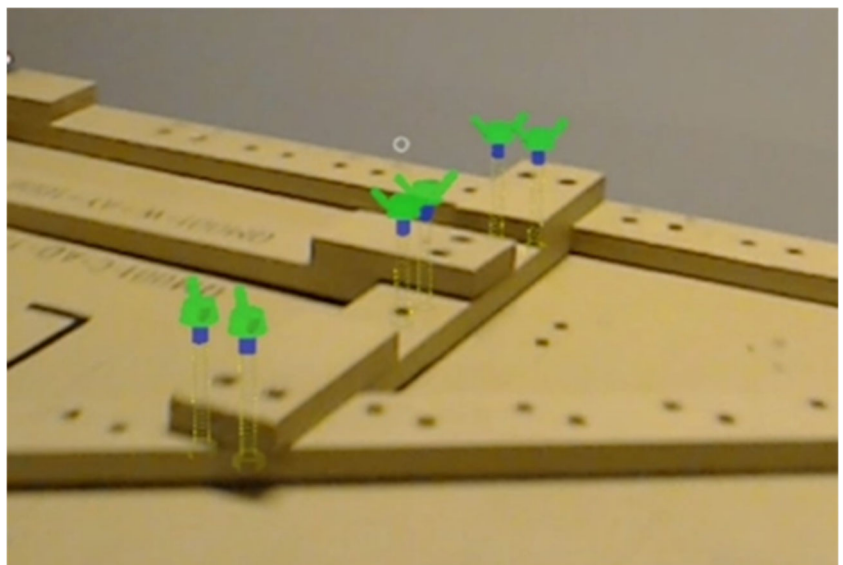
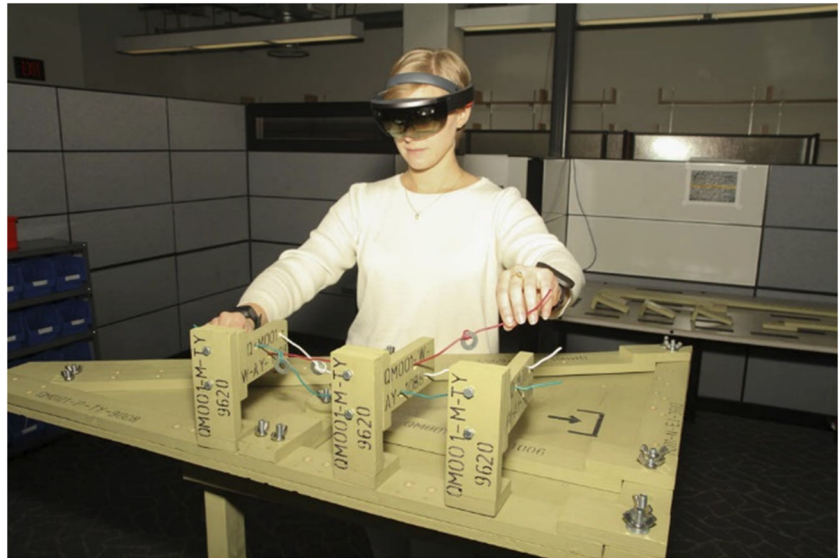
Fig. 7 Virtual bolts occluded by real parts

Fig. 8 Mock aircraft wing assembly



the HoloLens. After the start button was pressed, the user was led through the 46-step assembly process. The UI appeared over the current station and displayed pertinent information including text directions and navigational tools. The user's attention was guided through a series of square yellow gates to the proper position. Parts that need to be acquired were outlined in green, and then animations were given to demonstrate how to assemble each step as shown in Fig. 2. When all steps were completed, a finish button would appear which then terminated the session and stopped the data logging.

Wearable sensors such as Fitbit, Apple Watch, and the Empatica E4 can be used to collect important physiological data about the user. This data includes photoplethysmography, skin temperature, and electrodermal activity. Specifically, the Empatica E4 was chosen for this

application, as it is a high end, CE Medical 93/42/EEC Directive, class 2A compliant physiological sensor providing the most accurate data for health applications [38]. Using this device during training can lead to a better understanding of the user's experience and readiness level through an analysis of their physiological response.

