

Article

Distance Learning and Assistance Using Smart Glasses

Michael Spitzer ^{1,*}, Ibrahim Nanic ¹ and Martin Ebner ²

¹ Virtual Vehicle Research Center, Inffeldgasse 21/A, Graz 8010, Austria; ibrahim.nanic@v2c2.at

² Department Educational Technology, Graz University of Technology, Münzgrabenstraße 35a, Graz 8010, Austria; martin.ebner@tugraz.at

* Correspondence: michael.spitzer@v2c2.at

Received: 30 November 2017; Accepted: 25 January 2018; Published: 27 January 2018

Abstract: With the everyday growth of technology, new possibilities arise to support activities of everyday life. In education and training, more and more digital learning materials are emerging, but there is still room for improvement. This research study describes the implementation of a smart glasses app and infrastructure to support distance learning with WebRTC. The instructor is connected to the learner by a video streaming session and gets the live video stream from the learner's smart glasses from the learner's point of view. Additionally, the instructor can draw on the video to add context-aware information. The drawings are immediately sent to the learner to support him to solve a task. The prototype has been qualitatively evaluated by a test user who performed a fine-motor-skills task and a maintenance task under assistance of the remote instructor.

Keywords: distance learning; smart glasses; WebRTC; maintenance

1. Introduction

In recent years, many smart glasses devices emerged on the market with a reasonable price. The potential of such devices was already investigated in various domains. For example, smart glasses were used as an assistance system to guide visitors in a museum [1]. Also, in the educational domain, smart glasses are emerging. Google Glass was used in a context-aware learning use case—a physics experiment. The Google Glass measured the water fill level of a glass (camera) and frequency of the sound created by hitting the glass (microphone) and displayed a graph with the relation of the fill level and the frequency [2].

Also in medical education, smart glasses are used to solve communication and surgical education challenges in the operating room [3].

Smart glasses were already used in STEM education to monitor the students while they are performing a task in cyber security and forensic education. The smart glasses were used to record the field of view and the spoken words of the students. They were instructed to think out loud. The collected data will be used to analyze the learning success [4].

An investigation of promising pilot project studies was already published in a literature review study. The authors investigated smart glasses projects in education in the years 2013–2015. A high amount of the investigated studies focused on theoretical aspects as suggestions and recommendations without significant practical discoveries [5].

This issue motivated us to elaborate on the practical aspects of using smart glasses for distance learning scenarios.

Smart glasses provide various input possibilities such as speech, touch and gestures. The main advantage of using smart glasses is that users do not need their hands to hold the device. Therefore, these devices could be used to support participants in performing learning situations in which they

need their hands such as learning and training of fine-motor-skills. We defined the following research questions:

- RQ1: Are live annotations, such as arrows, rectangles or other types of annotations, in video streams helpful to perform different remote assistance and learning tasks?
- RQ2: Could the WebRTC infrastructure and software architecture be used to implement remote learning and assistance scenarios of different types?
- RQ3: Is a user participatory research suitable to derive software requirements for next iterations of the prototype?

The first research question (RQ1) evolved during our previous studies with smart glasses. We faced the issue that it was very difficult to describe locations of certain parts, tools or things which are important to perform the remote learning scenario; hence, we decided to mark the locations of these objects in the transmitted video stream in real-time to help the learner to find these objects. The second research question (RQ2) was defined to evaluate the capability of already developed real-time technology to support our learning scenarios. RQ3 evolved during other experiments where active participation of users was very effective to improve our software artifacts iteratively.

We already investigated the usage of smart glasses in several learning situations. For example, knitting while using smart glasses [6]. Additionally, we investigated the learning process how to assemble a Lego® Technic planetary gear while using smart glasses [7].

This study describes how to use video streaming technology to assist the wearer of smart glasses while learning a fine-motor-skill task. The instructor uses the web-based prototype to observe the subject during the learning situation. The instructor can see the learning and training situation from the learner's point of view because of the video stream coming from the camera of the smart glasses. The stream is also displayed in the display of the smart glasses. The remote user (instructor) can highlight objects by drawing shapes live in the video stream. The audio, video and the drawing shapes are transmitted via WebRTC [8]. We tested the prototype in two different settings. The first setting was a game of skill. Users had to assemble a wooden toy without any given instructions in advance. The second setting was an industrial use case in which the user had to perform a machine maintenance task. We evaluated this setting in our research lab on our 3D printer, which is a small-scale example of a production machine. The user had to clean the printing head of the 3D printer as a maintenance task without any prior knowledge of the cleaning procedure.

We did a qualitative evaluation of both scenarios to gather qualified information how to improve the learning scenarios and software prototype.

To keep this article clear and easy to read, the following personas were defined:

- **Instructor:** User of the web frontend who guides the smart glasses user remotely;
- **Subject:** Wearer of the smart glasses who uses the smart glasses app while performing a task.

2. Methodology

This paper describes the necessary work to develop a first prototype to be used to evaluate distance learning use cases with smart glasses.

The prototyping approach is appropriate for scenarios in which the actual user requirements are not clear or standing to reason. The real user requirements are often emerging during experimentation with the prototype. Additionally, if a new approach, feature or technology is introduced which has not been used before, experiments and learning is necessary before designing a full-scale information system [9].

The prototyping approach consists of 4 steps. At first, we identified the basic requirements. In our case it was necessary to communicate in real-time to support the remote learner during the learning scenario. The second step was to develop a working prototype. Then users test the prototype. The last step follows an agile approach. While users were testing the prototype, we identified necessary changes to the system and we provided a new version and tested it again in several iterations [10].

The prototyping approach is not only very effective by involving the end user into the development but also very effective from the developer's point of view. Since the software development process is very specific to the use case you only know how to build a system when you already have implemented major parts of it, but then it is often too late to change the system architecture fundamentally because of the experiences gained during the development phase [11].

The prototyping approach already prevents major software architecture and implementation flaws in early stage.

To test the prototype, two use cases were designed. They will also be used for future evaluation of distance learning scenarios. The first use case reflects a generic fine-motor-skills task, which can be validated with less effort. The second use case bridges the gap between a generic learning situation and a real-world industry use case. The learning transfer from a generic scenario to a real-world situation is a high-priority goal for a learning scenario. The main target of this study is to provide a suitable environment to perform future studies with smart glasses in the educational domain. In future work we will create guidance material for instructors and we will perform comparative studies to evaluate the performance of the distance learning approach with smart glasses compared to established learning and teaching approaches. The main purpose of this distance learning setting and the usage of smart glasses is to address issues with distance learning and assistance scenarios such as the instructor does not have a face-to-face communication to the subject. The expected impact of such a setting is that distance learning will be more effective as it is now without real-time annotations and video/audio streaming.

3. Technological Concept and Methods

This chapter elaborates on the details of the used materials. We describe the used server, web application, smart glasses app and the necessary infrastructure. Afterwards, we point at the method how we evaluated the two already introduced use cases: the fine-motor-skills task and the maintenance task.

