Felix von Haxthausen*, Yenjung Chen and Floris Ernst

Superimposing holograms on real world objects using HoloLens 2 and its depth camera

Abstract: Augmented Reality glasses such as HoloLens 2 may provide visual guidance during surgical interventions. To superimpose the holograms on real world objects (RWO), for instance a patient, spatial registration is required. In this work, we propose an approach to automatically register a hologram to the according RWO. To this end, the framework utilizes the depth camera of HoloLens 2 to acquire the point cloud (PC) of the RWO. A novel and recently published PC registration algorithm allows to register the PC of the RWO and the hologram after a rough initial placement without any need for pre-processing or outlier removal. The approach is evaluated by measuring displacements between certain known positions of the hologram and the RWO. The first metric relies on measuring points using an optically tracked stylus while the second is based on visually perceived positions. The median displacements were 22.3 mm, 35.6 mm, and 13.3 mm for the x-, y-, and z-axes in the first metric and 8.1 mm, 4.3 mm, and 11.9 mm for the second metric. Even though the accuracy is not yet adequate for many surgical interventions, the framework provides an initial step for a convenient marker less registration of holograms to an RWO.

Keywords: Augmented Reality, Registration, Point Cloud, Time-of-Flight camera, Research Mode

https://doi.org/10.1515/cdbme-2021-1024

1 Introduction

Augmented reality (AR) systems have the potential to support surgical interventions by adding visual assistance during those procedures. HoloLens 2 (Microsoft), as a head mounted computer with an optical see-through display, is such an AR system that allows to place holograms (or virtual objects) in the real world and allows the user to use hand gestures and voice commands for interaction. The registration and superimposition of holograms and the according real-world objects (RWO) is a crucial part of such a system. This task can be solved by AR markers [1], optical markers [2] or point clouds (PC). The former two require preparatory work by means of a registration between the marker and the virtual object as well as the manual placement of the markers. As HoloLens 2 contains a Time-of-Flight depth camera, it appears obvious to use it for a PC-based registration between the virtual object and an RWO, especially because this approach does not require any preparatory work.

Therefore, we propose a framework that utilizes the depth camera of HoloLens 2 to acquire the PC of an RWO and to register it with the PC of a corresponding virtual object. Our framework uses a recently published PC-based registration approach that does not require any pre-processing, sub-sampling, or outlier removal [3]. The goal of this study was to prove feasibility, namely that the proposed framework can register the virtual and real-world objects, and second, to evaluate the final superimposition by measuring the displacement between specific points visible on both objects.

***Corresponding author: Felix von Haxthausen:** Institute for Robotics and Cognitive Systems, University of Lübeck, Lübeck, Germany, E-mail: vonhaxthausen@rob.uni-luebeck.de
Yenjung Chen and Floris Ernst: Institute for Robotics and Cognitive Systems, University of Lübeck, Lübeck, Germany

2 Material and methods

2.1 System description and workflow

To access the PC of the depth camera, HoloLens' research mode [4] was incorporated in an application developed with Unity 2019.4.15f1 [5]. The depth camera operates in two modes: the AHAT (Articulated Hand Tracking) and long throw mode with 45 and 1-5 frames per second, respectively. As the names indicate, the modes are supposed to be used for different distances. Because the AHAT mode is more accurate and our application allows to acquire the PC at distances < 1 m, we decided to focus on this mode. After starting the application, the user can initially place the virtual object close to the RWO using hand gestures (Figure 1a). Thereafter, voice commands allow to acquire and display the PC of the RWO $\mathcal{P}_{RWO}$ from different point of views (Figure 1b). A user interface allows the user to connect to a MATLAB R2021a instance running on a computer. The acquired $\mathcal{P}_{RWO}$ and the pose of the virtual object are wirelessly sent to MATLAB. $\mathcal{P}_{RWO}$ and the PC of the virtual object $\mathcal{P}_{VO}$ are registered on the computer using the previously mentioned registration algorithm. Not having to remove outliers is of special interest, as the AHAT mode shows depth wrapping beyond one meter [6]. After registration, the resulting transformation is sent back to HoloLens and the virtual object is transformed such that it is superimposed on the RWO.