4.2 Drift correction

As stated in the introduction, this paper focuses on addressing the hardware limitations of the HoloLens. To validate this, the data from a user study, in a parallel project, was used. This data was comprised of 83 trials of an assembly application to better understand how the spatial drift of the HoloLens affected delivered work instructions. This data was taken directly from the HoloLens without the need of a secondary tracking system. Every time the user looked at the image target, a measurement was taken from where the image target was previously located to where it was recalibrated. Since the image target was stationary in the real world, this measurement captured the drift accumulated in between recalibration events. The first 12 trials were broken up into two separate conditions, one where the drift was corrected and one where it was not.

The first condition involved the AR system recalibrating and zeroing out the initialized locations each time the user viewed the image target. The relative position of the HoloLens was reset at each recalibration event. However, to reduce the effect of false positives in these measurements, the baseline position and orientation checks discussed in the methodology section were utilized. This ensures that any erroneous image target detection was not included in this analysis. The second condition involved the AR system calculating the drift difference but not recalibrating and zeroing out

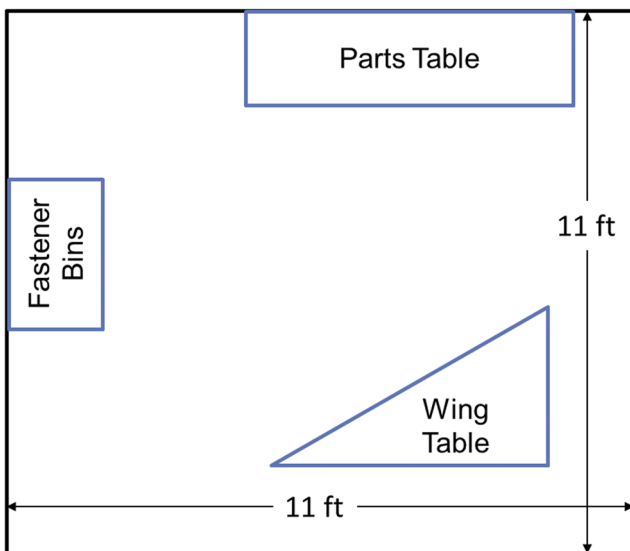


Fig. 9 Station layout

the image target location. While the HoloLens is not reset and recalibrated throughout the trial, the baseline position and orientation checks were utilized to ensure erroneous image target detection was not included in this condition either. After these initial trials, it was found that the correction condition significantly reduced drift error and the remaining 71 trials were run with the correction condition enabled. Full details of the study can be found in a corresponding publication [5].

The data collected during the corrected drift condition consisted of the measurements taken at each recalibration event. Figure 10 shows the interpolated trend of drift built up over time and resetting at each recalibration event. In addition, by summing the recalibration distances over the course of the trial, approximately 15 min, it was possible to get a graph of how drift would have theoretically accumulated over time if it was not corrected. For individual trials, this resulted in a stepwise function with a step at each recalibration event. As shown in Fig. 10, by periodically correcting for drift, the instantaneous drift was much lower than the theoretical drift accumulated/summed. Furthermore, by averaging all 77 trials of the summated drift, it was possible to examine a general trend of how drift accumulated. Figure 10 shows the result of averaging these 77 trials with a fitted linear regression line. On average, drift tended to accumulate at a rate of 0.0459 mm per second with the line fitting with a R^2 value of 0.9868. Figure 10 also shows the recalibration events for a single trial. An interpolated line was included between events to show approximately how the drift accumulated. This shows that the correction method significantly reduced drift at any given time during the trial.

To confirm the trend of drift accumulating over time, the 6 uncorrected condition trials were analyzed. For these trials, there were no recalibration events, so the accumulated/summated drift distance was the same as the instantaneous drift distance measured. However, since the measurement

was only taken when the user looked at the image target, for individual trials, a stepwise function was also produced. These 6 trials were averaged together to find a general trend for the uncorrected drift condition, shown in Fig. 11. Since only 6 trials were performed for this condition, small step functions are still present in the graph. However, a clear linear trend was found from the data. From the linear regression line, it was shown that drift accumulated a rate of 0.0445 mm per second with a R^2 value of 0.9897.

