UNIVERSITÀ POLITECNICA DELLE MARCHE

FACULTY OF ENGINEERING

MASTER DEGREE IN BIOMEDICAL ENGINEERING

Development of a Mixed Reality Application for Maxillofacial Surgery for Varjo XR3 to be compared with HoloLens2.

Supervisor:

Prof. Marco Mandolini

Graduating student:

Martina Dichiara

Co-supervisors:

Dr. Agnese Brunzini

Dr. Paolo Senesi

A.A. 2022/2023

I. INDEX

II. Figure's List

III. Table's List

IV. Abstract

Traditional approaches for maxillofacial surgery involves the use of 3D printed surgical guides and resin dental splints to assist surgeons during operations. These tools need to be customized for patients, thus requiring a lengthy 3D modelling process and being quite expensive. To overcome these limitations, an innovative technology like mixed reality can be introduced. Mixed reality allows blending virtual elements and real-world elements in a single hybrid environment. In the context of maxillofacial surgery, mixed reality can be used to visualize surgical guides on the patient's anatomy to aid surgeons. This thesis builds upon a previously developed augmented reality application for the Microsoft HoloLens 2 headset, enabling the visualization of cutting lines for maxillary osteotomy and the repositioning of the osteotomized bone fragment. The goal of this work is to develop a mixed reality application for the Varjo XR-3 headset that replicates the same functionalities as the previous one and integrates the visualization of cutting lines for mandibular osteotomy. The application, therefore, allows to choose between three modes: the maxillary osteotomy mode, where cutting lines and fixation holes are displayed, the mandibular osteotomy mode, which shows cutting lines on the mandible, and finally, the mandibular repositioning mode, which presents holograms guiding the surgeon in the movement of the maxilla until reaching the target position. For the development of the application, Unity and Visual Studio were used. Then, to test the application, a 3D model of a skull was created in Rhinoceros and subsequently printed using the ZPrinter 450, utilizing Color Jet Printing technology. Seven people tested the application for maxillary and mandibular osteotomies. The test involved tracing lines and holes with a pen following the holograms displayed on the phantom's anatomy. The tested geometries were then scanned using the Go!Scan3D scanner to reconstruct the traced cutting lines and drilled holes in Rhinoceros. Finally, quantitative results were extracted in CloudCompare. The same users were then asked to repeat the maxillary osteotomy using the HoloLens 2. In fact, another goal of this project is to compare the two headsets and the two developed applications. Users were subjected to two types of questionnaires, the User Experience Questionnaire, and the User Experience Survey, to evaluate maxillary osteotomy of the two applications and the overall experience wearing the two different headsets.

The quantitative results show that the average distance between the curves traced by the seven users and the nominal curves is too high to be considered significant data. To compare the results obtained in this study with previously collected data using the HoloLens 2, the standard deviation was considered, showing comparable results for genioplasty and higher values for all other types of osteotomies considered. This implies that not only the holograms are displayed in an incorrect position, but there is also some difficulty in tracking the displayed lines. The qualitative results of the User Experience Questionnaire show slightly higher usability for the maxillary osteotomy mode tested with Varjo XR3 compared to HoloLens2, while the results of the direct comparison between the two headsets, extrapolated from the User Experience survey, indicate that overall, HoloLens 2 is preferred over Varjo XR3 by the seven users who participated in the tests. So, critical issues have been highlighted in both the developed application and the chosen headset, which require improvements and adjustments to fully explore the potential of mixed reality in the field of maxillofacial surgery.

1. Introduction

Maxillofacial surgery is a field of surgery that deals with correcting defects and anomalies in the jaw and facial bones. A subset of maxillofacial surgery is orthognathic surgery, which focuses on the lower facial structure and airways issues, temporomandibular joint dysfunction and malocclusions resulting from asymmetries, and disharmonies in the jaw. The primary procedures performed in this type of surgery are osteotomy and repositioning. Osteotomy involves cutting the bone structures, while repositioning is the procedure that follows osteotomy, enabling the realignment and reorientation of bone fragments to correct the anatomical defect. The surgical procedure is preceded by acquiring the patient's anatomy for case study and the development of a pre-operative plan. This phase is traditionally followed by the creation of customized surgical guides for osteotomy and dental splints for repositioning using CAD software. These guides and splints are then 3D printed and used in the operating room to assist the surgeon. Nonetheless, the modelling and fabrication of these instruments entail a considerable amount of time and high costs, and they may not readily accommodate alterations to the pre-operative plan. To address these issues, innovative technologies such as mixed reality are emerging. Mixed reality allows the blending of real and virtual environments into a unique environment. In the surgical context, it can be a valuable tool for guiding procedures through virtual images. Virtual images must be correctly positioned onto real-world elements, which requires robust registration methods. This can be achieved either through the use of markers or through algorithms that can recognize features of real-world elements. The visualization of both real and virtual content relies on output devices. In most cases, these are head-mounted displays, which can be categorized into two main types: optical see-through and video see-through. The main difference between these two types concerns how real-world content is perceived. Optical see-through devices have lenses that enable direct viewing of the external environment, while video see-through devices use cameras to capture the external surroundings and then present it on monitors positioned in front of the eyes. The use of mixed reality in the field of surgical procedures is still in its early stages, but it has already been applied in some contexts. The starting point for this work is an already developed augmented reality application that allows for the visualization of cutting lines for the osteotomy and feedback for the repositioning of the maxilla through the Microsoft HoloLens 2. [1] In this thesis an analogous mixed reality application is developed to assist maxillofacial surgeons in maxillary and mandibular osteotomy and maxillary repositioning, utilizing another device, the Varjo XR3. The developed application utilizes three markers and offers a choice between two modes: osteotomy and repositioning. In the osteotomy mode, cutting lines for bone structures and holes for the final fixation of the bone fragment are displayed. The user can choose between maxillary and mandibular osteotomy. In the second mode, repositioning, the user views the final position established during the pre-operative plan and interactive feedback that guide the placement of the bone fragment in the aforementioned position. The application is then tested on a 3D-printed skull phantom, and the results are extracted for the validation. Additionally, the result extraction is aimed at a qualitative comparison with the augmented reality application through questionnaires. The hardware used for this project include the Varjo XR3, the ZPrinter450 printer to obtain the physical model for the testing, and a 3D scanner, Go!Scan3D, for result extraction. The software utilized comprises Rhinoceros7 for 3D modelling of the digital skull phantom, Unity for application development, Visual Studio for scripting to enable additional functionalities to the application, Varjo Base, the specific software for the device management, and finally, CloudCompare for extracting statistical results for comparison with those obtained through the use of HoloLens2. The accomplishment of this work required an in-depth study of the previously developed application and the differences between the two headsets. Starting from an understanding of these differences, it was possible to determine what could be replicated and what needed to be redesigned and how in order to achieve the objective. Mixed reality is a relatively new technology that is becoming increasingly utilized in the field of medicine and surgery. Some examples of applications indeed exist but are nearly entirely developed for HoloLens. The Varjo XR3 device is capable of providing a superior immersive experience, with a wider field of view and extremely high visual performances. For this reason, it has been considered extremely interesting to introduce this device within the field of maxillofacial surgery. To the best of my knowledge, there are no other studies of this kind reported in the literature.

In conclusion, the novelty of this study lies not only in the development of a mixed reality application in the field of maxillofacial surgery, which is still in its early stages, but also in the introduction of an exceptionally powerful device to assess if it could be a valuable and viable alternative in real surgical scenarios, identifying the limits and the potential improvements to ensure the introduction of this technology in actual surgical procedures.

2. State of the Art

In this section, all the necessary topics for understanding the foundational aspects of this thesis work are presented. Maxillofacial surgery is introduced, outlining the primary procedures and the traditional approach to this subject. The presentation then transitions to mixed reality, providing an explanation of the fundamentals of mixed reality technology, its key components, and an overview of the devices currently available on the market. Following that, a literature review on the application of mixed reality in the medical field is conducted, followed by an in-depth exploration of its use in maxillofacial surgery. Finally, a brief description of Color Jet Printing technology used in this thesis for the printing of the physical prototype and markers is provided.

2.1 Maxillofacial surgery

Maxillofacial surgery refers to a special kind of treatment that involves the face, head, and oral areas, from the jaw to the teeth and the neck. The definition of Maxillofacial Surgery given by the American Dental Association is reported:

"Oral and Maxillofacial Surgery is the specialty of dentistry which includes the diagnosis, surgical and adjunctive treatment of diseases, injuries and defects involving both the functional and aesthetic aspects of the hard and soft tissues of the oral and maxillofacial region."

The complex anatomical structures in the oral and maxillofacial region are involved in various physiological functions including speech, mastication, swallowing and breathing, not to mention facial aesthetics. In the treatment of diseases affecting this region, clinicians must consider the preservation and rehabilitation of vital organs and functions; hence, the emergence of a functional surgery approaches. Functional surgeries can be either preservative or reparative. [2]

There are many reasons a person may need maxillofacial surgery, including the treatment of certain health conditions, such as: jaw misalignment, tumours or cysts, cancer, facial injury or trauma, obstructive sleep apnea, cleft palate/lip, craniosynostosis, temporomandibular joint (TMJ) disorders, severe infection, broken bones, bone grafts.
Most common maxillofacial surgeries are Orthognathic surgery, Dental Implantology, Arthroplasty for Temporomandibular Joint (TMJ), Cleft Palates/Lips surgery and Surgery for Facial Trauma.

Orthognathic surgery is used to correct a wide range of skeletal and dental problems, including malocclusion, facial asymmetry, and obstructive sleep apnea. It involves repositioning the upper jaw, lower jaw, or both to improve the alignment of the teeth and the overall facial structure. The surgery involves cutting the jawbone and moving it to its new position. During the surgery, the oral and maxillofacial surgeon makes incisions inside the mouth to access the jaw and facial bones. The bones are then repositioned and stabilized using surgical plates and screws. Orthognathic surgery can be performed on both the upper and lower jaw.

Temporomandibular joint (TMJ) surgery is a type of maxillofacial surgery used to treat TMJ disorders, which affect the joint connecting the jawbone to the skull. Disorders of the mandibular joint can cause pain, clicking, and difficulty in opening and closing the mouth. During the surgery, the oral and maxillofacial surgeon will make an incision in the skin and muscle tissue surrounding the TMJ to access the joint. The surgeon may then reposition or replace the cushioning disc within the joint or remove damaged tissue. [3]

Dental implantology is a common surgical procedure used to replace missing teeth. Dental implants are artificial tooth roots that are surgically inserted into the jawbone to support a dental prosthesis, such as a crown or a bridge. During the surgery, the oral and maxillofacial surgeon makes an incision in the gum tissue and creates a hole in the jawbone to place the implant. The implant is then allowed to fuse with the surrounding bone tissue in a process called osseointegration. Once the implant has fully integrated, a dental prosthesis is attached to the implant.[4] [5]

Palates/Lips cleft surgery is a type of maxillofacial surgery used to repair a congenital defect where the lips, palate, or both do not properly fuse during fetal development. This defect can cause a wide range of functional and aesthetic issues, including feeding difficulties, speech problems, and facial asymmetry. The procedure is typically performed when the child is between 6 and 12 months old and involves surgical closure of the labiopalatine cleft. The surgery may involve using local tissue flaps or transplanting tissue from other parts of the body. [6]

Facial trauma surgery is a type of maxillofacial surgery used to repair injuries to the face and jaw, such as fractures, lacerations, and soft tissue injuries. These injuries can be caused by a wide range of accidents, including falls, sports injuries, and car accidents. The procedure begins with a thorough evaluation of the patient's medical and dental history, along with a physical examination of the affected area. The surgery may involve the use of surgical plates and screws to stabilize the facial and jaw bones, or the transplantation of tissue from other parts of the body to repair soft tissue injuries.

2.1.1 Osteotomies and repositioning

Focusing on orthognathic surgery, that is a major surgery in this field, the two basic steps are the osteotomy and the repositioning of the bony fragments. An osteotomy is a surgical procedure that involves the cutting and removal of a portion of a bone. Osteotomies are preferably conducted intraorally to avoid scars on the face while ensuring sufficient access to the underlying skeletal structures.

Maxillary osteotomies are performed to correct upper jaw deformities and open bites. Cuts are made through the gum above the upper teeth to access the jawbone, and then the upper jaw is cut using small saws to fracture the bone in a controlled manner. [7] The most common maxillary osteotomy is LeFort I. Figure 1.

Mandible osteotomies, instead, are performed to correct receded mandible or open bites. In this case cuts are made through the gum behind the back teeth to gain access to the jawbone, lower jaw is then cut with small saws. [7] Most common mandibular osteotomies are Bilateral sagittal split osteotomy (BSSO) (Figure 2), Mandible Angle Split Osteotomy (MASO) and Genioplasty. Figure 3. BSSO correct retrognathism and mandibular prognathism, MASO corrects prominent mandibular angles, while Genioplasty corrects disharmony in the chin, by reducing or augmenting it.

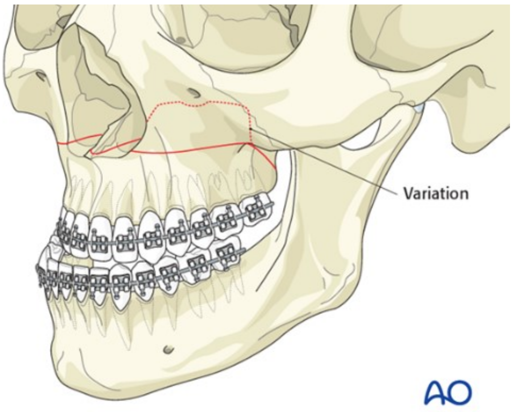


Figure 1 Cutting lines for Le Fort I maxillary osteotomy.

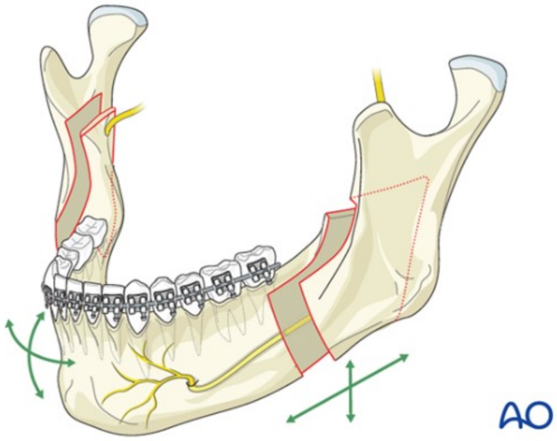


Figure 2 Cutting lines for BSSO mandibular osteotomy.

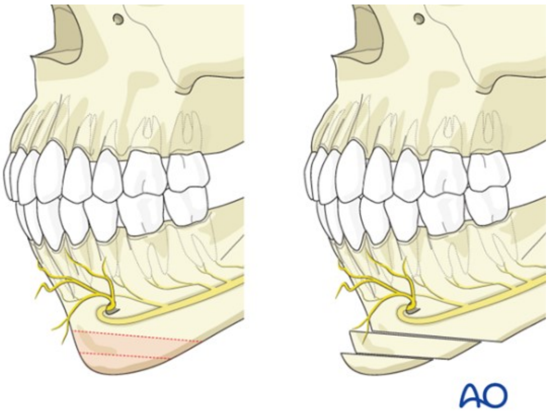


Figure 3 Cutting lines for genioplasty.

After the osteotomy, the surgery is completed with the repositioning phase. This surgical step involves the precise adjustment and realignment of the cut bony fragment to correct the intended defect. It involves the translation and rotation of the fragment along the roll-pitch-jaw axes for precise alignment. [8] Figure 4.

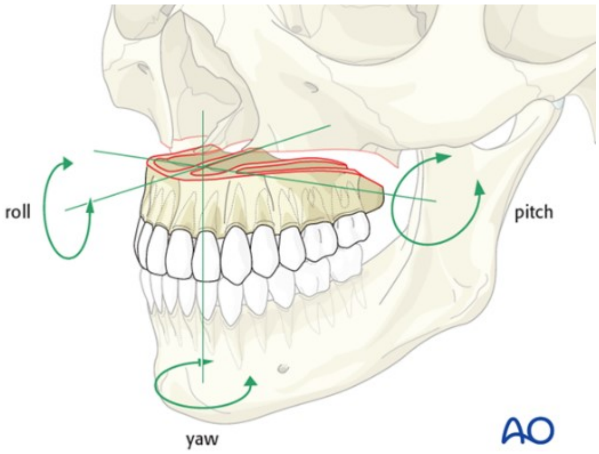


Figure 4 Moving axis for the maxillary repositioning.

2.1.2 Surgical Planning

Before proceeding with any type of maxillofacial surgery, the surgeons need to plan in detail the surgical procedure. Surgical planning is a preoperative method of visualizing a surgical procedure in order to predefine the surgical process itself. It has traditionally been carried out using 2D technology and 2D cephalograms. These techniques have limitations due to the fact that during the execution of the surgical planning, two-dimensional cephalometric images are not capable of providing comprehensive information about the three-dimensional anatomical structures of the face, which prevents the preoperative prediction of bone positions following osteotomies. Recent technological advancements have led to a wide variety of three-dimensional methods that allow for the acquisition in three dimensions of both surface structures and skeletal foundations within the space. These methods include intraoral arch scanning, Computed Tomography (CT), Cone-Beam Computed Tomography (CBCT), Magnetic Resonance Imaging (MRI), Fluoroscopy, etc. The medical image obtained as output of those methods is then translated to Digital Imaging and Communication in Medicine (DICOM) format to obtain volumetric data. Then, multiple software are available for the patient specific implant design and the virtual 3D planning of the surgical procedure, allowing an interaction with 3D images to simulate surgery and visualize the prediction of postoperative outcomes in soft and hard tissues. CAD tools allows to generate, shape and scale, analyse and hence optimize the design of a surgical instrument or object on the basis of the acquired three-dimensional anatomy. CAE tools allow to simulate, test, evaluate and possibly optimize the performances of the designed model or production processes before manufacturing. Rapid prototyping (RP), instead, includes different techniques used to fast fabricate a scale model of an object starting from 3D CAD data. Phantoms permit to test and validate functions, applications, and designs onto the patient data before testing them on real patients. Both models and implants are 3D printed, sterilized, and subsequently used for surgery. Most recent advancements include the use of Mixed reality (MR) for further evaluation and simulation before surgery and to assist the surgical team during surgery, allowing to facilitate data manipulations, to provide guidelines during the procedures and interactive feedback regarding the status of the surgical operation. [9]

3D Virtual Surgical Planning (VSP) includes the creation of a virtual occlusion, the placement of virtual osteotomy lines, and the virtual repositioning of the maxilla and mandible to an ideal position, based on real-time soft tissue simulation of the subsequent facial profile.

The transfer of the 3D VSP to the patient in the operating room is done, in the majority of cases, by using intermediate and final occlusal splints. The splints serve as an interoperative guide for the surgeon to position the maxilla and mandible at the planned location. [10] [11]

Surgical guides and splints can be designed and manufactured using computer-aided design/computer-aided manufacturing (CAD/CAM) technology, to avoid errors in the traditional model process that can lead to suboptimal outcomes. [12]
Surgical guides for the osteotomies are the lines that the surgeons must follow to properly cut the bones. As said, these lines are designed on a dedicated software and employed to generate custom-made CAD models of physical objects, that are then 3D-printed and used in the surgical operation. [13] Figure 5.
While the dental splints are those devices, usually made of resin, that help the repositioning of the bony fragments during the surgery. The dental splint is placed on the patient dental arch and then surgeon shift the cut bony fragment until teeth perfectly fit the splint. They are custom-made, starting from the registration of the subject bite that is then scanned to obtain the 3D model. The 3D model is given as input to CAD systems to design the appropriate dental splint that is finally manufactured. [14] Figure 6.

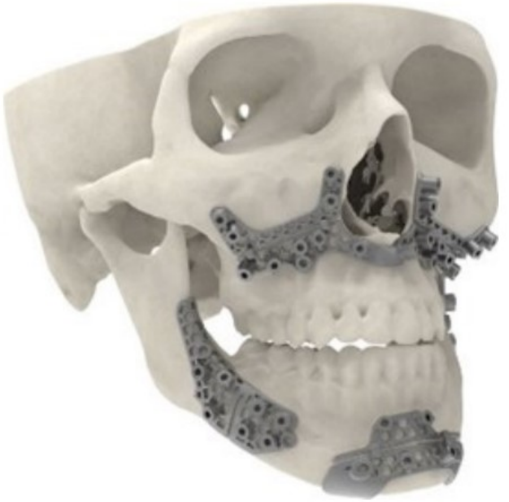


Figure 5 Printed surgical guides for maxillary and mandibular osteotomies.

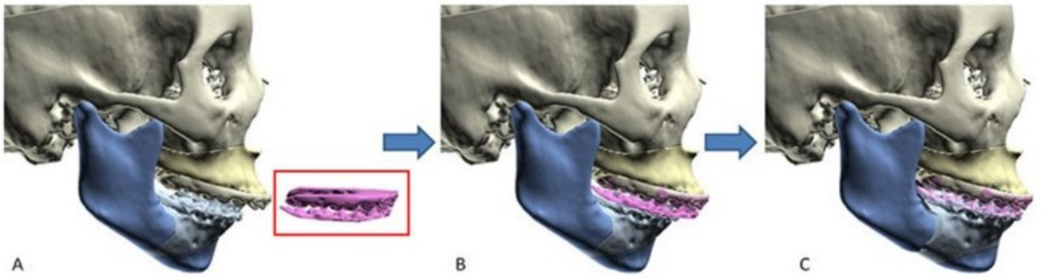


Figure 6 Dental splint example of usage.

Although 3D printed cutting guides and dental splints are extremely valid and affirmed in the planning and in the actual surgical procedure, some drawbacks exist. The long-time needed for the design and deliver of those products is not suitable for emergency situations. Moreover, this technique has high costs especially when multiple guides are required to treat the clinical case or when they need to be improved or adjusted due to inter-operative changes of the surgical plan. [15]

A suitable and innovative alternative lies in mixed reality technology. The idea is to replace the physical cutting guides and project the lines directly on the anatomy of the patient to guide the surgeons. Also, in this case there is the need for CAD software to draw and precisely place the lines on the right scale. Then a MR application need to be developed and the surgeons need to be equipped with a Head Mounted Display (HMD) to see the patient with the lines directly projected on the anatomy. MR can be used also to avoid the need for physical dental splints, the optimal position can be defined during the preoperative planning process and directly displayed on the HMD worn by the surgeon, that will move the bony fragment in accordance with the provided indications.

Once the planned position for the jaw is reached, the bony fragments need to be fixed and secured to complete the surgical procedure and to allow the healing of the treated structure. The standard in this field involves the usage of fixation plates and screws. To provide stable fixation various plating system have been developed. The gold standard material for fixation plates and screws is titanium. The mechanical properties of titanium as its strength, the ease of handling, the lack of dimensional changes, the minimal scatter on CT scanning, and the compatibility with radiography and magnetic resonance imaging led to its widespread adoption. As the need for fixation is only temporary and as metallic materials cause stress shielding of the underlying bone, these plates are often removed after the maxillofacial bony healing. For this reason, the usage of biodegradable or bioresorbable fixations is increasing. Biodegradable osteofixation implants are characterized by biomaterials that disintegrate after implantation with no sign of elimination from the body. The biodegradation process depends on contact with body fluids, temperature, motion, molecular weight, the crystal form and geometry of the material, and the tissue that is implanted. These materials are safe, effective, and sufficiently flexible for use at many maxillofacial bony surgical sites. Common constituents include polyhydroxyl acids: polymers and copolymers of PLLA, poly-d-lactic acid (PDLA), polyglycolic acid (PGA), and polydioxanone sulfate. Developed bioresorbable osteosynthesis materials comprise a composite of PLLA as a base, an osteoconductive material such as hydroxyapatite, and an accelerator of bioresorption such as polyglycolic acid. [16]

2.2 Mixed Reality

Mixed Reality is a technology that merge physical and virtual worlds to build a new environment. The concept of Mixed Reality was originally introduced in 1994 by Milgram and Kishino in relation to the "virtuality continuum". Figure 7. The virtuality continuum shows a range of environments from the real world to the virtual world, establishing the continuity between these two worlds. At one end of the continuum there is the real environment, while on the opposite extremum there is the virtual environment. The space in between the two extremes, where real elements start to mix with virtual elements, is the Mixed Reality. The actual definition proposed by Milgram and Kishino of MR is:

"The most straightforward way to view a Mixed Reality environment, therefore, is one in which real world and virtual world objects are presented together within a single display, that is, anywhere between the extrema of the virtuality continuum." [17]

In the MR space, different degrees of reality-virtuality are present, so, further definition were introduced such as Augmented Reality (AR) and Augmented Virtuality (AV).

The Augmented Reality environment is a physical environment whose elements are "augmented" by computer-generated objects that aims to improve our perception and understanding of the real worlds. This is achieved by superimposing virtual information on our vision of the real world. [18]

The concept of Augmented Virtuality (AV) refers, instead, to a computer-generated virtual environment where real-world objects are integrated, enriching the virtual world with elements from the real world. [18]

To provide a comprehensive description of the virtuality continuum, physical reality expresses the real environment with its tangible components, allowing users to interact directly with it through their five human senses. [19] In contrast, Virtual Reality constitutes a purely computer-generated environment.

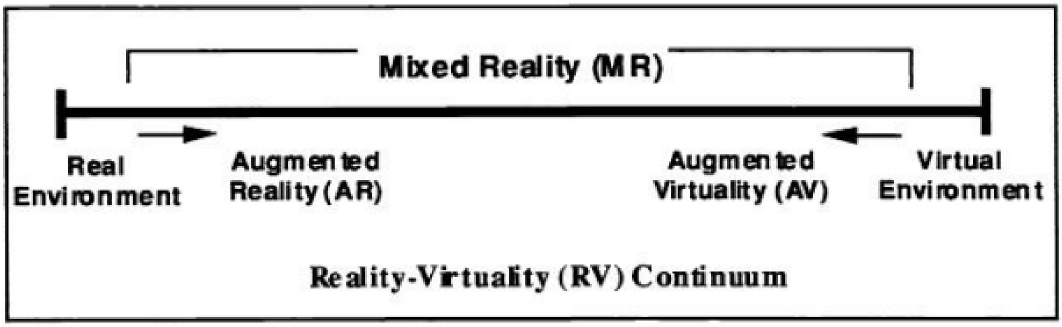


Figure 7 Reality-Virtuality Continuum.

2.2.1 Technologies

The overall architecture of a mixed reality system is complex and involves the integration of hardware, software, sensors, and sophisticated algorithms to blend real and virtual environments on a single display, or on multiple display for collaborative mixed reality. To create the MR environment the following steps are required:

1. Virtual content creation: to mix real and virtual world, first of all the virtual content need to be created in dedicated software.
2. Calibration: The calibration procedure is aimed to compute the transformation that enables the virtual objects to be represented in the same coordinate system as the real-world objects.
3. Space recognition: the implementation of a MR scene requires the modelling of the external real environment. The modelling of the space includes the modelling of large areas as rooms and the modelling of small tangible objects.
4. Object recognition and Tracking: the identification and the tracking of real objects allow to add a virtual content in the real environment to create a MR scenario.
5. Registration and Mapping: the interaction between real and virtual require the mapping and registration of the virtual model in the real space.

6. Visualization and Rendering: to display the MR content, appropriate algorithms are required. Those algorithms must add virtual object to the video images to enrich them, the virtual objects need to be as mush a possible close to the reality and this is achieved by means of lighting and shadows, the level of detail of the object must vary according to the position of the subject, objects that are not in the visible field of view need to be switched off , etc. [20]

2.2.1.1 Components

To accomplish all the steps cited before, there is the need for input devices, output devices, a computing hardware and software toolkits.

According to Ohlenburg, an input device is a combination of a hardware component and a software component, able to send data into the system, based upon input from reality. [21] Input devices are all those peripherals that sense and collect the user data and actions and allow the user to communicate with the application. The main input devices can be categorized into:

1. Desktop and Mobile Devices: mouse, joystick, trackpad, keyboard used to navigate the application.
2. Tracking devices: these devices detect the position and orientation of either the user's body parts or real-world objects, ensuring the alignment and consistency between the real and virtual worlds. Such devices encompass a variety of sensor types, including inertial sensors, optical sensors, acoustic sensors, magnetic sensors, mechanical sensors, and GPS.
3. Depth sensors: acquire three-dimensional images with a real-time compatible frame rate. Each pixel of the resulting image at each frame corresponds to the distance of the corresponding object to the sensor. In this manner is possible to extract a point cloud of the scene. The most common are the stereo-base, the structured-light and the Time-of-Flight depth sensors.
4. Direct Human Input devices: signals emitted directly by the user are acquired to control the system. The most common signal is the voice detected by microphones that is then processed by means of specific software for the speech recognition.

An output device is a combination of a hardware and a software component which is responsible to provide the user with the feedback computed by the system. [21] Output devices are able to present information to the user by stimulating one or more senses, visual, hearing, and touch are the most common but in very rare case also smell and taste are stimulated.

1. Visual display: most common output display, they are able to make the user perceive the mixed reality environment in three dimensions. These displays can be categorized in four classes: head-mounted display, hand-held display devices, monitor-based and projection-based displays.
2. Sound output devices: these devices allow to render the sounds generated by the system in the implementation of the virtual environment. They provide users with a more immersive experience and enhance the perception of a unified environment blending the real and the virtual. Two main auditory devices can be found: stereophonic headphones placed on user's ears and external speakers placed at different locations in the real world.
3. Haptic devices: allow the user to "touch" the virtual environment by sending haptic feedback. Feedback can be of two types: kinesthetic, to feel the force, and tactile, to feel textures, temperatures, or vibrations. Devices that allow the haptic feedback are data gloves, hand exoskeletons, game controllers that have some mechanical actuators and vibrotactile stimulators.
4. Smell output devices: very uncommon devices that try to introduce olfactory stimulations to increase the user immersion in the MR environment. An example was proposed by Nakamoto et al that presented a device composed by multiple solenoid valves able to blend 32 smells. [22]
5. Taste output devices: unusual devices to include the taste for a more immersive experience. Aminzade [23] proposed for example the BeanCounter, a discrete gustatory device that can dispense jellybeans with different flavours. Some researchers have attempted to simulate taste without the use of dedicated taste output devices. For instance, Narumi et al. [24] introduced the 'Pseudo-gustation' method, which alters the perceived taste of food by modifying its appearance and scent within an AR application. This approach capitalizes on the understanding that our perception of taste is influenced by our sense of smell and sight.

2.2.1.2 Tracking

The tracking of real-world objects is one of the crucial steps to ensure the visualization at the correct position of the virtual content in the real environment. Tracking systems can estimate the absolute or relative position and orientation of the objects. The pose of a rigid body in the space is described by six quantities, three translations and three rotations along or around the three axes x, y, and z. A tracker may monitor all six motions, a subset of them or a limited range of a variable.

Tracking methods can be subdivided into two categories: sensor-based or vision-based. Tracking in sensor-based systems is carried out by sensors that are able to detect position and/or orientation of an object, while vision-based methods use image information to this purpose.

Sensors give information to the processing unit over time by measuring coordinates and orientation of real environment. Example of sensors are optical, inertial, GPS, magnetic, or acoustic.

Optical tracking uses a video camera, that may be visible light or infrared type. For 3D tracking at least two cameras are required with direct light without occlusion. Markers are necessary and serve as reference points. When the camera seek them, specific algorithms reconstruct the position. They can be either active or passive. Active markers detect the infrared light, while passive markers are made by highly reflective material that reflect the incident light. [25] Cameras can be placed in the environment while the markers are on the headset, this is the case of outside-in tracking. Alternatively, a camera can be placed on the MR device that look at the environment, this is the inside-out method. A popular example of optical inside-out method is called SLAM (Simultaneous Localization and Mapping), in which sensors are able to produce a real time map of the environment.

Inertial tracking uses accelerometers and gyroscopes that measure, respectively, linear acceleration and angular velocity. Magnetic tracking involves magnetic field emitted from a transmitter, this induces current in a receiver according to distance and orientation.

GPS relies on signals transmitted by satellites to calculate positions, employing mathematical methods such as time of flight and triangulation. However, this technology is primarily suitable for outdoor use because satellite signals are obstructed by buildings, and they are not precisely detectable indoor.

Acoustic tracking is based on the transmission of ultrasonic frequency sound that is bounced off objects and then detected by acoustic sensors. The position and orientation are computed either measuring time-of-flight or acquiring phase information. [26]

Vision-based tracking systems use images from cameras and rely on image processing techniques for the tracking . These tracking systems can be categorized as marker-based or marker-less. In marker-based tracking, predefined fiducial markers are placed in arbitrary locations. When these markers are detected by the camera, they guide the computer algorithm in the tracking process. On the other hand, marker-less tracking techniques seek natural elements in the real world for tracking. Initially, the camera calculates its pose based on known natural features. Subsequently, the system dynamically identifies additional natural features and uses them for continuous pose calculation updates. [27]

2.2.1.3 Registration

Registration is the process of aligning in real-time the virtual content on the real environment, to create a new and unique mixed environment.

The simplest approach is the manual registration, in which the user sees the hologram and position it through software controls. This procedure, however, require additional time and introduce human error impacting the accuracy.

