UNIVERSITY OF TWENTE.

BSc Thesis Advanced Technology
Faculty of Science and Technology

Design and Assessment of an XR-guided MRI Interface and Intuitive Scan-plane Control

Irina Muriël Kramers

July 6, 2024

BSc Assignment Committee
Chair: prof.dr.ir. B. ten Haken
Supervisor: dr.ir. W. M. Brink
Member: dr.ir. R. G. J. Damgrave
External member: dr.ir. G. C. Vreman - de Olde

Research Chair
Department of Imaging & Sensing
Research group Magnetic Detection & Imaging
Faculty of Science and Technology

Design and Assessment of an XR-guided MRI Interface and Intuitive Scan-plane Control

Irina Muriël Kramers

July 6, 2024

Abstract

Real-time MR imaging is valuable in monitoring surgical procedures such as biopsies, ablations and radiotherapy. Major challenges during MRI-guided interventions include convoluted scan-plane control and limited scan visibility options inside the MRI room. This project proposes an interactive XR-interface with the MRI scanner as a solution. A proof-of-concept was realised through establishing bidirectional communication between an MRI scanner and XR-HMD via an intermediary PC, using the Siemens Access-i framework. During real-time MR image acquisition, the user can connect to the scanner via Wi-Fi in the holographic environment. Acquired images are shown on a holographic panel, and the user can modify the scan-plane position and orientation using hand gestures. The XR-HMD is MR-safe up to 40 mT. Data transmission latencies of 1.38 ± 0.18 s and 508 ± 194 ms were measured for image and command data, respectively. User test participants were positive about the system and its usefulness in clinical implementations, with ratings of 0.81 ± 0.18 and 0.95 ± 0.11 out of 1.00. Although the current prototype is subject to hardware limitations, it is believed that future XR-interface versions could reduce procedural time, complexity and cost of MRI-guided interventions.

Keywords: MRI-guided interventions, Mixed Reality, scan-plane control

Report contents

Definitions

Below follow the definitions of several technical terms as they are used in the document, based on [1].

Augmented Reality: the addition of digitally generated elements to (perceived) reality for an enhanced experience of a real-world environment.

Virtual Reality: visualisation of a completely digitally generated environment with no direct connection to reality.

Mixed Reality: a combination of Augmented and Virtual Reality, using reality as basis while allowing for addition of interactive 3D data.

Abbreviations

Below follows a list of abbreviations used in the report. Their meanings are also given at first occurrence in the document.

AR	*Augmented Reality*
CPU	*Central processing unit*
CT	*Computed Tomography*
DICOM	*Digital Imaging and Communications in Medicine*
FPS	*Frames per second*
HMD	*Head-mounted device*

IP	*Internet Protocol*
MATLAB	*Matrix Laboratory*
MR	*Magnetic Resonance*
MRI	*Magnetic Resonance Imaging*
PC	*Personal Computer*
RAM	*Random-access memory*
RF	*Radiofrequency*
ROI	*Region of interest*
SNR	*Signal-to-noise ratio*
SQL	*Structured Query Language*
TCP	*Transmission Control Protocol*
US	*Ultrasonography*
VR	*Virtual Reality*
XR	*Mixed Reality*

Symbols

Below follows a list of symbols used in the report. Their meanings are also given at first occurrence in the document.

Symbol	**Meaning**	**Unit**
B_0	Static magnetic field of MRI scanner	T
B_1	RF pulse magnetic field of MRI scanner	T
f_0	Larmor frequency	MHz
G_f	Gradient magnetic field coefficient in frequency-encoding direction	$\mathrm{T\,m^{-1}}$
G_p	Gradient magnetic field coefficient in phase-encoding direction	$\mathrm{T\,m^{-1}}$
G_ss	Gradient magnetic field coefficient in slice-selection direction	$\mathrm{T\,m^{-1}}$
$\vec{g}$	Unit vector in direction of gradient field	-
N	Background signal average of MR image	-
n	Number of measurements	-
S	Foreground signal average of MR image	-
T	Data transmission time	s
t	Time	s
$\vec{v}$	Vector containing measurement data	-
x	Axis direction	m
y	Axis direction	m
z	Axis direction	m
γ	Particle-specific reduced gyromagnetic ratio	$\mathrm{MHz\,T^{-1}}$
θ	Angle of scan-plane rotation around the z-axis	-
σ	Standard deviation of a dataset	-
φ	Angle of scan-plane rotation around the y-axis	-
χ	Angle of scan-plane rotation around the x-axis	-

1 Introduction

Imaging techniques such as Ultrasonography (US), Computed Tomography (CT), and Magnetic Resonance Imaging (MRI) can provide detailed visualisation of the human body in 2D and 3D, and nowadays play a key role in diagnostics and treatment planning [2, 3]. Real-time imaging during medical procedures facilitates potential increases in accuracy and time-efficiency [4, 5]. In this context, the mentioned imaging modalities each have unique benefits and disadvantages, which are discussed below.

US imaging is a fast, safe, affordable solution, but can be unreliable. Real-time US imaging has been implemented in, among others, cardiac, neurological, and biopsy surgeries [2, 3, 6]. The necessary equipment can be installed in regular operating rooms. Though effective, US has one significant disadvantage: the image resolution varies depending on the target location, since the sound waves have difficulty passing through dense tissues [7]. US is therefore only reliable at low penetration depths.

Although CT can be more accurate than US in certain diagnostics, it generally performs similarly at higher cost, less accessible set-up and longer procedural and waiting times [8, 9, 10, 11]. Furthermore, CT uses ionising radiation, posing a risk to the medical staff when used in interventional procedures [2].

MRI is relatively slow, but provides better soft-tissue contrast than CT and US, making it a good candidate for interventional use [12]. Additionally, MRI is safer than CT [2, 6, 11]. Real-time MRI has been used in various areas, such as biopsies, radiotherapy, monitoring of heat treatment, various ablation techniques, drug delivery and cardiac, upper airway and musculoskeletal imaging [6, 13, 14]. Figs. 1 and 2 illustrate possible MRI room lay-outs. Surgical procedures performed in the MRI room make use of MR-safe equipment, with real-time imaging results being displayed on a dedicated screen in the room [4]. There are three main disadvantages to the current MRI technologies for image-guided interventions. Firstly, it can be difficult for the surgeon to view the screen properly while reaching inside the narrow MRI bore to perform the procedure. Secondly, images are shown in 2D, whereas 3D imaging would provide more insights [16]. Thirdly, the scan-plane can only be controlled from outside the room, necessitating frequent interruptions which increase the overall procedural time, as well as require the presence of an additional staff member. This project focuses on improving efficiency and ease of use of real-time MR imaging, aiming to ultimately reduce cost and procedural time in the interest of healthcare.

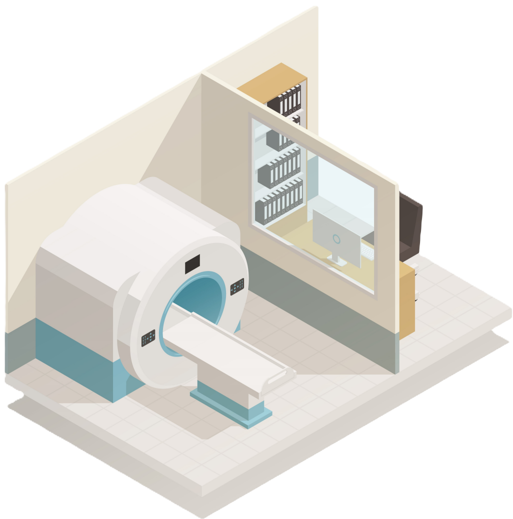


FIGURE 1: MRI room and operational control equipment outside. Figure taken from [15].

FIGURE 2: Example of an MRI-guided intervention. Figure taken from [17].

1.1 Current solutions

The main challenges in real-time MRI are speed, the need for MR-safe equipment, artifacts caused by said equipment, user-friendliness, scan visualisations being limited to 2D-representations, and cost [2, 4, 12, 16]. To fully leverage the potential of MRI as a widely applicable imaging modality, research pertaining to the mitigation of these challenges is ongoing. Current solutions and promising research directions are summarised below.

Speed

MRI acquisition rates have increased significantly in the past years, reportedly going from 5 frames per second (FPS) in 2006 to 50 FPS in 2023 [18, 19]. Image reconstruction is currently possible at 27 FPS [20]. The attainable frame-rates surpass 24 FPS, which is commonly used for cinematic purposes [21]. However, acquisition rate and image resolution of MRI are negatively related [22]. Efforts to decrease acquisition times have invariably been faced with the challenge of achieving image resolutions and signal-to-noise ratios (SNR) similar to those of 'traditional' high-resolution scans [23]. Compromises made between speed and image quality may result in lower frame-rates during MRI-guided interventions.

MR-safe equipment

Due to the magnetic field around the MRI scanner, equipment meant for use in the MRI room must meet additional standards to ensure safety and functionality. Ferromagnetic objects in the room can be attracted by the MRI system, potentially posing a serious hazard [24]. Non-magnetic materials with high conductivity are susceptible to the formation of Eddy currents, which can cause a noticeable torque on the material [25]. Therefore, devices designed to be MR-compatible, especially devices that produce electric currents, need to be tested thoroughly to ensure no unwanted forces or malfunctions occur. The recent development of low-field MRI has eased the constraints on MR-safe equipment due to the drastically weakened magnetic field [13], creating the opportunity to use more varied and sophisticated instruments in the MRI room.

Image artifacts

MR image artifacts are anomalies found on the processed image that do not correspond to the tissue structure [26]. Artifacts have several causes, including motion, heterogeneity in the imaged tissue, scan settings and hardware interference, even when using MR-safe equipment [27]. Often, appropriate changes in the scan settings can reduce unwanted artifacts. In MRI-guided interventions, however, needles and catheters often deliberately create small artifacts for localisation purposes [28, 29, 30].

Image dimension

Access to 3D-portrayed scans can aid in medical decision-making, both pre-operatively and during interventional procedures [31, 32]. However, the 3D information cannot be utilised fully when 2D screens are used to visualise the data [16]. Virtual Reality (VR), Mixed Reality (XR) or Augmented Reality (AR) can be used to portray the obtained scan information comprehensively in 3D. Accurate navigation and replication of the surroundings can be ensured using optical tracking [33].

User-friendliness

As mentioned above in section 1, there are two main concerns in the user-friendliness of an MRI-guided interventional procedure set-up. As Fig. 1 shows, the scan-plane can only be controlled outside the MRI room, creating the need for an extra staff member to operate the scanner. Fig. 2 illustrates the difficulties with patient access that a closed MRI bore presents. The surgeon has to reach into the bore, while looking back at the screen positioned outside. For these reasons, it has been common practice in MRI-guided biopsies and aspirations to position the needle using US-guidance, employing the MRI scanner only for planning and validation of the needle position [5]. Although interchanging US and MR imaging in this manner circumvents challenges related to access and needle-tracking, the need to reposition the patient multiple times during the procedure significantly increases procedural time and the risk of mispositioning the needle. A partial solution is the use of an open MRI bore. These are generally low-field, but nowadays can achieve field strengths up to 1.2 T [29]. A more generally applicable solution, however, is the use of an XR head-mounted device (HMD) that can show the MR data anyplace without impeding the surgeon's ability to look at the patient directly [16].

Cost

MRI scans are becoming more accessible and affordable, but are still quite costly. A notable portion of the expenses goes towards scan supplies and staffing costs [34]. Moving towards low-field MRI can reduce system costs [13]. Use of AR/XR/VR glasses to visualise data can lead to reduced procedural times, and fewer procedures needed per patient due to increased accuracy, thus making the process less expensive.

1.2 XR-guided MRI scan-plane control

Many of the remaining challenges can be solved or improved upon through implementation of an XR-HMD to not only visualise MR data holographically in real-time, but also access the MRI parameter controls. This would allow for viewing the scan at more convenient angles, eliminating the necessity of intermediary US-guidance in biopsies and aspirations. Additionally, the interface would facilitate visualising scan information intuitively in 3D, and make it possible to control the scan-plane directly using hand gestures (Fig. 3). This could potentially reduce procedural time, increase accuracy and hence reduce the number of procedure repetitions required, as well as reduce the number of staff members needed. As a result, MRI-guided interventions would become more effective and cost-efficient.

The use of extended reality in medical contexts comes with several concerns and drawbacks. Firstly, the implementation of the devices can lead to ethical issues. The sensors used by the HMD may record personal and/or confidential information [36]. If the HMD is connected to the MRI scanner via Wi-Fi or similar networks, care must be taken to ensure the security of the transmitted data. Secondly, it has been shown that use of HMDs can induce motion sickness symptoms such as nausea [36]. This could be dangerous to the patient in surgical procedures. It is therefore imperative for future clinical implementation that user tests are performed to evaluate this risk [1]. Thirdly, care needs to be taken to ensure the HMDs do not distort either the presented data or the real-world environment, since this would reduce medical accuracy and potentially make the device unpleasant to use. Proper calibration can mitigate this effect [33]. It is believed that the mentioned drawbacks are not insurmountable. Additionally, the proposed solution does not necessarily replace any medical equipment or procedures. Its use should therefore remain optional.

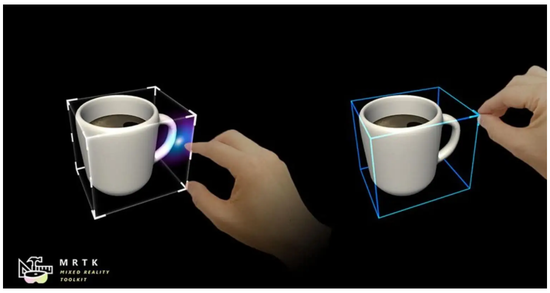


FIGURE 3: Example of grabbing and dragging an XR-hologram using hand gestures. Figure taken from [35].

1.3 Designing an interface between XR and MRI

The aim of this project was to give a proof-of-concept of using interactive XR in the MRI room, and to evaluate its potential in comparison to current practices. The research question was as follows: *"What is the potential to improve user-friendliness and efficiency of MRI-guided interventions through the use of interactive XR-guided visualisation?"*.

To create a functional interface between the MRI scanner and XR-HMD, an XR-application was created to run on an HMD independently, communicate bidirectionally with the MRI scanner via an intermediary PC to receive real-time MR images and adjust the scan-plane position and orientation. The project was divided in four separate phases. Firstly, an XR-application was designed to display MR images in the XR-environment. Secondly, a wireless connection between the XR-HMD and the MRI scanner was established to allow for live-streaming of the MR data. Thirdly, an XR-application was designed to allow for interactive control of the MRI scan settings. Lastly, all subcomponents were integrated into a suitable user interface. Relevant theory related to the mechanics of MRI scan-plane control and the interactions and data infrastructure between MRI and the XR-HMD is discussed in Section 2.

2 Theory

This section provides underlying theory supporting the design decisions and considerations made during the project. Since the focus lay primarily on software design, an in-depth knowledge of the inner workings of the MRI scanner itself is not required. Nevertheless, MRI basics are elaborated upon in section 2.1, to provide the reader with a better understanding of hardware concerns mentioned in section 2.2 and the underlying principles of interaction with the MRI scanner.

2.1 MRI operating principle

MRI scanners use the principle of nuclear magnetic resonance. Atomic nuclei with an odd number of protons inherently possess a magnetic moment with certain magnitude and direction. An example of atoms with this property is hydrogen, which is abundantly present in the human body. In a spin system containing several such particles, the combined magnetic field can be made measurable through external magnetic field variations at the particle-specific resonance frequency, referred to as the Larmor frequency. A more detailed explanation can be found in [37]. Relevant components of the MRI scan system, the relation between magnetic field variation and scan-plane orientation, and MRI coordinate transformations are discussed in Sections 2.1.1, 2.1.2 and 2.1.3, respectively.

2.1.1 MRI scanner components

Three main components of an MRI scanner are a magnet, gradient coils and radiofrequency (RF) coils. The magnet creates a homogeneous magnetic field B_0 inside the MRI bore, traditionally with magnitudes ranging from 0.5 T to 3.0 T [22]. There are three gradient coils, one for each direction (x, y, z), which are used to create small variations in the magnetic field along their respective axes. This allows for spatial localisation of the image information, as discussed in section 2.1.2. RF coils facilitate the actual measurement. Transmit RF coils send out a short magnetic pulse B_1 around the Larmor frequency. This pulse 'pushes' the nuclear magnetisation direction towards the transverse plane (orthogonal to the longitudinal axis), where its magnetic field can be measured by receive RF coils. Subsequently, the magnetisation returns to equilibrium while precessing about the longitudinal axis. An illustration of this process can be found in Fig. 4. [37, 38]

The time needed for the transverse component of the magnetic moment to return to zero and the longitudinal component to return to its original value is dependent on the imaged tissue. A measurement taken shortly after the excitation pulse therefore shows different measured magnetic field intensity levels, corresponding to anatomical structures. How the raw scan data is reconstructed to form a clear image and how scan settings can be optimised is not relevant to this project, but the interested reader may find a comprehensive explanation in [37].

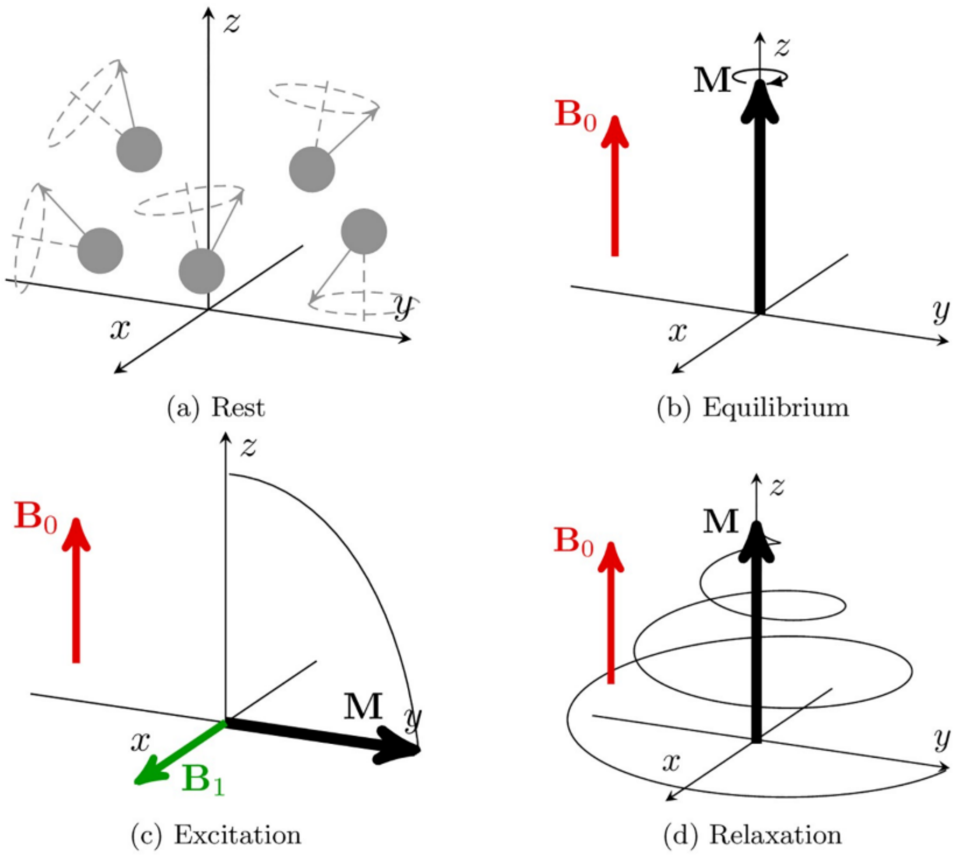


FIGURE 4: Visualisation of Larmor precession after RF excitation (B_1). It can be seen that the transverse magnitude decreases and the longitudinal magnitude increases as the magnetic moment realigns itself with the static magnetic field (in vertical direction). Figure taken from [39].

