4D Holographic Tutorials

Andrin Bertschi
bandrin@ethz.ch

Benjamin Hardin
bhardin@ethz.ch

Lukas Walker
luwalker@ethz.ch

Nicolas Wicki
nwicki@ethz.ch

ETH Zürich
Rämistrasse 101 8092 Zürich

Abstract

In recent years, augmented and virtual reality devices have seen significant improvements in usability and processing power, directly leading to increased usage in industrial settings. These devices have shown promising opportunities for creating immersive tutorials for training or general teaching. However, tutorial creation has been mostly restricted to professionals with graphic or 3D design experience. In this paper, we present a novel application to enable diverse users to create 4D Holographic Tutorials for the HoloLens 2. To the authors' best knowledge, no previous work exists trying to achieve a similar result. We focused on providing an easy to use user-interface, which allows users to create and watch tutorials step-by-step. Not only does it allow creating tutorials for specific locations, but it also provides the functionality to use the same tutorial for new locations. We describe the design of core features and publish the code as open-source software. To show the prevalent usability, a user study qualitatively measures the user experience. Given the broad domain of potential applications and research avenues, we lay a foundation for future work to create applications of this technology to diverse training scenarios.

1. Introduction

Mixed Reality has attracted increasing attention in today's organizations, and continues to do so. Especially in fields like medicine and manufacturing, where training costs of new employees are high, and many tasks are carried out routinely, Mixed Reality tutorials can facilitate education and serve as a support in task completion. Examples include Intel's manufacturing sites [6] to handle maintenance, repair tasks, troubleshooting, or remote communication, and BMW using AR goggles in manufacturing training processes [5]. In recent times, the global Covid pandemic made it difficult for experts to meet trainees personally. Mixed Reality offers a great opportunity as a way to train staff remotely. In this paper, we present an application for individuals to create a Mixed Reality training tutorial directly on the HoloLens. Once recorded, other users can view these tutorials overlaid on their current environment, leading to a more immersive and interactive training environment.

Modern Mixed Reality devices belong to a relatively young branch of technology, with many users being inexperienced in working with three-dimensional user interfaces. Thus, high user-friendliness and appropriate user-experience design are of the utmost importance as users experience the devices for the first time. Therefore, in this project, we strove to create a user interface that intuitively lets users record and replay simple hand movements. It is targeted at experts in any field, who need to capture their workflow to present it to trainees who are not present at the same time. When an expert has recorded their workflow, trainees can watch the objective tasks in first-person view both during training and later as a supporting guide during productive work.

The main contributions of this paper are:

- We present the first application for step-by-step tutorial creation.
- We allow the user to not only capture one's hand movements, but also to interact with various objects provided.
- We provide an easy to use user-interface evaluated by a user study.
- Our application allows for playback location invariance by orienting itself at a new location.
- We provide a proof-of-concept feature to extract real-life objects from a scene.

2. Related Work

There have been numerous studies evaluating the effectiveness and performance of Mixed Reality tutorials, many using the HoloLens. One such study [8] tested the applicability of a HoloLens 3D hand tutorial to teach surgical knot tying skills, showing promising results and giving an outlook on various applications for education in medicine. Likewise, another study explored using the HoloLens as a method for exploring and drilling the human temporal bone and found that the immersion of Augmented Reality (AR) made it ideal for practicing complex surgical operations in a real operating room without the need for an invasive procedure [9]. Another study from the automotive industry measured the impact of Mixed Reality learning environments and found that it improved the overall performance of trainees compared to traditional teaching [7]. They managed to reduce errors and completion time while performing complex tasks, with most participants giving positive feedback on the Mixed Reality training. In one study, participants used the HoloLens as a way to learn about nutrition and digestion [10]. It was found that 60% of participants preferred the HoloLens approach to a traditional computer desktop learning experience, indicating the potential of using AR for certain teaching scenarios. Another study has found that most participants found that AR tutorials were a satisfactory way to learn, but that higher computer science knowledge of participants led to greater satisfaction [14]. This may indicate that work still needs to be done to make these tutorial methods accessible and easy to use if the intended users have non-technical backgrounds.