Automate registration is carried out by computer vision algorithms that require the use of head mounted cameras. The general principle of computer vision tracking approaches is that a 2D image or 3D object with known dimensions is tracked by a digital camera, and a software algorithm determines the position and orientation of the tracking target relative to the camera. The computer vision capabilities are employed for registrations by placing targets at known locations. Fiducials markers or infrared markers can be used as reference for the positioning of the virtual content in relation to the position of the user. [28]

The accuracy of registration and the stability of visualized virtual content in the correct position are fundamental requirements for ensuring the quality of the mixed reality experience, particularly in critical fields such as surgery.

2.2.1.4 Display

A display is an essential element to visualize real and virtual together when developing a MR application. There are various examples of display, such as simple screens (PC monitors, televisions, smartphones, and tablets), multiple display systems (CAVEs) or Head Mounted Displays.

They have different characteristics that impact the performances and the level of immersion of the user. The main characteristics that are evaluated are:

- Field of View (FOV) is the angular extent of the observable world that is seen at any given moment on the display. The human field of view is about 160°.
- Field of Regard (FOR) is the physical space, measured in degrees, surrounding the user in which visual images are displayed.
- Spatial resolution is a measure of quality of the display, depends on the display size and resolution. It is expressed in dots per inch (dpi).
- Stereoscopy is a technique for creating or enhancing the illusion of depth in an image by means of stereopsis for binocular vision.
- Frame rate refers to the speed with which images are computed by the system and placed into the frame buffer. It is measure in Heartz.
- Refresh rate is the speed of the visual display device to refresh to display from the frame buffer. It is measure in Heartz.
- Head-based-rendering considers user's head position and rotation to create consistency between the virtual and the real world.

At present, Mixed Reality (MR) is predominantly experienced through Head-Mounted Displays (HMDs), which can be categorized into two types based on their display mechanism: optical see-through HMDs and video see-through HMDs.

Optical see-through HMDs (OST HDMs) allow a user to see the real world and the virtual world placing optical combiners, half-silvered mirrors, or beam splitters, in front of the user's eyes. Half-silvered mirrors are partially transmissive and reflective mirrors, so part of the light is transmitted while the remaining is reflected. Transmissive capability allows to directly see the real world, while reflective capability allows the perception of the virtual content. The virtual content is rendered on a 2D micro display outside the FOV of the user and light rays are redirected to the eye using a beam combiner. Lenses are placed between the beam combiner and the display to focus the virtual images on a semi-transparent surface of projection at a viewing distance comfortable for the wearer. [29] [30]

Video see-through HDMs (VST HDMs) make use of front-facing cameras to record and digitize the scene before combining it with virtual content. The mixed content is then shown to the user by means of two screens placed in front of each eye. In this case the user view is limited to the camera FOV, when the cameras are turned off the user is completely blind.

The block scheme of OST HDM and VST HDM are reported in Figure 8.

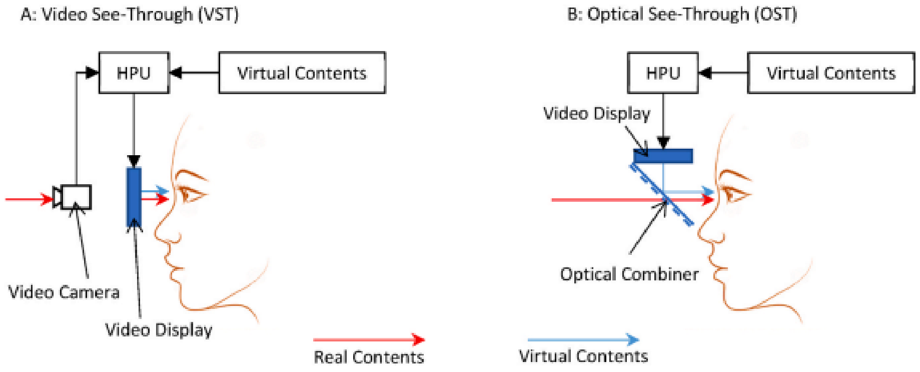


Figure 8 Block scheme of VST HDM (left) and block scheme of OST HDM (right).

Comparing these two technologies, a clear advantage of the OST approach is that the user can directly see the physical world, with natural stereo vision capabilities with no lag or loss of resolution associated with the real scene, and in case of display failure the user direct view is not compromised. Downsides are related to dynamic registration errors, latency when moving, static registration errors, complex calibration and issues related to virtual objects occlusion.

On the other hand, VST approach can offer a better registration between real and virtual contents. The real and virtual scenes are available in digital form and can be more flexibly merged. It introduces only negligible chromatic, temporal, and perspective alterations with respect to the naked eye, due to the fact that the point of view of the user is placed at the cameras' location. But the inevitable delay between actual motion and perception of the motion slows down the user activity and the resolution of the images depends on the resolution of the displays. In certain applications, such as surgical procedures, the potential loss of vision can be a critical concern. [30][31][32][33]

2.2.2 Head Mounted Display Present on Market

Due to the constant increase in popularity of mixed reality, various HMD have been launched on market. Leading vendors in this sector are Microsoft, Magic Leap, Varjo, HP, Acer, Samsun, ASUS. In this section, an overview of the most popular MR headsets is reported.

2.2.2.1 Microsoft HoloLens2

Microsoft HoloLens 2 (Figure 9) is a hands-free controllable computer with WLAN connectivity, it is controlled by voice and gestures, and it features full see-through holographic lenses, four visible light cameras, and two infrared cameras to provide mixed reality experience. The display has a resolution of 1440x936 per eye and a refresh rate of 60 Hz. [34]

Figure 9 Microsoft HoloLens2.

2.2.2.2 Magic Leap 2

Magic Leap 2 (Figure 10) is a OST, standalone, lightweight and ergonomic wearable device with built-in dynamic dimming technology and built-in technologies for running custom enterprise solution at scale. The display has 1440x1760 pixel resolution with a refresh rate of 120 Hz and 70° FOV. [35]

Figure 10 Magic Leap 2.

2.2.2.3 Varjo XR-3

Varjo XR-3 (Figure 11) is the professional-grade mixed reality solution available from the companies nowadays. It is a VST HMD, provides a realistic and true-to-life display through a 12-megapixel camera. It delivers images at human-eye resolution, with aspheric lenes and a 115° FOV. This headset includes Ultraleap hand-tracking to monitor hand gestures, allows to unlock depth awareness capabilities and has inside-out tracking capabilities for more immersive experiences. More detailed description is provided in Chapter 3. [36]

Figure 11 Varjo XR-3.

2.2.2.4 HP Windows Mixed Reality

HP Windows Mixed Reality headset (Figure 12) is a tethered device, equipped with built-in cameras and sensors that enable motion tracking and position detection, allowing users to interact with virtual environments and digital objects. It provides high quality visualization with high resolution images, 1440x1440 resolution per eye, and 90 Hz refresh rate with a FOV of approximately 100°. [37]

Figure 12 HP Windows Mixed Reality.

2.2.2.5 Acer AH101

Acer AH101 (Figure 13) is a tethered HMD which has two cameras on the front of the headset, that enables the spatial processing and the tracking of the headset's position relative to the computer. It uses a stereo display with a total resolution of 2,880 by 1,440, translating into a 1,440-pixel square for each eye, it delivers a 100° FOV and its refresh rate is 90Hz. The presence of the two cameras classifies the device as a mixed reality device, but it is primarily used for virtual reality application. [38]

Figure 13 Acer AH101.

2.2.2.8 Samsung HMD Odyssey+

Samsung HMD Odyssey+ (Figure 14) is a tethered, VST headset thanks to its two integrated cameras. It has dual 3.5" AMOLED displays, each with a 1440 x 1600 resolution, for a total of 2880 x 1600. It has a field of view of 110° and a refresh rate of 90 Hz. [39]

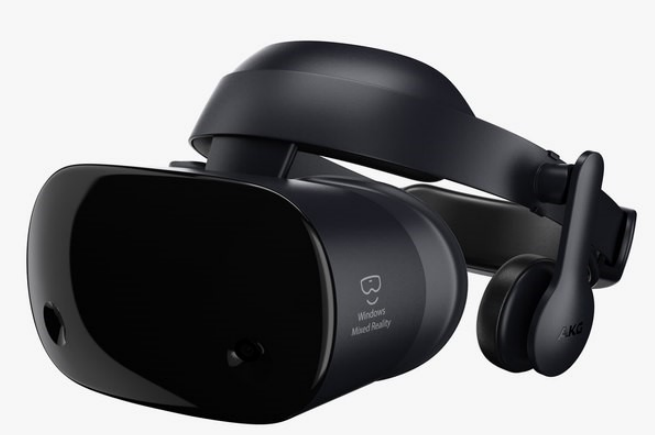


Figure 14 Samsung HMD Odyssey+

2.2.2.7 ASUS HC102 Windows Mixed Reality

ASUS HC102 Windows Mixed Reality (Figure 15) has two integrated cameras that allow the visualization of the external world and the inside-out tracking. The product includes a resolution of 2880 x 1440, as well as a 95° FOV and a 90Hz refresh rate. [40]

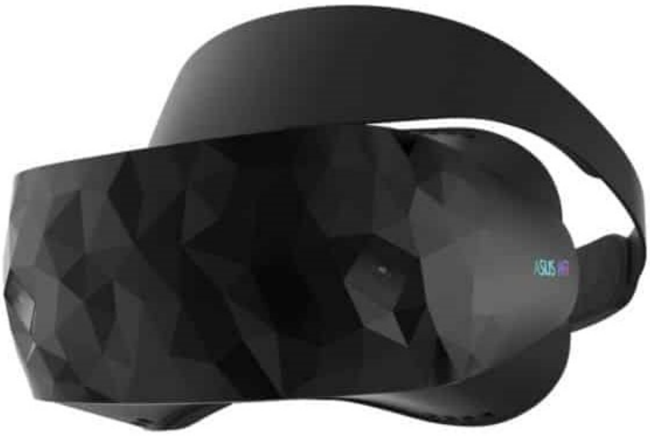


Figure 15 ASUS HC102 Windows Mixed Reality.

2.3 Applications of MR in the medical field

In recent years, MR have seen widespread application in society, most notably in the entertainment, engineering, and education industries. Also, the medical field has caught the potentials of these new technology. Even if MR in the medical filed is still at its infancy, the potential uses span the entirety of surgical practice and healthcare education. The increasing number of studies on MR-based approaches using head-mounted displays (HMD) has accelerated this development, and this trend is expected to continue in the coming years, paralleling the rapid technological advancements.

Main fields of application regards teaching, training, surgical planning, surgical aid and guidance, therapy, and telemedicine.

MR has opened a new era in the teaching and education of medical students and healthcare professionals. The 3D visualization of complex anatomy of the human body enables a deeper understanding and retainment of knowledge by the students. Certain applications allow the interaction with the anatomical structures, enabling users to scale, rotate, toggle layers, adjust contrast and transparency, and more. (Wu et al.) In other cases, the teacher/surgeon, while performing a surgical procedure wearing an HMD, can display its perspective on a monitor, and students can experience the procedure from a first-person viewpoint. [41] [42] Furthermore, the COVID-19 pandemic in recent years has heightened the demand for distance education, speeding up the integration of these technologies into the educational sector. [43]

Another significant application of MR in healthcare is in the training of surgeons. Through the immersive experiences made possible by Head-Mounted Displays (HMDs), realistic simulations of surgical procedures can be conducted. This allows for risk-free training in complex surgeries, enabling surgeons to build confidence in their skills without entering actual operating rooms or interacting with real patients. Surgeons can explore various strategies and discover the most effective approaches to successfully complete these procedures. Of particular interest is the simulation of emergency situations and mass casualty incidents. Users with MR can experience and practice an almost unlimited number of scenarios in which demands, and difficulty levels can be designed as needed. [44] [45]

MR has also proven to be a valuable tool for preoperative communication between doctors and patients. Through the 3D visualization, doctors can better explain the patient’s condition, the proposed treatment plan, and the surgical risks to the patient himself and to his family. This approach helps minimize misunderstandings and tensions, enhances the doctor-patient relationship, and alleviates apprehensions related to the surgical procedure.

Other applications, relates to the therapy and rehabilitation of physical and psychological conditions. Through immersive experiences fears can be faced in a controlled environment. Immersion represents the potential of technology to surround someone perceptually and to make them believe that the digital environment they are experiencing is “real”, then, following the therapist's guidance, the patient works on understanding the cognitive biases and thoughts that trigger their well-being, such as increased anxiety or panic. [46] Annaswamy et al. [47] tried to manage amputated patients with phantom pain. Virtual movements have been rendered onto a flat screen or a head-mounted display (HMD) to create the illusion of an intact limb to treat this condition.

Telemedicine represents another prominent field. The adoption of this innovative technology enables remote consultations, providing medical services to distant regions with challenges in accessing traditional healthcare facilities. Furthermore, remote communication facilitates collaboration among surgeons during surgical procedures or in the surgical planning. In fact, MR also plays a significant role in surgical planning. [48] Firstly, MR equips surgeons with a more comprehensive and intuitive understanding of the patient's anatomy, facilitating in-depth study and meticulous surgical procedure planning. Secondly, it supports surgeon cooperation, enabling them to integrate their knowledge and experience to formulate the most effective surgical plan.

Besides the surgical planning, MR is emerging also in the actual surgical procedure in the operating room. It can be employed to visualize subsurface anatomical structures in 3D from the surface without the need of great incisions, to consult a patient’s data through virtual 2D screens that display data collected before surgery or during surgery. Moreover, MR can be a valuable tool for the monitoring and tracking of instruments such as endoscopes. MR is expected to become a crucial tool for guiding a surgeon intraoperatively. The virtual objects, showing a patient’s data, can be consulted by a surgeon for decision-making and can also be moved around the operating room using mid-air hand gestures. The use of MR in surgical procedures allows to reduce the time required for the operation, to reduce bleeding and exposure to radiation when fluoroscopic imaging is required to monitor the progress of the procedure, to improve surgical accuracy and safety. Despite MR in surgical applications being in its early stages, some trials and experiments have been conducted, particularly in the fields of neurosurgery, maxillofacial surgery, urology, orthopedics, and generally in surgeries involving organs that exhibit minimal variations. This is because tracking and monitoring mobile organs pose significant challenges. In Table 1 examples of surgeries carried out with MR technologies are reported.

Table 1 Examples of Mixed Reality applications in surgeries.

Paper	Field	Procedure	Virtual content	HMD	Registration	Outcomes
Wei et al. [49]	Orthopedy – Spine surgery	Percutaneous kyphoplasty	Spinal 3D virtual images overlapped on the patient's body to assist the insertion of the needle	HoloLens	Manual - anatomical landmarks and confirmation with fluoroscopy	Reduced operating and fluoroscopy time
Liu Xiaoyang et al. [50]	Orthopedy – Spine surgery	Lumbar discectomy	Superimposition of 3D virtual images on the patients' anatomy to guide the insertion of the needle and positioning of the working sheath	HoloLens2	Manual - anatomical landmarks	Reduced operating time (103 ± 18 min vs. 116 ± 29 min, p = 0.01)
Gu et al. [51]	Orthopedy – Spine surgery	Percutaneous pedicle screw placement	3D virtual image of the target vertebra with the entry point of the needles and the direction of screws implantation on the patient anatomy	HoloLens2	Manual - anatomical landmarks and drawn-on markers	Reduced blood loss (382.27 ± 95.75 ml vs. 449.76 ± 91.69 ml, p = 0.014)
Gregory et al. [52]	Orthopedy – Shoulder arthroplasty	Reverse shoulder arthroplasty	Real-time display of patient's data combined with the data of the operative technique to compare at each step if what has been done is what should be done	HoloLens	Manual - anatomical landmarks	Similar operating time as traditional surgery (90 minutes)

Lei et al. [53]	Orthopedy – Hip arthroplasty	Total hip arthroplasty	Anatomical structure of the surgical area overlapped to the patient, visualization of osteotomy plane during the osteotomy and the location and size of the acetabular cup	HoloLens	Automatic - fiducial markers attached to pelvis via two K-wires	More precise individualized surgery without significantly increasing the operation time (4.5 hours)
Liu Shifeng et al. [54]	Neurosurgery – Brain surgery	Brain brachytherapy	Hologram with the tumour, peritumoral vessels location and six implantation needle paths aligned to the brain of the patient.	HoloLens	Manual - Anatomical landmarks	-
Li et al. [55]	Neurosurgery – Brain surgery	External ventricular drain insertio	Preoperative image and surgical plan, detailed access information such as the target point, entry point, and length of trajectory superimposed to their physical counterparts	HoloLens	Manual - Anatomical landmarks and adhesive skin markers	Additional time required for holographic guidance was 40.20 ± 10.74 minutes
Iizuka et al. [56]	Neurosurgery – Brain surgery	Resection of meningoencephalocoele	-	HoloLens	Manual - Anatomical landmarks	-
Kitagawa et al. [57]	General surgery	Laparoscopic cholecystectomy	3D models of gallbladder, cystic duct, common bile duct, common hepatic duct, cystic artery, and liver	HoloLens2	Manual	Similar mean operative durations (74 vs. 58 min) compared to standard technique, no blood loss
Aoki et al. [58]	General surgery	Laparoscopic hepatectomy	Needle guide	HoloLens2	-	The procedure required 285 min and involved the loss of 135 ml of blood
Wierzbicki et al. [59]	General surgery	Various open and laparoscopic	Preoperative CT or MRI data were transformed into 3D	HoloLens	Manual	Average operative duration of

		tumour resections	holograms and displayed in the surgeon FOV to better understand tumour size and location and setting of the needles			62.5 ± 21.2 min and inpatient stay of 5.5 ± 1.5 days
Li Guan et al. [60]	Urology	Laparoscopic partial and radical nephrectomy	Renal tumour and peritumoral tissue structure (including kidney, renal arteriovenous, collecting system, adrenal gland, liver, spleen, intestine, and bones) displayed	HoloLens	Manual – anatomical landmarks	Shorter operative time (60.7 ± 10.4 min vs 98.4 ± 11.7 min) and warm ischemia time and less estimated blood loss (15.5 ± 9.4 ml vs 45.9 ± 10.1 ml)
Li Guan et al. [61]	Urology	Laparoscopic partial nephrectomy	Kidney, renal vessels, renal collection system, tumour, renal capsule, and adherent perinephric fat area	HoloLens	-	Increased successful cases, shorten operating time (61.6 ± 14.2 min vs 100.5 ± 12.7 min), and decreased warm ischemia time and expected blood loss (23.4 ± 7.9 ml vs 43.7 ± 8.7 ml)
Porpiglia et al. [62]	Urology	Percutaneous nephrolithotomy	Renal vasculature (both arterial and venous), urinary collecting system, kidney shape, and kidney stone features	HoloLens	Manual – anatomical landmarks	Mean operative duration was significantly higher in MR-guided surgeries (median 108

						vs. 91 min, p = 0.02), while the rates of complications were similar (30% vs. 40%), and inpatient stay durations were the same (median 4) with respect to traditional surgery
Nuri et al. [63]	Plastic Surgery	Hemiglossectomy with anterolateral thigh (ALT) flap reconstruction	Vessels, muscles, and outer surface of the thigh with the reference points	HoloLens	Manual – Anatomical landmarks and drawn-on markers	-

2.3.1 Applications of MR in Maxillofacial Surgery

As said in the previous section, the use of MR is becoming more and more popular in the surgical procedures. Here a focus on the applications on maxillofacial surgeries is reported.

Widely applications are reported for the tumour resection. For example, in the context of head and neck cancer, traditional methods involve post-operative analysis of the surrounding tissues to be sure that the tumoral mass has been removed correctly, starting from this, Pepe et al. [64] proposed a mixed reality application able to display the geometry of the tumour superimposed to the patient's anatomy, by means of marker-less registration method to allow the surgeons to visualize intraoperatively if the tumour has been resected correctly. The geometry of the tumour has been reconstructed starting from the data acquired with PET-CT and then loaded on a MR application. For the superimposition of the virtual content with the anatomy, an algorithm was implemented to recognise facial landmarks extracted from the RGB camera of the HoloLens. Figure 16. The application was tested over a 3D-printed version of a patient CT scan, then the results were compared to an existing marker-based approach. The outcomes have revealed an error of − 4.5 ± 2.9 mm along the up-down dimension, an error of 3.3 ± 2.3 mm along the right-left dimension and, an error of − 9.3 ± 6.1 mm along the back-front dimension in the tumour resection. Moreover, the use of facial landmarks as markers appears to diminish the accuracy of the system. Tang et al. [65], instead developed an application to assist the tumour resection in which the preoperative plan was not superimposed to the patient's anatomy. The procedure involves the acquisition of anatomical data, through head and neck CT and MRI, that are used for the virtual planning of osteotomy planes. During the surgical procedure, after patient registration through laser surface scanning, the surgeon could visualize the preoperative STL model in the HoloLens and interact with the hologram using predetermined gesture controls. Before cutting, the surgeon used a handheld navigation probe to verify the surgical plan using predetermined reference points on the osteotomy lines. The mixed reality system had a 2 mm error tolerance, helping the surgeon quickly locate the precise position for the osteotomy lines. The pointer changed colour from red to green when the navigation probe was ≤2 mm from the reference point, ensuring accuracy. Through the 3D image and distance information displayed in the HoloLens, the surgeon could adjust the navigation probe's direction and position to confirm the osteotomy lines. Figure 17. Results were obtained comparing presurgical plan with CT scans taken one week after the surgery, and they revealed mean deviation of 1.68 ± 0.92 mm between preoperative virtual osteotomy plane and actual postoperative osteotomy plane, with the largest deviation being 3.46 mm.

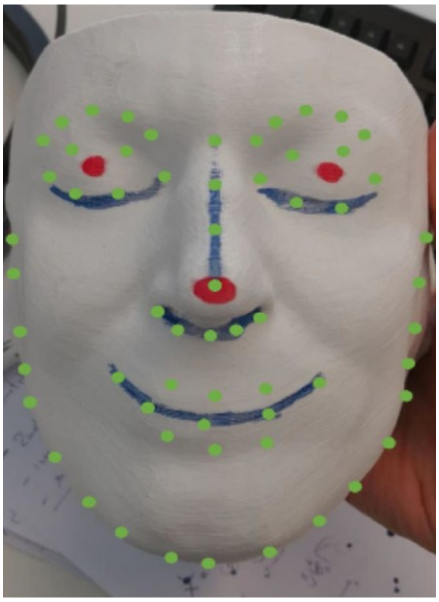


Figure 16 Facial landmarks (in green) detected over a phantom head.

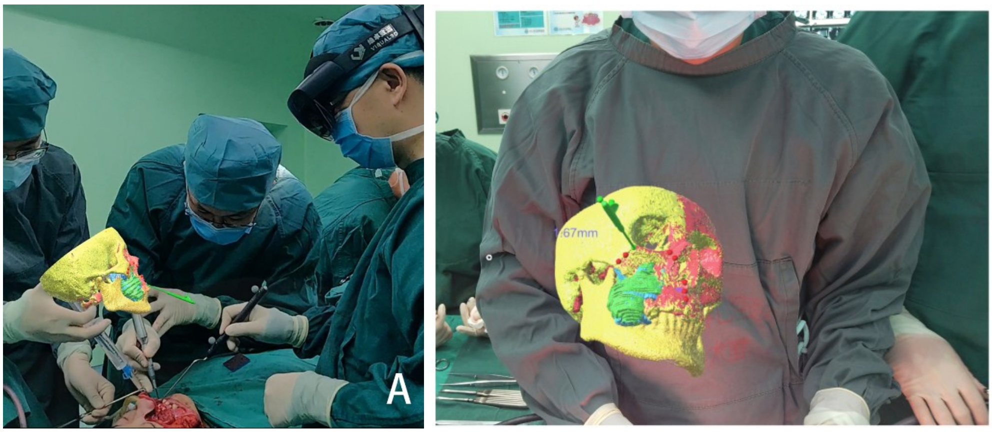


Figure 17 Virtual content displayed above the patient but in the surgeon field of view.

Sugahara et al. [66] has developed an application for calcified odontogenic tumours resection. Starting from CT data, the planning of the virtual operation was performed using Mimics, while the application was created with Unity and Visual Studio for the Microsoft HoloLens. The application provided the surgeon with the 3D image, that was able to adjust the transparency of the colours to perform the surgery with the superimposition of the image on the anatomy. Figure 18. In addition, a 3D model detailing the tumour resection domain was 3D printed. Then, the combination of the 3D printed model and the HoloLens enabled the surgeons to perform surgery without separating the operative and visual fields, while maintaining a 3D understanding of the communication region between the maxillary sinus and nasal cavity. This procedure has shown to be safer and was completed in less time with respect to traditional technique.

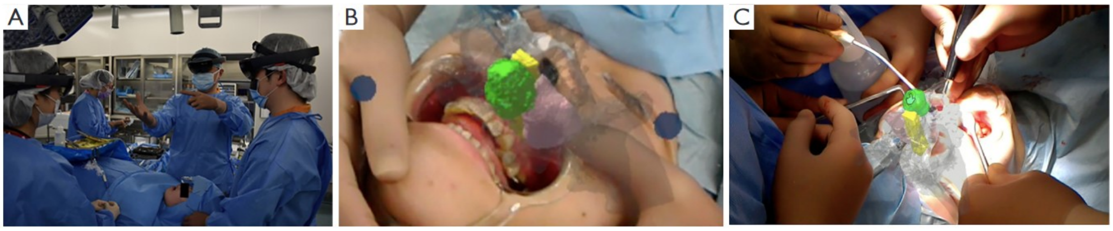


Figure 18 (A) The surgeon and all assistants wear head-mounted displays and share VR images. (B) Intraoperative photographs. (C) VR is superimposed on biological samples.

MR has been used also as a navigation method for dental implantation, as reported by Liu et al. [67] that proposed an application that helped the surgeon to position a dental implant. A method was developed to accomplish real-time display of the implant area model, target positioning, implant angulation, implantation instruments, and implant path by comparing pre-surgical plans with observed paths via MR-smart glasses. Starting from the acquisition of the dentition and jawbones through CBCT, and the 3D scan of the implantation handpiece, dental implant, and drilling instruments were fabricated and imported into Unity to for the development of the application. After the system and the surgical probe calibration, the surgeon was able to see on the MR glasses the computation of the spatial position of the implant relative to the planned entry point, apex point, and angulation, to complete the surgical procedure. Figure 19. Outcomes of this study revealed that the MR-based navigation system were more precise than freehand approach (FH) for entry deviation ((MR: 0.6914 ± 0.2507 mm, FH: 1.571 ± 0.5004 mm, P = 0.000), middle deviation (MR: 0.7156 ± 0.2127 mm, FH: 1.170 ± 0.3448 mm, P = 0.000) , apex deviation (MR: 0.7869 ± 0.2298 mm, FH: 0.9190 ± 0.3319 mm, P = 0.1082) and angular deviation (MR: 1.849 ± 0.6120°, FH: 4.933 ± 1.650°, P = 0.000).

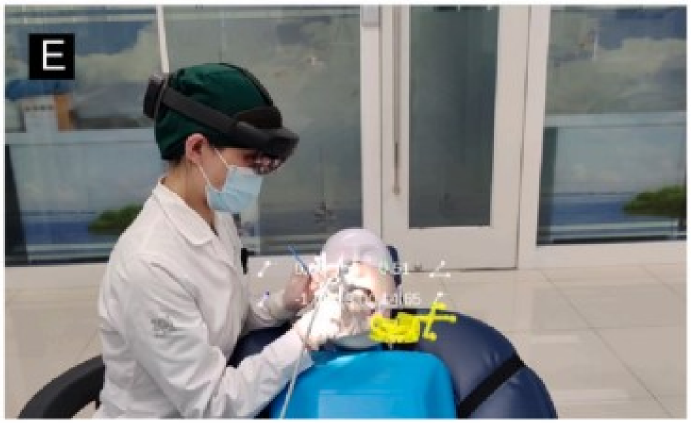


Figure 19 Intraoperative interface for navigation guidance, information on drill path.

Rahimov et al. [68] report the use of MR technology to simulate the orbital floor reconstruction. The CT scan of a healthy subject was imported in Mimics, then an artificial defect in the right orbit was created. The implant were designed within 3-Matic Medical 9.0, then it was loaded on Mimics to position it correctly on the anatomy. The obtained 3D data was used for the plastic model fabrication and for the creation of the virtual scene in Unity, that was loaded to Microsoft HoloLens MR headset to achieve 3D holograms in real space. The designed application gives two options to the user, one for turning on/off the virtual elements, and the second to navigate virtual objects in 3D space and place them in the correct spot. To accomplish the spatial registration of the virtual object to the plastic model, three cylinders were added both to the virtual and plastic model and then superimposed in the 3D space. Figure 20. Objective assessment by the means of comparison of virtual and real measurements data showed inaccuracy in a range of 0.3–0.5 mm.

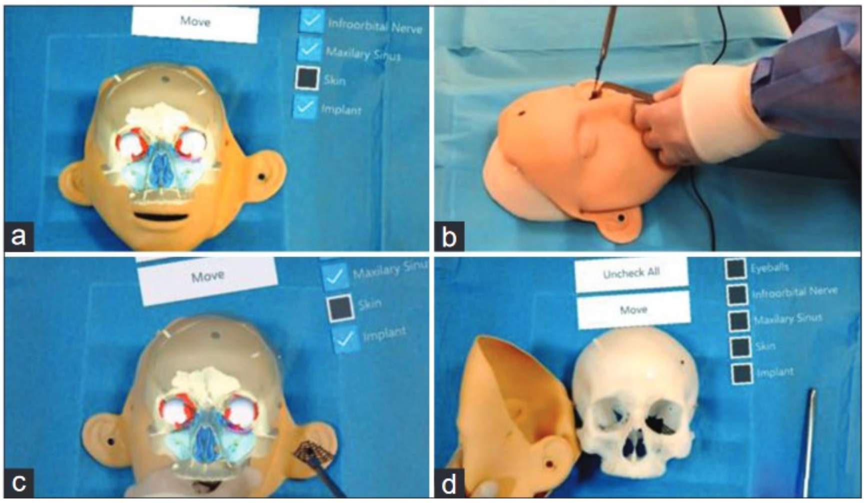


Figure 20 Mixed reality aided surgical simulation: (a) mapping of the virtual hologram to plastic model; (b) insertion of the implant (side view); (c) insertion of the implant (see-through display view); (d) inserted implant.

Still in the context of simulation, Lin et al. [69] used MR approach for the presurgical planning and presurgical discussion for mandible reconstruction, while the actual surgery were performed combining the 3D printed surgical guides with robotic-assisted navigation. All sorts of imaging data including CT, contrast-enhanced CT, and intraoral optical scan were collected. Then, the data was imported into the MIMICS system and 3-Matic system for reconstructing the mandible shape. The virtual content were loaded on HoloLens, in this manner the surgeon can see the arteriovenous and bone tissues in reality by wearing MR glasses, which provide a more convenient and real way for preoperative discussion and simulating the surgery procedure. Figure 21.

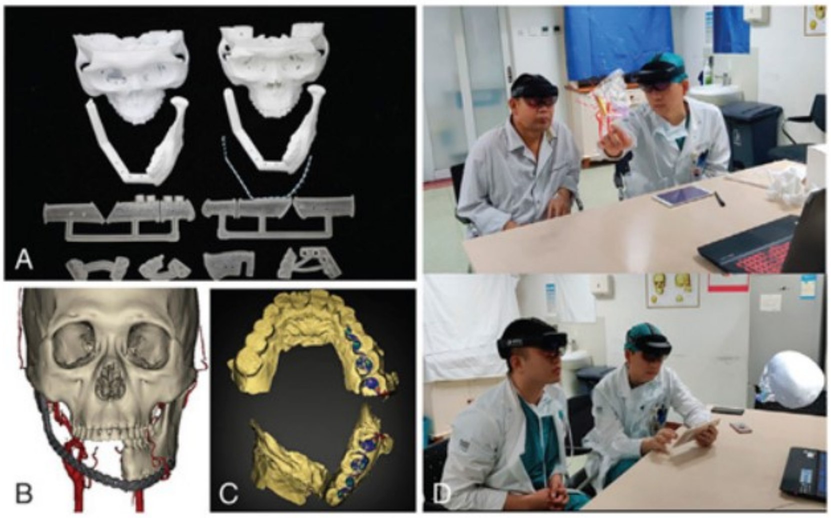


Figure 21 (A) Three-dimensional printing of maxillofacial lesions and surgical guide plate after preoperative CT reconstruction. (B) Images of preoperative enhanced CT reconstruction. (C) Preoperative intraoral occlusal scanning. (D) Preoperative communication and design of surgical plan by mixed reality. CT, computed tomography.

Also, Piramide et al. [70] presented a MR-based solution to support professionals in surgical guide positioning for the anatomical and functional restoration of the mandible structures after an oncological resection or trauma. CT scans of the patient’s mandible and of the lower leg as the donor site are performed to obtain DICOM images, then CAD simulation of the mandibular osteotomies to design and fabricate custom surgical guides. Vision-based tracking system used to align virtual surgical guides, created with Blender software, to the patient’s anatomy. MR display system is necessary to visualize and analyse the results of the guide alignment. In fact, on the MR display not only the virtual surgical guides are presented to the surgeon’s eyes but also feedback window that inform the user with the tracking status and quantitative measures of the overlap in real time. The outcomes show a distance error of 6 mm and a rotation error of 5.25 degrees in the placement over time of surgical guides.