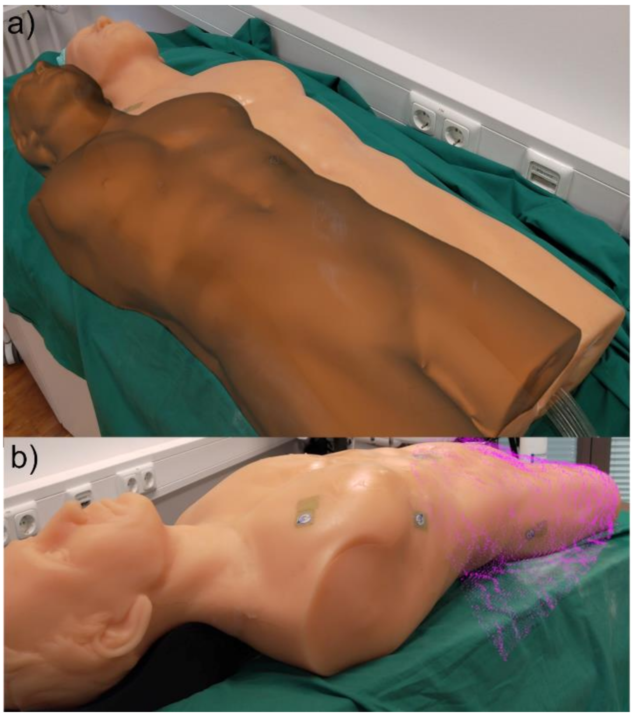

Figure 1: The HoloLens 2 user's point of view of a) the virtual object (left) and the RWO (right) after the initial manual placement and b) the RWO and a single acquisition of PC (magenta).

2.2 Evaluation

The aim of the evaluation was to quantify the displacements between certain known positions between the virtual object and the RWO. To this end, we used a torso phantom (FAST Ultrasound Training Model, Blue Phantom) and placed six lead balls (hereafter called markers) of 1 mm diameter (SL-10, Suremark) on the body surface. To create the virtual object, a 3D scan of the phantom was performed using a handheld scanner (Artec Leo, Artec3D). The 3D model including texture allowed to manually locate the markers on the virtual model. Afterwards, the model was exported as an OBJ and PLY file for Unity and MATLAB, respectively.

Finally, the objective is to measure the displacements between the virtual and real markers after the registration. The virtual marker positions ${}^{U}p_{VM}$ in the Unity coordinate system are saved within the application. To acquire the real marker positions ${}^{O}p_{RM}$ in an optical tracking coordinate system (fusionTrack 500, Atracsys), we pointed an optically tracked stylus to the lead balls. To calculate the displacement, it is necessary to acquire the positions in a common reference frame. For this reason, we performed a hand eye calibration [7] to calculate the transformation from the optical tracking coordinate system to the Unity coordinate system ${}^{O}T_{U}$. An optical marker was attached to HoloLens 2 and 50 different poses of the AR glasses ${}^{U}T_{HL}$ and the optical marker ${}^{O}T_{OM}$ were acquired (Figure 2).

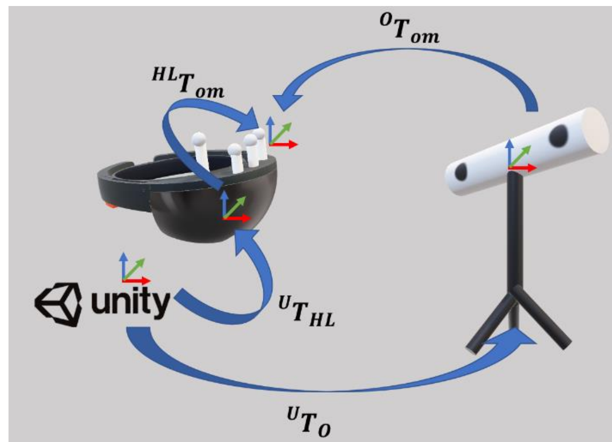

Figure 2: The transformations (blue arrows) used and calculated during the hand eye calibration for the evaluation. While ${}^{O}T_{om}$ and ${}^{U}T_{HL}$ can be measured, the hand eye calibration allows to calculate ${}^{U}T_{O}$ and ${}^{HL}T_{om}$.