However, these studies highlight the need for further software development in the field of Mixed Reality (MR). While very valuable in theory, many MR applications are not intuitive to use. Additionally, the majority of potential users have never used or experienced state-of-the-art MR devices. Generally, screen-independent three-dimensional user interfaces lack time-tested best practices, which further supports the need to explore such new concepts and to invest in user experience design. Likewise, specific tutorials for certain domains are created by domain experts. However, there exists no way for non-technical users to create AR tutorials for diverse demonstrations. Our project aims to address this gap.

3. Approach

Our approach includes various features such as an intuitive user interface, recording and playback of tutorials, location invariance, and object extraction from the scene. The following paragraphs will discuss into their implementations and usage scenarios.

3.1. Technologies

Our application was implemented using the Unity Engine version 2019.4.39f1 [11] with further functionalities provided by the Mixed Reality Toolkit version 2.7.3 [3]. To get access to HoloLens sensor data, we used the Microsoft HoloLens 2 Research Mode [12] and access short-throw (AHAT) sensor data stream. Given the Research Mode functionality is exposed as C++ APIs, we bundle access to sensor data as Windows Runtime (WinRT) language projections and access the WinRT component in C# through Unity's IL2CPP [1] build support. Even though newer versions of the Unity Engine exist, we found that some technologies offered by the Microsoft HoloLens 2 Research Mode are more suited for 2019 versions of Unity. In particular, our implementation relies on support for Windows XR SDK for Windows MR [4], which was marked as depreciated in 2019 and completely removed in later releases of Unity. The source code is released as open-source software and is accessible on Github[1].

3.2. User Interface

The main user interface consists of a single panel, which controls various extensions and allows the user to interact with holographic input elements. We will explain these elements in detail in the following sections. Additionally, we support a few speech commands to allow for hands-free control where applicable. Our interface is built on the Microsoft Mixed Reality Toolkit [3] which provides templates for certain user interface building blocks.

3.2.1 Feature Panel

The Feature Panel is the user interface panel that first appears when the user opens the application. The Feature Panel contains the buttons named Hand Mesh, Point Cloud, Tutorial, Objects, and a pin simply indicated by an icon on the far right. Each of them can be toggled, and any additional interface elements are spawned and moved in relation to the initial Feature Panel. The panel can be moved freely to any location in the scene by grabbing its plate and moving it to the desired location. To simplify location adjustment, the user may toggle the pin button and allow the panel to automatically adjust its position in relation to the user's movement. This can be helpful for recording tutorials in larger spaces where returning to the panel might be too far.

3.2.2 Hand Mesh Visualization

The Hand Mesh button activates a live visualization of the user's hands, allowing the user to better understand hand tracking. The relevancy for experienced users can be neglected, as the recording of the hands is independent from the live visualation of them.

3.2.3 Point Cloud Visualization

The Point Cloud button toggles a live visualization of the tracked short distance point cloud sensor data gathered by

[1] https://github.com/lukas-walker/4d-holographic-tutorials

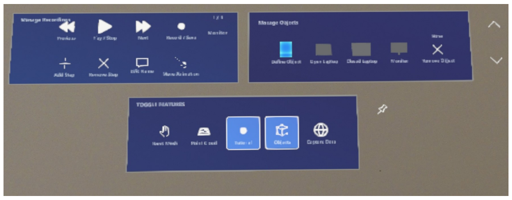

Figure 1. Shows an overview of the entire user interface available to the user.

the HoloLens. We integrated this functionality into our application through a plugin (see section 3.6) to access the HoloLens 2 Research Mode.

3.2.4 Recording Panel

The Tutorial button toggles the view of the Recording Panel (see Figure 1), which contains the following functionalities accessible through buttons:

- **Record.** Activating the *Record / Save* button starts the recording process with a countdown timer of three seconds to allow users to adjust their pose. While recording, both hands and all currently active objects are tracked. Deactivating the button, stops the recording and overwrites the current placeholder.
- **Play / Stop.** The *Play / Stop* button starts the playback of a chosen recording. Deactivating it stops the playback.
- **Previous.** The *Previous* button allows the user to choose an earlier recording made.
- **Next.** The *Next* button allows the user to choose a following recording made.
- **Add Step.** The *Add Step* button allows the user to add a placeholder for a new recording to be made.
- **Remove Step.** The *Remove Step* button deletes the chosen recording and removes it from the list of recordings.
- **Edit Name.** The *Edit Name* button displays a text prompt and allows the user to provide a description for the recording. This description will be displayed on the recording panel.
- **Move Animation.** The *Move Animation* button allows the user to relocate the animation using a handle. This is useful for aligning playback with the physical environment.