Koyachi et al. [71] used MR for right mandibular segment resection and iliac block graft under the diagnosis of ameloblastoma. CT data of head, neck and ilium were collected to prepare the virtual surgical planning. The osteotomy guide to determine the extent of tumour resection and the reconstruction plate and screw positions necessary for the mandibular reconstruction were designed on ProPlan and loaded on HoloLens. The first part of the surgery, incision in the mandible and in the ilium, insertion of osteotomy guides and temporary fixation were performed in the traditional manner. After, for deeper iliac bone cutting, the hologram was overlaid in MR to check the angle during the excision. Then, the iliac bone was fitted to the mandible with custom-made plate and screw, and the reconstruction at the planned position was confirmed using MR. Figure 22. Surgical accuracy was assessed comparing presurgical and three months post-surgery CT data of the mandible, the average error of the temporomandibular joint was −1.13 mm (range, −4.97 to 4.80 mm), and the average error of the reconstructed iliac bone was −0.76 mm (range, −5.82 to 4.52 mm).

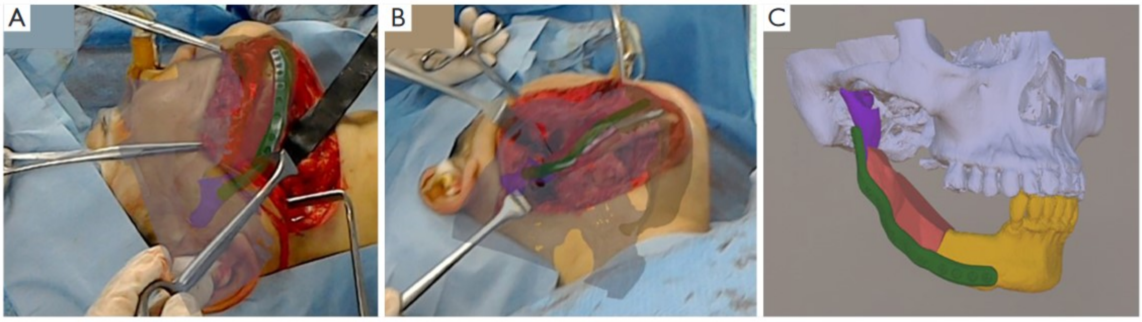


Figure 22 Intraoperative view of mandibular reconstruction using mixed reality technology. (A,B) In order to confirm whether the reconstruction was possible at the planned position, mixed reality was overlayed, and the positions were confirmed three-dimensionally from various directions. (C) Projected hologram.

Again, Koyachi et al. [72] employed MR in conjunction with CAD/CAM technologies to carry out Le Fort I osteotomy. The skull anatomy was acquired by means of CT scan, while a laser scanner was used to obtain the dentition. These data were combined to create a skull-dentition 3D digital model, a splint, an osteotomy device, and a repositioning device were designed. The final model was 3D printed and loaded into Unity to create the MR application for HoloLens. The hologram reconstructs the skull, the pre- and post-repositioned maxilla/mandible, the markers, and the arteries/veins from the contrast CT scans. ArUco markers were attached to the splint and used to align virtual contents with the anatomy of the patient. Figure 23. Postoperative evaluations were conducted by comparing the preoperative virtual operation 3D image (Tv) with the 1-month postoperative CT image (T1) and showed that the median error for each axis when comparing Tv and T1 was < 2 mm.

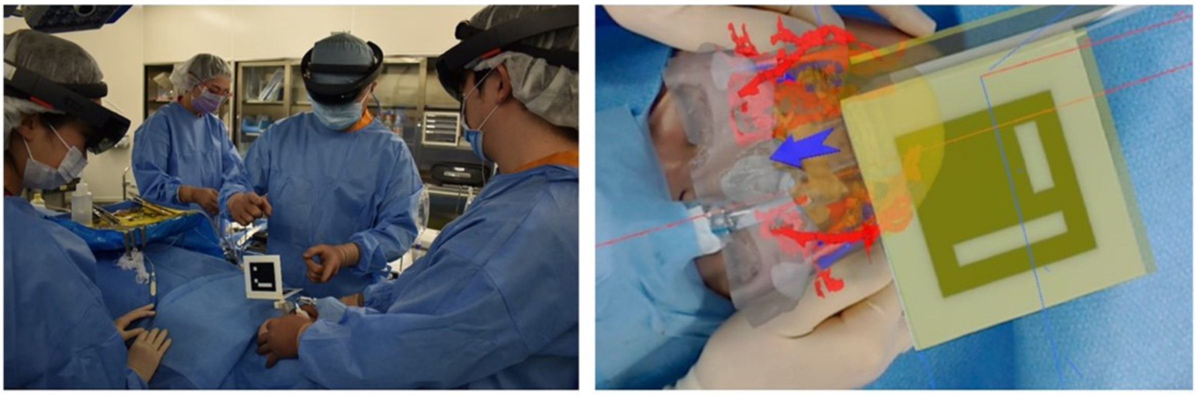


Figure 23 Intraoperative view with operators wearing HoloLens.

2.4 Color Jet Printing (CJP)

Color Jet Printing (CJP) technology was developed at Massachusetts Institute Technology (MIT) in the early 1990s. CJP is an additive manufacturing method in which a core material is bonded together using a binding agent. The core material is in the form of powder, which is spread in a thin layer over a building platform through a roller system. Then the other material, the binder, is selectively deposited onto the powder layer by means of an inkjet printer head. Figure 24. This binder selectively solidifies the powder. Different colored binders exist and are mixed to directly create the colors of the model while simultaneously creating the shape, thus producing three-dimensional full-color models. When one layer is completed, the platform is lowered, and a new layer of powder is spread out and bonded. This process is repeated iteratively until all the necessary layers are completed. Afterward, the resulting product must be cleaned to remove excess powder and finished through the infiltration of an acrylic compound .After the printing process, the model is still soft and fragile, and the addition of acrylic compound, known as “super glue”, aims to strengthen the entire product. The standard material for CJP is plaster powder which is bound by colored glue.

This technology enables full-color printing capabilities with cost efficiency and without the need for supporting structures for the model during the printing process. The limitations of CJP are attributed to the brittleness and low mechanical strength of the printed models, making them unsuitable for functional prototypes as the material tends to break easily.

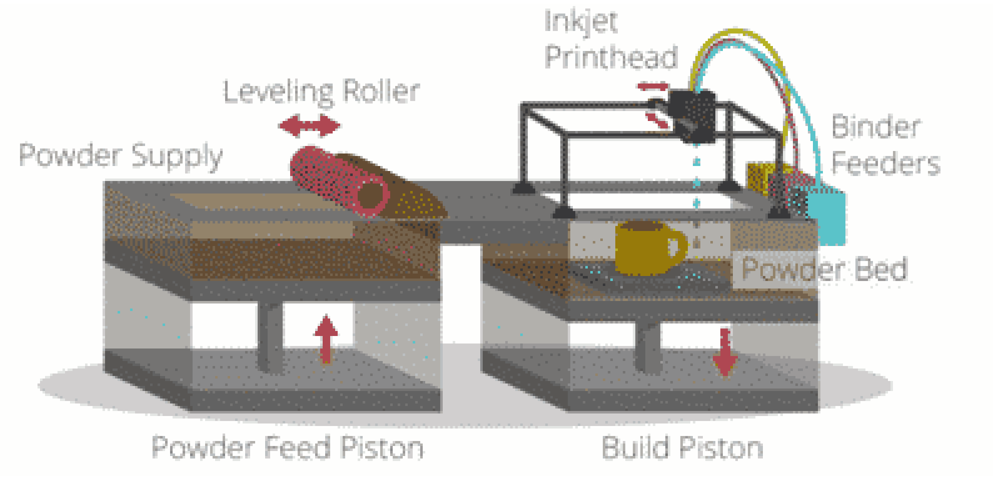


Figure 24 Color Jet Printing (CJP) process phases.

3. Material and Methods

In this section, all the software and hardware components required for the development of this thesis are presented. These include CAD software, specifically Rhinoceros 7, for modelling skull geometry, cutting lines, and markers. Additionally, the tools for application development consist of the Varjo XR3 head-mounted device, the Unity 2020.3.12f1 development environment, along with the programming environment, Visual Studio 2019, and the device management software, Varjo Base. Lastly, the 3D printer used to create the physical model for testing based on the digital 3D model, ZPrinter 450, the scanner for acquiring the tested geometries, Go!Scan3D, and the software for extracting statistical data from the conducted tests, CloudCompare, are introduced.

3.1 CAD Tools

Computer Aided Design (CAD) is a term that describes the usage of computer systems (or workstation) to aid the creation, design, modification, and optimization of a three-dimensional model. [73]

CAD systems are used for generating geometric virtual models. The model represents a nominal geometry, that can be exact or approximated, and it is rigid. Then, many CAD tools exist for each applicative sector, showing additional and specific features for that field.

In the medical field the first task that a CAD tool must accomplish is the codification of the scanned data into the DICOM format, the standard for the communication and management of medical imaging information and related data, to obtain the 3D model.

The 3D model can be optimized and integrated with additional components, in the context of this thesis, for example, CAD software allowed to create customized surgical guide to perfectly fit the patient's anatomy. More in general, in the biomedical field, CAD techniques allow a more rapid and accurate surgical planning.

The final model, at the end can be exported in the adequate format for specific purposed. The principal formats are the Virtual Reality Modelling Language (*.VRML) format, the Standard Triangulation Language (*.STL) format and the Object Files (*.OBJ) format.

Example of CAD software used in the medical field are Rhinoceros, MIMICS and ProPlan. Among the cited software, Rhinoceros is the one used for the development of the 3D model, and it is presented in the following section.

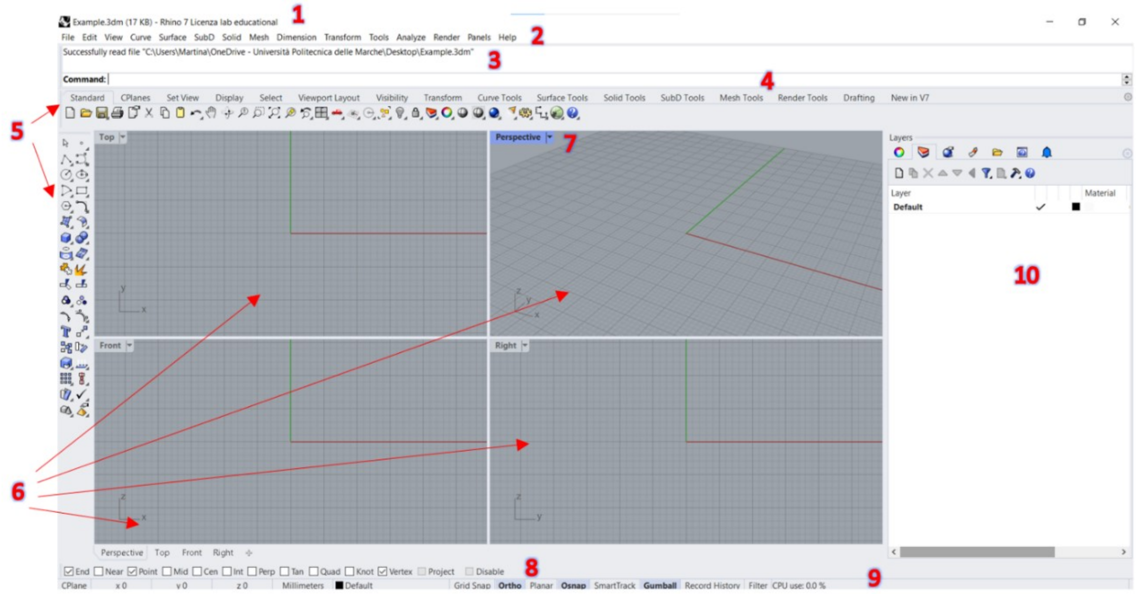
3.1.1 Rhinoceros

Rhinoceros is a commercial applicative software for free-form three-dimensional surfaces modelling created by Robert McNeel & Associates Company (Seattle,Washington, USA). Surface modelling is a modelling technique, used to define the external characteristics of objects with infinitesimal thickness. Rhinoceros allow to create, modify, analyse, document, render, animate, and translate curves, surfaces, solids, SubD, point clouds and polygonal meshes. All geometric entities are represented using Non-Uniform Rational B-Splines (NURBS). NURBS is a mathematical representation for 2D and 3D geometries, able to precisely define any shape, from lines, circles, arcs to complex solids and free-form surfaces. This software shows a variety of features and unlimited tools for modelling with the precision required in designs and prototyping. The models created are compatible with many other design, technical drawing, CAM, analysis, rendering, animation, and illustration programs. Rhinoceros is easy to use due to its simple and intuitive user interface and does not require any specific hardware to have efficient performances. Once the model is created, the file can be exported in more than 30 different formats. The combination of all these features, available at a relatively low price when compared with other concurrent software, has contributed to the popularity of Rhinoceros. [74] [75]

In Figure 25 the software interface is reported. It is possible to find:

1. Title Bar: where current file’s name and size are displayed.
2. Menu Bar: contains the Rhinoceros commands grouped by function.
3. Command History Window: shows the previous command and prompts.
4. Command Prompt: shows prompts for command actions and permits to type command names preventing from research in the Menu.
5. Toolbar: includes all the graphic icons for starting the commands.
6. Viewports: corresponds to Rhinoceros’ working environment, including the views’ title, the object’s visualization, the backgrounds, construction plane’s grid and world axis’s icon. The model can be subdivided into different layers.
7. Viewport Title: by double left mouse button clicking on these sections the viewport is activated and imposed as the unique principal view.

8. Osnap Control: these toggles constrain the marker to an exact location on a specified object, such as the end of a line, or the center of a circle, etc.
9. Status Bar: displays current coordinate system location of the marker, the unit of employed template, the layer pane with the selected layer and its color, the Grid Snap Pane, the Ortho Pane, the Planar Pane, the Osnap Pane, the SmartTrack Pane, the Gumball Pane, the Record History Pane, the Filter Pane, and the Info pane.
10. Layer Panel: displays the layers in the model and offers some tools for managing them.


Example.3dm (17 KB) - Rhino 7 Licenza lab educational
1
File Edit View Curve Surface SubD Solid Mesh Dimension Transform Tools Analyze Render Panels Help
2
3
Command:
4
Standard CPlanes Set View Display Select Viewport Layout Visibility Transform Curve Tools Surface Tools Solid Tools SubD Tools Mesh Tools Render Tools Drafting New in V7
5
Top
Perspective
7
Layers
Layer
Material
Default
10
Front
Right
6
Perspective Top Front Right
8
9

Successfully read file "C:\Users\Martina\OneDrive - Università Politecnica delle Marche\Desktop\Example.3dm"

☑ End ☐ Near ☑ Point ☐ Mid ☐ Cen ☐ Int ☐ Perp ☐ Tan ☐ Quad ☐ Knot ☑ Vertex ☐ Project ☐ Disable

CPlane x 0 y 0 z 0 Millimeters ■ Default Grid Snap Ortho Planar Osnap SmartTrack Gumball Record History Filter CPU use: 0.0 %

Figure 25 Rhinoceros graphical interface.

3.2 Development Tools

This section outlines the essential components required for the development of a MR Application. These components encompass the software tools used to create virtual content, generate scripts, manage Head-Mounted Display (HMD) settings, and the hardware device chosen for visualizing the mixed environment.

3.2.1 Hardware

The MR application developed in this thesis require an HMD to blend the real and virtual content in a unique environment. As seen in Chapter 2, there are several examples of displays, in this context the Varjo-XR3 HMD has been used.

3.2.1.1 Varjo-XR3

Varjo XR-3 (Figure 26) delivers the most immersive mixed reality experience, featuring photorealistic visual fidelity across the widest field of view of any XR headset. And with depth awareness, real and virtual elements blend together naturally.

Varjo's video pass-through technology make virtual objects look real instead of appearing as holographic augmentations. It is possible to explore true-to-life virtual visualizations as natural extensions of the real world thanks to the ultra-low latency (< 20 ms) and high-fidelity 12-megapixel video stream. The world's only human-eye resolution (over 70 PPD) mixed reality headset with the widest field of view (115 degrees) guarantees the most naturally immersive mixed reality experience ever constructed. Varjo XR-3 depth awareness combines the LiDAR sensor and stereo RGB video feed to set a new reference point for hand, object, and people occlusion, pixel-perfectly blending real and virtual elements. And with real-time 3D room reconstructions, realistic immersive environments of the surroundings can be easily built. Ultraleap is used for the hand tracking, it captures all the complexities of your natural hand movements to precisely replicate real-life interactions with either virtual or real contents. And with Varjo's built-in eye tracking at 200 Hz powering foveated rendering, the headset shows what the user is directly looking at in full resolution. Similar to how the human eye works. Eye tracking data can be used also for sophisticated training analysis or research purposes.

It is also possible to easily switch between XR, AR, and VR.

Varjo's XR/VR headsets are compatible with many professional 3D platforms and software, including Unity, Unreal Engine, OpenXR 1.0, Autodesk VRED, PREPAR3D, and VBS Blue IG among others.

Varjo XR-3 is designed with a 3-point precision fit headband to enable maximum comfort for all head shapes and sizes, even during longer immersive sessions. Automatic IPD, active cooling, 90 Hz frame rate, and ultra-wide, crystal clear non-Fresnel lenses ensure to work comfortably without any eye strain or simulator sickness. With hygienic, easy-to-clean materials, the headset is suitable for multi-user setups. Varjo XR-3 can be used also comfortably with glasses.

Key features are listed:

- Photorealistic, true-to-life mixed reality powered by low-latency, 12-megapixel video pass-through.

- The industry's highest resolution (over 70 ppd) and the widest field of view (115°).
- Depth awareness for pixel-perfect real-time occlusion and 3D world reconstruction.
- The widest-ever color gamut matches 99% with the sRGB color space for the most realistic scenes ever produced.
- Integrated Ultraleap hand tracking and integrated 200 Hz eye tracking for natural interactions.
- Inside-out tracking, offering flexibility for deployments without the need for base stations.
- Total comfort with a 3-point precision fit headband, active cooling, and ultra-wide optical design to reduce eye strain and simulator sickness.
- Complete software compatibility. Any software is easily ported into Varjo via Open XR 1.0 or Varjo native SDK. Varjo also supports OpenVR content. In addition, native support for Unity™, Unreal Engine™, and dozens of industrial 3D software, including Autodesk VRED™, Lockheed Martin Prepar3d™, VBS BlueIG™ and FlightSafety Vital™.

All the technical specifications and the minimum system requirements are reported, respectively, in Table 2 and Table 3.

Table 2 Technical specifications of Varjo XR-3

Technical Specification of Varjo-XR3	
Display and Resolution	Full Frame Bionic Display with human-eye resolution. Focus area (27° x 27°) at 70 PPD uOLED, 1920 x 1920 px per eye. Peripheral area at over 30 PPD LCD, 2880 x 2720 px per eye. Colors: 99% sRGB, 93% DCI-P3
Field of view	Horizontal 115°
Refresh rate	90 Hz
Mixed Reality	Ultra-low latency, dual 12-megapixel video pass-through at 90 Hz
XR Depth	LiDAR + RGB fusion, 40 cm–5m operating range
Hand tracking	Ultraleap Gemini (v5)

Eye Tracking	200 Hz with sub-degree accuracy; 1-dot calibration for foveated rendering
Comfort and Wearability	3-point precision fit headband Replaceable, easy-to-clean polyurethane face cushions Automatic interpupillary distance adjustment 59–71mm
Weight	594g + headband 386g
Dimensions	Width 200 mm, height 170 mm, length 300 mm
Connectivity	Two headset adapters in-box Two USB-C cables (5m) in-box PC Connections: 2 x DisplayPort 1.4 and 2 x USB-A 3.0+
Positional Tracking	Steam VR™ 2.0 tracking system Varjo inside-out tracking (Beta) utilizing RGB video pass-through cameras
Audio	3.5mm audio jack with microphone support

Table 3 Minimum and reccommended system requirements to use Varjo XR-3

Minimum system requirements	
Processor	8-core CPU
GPU	NVIDIA GeForce RTX 2080 Ti NVIDIA Quadro RTX 6000
Memory	32 GB
Storage space	2 GB
Connectivity	2 x DisplayPort 1.4 2 x USB-A 3.0/3.1
Operating system	Windows 10 (64-bit)
Recommended system requirements	
Processor	Premium 8-core CPU, For example, Intel Core i9-9900K, Intel Xeon W-2245 8-core, AMD Ryzen 7 3700X
GPU	NVIDIA GeForce RTX 3080 NVIDIA RTX A6000
Memory	32 GB
Storage space	2 GB
Connectivity	2 x DisplayPort 1.4 2 x USB-A 3.0/3.1

Operating System	Windows 10 (64 – bit)

Varjo XR-3 is not a standalone HMD, it needs to be connected to a high-performance computer. To this purpose two 10 meters long cables connect the device to a headset adapter. The adapter is then connected to the power supply and to the workstation by means of two DisplayPort ports on the graphics card and to two USB-A 3.0 / 3.1 ports on the motherboard.

Moreover, to know the exact position of the headset in the space a tracking system is required. Tracking allows the synchronization of the real movements with the virtual movements.

Varjo XR-3 supports three methods for the tracking that are:

- SteamVR Tracking uses base stations and optional controllers to track the device.
- Varjo inside-out tracking uses the built-in cameras of the headset to scan the room and track the position.
- Third-party tracking methods can be integrated with the headset.

In the context of this thesis the SteamVR Tracking method has been used. It is necessary to remove any obstacles in the space to let the user move and to cover, when possible, all the reflective surfaces. Then, SteamVR™ base stations 1.0 or 2.0 are positioned following the recommended distances: a minimum of 1.5 m (5 ft) between each base station and the headset and a maximum of 5 m (16 ft) between base stations. The base stations must be mounted at a minimum height of 2 m and angled down 30–45 degrees. Make sure that the base stations have a direct line of sight with the headset. Up to four base stations can be used at the same time. Four base stations are recommended if the user move around the room, otherwise two base stations are sufficient for sitting or stationary user. Base stations must be connected to the power supply.

Figure 26 Varjo XR-3 Head-Mounted Display.

3.2.2 Software

The development of a MR application requires the integration of multiple software and environments. The primary goal is to accurately display virtual elements within the real world, to achieve this goal, users need to manipulate the positions of holograms within dedicated software, ensuring that holograms are registered precisely in their intended locations. In addition to this engine, a code editor that facilitates the enhancement of scripting and development capabilities can be employed to implement further functionalities. In this context, Unity and Virtual Studio has been employed. Then, since the developed application will run on the Varjo-XR3 headset, Varjo Base software is necessary for the deployment of the application and the settings of the HMD.

3.2.2.1 Unity

Unity is a cross-platform engine distributed by Unity Technologies that allows video games and other interactive content development, such as MR applications. This editor runs on all the main operating systems, Microsoft Windows, macOS and Linux, and the developed product can be launched on Microsoft Windows, macOS, Linux, Xbox 360, PlayStation 3, PlayStation Vita, Wii, iPad, iPhone, Android, Windows Mobile, PlayStation 4, Xbox One, Wii U and Nintendo Switch. Since Unity can be supported on all the platforms listed above, it is necessary to define the target platform at the beginning to be aware of any platform-dependent constraints. When starting a new project, Unity offers multiple templates, such as 2D projects or 3D projects, that will have different assets, scene, and scripts as starting point. Then, it is possible to download and install any additional assets or package that must be integrated to the Unity environment for the application purposes. Assets can be downloaded for free or at a cost in the Asset Store, where the developer can find pre-designed objects, tools, functionalities, animations, or scripts. Further functionality can be implemented in the application by creating scripts in the C# language, using compatible API such as Visual Studio, and assigning them to the objects present in the scene. A basic scene contains a perspective camera and a directional light, depending on whether the project has been started in 2D or 3D mode. Unity allows to test the application during the development by means of Play Mode option. When starting the Play Mode, it is possible to visualize the virtual scene through the camera object and test the application. The Unity graphical interface (Figure 27) is composed in this manner:

1. Menu: contains all the commands grouped according to their function.
2. Toolbar: it consists of several groups of controls related to different parts of the editor.
3. Hierarchy Window: displays every object in the scene from which they can also be managed.
4. Scene View: allows to visualize and interact with the world created in the editor.
5. Inspector Window: allows to view and edit properties and settings for almost everything in the Unity Editor, including GameObjects, Unity components, Assets, Materials, and in-Editor settings and preferences.
6. Project Window: displays all of the files related to the project, allows to find assets and other project files in the application.
7. Console Window: displays errors, warnings, and other messages generated by the editor.

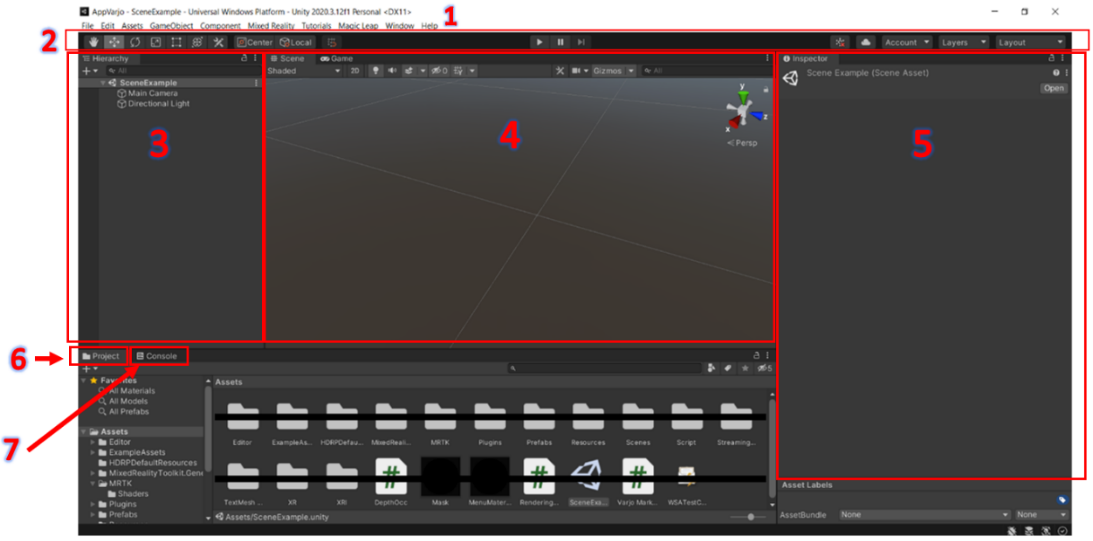
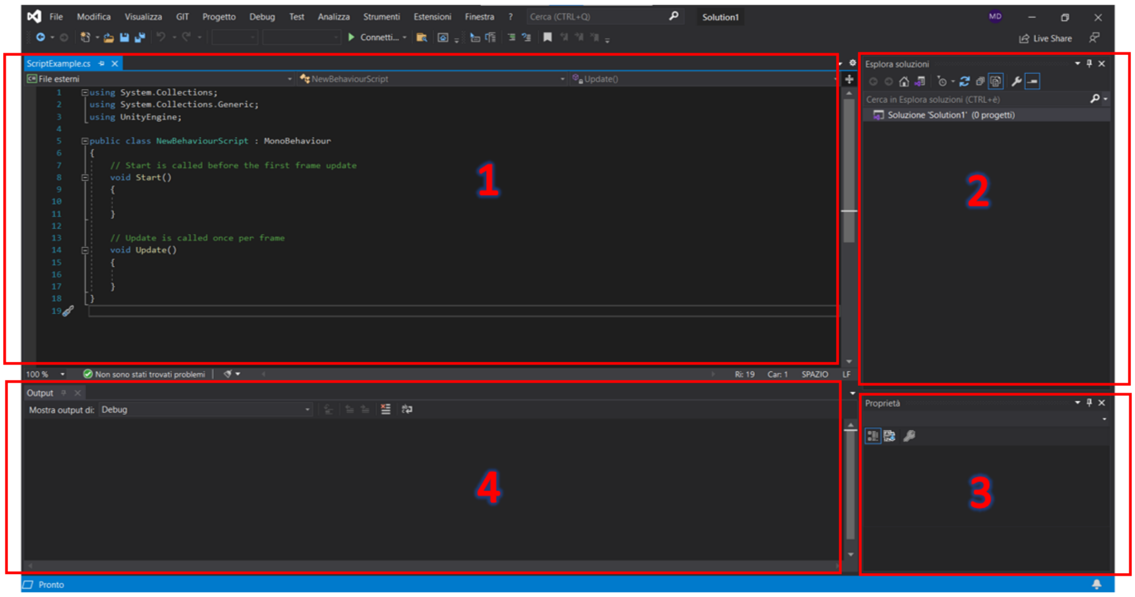

1
2
3
4
5
6
7
AppVarjo - SceneExample - Universal Windows Platform - Unity 2020.3.12f1 Personal <DX11>
File Edit Assets GameObject Component Mixed Reality Tutorials Magic Leap Window Help
Hierarchy
Scene
Game
Inspector
Project
Console
Assets

Figure 27 Unity graphical interface.

3.2.2.2 Visual Studio

Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, to produce native codes or to manage existing codes. A key feature of Visual Studio is the integrated debugging. It supports 36 different programming languages, including C# that makes it compatible with Unity. This software can be used also to deploy the application on the targeted device. When using Unity, the solution built in that engine can be opened in Visual Studio, compiled, and then distributed. The user interface (Figure 28) is composed by:

1. Code Editor: space in which is possible to write the code.
2. Solution Explorer: shows the files on which the user is currently working.
3. Properties Panel: contains information about the project.
4. Output Window: displays debugging and error messages, compiler warnings, and other output.