2.1.2 Scan-plane control

The gradient coils provide spatial encoding, which allows the scanner to determine the spatial origin of measured signals [40]. Spatial encoding facilitates scanning only select parts (slices or slabs) of the anatomy, as well as resolving in-plane contributions to the signal. In a 2D imaging experiment, which is typically used in real-time applications, a scan-plane or slice is selected in three steps. These are facilitated by short magnetic field pulses of the gradient coils: slice selection, phase encoding and frequency encoding. It must be noted that these steps are completely independent of the gradient directions x, y and z, but rather correspond to selecting the orientation and location of the plane, distinguishing the signal in 'horizontal' direction, and distinguishing the signal in 'vertical' direction. Fig. 5 illustrates the spatial encoding steps. In-depth explanations can be found in [37, 40].

2.1.3 MRI coordinate systems and transformations

Several different coordinate systems and transformations are in use during MR image acquisition. The xyz-coordinate system used in this report is fixed to the scanner and defined as shown in Fig. 6. The B_0 field of the scanner is aligned with the z-axis. To set a certain scan-plane orientation, the MRI scanner needs three orthogonal unit vectors corresponding to the slice selection, frequency encoding and phase encoding directions of the image. To convert an arbitrary rotation of the scan-plane to the appropriate vector inputs, the transformation matrix shown in Eq. (1) suffices [41].

$$R_{\chi,\varphi,\theta} = \begin{pmatrix} c(\varphi)c(\theta) & c(\chi)s(\theta) + s(\chi)s(\varphi)c(\theta) & s(\chi)s(\theta) - c(\chi)s(\varphi)c(\theta) \\ -c(\varphi)s(\theta) & c(\chi)c(\theta) - s(\chi)s(\varphi)s(\theta) & s(\chi)c(\theta) + c(\chi)s(\varphi)s(\theta) \\ s(\varphi) & -s(\chi)c(\varphi) & c(\chi)c(\varphi) \end{pmatrix} \quad (1)$$

Where $c() = \cos()$, $s() = \sin()$, *and* χ, φ, θ *are the rotations around the* x, y *and* z*-axis, respectively.*

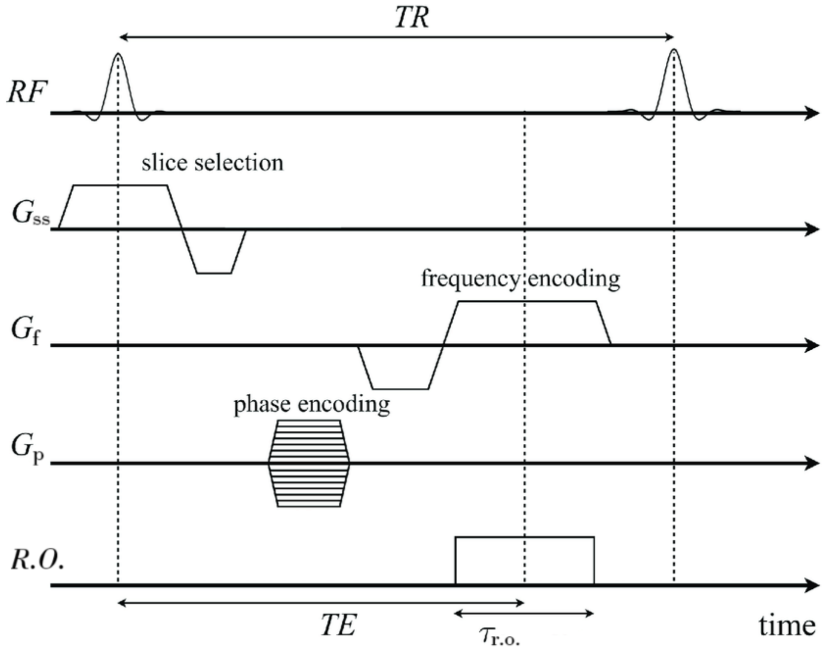


FIGURE 5: Diagram showing the spatial encoding steps for a 2D slice. 'r.o.' stands for 'read-out'. The phase differences induced by the slice selection are negated by an added negative pulse in G_{ss}-direction. Data is acquired during $\tau_{r.o.}$. At $t = TE$, the phase differences along the frequency-axis are zero, which leads to a peak in the received signal termed 'gradient recalled echo' Figure modified from [39].

FIGURE 6: Device-based coordinate system. Figure adapted from [42].

Hence, the direction of the slice selection, frequency and phase gradient fields after arbitrary rotation of the scan-plane can be easily determined. Equation (2) shows the final transformation step. The relative field strengths of gradient fields G_x, G_y and G_z are constructed by the MRI scanner using direction vector $\vec{g}$.

$$\vec{g}_{\text{new}} = R_{\chi,\varphi,\theta} \cdot \vec{g}_{\text{old}} \tag{2}$$

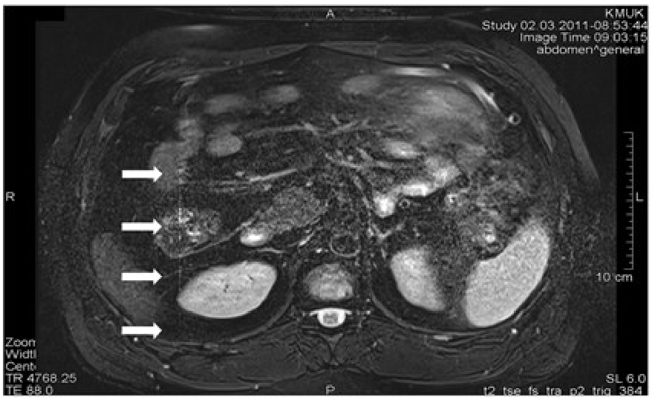
With $\vec{g}$ the unit vector in a given encoding direction and $R_{\chi,\varphi,\theta}$ the rotation matrix given in Eq. (1).

2.2 Hardware interactions

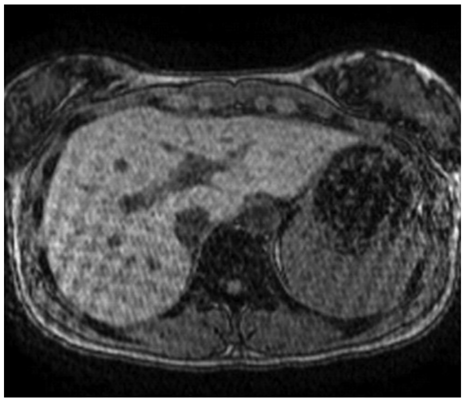
Since the XR-HMD is an electronic device intended for use in close proximity to the MRI scanner, possible unwanted interactions between the HMD and scanner must be taken into account. The concerns are bidirectional; the magnetic field of the scanner could influence the performance of the HMD, and the magnetic/electronic fields generated by the HMD could cause image artifacts on the scans [24, 25, 27]. These possibilities are elaborated upon in Sections 2.2.1 and 2.2.2.

2.2.1 Image artifacts

Given that MR image reconstruction depends on interpreting measured magnetic fields as a product of precisely induced field gradients, electromagnetic interference emitted by the HMD might cause degradation in image quality. Besides lowering the SNR of the image, two specific types of image artifacts might occur: zipper artifact (Fig. 7a) and herringbone or corduroy artifact (Fig. 7b) [27, 43]. Zipper artifacts tend to be caused by RF signals from the device at frequencies approaching the Larmor frequency of the imaged tissue. Herringbone artifacts can occur in a wider range of frequencies.



(A) Zipper artifact example. Figure taken from [44].



(B) Herringbone artifact example. Figure taken from [43].

FIGURE 7: Image artifacts caused by RF interference.

2.2.2 Attraction force and eddy currents

The magnetic field of the MRI scanner could have unwanted and even dangerous effects on the HMD. First and foremost, ferromagnetic components in the HMD are attracted by the static magnetic field B_0 [24]. If the HMD is not properly secured to the wearer or taken off and placed too close to the magnet bore, it could turn into a projectile. This could cause injuries to patients and staff, as well as damage the MRI system or HMD [45]. Alternatively, the attraction force or induced torque could unbalance the wearer, potentially causing them to err during precision procedures. Secondly, the gradient and RF magnetic fields might induce voltages and eddy currents in the HMD [25]. As a result, the device might heat up or malfunction. An overview of safety tests performed to rule out the aforementioned risks can be found in Section 3.5.1.

2.3 Data transmission

Data transmission between the Hololens and MRI scanner is a vital part of the device functionality. To maintain the intended ease of use and avoid bringing more electronics in the MRI room than strictly necessary, it was opted to use wireless data transmission rather than a cable connection. Two different types of data transmission used are elaborated upon in Sections 2.3.1 and 2.3.2. Multithreading is invaluable in the implementation of these methods, and is discussed in Section 2.3.3.

2.3.1 Sockets

The Transmission Control Protocol / Internet Protocol (TCP/IP) is ubiquitously used for wireless connections between devices and networks [46]. It preserves message order and content reliably, and is used by networking libraries such as ∅MQ and Windows Sockets [47, 48]. These libraries provide straightforward methods to connect and send data via sockets. When using ∅MQ with TCP/IP, sockets need only the IP address of the host device and a specified port to establish a connection [47]. For this project, the most appealing type of connection is to have the sockets send and receive independently without waiting for a reply, and to have the option to connect more than one receiver. This type of connection is provided by the ∅MQ library as the 'publisher-subscriber' pattern, an illustration of which is given in Fig. 8. The publisher socket sends data regardless of whether or not a subscriber is connected (though not infinitely), and subscriber sockets can connect and disconnect without interrupting the publisher.

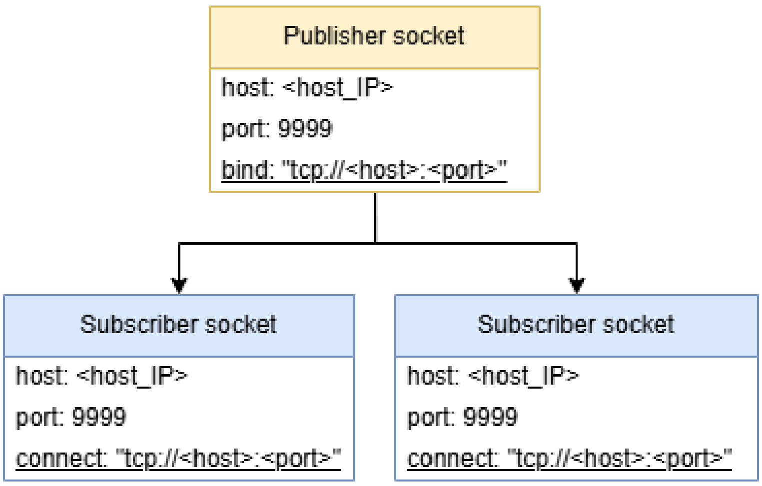


FIGURE 8: Example of a pub-sub connection pattern with two independent subscribers.

2.3.2 Databases

An alternative method to send data between devices is the use of online databases. Data is structured in tables and can be accessed by remote devices, provided their IP addresses are registered in the firewall of the database server. Structured Query Language (SQL) databases provide an excellent framework for scalable data organisation [49]. Whereas sockets are limited to sending only one or a few data strings simultaneously and are susceptible to queuing delays, SQL databases allow for targeted access of specific data components independently and asynchronously and have a higher base level of security [47, 49]. Conversely, connecting to the database might cause system delays in hardware with limited processing power.

2.3.3 Multithreading

Multithreading is essential to developing code that exchanges data with an external server or device. If a device has multiple central processing units (CPUs) at its disposal, code in separate threads can be executed simultaneously on different CPUs. If only a single CPU is available, an approximation of simultaneous code execution is realised by frequently switching between threads. All threads have real-time access to current system data and variables. Since creating and switching threads is costly and the resulting desynchronisation of processes can be unpredictable, multithreading should be implemented cautiously. Nevertheless, it is invaluable for incorporating arduous tasks or awaiting input. Executing tasks like these in a separate thread allows other code elements, such as the user interface, to remain responsive. [50]

3 Method

In this section, an overview of the design and assessment methodology is given. The first step of the project was to define hardware needed to create a functional XR-interface and communication with the MRI scanner. Secondly, software was chosen to develop the application and run relevant scripts. Section 3.1 describes choices made herein. An overview of the established hardware interactions and different system configurations is given in Section 3.2, and discussed in-depth in Sections 3.3 and 3.4. Section 3.5 describes how the XR-application performance and potential were evaluated.

3.1 Equipment and software

The necessary hardware components for the project were an MRI scanner, an HMD, and an intermediary desktop personal computer (PC) to manage the more laborious data processing aspects. In addition, a laptop (running Windows 11 Home, with Intel(R) Core(TM) i7-10750H processor, CPU @ 2.60 GHz and 16.0 GB random-access memory (RAM)) was used to develop the XR-application for the HMD and interface with the desktop PC. The available MRI scanner at the University of Twente is the Siemens MAGNETOM Aera [42]. The paired desktop PC runs Windows 10 Enterprise, with Intel(R) Core(TM) i9-9900K processor, CPU @ 3.60 GHz and 64.0 GB RAM. Communication and data transmission between the PC and MRI scanner was managed by the Access-i MR Scanner Interface provided by Siemens, through a dedicated Python library [51]. The Access-i framework allows third parties to connect to, control, and retrieve data from the MRI scanner remotely.

In the process of choosing a suitable HMD, both AR and XR devices were considered. The following criteria were taken into account for the feasibility comparison: whether the HMD supported XR or AR, price, whether native pass-through of the surroundings was possible, whether the HMD supported gesture input, whether the HMD supported voice input, battery life, whether the HMD could function wirelessly, whether the HMD could function as a stand-alone device, whether Wi-Fi connections were possible, and whether Bluetooth connections were possible. A full comparison overview is given in Appendix A.1. In addition to being readily available at the University of Twente, the Microsoft Hololens gen 1, henceforth referred to as the Hololens, was the only XR-HMD that met all inclusion criteria. The relatively unique combination of pass-through lenses and 3D holographic control provided by the Hololens has made it the top choice for a multitude of related studies and applications [1, 16, 52]. Therefore, this HMD was chosen to develop and test the XR-application.

Software and version choices were in part dictated by the chosen HMD. A major disadvantage of using the Hololens is the current lack of support from established XR development platforms such as Unreal Engine and Unity [53, 54]. Therefore, it was chosen to develop the XR-application on the laptop using Unity version 2019.4.40f1 and edit Unity scripts using Visual Studio Community 2017, rather than working with the newest software versions. On the desktop PC, PyCharm Community 2024 was used to run the intermediary processing script.

3.2 Overview of system connections

The data transmission methods and hardware connections used enable easy modification and adjustment between system configurations. Fig. 9 shows the relation between the different components. The XR-application is deployed from the laptop to the HMD once. Subsequently, the HMD can run the application independently. When connected to the smart devices Wi-Fi network hosted by the University of Twente, the HMD can exchange data with the desktop PC (connected to the University of Twente eduroam network via ethernet) from within the MRI room. The MRI scanner can only communicate with the desktop PC, which is is controlled by the laptop via remote desktop connection.

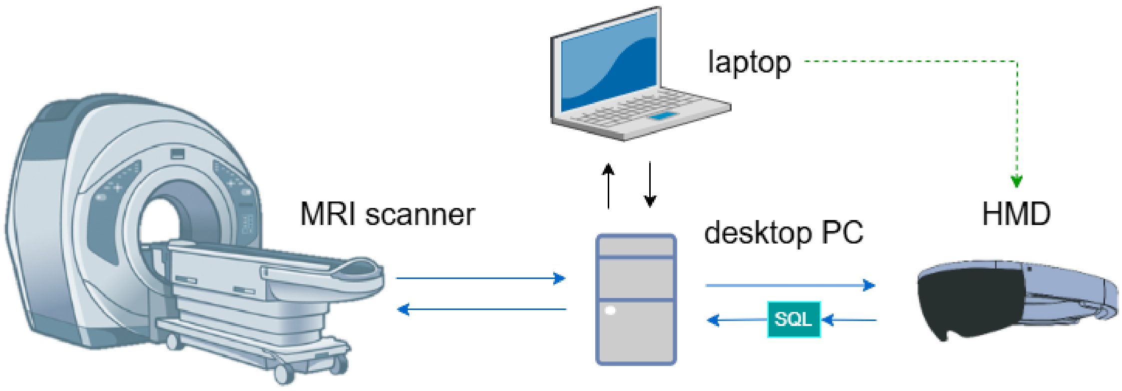


FIGURE 9: Illustration of the communication between the hardware components. Black arrows: remote control of the desktop PC via the laptop. Blue arrows: data exchange. Green arrow: XR application deployment from the laptop to the HMD, done only when changes to the application were made. MRI icon taken from [55]. HMD icon modified from [56].

In the rest of the report, the intermediary processing script facilitating the connection between the MRI scanner and the Hololens is referred to as 'Python code', and the developed XR-application is referred to as 'XR-application' or 'Unity application'. A bidirectional communication between the Python code and Unity application is implied. The different system configurations used during development are listed below.

- *MRI simulator - Unity Editor:* An MRI simulator application provided by Siemens is used to simulate the Access-i connection to the physical MRI scanner. The simulator and Python code run either on the desktop PC or on the laptop. The Unity application runs on the laptop.
- *MRI simulator - Hololens:* The MRI simulator and Python code run either on the desktop PC or on the laptop. The Unity application runs on the Hololens.
- *MRI scanner - Unity Editor:* The MRI scanner is controlled by the Python code running on the desktop PC. The Unity application runs on the laptop.
- *MRI scanner - Hololens:* The MRI scanner is controlled by the Python code running on the desktop PC. The Unity application runs on the Hololens. This configuration represents the intended use of the XR-application.

Both the Unity application and Python code can be started independently and connect to each other at a later point during code execution. To demonstrate the level of control possible from within the holographic environment and ensure images are not acquired unnecessarily after a test, disconnecting the Unity application also stops the current scan.

3.3 Unity application: holographic environment

The holographic scene was kept as simple as possible to allow for intuitive user interaction, and consists of three main components: a modifiable visualisation of the scan-plane position and orientation, a panel showing real-time MRI scans, and a small menu panel containing general settings (Fig. 10). On start-up, an information panel is shown with an explanation on how to use the environment. The application was developed as a Universal Windows Platform app, using Windows Mixed Reality and the Microsoft Mixed Reality Toolkit to ensure the application was compatible with the Hololens and hand-tracking inputs were interpreted correctly. For a complete overview of the installation and set-up of the hardware and software used, please refer to Appendix A.2.

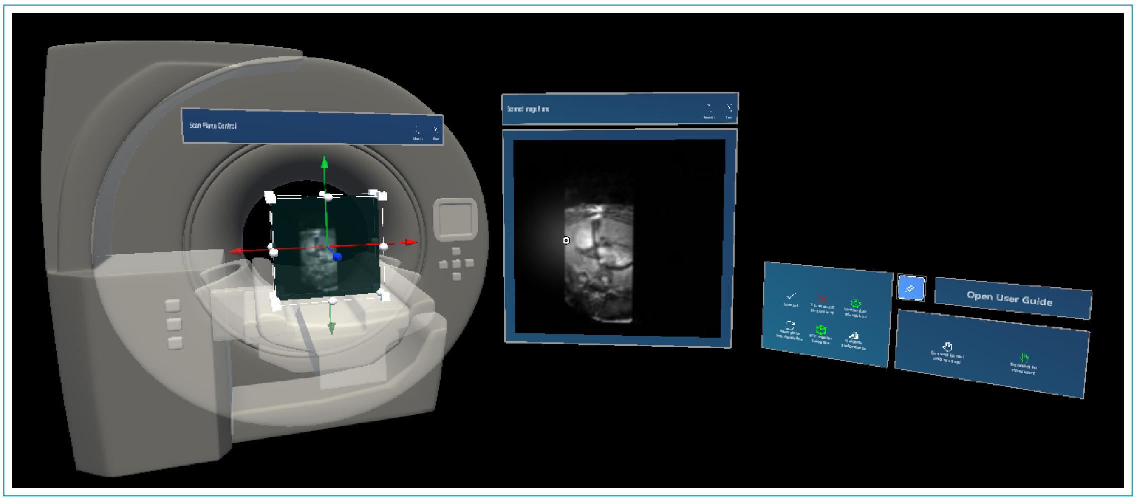


FIGURE 10: Standard lay-out of the holographic environment. Left to right: Scan Plane Control, Image Display and Menu Panel holograms.

