3D Drawing Evaluation Using Hololens 1

Martin Bernardi, Nour M. Abdelhamed

***Abstract*—The availability of Augmented reality technology enables the users to perceive the surroundings in a more informative way as the environment is augmented with illustrative objects [1]. Also, it aims to engage them with those virtual objects in a sense that makes an educational material easy to learn and solidified in their brains. In a series of two research courses, 2D/3D drawing applications are developed to make use of augmented reality features. For the first course [2], a 2D calligraphy learning application was implemented using a phone camera and means of object registration in marker and markerless settings. Conclusions were drawn related to the efficiency of imposing calligraphy character on the screen as well as to indicate the learner experience. For the second and current course, Hololens device as an HMD (i.e. Head Mounted Device) is used to implement a 3D drawing learning application, targeting lines, circles and spirals drawings. In-depth analysis on the HMD's sensors and technologies are recorded. Furthermore, technologies used to deliver the application such as input user tracking and drawing comparison are discussed in addition to evaluation of their limitations as well. Finally, scores of the user results following the drawing templates are attached indicating the similarity of how they follow the templates. A line comparison criteria is proposed for indicating the scores. In conclusion, direct interaction using Hololens whether by the hand gesture or marker tracking is impractical due to processing limitations of the device as well as limitations with respect to depth perception.**

I. Introduction

The main topic of this work concerns the use of Augmented Reality (AR). The definition of AR, together with Virtual Reality (VR) and Mixed Reality (MR), is not straightforward and there are conflicting views in the topic [3]. A common definition for this term involves the Milgram & Kishino's Reality–Virtuality Continuum [4]. In this definition, MR is a term which involves the merging between real and virtual worlds. It includes AR and VR to broadly differentiate if the experience is prominently real or virtual.

When talking about AR, the definition is still broad and has a strong relationship with the hardware used. Augmented reality research community, especially research related to mobile augmented reality, puts an emphasis on defining generic augmented reality technology in which re-usability and consensus over the used technology is achieved [5] [6]. One of the objectives of this work is the implementation of an AR experience using the Microsoft HoloLens as a device, so from now on we address AR, provided by a Head Mounted Device (HMD) as opposed to Hand Held Devices (HHD) or other methods [7].

In previous work [2], the project focus was done on reviewing the state of the art technologies and applications featuring calligraphy learning. A prototype of a calligraphy learning application was implemented. The main problem observed for this application is the stability of imposing the template character in the augmented environment using the HHD (.i.e the mobile phone). Irregularities, such as inaccurate positioning of the template due to user motion or insufficient marker tracking were identified as the main focus required to enhance the user experience. Three forms of calligraphy learning application were provided to indicate the quality augmenting calligraphy characters, including marker and markerless solutions. Marker based registration from the Vuforia library resulted to be the best performing solution for this application, while the other options suffered from greater instability of projection along the drawing period as shown in Figure 1. Even though the best registration feature was accomplished by Vuforia library, inconsistencies of millimeters between the expected projected and the actual one were visible when applying manual criteria to indicate the difference between both versions as indicated in Figure 2.

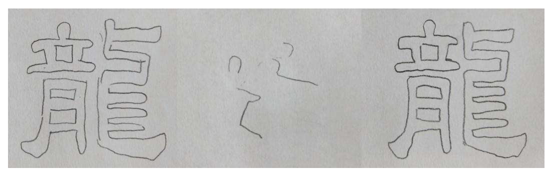

Fig. 1. Learning results of using three different augmented reality applications for calligraphy learning, from left to right: marker based using AR foundation marker, markerless based using AR foundation, marker based using Vuforia Library [2].

Fig. 2. Difference between actual and expected projection using the image tracking feature of Vuforia library [2].

Research outcomes of the previous project [2] suggest integrating a user input tracking algorithm to keep track of user's drawing and implementing an evaluation technique to assess virtual registration as well as providing the users feedback as numerical score for how good they follow the template. Those points are mainly addressed in this course.

Hololens 1, as a head mounted device, is considered for development in this work allowing the learner to be re-

martin-gustavo.bernardi@etu.u-bordeaux.fr
nour-mohamed.abdelhamed@etu.u-bordeaux.fr

lieved from holding the phone while following the character. Hololens leverage a great range of immersive features adapted by the research community in many disciplines such as in surgical, industrial, architecture and other fields of engineering [8]. Its set of sensors and built-in algorithms for tracking and recognition make the development of augmented reality powerful for wide range of applications.

To make use of the registration and tracking capabilities of the Hololens, the project is extended from calligraphy learning (i.e. learning 2D drawing) into 3D drawing learning. The main aspects of this work are registration and object placement, input tracking, and drawing evaluation. As registration and object placement, the markerless approach from the Hololens is utilized. User input tracking is done based on hand and marker tracking. In addition, means of drawing matching is investigated to provide a good quantitative metric for user feedback as well as evaluating the Hololens capabilities of augmenting objects in the real world.

The code developed for this project can be found at GitHub/martinber/holodemo.

This report is organized in the following structure:

- First, literature is reviewed about aspects of AR usage in the area of learning and teaching.
- In the next section, the work done in the previous semester is reviewed, indicating findings and the proposed approaches for this research course.
- In the Technical Framework section, details of the Hololens and related devices are listed. This is important since the limitations shape the decisions taken during the implementation.
- Next, the details concerning the implementation and the features of the software are explained and discussed.
- Finally, the user experience is evaluated and discussed to reach conclusions.

II. Literature review

A. AR in teaching and learning

The information technology research community shows great interest in mobile AR development for leveraging educational context applications, allowing to enhance learning experiences and to make material easily accessible to learners [9]. One way to categorize the AR applications that are available to educators is as *location-aware* and *vision-based* [10].