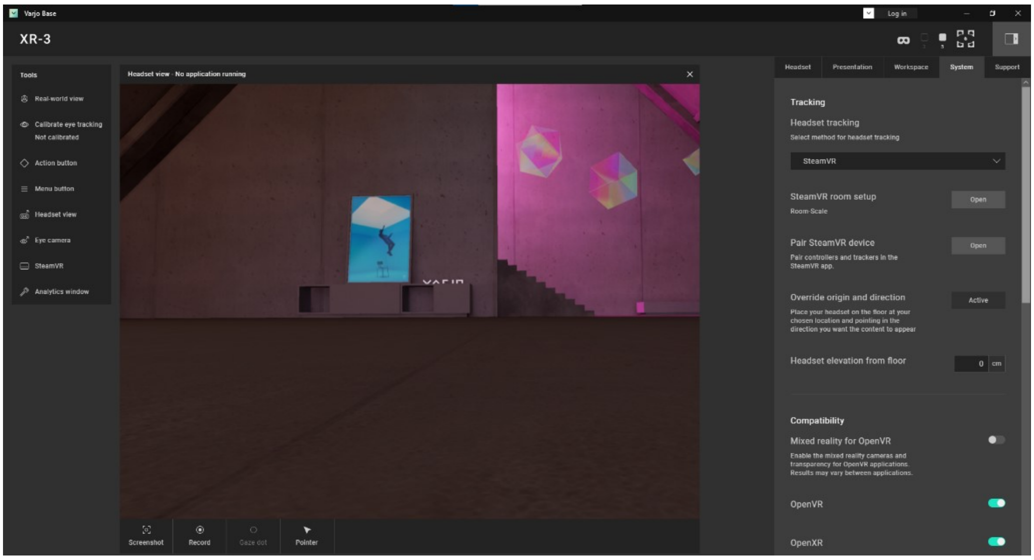
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class NewBehaviourScript : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
        