We implemented an Android-based [12] prototype for the smart glasses Vuzix M100 [13]. Additionally, a web-based prototype was implemented acting as server. Figure 1 shows the system architecture and the data flow. The instructor is using the web client which shows the live video stream of the smart glasses camera. Additionally, the audio of the instructor as well as the audio of the subject is transmitted. The server is responsible to transmit the WebRTC data stream to the web client and to the smart glasses Android app. WebRTC is a real-time communications framework for browsers, mobile platforms and IoT devices [8]. With this framework, audio, video and custom data (in our case: drawn shapes) are transmitted in real-time. This framework is used for both, the server and in the client Android application. Figure 1 shows the system architecture and infrastructure. All communication is transferred and managed by a NodeJS [14] server.

The generic learning scenario workflow consists of several steps:

- (1) The instructor logs on to the web frontend on a PC, tablet or any other device;
- (2) The next step is to start the video broadcast by clicking on "Broadcast". A separate room is then created and waiting until the wearer of smart glasses joins the session;
- (3) The subject starts the smart glasses client app and connects to the already created room;
- (4) The video stream starts immediately after joining the room.

Figure 2 shows the screen flow in detail.

The following sections describe the Android app and the NodeJS server implementation in detail. Figure 3 shows a screenshot of the web application and the smart glasses app while performing the 3D printer maintenance task.

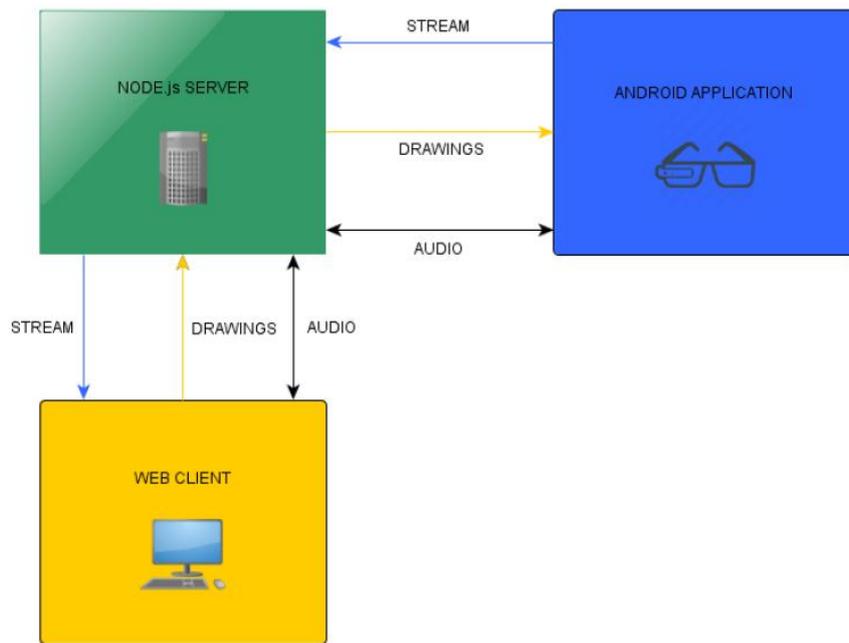


Figure 1. System architecture and infrastructure.

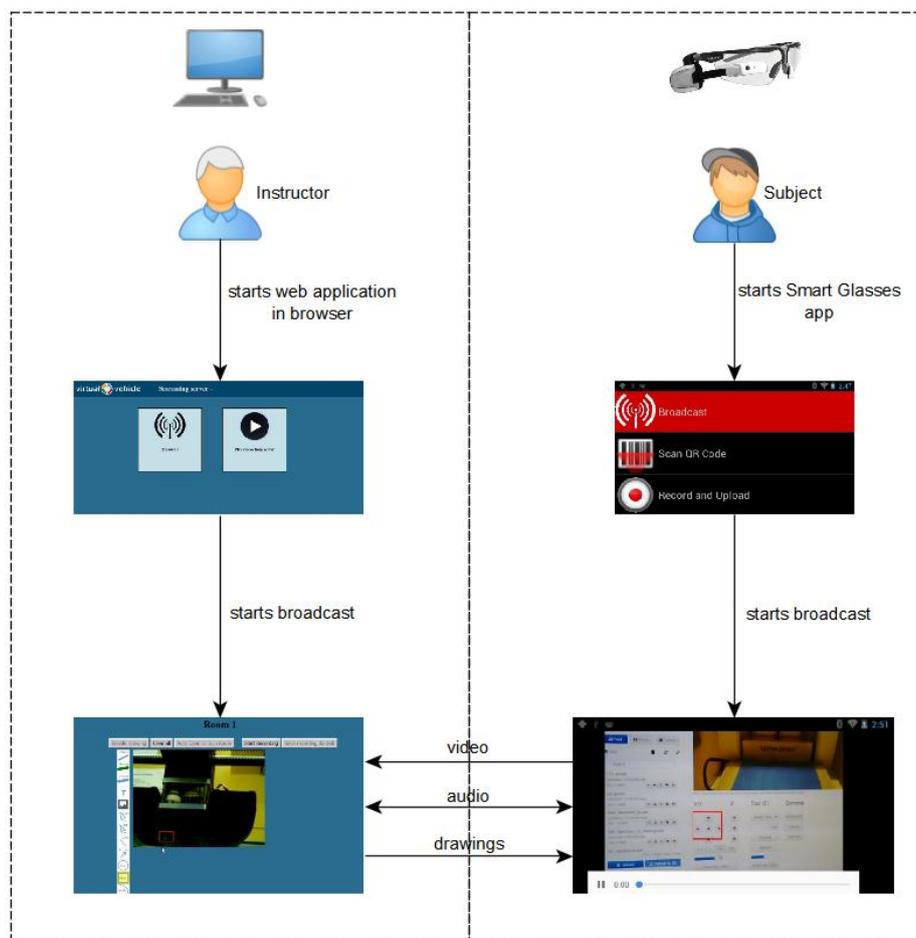


Figure 2. Screen flow.



Figure 3. Web frontend and smart glasses app.

3.1. Server and Web App Implementation

We used NodeJS to implement the server. The WebRTC multiconnection implementation is based on the implementation of Muaz Khan, the repository is available on GitHub [15]. The server is started on localhost, port 9001. The entry page of the web application offers two features:

- (1) Broadcast from local camera: The internal webcam of the computer is used;
- (2) Play video directly from server: A previously recorded video will be played.

Figure 4 shows the start screen.