Location-aware AR prompts digital media to learners as they move in the physical environment with location-enabled devices. Graphical media, presented as text, graphics, audio, video and 3D models, is augmented to the physical world with academic information relevant to the location of the user. This technology can be found in the area of industrial equipment and tool training where 2D images of certain objects in the industry are augmented with their 3D forms using a mobile augmented reality technology (.i.e mobile device with camera and GPS sensor) [11]. Also, specific applications related to tourism exist in the literature to guide tourists in specific places where historical information relevant to specific places are projected to guide the user [12].

In contrast, *vision-based* AR is associated by presentation of digital media to learners after pointing their camera to a target object (e.g., ARToolkit, ArUco and Vumark markers). Similar context in tourism can reflect this technology as well: a camera pointing to a static marker, such as an ArUco marker, would immerse the relevant information associated with this marker.

A real life example combining both AR technologies is available in the *Tuscany+* AR application for IOS [12]. Relevant information about located historical landmarks in Tuscany city guides users to sources where they can inspect relevant information about the area and other useful features of navigation.

These previous applications show the informative contribution of AR in aspects of guidance and training. It enables the users to discover the unknown in critical situations or educate them about the surroundings in a unique and easy way. Additionally, the engagement of the students with the augmented world enhance the learning outcomes as the knowledge becomes closely relatable [1]. Therefore, the recent research of augmented reality in the context of learning focus on the engagement part [13]. This trend brings development challenges to leverage good AR experience as well as enhancing the quality of education. Example of such applications includes [14] [15] for 2D drawing learning and [16] for training maintenance and assembly skills in industrial context.

B. Interactive learning using AR

As opposed to the previous AR applications that are focused in information learning, there is interest in AR for interactive learning which involves interaction of the user with the virtual world via movements and gestures. In this paper [17], an AR framework for 3D drawing is developed to enhance school learners experience learning geometrical shapes as well as manipulating their 3D viewing using hand gestures movements.

The quality of education leveraged by the AR depends on the availability of immersive devices with relevant sensors for surface detection and virtual object registration as well as to have a limitation-free design associated with teaching and learning. In this project, we are interested in the application of calligraphy learning as it is related to the topic of the first and the second research course.

In the area of calligraphy learning, usage of smartphone devices is sufficient in terms of sensor availability to produce immersive content in which the learner can observe an augmented character to a screen and follow it on a sheet of paper with proper view of the imposed character depending on the location provided by the sensors. The quality of character positioning as well as maintaining its position during the learning process is also critical [2]. Therefore, means of tracking and positioning corrections are in charge of avoiding any inconsistencies. For learning limitations, learners should not encounter any disturbance following the augmented character. For teaching limitations, the application should provide unsupervised feedback, without presence of an evaluator [18], to the learner as a means of graphical alarms or matching scores; that's why means of distance matching and content imposing such as text or color are loosing those challenges.

Additionally, holding the mobile phone with one hand and following the drawing on the paper with the other is a burden to the learner who might suffer from positioning inconsistency of the augmented character [2]. Therefore, a head mounted device would be a better solution for the learner.

III. Proposed approach

As aforementioned, this project is extension of calligraphy learning to a 3D case. Where the template instead of being a calligraphy character, consists of a 3D template. In this case, geometrical shapes such as a spiral, a line and a circle are the drawing targets to be followed by the hand of the user. Means of object registration, user input tracking and drawing comparison are core pillars of the application.

A. Object Registration and Recognition

Object registration and recognition is meant by the technique used to determine where to place virtual content to the real environment [19]. In previous work [2], object registration from a camera is delivered by image markers placed in the real world or by markerless mapping of the environment. This was implemented by the ARCore, ARFoundation and Vuforia libraries for Android. The result of marker based Vuforia has the most robust means of registration and recognition compared to the rest options producing the best drawing results as shown in Figure 1. In this work based on the Microsoft Hololens, registration is applied only by markerless means but with better results thanks to the greater sensing capabilities of the device.

B. User input tracking

In the context of drawing learning, user tracking includes user movement tracking in the real world to imitate the process of drawing virtually. This feature was not addressed in the previous project [2], but in this case user input is tracked using two techniques, hand gesture tracking and marker tracking. For the first tracking technique, the hand with a grab gesture is utilized for following the drawing template, where the center of the detected hand marks the place of the user's drawing. Another evaluated solution is a pen with a Vuforia marker as a pointer to create the drawing. Both input methods are shown in Figure 3.

C. Drawing comparison

The trajectory of template curve is generated from discrete vertices, and the user drawing is obtained by tracking methods as described above. Then, as a final part of the application, curve comparison is applied to evaluate the users' drawing with respect to the template (see Section V-A).

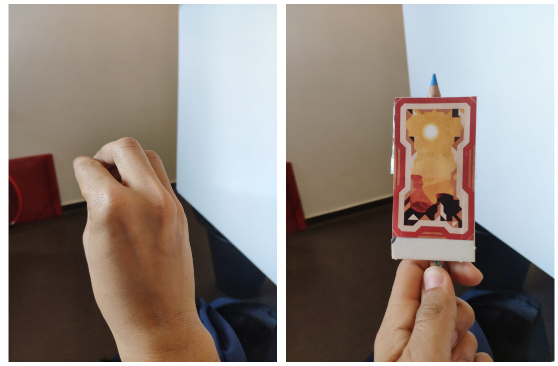

Fig. 3. User input methods considered for drawing

IV. Technical framework

A. Hardware

The experiences provided to the user in AR depend greatly on the hardware used since there are many technical challenges yet to be solved. When talking about HMD, main differences in hardware include the display, sensing capabilities and processing power. In this section, we focus on the Microsoft Hololens which will be used in the implementation.