The Menu Panel shown in Figs. 10 and 11 coordinates the main functionality of the application. It allows the user to establish a connection to the MRI scanner and to customise the scene elements. A list of the menu buttons and their functionality is given in Table 1. To receive data, the Unity application uses the NetMQ library, a C# compatible variant of ∅MQ. In a separate thread, a subscriber socket awaits the arrival of an image data string from the paired Python publisher socket (JPEG encoded as base64 string). The application then converts the image to a Sprite and displays it in the holographic scene.

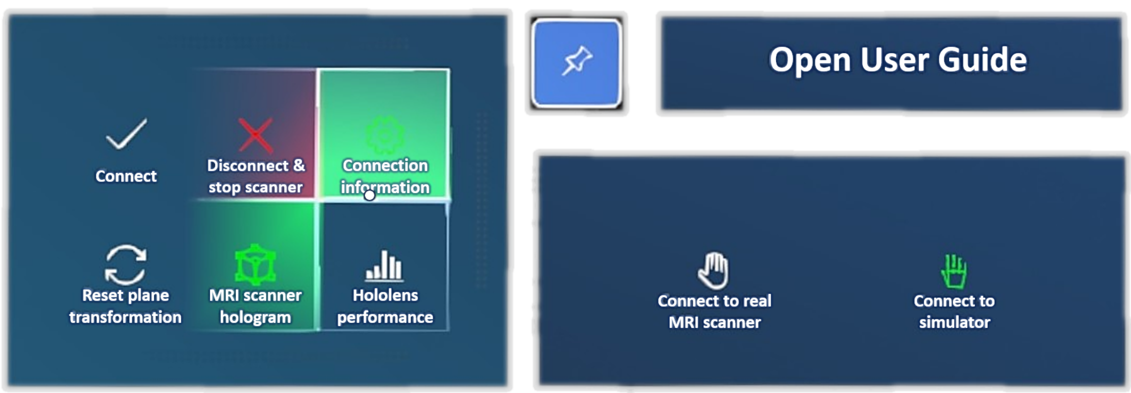


FIGURE 11: Detailed overview of the Menu Panel in the holographic environment.

Table 1: Menu Panel options in the holographic environment.

Label	Type	Function
Connect	Button	Connect the database and subscriber socket to exchange data with the Python interface
Disconnect & stop scan	Button	Signal the MRI scanner to stop the measurement and disconnect
Reset plane transformation	Button	Reset the Scan Plane Control image plane (and, consequently, the scan-plane orientation settings of the MRI scanner) to the initial position and orientation
Connection information	Toggle	Show/hide socket information: host IP address, port and whether or not the connection is active
MRI scanner hologram	Toggle	Show/hide MRI scanner hologram
Hololens performance	Toggle	Show/hide CPU and memory status of the Hololens
Open User Guide	Toggle	Show/hide the User Guide panel
Connect to MRI scanner	Button	Set subscriber socket IP address to that of the desktop PC
Connect to MRI simulator	Button	Set subscriber socket IP address to that of a third party laptop (pre-set IP address)

Data is not sent directly from the Unity application to the Python code. An intermediary SQL database table is used, with format as shown in Table 2. Since connecting and sending information to the database is a time-consuming operation, this code runs in a background thread as well. If the user changes the position and/or orientation of the Scan Plane Control cube shown on the left in Fig. 10, the new parameters are converted to a string and sent to the database. This replaces the previous command string in the 'entryCommand' column. Unity works with a left-handed coordinate system. To conform to the coordinate system used by the MRI scanner, the z-position and rotation around the z-axis are inverted. The position and orientation of the image shown inside the Scan Plane Control cube only updates when new images arrive.

Table 2: SQL database table used to store the most recent data sent from the Unity application.

entryNumber	entryCommand	imageDiagnostics
1	"latest command"	"latest image timestamps"

When an image is received by the subscriber socket, the associated timestamps are extracted from the data string and saved as intermediate latency data. Timestamps of relevant image processing steps in Unity are appended to this new string. Subsequently, the updated string is sent to the SQL database to replace the previous string in the 'imageDiagnostics' column. Activity diagrams related to the Unity data infrastructure can be found in Figs 28, 29 and 30 in Appendix B.

3.4 Python code: data conversion and infrastructure

The Python code coordinates the data flow between the Unity application and MRI scanner. It controls the settings of the MRI scanner, receives MR images, and exchanges data with the Unity code. The code is structured using the Mediator principle: all communication and data transmission between classes is regulated in main.py, to avoid convoluted interconnections and keep the code more modular in nature [57]. A list of the used classes and their main function is given in Table 3. A sequence diagram depicting the communication between the Python code and MRI scanner is shown in Fig. 12.

With regard to the different system configurations described in Section 3.2, the SocketInfo class allows the user to set connection IP addresses and ports for socket communication, and server information for the connection to the SQL database. In main.py, the user can specify whether the Access-i library should connect to the physical MRI scanner or the MRI simulator, and whether data is received via a subscriber socket in the SubscriberFromUnity class or retrieved from the database in the SubscriberFromDatabase class. Upon starting, the ConnectToScanner class runs a series of tests to establish whether the Access-i library is able to establish a fully functional connection with the scanner or simulator.

Table 3: Overview of the Python classes, the external systems or devices they connect to, and their function. Except for the SystemDiagnostics class, all intercommunication is managed by main.py. Blue cells: this class operates in a background thread.

Class	Connects to	Main function
SocketInfo	-	Save IP, port and database server info
ConnectToScanner	MRI scanner	Establish a connection to and control over the MRI scanner via Access-i, ascertain that all relevant operations function correctly
ManageImageAcquisition	MRI scanner	Start or stop image acquisition, disconnect from MRI scanner
PublisherToUnity	Unity application	Process and send images to the Unity application via a publisher socket
SubscriberFromDatabase	SQL database	Retrieve position/orientation changes and image latency data from SQL database
SubscriberFromUnity	Unity application	Receive position/orientation changes and image latency data via a subscriber socket
CommandHandler	-	Convert position/orientation changes to appropriate Access-i inputs and change MRI scan parameters accordingly
SystemDiagnostics	-	Write image and command latency data to .txt files for future processing. This class is called from SubscriberFromDatabase/SubscriberFromUnity and CommandHandler instead of main.py

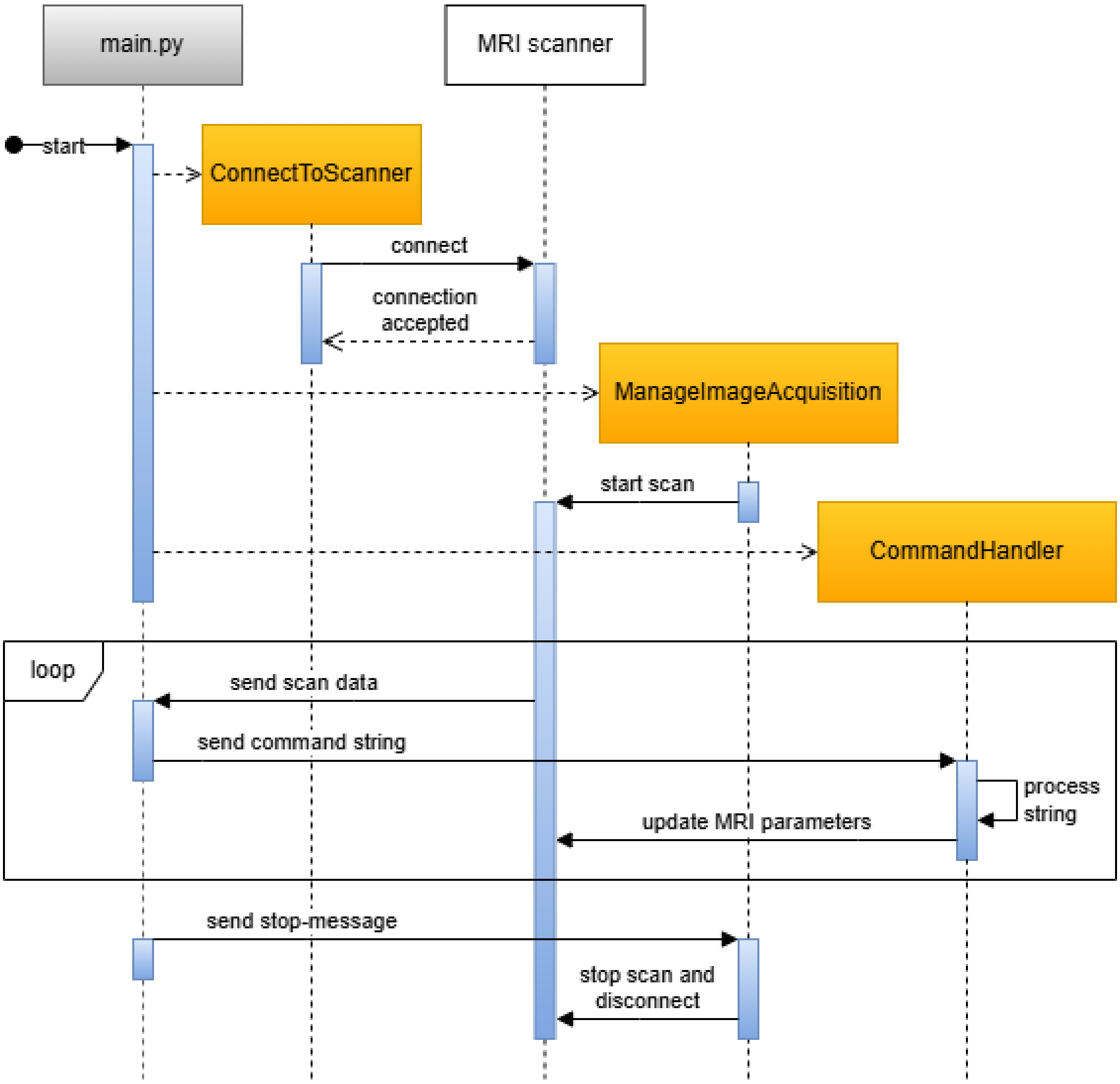


FIGURE 12: Sequence diagram depicting the interaction between the Python code and MRI scanner.

Image data conversion and transmission

When the MRI scanner is properly connected, a scan is started by the ManageImageAcquisition class, and the MRI scanner sends continuous real-time Digital Imaging and Communications in Medicine (DICOM) images to the Python code for processing. The data string is first converted to an accessible format. Next, the image and metadata are extracted and the image is converted to JPEG format base64 string, which is more easily processed by the Unity application than the original format. Finally, the image is checked one last time for validity and sent to the Unity application via a ∅MQ publisher socket in the PublisherToUnity class. The image data flow is shown in the activity diagram in Fig. 13.

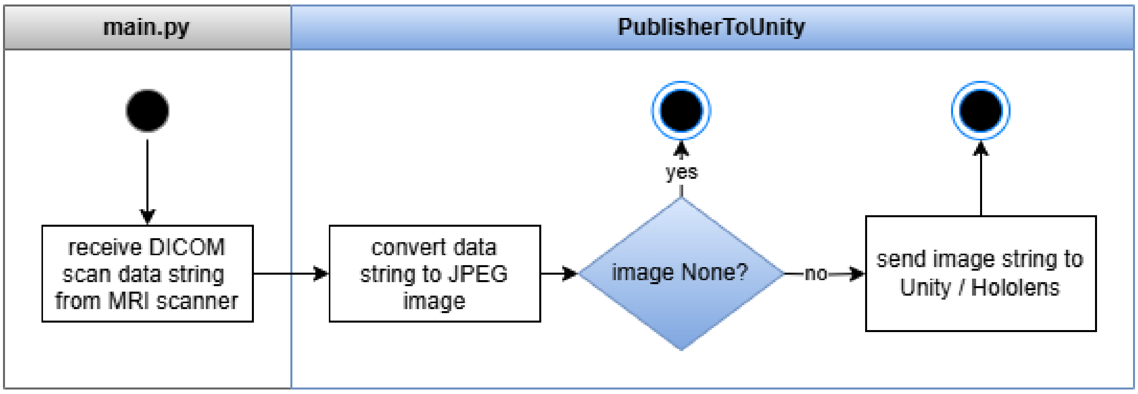


FIGURE 13: Activity diagram depicting the Python part of the image data flow from MRI scanner to Unity code.

Command data conversion and transmission

To receive and store incoming commands from the Unity application, Python runs the SubscriberFromUnity or SubscriberFromDatabase class in a separate thread. SubscriberFromUnity connects a subscriber socket to a publisher socket hosted by the Unity application. For a given timeframe, the code checks whether new messages have arrived in the queue. Accordingly, it determines whether the message is an MRI parameter change command or a diagnostic message. SubscriberFromDatabase connects to the SQL database and continuously retrieves the latest command and diagnostic message from the corresponding columns depicted in Table 2. In both classes, commands are saved as a class attribute and diagnostic messages are sent to the SystemDiagnostics class to be saved in a .txt file. An activity diagram of data retrieval in the SubscriberFromDatabase class is shown in Fig. 14. An activity diagram of the SubscriberFromUnity class is shown in Fig. 31 in Appendix B.

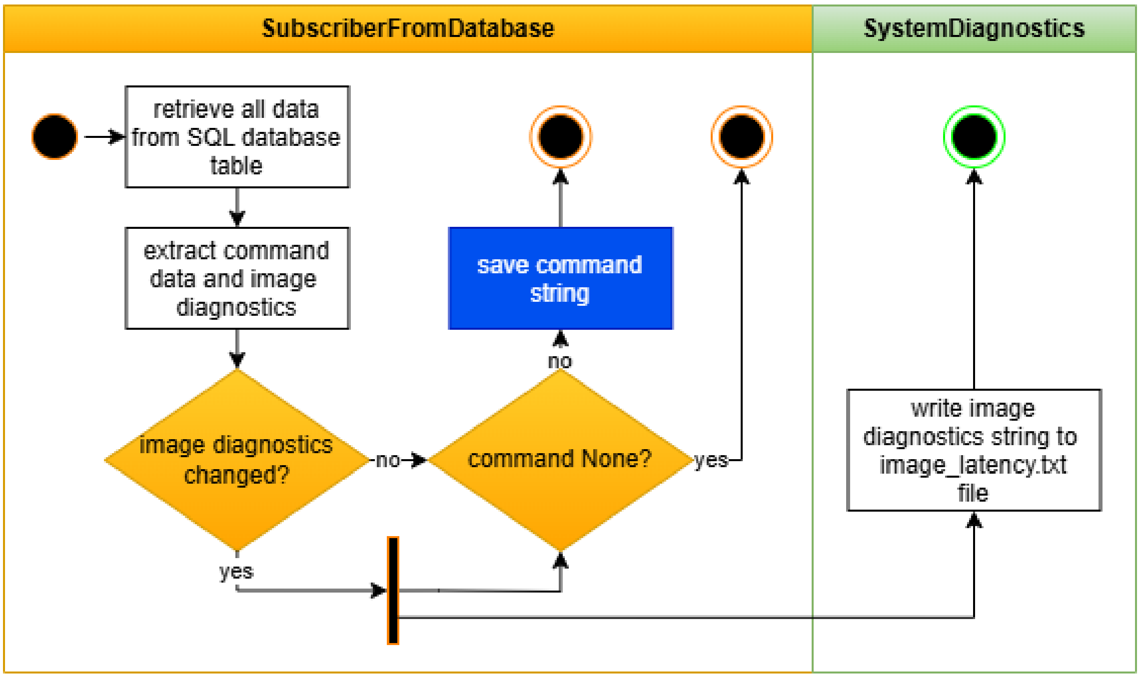


FIGURE 14: Activity diagram depicting reception of commands from the Unity application in the SubscriberFromDatabase class.

Whenever an image is sent from the MRI scanner, main.py retrieves the latest saved command from the SubscriberFromUnity or SubscriberFromDatabase class. If the command is not *"stop_measurement"*, it is sent to the CommandHandler class, which modifies the MRI parameters accordingly. For position changes, data processing is straightforward: the x, y, z parameters sent by the Unity code correspond to the MRI x, y, z system, albeit scaled up. Orientation changes are sent by the Unity application as Euler angles around the coordinate system axes. To convert these to a set of orthogonal direction vectors, Eq. (2) is used. An activity diagram illustrating further processing of command signals is shown in Fig. 32 in Appendix B.

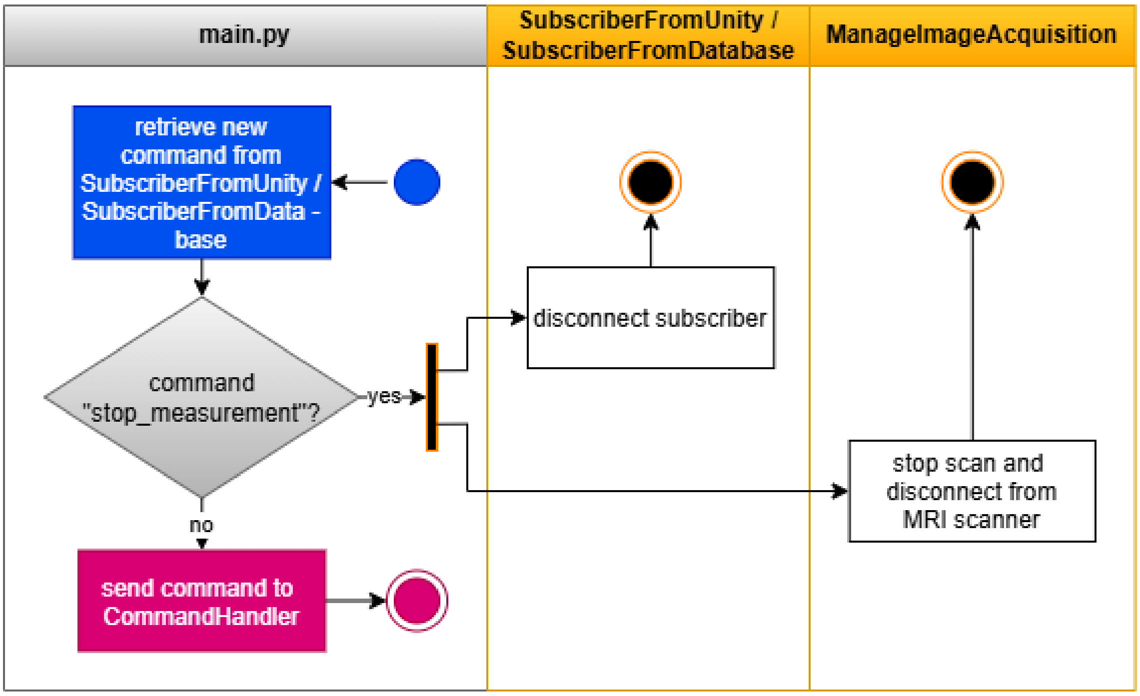


FIGURE 15: Activity diagram depicting retrieval of commands by main and subsequent actions.

3.5 Evaluation

Since the project was a proof-of-concept, no strict design constraints applied and the most important results pertained to the perceived future potential of the application. Nevertheless, data was gathered to evaluate several key aspects of the device functionality: MR-compatibility (Section 3.5.1) and data transmission latency (Section 3.5.2). To answer the research question, a user test was organised and the experiences of the participants were recorded by means of a brief questionnaire (Section 3.5.3). Data processing was executed in Matrix Laboratory (MATLAB). Averages and standard deviations were calculated as shown in Eq. (3).

$$\bar{v} = \frac{1}{n}\sum_{i=1}^{n} v_i, \qquad \sigma = \sqrt{\frac{1}{n-1}\sum_{j=1}^{n} |v_j - \bar{v}|^2} \tag{3}$$

With $\bar{v}$ the mean of data vector $\vec{v}$, n the number of measurements (elements in $\vec{v}$) and σ the standard deviation.