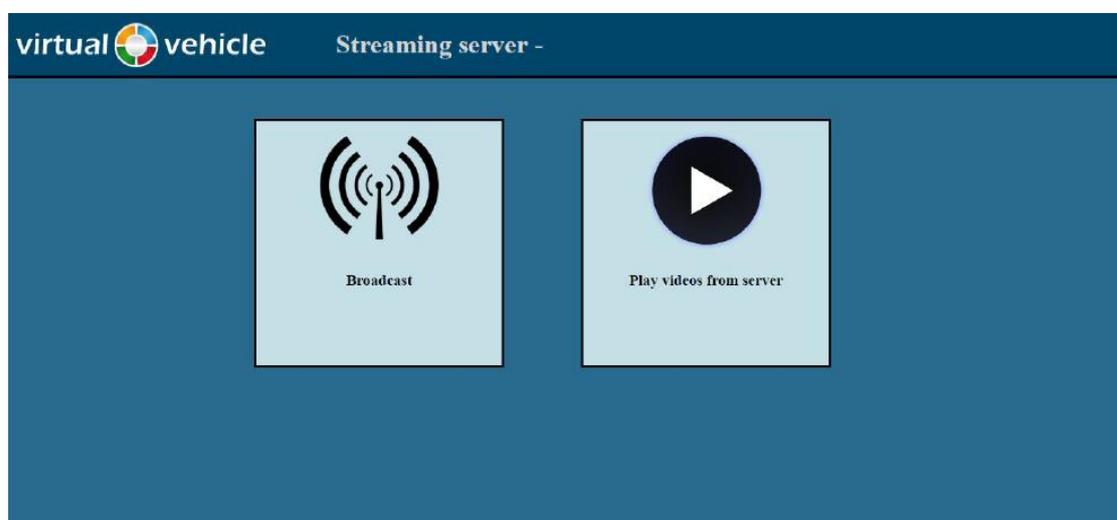


Figure 4. Start screen of the web application.

When the instructor selects the first option, a video stream of the local camera is shown. The video stream will be switched to the camera stream of the smart glasses immediately after the subject connects to the system. With no subject present, the instructor can record a video, for example a training, how to assemble or disassemble an object or any other challenging fine-motor-skills task. The recording is stored on the server; additionally, the instructor can download the video to the hard disk. When a subject connects to the system, the instructor can enable drawing as an additional feature. With this feature enabled, the instructor can augment the video with several objects as colored strokes, arrows, rectangles, text, image overlays and Bezier curves [16]. The drawn objects and the video stream are transferred in real-time to the subject. The second option plays videos already created and uploaded to the server. Figure 5 shows a video streamed by the smart glasses app. In the displayed web app, the drawing feature is enabled.

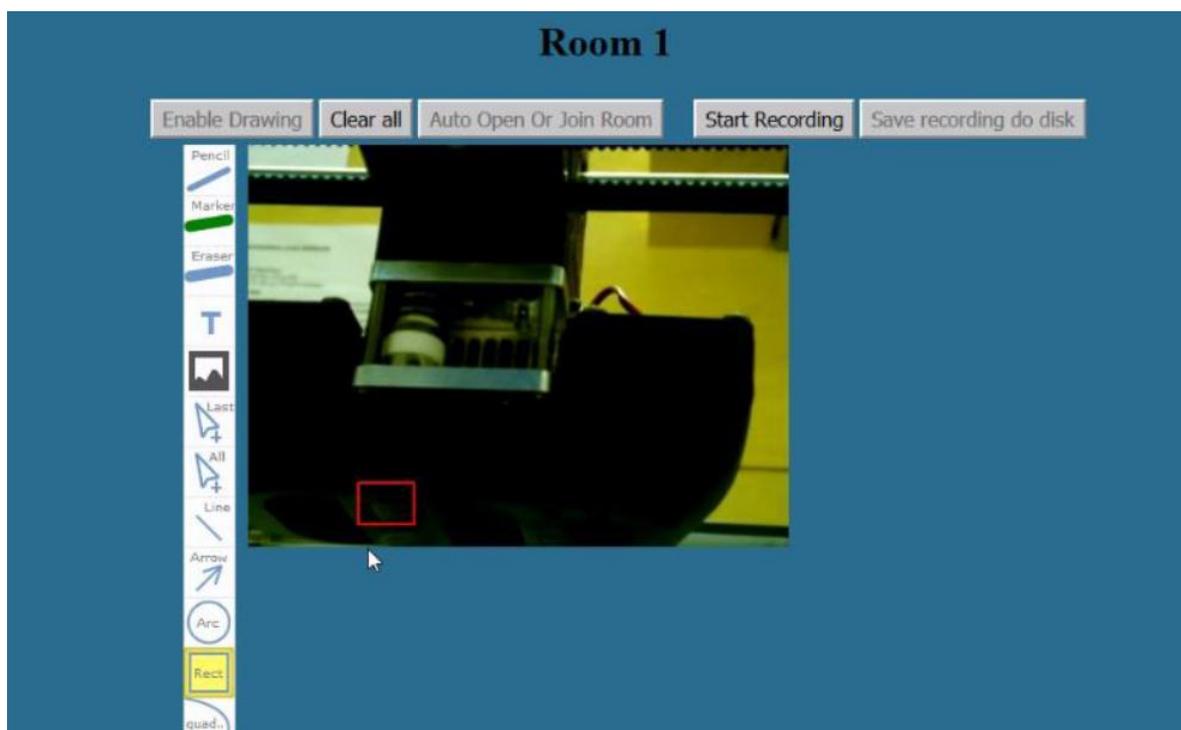


Figure 5. Broadcast option.

3.2. Smart Glasses App Implementation

The smart glasses we used were the Vuzix M100. This device was chosen because it was the only one which was available for us to buy as we started the project. Table 1 shows the specifications of the Vuzix M100.

The conventional way to develop applications for Android-based devices is to write a Java [18] application with Android Studio [19] in combination with the Android SDK [20]. The first approach was to implement such an app. An Android WebView [21] was used to display the WebRTC client web app, served from our server. With an Android WebView, a web page can be shown within an Android app. Additionally, JavaScript execution was enabled to support the WebRTC library. We built the app against API level 15 for the smart glasses. A big issue emerged in this phase; because of the low API level, the WebView was not able to run the used WebRTC Multiconnection library. Therefore, we switched our system architecture from an ordinary Java Android application to a hybrid app. Our new app is based on Cordova [22] and uses a Crosswalk [23] WebView. Apache Cordova is an open source mobile development framework. Figure 6 shows the architecture of such a Cordova

app. The smart glasses app is implemented as a web app and uses Cordova to be able to run on an Android device.

Table 1. Vuzix M100 specifications [17].

Feature	Vuzix M100 Specification
Display	Widescreen 16:9 WQVGA
CPU	OMAP4460@1.2GHz
Memory	1 GB RAM 4 GB flash
Sensors	3 DOF (degree of freedom) gesture engine Ambient light GPS Proximity
Connectivity	Micro USB Wi-Fi 802.11 b/g/n Bluetooth
Controls	4 control buttons Remote control app paired via Bluetooth (available for iOS and Android) customizable voice control gestures
Operating system	Android ICS 4.04, API level 15