The Microsoft Hololens is a completely enclosed passively-cooled HMD device that contains mainly a see-through display, a wide range of sensors (see Figure 4), processing units and battery. It runs the Windows 10 Operating System over its 32-bit x86 intel CPU/GPU, with the aid of a custom-designed Holographic Processing Unit (HPU) dedicated to spatial mapping and understanding of gestures in real time [20] [21]. WiFi and Bluetooth connections are available, and the battery of the unit allows for 2-3 hours of totally independent wire-free use of the device.

Fig. 4. Sensor bar present in the Hololens 1. [21].

To detect posture and relative movements of the device, four environment understanding cameras with the Iterative Closest Point algorithm are used, together with an Inertial Measurement Unit (IMU) with a gyroscope, accelerometer and magnetometer [22] [23]. This estimation produces better performance against flat surfaces in bright lighting conditions, while the biggest errors are produced by high movements speeds or by quick rotations against the X axis [22].

The spatial mapping of the world is done with the depth camera, the environmental understanding cameras, and the KinectFusion algorithm. This spatial mapping allows the retrieval of meshes representing the environment, serving mainly as anchors for the placement of virtual objects in the real world. The best results are achieved at distances of 1.5-2.5m, which may be beneficial for high-accuracy tasks such as mapping mechanical components [22] [24].

Regarding the display, it is see-through transparent display. Planar waveguides that make use of Total Internal Reflection are able to guide projected images from two small Liquid Crystal on Silicon Displays (LCoD) into the user's eyes, although the details of the working principle are not completely available [25] [26] [23]. To ensure that the displayed objects are stable and rendered at an appropriate apparent distance, the Interpupillary Distance (IPD) must be calibrated [27].

The biggest limitations of the displays that breaks the immersion and shapes how the applications are designed, are the relatively small field of view and the working principle that only allows to project light but not to produce opaque images. A small field of view produces not only cropping of big virtual objects but also produces that small objects become only visible when pointing the head directly towards them, requiring the user to move the head instead of the eyes to look around. As mentioned, the display guides project light into the user's eyes, but cannot prevent the passage of light from the world, this means that it is impossible to render a black object or a virtual object that completely occludes a real one.

Another problem is the Vergence-accommodation conflict that can create eye fatigue and focusing problems. When looking at far and close objects in the real world, eyes have to compensate for vergence angle and for focusing distance (accommodation), these two mechanisms are involuntarily linked via the accommodation-convergence reflex. When using an HMD, the depth is simulated by displaying different images for each eye, but the brain receives mismatching cues because the focusing distance is still the same [28] [29]. The current workaround used in AR applications is to simply design applications that do not show virtual objects too close to the user.

In the case of the Microsoft Hololens 2, the field of view of the screen is bigger but the pixel density is reduced [30]. The biggest improvement is the availability of better input methods, where hand and finger detection is used to provide direct manipulation of virtual objects [31]. Another similar device is the Magic Leap One [32] but there is no significant literature about its capabilities.

B. Hololens development

The development in the Microsoft Hololens has certain particularities because of the uniqueness of the device. As AR is an area in active research, the development tools recommended vary considerably over time so special care must be taken to utilize compatible tools and frameworks, especially since the HoloLens 1 currently is a deprecated device.

In a low level, developed applications leverage APIs part of the Windows 10 SDK that implements all the processing needed to allow the device to understand the surroundings. This is normally accessed via Extended Reality (XR) libraries that may provide cross platform interfaces to different devices.

When developing for the HoloLens [33] it is possible to create a native application in C/C++ directly with the OpenXR API, or use an engine. One of the supported engines is Unity that allows to create applications both visually and with C# scripting, but also the Unreal Engine 4 can be used to develop in C++.

Legacy XR, Windows XR or OpenXR can be used in the Hololens depending on the application requirements, and Legacy XR was chosen since it is appropriate for Unity 2019 [34]. At a higher level, the HoloToolkit is a set of high level tools developed for Hololens 1 that later evolved to become the Mixed Reality Toolkit (MRTK), providing cross platform input and GUI elements for various AR devices [35].

When working with Unity, the program is organized around GameObjects. The GameObjects represent an entity, which commonly has certain behaviours coded in scripts, a defined position and rotation in space (represented in its Transform component) and a visual representation controlled by the Renderer component. The coordinate system used is fixed to the real world, units represent meters and the origin of coordinates located where the device was when the application started. Similarly, the Z axis points forward and the X axis to the right of the initial orientation, but in the other hand the Y axis always points up relative to the real world.

The device is represented in Unity as the Camera GameObject, which will move automatically in the world-fixed coordinate system according to the movements detected by the sensors on the device.

The application which is the focus of this work is written in C# taking advantage of Unity 2019 to build a Visual Studio project, which after is compiled for the Hololens device in particular. It is possible to connect the Microsoft HoloLens via USB but it is recommended to deploy applications via WiFi. In this case, both the development PC and the HoloLens must be connected in the same network and the IP address of the target device must be indicated in the deployment settings of the Visual Studio IDE.

C. Hand tracking

One of the built-in capabilities of the SDK is hand tracking and gesture detection, where the position of the hand and four gestures can be recognised in a specific area in front of the user [36].

It is possible to retrieve only the position of the center of a hand; the orientation is unknown, there is no differentiation between left and right hand, and the position of the fingers is not available. One of the limitations is that the hand has to be held in a certain shape that resembles one of the detectable Hololens default gestures [36]. Finally, the latency is very high and limits the usage to mostly indirect input instead of direct manipulation of objects.