Finally, the displacement vector d_{OT} based on the optical tracking system can be calculated using

$$\boldsymbol{d_{OT}} = {}^{U}\boldsymbol{T_{O}}\ {}^{O}\boldsymbol{p_{RM}} - {}^{U}\boldsymbol{p_{VM}} \quad (1)$$

Note that whenever a Unity application is started, the origin of the coordinate system is located at the current HoloLens users' pose. Thus, a calibration would need to be performed each time the application is terminated. To overcome this drawback, we implemented spatial anchors [8] that allow to have a common coordinate system throughout different applications and HoloLenses.

As a second metric, the subjectively perceived displacements are measured. After the registration and visually hiding the virtual object and markers, we manually placed additional virtual spheres of 1 mm diameter on the real markers within the Unity application. Then, the positions of virtual spheres ${}^{U}p_{VS}$ and the virtual markers ${}^{U}p_{VM}$ are saved and the displacement vector d_{SP} can be calculated without any transformation needed:

$$\boldsymbol{d_{SP}} = {}^{U}\boldsymbol{p_{VS}} - {}^{U}\boldsymbol{p_{VM}} \quad (2)$$

3 Results

The spatial anchors allowed to have a common coordinate system throughout different applications. Therefore, one calibration is potentially enough for different instances of the application. A rough initial alignment of the virtual object was sufficient to register both objects using the proposed framework (Figure 3).

Figure 3: The HoloLens 2 user's point of view of the virtual object with the virtual markers superimposed on the real object after the PC-based registration.

The registration of the PCs took around 1.2 s for the clouds consisting of 738 125 and 165 959 vertices. The median displacements between the virtual and real markers were 22.3 mm, 35.6 mm, and 13.3 mm for the x-, y-, and z-axes, respectively (Figure 4a). A systematic drift is visible in each direction. Figure 4b shows the subjective perceived displacements of the virtual markers and virtual spheres. The median values were 8.1 mm, 4.3 mm, and 11.9 mm for the x-, y-, and z-axes, respectively.

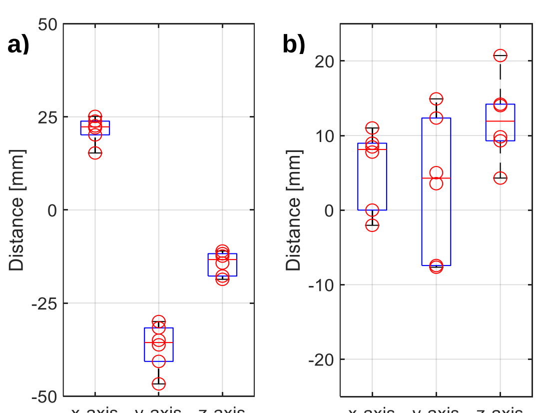

Figure 4: Boxplots with scatter points of d_{OT} and d_{SP} along the different axes from a) points acquired with the optical tracking system and b) manually placed virtual spheres on the real markers relative to the virtual markers after registration.