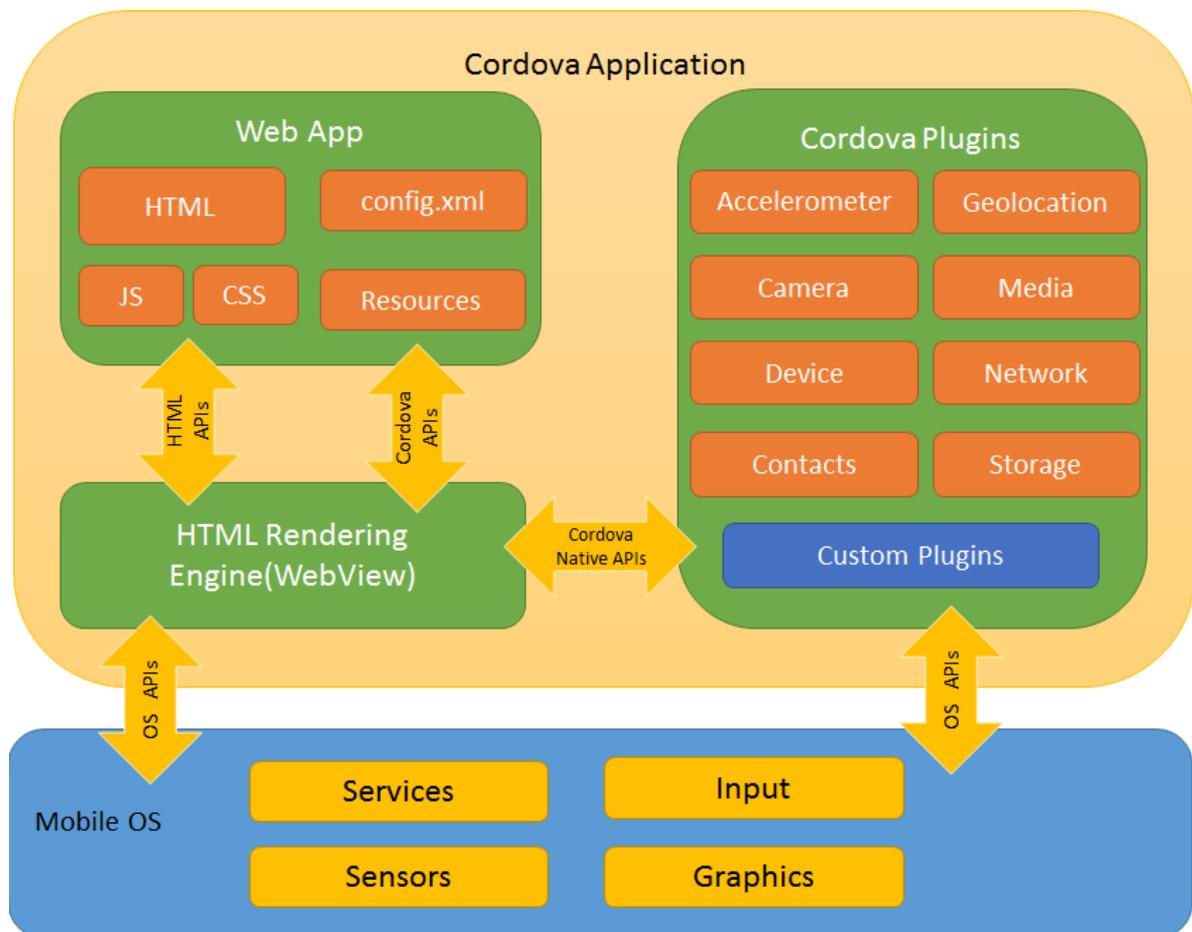


Figure 6. Cordova app architecture [24].

Crosswalk was started in 2013 by Intel [25] and is a runtime for HTML5 applications [23]. These applications can then be transformed to run on several targets as iOS, Android, Windows desktop and Linux desktop applications. Figure 7 shows the workflow, how to set-up a Cordova app with Crosswalk.

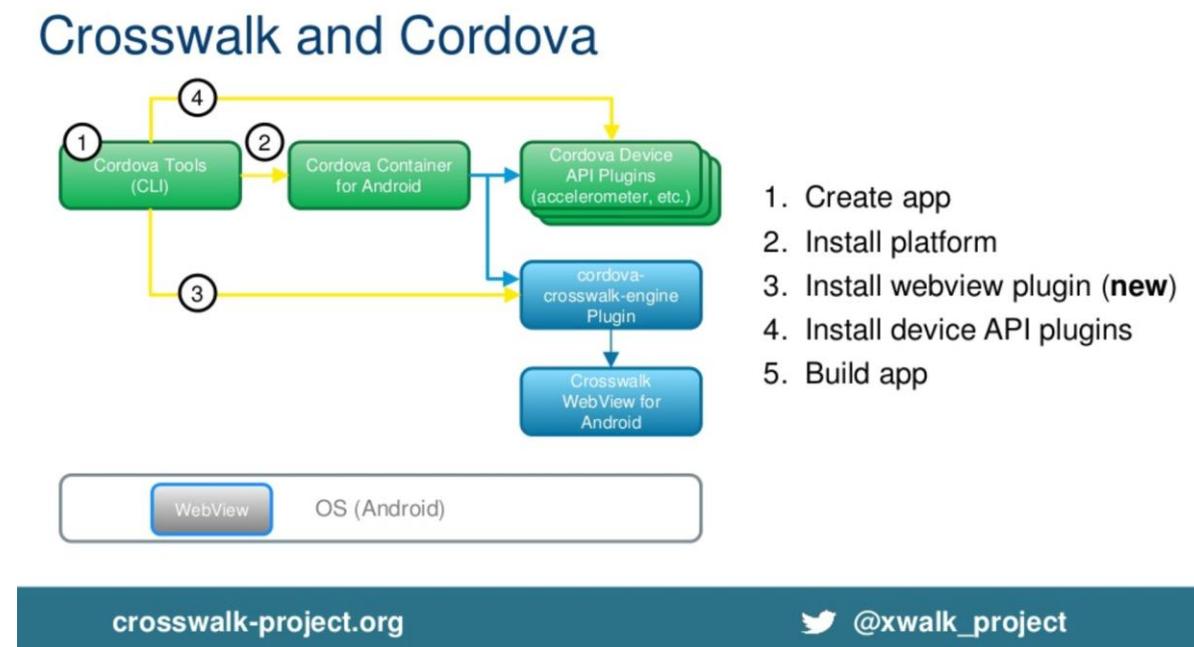


Figure 7. Crosswalk and Cordova workflow [26].

The main reason why we had to switch to such a hybrid solution is that WebViews from older Android systems are not updated to be able to run state-of-the-art web code. Crosswalk is now discontinued because the code and features of Google Chrome are now shared with newer Android WebView implementations; hence, the newer WebViews are now kept up-to-date and support state-of-the-art web applications. Additionally, progressive web apps can provide native app features [25]. Therefore, for the next prototype, we can use the implemented WebView of newer Android systems because they are actively maintained by Google and will share current features of the Google Chrome development.

Programmers can now rely on active support for new web features of the WebView. A requirement is that the Android version of newer smart glasses must be at least Android Lollipop (Android 5.0) [27].

Since this is not the case for the Vuzix M100 as well as for other Android-based smart glasses, our hybrid solution is still necessary for these devices.

The UI of the Android app is optimized for small displays as smart glasses' displays. After the start of the app a menu with three options is displayed:

- (1) **Broadcast:** This starts the WebRTC connection and connects to the server and joins the session of the running web app automatically.
- (2) **Scan QR code:** This feature enables context-aware information access. Users can scan a QR code, e.g., on a machine to access certain videos related to the machine. This approach was already used in another smart glasses study to specify the context (the machine to maintain) because using QR codes to define the context is more efficient than browsing information and selecting the appropriate machine in the small display (UI) of the smart glasses [28].
- (3) **Record and upload.** Smart glasses users can record their own videos in their point-of-view. The videos are then uploaded to the server and are then accessible by the web frontend users.