The ideal tracking setting for 3D drawing, would be the tracking of fingers which is only possible on Hololens 2. A workaround for Hololens 1 would be running custom computer vision algorithms that make use of raw sensor data streams, the access and usage of them are described in the next section.

D. Marker tracking

The tracking capabilities of the Hololens 1 are very limited since by itself it does not include any form of object or marker tracking apart from the hand. Alternatives include the processing of raw sensor streams using Vuforia or libraries based on OpenCV. In both cases, the target recognition and pose estimation is generally done solely from the RGB camera.

If using libraries such as OpenCV and ARToolkit, the main challenge is to include these C/C++ libraries to be compiled for the Hololens processor, and to link them to the C# code developed in the Unity engine. There is little documentation on the procedure, but here are samples available online which vary greatly in the libraries and versions utilized [37] [38].

Another alternative to running OpenCV in the device, is to wirelessly stream in real time the camera data to a computer and receive the results [37]. This has the disadvantage that the latency is increased and that a continuous connection to a computer is needed.

Fig. 5. On the left is a vumark, On the right is an image target of Vuforia markers

After unsuccessful attempts to run OpenCV in the device, it was decided to use the Vuforia library which is available for a big variety of devices including the Hololens. This library provides a stand alone solution for tracking, supporting a number of marker options as image targets, Vumarks, model targets and others. The suitable marker options for this project are the Vumark and the image target.

Both markers are similar in terms of how they are tracked by Vuforia. The difference is in the visual representation of each. Vumarks are closer to bar codes. They has enough features of interest that make their detection and tracking robust. Image targets are ordinary images which make them less unique to track compared to Vumarks.

E. Spatial Mapping

The meshes related to the shape of the world around the device can be accessed via the usage of the *SpatialMapping-Manager* singleton object from the HoloToolkit library [27]. To avoid the usage of too much computing power, it is possible to configure the precision of the mapping by setting the amount of triangles per cubic meter to generate. The obtained data structure consists of a list of several *MeshFilters*, where each one represents a portion of the room and contains a *Mesh* with the vertices in local coordinates and a *Transform* that allows to position the *Mesh* in global coordinates.

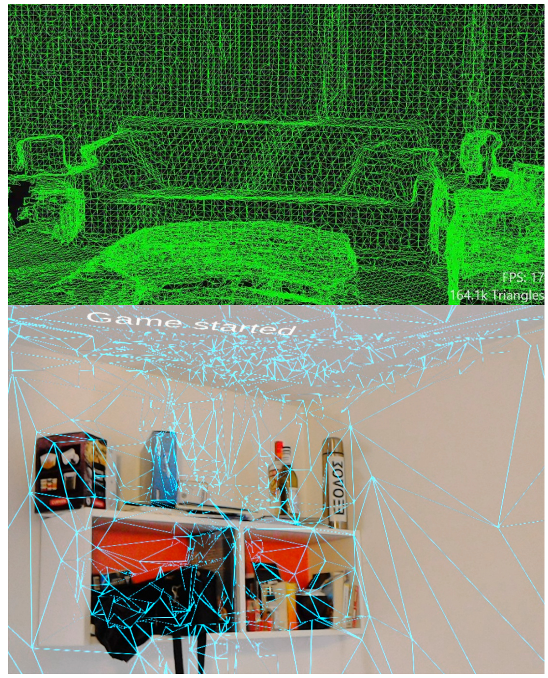

Fig. 6. Example of a Spatial Mapping mesh obtained with Hololens (from [39]) and a screenshot of our test application. Meshes with lower triangle count can be used for performance reasons.

A simple Unity project was developed to demonstrate the working principle of this feature. It is possible to show the meshes over imposed in the real world but also transfer the data to the computer for later use. In Figure 7, vertex data from the mapping of a room is plotted, and the partitioning of the room in different *MeshFilters* can be observed. With this data, it is possible to develop applications with objects that interact with the environment and avoid collisions with the real world. Furthermore, these meshes can be analyzed to interpret the localization of flat walls or surfaces where planar virtual objects can be placed.

The Spatial Mapping feature in the Hololens has the disadvantage that it is meant for static objects and cannot react quickly to object displacement or moving people.

V. Implementation

The application, named Holodemo, is aimed for 3D drawing learning. A user wearing the Hololens will have three kind of targets to follow by drawing: a straight line, a circle and a spiral. Speech commands are used to choose one of the drawing options as well as to mark the start and stop points of the drawing. Hand and Vuforia trackers are available as options of painting, allowing to assess the tracking as is done later in the evaluation section. Finally, a score is projected as a graphical text object above the users' head to indicate how accurate they followed the drawing template.

Regarding the general development process, the Unity Engine with the Windows 10 SDK was installed in Windows 10 machines. Git was also set up to allow version control and

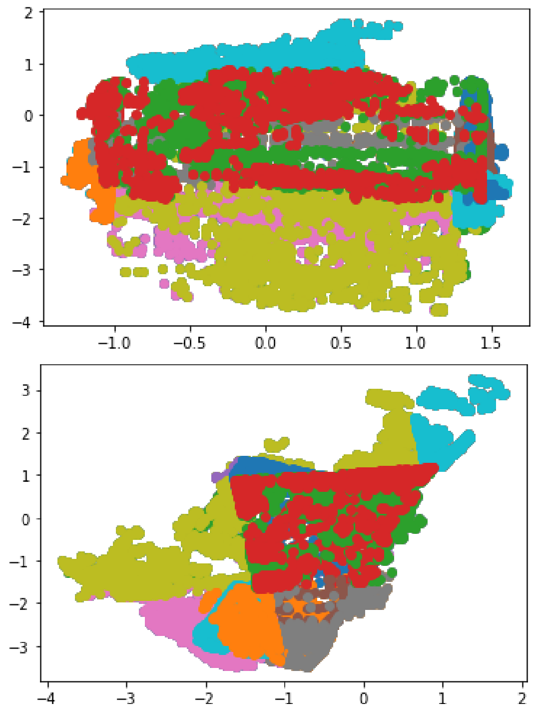

Fig. 7. Vertices exported from a mapping of a room made with the Microsoft Hololens, plotted with Matplotlib as a side view and a top-down view respectively. Each MeshFilter is represented in a different color and the units are in meters.

collaborative development. To upload the compiled software to the device, a WiFi connection is used.