3.5.1 MR-compatibility of the Hololens

To address concerns raised in Section 2.2, an experiment was conducted to assess the MR-safety of the Hololens. The experiment was comprised of three stages, to assess potential hardware interactions in descending order of severity.

1. Magnetic attraction force. During this experiment, the closest safe distance to the MRI scanner was determined. Since initial tests with a small magnet and the Hololens had indicated some ferromagnetic interactions with the device, it was expected that magnetic attraction would occur in the MRI room. The Hololens was brought into the room and moved slowly towards the scanner along the z-axis (as defined in Fig. 6). No images were acquired during this test, and the Hololens was turned off. The device was held loosely by the headband. The location at which the suspension angle shifted was recorded and the corresponding static magnetic field strength was noted.

2. Electromagnetic interference in the Hololens. The Hololens performance was evaluated first with the MRI scanner inactive, later while running a series of scans. While moving the device between field gradients, a series of standard actions was performed. Navigation of the device main menu and settings menu was used to assess any differences in performance and behaviour for the actions described in Table 4. Since the inertial measurement unit of the device has a magnetometer, it was hypothesised that some functionality regarding hologram placement and spatial mapping might be compromised by the magnetic field of the MRI scanner [52].

3. Electromagnetic interference in the MRI scanner. With the Hololens operational in the MRI room, a series of phantom scans was made. The scans were inspected for image artifacts consistent with RF interference, and the SNR was calculated according to one of the National Manufacturers Electrical Association standard methods shown in Eq. (4) [58]. An overview of the main scan settings used during image acquisition can be found in Appendix D.1.

$$\text{SNR} = \frac{\bar{S}}{\sqrt{\sigma_N}} \tag{4}$$

With $\bar{S}$ the signal mean inside the phantom and σ_N the standard deviation of the background signal.

Data transmission via Wi-Fi was not expected to cause unwanted effects. The larmor frequency was calculated according to Eq. (5). For the commonly imaged hydrogen nucleus, the gyromagnetic ratio γ equals 42.6 MHz T^{-1} [59]. With a static magnetic field strength of 1.5 T, it follows that the Larmor frequency of hydrogen equals 63.9 MHz [42]. Standard Wi-Fi communication frequencies are in the GHz range, and should therefore not cause image degradation [60]. Nevertheless, other Hololens system processes or induced eddy currents might still have an impact on the image quality.

$$f_0 = \gamma B_0 \tag{5}$$

With f_0 the Larmor frequency, γ the reduced gyromagnetic ratio of the imaged particles and B_0 the applied magnetic field.

3.5.2 Latency

System latency at different data transmission steps was measured during operation of the four different hardware configurations defined in Section 3.2. The latency results were categorised corresponding to the process of receiving images within either the Unity Editor or the Hololens, and sending commands to the MRI scanner or simulator. Timestamps of certain reference points during data transmission were added to the relevant data strings, to be extracted later by the Python code and saved to .txt files. The measured timepoints are specified in Section 4.3.

3.5.3 User-friendliness and potential

To evaluate user-friendliness and usefulness, seven participants were asked to test the XR-application. All had prior experience in using the MRI scanner or doing MRI-related research. Afterwards, the participants were asked to fill in a questionnaire regarding their opinions on the system. The questionnaire was based mainly on the Post-Study System Usability Questionnaire developed by [61]. Some changes were made to incorporate a section with focus on usefulness, following the vision of the Perceived Usefulness and Ease of Use survey [62]. Appendix C shows the complete questionnaire, denoting the source of each question and whether or not it was changed significantly. Except for an open question at the end, all questions were phrased as positive statements about the system functionality and potential, with the following multiple choice options: 'Strongly disagree', 'Disagree', 'Neutral', 'Agree', 'Strongly agree'. These answers were interpreted as scores between 0.00 and 1.00, and divided into six different categories: ease of use, usefulness, information quality, interface quality, overall satisfaction and potential [61, 62]. Results are shown in Section 4.4.

4 Results

This section provides an overview of the final application design and relevant test results. The performance evaluation of the XR-application was conducted according to the methods described in Section 3.5. Firstly, the MR-safety of the Hololens was tested. Secondly, the latency of data transmission in different system configurations was recorded. Thirdly, user tests were performed. More extensive results can be found in Appendix D.

4.1 Overall design and use

To document the interactivity and usage of the XR-application, pictures were taken during the user tests. Fig. 16 shows how the MRI room was set up during the demonstrations. Real-time images of a phantom were acquired during tests, and displayed on a screen inside the MRI room for comparison with the holographic environment observed on the Hololens (shown in Fig. 17).

Figure 16: Set-up of the MRI room during the user tests.

The user has the option to hide several parts of the holographic scene, such as a holographic representation of the MRI scanner and the User Guide panel. Additionally, the user can place the holographic components at any position and at various orientations in the MRI room, depending on personal preference. By using buttons on the Menu Panel, the connection target can be set to either the MRI simulator or physical MRI scanner, after which a live connection can be made. Subsequently, changes can be made to the position and orientation of the Scan Plane Control cube shown in Figs. 18a and 18b. More detailed information on the Menu Panel and Scan Plane Control holograms is given in Section 3.3.

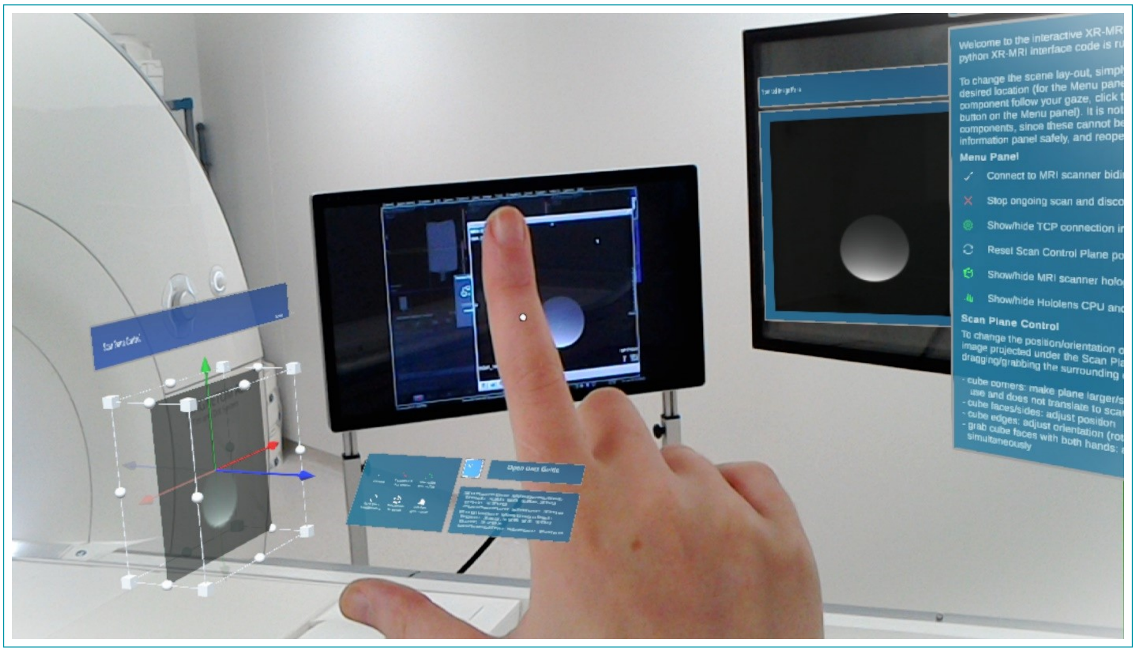
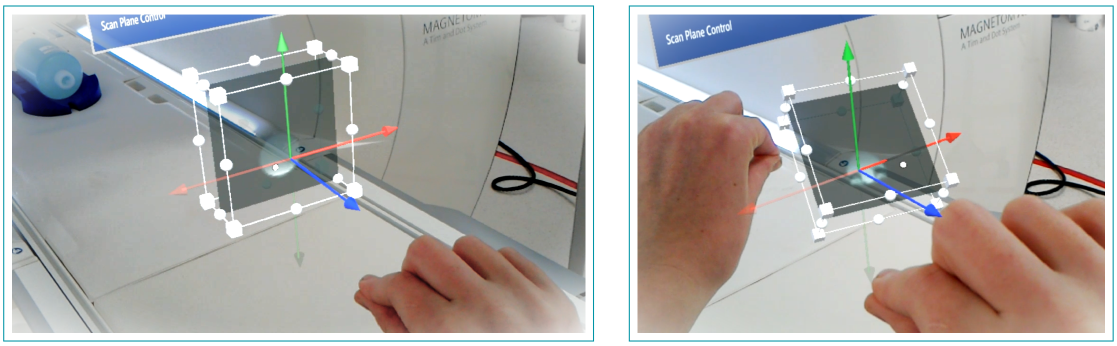


FIGURE 17: Example lay-out of the holographic scene in the MRI room, as seen by the wearer of the Hololens.

All changes are made using the 'air tap and hold' gesture. By looking at different parts of the cube, the user can selectively change position or orientation using one hand. Making the gesture with two hands, as illustrated in Fig. 18b, allows the user to change position and orientation simultaneously. The XR-application continuously sends the current position and orientation to the SQL database via Wi-Fi, from where it is retrieved by the Python code and sent to the MRI scanner. Fig. 19 shows a side-by-side overview of the physical and holographic image displays in the MRI room after a scan-plane change.



(A) Scan Plane Control hologram showing real-time MR images of the transverse plane.

(B) Illustration of changing the scan-plane using hand gestures.

FIGURE 18: Example of changing the scan-plane position and orientation simultaneously, using the air tap and hold gesture with two hands. Initial position and imaged phantom shown in Fig. 18a, final position shown in Fig. 18b.

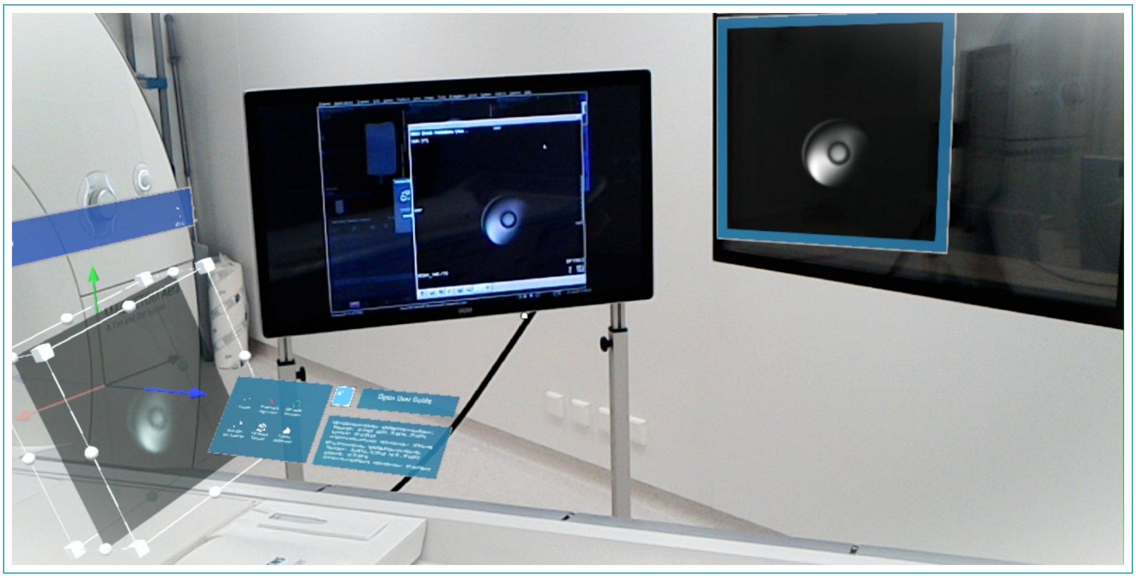


FIGURE 19: Real-time MR images displayed on the Scan Plane Control hologram (left), physical screen (middle) and Image Display hologram (right).

4.2 MR-compatibility of the Hololens

A noticeable magnetic attraction force was detected at distances smaller than approximately 60 cm from the MRI bore edge along the z-axis. These distances correspond to a magnetic field strength greater than 40 mT. The attraction limit is indicated in Fig. 20. No noticeable heating of device parts occurred.

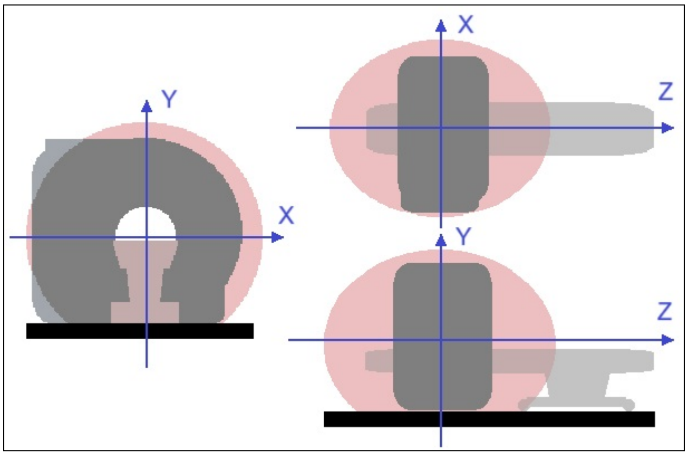


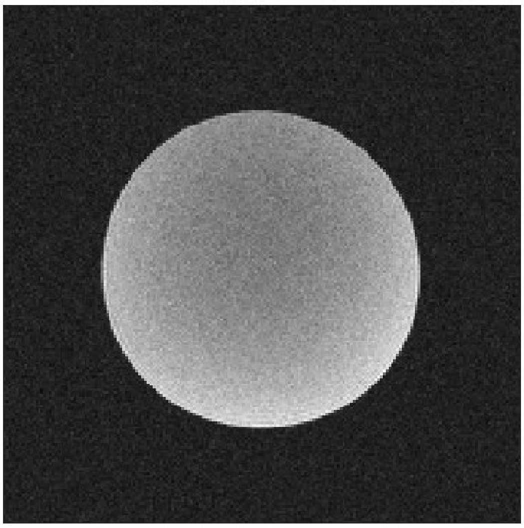
FIGURE 20: Illustration of the magnetic field strength around the MRI scanner, roughly to scale. the 40+ mT field is shown in red. Left: front view. Top right: top view. Bottom right: side view.

Outside the red areas indicated in Fig. 20, no changes in Hololens performance were detected. At magnetic field strengths surpassing 50 mT, holograms would flicker or disappear. Moving the Hololens away from the scanner would cause the holographic scene to reappear unchanged. A list of considered performance markers can be found in Table 4.

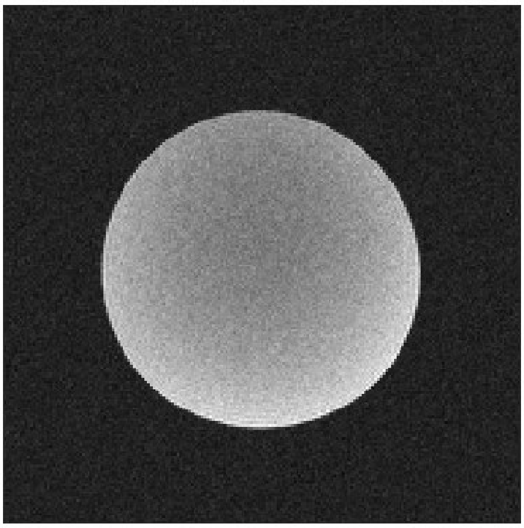
Table 4: Performance evaluation elements during the MR-compatibility experiment. The 'Performance' column denotes whether or not any change in performance was detected for the listed action or functionality.

	Command / action	Performance
Projection	-	Changed
Gaze recognition	-	Unchanged
Hand gesture recognition	Bloom	Unchanged
	Air tap	Unchanged
	Tap and hold	Unchanged
Spatial mapping	Object anchors	Unchanged
	Spatial awareness	Unchanged
Miscellaneous	Moving and resizing objects	Unchanged
	Navigating menu screens	Unchanged

No image artifacts were found in scans made while the Hololens was operational around the 40 mT field. A baseline scan is shown in Fig. 21a. Fig. 21b shows a phantom scan made with identical settings. During acquisition of this scan, the Hololens was turned on and in operation around the 40 mT field. Twenty scans were made for both conditions, with variable slice positioning along the z-axis. The scans shown in Fig. 21 were made close to the centre of the phantom. The SNR averages and standard deviations were calculated according to Eqs. (4) and (3), respectively. The baseline SNR was found to be 23.5 ± 1.0, whereas the SNR under influence of the Hololens was found to be 23.3 ± 1.1. Selected regions of interest (ROI) are depicted in Figs. 33 and 34 in Appendix D.1.



(a) Phantom scan made without the Hololens present.



(b) Phantom scane made with the Hololens operational in the MRI room.

Figure 21: Phantom scans made with (Fig. 21b) and without (Fig. 21b) the Hololens in the MRI room.

4.3 Latency

Latency was measured separately for the image and command data transmission. Depending on the system configuration, the images originated either from the MRI scanner or from the MRI simulator, and were received either on the laptop or Hololens. The latency results are shown in Figs. 22 and 23.

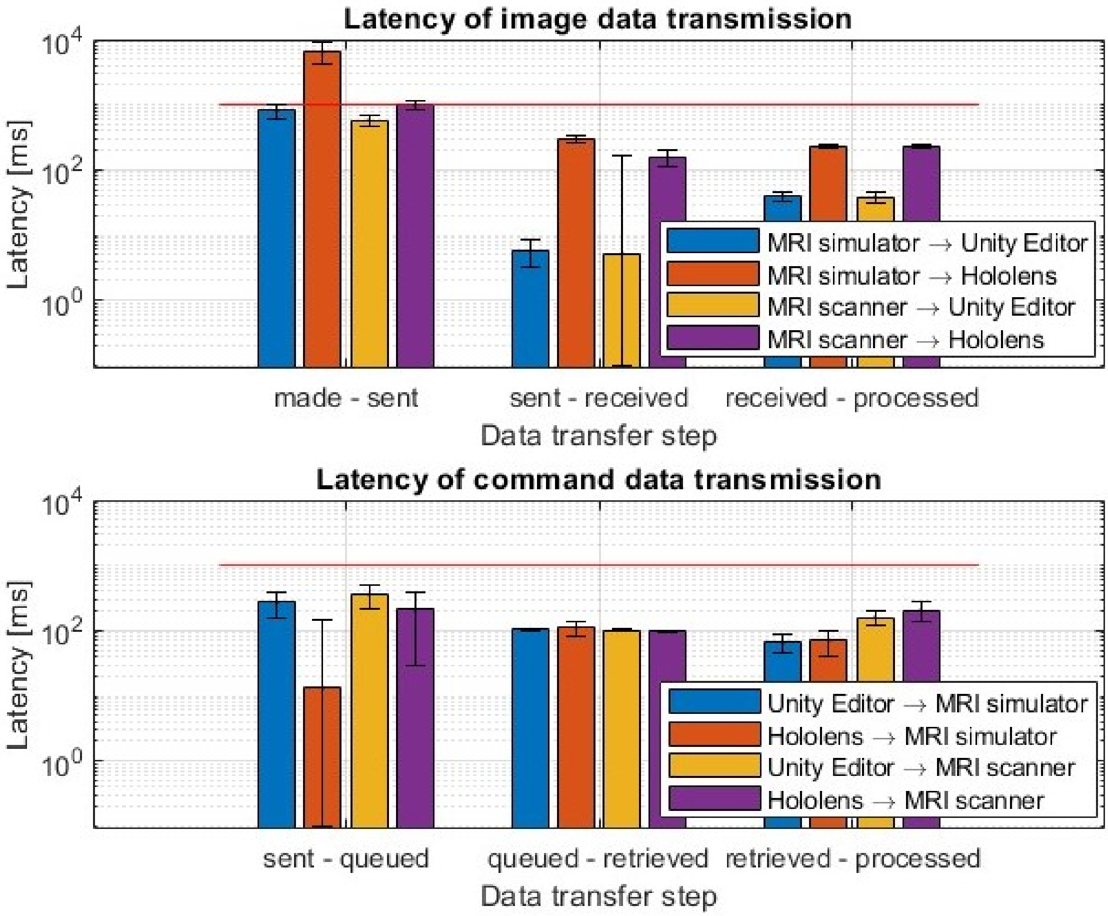


FIGURE 22: Latency of the separate measured data transmission data steps. Different timepoints were appended to the data strings, and saved to .txt files. Afterwards, latencies between these timepoints were calculated. The target latency limit of 1000 ms is indicated in red.