Figure 8 shows the start screen of the smart glasses app.

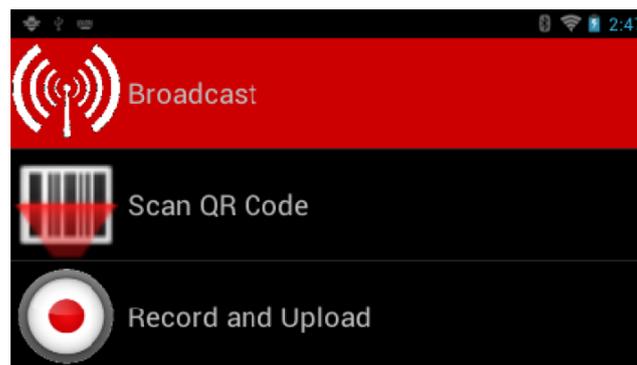


Figure 8. Start screen of the smart glasses app.

When the user selects the first option, a video stream of his own point of view (the video stream of the camera of the smart glasses) is shown. Additionally, drawn objects of the web application user (instructor) are displayed in real-time. This is very helpful when the subject investigates a complicated UI or machine. Then the instructor can guide the subject with drawn arrows or rectangles to the correct direction to perform the maintenance or fine-motor-skills task correctly. Figure 9 shows the 3D printer maintenance UI. Since the subject has never used the UI before, he/she needs guidance on how to use the UI. The highlighted area (red rectangle) shows the buttons which must be used for the maintenance procedure. Along with audio instructions, the subject in our evaluation managed to perform the maintenance task efficiently. The maintenance task will be explained in detail in Section 3.2.

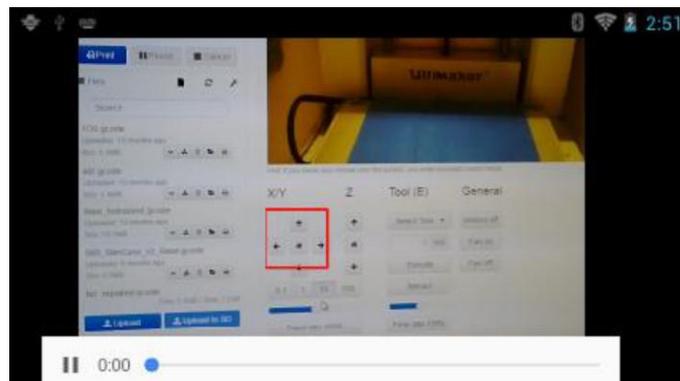


Figure 9. Highlighted parts of the maintenance UI displayed on the smart glasses.

3.3. Evaluation Methods

To evaluate the two use cases, an end-user participatory research was performed with one test person of the target group to get qualitative feedback of the used technology as well as feedback on the design of the learning scenario.

In a user participatory research study, the subject is not only a participant, but also is highly involved in the research process [29].

In our case, the subject gave qualified feedback to the used device and the concept of the learning scenarios which are now considered for the next iteration of the prototype.

A quantitative research with a larger group of users of the target group will be performed after the qualitative feedback is considered. This follows the approach we used in our previous studies [6,7].

4. Evaluation Scenarios

We decided to use two different scenarios. The first scenario is a general approach to investigate a more generic use case. The introduced software and infrastructure was used to help subjects to

perform general fine-motor-skills tasks. The second use case is a specific maintenance use case which is often seen in industrial domain. In the project FACTS4WORKERS we investigate a use case how to clean a lens of a laser cutter [30]. The whole procedure can be split into small fine-motor-skills tasks; hence, our use cases one and two fit well to real industry use cases.

4.1. Fine-Motor-Skills Scenario

Since a lot of tasks can be separated into smaller fine-motor-skills tasks, the first approach was to evaluate our solution with such a task. We chose a wooden toy which reflects an assembly task. The subject gets the wooden parts and must assemble the toy without a manual or pictures of the finished assembly. Only by getting voice assistance and augmentation of the video stream with drawings, the subject must assemble the toy. The toy consists of 30 small wooden sticks, 12 wooden spheres with five holes each and one sphere without any holes. Figure 10 shows the parts of the toy.



Figure 10. Parts of the wooden toy.

The reason why we chose this toy was that it is not likely that the subject is able to assemble the toy without any help hence the subject must rely on an assistance system to assemble the toy with help of a remote instructor. Figure 11 shows the assembled toy.



Figure 11. Assembled toy.

4.2. Maintenance Scenario

The second evaluation use case is a maintenance task. We used our 3D printer as an industry machine. A maintenance task on the 3D printer can be compared to a real-world maintenance use case in the production domain. Since it is very difficult to test such scenarios on a real industry machine, the first approach was to test the scenario with our 3D printer in a safe environment. The task was to clean the printhead of the 3D printer. The following steps must be performed:

- (1) Move printhead to the front and to the center;
- (2) Heat the printhead to 200 °C;
- (3) Use a thin needle to clean the printhead;
- (4) Deactivate printhead heating;
- (5) Move printhead back to the initial position.

The heating and cooling of the printhead can be achieved by using a web-based UI of the 3D printer. Additionally, users can move the printhead by pressing position adjust buttons in the 3D printer web UI. Our 3D printer uses the Web UI provided by OctoPrint [31]. Two screens are necessary to perform the maintenance operation. Figure 12 shows the printhead movement UI. The arrows indicate the movement directions. The area of interest is marked with a green frame and a speech balloon.