A problem encountered is that the network available in the campus is protected via WPA2 Enterprise authentication method which is unsupported in the device, as opposed to the more common WPA2 Personal authentication. The workaround involved the usage of the development PC network as a WiFi hotspot to which the HoloLens is connected. As the HoloLens was unable to obtain an IP address via DHCP, a static address was configured. The setting of a static IP address in the device requires the access to the advanced configuration portal accessible via a USB connection.

Other details to consider when developing in this device, is that the applications have to request the corresponding permissions to access the hardware. Mainly, it is required to request permission to use the microphone, camera and to enable spatial mapping. This is done by adding the corresponding directives in the Package.appxmanifest file generated by the Unity build.

A. Line drawing and comparison

For line drawing, two Unity LineRenderer components were used. The first LineRenderer, called TargetLine, draws a line generated at runtime when the program starts. The second one, named UserLine is the line drawn by the user and is generated at runtime from the position of the hand or the Vuforia marker.

The target line is generated from a parametric equation to define a spiral, a circle or a straight segment. The case of the spiral is shown in Equation 1, where L is the number of loops, r_1 and r_2 the radius and t is incremented between 0 and 1.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} (r_1(1-t) + r_2 t)\sin(2\pi t L) \\ t \\ (r_1(1-t) + r_2 t)\cos(2\pi t L) \end{pmatrix} \tag{1}$$

In the case of the user line, each vertex is simply obtained as the position of the hand or the marker. To avoid oversampling and to filter small movements, new vertices are added only if there is a certain distance respect to the previous point. Figure 8 shows the user and target lines as they look in the software.

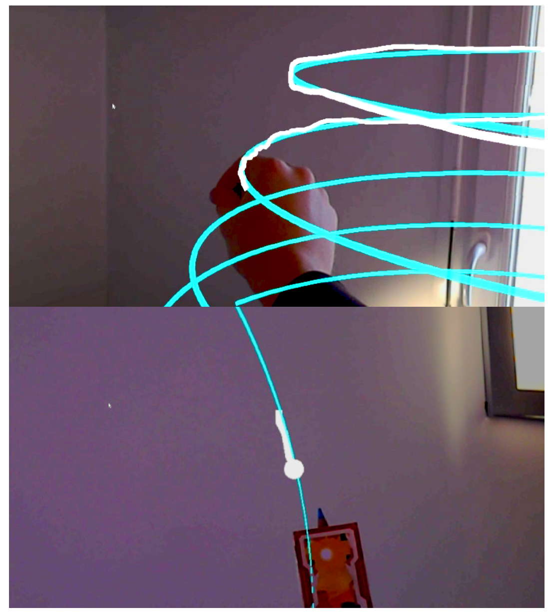

Fig. 8. Screenshots from the software. The target line is drawn in cyan while the user line is in white. In the first case a spiral is being followed with the hand, in the second example a circle is followed with the pointer (the mismatch between the pointer and the drawing is not present in-device, it is due to video recording limitations).

After the user finishes drawing the line, a comparison between the TargetLine and UserLine is done to obtain a similarity metric. The chosen metric is the Directed Average Hausdorff distance [40], from the user to the target line. This distance can be written mathematically as shown in Equation 2, where U and T are the user and target lines, $d(.,.)$ is the euclidean distance and N_U is the number of points in the user line:

$$\frac{1}{N_U} \sum_{u_i \in U} \min_{t_j \in T} d\left(u_i, t_j\right) \tag{2}$$

The objective is to find the shortest distance from each vertex in the user line to the target curve, but when the target line has enough sample rate, it is possible to obtain an acceptable approximation. Figure 9 represents the distances involved in the calculation of a Directed Average Hausdorff distance.

This metric provides good results in our use case given some assumptions, mainly related to the sampling of both lines. The spatial sampling frequency of the user line has to be considerably lower than the target line but high enough to capture the general user movement. Also, the assumption is taken into consideration that users follow the track of

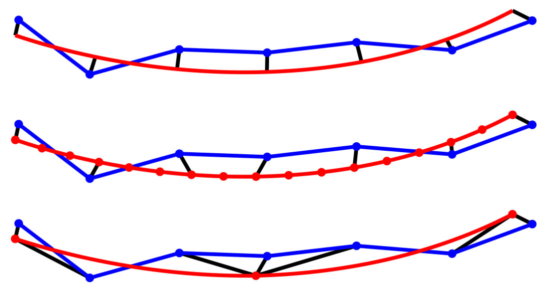

Fig. 9. The user line is represented in blue while the target line on red. The first case shows the real distances (in black) from the user line vertices to the target line. The second and third cases shows the distances involved in our approximation, calculated as the Directed Average Hausdorff distance. In the third case incorrect results are obtained due to bad sampling.

the drawing template from the start to the end without any skipping of the template segments.

Other possible metrics to use include the Frechet distance, Dynamic Time Warping or similar measures [41] [42] [43].

B. Speech recognition

Another input method used is the speech recognition from the device microphone via the *UnityEngine.Windows.Speech* API. After the instantiation of a *KeywordRecognizer* object provided a list of keywords to detect, a callback to *OnPhraseRecognized* can be added. When the callback is executed, the keyword detected from voice input is obtained, together with information such as the duration and detection confidence [44]. For the best performance, it is recommended to use concise voice commands, with multiple syllables and sufficiently different one from the other [45].