4 Discussion

The proposed framework was able to register the virtual and real objects using the depth camera of HoloLens 2 without prior pre-processing or outlier removal (Figure 5). However, the displacements measured using the first metric are relatively high. We believe that the cause is not the registration process itself but the evaluation workflow. It relies on the assumption that the registration between the Unity and optical tracking coordinate system is accurate. This registration depends on the accuracy of the spatial anchors and the hand eye calibration. With a perfect transformation ${}^{O}T_{U}$, the real marker positions acquired with the optical tracking system in the Unity coordinate system ${}^{U}p_{RM}$ should be close to the according markers on the point cloud $\mathcal{P}_{RWO}$. To verify that the registration is not perfect, the displacement between ${}^{U}p_{RM}$ and the markers on $\mathcal{P}_{RWO}$ could be calculated. Because of the limited quality of $\mathcal{P}_{RWO}$, it is not possible to identify the markers on $\mathcal{P}_{RWO}$. Instead, we calculated Hausdorff distances d_H between ${}^{U}p_{RM}$ and $\mathcal{P}_{RWO}$ (Table 1). Even though the Hausdorff distances are probably even lower than the true displacement, the mean d_H is 14 mm. Another indicator for an inaccurate registration is the systematic drift in the displacements. Even after several calibration attempts, a relatively high d_{OT} was observed.

Table 1: Hausdorff distances d_H between the real markers in the Unity coordinate system ${}^{U}P_{RM}$ and the PC acquired with the depth camera of HoloLens 2 $\mathcal{P}_{RWO}$.

Marker	1	2	3	4	5	6
d_H [mm]	10.5	4.4	14.5	38.1	15.7	1.2

On the other hand, the values of the subjectively perceived displacements are lower, partially in the sub-centimetre area. As this metric does not rely on any registration but is computed completely in the Unity space, we assume that those displacements are closer to the true values. Nevertheless, the virtual spheres are manually placed. Therefore, the measured displacements still have a human-influenced error. Another source of error applicable to both metrics may be a limited absolute accuracy of the AHAT mode.

To account for movement of the RWO it would be feasible to implement a workflow that performs registrations if the RWO has moved. However, in this work we performed a registration with a fixed pose of the RWO to validate feasibility and measure accuracy.

For a clinical application with sub-millimetre accuracy required, the accuracy of this framework is not satisfying. However, it shows a general feasibility without cumbersome manual outlier removal. Possible ways for improvement could