To the top right of the *Recording Panel*, the user sees two numbers, where the first indicates the number of the current recording and the second indicates the total number of recordings made.

3.2.5 Speech Commands

Speech Commands are a feature provided by the Mixed Reality Toolkit [3] which allows developers to select words that trigger actions when users speak them. We use speech commands to control recording and playback. The ability to control the recording with voice is aimed at experienced users since using the visual user interface can feel cumbersome in certain situations.

We support the following Speech Commands in our application:

- **Record.** Starts the recording process similar to the functionality of activating the *Record / Save* button, as introduced in section 3.2.4.
- **Save.** Mirrors the same functionality of deactivating the *Record / Save* button introduced before.
- **Play.** Starts playback similar to the functionality of activating the *Play / Stop* button, as introduced in section 3.2.4.
- **Stop.** Mirrors the same functionality of deactivating the *Play / Stop* button introduced before.
- **Previous, Next, Add Step, Remove Step.** See section 3.2.4.

3.2.6 Object Panel

The view of the Object Panel displays the following buttons:

- **Objects.** The first four buttons display object icons and allow the user to spawn object instances in the scene. Using the pagination buttons to the right of the panel, the user can conveniently scroll through a selection of different everyday CAD objects provided by the ObjectNet3D dataset [13]. The first of all object buttons is named *Define Object* and lets the user spawn a bounding box, which can be used to manually define a 3D subspace. Sensor data in the subspace can be extracted to create a new object based on the captured sensor data.
- **Remove Object.** The last button displayed lets the user remove spawned object instances from the scene. The button displays the name of a potential target object to be removed. To select an object, the user only needs to interact with an object in some arbitrary way (hovering over the target object will suffice).

3.3. Recording

The recording is either initiated using the functionality provided by the *Record / Save* button 3.2.4 or by using the *Record* speech command 3.2.5. The tracking of each entity throughout the recording process is split up between *Hand Recording* and *Object Recording*.

3.3.1 Hand Recording

To record all state parameters of our hand models, we use a duplicate pair of invisible instantiated hand models. They mirror the state of our actual hand models at specific points in time and provide information on rotation and position for every joint of both hands. As the joints are dependent on the wrist, we only need to track the position of the wrist. Tracking the rotations of all entities then allows us to reproduce the whole state. The used hand model limits joint movement to rotation and does not consider displacement. Hands of different sizes and shapes are directly handled by our used hand models by abstracting away those differences and by using the same size and shape for all users.

3.3.2 Object Recording

Sometimes, recordings of a user's hands are not enough to determine the action that is being performed. In this case, having virtual objects can help identify the interaction and can allow the user to orient the recording to their own environment. To record virtual objects, we use a similar approach as for the hand models. An invisible instance for each object is created which records the state parameters at specific points in time. However, in this case we need to record the position, rotation, and scale of all objects at all times and there is no underlying dependency between them. Additionally, we keep track of the type of object in question.

3.3.3 Serialization

All state parameters are tracked in relation to some point of reference (generally the origin, see section 3.5.2) and stored in a buffer during recording. Once the recording is stopped, the buffer data is written to a binary file. Additional metadata such as name, description, and the point of reference are stored to a separate file in XML format. It contains the list of recordings step-by-step and allows managing them.

3.4. Playback

The playback is controlled by the *Play / Stop* button 3.2.4 or the *Play / Stop* speech command 3.2.5. Each recorded entity (hand or object) is visually displayed in the scene accordingly, where the process handles *Hand Playback* and *Object Playback* separately.

3.4.1 Hand Playback

Here, we take the reverse approach from recording serialization. We read the state parameters of our hand models from the binary file and use them to reconstruct the scene. For each recorded state in time, we create interpolated states to enable a smooth animation. We use the same hand models as before and make them visible to the user. To animate the hands, we apply the recorded and interpolated state parameters for each frame. Figure 2 shows an example of the right and left hand model during playback.