In the developed program, several voice commands were created:

start line
: Starts creating the user line from the position of the hand.

stop line
: Stops the line drawing and shows the similarity score between both lines.

next target
: Changes the target line trajectory between straight segments, spiral or circles of different orientations.

move target
: Allows to change the position of the target line.

VI. Evaluation

There are several points to consider for the evaluation in this course. Holodemo application is evaluated regarding its features such as template drawing registration and user input tracking. Moreover, limitations of the Hololens are recorded to indicate the challenges related to 3D drawing learning. This covers the tracking precision in relation to Hololens sensors in addition to depth estimation precision in relation to the concept of parallax error.

The same user, which is familiarized with the hardware, evaluated the different scenarios in order to extract conclusions not only about the drawing experience but also about the technical capabilities of the device and the libraries used.

A. Quantitative results

Four trials per template were made by the same user, and the results correspond to the Average Directed Hausdorff distance between the drawing and the template. When drawing the spiral with the hand (see Figure 10), the average distance ranged between 13.9mm and 20.3mm. A harder case was the vertical circle, with metrics between 16.1mm and 23.2mm. In the horizontal circle we obtained 11.7-14.7mm and finally, the easiest scenario is the horizontal line with scores between 4.3mm and 9.4mm.

For the pointer with the image marker, only results for the horizontal line are considered as relevant, with distances between 3.7mm and 4.8mm. Good precision is achieved thanks to the virtual small white ball that is rendered to help the user understand exactly the detected position. Although accurate results were obtained with this pointer, the tracking is easily lost which rendered this method unusable for the other longer templates.

Fig. 10. Screenshot of the result from the drawing of a spiral.

B. Object registration

One of the challenges that was recorded from TRDP I is object registration. This was the bottleneck of the Calligraphy learning application since calligraphy character registration was done using a marker based solution that suffers from small movement changes due to user movements. Markerless registration was impractical in the context of calligraphy learning as the character is not maintained well during the learning process.

Instead of the phone camera, Hololens utilizes its four cameras to leverage means of Spatial mapping and apply markerless tracking and object registration. It was observed that 3D virtual drawing targets, such as line segments, circle and spiral, are imposed on the real world accurately without the aforementioned problems and the projection is persistent during the learning process.

C. Tracking limitations

Available tracking options are hand gesture and Vuforia Vumarks. Regarding the hand tracking, it was observed that the drawing template as well as user's hand should be in the field of view of the Hololens for position estimation to be easy and consistent during the drawing process. Furthermore, the user should maintain the hand gesture during the drawing period for position estimation to be present. The hand tracking does not work correctly when the hand is close to a wall, suggesting that the algorithm depends considerably on the depth camera.

Hololens detects the position of the hand with great precision; however, a big latency is observed. This is related to the processing power of the hololens as gesture detection is not meant for precise tracking of the hand position. Also, drawing using the hand gesture is not accurate in a sense that the line drawing marked by the center of the detected hand gesture not the tip of the hand or even the fingers. A possible workaround would be showing a dot mark on the hand where the drawing is tracked by the hololens.

Tracking using markers comes with different challenges compared the hand tracking. Marker should be clearly present in the Hololens field of view for consistent tracking so occlusion or side views of the marker would not guarantee proper tracking. Moreover, Hololens' access to the camera also presents important latency that produces very noticeable lag.

The marker tracking is easily lost rendering this input method useless for most applications. Especially in low light environments, rapid movement of the marker while drawing results in motion blur of the frames and marker detection is not possible as shown in figure 11.

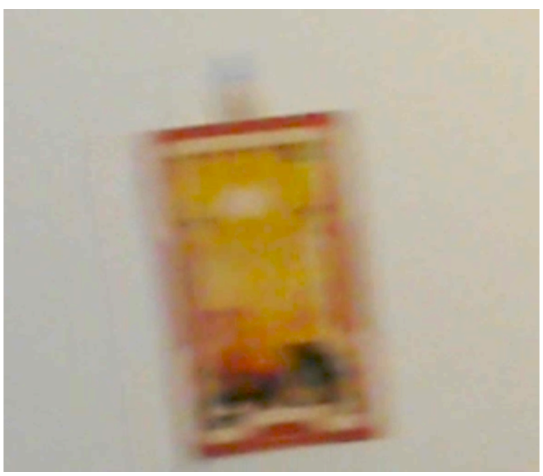

Fig. 11. Screenshots from the software showing the motion blur and noise captured by the camera in low light conditions.

D. Depth perception limitation

A big limitation discovered during the evaluation is related to the displays. In the scenario of a user following a template in the air, it is very important that a the correct depth is perceived. The drawing and the template may appear to be aligned while drawing, but when the user is unable to estimate the depth correctly it will be evident that there is a displacement when the result is observed from the side (see Figure 12).

The depth is perceived by the user thanks to the binocular vision. The HMD deviced renders the virtual object with a certain displacement in each display which depends mainly in the distance of the virtual object and in the interpupillary distance, forcing the eyes to maintain a certain vergence angle when looking at the virtual object. It was observed that even after display callibration [27], incorrect sensation of depth can cause the user to make errors of 1-6cm.