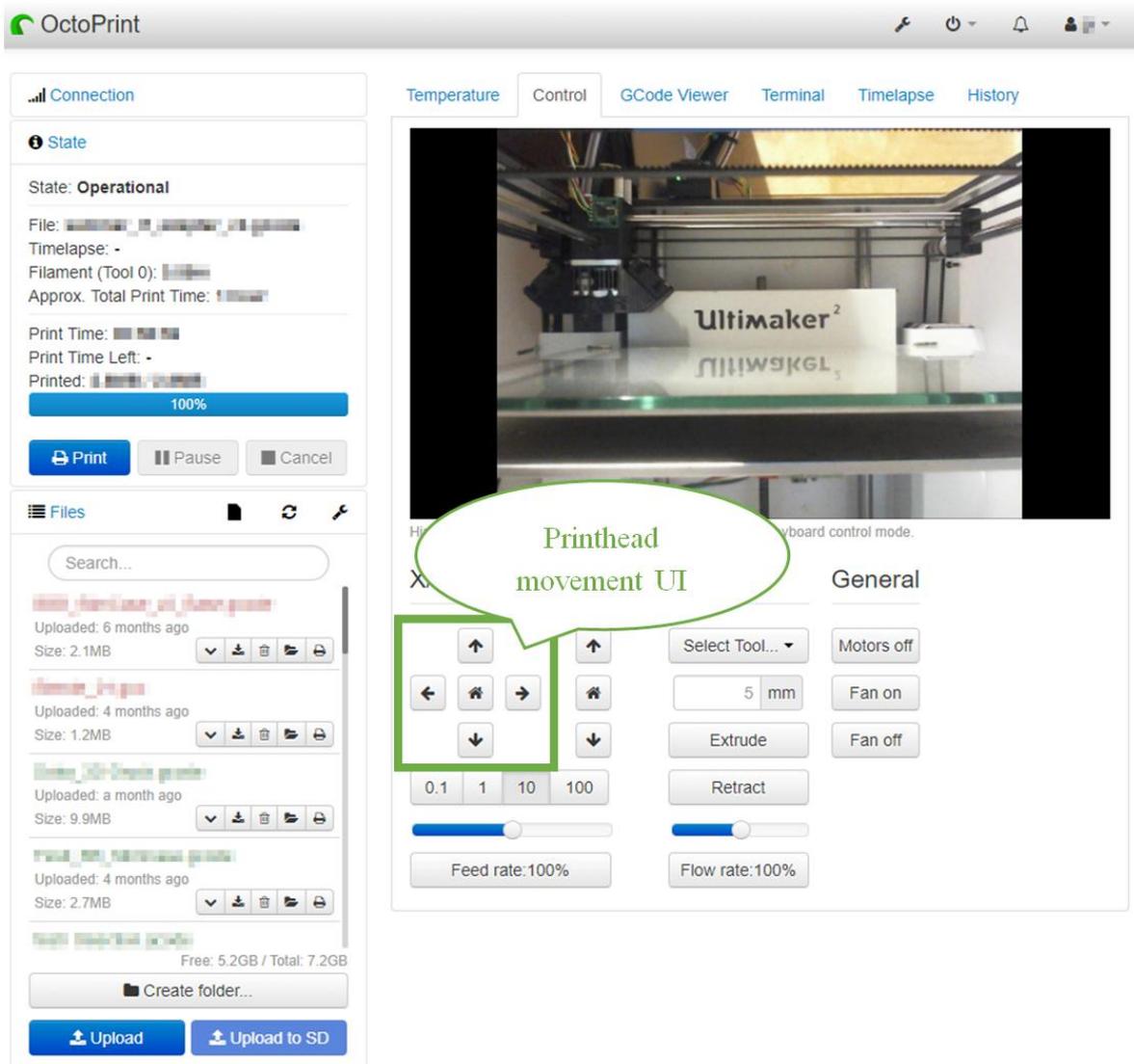


Figure 12. Printhead movement UI.

This UI is used to move the printhead to the front to access it with the needle. The next step is to heat it to a very high temperature to fluidize the printing material in the printhead to make the cleaning procedure possible. Figure 13 shows the printhead heating UI. Again, the area of interest is marked with a green frame and a speech balloon.

After the printhead is heated to the appropriate temperature, the cleaning process starts. The subject inserts the needle into the material hole to clean the print head. After the cleaning process, the subject sets the temperature of the printhead (tool) back to normal temperature (23 °C) and moves the printhead back to the home position. All of these steps are comparable to a real industry use case. At first the machine must be set into a maintenance mode and then some tools must be used to perform a certain maintenance task.

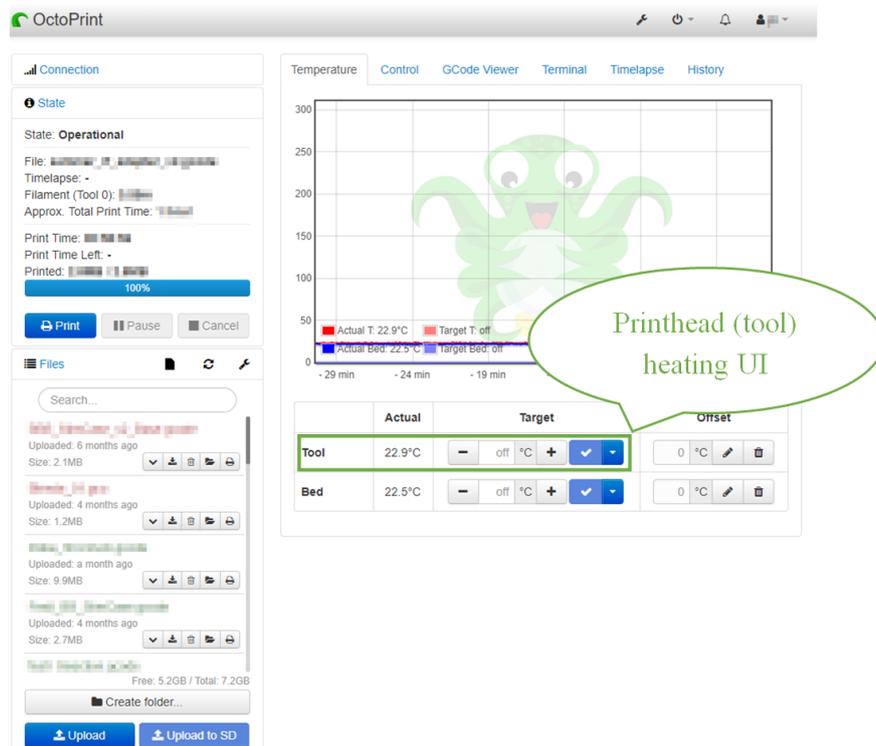


Figure 13. Heat printhead UI.

5. Results

This section elaborates on the details of the qualitative evaluation of the previously described evaluation scenarios. Both scenarios were performed by the same subject without knowing the scenarios a priori. At first, the subject performed the general scenario (fine-motor-skills training with a toy). Figure 14 shows screenshots of the assembly of the toy. The instructor first used the drawing feature of the app to explain the different parts. Then the instructor tried to draw arrows to explain the shape and the assembly procedure of the toy. It turned out that explaining the assembly process only by drawing arrows or other shapes into the screen was not enough because of the challenging assembly procedure. Therefore, the instructor switched to verbal instructions which worked considerably better. During the learning scenario, the subject ignored the video stream and the drawings and just listened to the voice explanations of the instructor. This behavior justifies our approach, not only to stream video and drawings, but also voice.



Figure 14. Screenshots made by the instructor during the assembly of the toy.

The second scenario was performed by the same subject. We tested our solution while performing a maintenance task. Figure 15 shows our 3D printer research lab. The computer is used to remote control the 3D printer.