Both the corrected and uncorrected conditions of the collected drift data showed a clear linear trend as drift accumulated over time. The rate at which this drift accumulated was similar for both conditions, as shown in Figs. 10 and 11. This shows that correcting the drift throughout the trial is critical to ensure that the work instructions are accurately spatially located. If drift is not corrected, the error in displacement of spatially located content will follow the accumulated drift trendlines shown in Figs. 10 and 11. However, if drift is corrected periodically, the displacement error in spatially located content will be minimized to the small amount of drift accumulated between recalibration events, shown as T1 in Fig. 10, on the order of 10 mm.

4.3 Data visualization tool

The data visualization tool was developed to virtually recreate the user’s assembly session. From this recreation, an analyst, referring to anyone who would use this tool to review the data, may validate the instructions and explore any potential trends that occurred during the session. This analysis would be highly advantageous to identify if the user is prepared but also to pinpoint inefficiencies in the assembly process. The data visualization tool was developed using Unity3D for this purpose. Data from the HoloLens, Empatica E4, and post-

Fig. 10 Drift over 15 minutes for the corrected condition

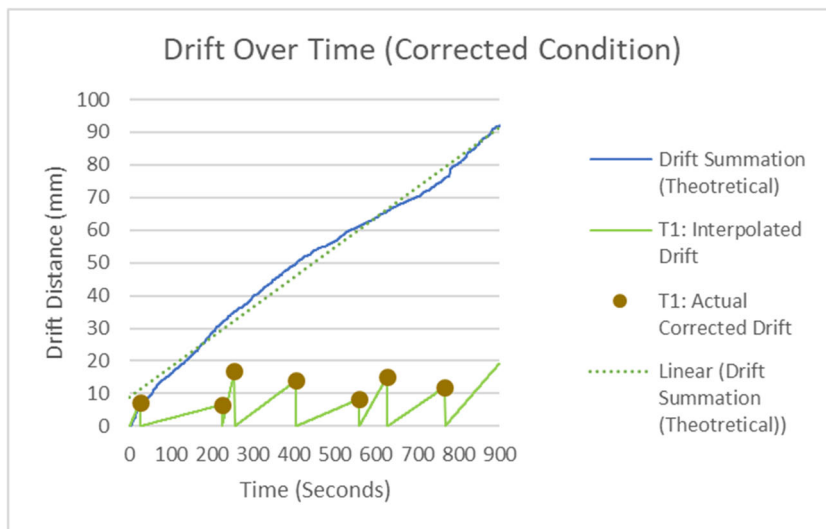
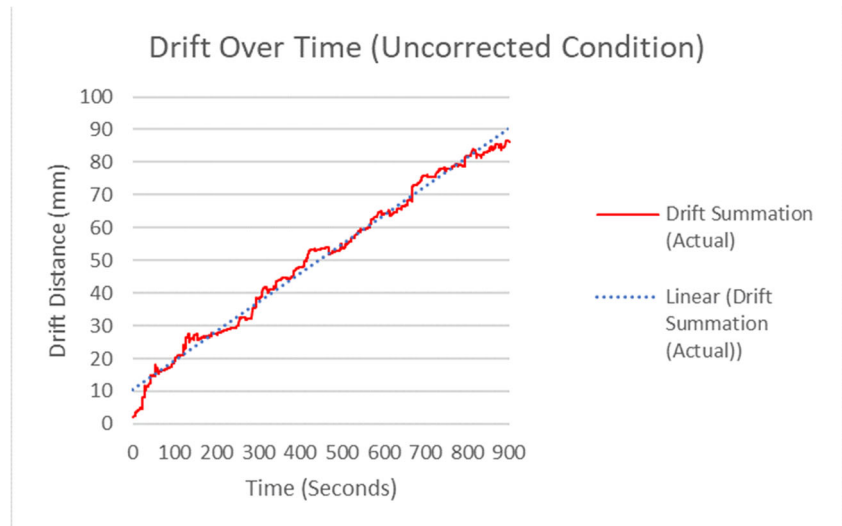