MR image data transmission

During real-time image acquisition, the following four timepoints were defined: time of image acquisition as given in the DICOM metadata, time the image was sent from Python to Unity, time of image reception in Unity and the time at which the image was shown in the holographic scene. It must be noted that image reconstruction is included in the measured latency between the first and second timepoint. The total time elapsed between the first and last checkpoint is shown in Table 5. Latency of the individual steps can be found in Table 13 in Appendix D.2.

The value shown in red in Table 5 was originally calculated to be 541 ms. However, this value resulted from a measured latency of -81.8 ms for the 'sent - received' intermediary

TABLE 5: Image transmission latency. The value in red was altered from the original measured value, as explained in Section 4.3.

	Sample size	**Total time [ms]**
MRI simulator → Unity Editor	158	853 ± 186
MRI simulator → Hololens	120	7058 ± 2361
MRI scanner → Unity Editor	122	628 ± 179
MRI scanner → Hololens	137	1381 ± 182

step (Table 13). A separate measurement was conducted to determine the actual latency of this step, through sending test data bidirectionally between the desktop PC and laptop via publisher-subscriber socket connections and recording the measured latency. The 'true' transmission latency was calculated from the measured difference using Eq. (6), with a sample size of 500. The inaccurate value was replaced, and the total latency of the image data transmission was calculated with the new value. Since the erroneous value was believed to be the result of an offset in system times, the standard deviation remained unchanged.

$$T_{\text{transmission}} = \frac{T_{\text{PC}\rightarrow\text{laptop}} + T_{\text{laptop}\rightarrow\text{PC}}}{2} \tag{6}$$

Where $T_{transmission}$ is the 'true' time needed to send data between the laptop and PC, $T_{PC\rightarrow laptop}$ is the measured time difference for data sent from the PC to the laptop, and $T_{laptop\rightarrow PC}$ is the measured time difference for data sent from the laptop to the PC.

Command data transmission

For the command data, four timepoints were defined as well: time the command was sent from Unity to Python, time of command reception in Python, time of command retrieval in the Python main thread and the time at which the MRI scan parameters were updated. The total time elapsed for different system configurations is shown in Table 6. The latency of the individual steps can be found in Table 14 in Appendix D.2.

TABLE 6: Command latency.

	Sample size	**Total time [ms]**
Unity Editor → MRI simulator	100	439 ± 122
Hololens → MRI simulator	66	194 ± 145
Unity Editor → MRI scanner	73	618 ± 150
Hololens → MRI scanner	69	508 ± 194

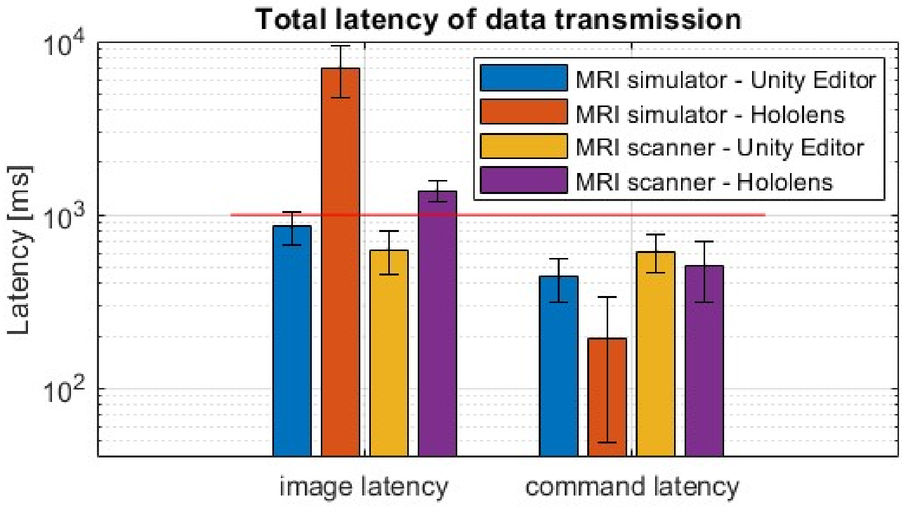


FIGURE 23: Total latency of image and command transmission. The target latency limit of 1000 ms is indicated in red.

4.4 User-friendliness and potential

Seven participants tested the XR-application and filled in the user experience questionnaire shown in Appendix C. Fig. 24 shows the averaged results per category, with 1.00 being the highest possible score. One participant tested the MRI scanner - Hololens configuration outside the MRI room, but was not able to send commands to the MRI scanner. Another tested only the functional MRI simulator - Hololens configuration. The remaining five participants tested the functional MRI scanner - Hololens configuration inside the MRI room. Participants that had been unable to test the functional MRI scanner - Hololens

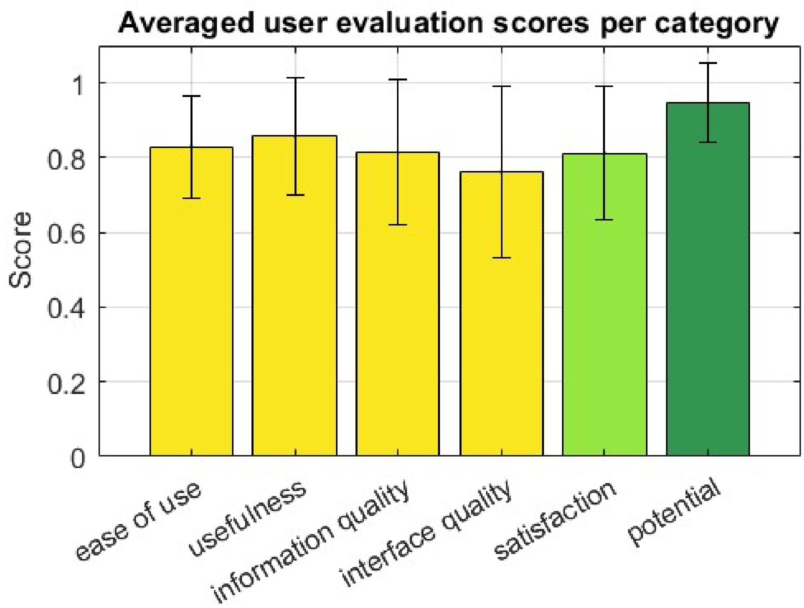


FIGURE 24: User opinions on the XR-application quality and potential for future implementation. The 'potential' category averages survey questions 8 and 18 as listed in the survey shown in Appendix C. The 'satisfaction' category averages questions 1-7 and 9-17.

configuration gave the system a total score of 0.83 ± 0.20 out of 1.00 (averaged over all multiple-choice questions). Participants that had tested with the functional MRI scanner - Hololens configuration rated the device at 0.83 ± 0.17. Overall satisfaction with the current device scored 0.81 ± 0.18, whereas overall future potential scored 0.95 ±0.11.

In question 19 of the survey, users were asked whether they had any additional comments on the application. Relevant answers were paraphrased and organised in three categories: positive remarks on the system and setting, suggestions for improvement of the current system and suggestions for added features. The responses are listed below.

Positive remarks:

- Nice and well-timed test-setting
- Using the system is easy and can be learned in 10 minutes
- Nice concept, the system would definitely be useful in MRI-guided interventions
- The current system delay is not problematic

Suggestions for improvement:

- Minimise delays caused by queue overflow and system latency
- Improve resolution of displayed MR image
- Make panel and button text in the holographic environment more legible
- It can take some time to get the Hololens comfortably positioned on the head. If developing a new HMD, pay attention to the wearer's comfort
- Ensure that only the wearer can provide system input

Features to be added:

- Undo button on the scan position/orientation
- Option to run different MR sequences from the HMD
- Pause button
- Haptic feedback of sorts, so attention can be allocated elsewhere
- Simple transform buttons in addition to current scan-plane control input
- Option to save certain plane transforms to revisit later
- Option to include a 3D pre-made scan for reference
- Option to switch between voice/physical/gesture input
- Option to start a scan from the Hololens
- Option to save scene pre-sets in the XR-application

5 Discussion

MR-safety tests with the Hololens revealed no altered behaviour or noticeable electromagnetic interference below 40 mT. Data transmission latencies for the MRI scanner - Hololens configuration were measured to be 1.38 ± 0.18 s between image acquisition and display in the holographic interface, and 508 ± 194 ms between sending parameter change commands and updating MRI scan settings. Participants in the user experience test rated the developed XR-MRI interface application at 0.81 ± 0.18 out of 1.00, and asserted confidence in future usefulness of the application in clinical settings with a score of 0.95 ± 0.11. Result implications and limitations are discussed below.

MR-compatibility

Hypotheses made in Section 3.5.1 were largely confirmed by the measurement data. As expected, the Hololens is weakly susceptible to magnetic attraction. Spatial anchor permanence during image acquisition was expected to be inferior to standard behaviour, but showed no changes below the 40 mT limit. It is probable that the magnetometer was indeed compromised during the measurement, and that the discrepant sensor output was rejected in favour of accelerometer, gyrometer and camera data [52]. A slight difference in SNR was found between images acquired with and without the Hololens active in the MRI room. However, the considerable overlap in error margin between the values suggests that no definite conclusions about possible electromagnetic interference can be made based on the obtained results. Leastways, the Hololens does not appear to significantly affect MR image quality, supporting in part the theory that Wi-Fi communication frequencies differ sufficiently from the measured Larmor frequency to remain undetected [37, 60]. Performing MRI-guided procedures might require the wearer to be nearer to the MRI bore than the currently safely attainable 60 cm limit. Nevertheless, this proximity to the MRI bore edge is near enough to the scanner for development and testing purposes.

Latency

As mentioned in Section 4, one of the latency measurements resulted in a negative value, as a result of asynchronous system times across devices with fluctuating differences. Effects of this issue were negated by determining the latency of the concerned data transmission step in a separate measurement, exchanging data over ∅MQ sockets and recording the difference in measured latency. However, this method could only be employed for transmission between the desktop PC and laptop, since communication pathways between the MRI scanner and desktop PC and communication pathways between the desktop PC and Hololens are not bidirectionally equivalent. The validity of the conducted latency measurements can therefore not be guaranteed. Even so, the greatest observed discrepancies did not exceed 100 ms, suggesting that the measured delays might still provide an adequate indication of the true interface latency. Generally, system latency exceeding 63 ms in extended reality applications should be avoided, since it has been reported to increase the probability of inducing motion sickness symptoms [63]. In the current application, this effect is deemed to be less relevant due to the data transmission latencies only affecting a small part of the holographic scene.

User-friendliness and potential

Since the participants in the user evaluation all had experience working with the MRI scanner or researching MRI-related topics, they were considered adequately representative of the intended target audience (clinicians and surgeons) for the first testing stage of the prototype. Though all participants had suggestions for improvement or added features, there was little overlap. This suggests that personal preference might play a large role in perceived ease of use and usefulness by the target audience. A limitation of the user test is the sample size of seven, which is not enough to get a conclusive overview of the importance of certain features in the application.

5.1 Current interface limitations

The performance of the current XR-interface prototype is somewhat impeded by hardware limitations. An important limiting factor in project progression and future scalability of the XR-application has been the lack of current software support for the Hololens (see Section 3.1). Due to limited processing power, the application prototype was kept as simple as possible for the purposes of this project. Since keyboard input proved to be unstable in the holographic environment, IP address and server credentials have been hard-coded in the application. Additionally, command data transfer from the Hololens to the desktop PC is currently only possible through the SQL database. A direct socket implementation was developed in both the Python code and Unity application, and tested succesfully in both the MRI simulator - Unity Editor and MRI scanner - Unity Editor configurations. The most probable cause for the lack of data transmission from the Hololens lies in the firewall outbound rules, which are not accessible on the device itself. The communication pathways were kept in the Python code and Unity application for possible future revisitation of the method.

5.2 Recommendations

For future research into the development of bidirectional XR-MRI interfaces, it is advised to focus on developing an MR-safe HMD, minimising latency, and adding functionality to the interface.

Developing a new XR-HMD. For this project, the Hololens gen 1 was deemed adequate and an attractive choice due to being readily available for use at the University of Twente. For future development, however, the device will not suffice. The Hololens 2 has greater memory, processing power and connectivity options as well as more sophisticated and intuitive interaction tracking [64]. However, it is improbable that the Hololens 2 is fully MR-safe, conceivably giving rise to the need for a new HMD. For development of this device, it is recommended to focus on MR-safety first and foremost. Rather than optimising CPU, the device could be designed to run all costly operations on a remote PC via wireless connection.

Minimising latency. System latency should decrease with increased processing power, but could be optimised even further. Code in both Python and Unity is already divided into separate threads to streamline the execution, but long processing times at image conversion steps can still cause delays. Currently, the MRI simulator - Unity Editor and MRI simulator - Hololens configurations suffer from a queue build-up due to the unrealistically high image acquisition rate, which is not accounted for in the Python code. This issue is not present in the MRI scanner - Unity Editor and MRI scanner - Hololens configurations, but could potentially surface as image acquisition rates improve. Adding an extra thread to handle image reception in the Python code should eliminate this issue. Image transmission latency, specifically, could be reduced even further through improving image reconstruction rates. The measured latency of the made - sent transmission step (as seen in Fig. 22 and Table 13) for the MRI scanner → Hololens configuration suggests that image reconstruction could account for up to 60% or more of the total transmission latency. It is assumed here that the latency of data transmission between the MRI scanner and desktop PC is comparable to the measured latency between the desktop PC and Hololens.

Adding features. Due to time constraints and limitations of the Hololens, several useful features were not yet implemented. Firstly, for future versions it is advised to allow the user to manually input the desired connection ports and IP addresses. Secondly, the holographic scene assumes a fixed initial orientation of the scan-plane. To allow for different initial scan-plane orientations, the structure of the MR image stream could be expanded to include the initial orientation. Alternatively, the holographic scene could be expanded to allow the user to specify the orientation before starting the scan. Additional MRI settings could be included as well. Thirdly, the option to add a pre-made 3D-scan into the holographic scene to supplement the 2D images would be beneficial to clinical outcomes [16]. Fourthly, rotating the scan-plane by more than 45 ° causes unexpected shifts in orientation. This is a known phenomenon on the regular MRI scanner interface, but identifying the underlying patterns might allow for mathematical correction of the shift via the Python code. Lastly, the currently used sockets are not secure. If the IP address and port are known, any third party could subscribe to the image stream undetected. To protect patient information sent to the HMD, the sockets should limit the number of allowed subscribers and incorporate an added layer of protection, such as a license or certificate. For the complete list of user-suggested features, refer to Section 4.4.

6 Conclusion

An XR-application was designed to allow for interactive MR image visualisation and scan-plane control during MRI-guided interventions. A proof-of-concept of the proposed system was realised and evaluated to answer the research question: *"What is the potential to improve user-friendliness and efficiency of MRI-guided interventions through the use of interactive XR-guided visualisation?"*.

The application provides a solution to challenges related to restricted MR image display positioning options and a convoluted scan-plane control workflow. Participants in the user experience tests found the system easy to use, and were unanimously positive regarding the potential for further developed versions to improve the efficiency of MRI-guided interventions. Clinical implementation of the proposed system is believed to facilitate reductions in procedural time, complexity and cost.

It is recommended for future research to focus on three main aspects: designing a suitable XR-HMD, minimising latency and adding practical features. In the development of a new HMD, MR-safety is the most important design restriction. In all other respects, the Hololens 2 could be used as inspiration for the necessary functionality. User-suggested missing features include version control of scan-plane and scene settings, access to additional MRI settings from within the holographic environment, the option to include pre-made 3D scans in the holographic environment and full support for input modalities besides gesture control.

Acknowledgements

I would like to extend special thanks to dr.ir. W. M. Brink for offering advice, feedback and general support during the project. Additionally, I thank dr.ir. R. G. J. Damgrave, M. Reinok and L. Marx for their invaluable advice on software implementations, prof.dr.ir. B. ten Haken, dr.ir. W. M. Brink, dr.ir. R. G. J. Damgrave and dr.ir. G. C. Vreman - de Olde for overseeing the project as bachelor assignment committee, and all participants of the user experience test for their time and feedback.

References

[1] O. M. Tepper et al. "Mixed Reality with HoloLens: Where Virtual Reality Meets Augmented Reality in the Operating Room". In: *Plastic & Reconstructive Surgery* 140 (5 2017), pp. 1066–1070. DOI: 10.1097/PRS.0000000000003802.

[2] A. C. Linte et al. "From pre-operative cardiac modeling to intra-operative virtual environments for surgical guidance: an in vivo study". In: *Medical Imaging 2008: Visualization, Image-Guided Procedures, and Modeling.* Ed. by Michael I. Miga and Kevin Robert Cleary. Vol. 6918. International Society for Optics and Photonics. SPIE, 2008, p. 69180D. DOI: 10.1117/12.772028.

[3] L. Olah and M. Olah. "Potential benefits of intraoperative ultrasound in neurosurgery". In: *Journal of Clinical Ultrasound* 51 (4 2023), pp. 739–741. ISSN: 00912751. DOI: 10.1002/jcu.23429.

[4] R. K. Mukherjee et al. "Advances in Real-Time MRI–Guided Electrophysiology". In: *Current Cardiovascular Imaging Reports* 12 (6 2019). DOI: 10.1007/s12410-019-9481-9.

[5] C. R. Weiss, S. G. Nour, and J. S. Lewin. "MR-guided biopsy: A review of current techniques and applications". In: *Journal of Magnetic Resonance Imaging* 27 (2 2008), pp. 311–325. DOI: https://doi.org/10.1002/jmri.21270.

[6] D. A. Ortendahl and L. Kaufman. "Real-Time Interactions in MRI". In: *Computers in Biology and Medicine* 25 (2 1995), pp. 293–300. DOI: https://doi.org/10.1016/0010-4825(94)00012-F.

[7] M. A. Iqbal et al. "Comparison of the Efficiency of Diagnostic Ultrasound and Magnetic Resonance Imaging of Cervical Lymph Nodes in Papillary Thyroid Carcinoma". In: *Journal of X-Ray Science and Technology* 29 (6 2021), pp. 1033–1044. DOI: 10.3233/XST-210927.

[8] E. J. Balthazar et al. "Acute appendicitis: CT and US correlation in 100 patients". In: *Radiology* 190 (1 1994), pp. 31–35. ISSN: 00338419.

[9] W. Abdalla et al. "Lung ultrasound versus chest radiography for the diagnosis of pneumothorax in critically ill patients: A prospective, single-blind study". In: *Saudi Journal of Anesthesia* 10 (3 2016), pp. 265–269. DOI: 10.4103/1658-354X.174906.

[10] M. Mychajlowycz, A. Alabousi, and O. Mironov. "Ultrasound- Versus CT-Guided Subpleural Lung and Pleural Biopsy: An Analysis of Wait Times, Procedure Time, Safety, and Diagnostic Adequacy". In: *Canadian Association of Radiologists Journal* 72 (4 2021), pp. 883–889. ISSN: 08465371. DOI: 10.1177/0846537120939073.

[11] J. Obernauer et al. "Ultrasound-guided versus Computed Tomography-controlled facet joint injections in the middle and lower cervical spine: A prospective randomized clinical trial". In: *Medical Ultrasonography* 15 (1 2013), pp. 10–15. ISSN: 20668643. DOI: 10.11152/mu.2013.2066.151.jo1ugc2.