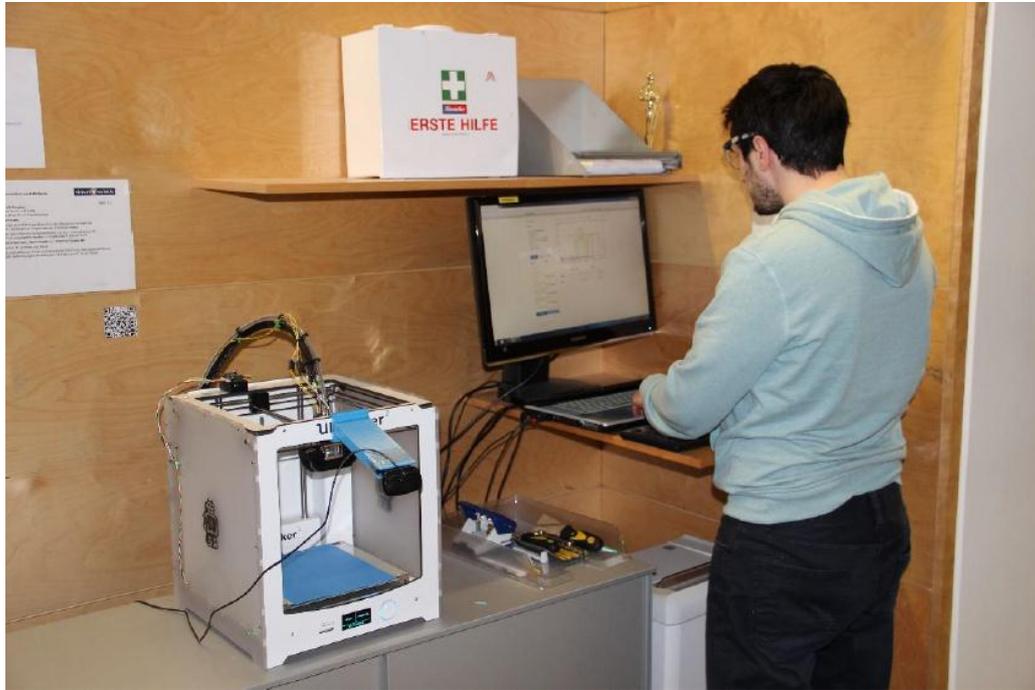


Figure 15. 3D printer research lab @ Virtual Vehicle Research Centre.

The subject was not familiar with the web UI of the 3D printer; he had never used the UI before. In this maintenance phase, the drawing feature was very helpful to guide the subject through the process. Figure 16 shows the red rectangle drawn by the instructor to explain the subject which buttons he had to push to move the printhead. Marking the area of interest of the UI by using a drawn shape was very effective because with audio it is very difficult to explain which buttons are the correct buttons for this task. Imagine an industry machine where pressing the wrong button could cost a lot of money or could be very dangerous. In this case, it is better to mark the buttons clearly by drawing a shape (arrow, rectangle...). One issue of the drawing shapes came up while testing this use case. The shapes are positioned in the video stream related to the screen dimensions. This means that if the subject turns his head, the video stream will then have a new field of view, but the rectangle will be on the same position relative to the screen. Therefore, the rectangle marks now another part. This will be solved in future iterations of the prototype. The shapes are then positioned correctly in space, even if the smart glasses user turns his head, the shapes will be stick to the correct spatial position. In this maintenance use case this was not a big issue because if the subject is focusing on a certain process of the maintenance task (interaction with the web UI, interaction with the printhead) his/her head position is quite fixed while concentrating on the current step. In other learning scenarios or industry use cases this could be a bigger issue, then the shapes should be placed with spatial awareness.

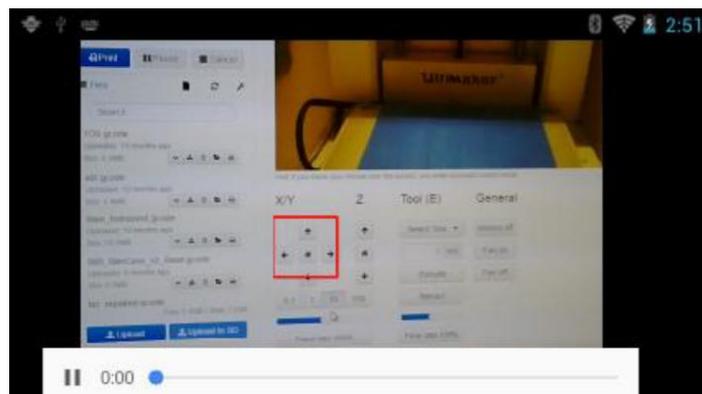


Figure 16. Red rectangle visible in the smart glasses UI.

After the subject finished all tasks in the web UI he started to clean the printhead. This was very challenging for him because he did not even know how the printhead and the drawing pin looked. Figure 17 shows the drawing pin of the printhead marked by the instructor. In this phase the drawing feature of our system was very effective because to explain the exact position of the pin by voice was very difficult but by drawing the red rectangle in the point-of-view of the subject, he found the drawing pin in a very short time.

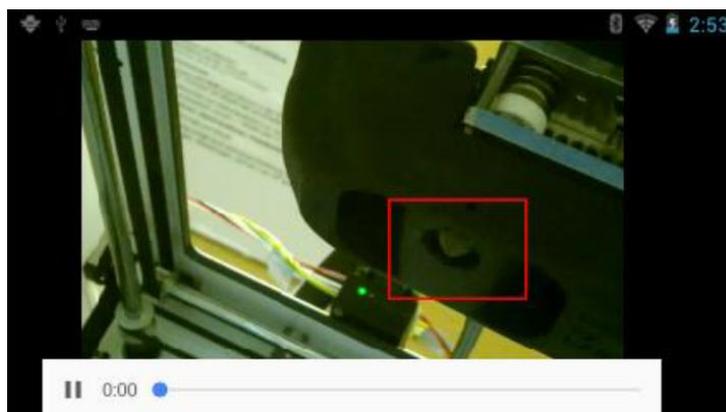


Figure 17. Marked drawing pin of the printhead.

The next step was to use the very small needle to clean the drawing pin. During the cleaning procedure the subject had to focus on the process and did not use the information of the smart glasses display. Additionally, it was quite challenging for the subject to focus his eyes alternating between the drawing pin and the smart glasses display. This problem could be solved by using see-through devices such as the Microsoft HoloLens [32].

Eventually, the subject totally focused his eyes on the drawing pin to solve the task. In such a high-concentration phase, the smart glasses display should be turned off to not distract the subject. Figure 18 shows the subject while cleaning the printhead.



Figure 18. The subject performed the cleaning procedure.

6. Discussion and Conclusions

A WebRTC server and a smart glasses app were developed to implement remote learning and assistance. WebRTC was used to implement the streaming functionality, which fit for both use cases (RQ2). We tested our solution in a user-participatory research by performing a qualitative evaluation of two different learning scenarios. The first scenario (toy) was a more generic scenario to evaluate fine-motor-skills tasks and the second scenario was more industry-related, a maintenance use case. We derived requirements for our solution (Table 2) for the next iteration of our prototype (RQ3). The live annotations (drawings) were very helpful, especially during the maintenance task. With drawings, the focus of interest of the subject can be set very effectively (RQ1). After the next prototype is implemented, we will perform a quantitative research study with more users of the target group to validate our system. In future studies, detailed transcriptions and recordings of the whole learning scenario will be analyzed.

Table 2. Derived requirements for the next iteration of the prototype.

Derived Requirements	Description
Spatial awareness for drawing the shapes	The shapes should be drawn with spatial awareness. When the subject turns his head, the drawings should stay on the same object in space.
Voice command to switch off the Smart glasses screen	This feature is necessary in situations in which the subject needs to totally focus on the task and not on the display of the smart glasses
See-through devices	Since smart glasses are now easily available we will implement the prototype on other devices with see-through displays as the Microsoft HoloLens

The whole software artifact and setting can now be used for several educational scenarios. The next step is to create coaching material to support the instructor to get familiar with this new way of assisting subjects remotely.