Fig. 11 Drift over 15 minutes for the uncorrected condition

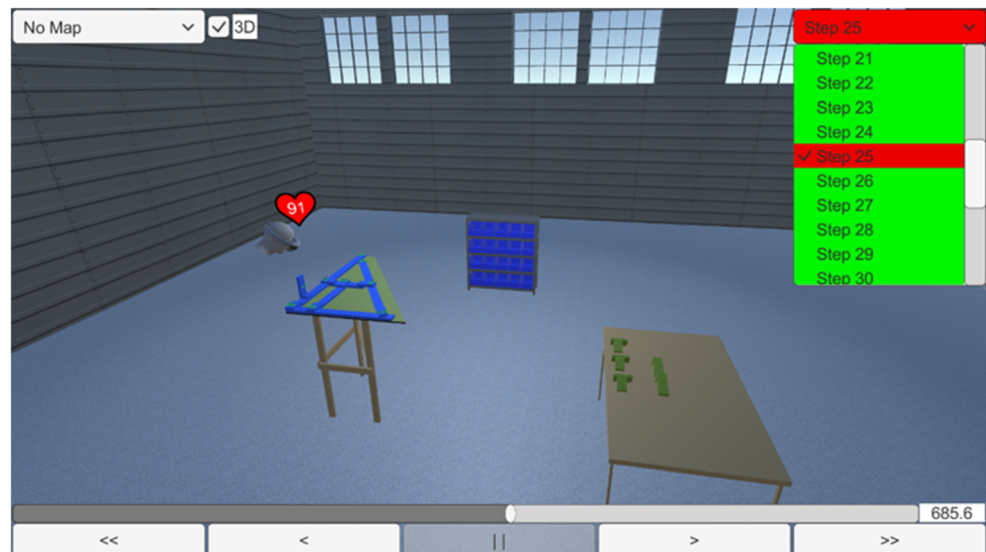


processed assembly error data was parsed and synced within the application.

For the fusion of data within the application, there were various tools for analysis. A playback tool was available to scrub through the timeline of the training session; see Fig. 12. A scrubber and timestamp showed the current time location. The analyst could then use buttons to pause, play, play-2x, rewind, and rewind-2x to navigate along the timeline. The analyst was also able to navigate around the environment to view the session from various vantage points. To achieve this, various technical and usability challenges had to be overcome. An avatar, representing the user, was displayed and followed the same path of the user within the virtual recreation of the training environment. To accomplish this, the positional and rotational data set were parsed into the application from the CSV files generated during the trials. The pose of the avatar

was then transformed relative to the new virtual environment to ensure the collected data points in the real world aligned with the virtual environment. In addition, the heart rate of the user was displayed over the avatar's head to show their physiological responses during the session; see Fig. 13. To enable this functionality, the heart rate data collected from the Empatica E4 was parsed into the application and synchronized with the positional data set via the UTC timestamps. Finally, the post-processed assembly error data and step time data set were synchronized with the previously mentioned data sets using the same UTC timestamp method. This allowed the analyst to quickly navigate to specific points where the user may have struggled or misunderstood directions. The navigational tools and data represented would allow the analyst to quickly assess if the instructions were delivered accurately and identify high-level trends during the session.