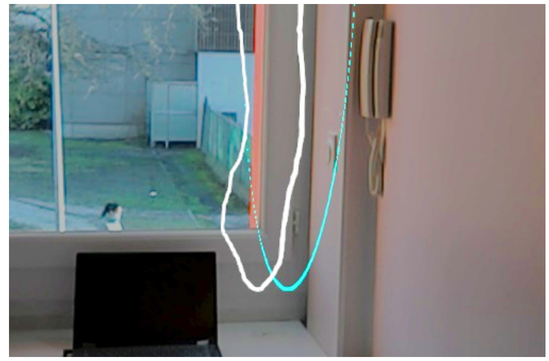

Fig. 12. Depth error when drawing a circle. While the drawing was in progress, circles looked aligned, but a displacement is evident when looking from the side.

VII. Conclusions and future work

Overall, the quantitative results roughly suggest that the precision can be expected from these tracking methods is around 20mm if the user has sufficient training. The built-in Hololens hand detection produces good results if the limitations are taken into account in the application design, meanwhile the loss of Vuforia marker tracking makes impractical its usage as input method.

To implement precise 3D input in the Hololens, we suggest that hand tracking should be studied for indirect interaction, since latency and display depth perception makes it unpractical for most direct manipulation cases.

References

[1] E. Klopfer and J. Sheldon, "Augmenting your own reality: Student authoring of science-based augmented reality games," *New Directions for Youth Development*, vol. 2010, no. 128, pp. 85–94, 2010. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/yd.378

[2] M. Bernardi and N. Abdelhamed, "Augmented reality for calligraphy learning," Universidad Autonóma de Madrid, Spain, 2021, [Tutored research and development project I, Image Processing and Computer Vision Master's, Erasmus Mundus].

[3] M. Speicher, B. D. Hall, and M. Nebeling, *What is Mixed Reality?* New York, NY, USA: Association for Computing Machinery, 2019, p. 1–15. [Online]. Available: https://doi.org/10.1145/3290605.3300767

[4] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays," *IEICE Trans. Information Systems*, vol. vol. E77-D, no. 12, pp. 1321–1329, 12 1994.

[5] A. H. Behzadan, B. W. Timm, and V. R. Kamat, "General-purpose modular hardware and software framework for mobile outdoor augmented reality applications in engineering," *Advanced Engineering Informatics*, vol. 22, no. 1, pp. 90–105, 2008, intelligent computing in engineering and architecture. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474034607000481

[6] H.-K. Wu, S. W.-Y. Lee, H.-Y. Chang, and J.-C. Liang, "Current status, opportunities and challenges of augmented reality in education," *Computers & Education*, vol. 62, pp. 41–49, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360131512002527

[7] O. Bimber and R. Raskar, *Spatial Augmented Reality Merging Real and Virtual Worlds*. AK Peters/CRC Press, 08 2005.

[8] S. Park, S. Bokijonov, and Y. Choi, "Review of microsoft hololens applications over the past five years," *Applied Sciences*, vol. 11, no. 16, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/16/7259

[9] T. Liao, "Future directions for mobile augmented reality research: Understanding relationships between augmented reality users, nonusers, content, devices, and industry," *Mobile Media & Communication*, vol. 7, p. 205015791879243, 09 2018.

[10] M. Dunleavy and C. Dede, *Augmented Reality Teaching and Learning*. New York, NY: Springer New York, 2014, pp. 735–745. [Online]. Available: https://doi.org/10.1007/978-1-4614-3185-5_59

[11] E. A. Chicaiza, E. I. De la Cruz, and V. H. Andaluz, "Augmented reality system for training and assistance in the management of industrial equipment and instruments," in *Advances in Visual Computing*, G. Bebis, R. Boyle, B. Parvin, D. Koracin, M. Turek, S. Ramalingam, K. Xu, S. Lin, B. Alsallakh, J. Yang, E. Cuervo, and J. Ventura, Eds. Cham: Springer International Publishing, 2018, pp. 675–686.

[12] C. Kounavis, A. Kasimati, and E. Zamani, "Enhancing the tourism experience through mobile augmented reality : Challenges and prospects," *International Journal of Engineering Business Management*, vol. D, 07 2012.

[13] P. Chen, X. Liu, W. Cheng, and R. Huang, "A review of using augmented reality in education from 2011 to 2016," in *Innovations in Smart Learning*, E. Popescu, Kinshuk, M. K. Khribi, R. Huang, M. Jemni, N.-S. Chen, and D. G. Sampson, Eds. Singapore: Springer Singapore, 2017, pp. 13–18.

[14] SketchAR, "The method of "progressive markers" in computer vision and augmented reality," https://medium.com/sketchar/the-method-of-progressive-markers-in-computer-vision-and-augmented-reality-f811c6e49e28, 2018, [Online; accessed 19-May-2021].

[15] J. Friedhoff, "Stuff we learned building drawalong ar," https://jfriedhoff.medium.com/drawalong-ar-90894cf56010, 2019, [Online; accessed 10-May-2021].

[16] S. Webel, U. Bockholt, T. Engelke, N. Gavish, M. Olbrich, and C. Preusche, "An augmented reality training platform for assembly and maintenance skills," *Robotics and Autonomous Systems*, vol. 61, no. 4, pp. 398–403, 2013, models and Technologies for Multi-modal Skill Training. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889012001674

[17] H.-Q. Le and J.-I. Kim, "An augmented reality application with hand gestures for learning 3d geometry," in *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2017, pp. 34–41.