Additionally, the effectiveness of this system has to be investigated and statistical data such as how often the instructor had to repeat the voice commands has to be gathered. These issues will be addressed in further studies.

Acknowledgments: This study has received funding from FACTS4WORKERS—Worker-centric Workplaces for Smart Factories—the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 636778. The authors acknowledge the financial support of the COMET K2—Competence Centers for Excellent Technologies Programme of the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT), the Austrian Federal Ministry of Science, Research and Economy (BMWFW), the Austrian Research Promotion Agency (FFG), the Province of Styria and the Styrian Business Promotion Agency (SFG). The project FACTS4WORKERS covers the costs to publish in open access.

Author Contributions: Michael Spitzer did the use-cases development, research design, software architecture, software development support and wrote the paper. Ibrahim Nanic was responsible for the software implementation. Martin Ebner reviewed and supervised the research study.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Tomiuc, A. Navigating culture. Enhancing visitor museum experience through mobile technologies. From smartphone to google glass. *J. Media Res.* **2014**, *7*, 33–47.
2. Kuhn, J.; Lukowicz, P.; Hirth, M.; Poxrucker, A.; Weppner, J.; Younas, J. gPhysics—Using Smart Glasses for Head-Centered, Context-Aware Learning in Physics Experiments. *IEEE Trans. Learn. Technol.* **2016**, *9*, 304–317. [[CrossRef](#)]
3. Moshtaghi, O.; Kelley, K.S.; Armstrong, W.B.; Ghavami, Y.; Gu, J.; Djalilian, H.R. Using Google Glass to solve communication and surgical education challenges in the operating room. *Laryngoscope* **2015**, *125*, 2295–2297. [[CrossRef](#)] [[PubMed](#)]
4. Kommera, N.; Kaleem, F.; Harooni, S.M.S. Smart augmented reality glasses in cybersecurity and forensic education. In Proceedings of the 2016 IEEE Conference on Intelligence and Security Informatics (ISI), Tucson, AZ, USA, 28–30 September 2016; pp. 279–281. [[CrossRef](#)]
5. Sapargaliyev, D. Learning with wearable technologies: A case of Google Glass. In Proceedings of the 14th World Conference on Mobile and Contextual Learning, Venice, Italy, 17–24 October 2015; pp. 343–350. [[CrossRef](#)]
6. Spitzer, M.; Ebner, M. Use Cases and Architecture of an Information system to integrate smart glasses in educational environments. In Proceedings of the EdMedia—World Conference on Educational Media and Technology, Vancouver, BC, Canada, 28–30 June 2016; pp. 57–64.
7. Spitzer, M.; Ebner, M. Project Based Learning: From the Idea to a Finished LEGO® Technic Artifact, Assembled by Using Smart Glasses. In *Proceedings of EdMedia*; Johnston, J., Ed.; Association for the Advancement of Computing in Education (AACE): Washington, DC, USA, 2017; pp. 269–282.
8. WebRTC Home | WebRTC. Available online: <https://webrtc.org> (accessed on 26 November 2017).
9. Alavi, M. An assessment of the prototyping approach to information systems development. *Commun. ACM* **1984**, *27*, 556–563. [[CrossRef](#)]
10. Larson, O. Information systems prototyping. In Proceedings of the Interez HP 3000 Conference, Madrid, Spain, 10–14 March 1986; pp. 351–364.
11. Floyd, C. A systematic look at prototyping. In *Approaches to Prototyping*; Budde, R., Kuhlenkamp, K., Mathiassen, L., Züllighoven, H., Eds.; Springer: Berlin/Heidelberg, Germany, 1984; pp. 1–18.
12. Android. Available online: <https://www.android.com> (accessed on 26 November 2017).
13. Vuzix | View the Future. Available online: <https://www.vuzix.com/Products/m100-smart-glasses> (accessed on 26 November 2017).
14. Node.js. Available online: <https://nodejs.org/en> (accessed on 26 November 2017).

15. WebRTC JavaScript Library for Peer-to-Peer Applications. Available online: <https://github.com/muaz-khan/RTCMulticonnection> (accessed on 26 November 2017).
16. De Casteljau, P. *Courbes a Poles*; National Institute of Industrial Property (INPI): Lisbon, Portugal, 1959.
17. Vuzix M100 Specifications. Available online: <https://www.vuzix.com/Products/m100-smart-glasses#specs> (accessed on 26 November 2017).
18. Java. Available online: <http://www.oracle.com/technetwork/java/index.html> (accessed on 26 November 2017).
19. Android Studio. Available online: <https://developer.android.com/studio/index.html> (accessed on 26 November 2017).
20. Introduction to Android | Android Developers. Available online: <https://developer.android.com/guide/index.html> (accessed on 26 November 2017).
21. WebView | Android Developers. Available online: <https://developer.android.com/reference/android/webkit/WebView.html> (accessed on 26 November 2017).
22. Apache Cordova. Available online: <https://cordova.apache.org> (accessed on 28 November 2017).
23. Crosswalk. Available online: <http://crosswalk-project.org> (accessed on 26 November 2017).
24. Architectural Overview of Cordova Platform—Apache Cordova. Available online: <https://cordova.apache.org/docs/en/latest/guide/overview/index.html> (accessed on 26 November 2017).
25. Crosswalk 23 to be the Last Crosswalk Release. Available online: <https://crosswalk-project.org/blog/crosswalk-final-release.html> (accessed on 26 November 2017).
26. Crosswalk and the Intel XDK. Available online: <https://www.slideshare.net/IntelSoftware/crosswalk-and-the-intel-xdk> (accessed on 26 November 2017).
27. WebView for Android. Available online: <https://developer.chrome.com/multidevice/webview/overview> (accessed on 26 November 2017).
28. Quint, F.; Loch, F. Using smart glasses to document maintenance processes. In *Mensch & Computer—Tagungsbände/Proceedings (n.d.)*; Oldenbourg Wissenschaftsverlag: Berlin, Germany, 2015; pp. 203–208. ISBN 978-3-11-044390-5.
29. Cornwall, A.; Jewkes, R. What is participatory research? *Soc. Sci. Med.* **1995**, *41*, 1667–1676. [CrossRef]
30. Spitzer, M.; Schafler, M.; Milfelner, M. Seamless Learning in the Production. In *Proceedings of the Mensch und Computer 2017—Workshopband*. Regensburg: Gesellschaft für Informatik e.V., Regensburg, Germany, 10–13 September 2017; Burghardt, M., Wimmer, R., Wolff, C., Womser-Hacker, C., Eds.; pp. 211–217. [CrossRef]
31. OctoPrint. Available online: <http://octoprint.org> (accessed on 27 November 2017).
32. Microsoft HoloLens | The Leader in Mixed Reality Technology. Available online: <https://www.microsoft.com/en-us/hololens> (accessed on 27 November 2017).



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).