Fig. 12 Visualization tool UI



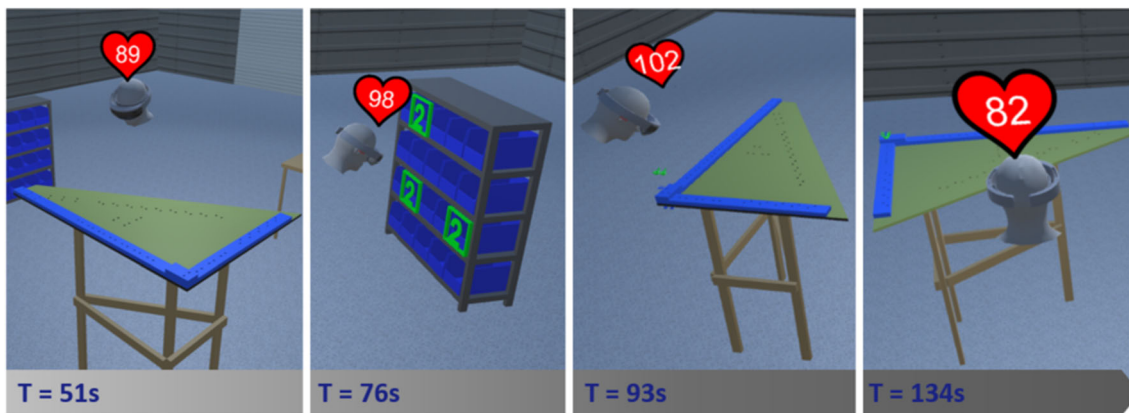


Fig. 13 Analysis of heart rate data over time

To further explore these trends within the training session, heat maps were available for positional and heart rate data. To create the positional heat map, a square mesh was generated over the virtual recreation of the training environment. Each vertex of this mesh was then compared with the positional data set and was raised in the upwards *y* direction if it was within a threshold distance, shown in Fig. 14. The height of the mesh was then normalized to enhance visibility. To generate the heart rate heat map, the same square mesh was used. However, to get positional context of the heart rate data set, the heart rate data set was synchronized with the positional data set, utilizing the UTC timestamps included in each data set. From there, the average heart rate was found for each point, and the closest vertex to that location was raised to that value. 2D heat maps were also available from storing the value of each vertex in an array and not raising the vertex, shown in Fig. 15. A linear interpolation of color is used to show the variation from high to low, red representing high values and blue representing low values. Since both maps represent

where the participant traveled, any unusual points could represent issues within the tracking system of the HoloLens during the training session.

The development of this application using Unity3D allowed for multi-platform build support. The data visualization tool’s codebase was build platform agnostic meaning that it could be deployed on any platform supported by Unity3D. While a major use case of this application was to give additional evidence during statistical analysis, commonly performed on a computer, it could also be highly beneficial to make a quick high-level analysis and validation in the field. For this purpose, the application was deployed on a commodity smartphone. This would allow the analyst the ability to review the session with all the collected data shortly after the training session to make sure that the instructions were being delivered accurately so adjustments may be made during the training process.

The capabilities of this application present many unique opportunities regarding data analysis. The purpose of this

Fig. 14 3D positional heat map

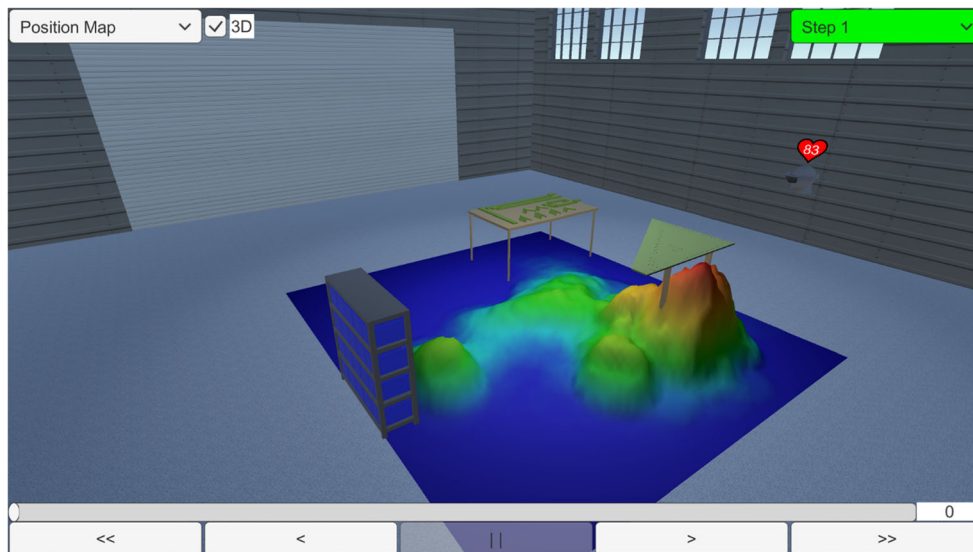
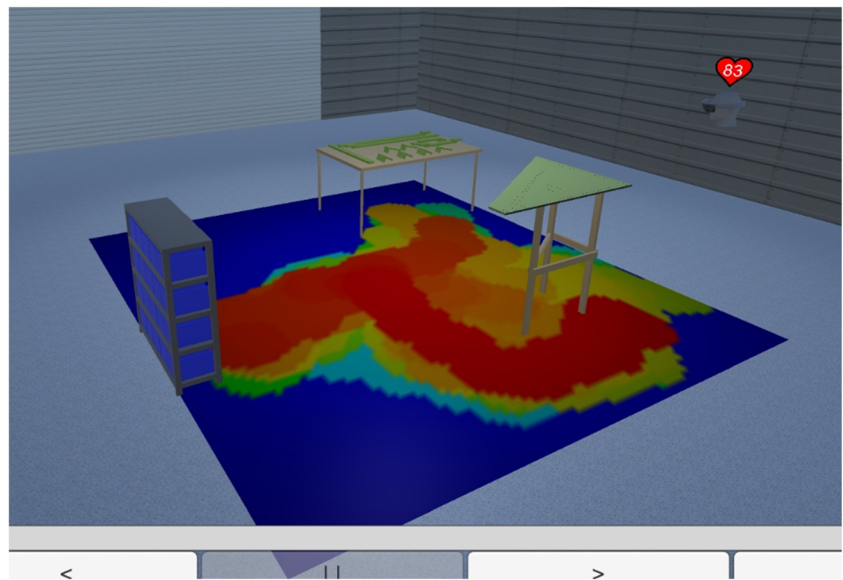