[12] S. Feig. "Cost-Effectiveness of Mammography, MRI, and Ultrasonography for Breast Cancer Screening". In: *Radiologic Clinics of North America* 48 (5 2010), pp. 879–891. ISSN: 0033-8389. DOI: https://doi.org/10.1016/j.rcl.2010.06.002.

[13] K. S. Nayak et al. "Real-Time Magnetic Resonance Imaging". In: *Journal of Magnetic Resonance Imaging* 55 (1 2022), pp. 1–312. DOI: https://doi-org.ezproxy2.utwente.nl/10.1002/jmri.27411.

[14] A. E. Campbell-Washburn et al. "MR sequences and rapid acquisition for MR guided interventions". In: *Magnetic Resonance Imaging Clinics of North America* 23 (4 2015), pp. 669–679. DOI: https://doi.org/10.1016/j.mric.2015.05.006.

[15] Macrovector. *Medical Diagnostic Equipment Isometric*. Freepik. URL: https://www.freepik.com/free-vector/medical-diagnostic-equipment-isometric_6088345.htm#page=2&query=mri%20scanner&position=4&from_view=keyword&track=ais_user&uuid=7972c78d-6375-42e0-8668-8ef8d7e38173.

[16] C.M. Morales-Mojica et al. "A Holographic Augmented Reality Interface for Visualizing of MRI Data and Planning of Neurosurgical Procedures". In: *Journal of Digital Imaging* 34 (2021), pp. 1014–1025. DOI: https://doi.org/10.1007/s10278-020-00412-3.

[17] Imricor. *Maastricht University Medical Center Commences Procedures*. 2021. URL: https://imricor.com/blog/2021/02/24/maastricht-univerisity-medical-center-commences-procedures/.

[18] F. W. Hirsch et al. "Real-time MRI: a new tool of radiologic imaging in small children". In: *European Journal of Pediatrics* 182 (2023), pp. 3405–3417. DOI: 10.1007/s00431-023-04996-0.

[19] M. Bock et al. "Active catheter tracking using parallel MRI and real-time image reconstruction". In: *Magnetic Resonance in Medicine* 55 (6 2006), pp. 1545–1459. DOI: https://doi.org/10.1002/mrm.20902.

[20] A. Basit, O. Inam, and H. Omer. "Accelerating GRAPPA reconstruction using SoC design for real-time cardiac MRI". In: *Computers in Biology and Medicine* 160 (2023). DOI: https://doi.org/10.1016/j.compbiomed.2023.107008.

[21] D. Brunner. *Frame Rate: A Beginner's Guide*. 2023. URL: https://www.techsmith.com/blog/frame-rate-beginners-guide/.

[22] V. P. B. Grover et al. "Magnetic Resonance Imaging: Principles and Techniques: Lessons for Clinicians". In: *Journal of Clinical and Experimental Hepatology* 5 (3 2015), pp. 246–255. DOI: 10.1016/j.jceh.2015.08.001.

[23] E. Plenge et al. "Super-Resolution Reconstruction in MRI: Better Images Faster?" In: *Progress in Biomedical Optics and Imaging - Proceedings of SPIE* 8314 (2012). DOI: 10.1117/12.911235.

[24] N. M. Cross, M. N. Hoff, and K. M. Kanal. "Avoiding MRI-Related Accidents: A Practical Approach to Implementing MR Safety". In: *Journal of the American College of Radiology* 15 (12 2018), pp. 1738–1744. DOI: https://doi.org/10.1016/j.jacr.2018.06.022.

[25] G. Schaefers. "Testing MR Safety and Compatibility". In: *IEEE Engineering in Medicine and Biology Magazine* 27 (3 2008), pp. 23–27. DOI: 10.1109/EMB.2007.910267.

[26] L. J. Erasmus et al. "A short overview of MRI artefacts". In: *South African Journal of Radiology* 8 (2 2004). DOI: https://doi.org/10.4102/sajr.v8i2.127.

[27] C. Noda et al. "Primer on Commonly Occurring MRI Artifacts and How to Overcome Them". In: *Radiographics* 42 (3 2022). DOI: 10.1148/rg.210021.

[28] H. D. Kocasarac et al. "Evaluation of artifacts generated by titanium, zirconium, and titanium–zirconium alloy dental implants on MRI, CT, and CBCT images: A phantom study". In: *Oral Surgery, Oral Medicine, Oral Pathology and Oral Radiology* 127 (6 2019), pp. 535–544. DOI: https://doi.org/10.1016/j.oooo.2019.01.074.

[29] S. Okamoto et al. "Needle artifact characteristics and insertion accuracy using a 1.2 T open MRI scanner: A phantom study". In: *Diagnostic and Interventional Imaging* 102 (6 2021), pp. 363–370. DOI: https://doi.org/10.1016/j.diii.2020.12.007.

[30] F. Zijlstra, M. A. Viergever, and P. R. Seevinck. "SMART tracking: Simultaneous anatomical imaging and real-time passive device tracking for MR-guided interventions". In: *Physica Medica* 64 (2019), pp. 252–260. DOI: https://doi.org/10.1016/j.ejmp.2019.07.019.

[31] D. Abjigitova et al. "Virtual Reality in the Preoperative Planning of Adult Aortic Surgery: A Feasibility Study". In: *Journal of Cardiovascular Development and Disease* 9 (2 2022). DOI: https://doi.org/10.3390/jcdd9020031.

[32] K. H. Kim. "The Potential Application of Virtual, Augmented, and Mixed Reality in Neurourology". In: *International Neurology Journal* 20 (3 2016), pp. 169–170. DOI: 10.5213/inj.1620edi005.

[33] F. Cutolo et al. "Robust and Accurate Algorithm for Wearable Stereoscopic Augmented Reality with Three Indistinguishable Markers". In: *Electronics* 5 (59 2016). DOI: https://doi.org/10.3390/electronics5030059.

[34] N. Nicolaou. "Magnetic Resonance Imaging Department Cost Study: A Glance from Cyprus". In: *Journal of Health Science* 5 (2017), pp. 340–344. DOI: 10.17265/2328-7136/2017.06.009.

[35] *Bounds control - MRTK2*. Microsoft, 2024. URL: https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/features/ux-building-blocks/bounds-control?view=mrtkunity-2022-05.

[36] R. Kabha. "Ethical Challenges of Digital Immersive and VR". In: *Journal of Content, Community & Communication* 9 (2019), pp. 41–49. ISSN: 2456-9011.

[37] A. S. Minhas and R. Oliver. "Magnetic Resonance Imaging Basics". In: *Electrical Properties of Tissues: Quantitative Magnetic Resonance Mapping*. Ed. by R. Sadleir and A. S. Minhas. Cham: Springer International Publishing, 2022, pp. 47–82. ISBN: 978-3-031-03873-0. DOI: 10.1007/978-3-031-03873-0_3.

[38] W. E. Kwok. "Basic Principles of and Practical Guide to Clinical MRI Radiofrequency Coils". In: *RadioGraphics* 42.3 (2022). PMID: 35394887, pp. 898–918. DOI: 10.1148/rg.210110.

[39] T. Puiseux et al. "Numerical simulation of time-resolved 3D phase-contrast magnetic resonance imaging". In: *PLOS ONE* 16 (3 Mar. 2021). DOI: 10.1371/journal.pone.0248816.

[40] R. Deichmann et al. "The Basics of Functional Magnetic Resonance Imaging". In: *EEG - fMRI: Physiological Basis, Technique, and Applications*. Ed. by Christoph Mulert and Louis Lemieux. Cham: Springer International Publishing, 2022, pp. 49–77. ISBN: 978-3-031-07121-8. DOI: 10.1007/978-3-031-07121-8_3.

[41] R. Deakin. "3-D Coordinate Transformations". In: *Surveying and Land Information Systems* 58 (4 1998), pp. 223–34. URL: https://www.researchgate.net/publication/228608056_3-D_Coordinate_Transformations.

[42] *MAGNETOM Aera*. Siemens Healthineers. URL: https://www.siemens-healthineers.com/en-us/magnetic-resonance-imaging/0-35-to-1-5t-mri-scanner/magnetom-aera.

[43] M. J. Graves et al. "Body MRI artifacts in clinical practice: A physicist's and radiologist's perspective". In: *Journal of Magnetic Resonance Imaging* 38 (2 2013), pp. 269–287. DOI: `https://doi.org/10.1002/jmri.24288`.

[44] T. Budrys et al. "Artifacts in magnetic resonance imaging: how it can really affect diagnostic image quality and confuse clinical diagnosis?" In: *Journal of Vibroengineering* 20.2 (2018), pp. 1202–1213. DOI: `10.21595/jve.2018.19756`.

[45] J. Delfino et al. "MRI-related FDA adverse event reports: A 10-yr review". In: *The International Journal of Medical Physics Research and Practice* 46 (12 2019), pp. 5562–5571. DOI: `https://doi.org/10.1002/mp.13768`.

[46] L. Parziale et al. "TCP/IP Tutorial and Technical Overview". In: 8th ed. IBM Redbooks, 2006. Chap. 1.

[47] P. Hintjens. *ZeroMQ: Messaging for Many Applications*. 1st ed. O'Reilly Media, Inc., 2013. ISBN: 978-1-449-33406-2.

[48] *Windows Sockets 2*. Microsoft, 2021. URL: `https://learn.microsoft.com/en-us/windows/win32/winsock/windows-sockets-start-page-2`.

[49] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems*. 7th ed. Pearson Education, 2016. ISBN: 1292097620.

[50] J. Albahari. "Multithreading in C#". In: O'Reilly Media, Inc., 2006. Chap. 1.

[51] M. Reinok. *Library for Siemens Access-i MR Scanner Interface to integrate and control the MR Scanner*. University of Twente, 2024. URL: `https://pypi.org/project/accessi/`.

[52] R. Mamdouh et al. "Improve Speed Real-Time Rendering in Mixed Reality HOLOLENS during Training". In: *International Journal of Advanced Computer Science and Applications* 11 (2020). DOI: `10.14569/IJACSA.2020.0110339`.

[53] *Choosing a Unity version and XR plugin*. Microsoft, 2024. URL: `https://learn.microsoft.com/en-us/windows/mixed-reality/develop/unity/choosing-unity-version`.

[54] *Supported XR Devices*. Epic Games. URL: `https://dev.epicgames.com/documentation/en-us/unreal-engine/supported-xr-devices-in-unreal-engine?application_version=5.3`.

[55] *Mri Scanner PNG*. PlusPNG. URL: `https://pluspng.com/png-73944.html`.

[56] *360x240 Search Photos Hololens*. Vectorified. URL: `https://vectorified.com/download-image#hololens-icon-27.jpg`.

[57] *Mediator*. Refactoring Guru. URL: `https://refactoring.guru/design-patterns/mediator`.

[58] *NEMA Standards Publication MS 1-2008: Determination of Signal-to-Noise Ratio (SNR) in Diagnostic Magnetic Resonance Imaging*. 2008. URL: `https://www.mriquestions.com/uploads/3/4/5/7/34572113/nema_snr_standards_2008.pdf`.

[59] P. A. Rinck. "Magnetic Resonance in Medicine: A Peer-Reviewed Critical Introduction". In: 14th ed. TwinTree, 2024. Chap. 2. ISBN: 3746095182.

[60] K. R. Foster. "Radiofrequency Exposure from Wireless LANs Utilizing Wi-Fi Technology". In: *Health Physics* 92 (3 2007), pp. 280–289. DOI: `10.1097/01.HP.0000248117.74843.34`.

[61] J. Lewis. "IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use". In: *International Journal of Human-Computer Interaction* 7 (1 1995), pp. 57–78. DOI: 10.1080/10447319509526110.

[62] F. Davis and F. Davis. "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology". In: *MIS Quarterly* 13 (3 1989), pp. 319–339. DOI: 10.2307/249008.

[63] P. Caserman, M. Martinussen, and S. Göbel. "Effects of End-to-end Latency on User Experience and Performance in Immersive Virtual Reality Applications". In: *Entertainment Computing and Serious Games*. Ed. by E. van der Spek et al. Cham: Springer International Publishing, 2019, pp. 57–69. ISBN: 978-3-030-34644-7.

[64] J. Kosciesza. *HoloLens 2 vs HoloLens 1: what's new?* 2020. URL: https://4experience.co/hololens-2-vs-hololens-1-whats-new/.

[65] R. Brown. *Compare Augmented Reality Headsets & Glasses*. URL: https://vr-compare.com/ar.

[66] *Get your HoloLens (1st gen) ready to use*. Microsoft, 2021. URL: https://learn.microsoft.com/en-us/hololens/hololens1-setup.

[67] *Wireless network 'UThings' for smart devices*. University of Twente, 2022. URL: utwente.nl/uthings.

[68] *Using Visual Studio to deploy and debug*. Microsoft, 2022. URL: https://learn.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/using-visual-studio?tabs=hl2.

[69] *Connect Hololens to a network*. Microsoft, 2022. URL: https://learn.microsoft.com/en-us/hololens/hololens-network.

[70] *Create with Unity in three steps*. URL: https://unity.com/download.

[71] *Unity download archive*. URL: https://unity.com/releases/editor/archive.

[72] *Install the tools*. Microsoft, 2023. URL: https://learn.microsoft.com/en-us/windows/mixed-reality/develop/install-the-tools.

[73] *Installation guide*. URL: https://hololabinc.github.io/MixedRealityToolkit-Unity/Documentation/Installation.html#get-the-latest-mrtk-unity-packages.

[74] MaxWang-MS. *Microsoft Mixed Reality Toolkit v2.8.3*. 2022. URL: https://github.com/Microsoft/MixedRealityToolkit-Unity/releases.

[75] *What's New in PyCharm 2024.1*. JetBrains, 2024. URL: https://www.jetbrains.com/pycharm/whatsnew/2024-1/.

[76] *Quickstart: Create a single database - Azure SQL Database*. Microsoft, 2024. URL: https://learn.microsoft.com/en-us/azure/azure-sql/database/single-database-create-quickstart?view=azuresql&tabs=azure-portal.

[77] ParthManiyar. *Basic Database Operations Using C#*. 2023. URL: https://www.geeksforgeeks.org/basic-database-operations-using-c-sharp/.

A Hardware choices and software installation guide

At the start of the project, a suitable HMD was chosen. For this purpose, a number of different AR and XR HMD types were compared (Appendix A.1). For future reference, an installation and start-up guide for the used software is included in Appendix A.2.

A.1 AR/XR HMD specifications

A comparison between different features of AR/XR HMDs was made. In Table 7, general features are listed. Prices above €1000 were considered an excluding factor. Table 8 lists

Table 7: Overview of general HMD features and specifications. Red cells: the feature or HMD falls outside the inclusion criteria. Green cells: positive features.

HMD	AR/XR	Price range	Platform	Release year
Microsoft Hololens gen 1	XR	€300+ (eBay)	Windows Mixed Reality	2016
Microsoft Hololens 2	XR	€1700+ (eBay) €3200 (new)	Windows Mixed Reality	2019
Everysight Raptor	AR	€500-600		2017
MAD Gaze Vader	AR	€700-800		2017
Dream Glass	AR	€300-400	Unity	2018
ThirdEye X2	AR	€2200-2300	VisionEye	2018
Google Glass Enterprise Edition 2	AR	€900-1000		2019
MAD Gaze Glow Plus	AR	€500-600		2020
Dream Glass 4K	AR	€500-600		2020
Dream Glass 4K Plus	AR	€700-800	Android	2020
Nreal Light	AR	€400-500	Nebula	2020
Realmax Qian	AR	€900-1000		2020
Julbo EVAD-1	AR		ActiveLook	2020
Vuzix Blade Upgraded	AR	€700-800	Vuzix	2020
Epson Moverio BT-40	AR	€500-600		2021
Epson Moverio BT-40S	AR	€800-900		2021
Snap Spectacles (2021)	AR		Lens Studio	2021
AjnaLens AjnaXR PRO	XR	€1100-1200		2021
Engo Eyewear	AR	€300-400	ActiveLook	2021
Tilt Five	AR	€300-400	Steam	2021
Cosmo Vision	AR	€500-600	ActiveLook	2022
Dream Glass Lead Plus	AR	€700-800		2022
NuEyes Pro 3e	AR	€400-500		2022
Oppo Air Glass	AR	€700-800		2022
MAD Gaze Wave	AR	€300-400		2022
INMO Air	AR	€300-500	INMO OS	2022
Vuzix Blade 2	AR		Vuzix	2022
P&C Solution METALENSE	AR			2022
INMO Air2	AR	€500-600	INMOVERSE	2023

the device input types. Exclusion criteria were a lack of native pass-through or no hand tracking/gesture control. [65] was used as main source for the comparison. HMDs excluded in Table 7 are marked in grey. Table 9 lists connection options of the HMDs. Exclusion criteria were a lack of wireless, stand-alone or Wi-Fi connection options. HMDs excluded in Tables 7 and 8 are marked in grey. Finally, Table 10 shows a short list of all HMDs meeting the inclusion criteria. The Microsoft Hololens gen 1 was chosen because XR allows for 3D holographic interaction.

TABLE 8: Overview of HMD input specifications. Red cells: the feature or HMD falls outside the inclusion criteria. Green cells: positive features. Grey cells: the HMD has been excluded by earlier considered criteria.

HMD	Native pass-through	Hand tracking / gesture control	Voice control	Battery life (hours)
Microsoft Hololens gen 1	Yes	Yes	Yes	2-3
Microsoft Hololens 2	Yes	Yes	Yes	2-3
Everysight Raptor	Yes	No		8
MAD Gaze Vader	Yes	Yes		
Dream Glass	Yes	Yes		-
ThirdEye X2	Yes			
Google Glass Enterprise Edition 2	Yes	No	Yes	8
MAD Gaze Glow Plus	Yes	Yes		-
Dream Glass 4K	Yes	No		5
Dream Glass 4K Plus	Yes	No		5
Nreal Light	Yes	Yes		-
Realmax Qian	Yes	No		5
Julbo EVAD-1	Yes	No		12
Vuzix Blade Upgraded	Yes	No		8
Epson Moverio BT-40	Yes	No		-
Epson Moverio BT-40S	Yes	No		
Snap Spectacles (2021)	Yes			
AjnaLens AjnaXR PRO		Yes		3+
Engo Eyewear	Yes	No		12
Tilt Five	Yes	No		-
Cosmo Vision	Yes	No		12
Dream Glass Lead Plus	Yes	No		3.5
NuEyes Pro 3e	Yes	No		-
Oppo Air Glass	Yes	No		
MAD Gaze Wave	Yes	Yes		-
INMO Air	Yes			3.5
Vuzix Blade 2	Yes	No		
P&C Solution METALENSE	Yes	Yes		3
INMO Air2	Yes			

Table 9: Overview of HMD connectivity specifications. Red cells: the feature or HMD falls outside the inclusion criteria. Green cells: positive features. Grey cells: the HMD has been excluded by earlier considered criteria.

HMD	Wireless	Stand-alone	Wi-Fi	Bluetooth
Microsoft Hololens gen 1	Yes	Yes	Yes	Yes
Microsoft Hololens 2	Yes	Yes	Yes	Yes
Everysight Raptor	Yes	Yes	Yes	Yes
MAD Gaze Vader	Yes	Yes	Yes	Yes
Dream Glass	No	No	No	No
ThirdEye X2	Yes	Yes	Yes	Yes
Google Glass Enterprise Edition 2	Yes	Yes	Yes	Yes
MAD Gaze Glow Plus	No	No	No	No
Dream Glass 4K	Yes	Yes	Yes	Yes
Dream Glass 4K Plus	Yes	Yes	Yes	Yes
Nreal Light	No	No	No	No
Realmax Qian	Yes	Yes	Yes	Yes
Julbo EVAD-1	Yes	Yes	No	Yes
Vuzix Blade Upgraded	Yes	Yes	Yes	Yes
Epson Moverio BT-40	No	No	No	No
Epson Moverio BT-40S	Yes	Yes	Yes	Yes
Snap Spectacles (2021)	Yes	Yes		
AjnaLens AjnaXR PRO	Yes		Yes	Yes
Engo Eyewear	Yes	Yes	No	Yes
Tilt Five	No	No	No	Yes
Cosmo Vision	Yes	Yes	No	Yes
Dream Glass Lead Plus	Yes	Yes	Yes	Yes
NuEyes Pro 3e	No	No	No	No
Oppo Air Glass	Yes	Yes	Yes	Yes
MAD Gaze Wave	No	No	No	No
INMO Air	Yes	Yes	Yes	Yes
Vuzix Blade 2	Yes	Yes	Yes	Yes
P&C Solution METALENSE	Yes	Yes	Yes	Yes
INMO Air2	Yes	Yes	Yes	Yes

Table 10: HMDs meeting all inclusion criteria. Green cells: positive features.