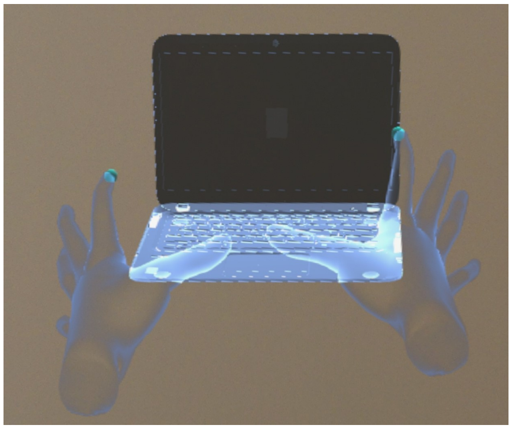

Figure 2. Animated hand & object during playback, showing a task from the viewpoint of a user wearing the HoloLens

3.4.2 Object Playback

The process for the objects is very similar when compared to the hand animation. We create instances for each object recorded and apply either the recorded and interpolated state parameters (rotation, position, scale) for each frame to again allow for a smooth animation of each moving object. Figure 2 also shows an example of an object during playback.

3.5. Location Invariance

The HoloLens 2 is very good at recognizing and remembering rooms. However, in most use cases of our application, the objective task is not tied to a physical location, but rather to whatever the user is interacting with during the tutorial. The HoloLens does not natively support object recognition, nor does it register small changes in a room, such as the displacement of an object of interest. We saw the need to address this issue to allow for location independent playback of various tutorial scenes (e.g. operating the same machine at different locations). We therefore want to adjust the animation point of reference to any room and any object during runtime.

3.5.1 QR Code Alignment

The approach is relatively simple and allows animations to be adjusted to the current scene. A QR code is placed at a specific point of interest related to a specific tutorial and scanned before recording and playing tutorials. In Figure 3, a button-pressing motion is aligned to a QR code.

The HoloLens scans QR Codes natively and continuously during runtime, independent of the running applica-

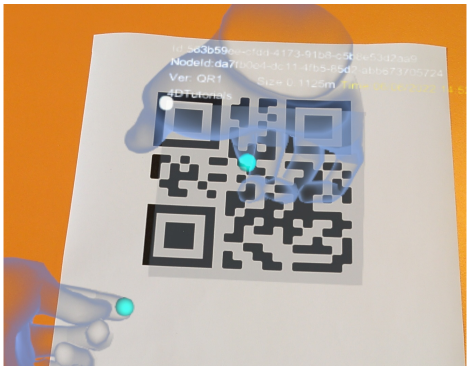
Id:563b59ee-cfdd-4173-91b8-c5b8e53d2aa9
NodeId:da7fb0e4-dc11-4fb5-85d2-abb673705724
Ver: QR1
Size:0.1125m
Time 06/06/2022 14:5
4DTutorials

Figure 3. Aligning an animation with a QR Code

tion. Using QR codes does not decrease performance, even if detecting image trackers in real time can be a challenging task. A package by Microsoft provides access to the QR Code API [2]. Our application tracks a relevant QR code constantly and updates the coordinate system accordingly whenever the code is moved. This allows for simple and intuitive use, as users immediately see the change should the QR code be moved. However, constantly updating the QR code introduces slight jittering as the tracking does not capture the position and rotation perfectly. Using different modes, we allow easy adaptation of the application for different use cases. The application can easily be configured to align to a code only once, to recognize different QR codes, or to accept all QR codes.

3.5.2 Manual Position Adjustment

The process of recording and editing high quality step-by-step instructions is time-consuming. Therefore, being able to adjust animations to deal with small changes in the environment is a big advantage. For example, if the same tutorial should be used for a new coffee machine model that is very similar to the old one but has one button moved slightly, users should not have to re-record the entire tutorial. To address this issue, we introduced an adjustment handle on a per-animation basis that can be toggled in the recording panel. Each animation has a second specific origin in the hierarchy between the global origin and the hand model itself. This can be moved and rotated before recording or before playback, and changes are stored to the metadata file.

3.6. Object Segmentation

A virtual object library will never be comprehensive for the diverse needs of the user. For instance, if the user is trying to demonstrate how to work an espresso machine, it is unlikely that we could provide the exact model of the machine in our library. Because of this, it is helpful to give the user the ability to define their own object. Rather than requiring the user to upload a 3D model, we use the depth sensor of the HoloLens to scan a physical object and construct a mesh from the depth sensor's point cloud. We allow the user to define one object that can be added to the scene, but the functionality could be extended to include as many objects as the user wishes to capture.