Fig. 15 2D average heart rate heat map



application was to provide a high-level analysis to augment traditional statistical methods by helping to explain trends already found or generating new leads to explore further. In addition, time studies would also benefit greatly from a tool of this nature. Instead of the time-consuming process of overserving how a task is performed, the analyst could quickly scrub through the session to identify major trends paired along with the heat maps generated.

For example, a prior study, using the same experimental setup, found that participants had significantly higher average heart rates on parts picking and assembly steps compared with parts placing steps [39]. However, there was no explanation as to why this had occurred. By combining positional, rotational, step time, and heart rate data into the data visualization tool, a hypothesis could be formed. Figure 13 shows a timeline of a user who experienced a change in heart rate over the course of a picking and assembly step. The user moved from a placing step to a picking step requiring them to bend down to reach the fasteners required. Their heart rate spiked, and it was possible to see this physiological response continue into the assembly following step. After this, their heart rate returned to normal. From this high-level analysis, it is possible to hypothesize that the increased heart rate was caused by poor ergonomic conditions during the picking step. While the statistical analysis in Hoover et al. showed significantly higher heart rates for picking and assembly steps, this additional evidence shows that the elevated assembly step heart rate may be due to the physiological response carrying over from the picking step preceding it [39].

5 Conclusion and future work

The research presented in this paper explored novel solutions to overcome the limitations of the Microsoft HoloLens 1 for

delivering assembly work instructions and an approach for analyzing the data collected during the training session. Due to the HoloLens' various limitations including input, user interface, navigation, spatial registration, and occlusion, the work instructions had to be developed in a way that mitigated these issues. Specifically, the spatial registration is a key to deliver accurate work instructions, and the methodology described in this paper provides a novel approach to ensure consistent spatial registration over an extended period. Furthermore, the approach described in this paper allowed for an accurate delivery of work instructions for a 46-step mock wing assembly. To ensure these work instructions were delivered accurately through the HoloLens, a data visualization application was developed to validate the assembly training session along with an analysis of spatial drift data collected through 83 trials. In addition, it is important to ensure the user is properly trained for the assembly task. Through the same data visualization application, an analyst may explore trends to establish the overall readiness level of the user for the field.

Future work will explore new transparent HMDs that improve upon the limitations of the HoloLens. While this paper gives a novel approach to mitigating the current limitations, improvements in hardware would likely further the success of AR-delivered work instructions. The Meta 2, Daqri Smart Glasses, Magic Leap, and HoloLens 2 are four such products that have been released after the HoloLens 1 and boast improved performance. Assessing the limitations of these new transparent HMDs may allow for a better understanding of the ideal method of delivering AR work instructions. However, despite any hardware improvements that these devices may bring, additional work is needed to explore the methods of input for AR work instruction applications. Ideally, the user would be able to deliver input to the device hands-free. The

input methods for current transparent HMDs often require a clicker device or unintuitive gesture controls. The usability of these systems could be greatly improved by a better understanding of how users will best interact with them.