    }

    // Update is called once per frame
    void Update()
    {
        
    }
}
```

Figure 28 Visual Studio grphical interface.

3.2.2.3 Varjo Base

Varjo Base is the software that allow to control the owned Varjo headset. Figure 29. This software offers a variety of tools, the most notable of which are the real-world view, to turn on the integrated cameras to see the surrounding physical environment; the calibration of eye-tracking; the eye-camera, to show or hide the eye camera view; the headset view, to show or hide the headset view; the analytics window. The analytics window is extremely interesting since it allows to see the headset view without the headset. More in detail, from this window it is possible to enable or disable viewports, to capture screenshots or record video of the headset view, to open further options related to the eye tracking, to display live data about the performance of the application, to toggle the depth buffer, to open options about foveated rendering, to simulate the headset view and movements when the headset is not connected. Figure 30.

From Varjo Base is also possible to start the room setup for the positional tracking, overriding the origin point, direction, and floor level of the room.



Tracking

Headset tracking
Select method for headset tracking
SteamVR

SteamVR room setup — Open
Room-Scale

Pair SteamVR device — Open
Pair controllers and trackers in the SteamVR app.

Override origin and direction — Active
Place your headset on the floor at your chosen location and pointing in the direction you want the content to appear

Headset elevation from floor: 0 cm

Compatibility

Mixed reality for OpenVR ○
Enable the mixed reality cameras and transparency for OpenVR applications. Results may vary between applications.

OpenVR ●

OpenXR ●

Figure 29 Varjo Base graphical interface.

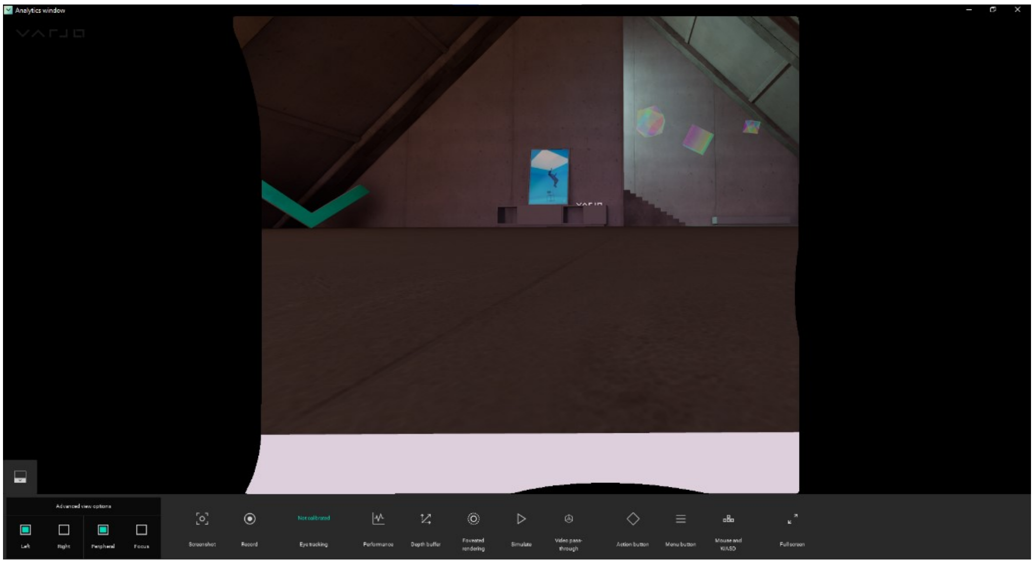


Figure 30 Analytics Window.

3.3 3D Printing

3D Printing is the construction of a three-dimensional object from a CAD model or a digital 3D model. 3D Printing is also called Rapid Prototyping (RP), and it is an additive technology through which the object is generated adding layer by layer of material. Multiple technologies exist that differ from the modality by which the layers are printed. In this thesis, 3D Printing is used to obtain a skull phantom and the markers mounted on it. The technology used is the Color Jet Printing (CJP), the printer is ZPrinter 450.

3.3.1 ZPrinter 450

The ZPrinter 450 is a widespread CJP 3D-printer commercialized by 3D-Systems company. Figure ZPrinter 450 incorporates many features that automate and streamline the 3D printing process. The printer is composed of 2 main chambers, the build chamber, and the fine-powder removal chamber.

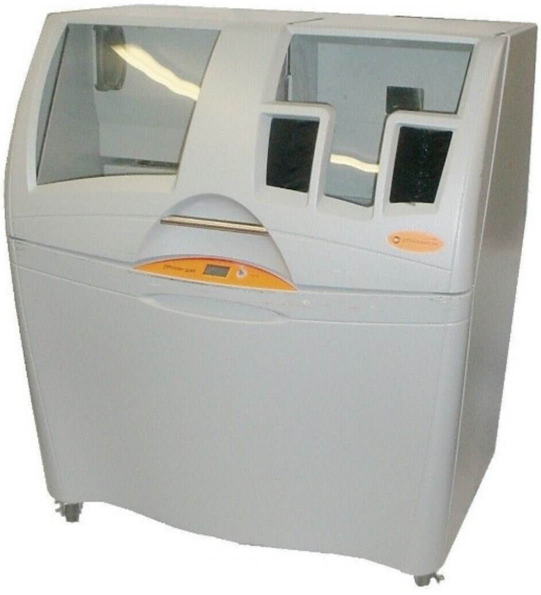


Figure 31 ZPrinter 450.

The build chamber is where the model is printed, while the powder removal chamber is used to refine the final product from any residual powder. The build area dimensions are 203 x 254 x 203 mm. It can print up to 23 mm per hour with a layer resolution that ranges from 90 to 100 microns and a print resolution of 300 x 450 DPI. An LCD panel, controlled by a wheel knob, allows to select among an array of functions ranging from de-powdering, status details, and platform raising controls. The printer runs alongside the ZPrint software that acts as both a slicer and interface to communicate with the printer. It can read STL, VRML, PLY, 3DS, ZPR file types. All the technical specifications are reported in Table 4.

Table 4 Technical specifications of the ZPrinter 450.

ZPrinter 450 Technical Specifications	
Printer Dimensions and Type	122 wide x 79 deep x 140/183 high cm (top cover closed/open) Integrated vacuum system and fine powder

	removal chamber Full color capability with color shell thickness 0.025”
Resolution	300 x 450 dpi
Build Volume and Speed & Layer Thickness	203 x 254 x 203 mm and 2-4 layers per minute 0.089 – 0.102 mm
Operating Environment	Maximum Temperature Range 10-29°C); Recommended Temperature Range (12.7-26.6°C); Maximum Relative Humidity (15-70%); Recommended Relative Humidity Range (20-60%)
Network Connectivity	TCP/IP 100/10 Base T
Noise Levels	57dB (during printing) 66dB (emptying) 86dB (vacuuming – cover open) 80dB (fine powder removal)
Materials – High Performance Composite	zp® 150 Powder with zb® 63 Binder
Power Requirements	Type 1 and 2: 100V, 15A, 50Hz – 115V; 15A, 60Hz – 230V, 7A, 50Hz Type 3 100-120V, 7A, 50-60Hz ; 230V, 4A, 50-60Hz

3.4 Validation Tools

After printing the 3D model and conducting tests to verify its performance, the tested model is scanned. The scanner captures the model that underwent testing, allowing for a comparison with the intended design to identify deviations and errors. In this section the 3D scanning technology, Go!Scan3D device, and the point cloud software for the three-dimensional model creation from the scanned data, CloudCompare, are presented.

3.4.1 Go!Scan3D

The Go!Scan3D is a scanner released by CreaForm3D in 2019. Figure 32. It is a structured light-based handheld scanner that allows users to scan objects easily, quickly, and accurately while also capturing texture information. The scanner is equipped with three cameras, one white light projector with 99 light stripes, and one-color camera. To enhance the scanning process, the scanner comes with two marker packages that are placed on a reference plane where the object is positioned, facilitating calibration. Additionally, there are two lights on the sides of the scanner to help users ensure the correct placement relative to the reference plane. The scanner has some multifunctional buttons for starting and ending the scanning process, as well as for zooming in and out. It also includes a flexible wrist strap for handheld use. The scanner works with a dedicated software named ‘’VXelements’’, which allows user to manage and process the scans and export them in various formats such as .dae, .fbx, .ma, .obj, .ply, .stl, .txt, .wrl, .x3d, .x3dz, .zpr, .3mf. All the technical specification of the Go!Scan3D are reported in Table 5.

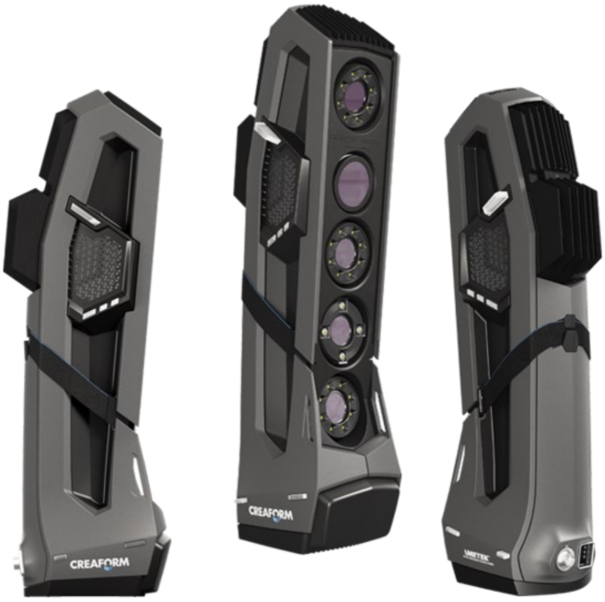


Figure 32 Go!Scan3D Scanner.

Table 5 Technical specifications of the Go!Scan3D Scanner.

Go!Scan3D Technical Specifications	
Accuracy	Until 0.05 mm
Volumetric Precision	0.05 mm + 0.15 mm/m

Measurement Resolution	0.100 mm
Meshes Resolution	0.200 mm
Measurement Frequency	1500000 measurements/s
Light Source	White Light
Positioning Method	Geometry and/or color and/or target
Scan Area	390 x 390 mm (250 x 250 mm : 700 x 700 mm)
Working Distance	400 mm
Depth of Field	450 mm
Texture Resolution	50 : 200 DPI
Texture Color Depth	24 bits

3.4.2 CloudCompare

CloudCompare is a versatile software designed for processing three-dimensional point clouds and triangular meshes, and it is compatible with Windows, Linux, and Mac OS X platforms. This software offers a wide range of tools for editing and rendering, it includes several advanced processing algorithms for registration, resampling, color/normal/scalar fields handling, statistical computations, sensors management, interactive or automatic segmentation, etc. CloudCompare allows users to translate and/or rotate single or multiple entities, enabling the selection of group of points and to open multiple models, aligning and comparing them. The comparison is conducted between the reference geometries (nominal models) and the tested objects, whose geometries are obtained through scanning, (scanned models). Once the nominal and scanned models are aligned, it becomes possible to extract deviations and errors between the nominal or reference model and the actual performance observed during the test.

4. MR Application

One of the objectives of this thesis is the development of a mixed reality application designed to assist maxillofacial surgeons. This application is intended for use with the Varjo XR-3 device and serves two main purposes: the visualization of surgical guides for maxillary and mandibular osteotomies, and the precise repositioning of the mandible to a predetermined position within the preoperative plan. In Figure 33 the entire workflow of the current project is reported. In this section the steps followed for the development of the application are explained.

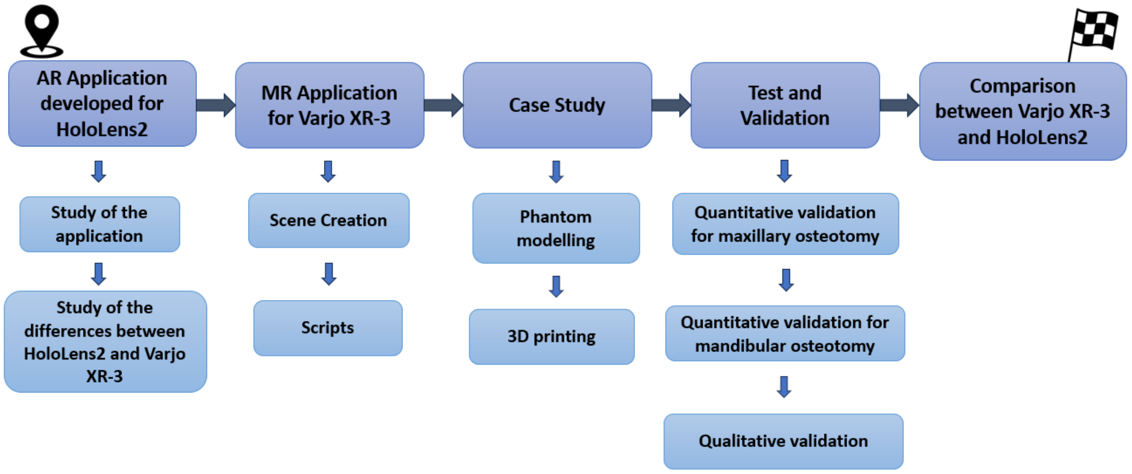
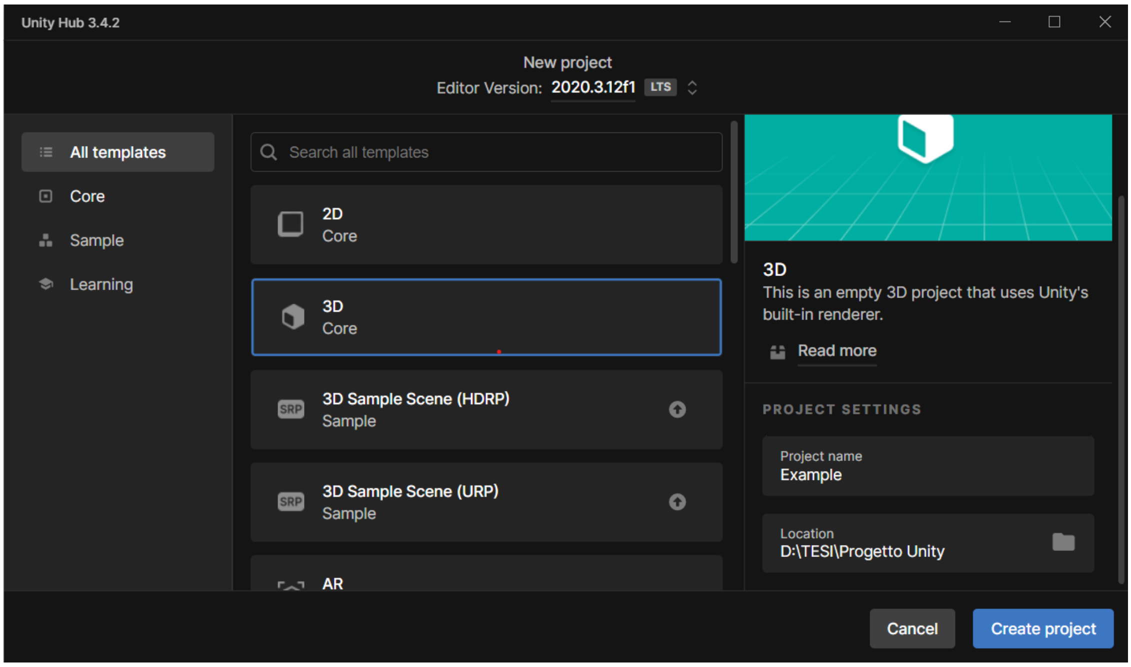

AR Application developed for HoloLens2
Study of the application
Study of the differences between HoloLens2 and Varjo XR-3
MR Application for Varjo XR-3
Scene Creation
Scripts
Case Study
Phantom modelling
3D printing
Test and Validation
Quantitative validation for maxillary osteotomy
Quantitative validation for mandibular osteotomy
Qualitative validation
Comparison between Varjo XR-3 and HoloLens2

Figure 33 Workflow of the project

The software used for the development of the MR application are Unity 2020.3.12f1 and Visual Studio 2019. To develop the application, the first step is the creation of a new 3D project in Unity. To do that the Unity Hub is employed, and here a 3D Project is selected, named, and located in the specific path. Figure 34. Once the project is created, all the project settings, assets, packages, prefabs, and scripts are automatically saved and located in the specified path, in a folder named as the project name. Starting from this new and empty project, the following steps concern the import of the necessary packages, the configuration of the project settings, the creation of the scene and the scripts. At the end of those steps, the application can be launched on the Varjo XR-3.



Editor Version: 2020.3.12f1 LTS

3D

This is an empty 3D project that uses Unity's built-in renderer.

Project name: Example

Location: D:\TESI\Progetto Unity

Figure 34 Unity Hub interface.

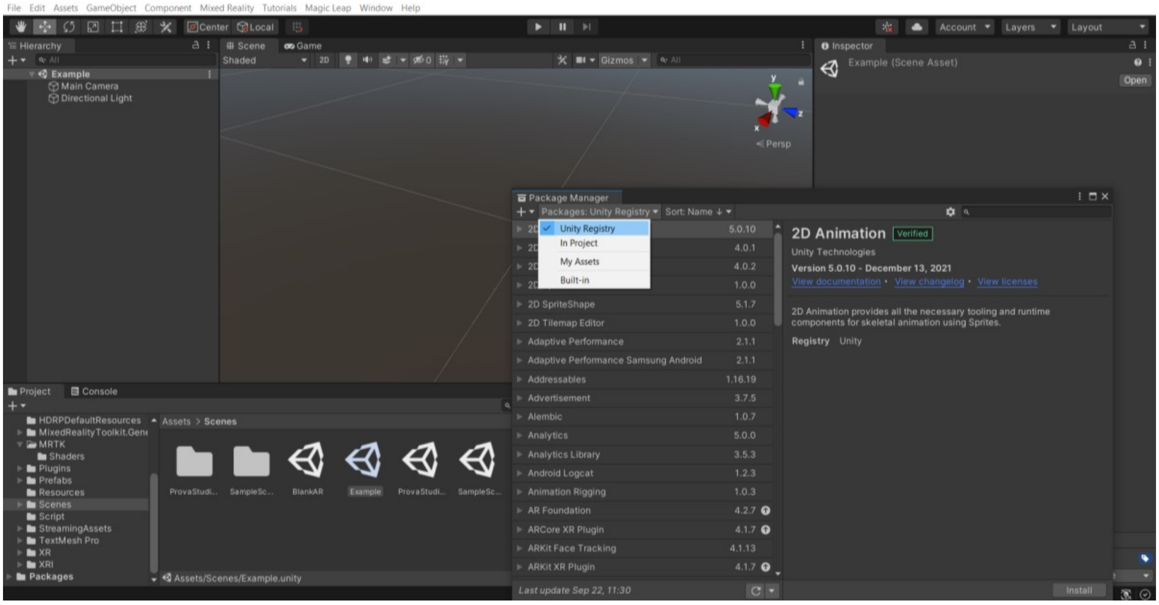
4.1 Packages Import

First step to obtain the necessary features and assets for the application set up is the packages import. Packages must be appropriately downloaded in the dedicated websites and then appropriately imported into Unity. The packages imported in this thesis are Universal Render Pipeline, Varjo XR and Ultraleap.

4.1.1 Universal Render Pipeline

The Universal Render Pipeline (URP) is a prebuilt Scriptable Render Pipeline, made by Unity. URP provides artist-friendly workflows that lets the user create quickly and easily optimized graphics across a range of platforms, from mobile to high-end consoles and PCs. It is a rendering method that supports many platforms and devices, and can be applied to various projects such as 2D, 3D, VR and AR.

To install URP: Window → Package Setting → Packages: Unity Registry → Universal RP → Install. Figure 35 and Figure 36.



2D…	5.0.10
2D…	4.0.1
2D…	4.0.2
2D…	1.0.0
2D SpriteShape	5.1.7
2D Tilemap Editor	1.0.0
Adaptive Performance	2.1.1
Adaptive Performance Samsung Android	2.1.1
Addressables	1.16.19
Advertisement	3.7.5
Alembic	1.0.7
Analytics	5.0.0
Analytics Library	3.5.3
Android Logcat	1.2.3
Animation Rigging	1.0.3
AR Foundation	4.2.7
ARCore XR Plugin	4.1.7
ARKit Face Tracking	4.1.13
ARKit XR Plugin	4.1.7

2D Animation Verified

Unity Technologies

Version 5.0.10 - December 13, 2021

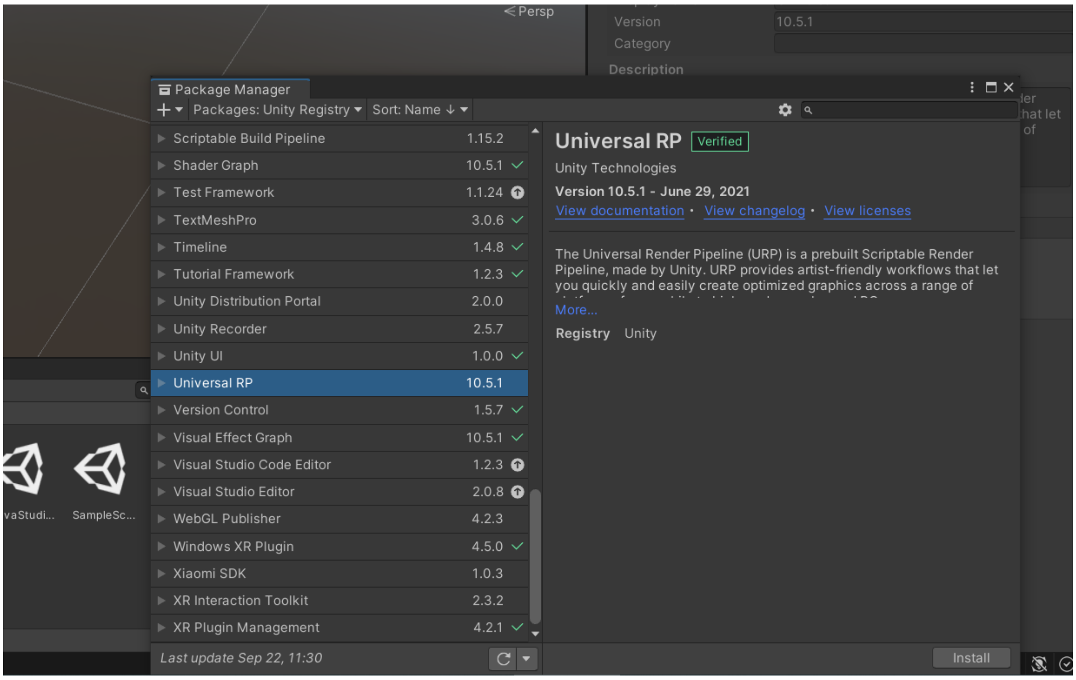
View documentation • View changelog • View licenses

2D Animation provides all the necessary tooling and runtime components for skeletal animation using Sprites.

Registry Unity

Last update Sep 22, 11:30

Figure 35 Universal Render Pipeline install procedure.



Scriptable Build Pipeline	1.15.2
Shader Graph	10.5.1 ✓
Test Framework	1.1.24
TextMeshPro	3.0.6 ✓
Timeline	1.4.8 ✓
Tutorial Framework	1.2.3 ✓
Unity Distribution Portal	2.0.0
Unity Recorder	2.5.7
Unity UI	1.0.0 ✓
Universal RP	10.5.1
Version Control	1.5.7 ✓
Visual Effect Graph	10.5.1 ✓
Visual Studio Code Editor	1.2.3
Visual Studio Editor	2.0.8
WebGL Publisher	4.2.3
Windows XR Plugin	4.5.0 ✓
Xiaomi SDK	1.0.3
XR Interaction Toolkit	2.3.2
XR Plugin Management	4.2.1 ✓

Universal RP Verified

Unity Technologies

Version 10.5.1 - June 29, 2021

View documentation • View changelog • View licenses

The Universal Render Pipeline (URP) is a prebuilt Scriptable Render Pipeline, made by Unity. URP provides artist-friendly workflows that let you quickly and easily create optimized graphics across a range of

More...

Registry Unity

Last update Sep 22, 11:30

Figure 36 Universal Render Pipeline install procedure.

4.1.2 Varjo XR Plugin

The specific package for the HMD chosen, Varjo XR3, is Varjo XR Plugin. Once chosen the render pipeline and verified the compatibility, Varjo XR Plugin can be installed. It provides display, input, and mixed reality feature support for Varjo HMDs.

To import this package, the steps followed are Window → Package Manager → '+'→ Add Package from disk. Figure 37 and Figure 38.

Once installed Varjo XR Plugin, to configure the Varjo XR-3 as one of the XR devices initialized at the startup: Edit → Project Settings → XR Plug-in Management → Varjo. Figure 39.

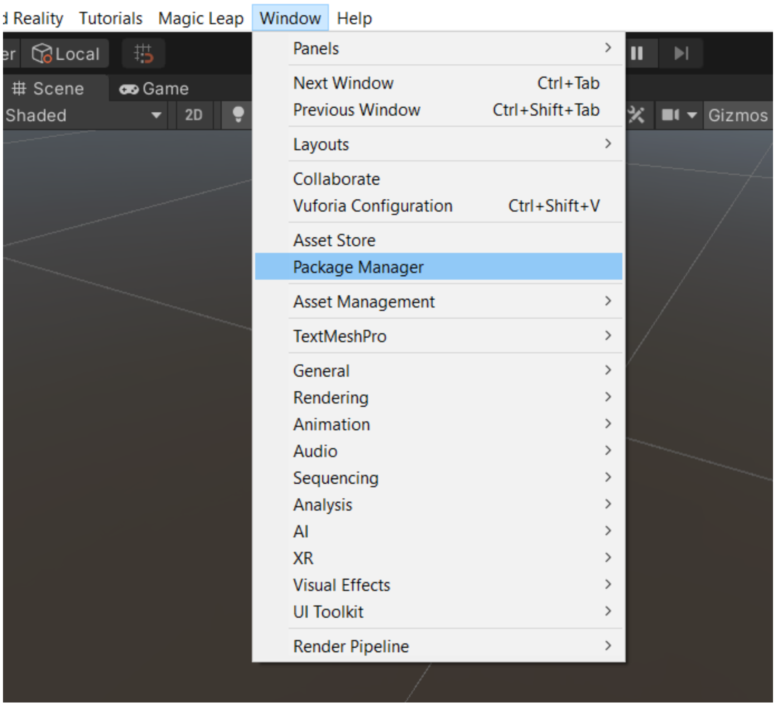
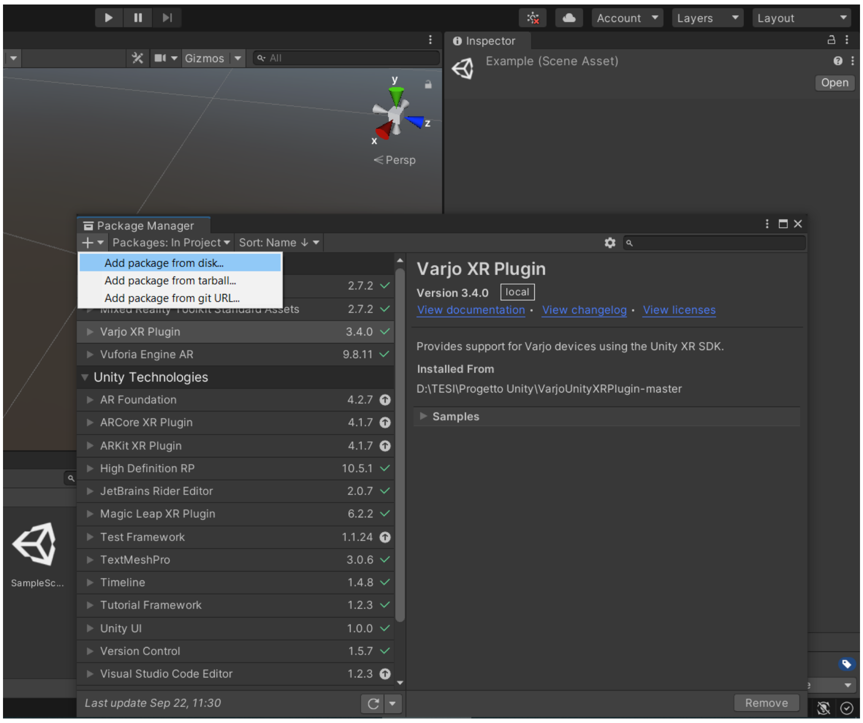


Figure 37 Varjo XR Plugin import procedure.



	2.7.2
Mixed Reality Toolkit Standard Assets	2.7.2
Varjo XR Plugin	3.4.0
Vuforia Engine AR	9.8.11
Unity Technologies	
AR Foundation	4.2.7
ARCore XR Plugin	4.1.7
ARKit XR Plugin	4.1.7
High Definition RP	10.5.1
JetBrains Rider Editor	2.0.7
Magic Leap XR Plugin	6.2.2
Test Framework	1.1.24
TextMeshPro	3.0.6
Timeline	1.4.8
Tutorial Framework	1.2.3
Unity UI	1.0.0
Version Control	1.5.7
Visual Studio Code Editor	1.2.3

Varjo XR Plugin

Version 3.4.0 local

View documentation • View changelog • View licenses

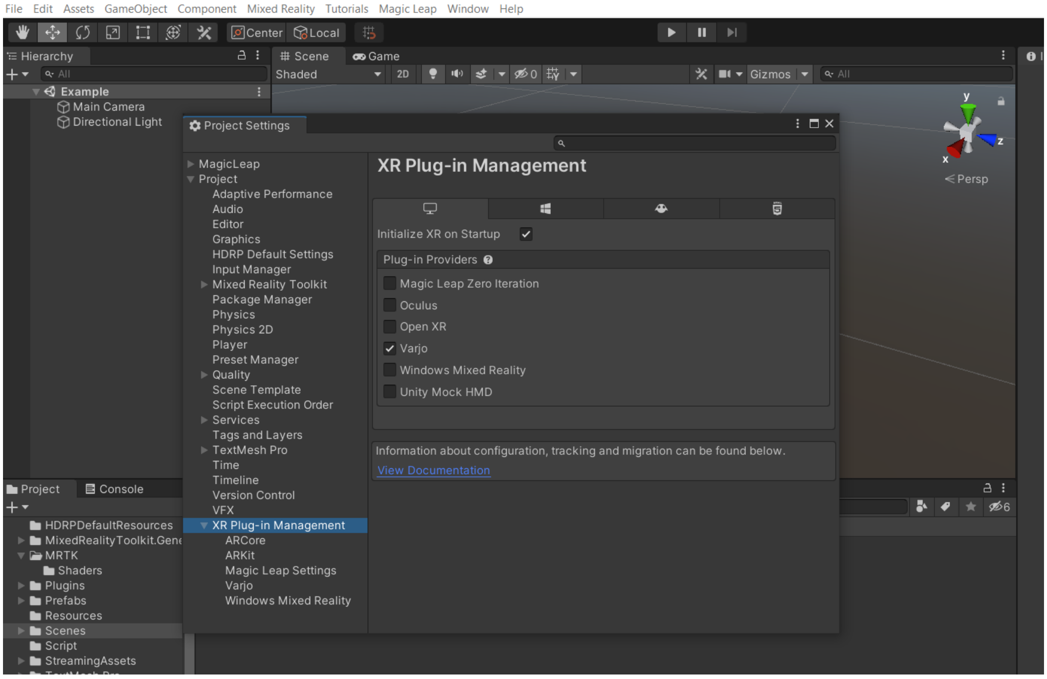
Provides support for Varjo devices using the Unity XR SDK.

Installed From

D:\TESI\Progetto Unity\VarjoUnityXRPlugin-master

Samples

Figure 38 Varjo XR Plugin import procedure.



XR Plug-in Management

Initialize XR on Startup ☑

Plug-in Providers

☐ Magic Leap Zero Iteration
☐ Oculus
☐ Open XR
☑ Varjo
☐ Windows Mixed Reality
☐ Unity Mock HMD

Information about configuration, tracking and migration can be found below.
View Documentation

Figure 39 Configuration of the Varjo Headset.

4.1.3 Ultraleap Tracking Plugin

The hand tracking with Varjo XR-3 headset works with Ultraleap Tracking Plugin. The unity package is downloaded from Ultraleap websites and imported in Unity. To import this package: right click on the Project Window → Import Package → Custom Package, and then search for the Ultraleap package. Figure 40 and Figure 41. This package includes a minimal interface between Unity and Ultraleap Hand Tracking Software.

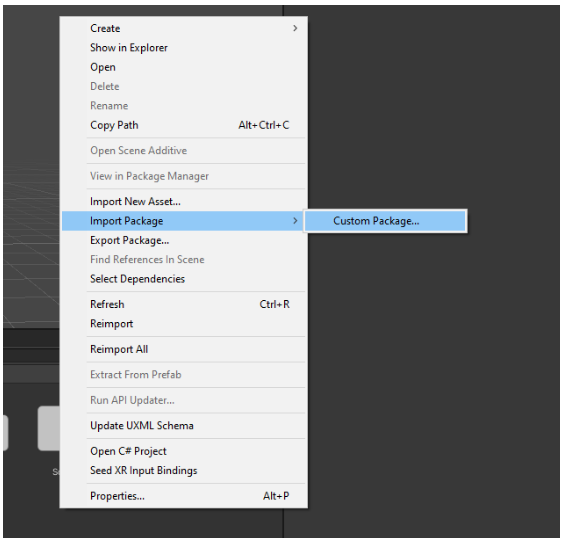
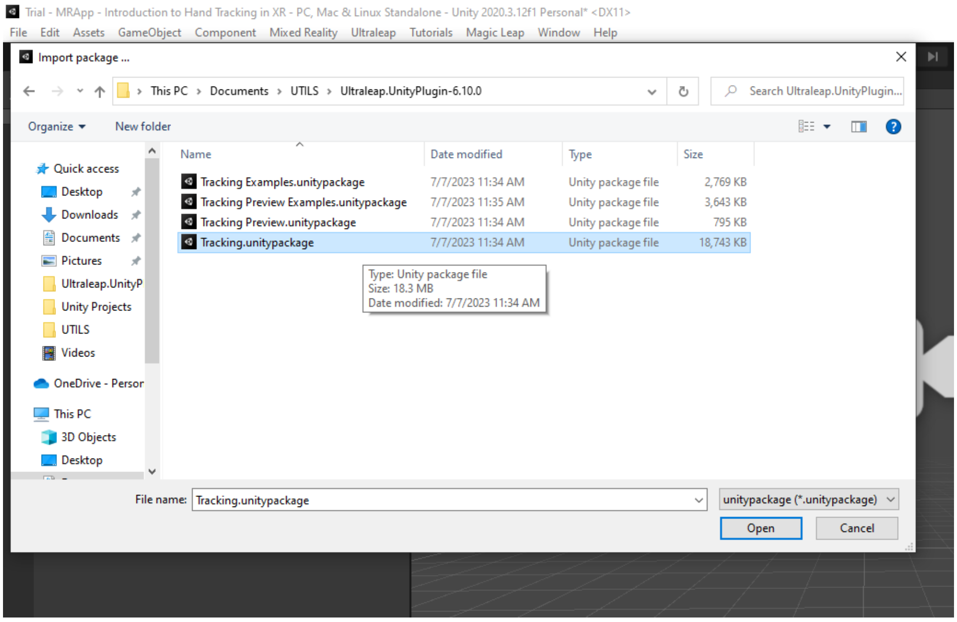


Figure 40 Ultraleap import procedure.


Trial - MRApp - Introduction to Hand Tracking in XR - PC, Mac & Linux Standalone - Unity 2020.3.12f1 Personal* <DX11>
File Edit Assets GameObject Component Mixed Reality Ultraleap Tutorials Magic Leap Window Help
Import package ...
This PC › Documents › UTILS › Ultraleap.UnityPlugin-6.10.0
Search Ultraleap.UnityPlugin...
Organize
New folder
Quick access
Desktop
Downloads
Documents
Pictures
Ultraleap.UnityP
Unity Projects
UTILS
Videos
OneDrive - Persor
This PC
3D Objects
Desktop
Open
Cancel

Name	Date modified	Type	Size
Tracking Examples.unitypackage	7/7/2023 11:34 AM	Unity package file	2,769 KB
Tracking Preview Examples.unitypackage	7/7/2023 11:35 AM	Unity package file	3,643 KB
Tracking Preview.unitypackage	7/7/2023 11:34 AM	Unity package file	795 KB
Tracking.unitypackage	7/7/2023 11:34 AM	Unity package file	18,743 KB

Type: Unity package file
Size: 18.3 MB
Date modified: 7/7/2023 11:34 AM

File name: Tracking.unitypackage

unitypackage (*.unitypackage)

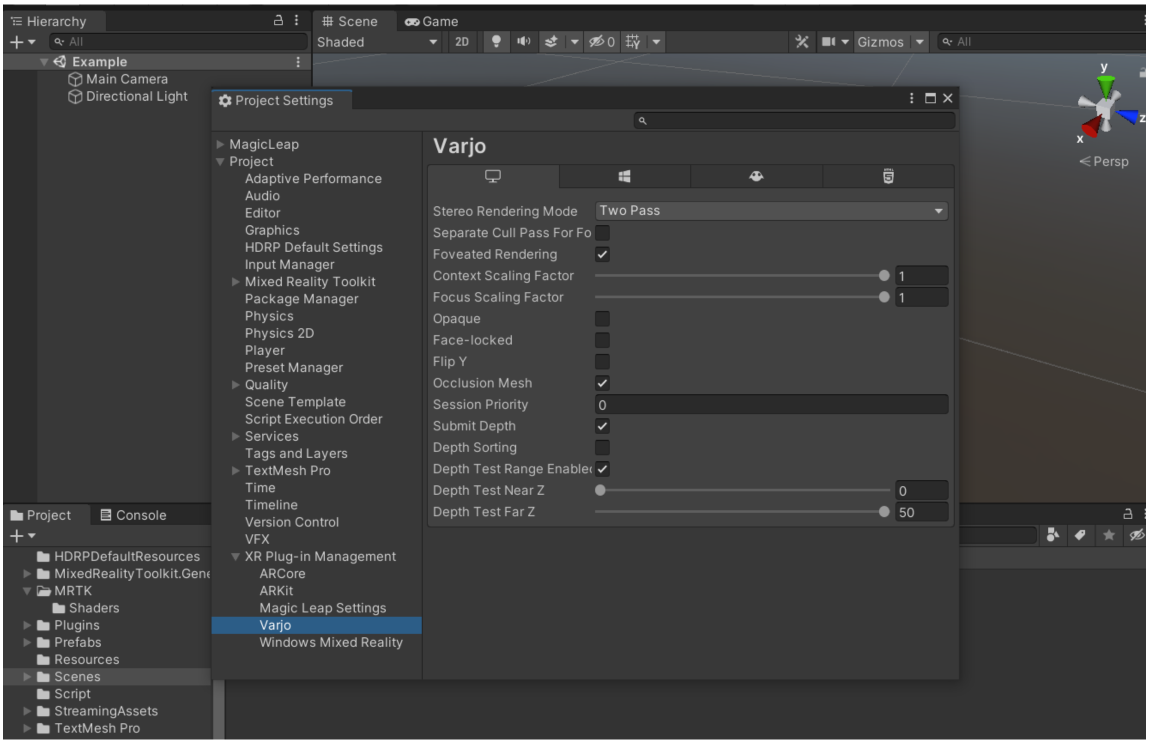
Figure 41 Ultraleap package selection.

4.2 Mixed Reality Settings

To develop a MR application, some settings need to be configured to enable the rendering of the images from the video pass-through cameras.

First thing to do is the conversion of the Main Camera to an XR Rig, to enable HMD tracking. It is converted by clicking on Game Object → XR → Convert Main Camera To XR Rig.

Then, in the Varjo XR plugin settings, it is necessary to uncheck the Opaque option. Under the XR Plug-in Management tab, select Varjo and disable the Opaque option as shown in Figure 42.



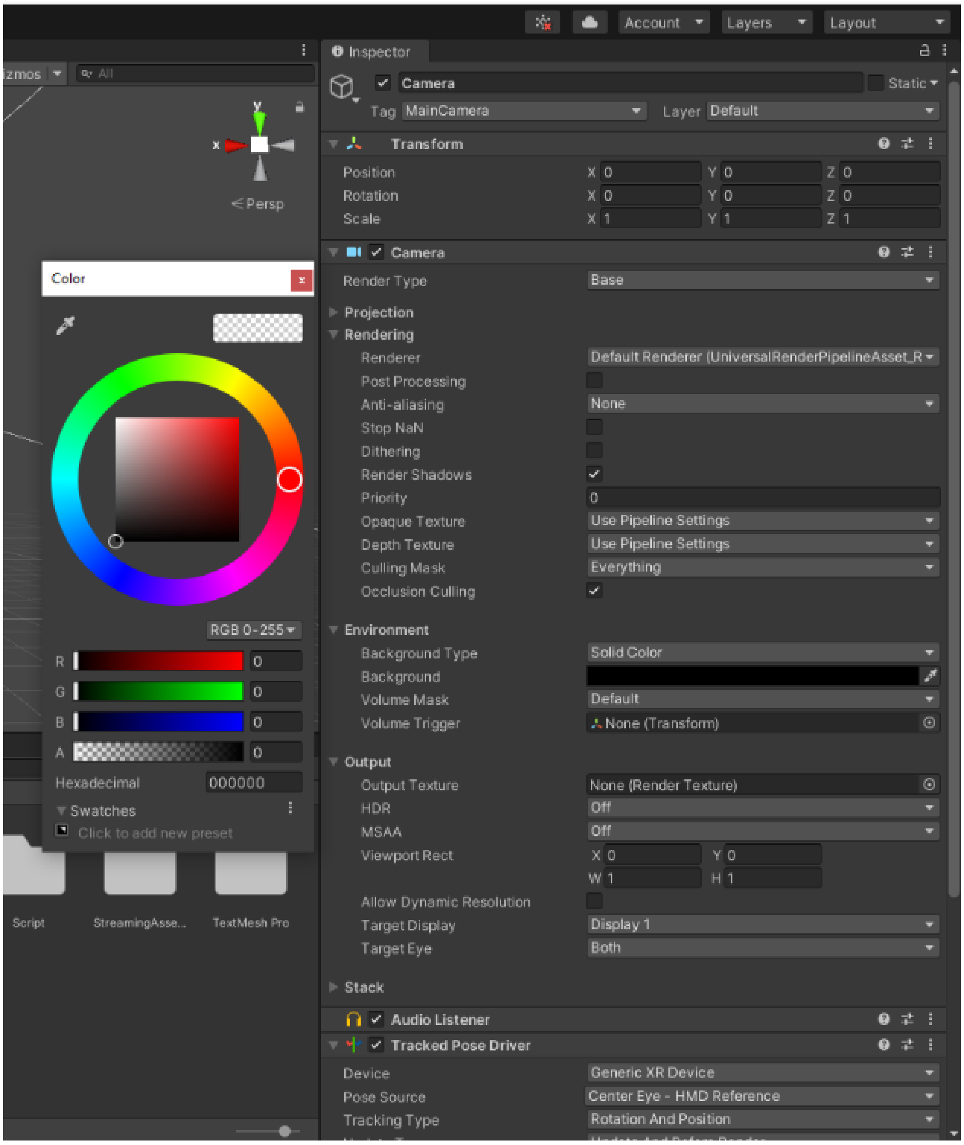
Varjo

Stereo Rendering Mode	Two Pass
Separate Cull Pass For Fo	□
Foveated Rendering	☑
Context Scaling Factor	1
Focus Scaling Factor	1
Opaque	□
Face-locked	□
Flip Y	□
Occlusion Mesh	☑
Session Priority	0
Submit Depth	☑
Depth Sorting	□
Depth Test Range Enablec	☑
Depth Test Near Z	0
Depth Test Far Z	50

Figure 42 Varjo XR Plugin settings window.

After that, the script “Start_MR” is assigned to the XR Rig to enable or disable the rendering of the image when the application is running. When enabled, the system will start to render the video pass-through images when the camera is clear to 0.

When using URP, camera settings to write into the alpha channel of the color buffer need to be modified. For video pass-through to work, the camera’s Post Processing and HDR settings must be disabled. Now, to make the camera clear to 0, set the camera’s Background Type to Solid Color and set RGBA(0,0,0,0) as the background color. Figure 43.



R 0
G 0
B 0
A 0
Hexadecimal 000000

☑ Camera □ Static
Tag MainCamera Layer Default

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

☑ Camera
Render Type: Base
Projection
Rendering
Renderer: Default Renderer (UniversalRenderPipelineAsset_R
Post Processing: □
Anti-aliasing: None
Stop NaN: □
Dithering: □
Render Shadows: ☑
Priority: 0
Opaque Texture: Use Pipeline Settings
Depth Texture: Use Pipeline Settings
Culling Mask: Everything
Occlusion Culling: ☑
Environment
Background Type: Solid Color
Background
Volume Mask: Default
Volume Trigger: None (Transform)
Output
Output Texture: None (Render Texture)
HDR: Off
MSAA: Off
Viewport Rect: X 0 Y 0 W 1 H 1
Allow Dynamic Resolution: □
Target Display: Display 1
Target Eye: Both
Stack

☑ Audio Listener

☑ Tracked Pose Driver
Device: Generic XR Device
Pose Source: Center Eye - HMD Reference
Tracking Type: Rotation And Position

Figure 43 MainCamera Inspector.

4.3 Hand Tracking

To enable the hand tracking in this project, as said before it is necessary to import the Ultraleap Plugin. In addition, some Ultraleap prefabs are imported. These prefabs are XR Leap Provider Manager, Hand Prefab, Interaction Manager, Interaction Hands and Attachment Hands, that allow the user to interact with 3D objects and to spawn objects via a hand menu UI.

XR Leap Provider Manager prefab simplifies the setup process for adding hand tracking to an XR based scene, providing a single access point to various possible source of tracking, selecting the most suitable.

Hand Prefab is used to visualize hand tracking data using custom mesh-based hands and provide a variety of hand styles. Figure 44. They are pre-built, optimized and rigged meshes that use a transform hierarchy to move the hands.

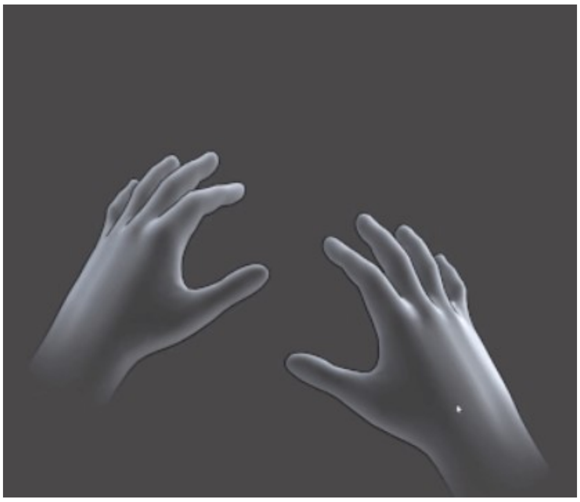


Figure 44 Ghosted prefab hands.

The Interaction Manager is responsible for creating physics representations of the user's hands through the use of the Interaction Hands. These physics enabled hands will match the size of the user's hands as closely as possible given the tracking data and will enable the user with interaction capabilities. Interaction Hands enables interaction capabilities in a scene. This prefab, with the appropriate XR Controller prefabs, allows to use hand tracking in combination with controllers for input. Lastly the Attachment Hands prefab allows to easily attach virtual objects to 21 possible data points tracked in real-time on the user's hand.

4.4 Varjo Markers

Varjo Markers are specialized markers employed by Varjo technologies to facilitate the recognition of physical objects, enabling them to serve as reference anchors for projecting holograms. This capability enables precise alignment and positioning of virtual content within the real-world environment. Each marker has a unique ID, and the same marker shouldn't be used more than once in any given environment. The Varjo Markers used in this thesis are the smaller-sized markers measuring 25x25x25 mm, which are capable of tracking objects up to a distance of 1 meter. These markers an outer border such that the final dimensions are 32.5x32.5x32.5 mm. They have been selected from the marker database on Varjo Developer website.

A total of 14 markers were selected and assigned to the faces of the markers placed on the forehead, on the maxilla, and the marker on the mandible (the posterior one is never visible).

The forehead marker is built as follows: the frontal face has the ID 101, the left face has ID 100, the right face has ID 102, the bottom face has ID 103, and the top face has ID 104. Figure 45.

The maxilla marker is built as follows: the frontal face has the ID 106, the left face has ID 105, the right face has ID 107, the bottom face has ID 108, and the top face has ID 109. Figure 46.

The mandible marker is designed in a different manner. It is an allow cube without the bottom face. It is built as follows: the frontal face has the ID 111, the left face has ID 110, the right face has ID 112, and the top face has ID 114. Figure 47.

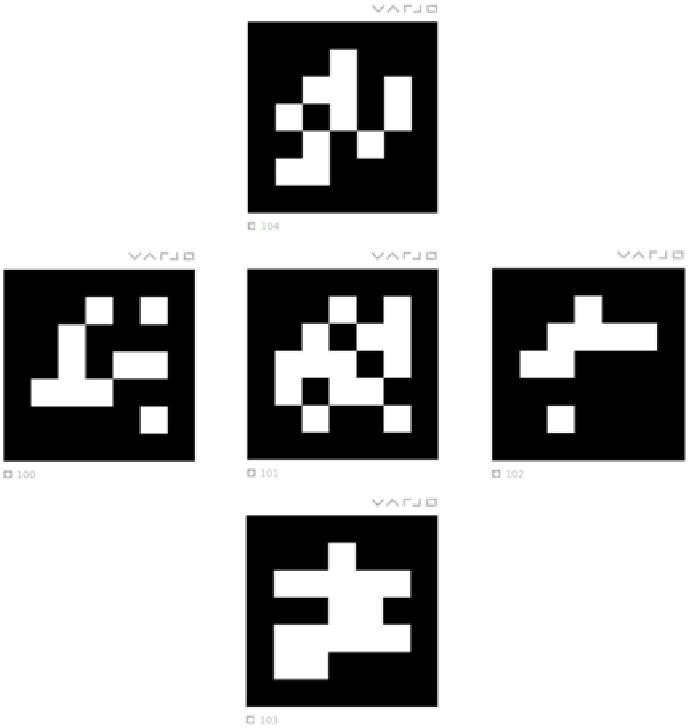


Figure 45 Faces of M1: ID 104 (top), ID 100 (left), ID 101 (center), ID 102 (right), ID 103 (bottom).

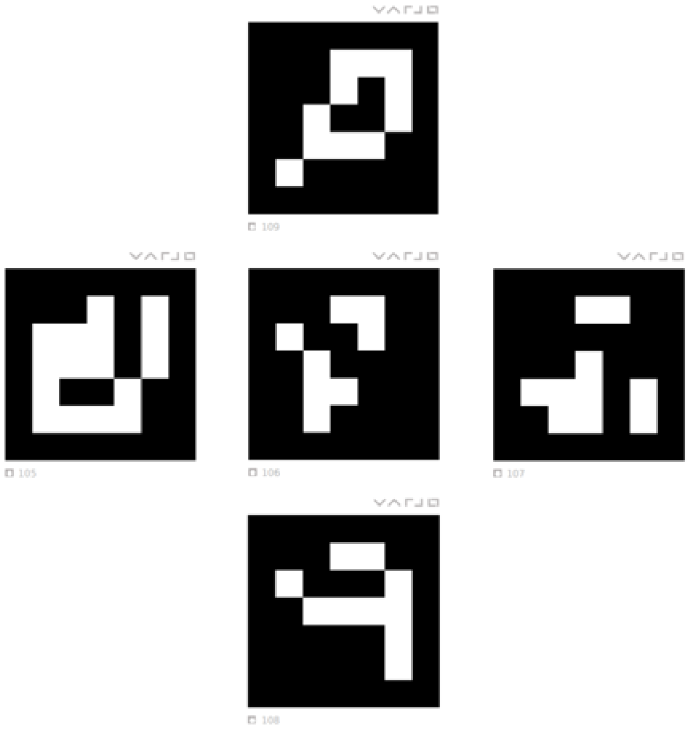


Figure 46 Faces of M2: ID 109 (top), ID 105 (left), ID 106 (center), ID 107 (right), ID 108 (bottom).

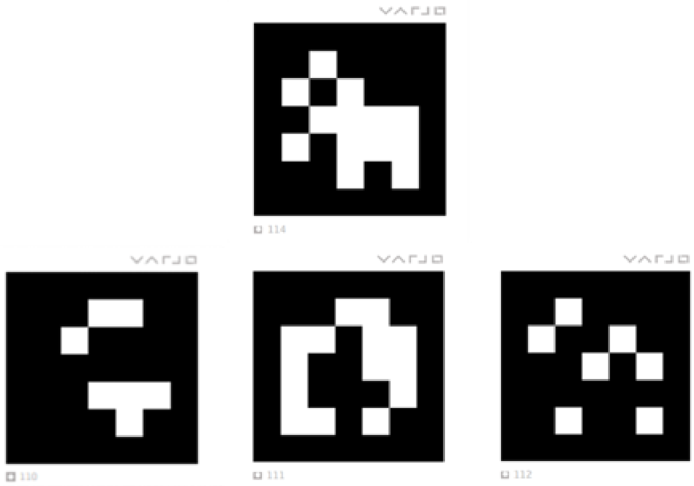


Figure 47 Faces of M3: ID 114 (top), ID 110 (left), ID 111 (center), ID 112 (right).

4.5 Scene Creation

First step in the scene creation is the import of the prefabs for the hand tracking in the Hierarchy of the scene. These prefabs, introduced in 4.1.3 contain all the assets, settings, and scripts necessary to enable the tracking of the user's hands when the application is launched. The initial outlook of the scene after the import of these prefabs is reported in Figure 48.

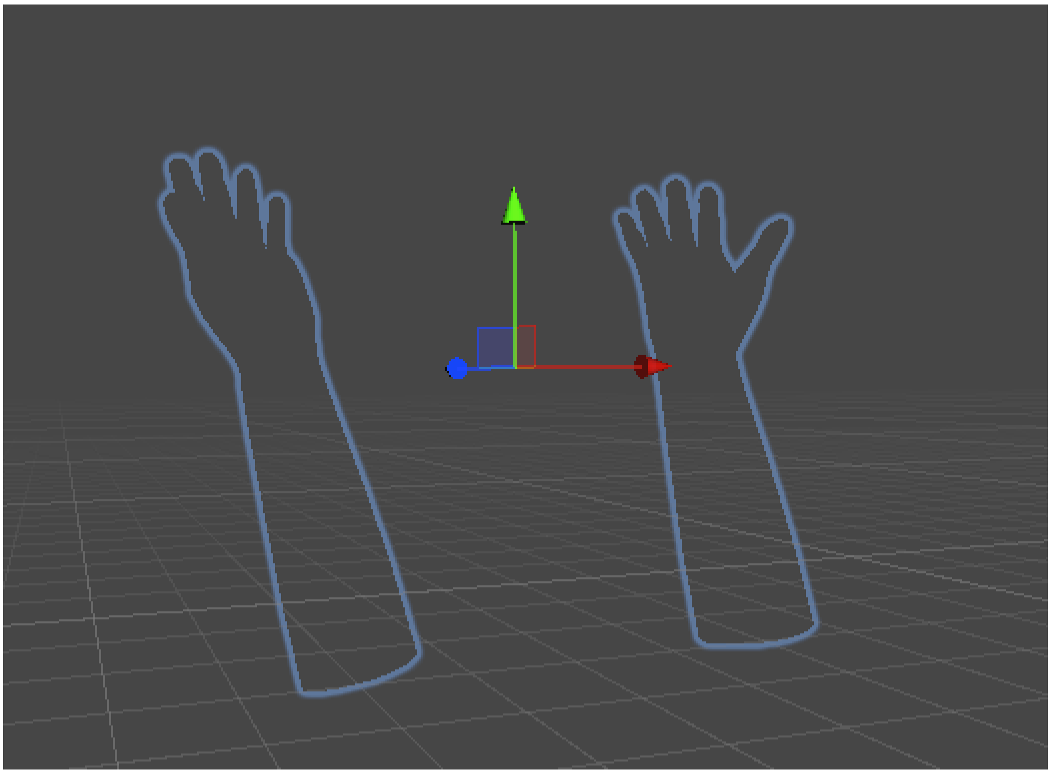


Figure 48 Prefab hands in the initial scene.

Then the targets are created. Following the process: GameObject → 3D Object → Quad, the 18 faces of the three marker cubes are introduced in the scene, rotated, and brought closer together to reconstruct the structure of the three cubes. Figure 49. The scale of the Quad is set to 0,0325 for the X and Y dimensions to match the real cube's side length. Figure 50. In the Hierarchy all the planes are set as child of the XR Rig. Each of these GameObjects will follow a Varjo Marker in the tracking space, so the GameObjects must be set as sibling of the XR camera. In the Inspector of each cube face, the static option next to the GameObject name is unflagged and in the material section the shader and the color are chosen. The shader chosen is URP/Unlit while the color is RGBA (0,0,0,0). Figure 50.

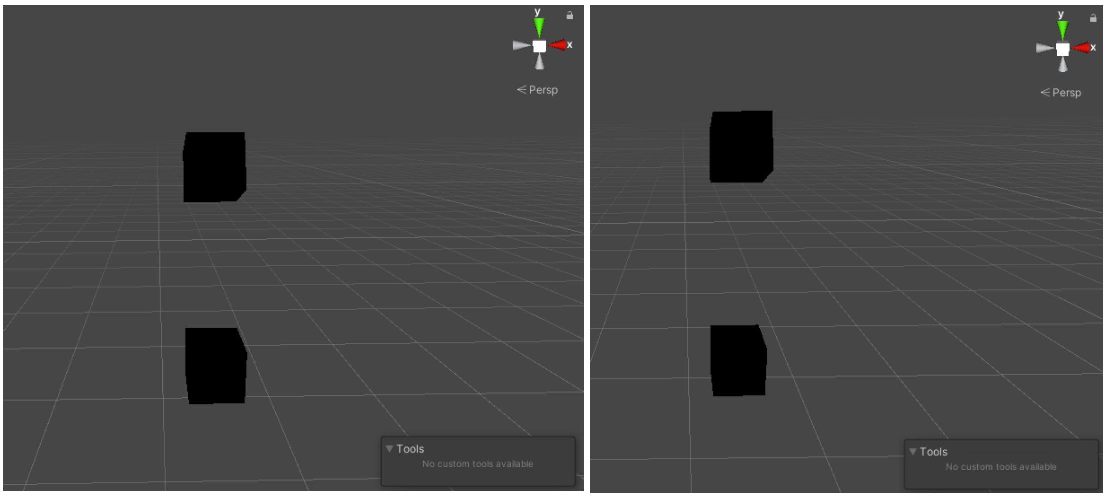
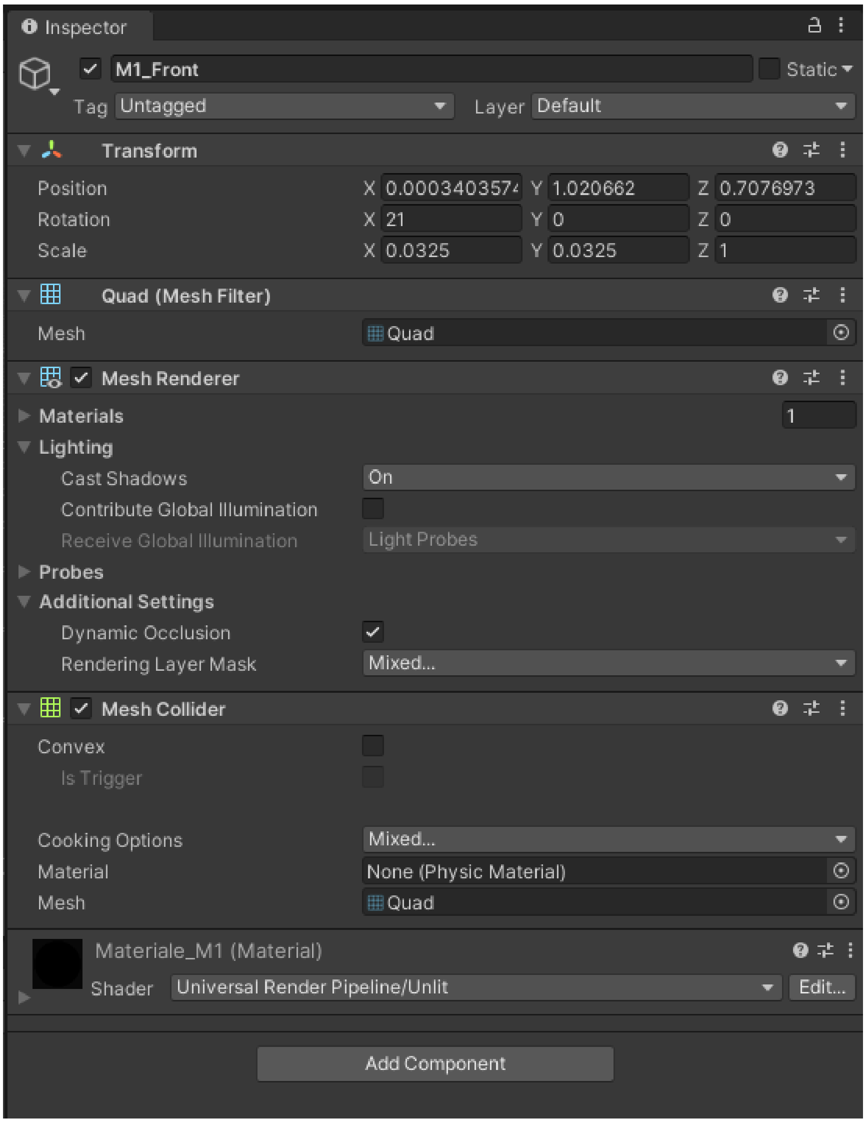


Figure 49 M1 and M2 (left), M1 and M3 (right)



Inspector

☑ M1_Front □ Static

Tag Untagged Layer Default

Transform

Position	X 0.000340357	Y 1.020662	Z 0.7076973
Rotation	X 21	Y 0	Z 0
Scale	X 0.0325	Y 0.0325	Z 1

Quad (Mesh Filter)

Mesh Quad

☑ Mesh Renderer

Materials 1

Lighting

Cast Shadows On

Contribute Global Illumination □

Receive Global Illumination Light Probes

Probes

Additional Settings

Dynamic Occlusion ☑

Rendering Layer Mask Mixed...

☑ Mesh Collider

Convex □

Is Trigger □

Cooking Options Mixed...

Material None (Physic Material)

Mesh Quad

Materiale_M1 (Material)

Shader Universal Render Pipeline/Unlit Edit...

Add Component

Figure 50 Inspector of M1 frontal face.

To establish a correspondence between the GameObjects in the scene and the physical markers, the 'Varjo_Marker_Manager' script, responsible for managing the tracked markers in the scene, was assigned to the XR Rig. Subsequently, within the XR Rig's Inspector, in the fields generated by the script, the GameObject, its corresponding ID, and the tracking type (static or dynamic) were specified.

After, the virtual content developed in the previous work for the AR application is loaded in the scene. [1] The imported assets are the osteotomy lines and drilled holes, the TargetPosition cube hologram, the 3D arrows and the DeltaPanelOut. The first two elements are necessary for the mandibular and maxillary osteotomy while the other three are necessary for the repositioning of the maxilla. The scale of the TargetPosition cube hologram has been modified to match the dimension of the physical markers used in this application.

Even if the positioning of the holograms with respect to the targets depends on the position in the space of the physical markers and is regulated by script, to enhance clarity and resemblance to the previous AR application, the elements within the scene have been aligned. Figure 51 and Figure 52. In the material section of the Inspector, the colors of each hologram has been maintained while the shader has been changed with the URP/Unlit.

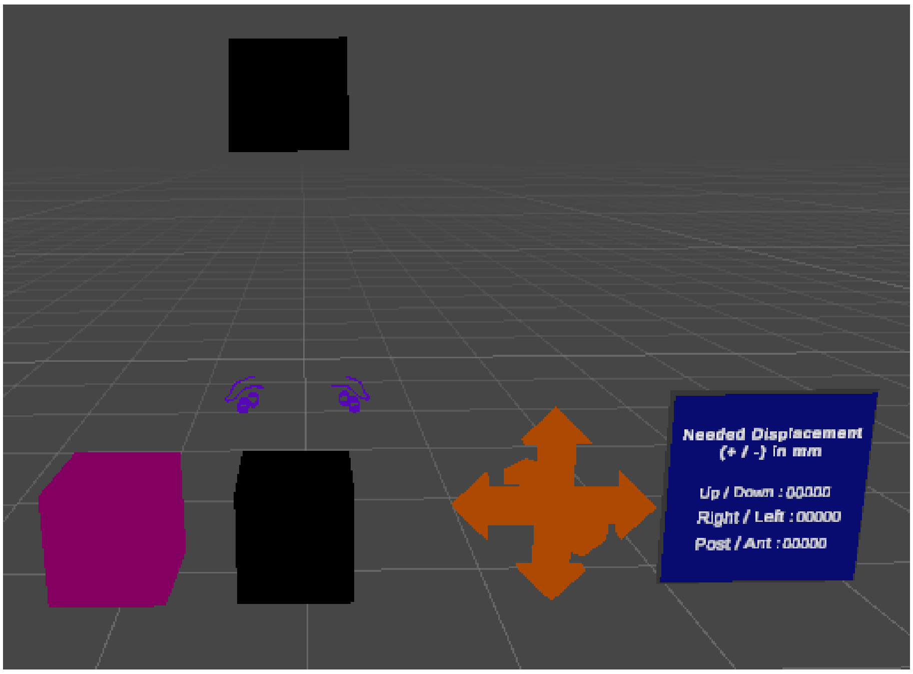


Figure 51 Scene with the holograms for maxillary osteotomy and repositioning outlook.

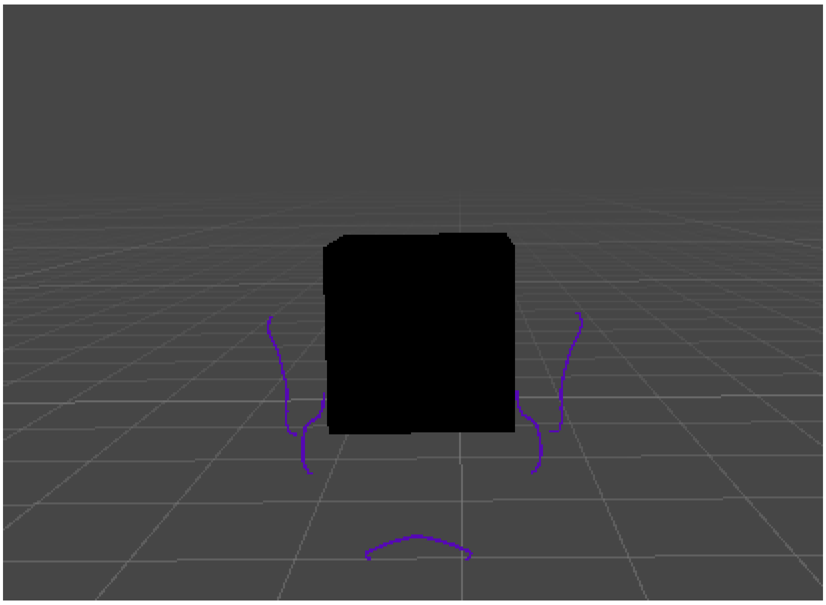


Figure 52 Scene with the holograms with the mandibular osteotomy outlook.

Three empty objects, M1, M2 and M3, and three cubes, Cube, Cube2 and Cube3, are added in the scene. To each empty object is assigned the script "CalcolaCentroCubo", that allow to compute the center of the physical marker and to position at that point a cube hologram, respectively Cube, Cube2 and Cube3.

Last step is creation of the three buttons on the Panel UI imported in the Attachment Hand prefab. In that panel there was already two buttons, a third button is added by copying and pasting one of them. On each button, the name of one of the three modalities is specified. Figure 53.

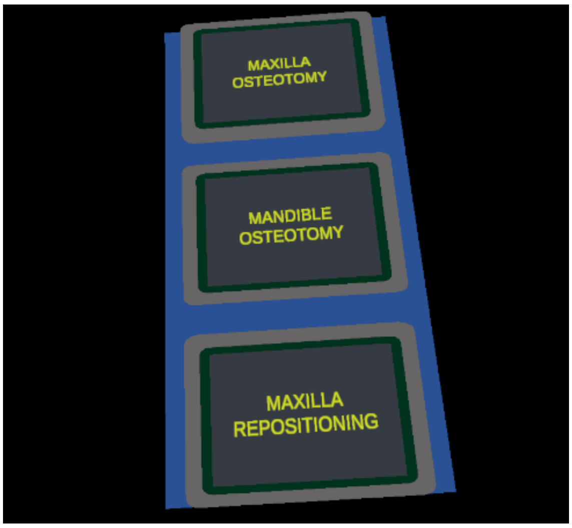
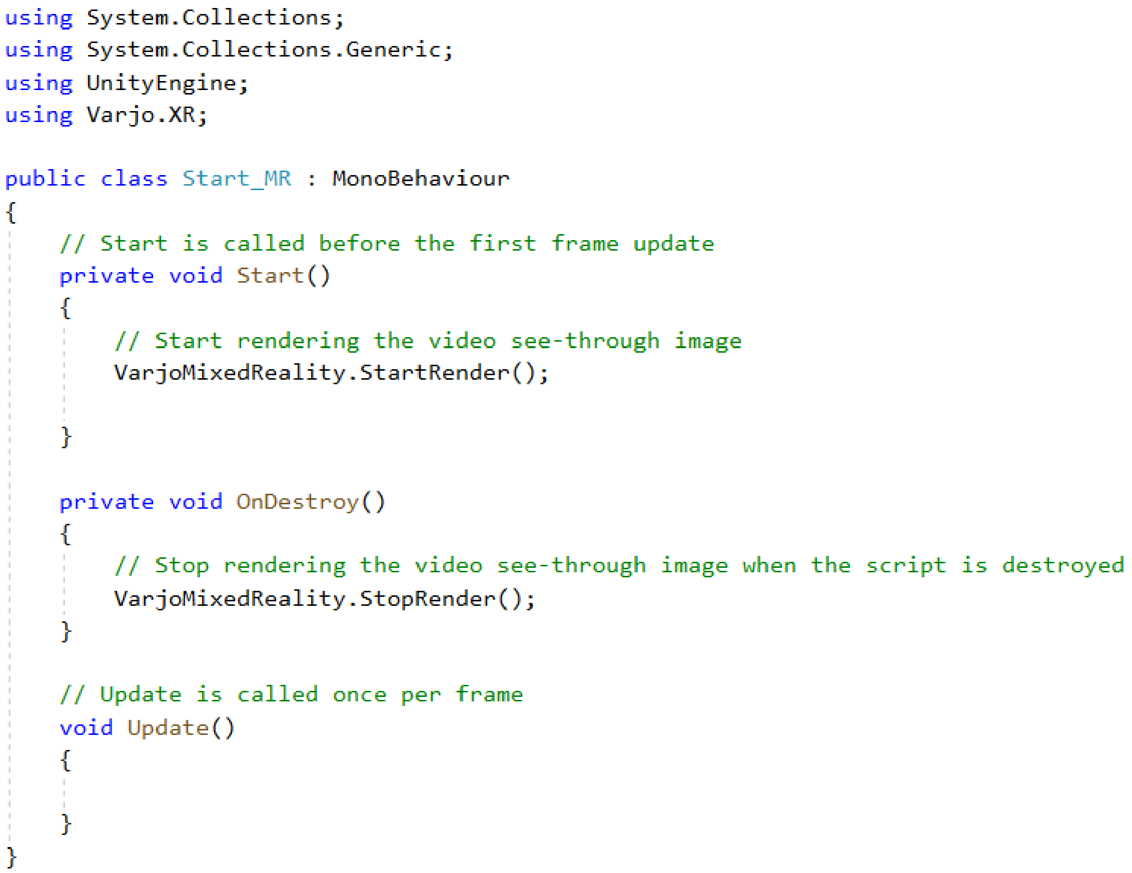


Figure 53 Panel with the buttons to select the modality.

4.6 Scripts

To mediate the functionalities expected by the MR application, six C# scripts have been implemented in Visual Studio.

The first one is Start_MR, it enables the video pass-through telling to Varjo to start the mixed reality streaming. First thing to do is to add the Varjo Plugin writing “using Varjo.XR” to enable all the Varjo XR functions in the project. Then in the start, whenever the scene initializes, the video see-through cameras are enabled calling the function “VarjoMixedReality.StartRender”. To stop the rendering of the video pass-through cameras, the private method “OnDestroy()” is added, here the function “VarjoMixedReality.StopRender” is called to do that. Figure 53. The full script is reported in Figure 54. This script is assigned to the XR Rig.



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Varjo.XR;

public class Start_MR : MonoBehaviour
{
    // Start is called before the first frame update
    private void Start()
    {
        // Start rendering the video see-through image
        VarjoMixedReality.StartRender();

    }

    private void OnDestroy()
    {
        // Stop rendering the video see-through image when the script is destroyed
        VarjoMixedReality.StopRender();
    }

    // Update is called once per frame
    void Update()
    {

    }
}
```