[18] M. Sousa, J. a. Vieira, D. Medeiros, A. Arsenio, and J. Jorge, "Sleevear: Augmented reality for rehabilitation using realtime feedback," in *Proceedings of the 21st International Conference on Intelligent User Interfaces*, ser. IUI '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 175–185. [Online]. Available: https://doi.org/10.1145/2856767.2856773

[19] W. A. Hoff, K. Nguyen, and T. Lyon, "Computer-vision-based registration techniques for augmented reality," in *Intelligent Robots and Computer Vision XV: Algorithms, Techniques,Active Vision, and Materials Handling*, D. P. Casasent, Ed., vol. 2904, International Society for Optics and Photonics. SPIE, 1996, pp. 538 – 548. [Online]. Available: https://doi.org/10.1117/12.256311

[20] T. Holmdahl, "Build 2015: A closer look at the microsoft hololens hardware," https://blogs.windows.com/devices/2015/04/30/build-2015-a-closer-look-at-the-microsoft-hololens-hardware/#SJdFVbXXJdM2BCl0.97, 2015, [Online; accessed 29-Dec-2021].

[21] Microsoft, "Hololens (1st gen) hardware," https://docs.microsoft.com/en-us/hololens/hololens1-hardware, 2021, [Online; accessed 29-Dec-2021].

[22] Y. Liu, H. Dong, L. Zhang, and A. E. Saddik, "Technical evaluation of hololens for multimedia: A first look," *IEEE MultiMedia*, vol. 25, no. 4, pp. 8–18, 2018.

[23] V. Stropnik, K. Babuder, V. Crmelj, R. Vizintin, and M. Pogacnik, "A look into the future of sports: A study of the actual state of the art - the microsoft hololens and augmented reality," 07 2018, pp. 1–6.

[24] P. Hübner, K. Clintworth, Q. Liu, M. Weinmann, and S. Wursthorn, "Evaluation of HoloLens tracking and depth sensing for indoor mapping applications," *Sensors*, vol. 20, no. 4, p. 1021, Feb. 2020. [Online]. Available: https://doi.org/10.3390/s20041021

[25] S. Colaner, "What's inside microsoft's hololens and how it works," https://www.tomshardware.com/news/microsoft-hololens-components-hpu-28nm,32546.html, 2016, [Online; accessed 29-Dec-2021].

[26] J. Ashley, "How hololens displays work," https://www.imaginativeuniversal.com/blog/2015/10/18/how-hololens-displays-work/, 2015, [Online; accessed 29-Dec-2021].

[27] Microsoft, "Improve visual quality and comfort - calibrating your hololens," https://docs.microsoft.com/en-us/hololens/hololens-calibration#calibrating-your-hololens-1st-gen, 2021, [Online; accessed 29-Dec-2021].

[28] G. Kramida, "Resolving the vergence-accommodation conflict in head-mounted displays," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 7, pp. 1912–1931, 2016.

[29] J. VR, "Vergence-accommodation conflict," https://xinreality.com/wiki/Vergence-Accommodation_Conflict, 2019, [Online; accessed 29-Dec-2021].

[30] K. Guttag, "Hololens 2 display evaluation (part 2: Comparison to hololens 1)," https://kguttag.com/2020/07/08/hololens-2-display-evaluation-part-2-comparison-to-hololens-1/, 2020, [Online; accessed 29-Dec-2021].

[31] Microsoft, "Hololens (1st gen) hardware," https://docs.microsoft.com/en-us/windows/mixed-reality/design/direct-manipulation, 2021, [Online; accessed 29-Dec-2021].

[32] M. leap 1, "Magic leap," https://www.magicleap.com, 2018, [Online; accessed 31-Dec-2021].

[33] Microsoft, "Choosing your engine," https://docs.microsoft.com/en-us/windows/mixed-reality/develop/choosing-an-engine?tabs=unity, 2021, [Online; accessed 29-Dec-2021].

[34] ——, "Setting up your xr configuration," https://docs.microsoft.com/en-us/windows/mixed-reality/develop/unity/xr-project-setup?tabs=openxr, 2021, [Online; accessed 29-Dec-2021].

[35] ——, "What is the mixed reality toolkit," https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/?view=mrtkunity-2021-05, 2021, [Online; accessed 29-Dec-2021].

[36] ——, "Gaze and commit - composite gestures," https://docs.microsoft.com/en-us/windows/mixed-reality/design/gaze-and-commit#composite-gestures, 2021, [Online; accessed 29-Dec-2021].

[37] ——, "Hololensforcv," https://github.com/Microsoft/HoloLensForCV, 2021, [Online; accessed 30-Dec-2021].

[38] M. Doughty, "Arucodetectionhololens," https://github.com/doughtmw/ArUcoDetectionHoloLens-Unity, 2019, [Online; accessed 31-Dec-2021].

[39] Microsoft, "Spatial mapping," https://docs.microsoft.com/en-us/windows/mixed-reality/design/spatial-mapping, 2021, [Online; accessed 29-Dec-2021].

[40] O. Aydin, A. A. Taha, A. Hilbert, A. Khalil, I. Galinovic, J. Fiebach, D. Frey, and V. Madai, "On the usage of average hausdorff distance for segmentation performance assessment: hidden error when used for ranking," vol. 5, 01 2021.

[41] M. Shokoohi-yekta, B. Hu, H. Jin, and E. Keogh, "Generalizing dynamic time warping to the multi- dimensional case requires an adaptive approach."

[42] A. Efrat, Q. Fan, and S. Venkatasubramanian, "Curve matching, time warping, and light fields: New algorithms for computing similarity between curves," *Journal of Mathematical Imaging and Vision*, vol. 27, pp. 203–216, 04 2007.

[43] T. Eiter and H. Mannila, "Computing discrete frechet distance," 05 1994.

[44] Unity, "Keywordrecognizer - unity documentation," https://docs.unity3d.com/ScriptReference/Windows.Speech.KeywordRecognizer.html, 2021, [Online; accessed 29-Dec-2021].

[45] Microsoft, "Hololens (1st gen) input 212: Voice," https://docs.microsoft.com/en-us/windows/mixed-reality/develop/unity/tutorials/holograms-212, 2021, [Online; accessed 29-Dec-2021].