Funding information This work has been funded by the Boeing Company.

References

- Henderson S, Feiner S (2011) Exploring the benefits of augmented reality documentation for maintenance and repair. *IEEE Trans Vis Comput Graph* 17:1355–1368. <https://doi.org/10.1109/TVCG.2010.245>
- Ockerman JJ, Pritchett AR (1998) Preliminary investigation of wearable computers for task guidance in aircraft inspection. In: *IEEE Proceedings of the 2nd International Symposium on Wearable Computers*. IEEE Comput. Soc, pp 33–41
- Dini G, Mura MD (2015) Application of augmented reality techniques in through-life engineering services. *Procedia CIRP* 38:14–23. <https://doi.org/10.1016/j.procir.2015.07.044>
- Nee AYC, Ong SK (2013) Virtual and augmented reality applications in manufacturing. *IFAC Proc* 46:15–26. <https://doi.org/10.3182/20130619-3-RU-3018.00637>
- Hoover M, Miller J, Gilbert S, et al (2018) Evaluating augmented reality assembly instructions delivered via Microsoft HoloLens. In: *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*
- Azuma R (1997) A survey of augmented reality. *Presence Teleoperators virtual Environ*
- Caudell TP, Mizell DW (1992) Augmented reality: an application of heads-up display technology to manual manufacturing processes. In: *Proceedings of the Twenty-fifth Hawaii International Conference on System Sciences*, pp 659–669 vol 2
- Azuma RTA (1997) Survey of augmented reality. *Presence Teleoperators Virtual Environ* 6:355–385
- Van Krevelen DWF, Poelman R (2010) A survey of augmented reality technologies, applications and limitations. *J Virtual Real* 9:1–20. <https://doi.org/10.1155/2011/721827>
- Palmarini R, Erkoyuncu JA, Roy R, Torabmostaedi H (2018) A systematic review of augmented reality applications in maintenance. *Robot Comput Integr Manuf* 49:215–228. <https://doi.org/10.1016/j.rcim.2017.06.002>
- Baird KM, Barfield W (1999) Evaluating the effectiveness of augmented reality displays for a manual assembly task. *Virtual Reality* 4:250–259. <https://doi.org/10.1007/BF01421808>
- Loch F, Quint F, Brishtel I (2016) Comparing video and augmented reality assistance in manual assembly. In: *Proceedings - 12th International Conference on Intelligent Environments, IE 2016*. pp 147–150
- Tatić D, Tešić B (2017) The application of augmented reality technologies for the improvement of occupational safety in an industrial environment. *Comput Ind* 85:1–10. <https://doi.org/10.1016/j.compind.2016.11.004>
- De Crescenzo F, Fantini M, Persiani F et al (2011) Augmented reality for aircraft maintenance training and operations support. *IEEE Comput Graph Appl* 31:96–101. <https://doi.org/10.1109/MCG.2011.4>
- Wagner D, Pintaric T, Ledermann F, Schmalstieg D (2005) *Towards massively multi-user augmented reality on handheld devices*. Springer, Berlin, Heidelberg, pp 208–219
- Microsoft (2018) Microsoft HoloLens | The leader in mixed reality technology. <https://www.microsoft.com/en-ca/hololens>. Accessed 3 Jun 2018
- Daqri (2018) Smart Glasses™ - DAQRI. <https://daqri.com/products/smart-glasses/>. Accessed 3 Jun 2018
- Meta (2018) Meta | Augmented Reality. <http://www.metavision.com/>. Accessed 3 Jun 2018
- Leap M Magic Leap. <https://www.magicleap.com/>. Accessed 6 Jan 2019
- Evans G, Miller J, Iglesias Pena M, et al (2017) Evaluating the Microsoft HoloLens through an augmented reality assembly application. In: *Degraded environments: sensing, processing, and display*. International Society for Optics and Photonics, p 101970V
- Tang A, Owen C, Biocca F, Mou W (2003) Comparative effectiveness of augmented reality in object assembly. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. pp 73–80