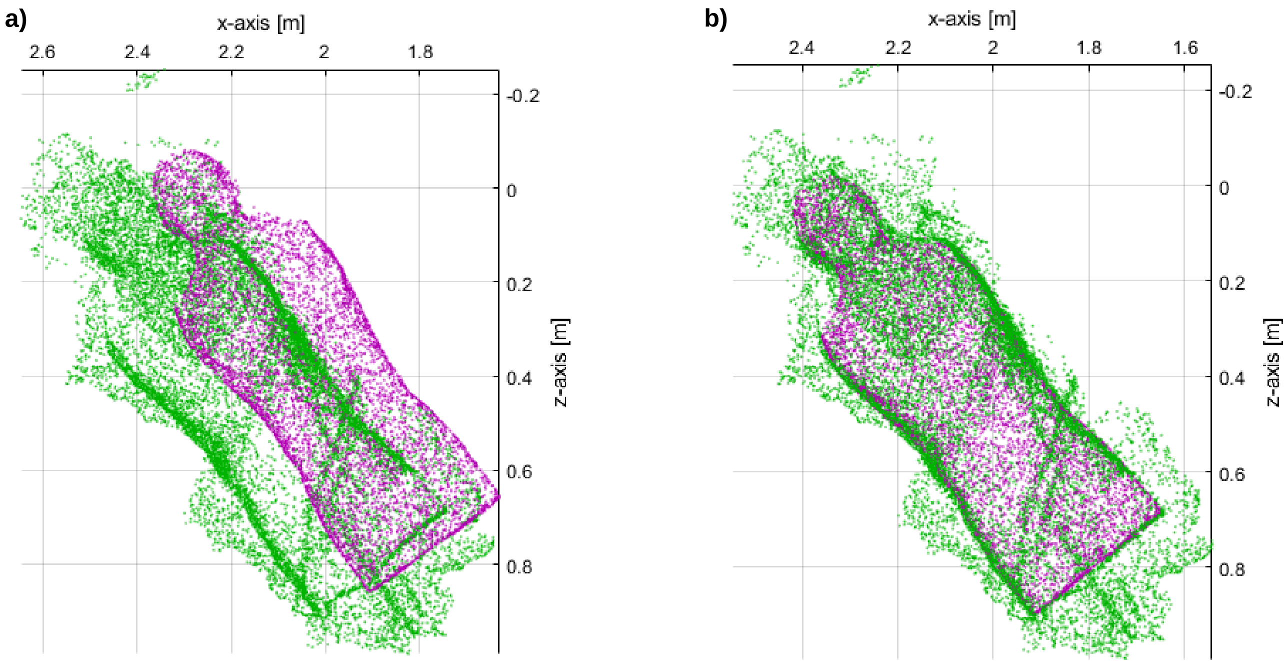

Figure 5: PCs of the virtual object (magenta) and the RWO (green) obtained with the depth camera of HoloLens 2 a) before and b) after registration. The registration algorithm used [3] does not require any pre-processing or outlier removal. Note that for visualization purposes, both PCs were down sampled.

be the pre-processing of the PC before registration, a registration approach including the texture information and potentially combining the AHAT and long throw modes for PC acquisition.

5 Conclusion

In this work, we could show a framework for HoloLens 2 that automatically registers an RWO and a virtual object based on the point cloud acquired with the depth camera of the AR glasses. No pre-processing or outlier removal was needed to perform the registration. The resulting superimposition may improve patient safety and ergonomics for the physician. However, the accuracy of the system is in the range of centimetres. Therefore, future work will focus on improving the accuracy by incorporating point clouds with texture information as an additional input parameter for the registration approach.

Author Statement

Research funding: This study was supported by the German Federal Ministry of Education and Research (grant number 13GW0228). Conflict of interest: Authors state no conflict of interest.
Informed consent: NA
Ethical approval: NA

References

[1] Moreta-Martinez R, García-Mato D, García-Sevilla M, Pérez-Mañanes R, Calvo-Haro J, Pascau J. Augmented reality in computer-assisted interventions based on patient-specific 3D printed reference. Healthc Technol Lett. 2018;5(5):162–166.

[2] Kunz C, Maurer P, Kees F, Henrich P, Marzi C, Hlaváč M, et al. Infrared marker tracking with the HoloLens for neurosurgical interventions. Curr Dir Biomed Eng. 2020;6(1):20200027. Available from: https://www.degruyter.com/view/journals/cdbme/6/1/article-20200027.xml

[3] Jauer P, Kuhlemann I, Bruder R, Schweikard A, Ernst F. Efficient registration of high-resolution feature enhanced point clouds. IEEE Trans Pattern Anal Mach Intell. 2019;41(5):1102–1115.

[4] Ungureanu D, Bogo F, Galliani S, Sama P, Duan X, Meekhof C, et al. HoloLens 2 eesearch mode as a tool for computer vision research. CoRR. 2020;abs/2008.1.

Available from: https://arxiv.org/abs/2008.11239

[5] Gu W. HoloLens2-ResearchMode-Unity: unity plugin for using research mode functionality in HoloLens 2. Modified based on HoloLens2ForCV. [Internet]. [cited 2021 May 25]. Available from: https://github.com/petergu684/HoloLens2-ResearchMode-Unity

[6] Pollefeys M. Microsoft HoloLens 2: Improved Research Mode to facilitate computer vision research - Microsoft Research [Internet]. 2020 [cited 2021 May 25]. Available from: https://www.microsoft.com/en-us/research/blog/microsoft-hololens-2-improved-research-mode-to-facilitate-computer-vision-research/

[7] Ernst F, Richter L, Matthäus L, Martens V, Bruder R, Schlaefer A, et al. Non-orthogonal tool/flange and robot/world calibration. Int J Med Robot Comput Assist Surg. 2012;8(4):407–420. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1002/rcs.1427

[8] Azure spatial anchors overview - azure spatial anchors | microsoft docs [Internet]. [cited 2021 May 25]. Available from: https://docs.microsoft.com/en-us/azure/spatial-anchors/overview