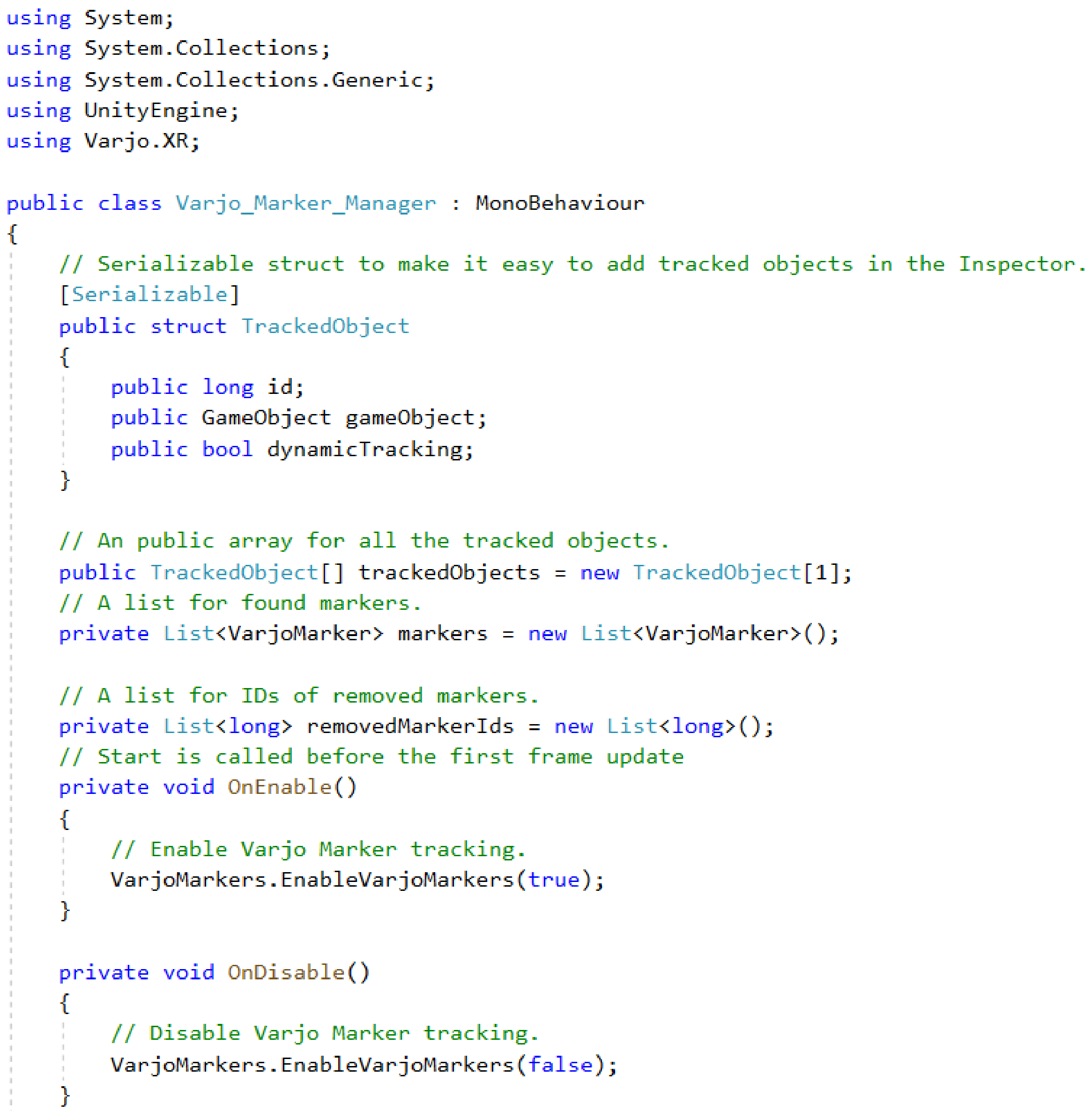
Figure 54 Start_MR script.

Second script, named as Varjo_Marker_Manager, is a manager script to handle all the tracked markers in the Unity scene. In the first part of the script an array of structs to store

ID-GameObject pairs and a Boolean for controlling the tracking mode of the marker are defined. The array and the tracking mode can be edited in the Unity Inspector. Since the manager script needs to keep track of all the markers in the environment, one list to store found markers and another one for the IDs of markers that have detected before but have not been visible for longer than the duration of the marker timeout, are added. Then, before getting the markers data, to have Varjo Markers enabled only when the object is active and enabled, two private methods, OnEnable() and OnDisable(), are used. These methods employ the “VarjoMarkers.EnableVarjoMarkers()” function, specifying 'true' in OnEnable and 'false' in OnDisable within the brackets. Figure 55.

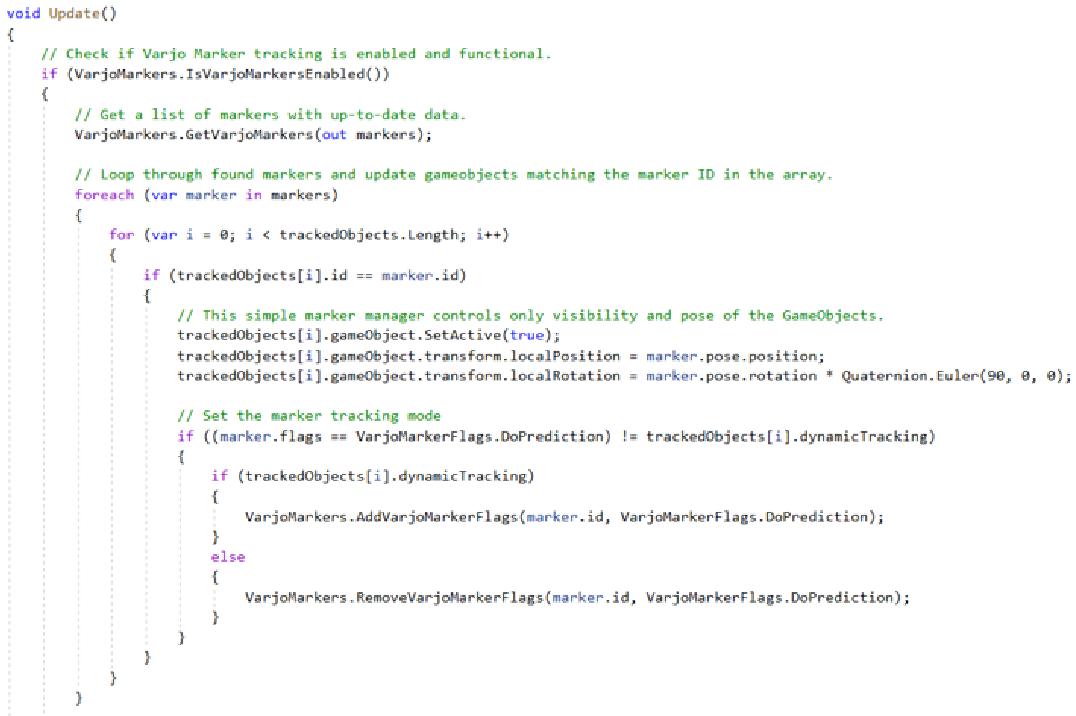
Now, in the 'Update', the logic of the script is implemented. If Varjo Marker tracking is enabled, a list of the latest marker data is obtained, and this updates all GameObjects in the 'trackedObjects' array whose ID matches that of a marker in the 'markers' array. First of all, it checks whether Varjo Marker tracking is enabled and functional. If it is, the script retrieves a list of markers with updated data. Then, the script iterates through the found markers and updates the GameObjects whose ID matches that of a marker in the "trackedObjects" array. It controls the visibility and position of GameObjects based on marker data and sets the tracking mode of the marker if necessary. After that, the scripts obtains a list of IDs for removed markers. It then iterates through the IDs of removed markers and deactivates the GameObjects that correspond to these IDs in the "trackedObjects" array. Figure 56.

This script is assigned to the XR Rig and the Inspector is filled with the required information as reported in Figure 57 and Figure 58.

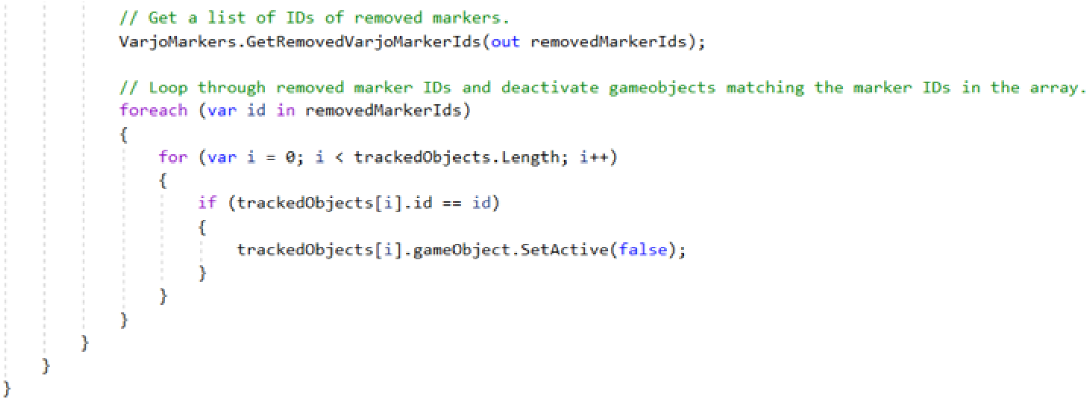


```
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Varjo.XR;

public class Varjo_Marker_Manager : MonoBehaviour
{
    // Serializable struct to make it easy to add tracked objects in the Inspector.
    [Serializable]
    public struct TrackedObject
    {
        public long id;
        public GameObject gameObject;
        public bool dynamicTracking;
    }

    // An public array for all the tracked objects.
    public TrackedObject[] trackedObjects = new TrackedObject[1];
    // A list for found markers.
    private List<VarjoMarker> markers = new List<VarjoMarker>();

    // A list for IDs of removed markers.
    private List<long> removedMarkerIds = new List<long>();
    // Start is called before the first frame update
    private void OnEnable()
    {
        // Enable Varjo Marker tracking.
        VarjoMarkers.EnableVarjoMarkers(true);
    }

    private void OnDisable()
    {
        // Disable Varjo Marker tracking.
        VarjoMarkers.EnableVarjoMarkers(false);
    }
```

Figure 55 Lines of code Varjo_Marker_Manager.

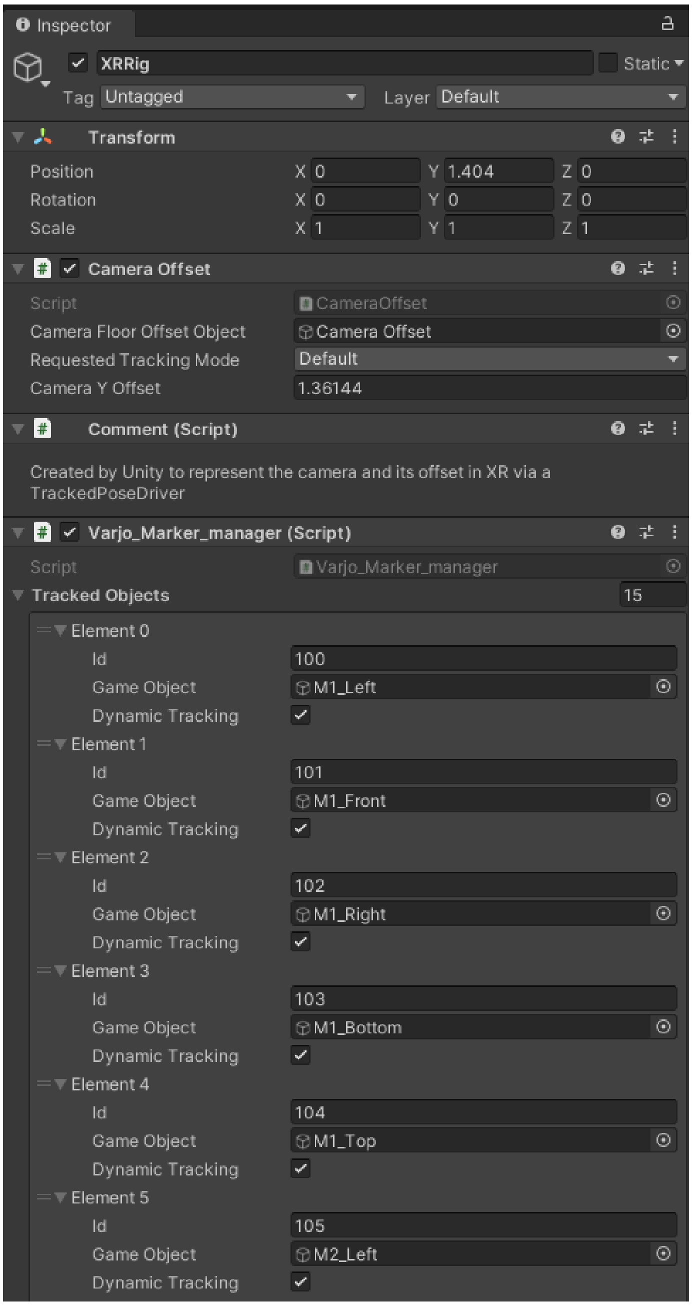


```
void Update()
{
    // Check if Varjo Marker tracking is enabled and functional.
    if (VarjoMarkers.IsVarjoMarkersEnabled())
    {
        // Get a list of markers with up-to-date data.
        VarjoMarkers.GetVarjoMarkers(out markers);

        // Loop through found markers and update gameobjects matching the marker ID in the array.
        foreach (var marker in markers)
        {
            for (var i = 0; i < trackedObjects.Length; i++)
            {
                if (trackedObjects[i].id == marker.id)
                {
                    // This simple marker manager controls only visibility and pose of the GameObjects.
                    trackedObjects[i].gameObject.SetActive(true);
                    trackedObjects[i].gameObject.transform.localPosition = marker.pose.position;
                    trackedObjects[i].gameObject.transform.localRotation = marker.pose.rotation * Quaternion.Euler(90, 0, 0);

                    // Set the marker tracking mode
                    if ((marker.flags == VarjoMarkerFlags.DoPrediction) != trackedObjects[i].dynamicTracking)
                    {
                        if (trackedObjects[i].dynamicTracking)
                        {
                            VarjoMarkers.AddVarjoMarkerFlags(marker.id, VarjoMarkerFlags.DoPrediction);
                        }
                        else
                        {
                            VarjoMarkers.RemoveVarjoMarkerFlags(marker.id, VarjoMarkerFlags.DoPrediction);
                        }
                    }
                }
            }
        }
```



```
            // Get a list of IDs of removed markers.
            VarjoMarkers.GetRemovedVarjoMarkerIds(out removedMarkerIds);

            // Loop through removed marker IDs and deactivate gameobjects matching the marker IDs in the array.
            foreach (var id in removedMarkerIds)
            {
                for (var i = 0; i < trackedObjects.Length; i++)
                {
                    if (trackedObjects[i].id == id)
                    {
                        trackedObjects[i].gameObject.SetActive(false);
                    }
                }
            }
        }
    }
}
```

Figure 56 Update of Varjo_Marker_Manager script.



Inspector

☑ XRRig □ Static

Tag Untagged Layer Default

Transform

Position X 0 Y 1.404 Z 0

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

☑ Camera Offset

Script CameraOffset

Camera Floor Offset Object Camera Offset

Requested Tracking Mode Default

Camera Y Offset 1.36144

Comment (Script)

Created by Unity to represent the camera and its offset in XR via a TrackedPoseDriver

☑ Varjo_Marker_manager (Script)

Script Varjo_Marker_manager

Tracked Objects 15

Element 0
Id 100
Game Object M1_Left
Dynamic Tracking ☑

Element 1
Id 101
Game Object M1_Front
Dynamic Tracking ☑

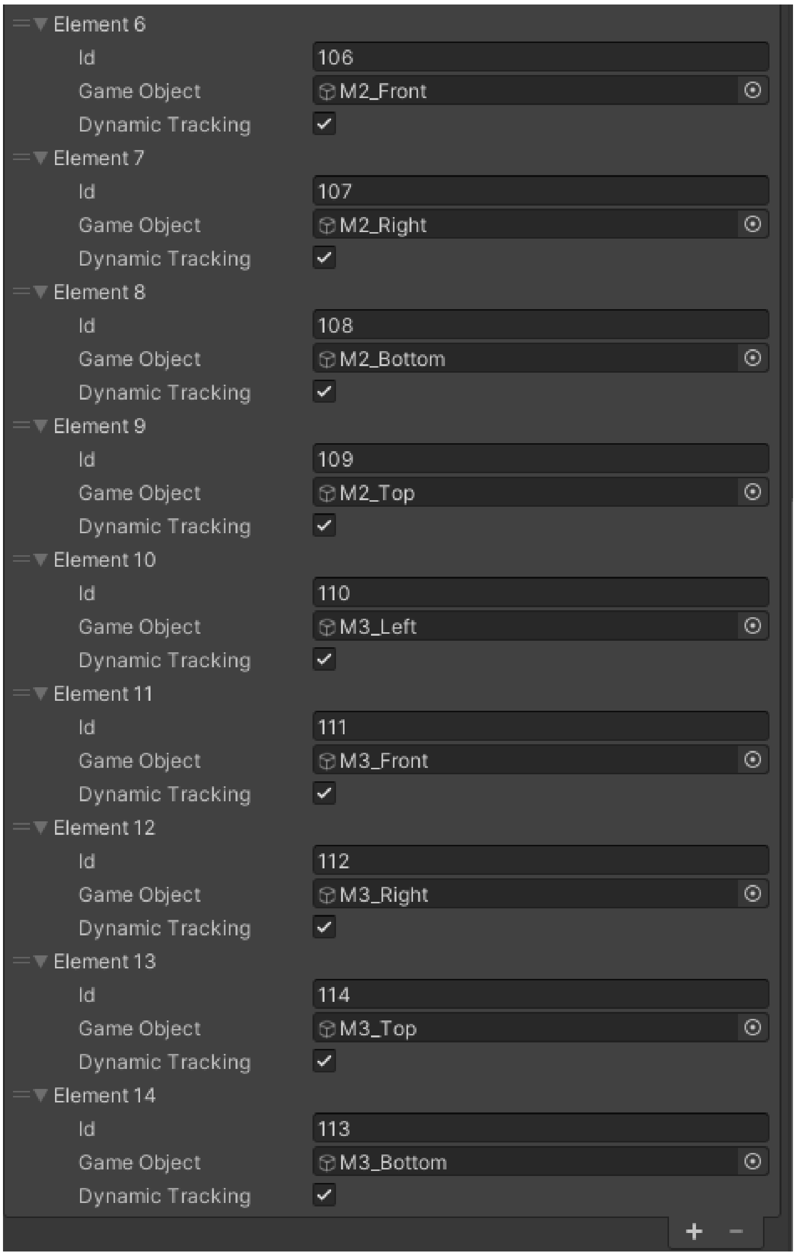
Element 2
Id 102
Game Object M1_Right
Dynamic Tracking ☑

Element 3
Id 103
Game Object M1_Bottom
Dynamic Tracking ☑

Element 4
Id 104
Game Object M1_Top
Dynamic Tracking ☑

Element 5
Id 105
Game Object M2_Left
Dynamic Tracking ☑

Figure 57 XR Rig Inspector.



Element 6
Id: 106
Game Object: M2_Front
Dynamic Tracking: ✓
Element 7
Id: 107
Game Object: M2_Right
Dynamic Tracking: ✓
Element 8
Id: 108
Game Object: M2_Bottom
Dynamic Tracking: ✓
Element 9
Id: 109
Game Object: M2_Top
Dynamic Tracking: ✓
Element 10
Id: 110
Game Object: M3_Left
Dynamic Tracking: ✓
Element 11
Id: 111
Game Object: M3_Front
Dynamic Tracking: ✓
Element 12
Id: 112
Game Object: M3_Right
Dynamic Tracking: ✓
Element 13
Id: 114
Game Object: M3_Top
Dynamic Tracking: ✓
Element 14
Id: 113
Game Object: M3_Bottom
Dynamic Tracking: ✓

Figure 58 XR Rig Inspector.

The third script is “CalcolaCentroCubo”, it aims to track the center of a cube based on its visible faces, and once this point is tracked, another object is positioned at that location. It is important to note that the physical marker is a cube, on which there is a specific VarjoMarker on each face, and the Varjo XR3 is capable of tracking VarjoMarkers as separate objects. So, it is necessary to reconstruct the structure of the physical cube based on the faces visible to the HMD. To achieve this, the center of the cube is calculated from the visible faces, and at this point, a cube of the same dimensions as the physical cube is displayed. In Figure 59, the activities of this script are summarized with an example in which there are three visible faces (three visible VarjoMarkers) of the physical cube. The system is able to track these faces and to compute the center of each face. Then from the computed centers, the center of the physical cube is reconstructed and in this point a virtual cube, of the same dimensions of the physical one, is placed.

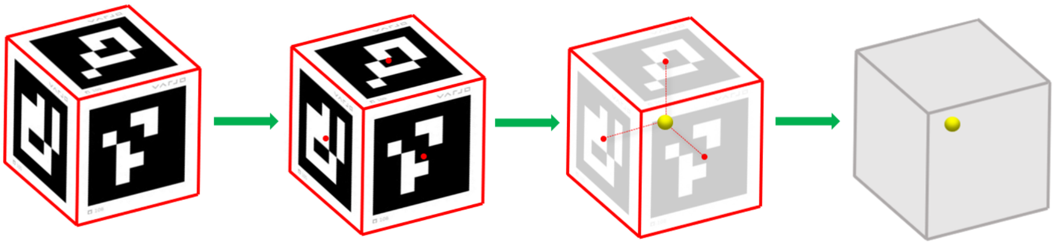
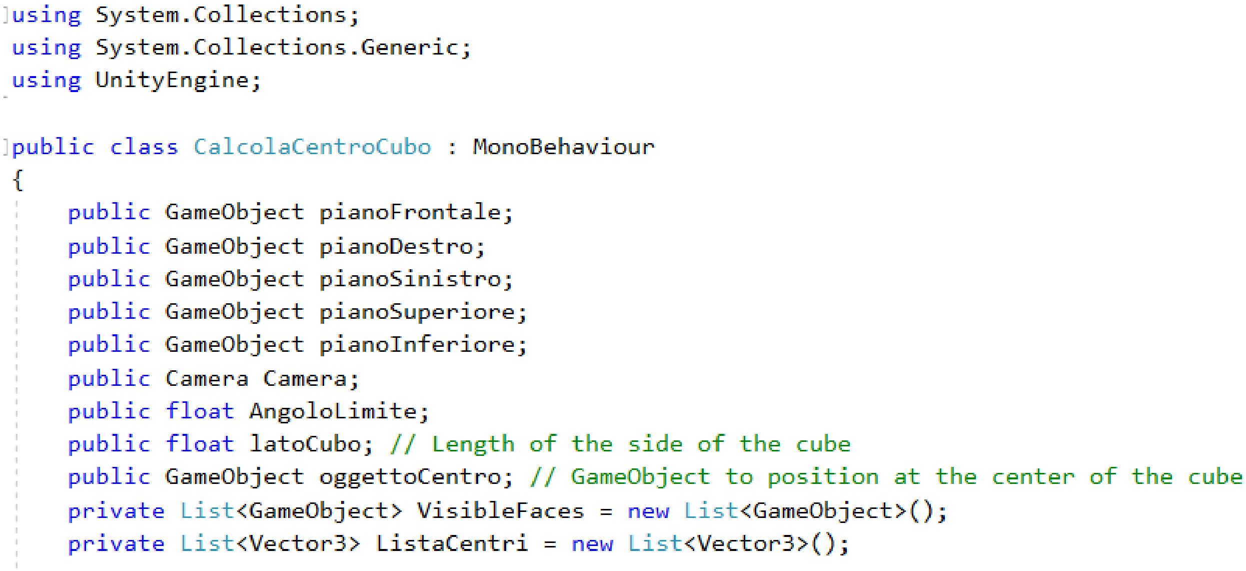


Figure 59 Summary of the 'CalcolaCentroCubo' script.

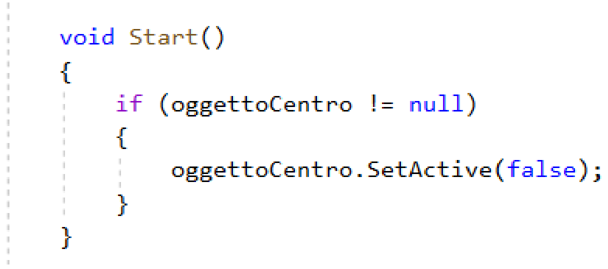
At the beginning of the script, a series of public variables and two empty lists are declared. The public variables, that can be set in Unity's Inspector, include the front, right, left, top, and bottom planes (the back plane is never visible), the length of the cube's side, the object to be positioned once the center is calculated, the main camera, and the angle limit. The empty lists are where the visible faces and the centers of each visible plane are added during each frame, which will be used to calculate the center of the cube. Figure 60.



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CalcolaCentroCubo : MonoBehaviour
{
    public GameObject pianoFrontale;
    public GameObject pianoDestro;
    public GameObject pianoSinistro;
    public GameObject pianoSuperiore;
    public GameObject pianoInferiore;
    public Camera Camera;
    public float AngoloLimite;
    public float latoCubo; // Length of the side of the cube
    public GameObject oggettoCentro; // GameObject to position at the center of the cube
    private List<GameObject> VisibleFaces = new List<GameObject>();
    private List<Vector3> ListaCentri = new List<Vector3>();
```

Figure 60 Variables of Calcola_Centro_Cubo script.

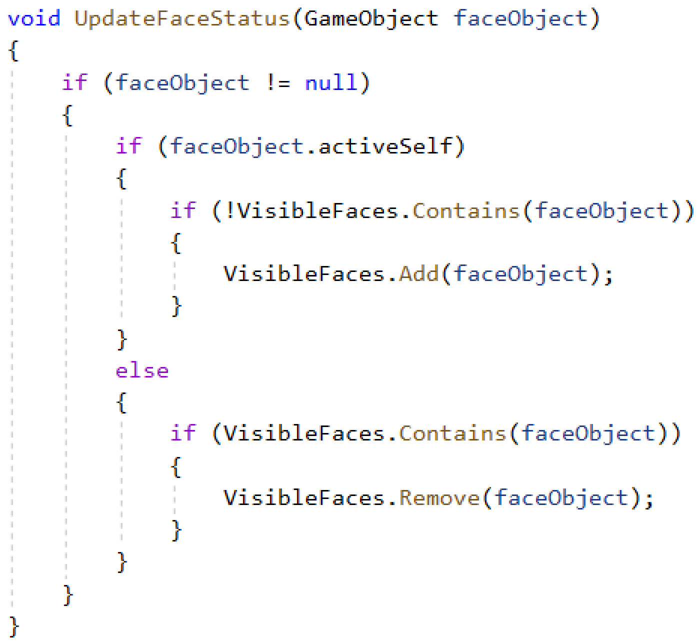
In the Start it checks whether the object to be positioned at the center of the cube is not null, and if it isn't, it is deactivated. This deactivation is done because it will later be activated and displayed once the center is found. Figure 61.