First, the system must understand which depth points are relevant for capture. We enable the user to place a floating, resizable bounding box in the scene. Once the bounding box has been placed, the user can select to capture all points inside the box. The system finds all points with coordinates inside the box and then constructs the convex hull of the points to generate a mesh. A virtual object clone of the physical object appears for the user. This new object can be moved, re-scaled to fit the scene and will also be part of recordings like other virtual objects.

In practice, we find that the lack of precision in the HoloLens depth sensor limits the ability to capture fine details. Thus, most objects lack a clearly defined shape. Additionally, the depth sensor will occasionally think it sees points that do not actually exist. In this case, these points can skew the convex hull calculation and cause the object mesh to have sharp protrusions that do not exist in real life.

Capturing an object from a single viewpoint does not provide complete object details. Therefore, we tried to have the user walk around the object and allow the algorithm to generate a complete view of the object's depth. However, in practice, this method did not work well and was abandoned on the final software version. As previously mentioned, there is significant noise in the depth sensor readings. Thus, noise from multiple angles around the object accumulates and causes the object to lose its scale and gain many protrusions that do not actually exist on the object. Additionally, capturing all angles causes the HoloLens to capture many points, increasing the complexity of the convex hull algorithm and causing it to run quite slowly and inefficiently on the power-efficient CPU. For these reasons, we decided to abandon this method and only allow single-viewpoint object capture.

For best results, the object must be placed on a flat surface with no nearby objects protruding into the bounding box of the object. Otherwise, these objects may interfere with the mesh generation.

4. Results

To evaluate our application, we conducted a user study to qualitatively measure the user experience. Our study involved 6 participants performing 2 experiments. The participants start by getting accustomed to the AR environment, the usage of the HoloLens, and our application in general. They could test and try out any feature available to them in the environment provided by the application. In the second experiment, their goal was to set up a real desktop workplace in augmented reality using four objects, namely a mouse, a keyboard, a laptop, and a monitor which should be placed in a way one would expect to arrive at one's workplace. This was meant to be done in multiple steps where the participants use various features to accomplish the task such as recording, playback, spawning of objects, naming of different steps, deleting and adding steps, etc. Once the tutorial was created, we also asked them to relocate the tutorial to a new location to test the location invariance provided based on the QR code with slight manual adjustment. After completing the experiments, we gathered the participants' feedback on how our application performed. We asked questions regarding the overall ease of use, the physical fatigue experienced, the visibility of the playback, the accuracy of the recordings, the ease of relocation, and how easy it was to manage the recorded steps of the tutorial. The average ease of use was evaluated at a rating of 4.5 / 5. The accuracy (4.66 / 5) and visibility (4.33 / 5) of recordings were appropriate, and managing the different steps of the tutorial was very convenient (4.25 / 5). The location invariance was easily handled using a QR code (5 / 5) and slight changes in the scene, though handled using manual position adjustment, posed quite some problems (3.25 / 5). Nevertheless, we conclude that our application enables users to conveniently and with little effort create tutorials in augmented reality.

5. Discussion and Future Work

For future work, we advise to especially extend on our work regarding object segmentation. A scenario most helpful for users would be to interact with real objects and for our recording process to automatically extract a 3D model to be animated during playback. A first step would be to let the user define new objects with higher accuracy and in a more user-friendly way. During our work, we noticed the rather limited computational resources provided by the HoloLens, which would encourage future work to use external resources such as cloud computing or local network access to more resourceful machines. Some initial programming work to establish a TCP socket to a remote server has already been started. More improvement in the area of user-friendliness can be achieved through more annotations on objects interacted with or during playback for various parts to indicate focus for the user watching the tutorial. Those annotations may be of visual or audible nature and should make the instructions clearer. Furthermore, including a documentation for various features available directly into the application would certainly improve usability.

Object segmentation provides the most opportunities for future work. Future work could consider adding other shapes for the point cloud capture bounding area, such as cylinders or spheres, that might allow for a closer fit on the object. The generated virtual object is given a basic gray mesh, but future work could investigate using camera data to reconstruct a realistic texture. Our method of generating a convex hull has limitations for creating a mesh from a point cloud. Visual details can be lost in a convex hull, since it merely takes the smallest shape that contains all the points. For instance, a coffee cup appears filled in. Future work should investigate a marching cubes approach to generate a mesh that most closely fits the points. To fuse data capture from multiple angles, future work should investigate Iterative Closest Point (ICP) algorithms. Besides object clarity and detail, the largest limitation of our object segmentation process is that it requires the object to be physically distanced from other objects. Future work should investigate how to distinguish between multiple, adjacent objects.