HMD	AR/XR	Price range
Microsoft Hololens gen 1	XR	€300+ (eBay)
MAD Gaze Vader	AR	€700-800
Snap Spectacles (2021)	AR	
INMO Air	AR	€300-500
P&C Solution METALENSE	AR	
INMO Air2	AR	€500-600

A.2 Unity app development for the Hololens gen 1

Since the Hololens gen 1 was released in 2016 and succeeded by the Hololens 2 in 2019 [65], most newer software does not directly support the Hololens gen 1. It was therefore chosen to use older versions of Unity and Visual Studio to develop the XR-application. Below follow instructions to set up a Unity project and deploy it to the Hololens. The required components are: Hololens gen 1, a laptop and a desktop PC. At the end of this instruction set, you should have the correct system configurations to develop an XR-application that runs independently on the Hololens and communicates bidirectionally with the desktop PC.

Note: it is assumed that the PC's or laptops used are running Windows 10 or Windows 11, and that the user has a basic knowledge of Unity.

Hololens instructions

Below follow instructions on how to set up the Hololens for a PC connection, and deploy your Unity project to the Hololens.

1. Turn on the Hololens according to the quick guide instructions (if you don't have the guide, refer to [66] and related sites).
2. Connect to a Wi-Fi network. This needs to be the same network as your PC! The Hololens cannot connect to Wi-Fi networks with a complex authorisation set-up, like eduroam. When working at the University of Twente, there are two alternatives:
   (a) A connection to the smart devices network 'UThings' can be made. Refer to [67] to get access to the network.
   (b) Set up a mobile hotspot on your PC using Settings > Network & Internet. Note that the Hololens gen 1 cannot connect to the 5G network, so you will have to edit your hotspot to use the 2.4GHz network band (or similar).
3. Enable developer mode on the Hololens [68]. To do this, use the 'start' gesture to open the main menu panel. Then go to Settings > Update > For developers and enable "Use developer features". You may need to also enable "Device discovery".
4. Get the IP address of your Hololens. There are at least three ways to do this (refer also to [69]):
   (a) In the Microsoft Store (on the Hololens), install 'Holographic Remoting Player'. Launch the app. It will display the IP address.
   (b) Say "Hey Cortana, what's my IP address?".
   (c) Go to Settings > Network & Internet. Underneath the available Wi-Fi networks, select "Hardware Properties".

Laptop instructions

1. Enable developer mode on the laptop. Refer to [68] for the steps.
2. Install Unity: follow the steps on [70]:
   (a) Download the Unity Hub. The hub manages your Unity projects and editor versions.
   (b) Choose Unity version 2019.4.40 from the download archive [71].

3. Visual Studio should be installed together with Unity. If not, install Visual Studio 2022 Community (free for students). Refer also to [72], make sure to install the necessary workloads and components. *Note: the author used Visual Studio Community 2017 for editing scripts inside Unity. This version was installed together with Unity 2017. To deploy the application to the Hololens, either Visual Studio 2017 or 2022 was used.*

4. Create an empty project. Open the Unity Hub, click "New Project". Make sure to set the Editor Version to 2019.4.40f1 and select the "3D (Built-In Render Pipeline) Core" template. Give the project a suitable name and set the path. See also Fig. 25. Unity has a character limit on the path length, so make sure to put the project in a folder close to C:\. Click "Create Project". This may take a while.

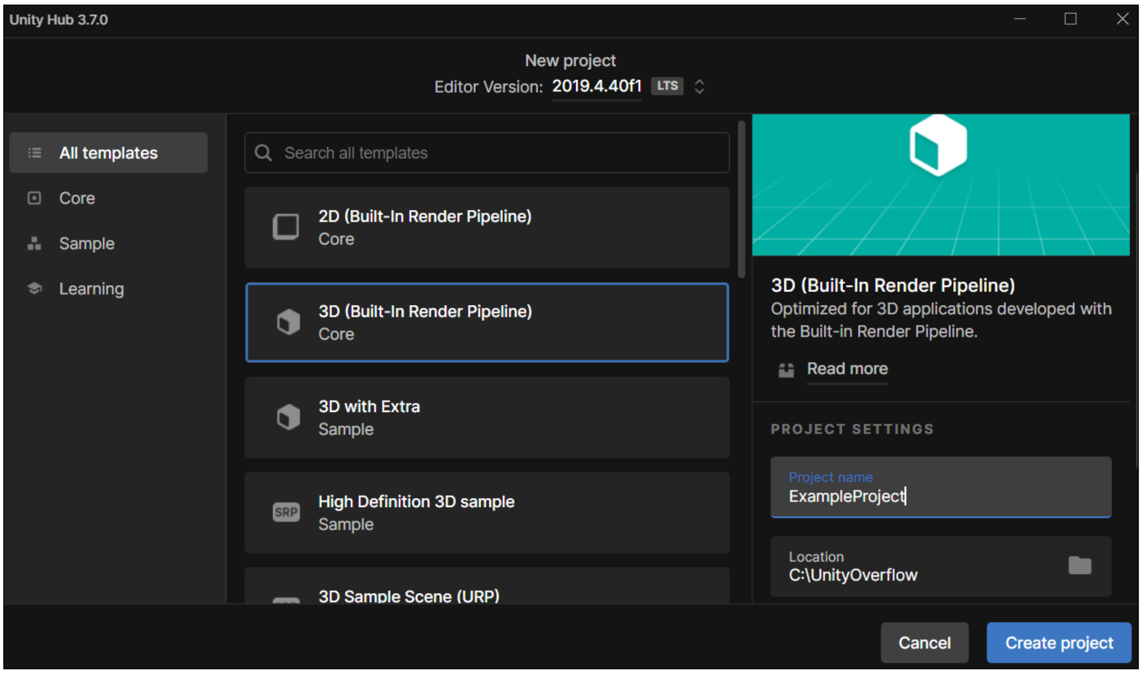


FIGURE 25: Unity Hub project settings window.

5. Download and install the Mixed Reality Toolkit packages (these include proper interpretation of the Hololens gen 1 sensor input and pre-fabricated objects like menu panels): follow the instructions on [73]:

   (a) Go to [74] and download the packages you want (Microsoft.MixedReality.Toolkit.Unity.Foundation.2.8.3.unitypackage is required, the rest is optional).

   (b) In the File Explorer, move the downloaded MRTK package(s) into the "Assets" folder of your Unity project (You might want to move it into a subfolder called 'MRTK' to keep things organised). In the Unity Editor, you should now see the package(s) appear. Double-click on them, and on the pop-up menu select "Import". After a while, you get the MRTK Project Configurator pop-up. Double-click "Legacy XR". Follow the instruction steps, make sure to select "Import TMP Essentials". Next, you can read up on the documentation, or click "Done".

   (c) In the top menu bar, select File > Build Settings. Set the platform to Universal Windows Platform. You may leave the other settings as they are, or change them to the settings shown in Fig. 26. Click "Switch Platform". You may get the MRTK Project Configurator again, just select the same settings.

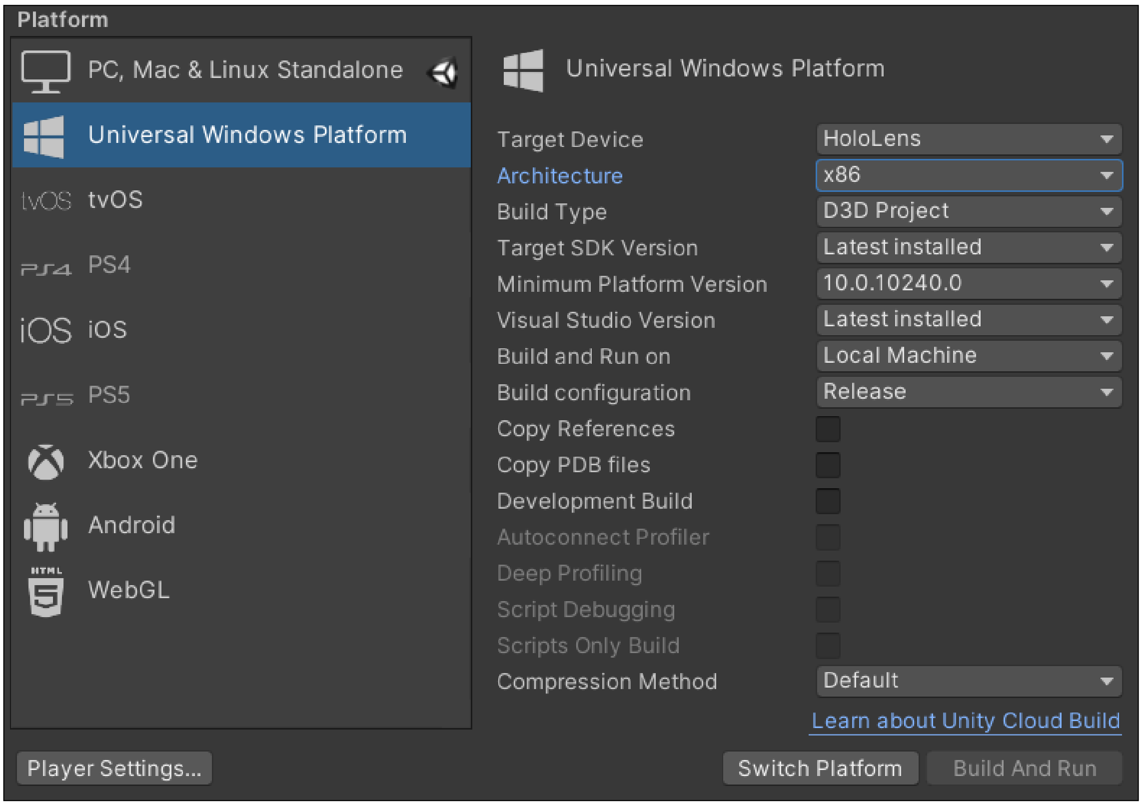


FIGURE 26: Build settings for Hololens development.

(d) In the top menu bar, select Mixed Reality > Toolkit > Add to Scene and Configure. On the right, you will now see MixedRealityToolkit in the inspector.

(e) Make sure to select the correct configuration in the attached MixedReality-Toolkit script. For Hololens gen 1, this is "DefaultHololens1ConfigurationProfile".

6. If you want to test your project, just drag a prefab from Assets > MRTK > SDK > Features > UX > Prefabs > Menus into the hierarchy panel. Make sure to drag it onto the MixedRealitySceneContent object, to maintain an organised hierarchy. Click Play at the top, and see what happens! You can move the menu, unpin it so it follows you, and click buttons. For simulated hand gesture input, see [73].

7. Optional: for better performance, you might need to lower the quality settings of the project. In the top menu bar, go to Edit > Project Settings > Quality and set the default for UWP to "Very Low". Next, navigate to Edit > Project Settings > Player and find the UWP "XR settings". Set the Depth Format to 16-bit depth.

8. Now, you can build the project. Again, go to File > Build settings. Check if the settings are correct (see Fig. 26), and if the scene you want to build is added to the "Scenes in Build" field. If not, click "Add Open Scenes" and make sure your desired scene is checked. Next, click "Build". In the File Explorer pop-up, create a new folder called "App" and select this as your target folder. The build may take a while.

9. In the File Explorer, navigate to the "App" folder in your Unity project. Open the folder, then open the .sln file inside with Visual Studio. In Visual Studio, set the configuration to "Release", the platform to "x86" and the play setting to "Remote Machine". See also Fig. 27.

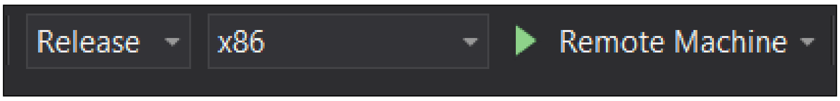


FIGURE 27: Visual Studio deployment settings.

   In the top bar, navigate to Debug > <YOUR PROJECT NAME>. Properties..., and in the left panel of the pop-up, select "Debugging" and set the "Machine Name" property to your Hololens' IP address. Make sure to click "Apply" before closing the Window!

10. Make sure the Hololens is turned on and connected to the same Wi-Fi network as the laptop. In Visual Studio, click "Start without debugging". After a few minutes, your app should open in the Hololens.

Desktop PC instructions

The Hololens will communicate with a Python script running on the desktop PC. In order to do this, the PC needs to have a Python interpreter installed. Firewall rules and relevant settings need to allow for the data to go through.

1. Enable developer mode on the PC. Refer to [68].

2. Install PyCharm 2024.1 Professional Edition or PyCharm 2024 Community Edition [75]. If no python interpreter is installed, PyCharm should direct you to a link to download the newest version when you create or open a new project. If you are missing Python packages required to execute a code, go to Python Packages in the bottom left corner of the IDE, and search for the packages by name to install them.

3. Set up the Windows Defender Firewall to allow certain ports to send and receive data. Go to Windows Defender Firewall > Advanced settings > Inbound Rules and add a rule to allow local port <YOUR PORT> through the firewall. Set the remote port field to 'All ports'. Add a new outbound rule with the same settings.

Database instructions

You might want to store data in a database to circumvent firewall difficulties or have a more structured process. In this case, you can create an SQL database and server. This can be done from any laptop or PC.

1. Follow the steps on [76] to create an Azure SQL database.

2. On your Microsoft Azure dashboard, navigate to <YOUR DATABASE NAME> > Query editor (preview) and log in using your credentials. In a new query, type the code given in [77] to create a new table.

3. Navigate to <YOUR DATABASE NAME> > Set server firewall. Add new rules for all IP addresses you want to give access to your server (most likely, these are your

desktop PC and Hololens IP addresses). Be aware that you will have to create a new rule every time an IP address changes, and sometimes even if the IP address is still the same.

A.3 Troubleshooting the application

If you are running the XR-application and run into errors or the Hololens is not sending/receiving properly, there are a few common causes you might want to check. They are listed below.

- *Error connecting to MRI scanner and opening template.* Make sure Access-i is properly set up on the MRI interface computer, the BEAT interactive template is selected and remote connections are allowed.

- *Connected to MRI scanner but cannot request control.* Turn off remote connections on the MRI interface computer or simulator for five seconds, then turn it on again. Most likely, the connection was not properly severed on the last interaction.

- *Sockets not connecting.* Socket information not defined correctly. In the Unity Editor or on the Hololens XR-MRI-interface menu panel, check if the IP addresses and ports listed are all correct. They need to meet three criteria:

  1. IP addresses and ports need to match in Python and Unity for a successful connection. The host IP is the IP address of the device running the publisher socket of the publisher-subscriber pair.
  2. IP addresses need to match the current operational IP of the used devices. The desktop PC has a static IP address, but the Hololens and third party laptop might have changing IP addresses.
  3. The ports used need to have an inbound and outbound rule allowing data sent over the specified local port to pass through the firewall of both the sending and receiving device.

- *Error during SQL interactions.* SQL database information not defined correctly. Currently, the database connection information is hard-coded in the device and should not cause issues. If the database or server is changed, however, make sure to not only change the connection information in both Python and Unity, but also check if the column tags used in Unity (DatabaseConnection.cs) still match the SQL table.

- *Error during SQL login.* SQL database server firewall issues. If the IP address of any of the connecting devices changes, the server firewall needs to be updated. If you still get errors, try updating every IP address involved once again for good measure. Might be an altered subnet mask or the like, or just the server being stubborn.

- *Connection start-up successful, but no images received in Python after scan start.* This most commonly happens after an error during SQL login or unexpected crash during the main loop in Python. The best way to fix this is by restarting the device running the Python code.

- *XR-application not responding to tapping gesture.* If the problem persists after a few seconds, try calling the in-app menu bar with the bloom gesture, then closing it again with the bloom gesture. Gesture input should be responsive again.

B Additional activity diagrams

This section contains several additional activity diagrams depicting data transmission and processing in the Unity application and Python code. Data connections between diagrams are indicated with matching colours. Green components in Fig. 28 and red components in Fig. 29 are reviewed in Fig. 30. The blue component in Fig. 31 is retrieved in Fig. 15. The magenta component in Fig. 32 is received from Fig. 15.

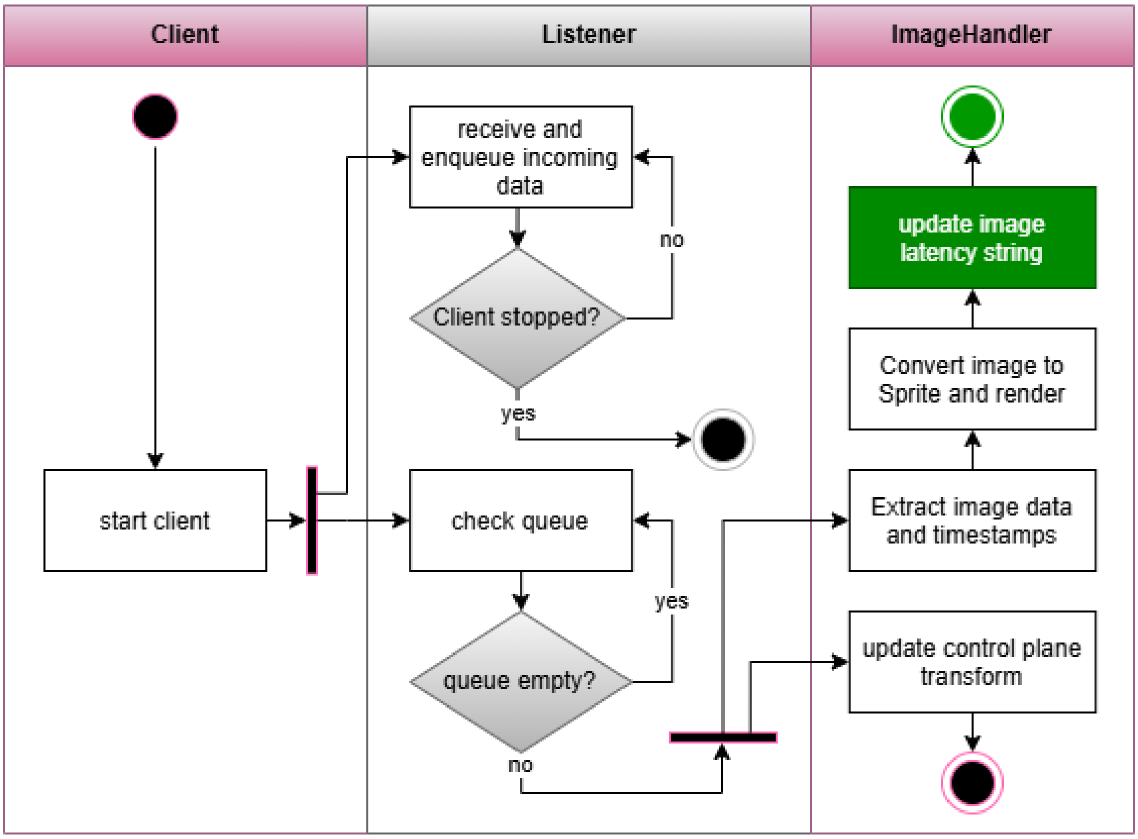


FIGURE 28: Activity diagram depicting data reception in the Unity application.