```
void Start()
{
    if (oggettoCentro != null)
    {
        oggettoCentro.SetActive(false);
    }
}
```

Figure 61 Start of Calcola_Centro_Cubo script.

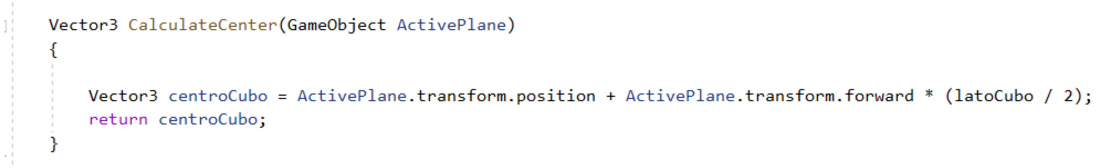
A function is defined “Update Face Status” that takes a GameObject called “faceObject”, as a parameter and updates a list called “VisibleFaces” based on whether the “faceObject” is active or not. It first checks if the “faceObject” parameter is not null, then checks if it is active. If these two conditions are respected, it checks if the “faceObject” is already part of the list “VisibleFaces” and if not, it adds it to the list. Otherwise, if “faceObject” is not active and is part of the list, it is removed. Figure 62.



```
void UpdateFaceStatus(GameObject faceObject)
{
    if (faceObject != null)
    {
        if (faceObject.activeSelf)
        {
            if (!VisibleFaces.Contains(faceObject))
            {
                VisibleFaces.Add(faceObject);
            }
        }
        else
        {
            if (VisibleFaces.Contains(faceObject))
            {
                VisibleFaces.Remove(faceObject);
            }
        }
    }
}
```

Figure 62 Lines of code of the UpdateFaceStatus method.

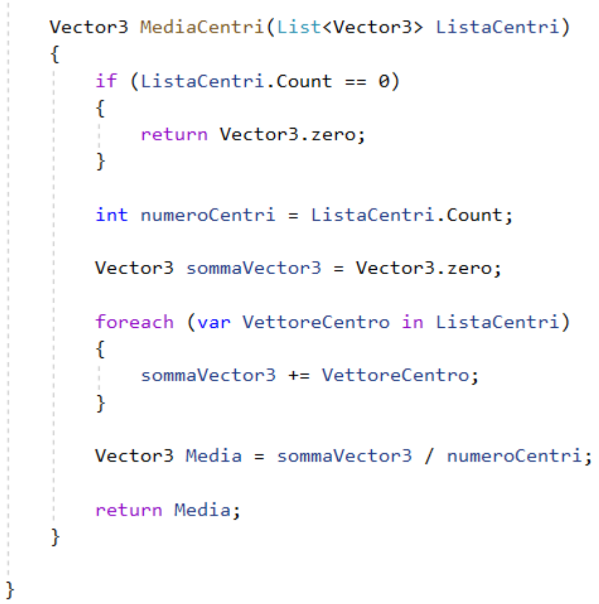
Then a method is defined, “CalculateCenter”, which calculates the center of a cube starting from a GameObject called “ActivePlane”. The position vector of the center of the cube is computed as the sum between the position of “ActivePlane” adding half of the cube’s side length in the forward direction. This method returns a vector with the coordinates of the cube center starting from the knowledge of the position of only one plane. Figure 63.



```
Vector3 CalculateCenter(GameObject ActivePlane)
{

    Vector3 centroCubo = ActivePlane.transform.position + ActivePlane.transform.forward * (latoCubo / 2);
    return centroCubo;
}
```

Figure 63 Lines of code of the CalculateCenter method.

One other method is defined, “MediaCentri” that takes as input a list of Vector3 called “ListaCentri”. First of all, it counts the number of elements in the list, if it is empty return a null vector. Otherwise, it counts the number of elements in the list, initialize a vector called “sommaVector3” as a null vector. Then each vector in “ListaCentri” it summed to “sommaVector3”, and “sommaVector3” is divided by the number of elements in “ListaCentri” to compute the mean vector, “Media”. At the end the “Media” vector is returned. Figure 64.



```
    Vector3 MediaCentri(List<Vector3> ListaCentri)
    {
        if (ListaCentri.Count == 0)
        {
            return Vector3.zero;
        }

        int numeroCentri = ListaCentri.Count;

        Vector3 sommaVector3 = Vector3.zero;

        foreach (var VettoreCentro in ListaCentri)
        {
            sommaVector3 += VettoreCentro;
        }

        Vector3 Media = sommaVector3 / numeroCentri;

        return Media;
    }

}
```

Figure 64 Lines of code of the MediaCentri method.

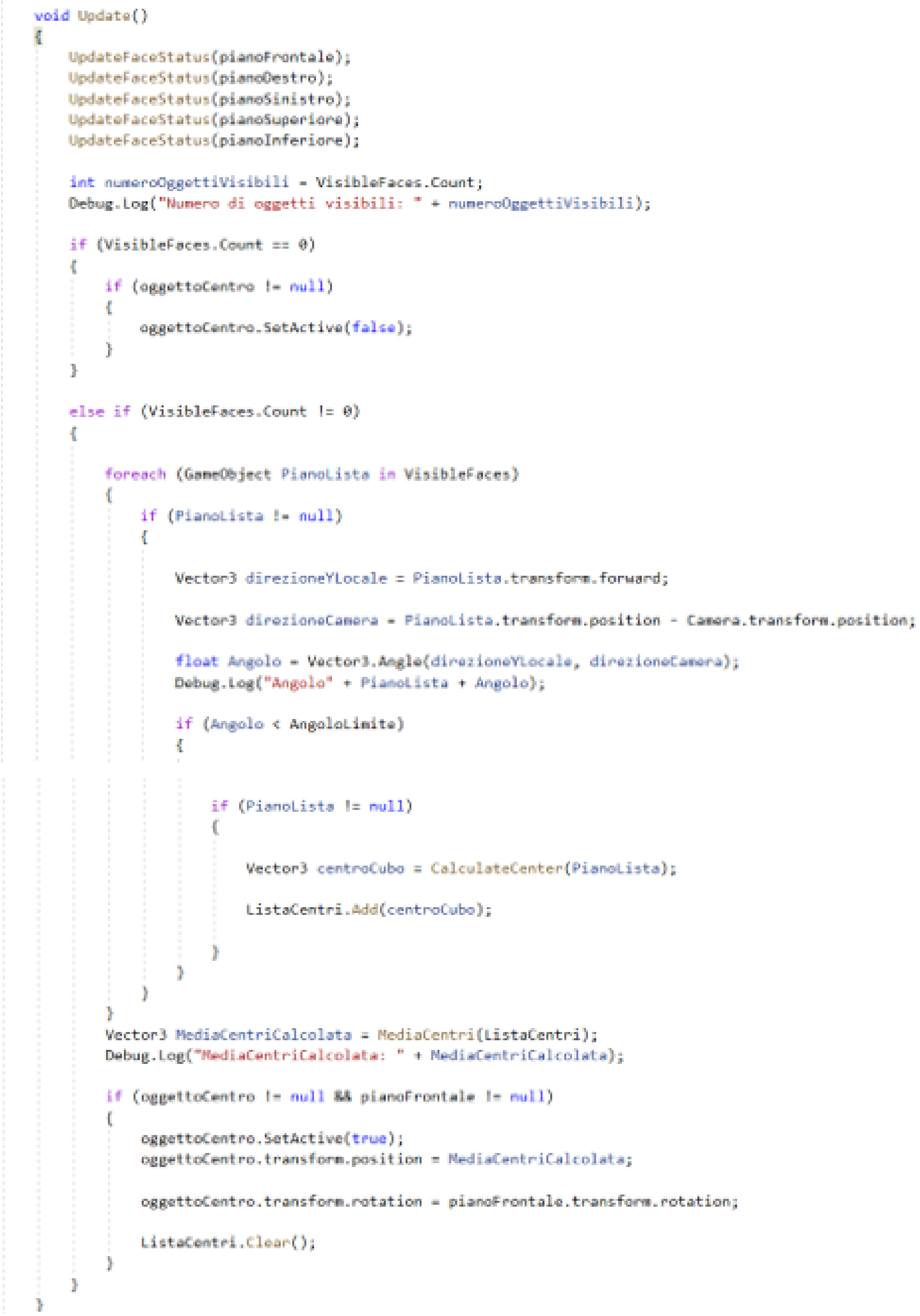
Now, let’s analyse the Update to understand what this script does at each frame. Initially, recalling the function “UpdateFaceStatus”, it checks the status of the front, right, left, top and bottom planes and according to the function, these planes are added o removed to the “VisibleFaces” list. The number of elements inside the list is counted and displayed on the console.

Then if “VisibleFaces” list is empty, and the GameObject “oggettoCentro” is not null, it is deactivated. Otherwise, for each visible plane inside the list, the script calculates the direction of the local Y axis of the plane, “direzioneYLocale”, and the direction between the visible plane and the Main Camera’s point of view, “direzioneCamera”. The angle between these two directions is calculated and displayed on the console. If this angle is smaller than a predefined angle, “AngoloLimite”, and if the plane is not null, the “CalculateCenter” method is called computing the center of the cube starting from that plane. The coordinates of this calculated center are then added to the list “ListaCentri”.

Then according to the vectors in the list “ListaCentri”, the method “MediaCentri is called to compute the mean of these vectors. This allows to determine the position of the cube’s center by averaging the information obtained from each visible plane.

Now, if the object to position at the center of the cube and the frontal plane are not null, the object is activated and positioned at the computed center of the cube and oriented according to the frontal plane.

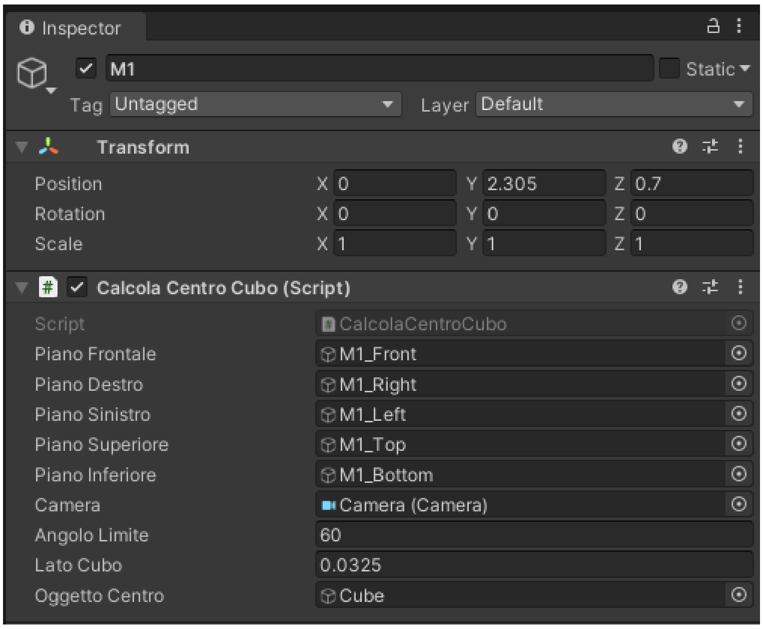
At the end, at each frame the list “ListaCentri” is cleared to compute the center of the cube using only the planes that are actually visible. Figure 65.



```
void Update()
{
    UpdateFaceStatus(pianoFrontale);
    UpdateFaceStatus(pianoDestro);
    UpdateFaceStatus(pianoSinistro);
    UpdateFaceStatus(pianoSuperiore);
    UpdateFaceStatus(pianoInferiore);

    int numeroOggettiVisibili = VisibleFaces.Count;
    Debug.Log("Numero di oggetti visibili: " + numeroOggettiVisibili);

    if (VisibleFaces.Count == 0)
    {
        if (oggettoCentro != null)
        {
            oggettoCentro.SetActive(false);
        }
    }

    else if (VisibleFaces.Count != 0)
    {

        foreach (GameObject PianoLista in VisibleFaces)
        {
            if (PianoLista != null)
            {

                Vector3 direzioneYLocale = PianoLista.transform.forward;

                Vector3 direzioneCamera = PianoLista.transform.position - Camera.transform.position;

                float Angolo = Vector3.Angle(direzioneYLocale, direzioneCamera);
                Debug.Log("Angolo" + PianoLista + Angolo);

                if (Angolo < AngoloLimite)
                {


                    if (PianoLista != null)
                    {

                        Vector3 centroCubo = CalculateCenter(PianoLista);

                        ListaCentri.Add(centroCubo);

                    }
                }
            }
        }
        Vector3 MediaCentriCalcolata = MediaCentri(ListaCentri);
        Debug.Log("MediaCentriCalcolata: " + MediaCentriCalcolata);

        if (oggettoCentro != null && pianoFrontale != null)
        {
            oggettoCentro.SetActive(true);
            oggettoCentro.transform.position = MediaCentriCalcolata;

            oggettoCentro.transform.rotation = pianoFrontale.transform.rotation;

            ListaCentri.Clear();
        }
    }
}
```

Figure 65 Update of CalcolaCentroCubo script.

This script is assigned to the empty objects M1, M2 and M3. Figure display the Inspectors for M1, M2, and M2 where the script's public fields are filled with the necessary information. Figure 66, Figure 67, and Figure 68.



Inspector

☑ M1 □ Static

Tag Untagged Layer Default

Transform

Position X 0 Y 2.305 Z 0.7

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

☑ Calcola Centro Cubo (Script)

Script CalcolaCentroCubo

Piano Frontale M1_Front

Piano Destro M1_Right

Piano Sinistro M1_Left

Piano Superiore M1_Top

Piano Inferiore M1_Bottom

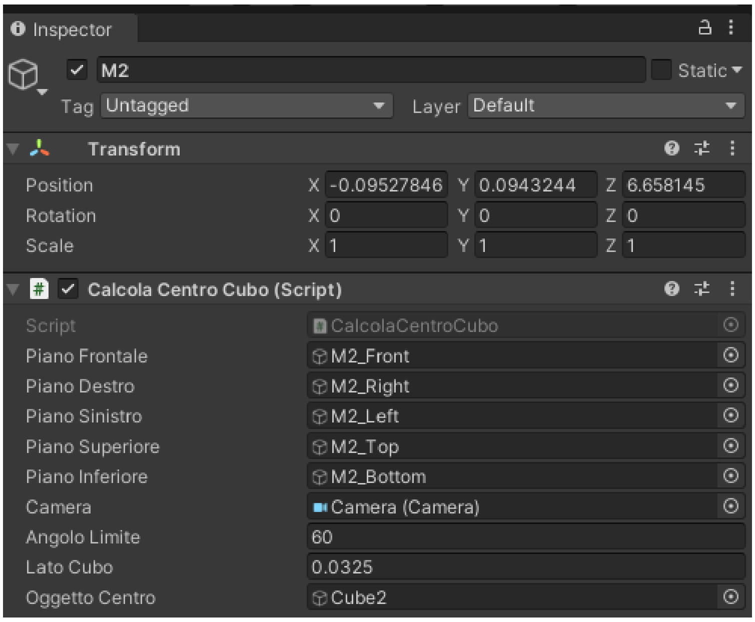
Camera Camera (Camera)

Angolo Limite 60

Lato Cubo 0.0325

Oggetto Centro Cube

Figure 66 Inspector of M1 empty object.



Inspector

☑ M2 □ Static

Tag Untagged Layer Default

Transform

Position X -0.09527846 Y 0.0943244 Z 6.658145

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

☑ Calcola Centro Cubo (Script)

Script CalcolaCentroCubo

Piano Frontale M2_Front

Piano Destro M2_Right

Piano Sinistro M2_Left

Piano Superiore M2_Top

Piano Inferiore M2_Bottom

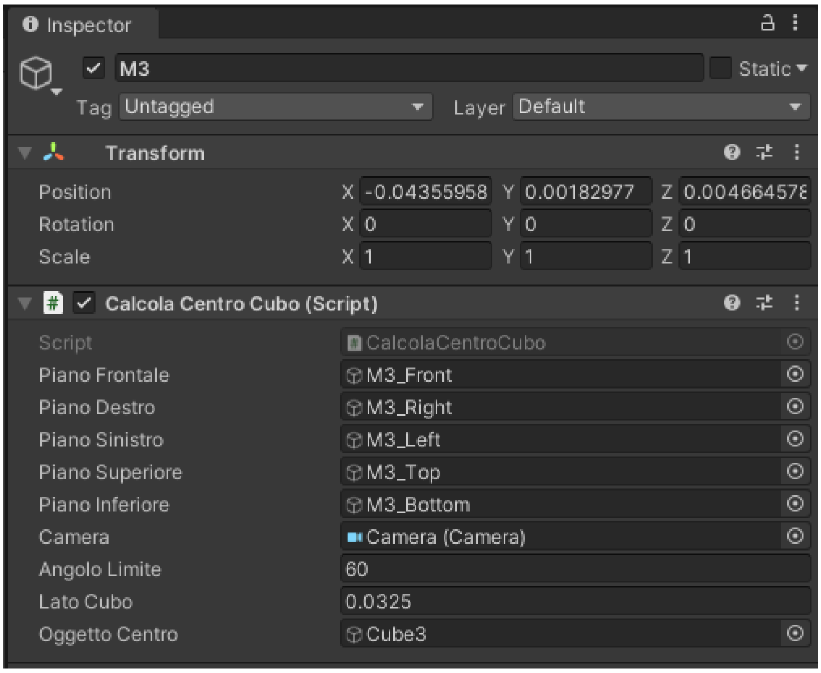
Camera Camera (Camera)

Angolo Limite 60

Lato Cubo 0.0325

Oggetto Centro Cube2

Figure 67 Inspector of M2 empty object.



M3 ☐ Static

Tag Untagged Layer Default

Transform

Position X -0.04355958 Y 0.00182977 Z 0.004664578

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

☑ Calcola Centro Cubo (Script)

Script: CalcolaCentroCubo

Piano Frontale: M3_Front

Piano Destro: M3_Right

Piano Sinistro: M3_Left

Piano Superiore: M3_Top

Piano Inferiore: M3_Bottom

Camera: Camera (Camera)

Angolo Limite: 60

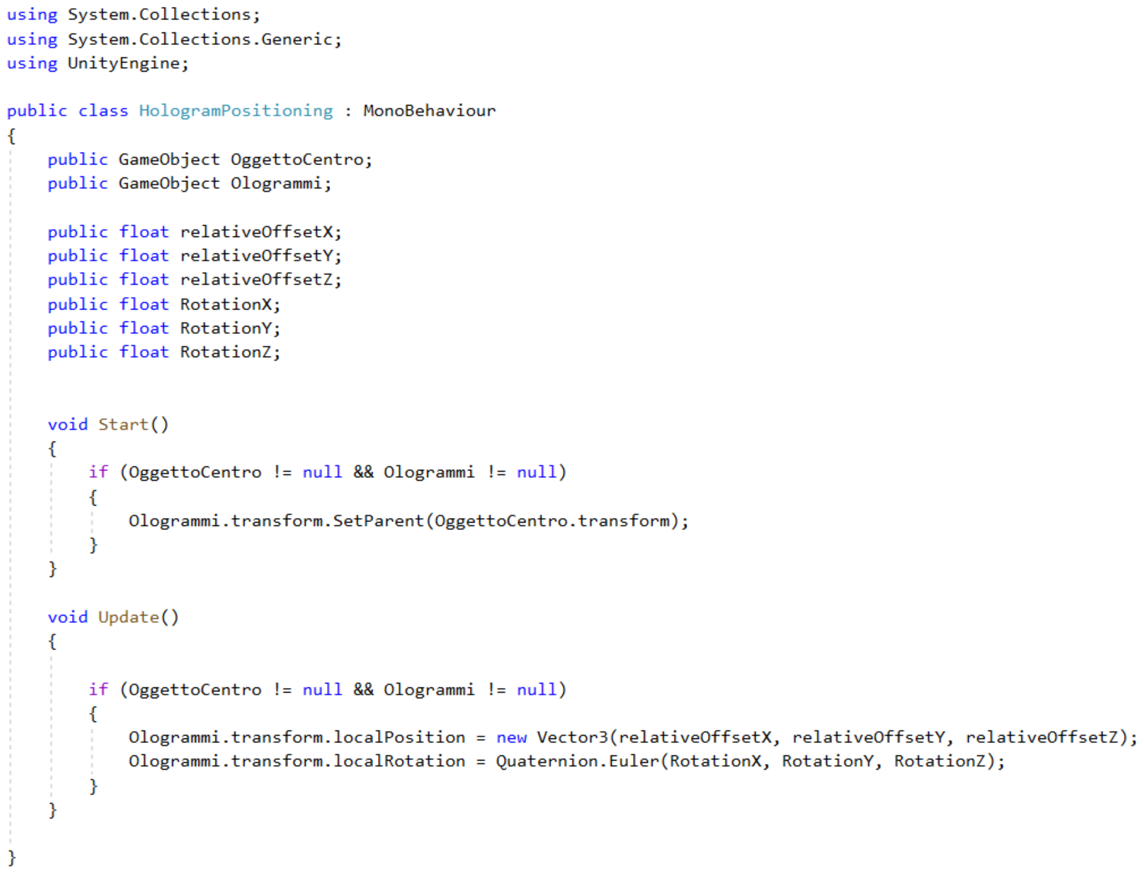
Lato Cubo: 0.0325

Oggetto Centro: Cube3

Figure 68 Inspector of M3 empty object.

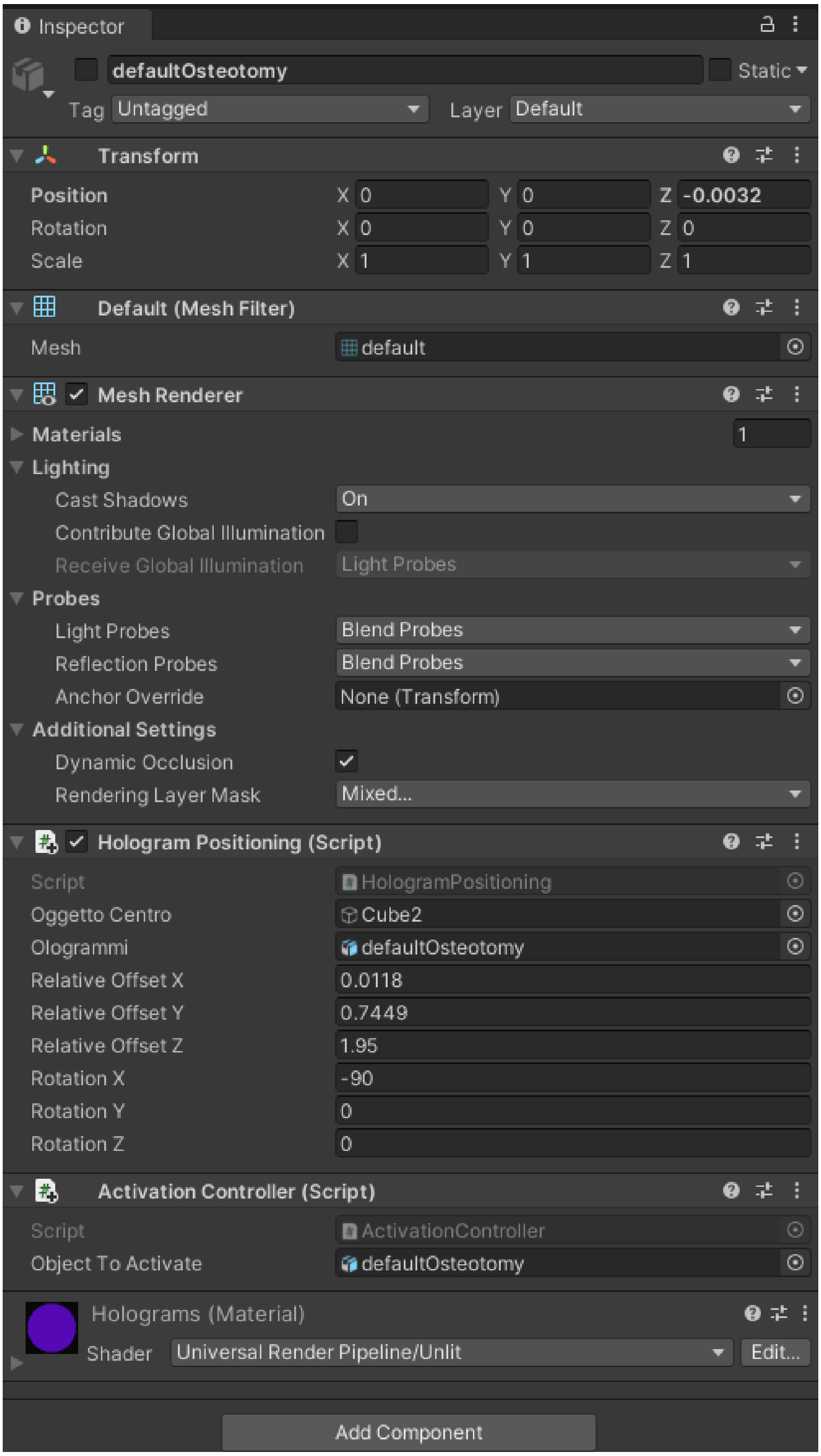
The fourth script is named "HologramPositioning" that is created to position a GameObject relative to another GameObject. At the beginning of the script, two public variables are defined: "OggettoCentro", the same GameObject of the script "CalcolaCentroCubo", and "Ologrammi", that is the GameObject that must be positioned and rotated in relation to "OggettoCentro". Additionally , there are variables that define: the offset position along the three axes (X, Y, and Z) for "Ologrammi" relative to "OggettoCentro"; the rotation angles around the three axes (X, Y, and Z) for "Ologrammi" relative to "OggettoCentro". In the Start function, the script checks if "OggettoCentro" and "Ologrammi" are not null, and if it is true, it set the "Ologrammi" GameObject as a child of "OggettoCentro". Then, in the Update, if the same condition present in the Start is valid, "Ologrammi" is positioned and rotated according to the values specified from the user in the Inspector of Unity. Figure 69.

This script is assigned to the Osteotomy Lines for the maxilla and Drilled Holes, in the Unity scene they are grouped in Holograms, to the Osteotomy Lines Mandible and to the TargetPosition, DeltaPanelOut, 3DArrows. The Inspector of each of these components is reported in Figure 70,71, 72 and 73.



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class HologramPositioning : MonoBehaviour
{
    public GameObject OggettoCentro;
    public GameObject Ologrammi;

    public float relativeOffsetX;
    public float relativeOffsetY;
    public float relativeOffsetZ;
    public float RotationX;
    public float RotationY;
    public float RotationZ;


    void Start()
    {
        if (OggettoCentro != null && Ologrammi != null)
        {
            Ologrammi.transform.SetParent(OggettoCentro.transform);
        }
    }

    void Update()
    {

        if (OggettoCentro != null && Ologrammi != null)
        {
            Ologrammi.transform.localPosition = new Vector3(relativeOffsetX, relativeOffsetY, relativeOffsetZ);
            Ologrammi.transform.localRotation = Quaternion.Euler(RotationX, RotationY, RotationZ);
        }
    }

}
```

Figure 69 HologramPositioning script.



Inspector

☐ defaultOsteotomy ☐ Static

Tag Untagged Layer Default

Transform

Position	X 0	Y 0	Z -0.0032
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Default (Mesh Filter)

Mesh default

☑ **Mesh Renderer**

Materials 1

Lighting

Cast Shadows On

Contribute Global Illumination ☐

Receive Global Illumination Light Probes

Probes

Light Probes Blend Probes

Reflection Probes Blend Probes

Anchor Override None (Transform)

Additional Settings

Dynamic Occlusion ☑

Rendering Layer Mask Mixed...

☑ **Hologram Positioning (Script)**

Script HologramPositioning

Oggetto Centro Cube2

Ologrammi defaultOsteotomy

Relative Offset X 0.0118

Relative Offset Y 0.7449

Relative Offset Z 1.95

Rotation X -90

Rotation Y 0

Rotation Z 0

Activation Controller (Script)

Script ActivationController

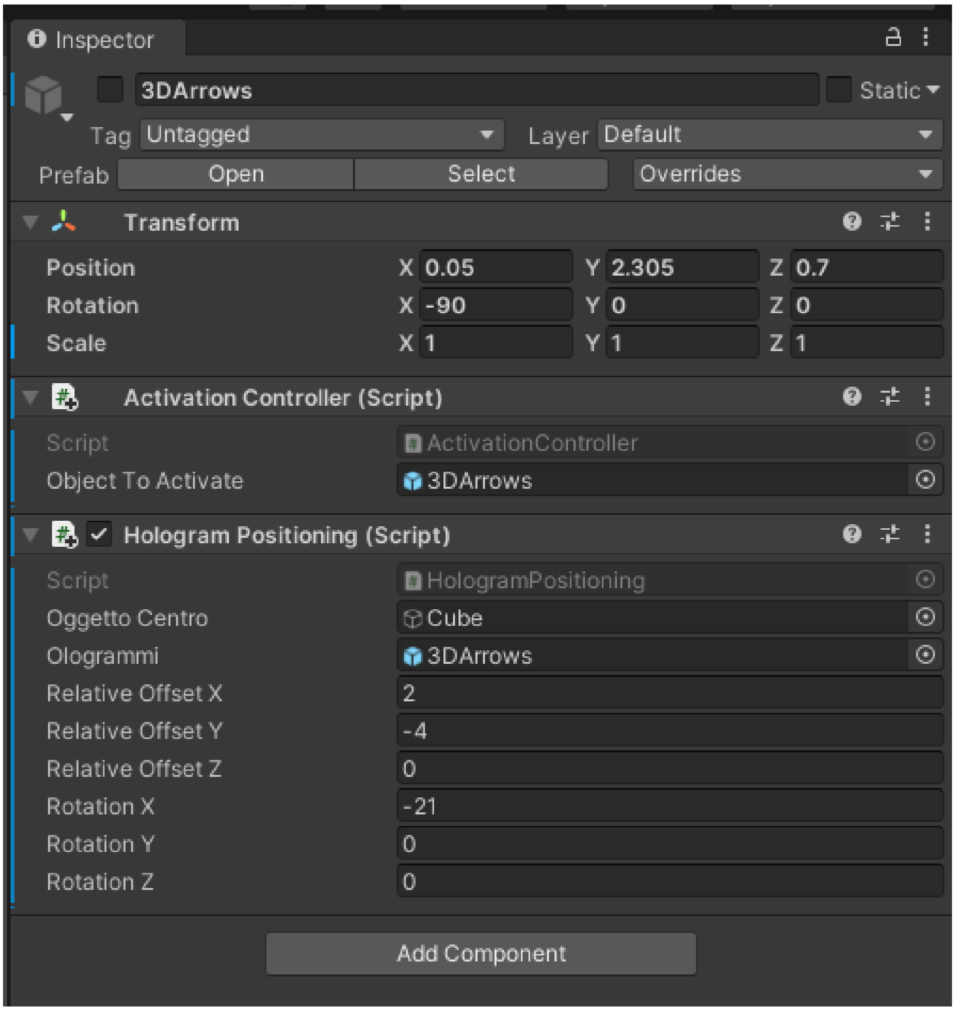
Object To Activate defaultOsteotomy

Holograms (Material)

Shader Universal Render Pipeline/Unlit Edit...

Add Component

Figure 70 Osteotomy lines and Drilled Holes Inspector.



3DArrows

Tag Untagged

Layer Default

Transform

Position X 0.05 Y 2.305 Z 0.7

Rotation X -90 Y 0 Z 0

Scale X 1 Y 1 Z 1

Activation Controller (Script)

Script ActivationController

Object To Activate 3DArrows

☑ Hologram Positioning (Script)

Script HologramPositioning

Oggetto Centro Cube

Ologrammi 3DArrows

Relative Offset X 2

Relative Offset Y -4

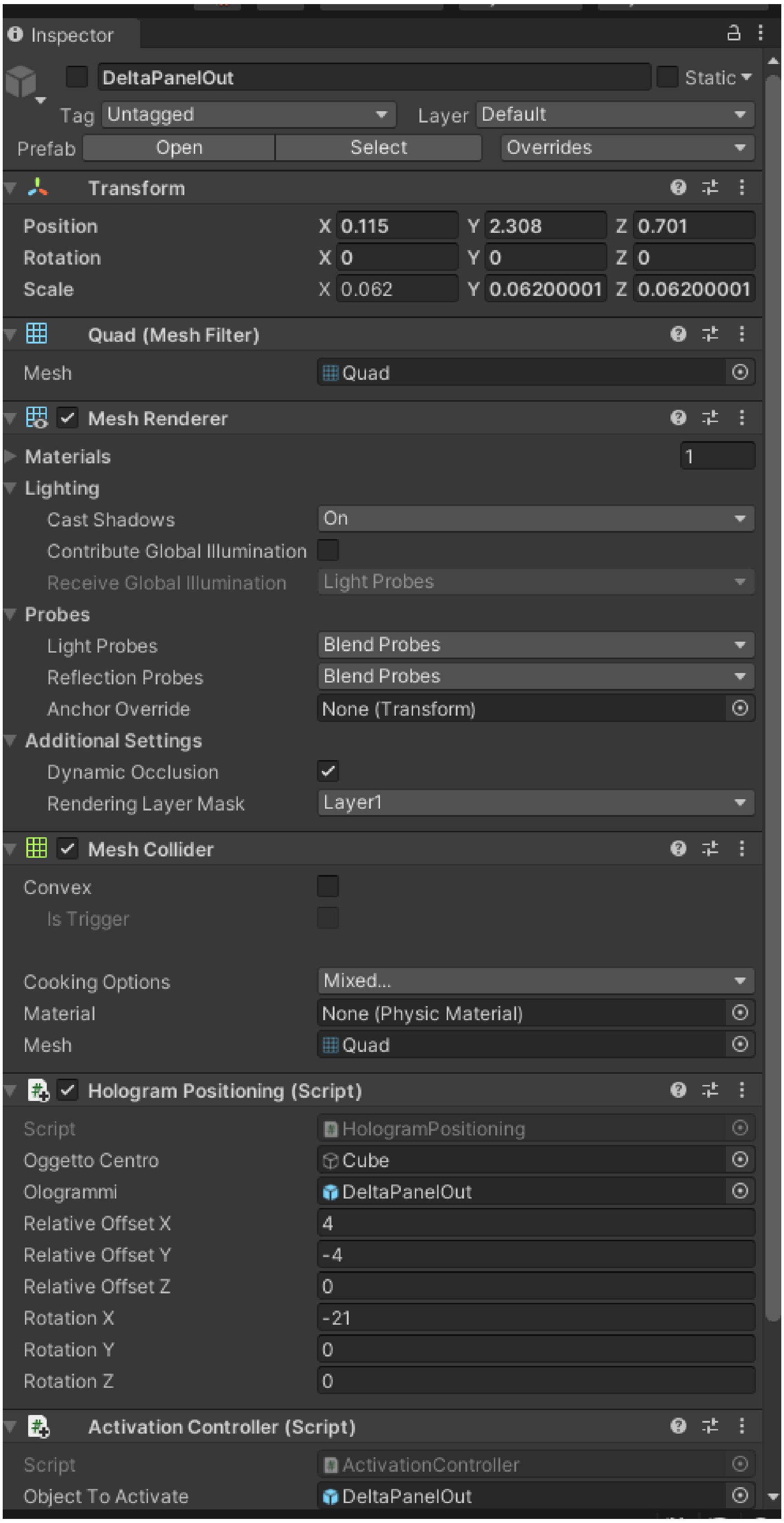
Relative Offset Z 0

Rotation X -21

Rotation Y 0

Rotation Z 0

Figure 71 3D Arrows Inspector.