6. Conclusion

We present a user-friendly application for the HoloLens to create 4D holographic tutorials to give experts in various fields an option to explain details in their workflow otherwise inaccessible. We allow easily manageable step-by-step recordings with our user interface and a handful of ready to use objects. Tutorials can be reused at multiple similar locations thanks to the location invariance provided. We conducted a user study to test the application and found that we provide an easy and user-friendly way to create such tutorials. We released the source-code as open-source software and hope to have laid a foundation for future work.

7. Work Distribution

- **Andrin Bertschi.** Point cloud plugin for Unity based on Microsoft Research mode, point cloud management
- **Benjamin Hardin.** Recording panel, countdown for recordings, recording naming, object segmentation
- **Lukas Walker.** Hand animation, part of speech commands, file handling, animation list, location invariance
- **Nicolas Wicki.** Object panel, object animation, user interface structure, part of speech commands, improvements to recording panel

References

[1] Intermediate language to cpp, unity backend. `https://docs.unity3d.com/2018.4/Documentation/Manual/IL2CPP.html`, published: 05/12/2022.

[2] Qr code tracking overview. `https://docs.microsoft.com/de-de/windows/mixed-reality/develop/advanced-concepts/qr-code-tracking-overview`, published: 05/12/2022.

[3] What is the mixed reality toolkit? `https://docs.microsoft.com/de-de/windows/`

mixed-reality/mrtk-unity/mrtk2/?view=mrtkunity-2022-05, published: 06/02/2022.

[4] Xr sdk for windows mr, unity. https://docs.unity3d.com/Packages/com.unity.xr.windowsmr@5.4/manual/index.html, published: 05/12/2022.

[5] BMW Group Corporate Communications. Absolutely real: Virtual and augmented reality open new avenues in the bmw group production system. 2019.

[6] Alexis Goodrich. Exploring the intel manufacturing environment through mixed reality. https://community.dynamics.com/365/b/365teamblog/posts/exploring-the-intel-manufacturing-environment-through-mixed-reality, published: 18/11/2021.

[7] Alyson Langley, Glyn Lawson, Setia Hermawati, Mirabelle D'Cruz, J. Apold, F. Arlt, and K. Mura. Establishing the usability of a virtual training system for assembly operations within the automotive industry: Usability of a virtual training system for assembly operations. *Human Factors and Ergonomics in Manufacturing and Service Industries*, 26, 08 2016.

[8] Regina Leung, Andras Lasso, Matthew Holden, Gabor Fichtinger, and Boris Zevin. Exploration using holographic hands as a modality for skills training in medicine. page 28, 03 2018.

[9] Pavithran Maniam, Philipp Schnell, Lilly Dan, Rony Portelli, Caroline Erolin, Rodney Mountain, and Tracey Wilkinson. Exploration of temporal bone anatomy using mixed reality (hololens): development of a mixed reality anatomy teaching resource prototype. *Journal of visual communication in medicine*, 43(1):17–26, 2020.

[10] Hugo Rositi, Owen Kevin Appadoo, Daniel Mestre, Sylvie Valarier, Marie-Claire Ombret, Émilie Gadea-Deschamps, Christine Barret-Grimault, and Christophe Lohou. Presentation of a mixed reality software with a hololens headset for a nutrition workshop. *Multimedia tools and applications*, 80(2):1945–1967, 2020.

[11] Unity Technologies. Unity engine 2019.4.39f1, 2019.

[12] Dorin Ungureanu, Federica Bogo, Silvano Galliani, Pooja Sama, Xin Duan, Casey Meekhof, Jan Stühmer, Thomas J Cashman, Bugra Tekin, Johannes L Schönberger, et al. Hololens 2 research mode as a tool for computer vision research. *arXiv preprint arXiv:2008.11239*, 2020.

[13] Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Choy, Hao Su, Roozbeh Mottaghi, Leonidas Guibas, and Silvio Savarese. Objectnet3d: A large scale database for 3d object recognition. In *European Conference Computer Vision (ECCV)*, 2016.

[14] Hui Xue, Puneet Sharma, and Fridolin Wild. User satisfaction in augmented reality-based training using microsoft hololens. *Computers (Basel)*, 8(1):9–, 2019.