- Azuma R, Baillo Y, Behringer R, Feiner S, Julier S, MacIntyre B (2001) Recent advances in augmented reality. *IEEE Comput Graph Appl* 21:34–47. <https://doi.org/10.1109/38.963459>
- Comport AI, Marchand E, Pressigout M, Chaumette F (2006) Real-time markerless tracking for augmented reality: the virtual visual servoing framework. In: *IEEE Transactions on Visualization and Computer Graphics*. pp 615–628
- Martin P, Marchand E, Houlier P, Marchal I (2014) Decoupled mapping and localization for augmented reality on a mobile phone. In: *Proceedings - IEEE Virtual Reality*. IEEE, pp 97–98
- Rolland JP, Fuchs H (2000) Optical versus video see-through head-mounted displays in medical visualization. *Presence Teleoperators Virtual Environ* 9:287–309. <https://doi.org/10.1162/105474600566808>
- Shah MM, Arshad H, Sulaiman R (2012) Occlusion in augmented reality. *8th Int Conf Inf Sci Digit Content Technol (ICIDT 2012)* 2: 372–378
- Wloka MM, Anderson BG (1995) Resolving occlusion in augmented reality. In: *Proceedings of the 1995 Symposium on Interactive 3D Graphics - SI3D '95*. pp 5–12
- Microsoft (2018) HoloLens hardware details - Mixed Reality | Microsoft Docs. <https://docs.microsoft.com/en-us/windows/mixed-reality/hololens-hardware-details>. Accessed 22 May 2018
- Unity Technologies (2018) Fast Facts. <https://unity3d.com/public-relations>. Accessed 10 May 2017
- Microsoft (2018) Unity development overview - Mixed Reality | Microsoft Docs. <https://docs.microsoft.com/en-us/windows/mixed-reality/unity-development-overview>. Accessed 22 May 2018
- Soussi L, Spijkerman Z, Jansen S (2016) A case study of the health of an augmented reality software ecosystem: Vuforia. In: *International conference of software business*. Springer, Cham, pp 145–152
- Microsoft (2018) Using Vuforia with Unity - Mixed Reality | Microsoft Docs. <https://docs.microsoft.com/en-us/windows/mixed-reality/vuforia-development-overview>. Accessed 22 May 2018
- Rautaray SS, Agrawal A (2012) Vision based hand gesture recognition for human computer interaction: a survey. *Artif Intell Rev* 43: 1–54. <https://doi.org/10.1007/s10462-012-9356-9>
- Cheng H, Yang L, Liu Z (2016) Survey on 3D hand gesture recognition. *IEEE Trans Circuits Syst Video Technol* 26:1659–1673. <https://doi.org/10.1109/TCSVT.2015.2469551>
- Sekar LPN, Santos A, Beltramello O (2015) IMU drift reduction for augmented reality applications. In: *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*. Springer, Cham, pp 188–196
- Vuforia (2018) Optimizing target detection and tracking stability. <https://library.vuforia.com/articles/Solution/Optimizing-Target-Detection-and-Tracking-Stability.html>. Accessed 22 May 2018

37. MacAllister A, Hoover M, Gilbert S, et al (2017) Comparing visual assembly aids for augmented reality work instructions. In: Comparing Visual Assembly Aids for Augmented Reality Work Instructions. Interservice/Industry Training, Simulation, and Education Conference (IITSEC)
38. Garbarino M, Lai M, Bender D, et al (2015) Empatica E3 - a wearable wireless multi-sensor device for real-time computerized bio-feedback and data acquisition. In: Proceedings of the 2014 4th International conference on wireless mobile communication and healthcare - “Transforming healthcare through innovations in mobile and wireless technologies”, MOBIHEALTH 2014. pp 39–42
39. Hoover M, MacAllister A, Holub J, et al (2016) Assembly training using commodity physiological sensors. In: Assembly Training using Commodity Physiological Sensors. Interservice/Industry Training, Simulation, and Education Conference (IITSEC)

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.