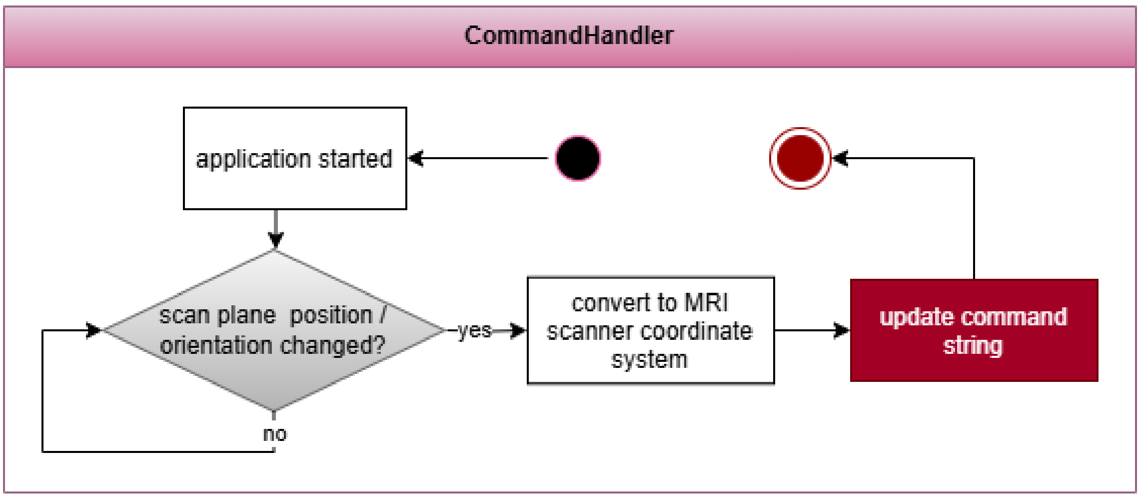


FIGURE 29: Activity diagram of command processing in the Unity application.

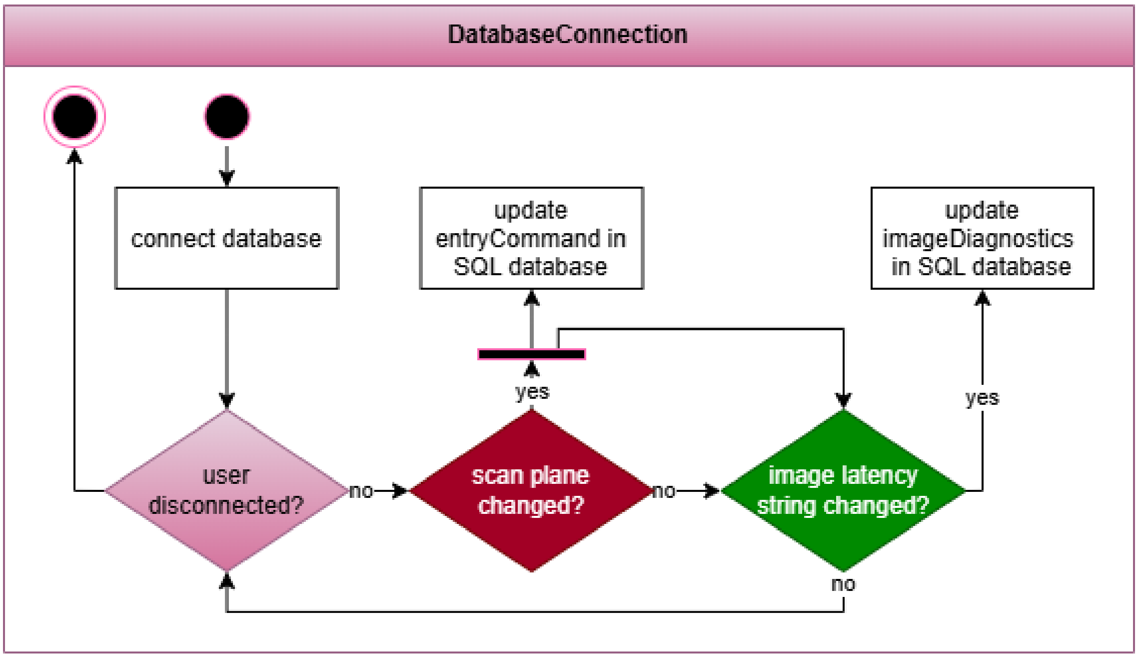


FIGURE 30: Activity diagram depicting database interaction in the Unity application.

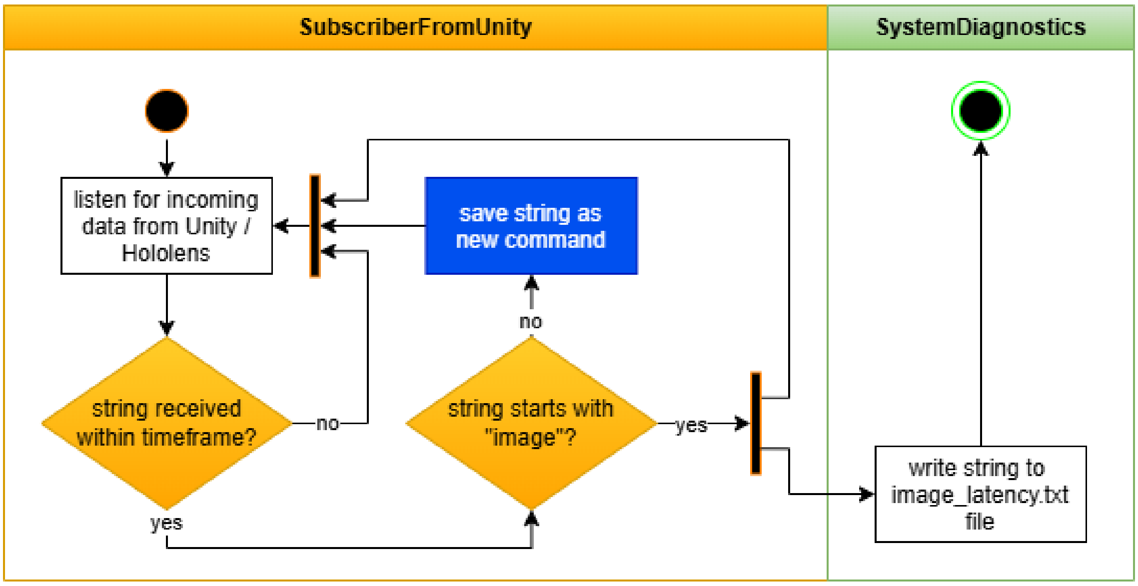


FIGURE 31: Activity depicting data reception via subscriber socket in the Python code.

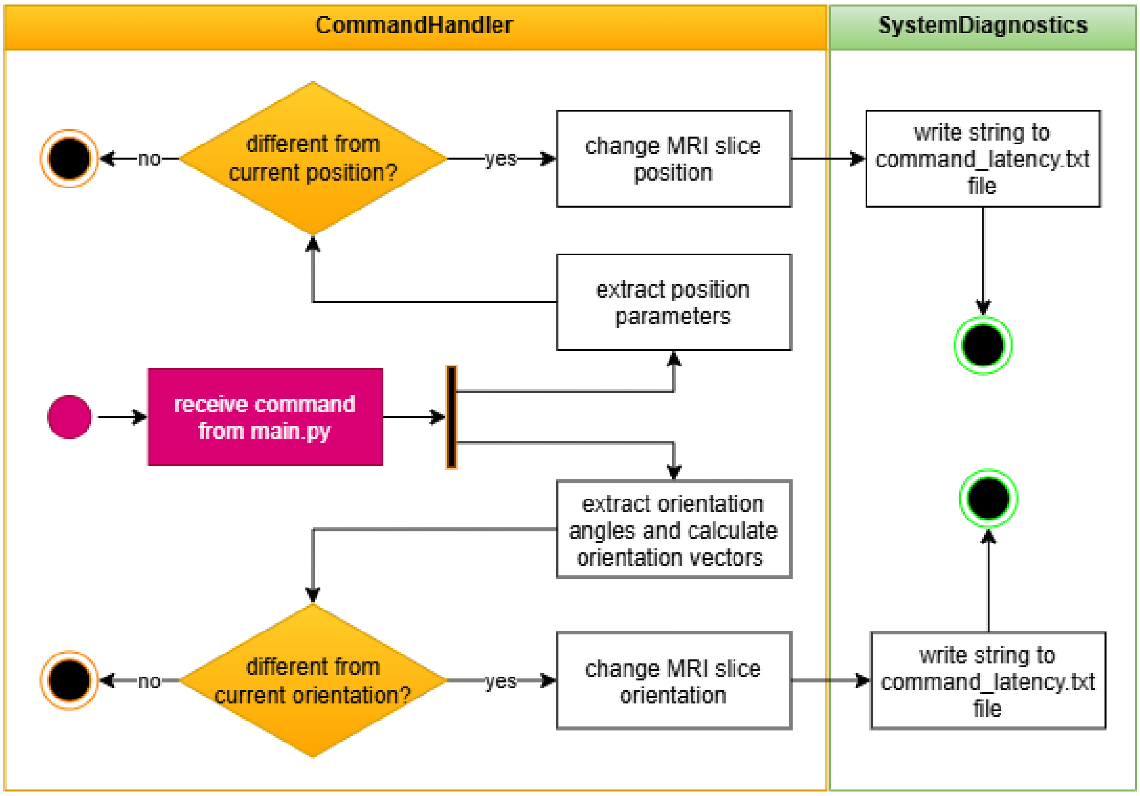


FIGURE 32: Activity diagram depicting command processing in the Python code.

C User experience questionnaire

The survey questions from the user-friendliness test can be found in Table 11. Since the survey is based on both [61] and [62], the source of each question is specified individually. The introductory paragraph at the top of the survey is as follows:

"A major challenge in improving the feasibility of MRI-guided interventions is efficient control over the position of the scanned image. Currently, scan settings can only be changed using a desktop interface outside the scan room. A proposed solution is the use of a mixed reality headset to allow for intuitive scan-plane control from within the MRI room.

In today's user-experience test, you have used a prototype of the new interface. In this survey, we ask for your opinion on the system and its potential for further development. Your - anonymous - response will be used in the project evaluation. Thanks in advance for your participation!"

TABLE 11: User questionnaire to accompany the user-friendliness test. Questions 1-18 are multiple choice with options: 'Strongly disagree', 'Disagree', 'Neutral','Agree', 'Strongly agree'. Question 19 is open.

	Question	Source
Ease of use		
1	Overall, I am satisfied with how easy it is to use this system.	Taken from [61]
2	It was simple to use this system.	Taken from [61]
3	I felt comfortable using this system.	Taken from [61]
4	I found it easy to get the system to do what I want it to do.	Modified from [62]
Usefulness		
5	I was able to effectively interact with the MRI scanner using this system.	Modified from [61]
6	I was able to efficiently interact with the MRI scanner using this system.	Modified from [61]
7	Using the system makes it easier to interact with the MRI scanner.	Modified from [62]
8	Overall, I think the system could be useful in MRI-guided interventions.	Based on [62]
Information Quality		
9	Whenever I made a mistake using the system, I could recover easily and quickly.	Taken from [61]
10	The User Guide and button/panel labels were clear.	Modified from [61]
11	It was easy to find the information I needed.	Taken from [61]
12	The information provided for the system was easy to understand.	Taken from [61]
13	The organisation of information on the system screens was clear.	Taken from [61]
Interface Quality		
14	The interface of this system was pleasant.	Taken from [61]
15	I liked using the interface of this system.	Taken from [61]
16	This system has all the functions and capabilities I expect it to have.	Taken from [61]
General		
17	Overall, I am satisfied with this system.	Taken from [61]
18	Overall, I think this system has the potential to improve the efficiency of MRI-guided interventions.	Taken from [61]
19	Do you have any further comments?	

D Additional results

This section follows the same structure as Section 4. Since the raw data obtained in the measurements was too extensive to be displayed here, the below sections show minimally processed data instead. The contents of this section are not vital for understanding the project, but rather provide some insights into result processing.

D.1 MR-compatibility of the Hololens

An overview of the settings of the MRI scanner used for the MR-compatibility measurements can be found in Table 12. With these settings, twenty phantom scans were obtained without the Hololens. Fig. 33 shows the ROI used to differentiate between signals originating from the phantom and background noise. Another set of twenty images was obtained with the Hololens operational in the MRI room. ROI and resulting masks of a scan made under these conditions can be found in Fig. 34. For both conditions, the phantom mask contains 8807 pixels and the background mask contains 2500 pixels in total.

TABLE 12: Overview of MRI scan settings used to obtain the phantom scans. 'z' in the 'ImagePositionPatient' parameter set refers to the variable slice position in the image sequence, where -34.2000 $\leq z \leq$ 34.2000 mm.

Setting	Value	Setting	Value
Format	'DICOM'	**NumberOfAverages**	1
FormatVersion	3	**ImagingFrequency**	63.6415
Width	192	**ImagedNucleus**	'1H'
Height	192	**EchoNumbers**	1
BitDepth	12	**MagneticFieldStrength**	1.5000
ColorType	'grayscale'	**SpacingBetweenSlices**	3.6000
ScanningSequence	'GR'	**PercentSampling**	100
SequenceVariant	'SK\SP'	**PercentPhaseFieldOfView**	100
ScanOptions	''	**PixelBandwidth**	500
MRAcquisitionType	'2D'	**FlipAngle**	15
SliceThickness	3	**ImagePositionPatient**	[-96;-81;z]
RepetitionType	1000	**ImageOrientationPatient**	[1;0;0;0;1;0]
EchoTime	2.3100	**SamplesPerPixel**	1

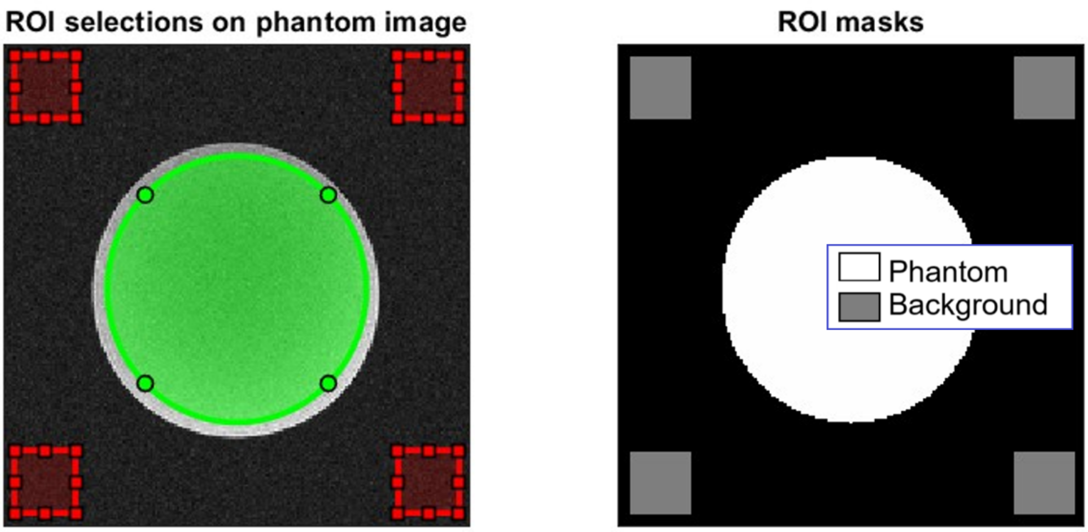


FIGURE 33: ROI and resulting masks used to differentiate between the phantom and background signals of the baseline scan. The 10th scan in the sequence is shown in the background of the ROI selections.

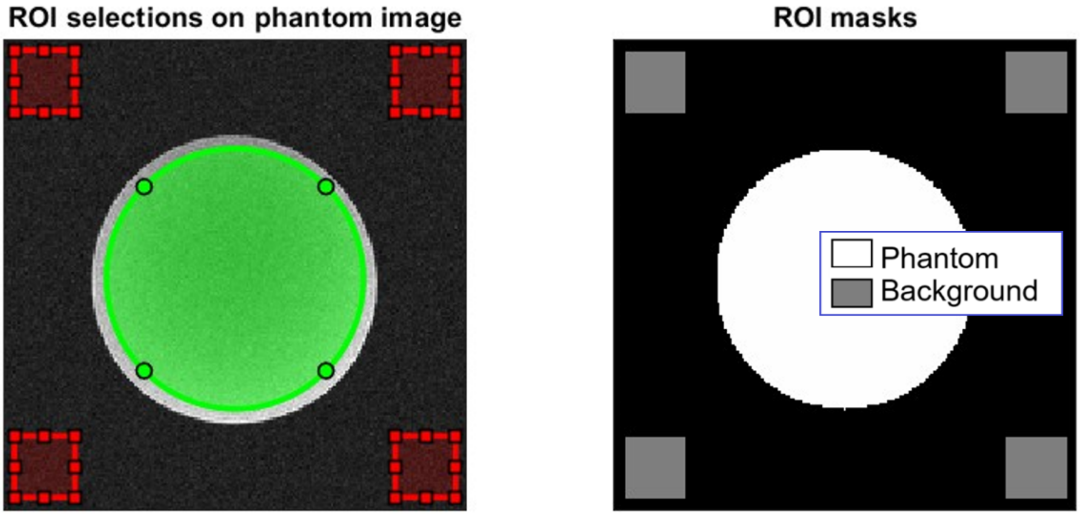


FIGURE 34: ROI and resulting masks used to differentiate between the phantom and background signals of the scan made under influence of the Hololens. The 10th scan in the sequence is shown in the background of the ROI selections.

D.2 Latency

Latency of every step in the data transmission between MRI scanner/simulator and Unity Editor/Hololens application. Image latency is shown in Table 13, command latency is shown in Table 14. The DICOM metadata time is defined with millisecond precision, whereas time recorded by the desktop PC, laptop and Hololens has microsecond precision.

TABLE 13: Image latency table showing all transmission steps. The value shown in red was originally measured to be -81.8 ms, and has been replaced by a separately determined value.

	made - sent [ms]	**sent - received [ms]**	**received - processed [ms]**
MRI simulator → Unity Engine	808 ± 186	5.79 ± 2.59	39.8 ± 6.3
MRI simulator → Hololens	6522 ± 2363	304 ± 43	231 ± 10
MRI scanner → Unity Engine	584 ± 104	5.23 ± 161.09	38.6 ± 6.6
MRI scanner → Hololens	999 ± 177	155 ± 41	228 ± 11

Table 14: Command latency table showing all transmission steps.

	sent - queued [ms]	**queued - retrieved [ms]**	**retrieved - processed [ms]**
Unity Engine → MRI simulator	269 ± 118	104 ± 4	65.5 ± 20.5
Hololens → MRI simulator	13.4 ± 134.3	110 ± 29	71.2 ± 29.9
Unity Engine → MRI scanner	357 ± 139	102 ± 4	159 ± 37
Hololens → MRI scanner	208 ± 179	95.1 ± 3.6	205 ± 72

D.3 User-friendliness and potential

Section 4 shows results from the user experience questionnaire averaged per category. Table 15 shows the averages per multiple-choice question. Scores between 0.00 and 1.00 were awarded to represent answers between 'Strongly disagree' and 'Strongly agree' with positive statements about the system functionality and potential.

Table 15: Average scores per survey question. The question numbers correspond to the questions shown in Table 11 in Appendix C.

Question number	**Score**
1	0.86 ± 0.13
2	0.82 ± 0.19
3	0.86 ± 0.13
4	0.79 ± 0.09
5	0.89 ± 0.13
6	0.79 ± 0.17
7	0.79 ± 0.17
8	0.96 ± 0.09
9	0.68 ± 0.24
10	0.86 ± 0.24
11	0.79 ± 0.17
12	0.86 ± 0.13
13	0.89 ± 0.13
14	0.79 ± 0.22
15	0.86 ± 0.13
16	0.64 ± 0.28
17	0.86 ± 0.13
18	0.93 ± 0.12