Inspector

□ DeltaPanelOut □ Static

Tag Untagged Layer Default

Prefab Open Select Overrides

Transform

Position X 0.115 Y 2.308 Z 0.701

Rotation X 0 Y 0 Z 0

Scale X 0.062 Y 0.06200001 Z 0.06200001

Quad (Mesh Filter)

Mesh Quad

☑ **Mesh Renderer**

Materials 1

Lighting

Cast Shadows On

Contribute Global Illumination □

Receive Global Illumination Light Probes

Probes

Light Probes Blend Probes

Reflection Probes Blend Probes

Anchor Override None (Transform)

Additional Settings

Dynamic Occlusion ☑

Rendering Layer Mask Layer1

☑ **Mesh Collider**

Convex □

Is Trigger □

Cooking Options Mixed...

Material None (Physic Material)

Mesh Quad

☑ **Hologram Positioning (Script)**

Script HologramPositioning

Oggetto Centro Cube

Ologrammi DeltaPanelOut

Relative Offset X 4

Relative Offset Y -4

Relative Offset Z 0

Rotation X -21

Rotation Y 0

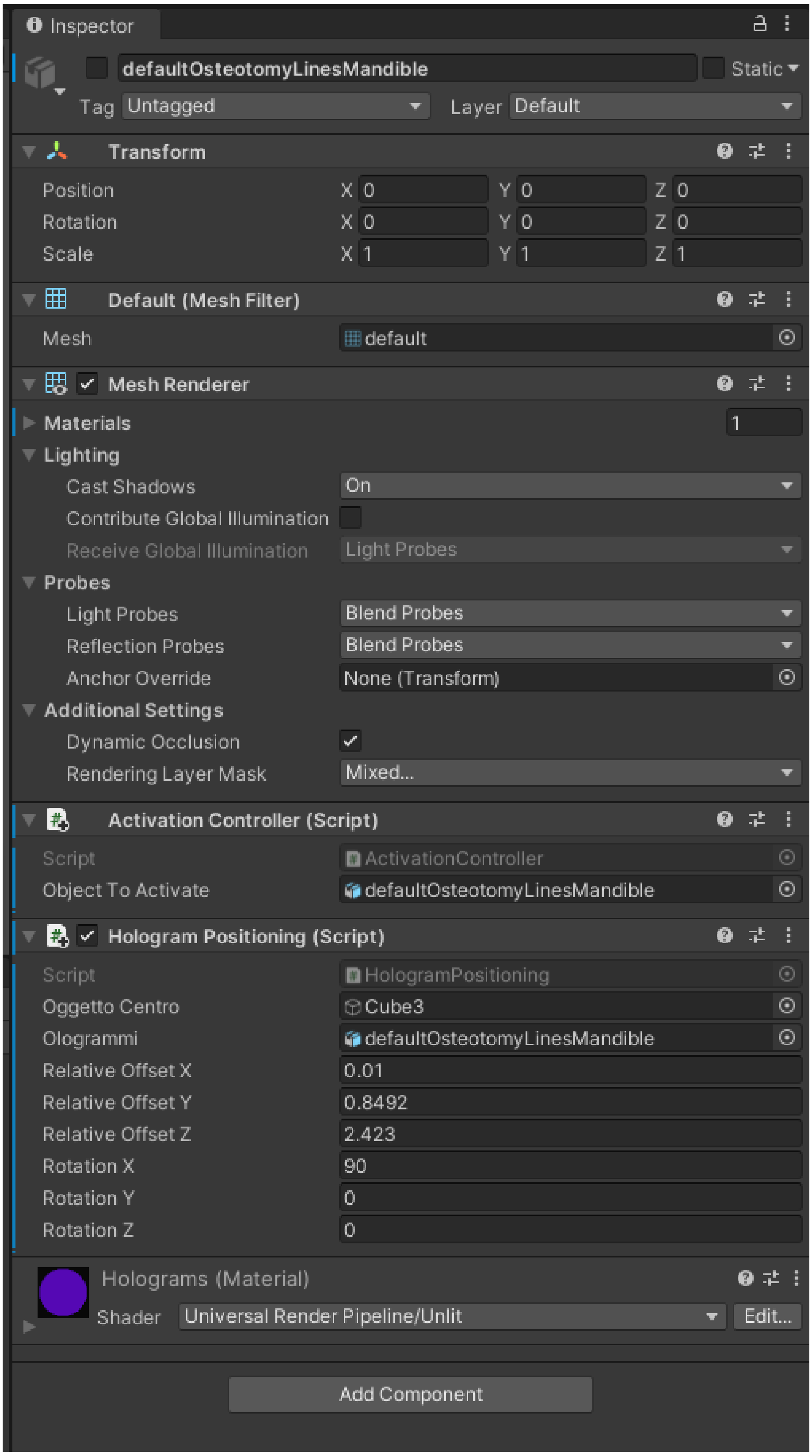
Rotation Z 0

Activation Controller (Script)

Script ActivationController

Object To Activate DeltaPanelOut

Figure 72 DeltaPanelOut Inspector.



Inspector

□ defaultOsteotomyLinesMandible □ Static

Tag Untagged Layer Default

Transform

Position X 0 Y 0 Z 0

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

Default (Mesh Filter)

Mesh default

☑ Mesh Renderer

Materials 1

Lighting

Cast Shadows On

Contribute Global Illumination □

Receive Global Illumination Light Probes

Probes

Light Probes Blend Probes

Reflection Probes Blend Probes

Anchor Override None (Transform)

Additional Settings

Dynamic Occlusion ☑

Rendering Layer Mask Mixed...

Activation Controller (Script)

Script ActivationController

Object To Activate defaultOsteotomyLinesMandible

☑ Hologram Positioning (Script)

Script HologramPositioning

Oggetto Centro Cube3

Ologrammi defaultOsteotomyLinesMandible

Relative Offset X 0.01

Relative Offset Y 0.8492

Relative Offset Z 2.423

Rotation X 90

Rotation Y 0

Rotation Z 0

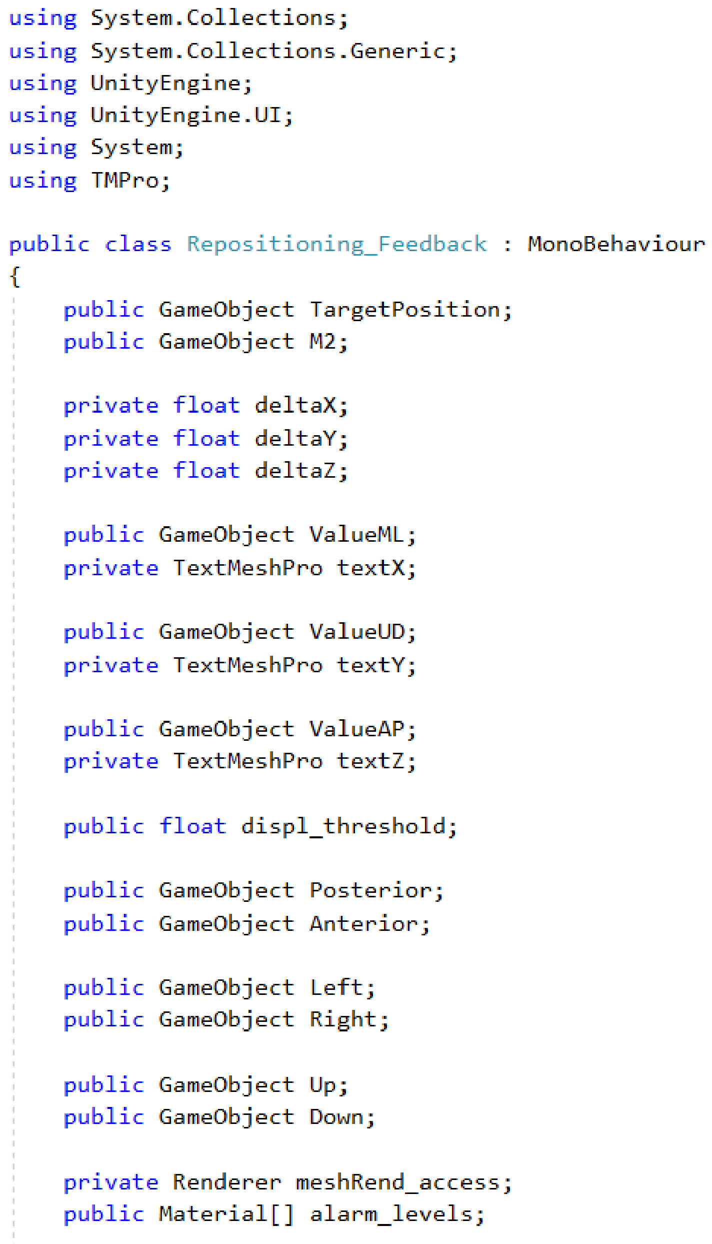
Holograms (Material)

Shader Universal Render Pipeline/Unlit

Figure 73 Mandible Osteotomy Lines Inspector.

The fifth script is Repositioning Feedback, it was implemented in the previous work. [1] Thanks to this script it is possible to manage all the feedback for the repositioning. The main functions to be executed through the script are the calculation of the delta between TargetPosition cube hologram and the current position of the maxilla marker; the management of the cube hologram color based on delta values; the management of 3DArrows based on delta values; the display of delta values in the dedicated dialog panel.

At the beginning of the script a list of public variables, that will be then specified in the Inspector, are defined. Those variables are reported in Figure 74.

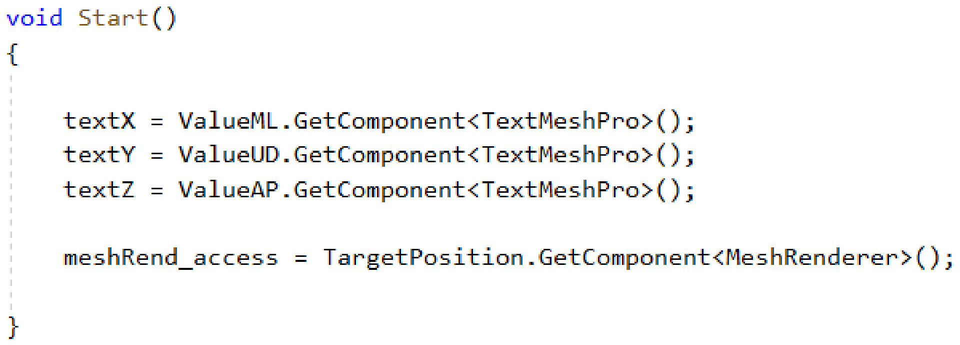


```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;
using TMPro;

public class Repositioning_Feedback : MonoBehaviour
{
    public GameObject TargetPosition;
    public GameObject M2;

    private float deltaX;
    private float deltaY;
    private float deltaZ;

    public GameObject ValueML;
    private TextMeshPro textX;

    public GameObject ValueUD;
    private TextMeshPro textY;

    public GameObject ValueAP;
    private TextMeshPro textZ;

    public float displ_threshold;

    public GameObject Posterior;
    public GameObject Anterior;

    public GameObject Left;
    public GameObject Right;

    public GameObject Up;
    public GameObject Down;

    private Renderer meshRend_access;
    public Material[] alarm_levels;
```

Figure 74 Variables of RepositioningFeedback script.

In the Start section, the variables textX, textY, and textZ are initialzed. These variables will store references to TextMeshPro components got from GameObjects named

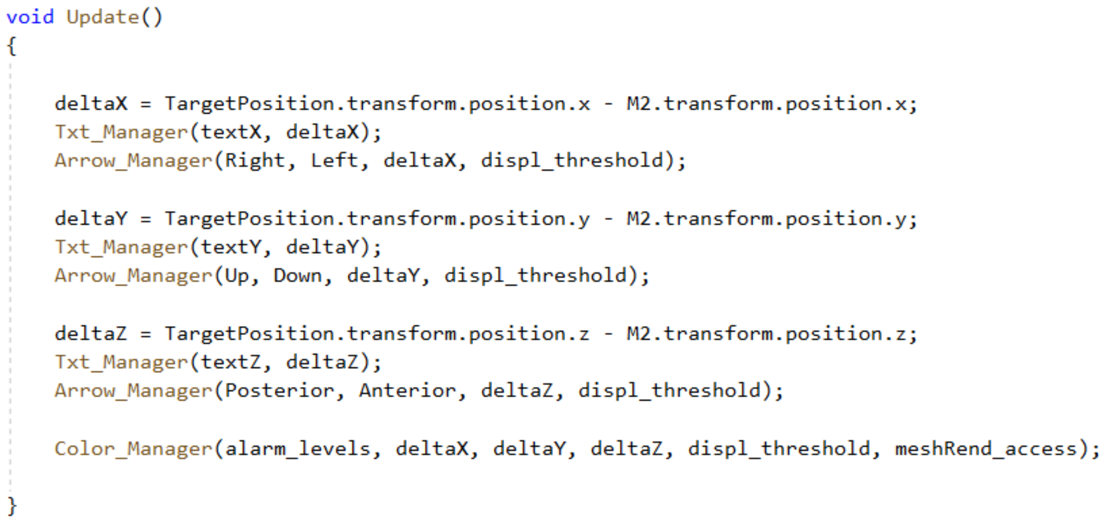
ValueML, ValueUD, and ValueAP. Instead, meshRend_access stores the reference of TargetPosition MeshRenderer component. Figure 75.



```
void Start()
{

    textX = ValueML.GetComponent<TextMeshPro>();
    textY = ValueUD.GetComponent<TextMeshPro>();
    textZ = ValueAP.GetComponent<TextMeshPro>();

    meshRend_access = TargetPosition.GetComponent<MeshRenderer>();

}
```

Figure 75 Start of RepositioningFeedback script.

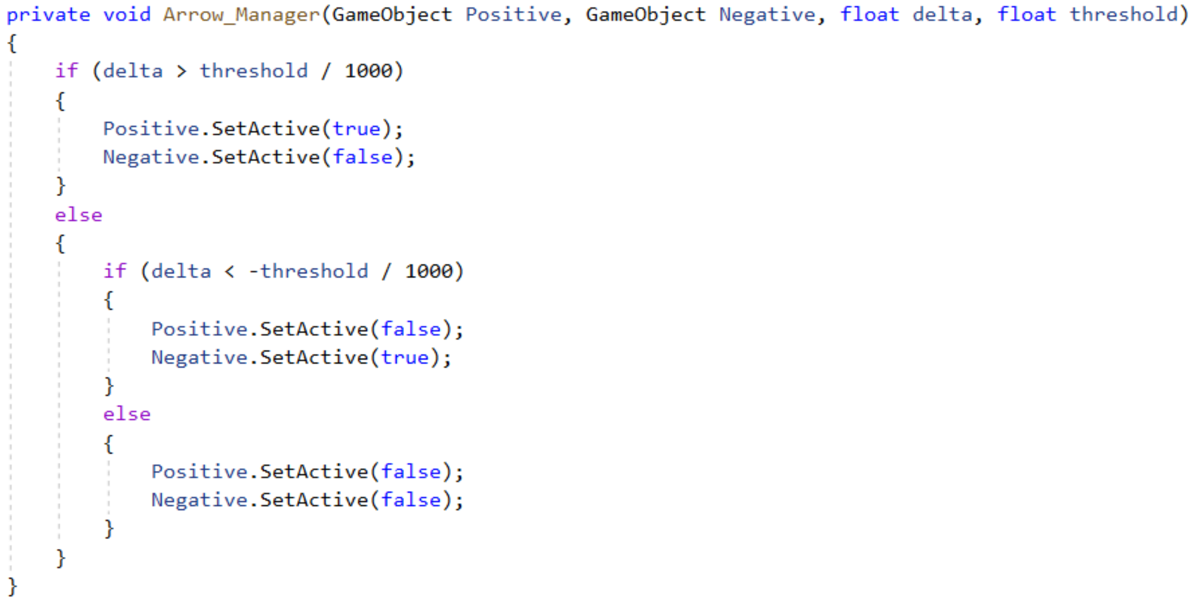
Within the Update block, to implement deltas calculation, the difference between ‘position.x’/’position.y’/position.z’ field of ‘Transform’ of cube hologram and, respectively, ‘position.x’/’position.y’/position.z’ field of ‘Transform’ component of M2 are assigned to 3 private float variables (‘deltaX’, ‘deltaY’ and deltaZ’). Then, these deltas will be employed to manage the various feedback elements. Figure 76.



```
void Update()
{

    deltaX = TargetPosition.transform.position.x - M2.transform.position.x;
    Txt_Manager(textX, deltaX);
    Arrow_Manager(Right, Left, deltaX, displ_threshold);

    deltaY = TargetPosition.transform.position.y - M2.transform.position.y;
    Txt_Manager(textY, deltaY);
    Arrow_Manager(Up, Down, deltaY, displ_threshold);

    deltaZ = TargetPosition.transform.position.z - M2.transform.position.z;
    Txt_Manager(textZ, deltaZ);
    Arrow_Manager(Posterior, Anterior, deltaZ, displ_threshold);

    Color_Manager(alarm_levels, deltaX, deltaY, deltaZ, displ_threshold, meshRend_access);

}
```

Figure 76 Update of RepositioningFeedback script.

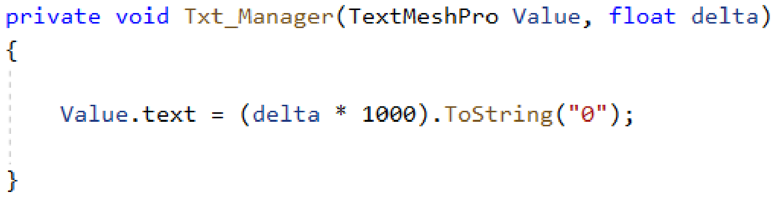
The private method “ArrowManager” manages the 3D-arrows based on delta values, it receives as inputs alternatively one of the delta values and the correspondent two 3D arrows (couples: ‘deltaX’ - ‘Right’, ’Left’; ‘deltaY’- ‘’Posterior’, ‘Anterior’; ‘deltaZ’ - ‘Up’, ’Down’). The method, through multiple if-then statements on the delta with respect to the displacement threshold and by mean of the ‘SetActive’ function, imposes as active the positive arrow when the delta is higher than the threshold or imposes as active the negative arrow when the delta is lower than minus the threshold or finally imposes as inactive both negative and positive arrows when delta is within the tolerance range. Since Unity works with meters, the conversion from meter to millimeters is implemented within the if statements. Figure 77.



```
private void Arrow_Manager(GameObject Positive, GameObject Negative, float delta, float threshold)
{
    if (delta > threshold / 1000)
    {
        Positive.SetActive(true);
        Negative.SetActive(false);
    }
    else
    {
        if (delta < -threshold / 1000)
        {
            Positive.SetActive(false);
            Negative.SetActive(true);
        }
        else
        {
            Positive.SetActive(false);
            Negative.SetActive(false);
        }
    }
}
```

Figure 77 Lines of code of Arrow_Manager private method.

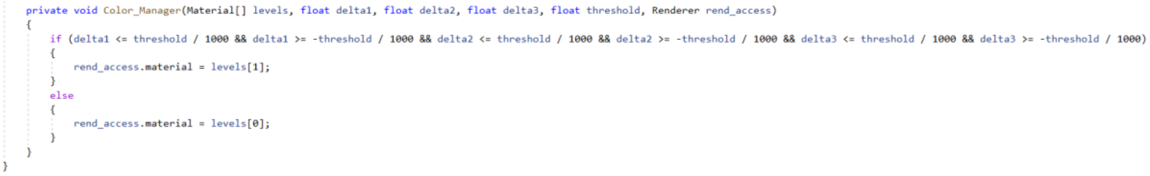
Then, in the Update block, the ‘Txt_Manager’ private method is called 3 times. This private method receives as inputs alternatively one of the delta values and the private ‘TextMeshPro’ variable with the access to the ‘TextMeshPro’ component of the correspondent box. Finally, within the private method, the ‘Text’ component of the input ‘TextMeshPro’ private variable is assigned as the delta value, converted from meters to millimetres and approximated at the first non-decimal value. Figure 78.



```
private void Txt_Manager(TextMeshPro Value, float delta)
{

    Value.text = (delta * 1000).ToString("0");

}
```

Figure 78 Lines of code of Txt_Manager private method.

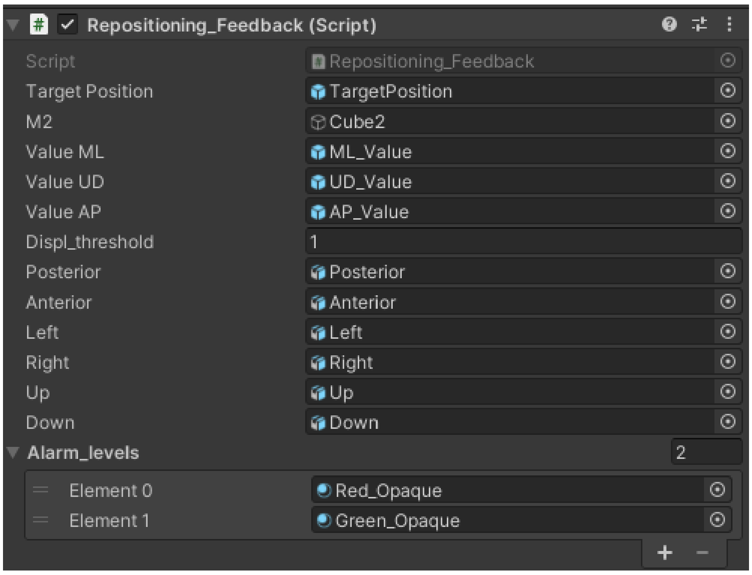
The private method ‘Color_Manager’ performed within the Update receives as input the list of materials, the 3 just calculated deltas, the displacement threshold, and a private variable of the ‘Renderer’ type that by the ‘GetComponent’ function in the Start block allow to get access to the ‘Material’ component of the cube hologram game object. The private method then with an if-else statements allows to assign top the ‘Material’ component of the cube hologram game object different materials based on the actual values of deltas with respect to the imposed thresholds. Essentially, the cube hologram material will be assigned to the first element of the material list (red) when all the deltas are outside the tolerance range, or to the second element of the material list (green) when all the deltas are within the tolerance range. Even here it must be noticed that deltas are extracted by Unity in meters while the threshold is expressed in millimeters, thus, the delta conversion from meters to millimeters is implemented within the if statements. Figure 79.



```
    private void Color_Manager(Material[] levels, float delta1, float delta2, float delta3, float threshold, Renderer rend_access)
    {
        if (delta1 <= threshold / 1000 && delta1 >= -threshold / 1000 && delta2 <= threshold / 1000 && delta2 >= -threshold / 1000 && delta3 <= threshold / 1000 && delta3 >= -threshold / 1000)
        {
            rend_access.material = levels[1];
        }
        else
        {
            rend_access.material = levels[0];
        }
    }
}
```

Figure 79 Lines of code of Color_Manager private method.

This script is assigned to M2 and the Inspector with the specified variables in the script’s public fields is reported in Figure 80.



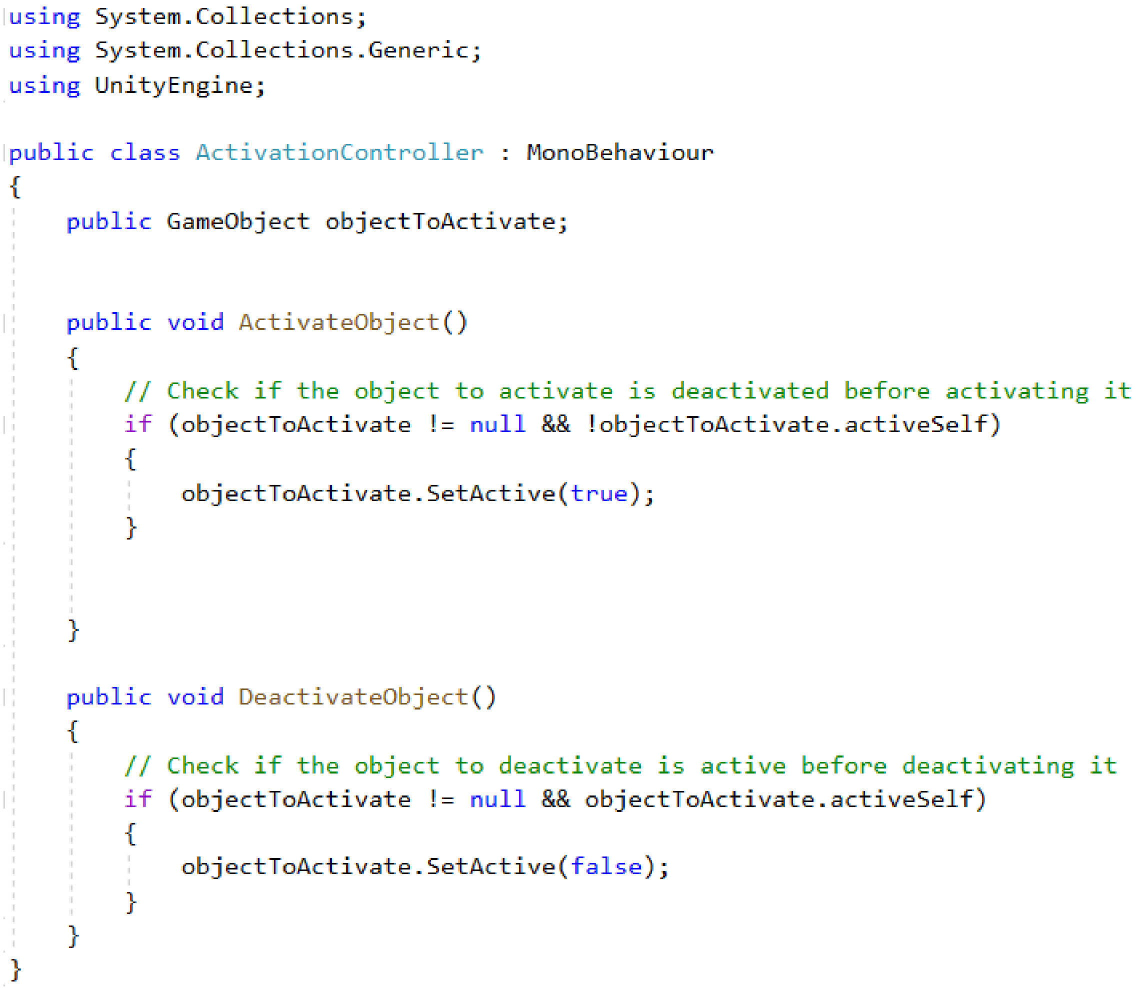
Repositioning_Feedback (Script)

Script	Repositioning_Feedback
Target Position	TargetPosition
M2	Cube2
Value ML	ML_Value
Value UD	UD_Value
Value AP	AP_Value
Displ_threshold	1
Posterior	Posterior
Anterior	Anterior
Left	Left
Right	Right
Up	Up
Down	Down
Alarm_levels	2
Element 0	Red_Opaque
Element 1	Green_Opaque

Figure 80 Input assignment of Repositioning_Feedback script in M2 Inspector.

The sixth script is “ActivationController” that controls the activation and deactivation of a GameObject. A public variable, “objectToActivate”, that represents the GameObject to activate or deactivate is declared at the beginning. Then two methods are defined. The “ActivateObject” method checks if the GameObject is not null and if it is not active, in this case it activates the GameObject. The other method, “DeactivateObject”, checks if the GameObject is not null and if it is active, in this case it deactivates the GameObject. Figure 81.

This script is assigned to the objects that we want to activate or deactivate by means of the buttons on the left-hand panel, they are Holograms, 3DArrows, TargetPosition, and DeltaPanelOut.



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ActivationController : MonoBehaviour
{
    public GameObject objectToActivate;


    public void ActivateObject()
    {
        // Check if the object to activate is deactivated before activating it
        if (objectToActivate != null && !objectToActivate.activeSelf)
        {
            objectToActivate.SetActive(true);
        }


    }

    public void DeactivateObject()
    {
        // Check if the object to deactivate is active before deactivating it
        if (objectToActivate != null && objectToActivate.activeSelf)
        {
            objectToActivate.SetActive(false);
        }
    }
}
```

Figure 81 ActivationController script.

5. Case Study

The developed MR application is designed to assist maxillofacial surgeons in performing osteotomies on both the maxilla and mandible, as well as repositioning the maxilla. The application's effectiveness is evaluated using a 3D printed skull that mimics a real patient's head. The 3D model is designed in Rhinoceros, it includes the skull geometry, splints, cutting lines, drilled holes, and the three markers. The osteotomy lines are designed as cylinders of 0.7 mm diameter, customized on the skull model, while drilled holes are developed as cylinders of 1.7 mm diameter. They are both exported in *.OBJ format and loaded into Unity as they are part of the virtual content to be displayed. Furthermore, osteotomy lines have been used to split the model in the various assembly components, the cranium, the maxilla, and the mandible. The three markers have been designed as three cubes measuring 32.5x32.5x32.5 mm, and a texture that corresponds to a VarjoMarker has been assigned to each face. The 3D model has been exported in *.VRML to be 3D printed using the ZPrinter450, a 3D printer that employs CJP technology. At this point the MR application described in section 4.1 has been tested on the printed phantom. This printed skull enables the execution of the three tasks expected from the application and consists of three parts: the cranium, the maxilla, and the mandible, which can be easily assembled. As the application relies on marker-based tracking, also the markers need to be anchored to the phantom. Actually, the forehead marker (M1) and the maxilla marker (M2) are printed separately and then assembled, while the last marker (M3) is directly printed with the mandible. When the user intends to perform the maxillary osteotomy, the cranium, the maxilla and M2, the marker required for displaying cutting lines and drilled holes, are assembled. Whereas, when the user selects mandibular osteotomies, only one part of the skull is needed, the mandible with the integrated marker.

Finally, if the user selects the maxillary repositioning, M1 and M2 are appropriately anchored, respectively on the cranium and on the maxilla. Meanwhile, the cranium and mandible are disassembled. This allows the user to freely manipulate the maxilla fragment while keeping the cranium fixed when the application displays the target position along with feedback holograms. Target position is reached moving the maxilla fragment 20 mm leftward and 20 mm forward from the skull perspective. The application is tested over the 3D printed phantom multiple times asking to the users to begin by performing the maxillary osteotomy, followed by the mandibular osteotomy. Between one modality and another, the user is asked to fill out a questionnaire, the UEQ. While, at the end of the test the user is required to complete the UX questionnaire. The maxillary osteotomy test is performed by tracing the cutting lines with a pen, following those displayed on the phantom, and making two circles where the drilled holes appear. Then, the tested geometry is scanned to proceed with the result extraction. The same procedure is repeated for the mandibular osteotomy. The 3D scans are conducted using the Go!Scan3D scanner, while the results are extracted through the CloudCompare software.

This entire procedure is carried out to test the MR application developed for Varjo XR3 and is repeated for each user. In order to enable a comparison between this application and the AR application for HoloLens2, the same users repeated the same tests with the AR application while wearing the HoloLens2.

5.1 Modelling

As previously mentioned, the 3D model was created in Rhinoceros. Starting with a 3D skull downloaded from Sketchfab, which had already been scaled and lightened by trimming it behind the condyles and removing the internal volume. This 3D model included the splints and bridge structures with holes for marker anchoring. The initial outlook of the 3D model is reported in Figure 82. The initial 3D model from which the modelling of the markers and mandible began already contained the cutting lines for maxillary and mandibular osteotomy and the drilled holes for fixation after repositioning. Using this 3D model as a starting point, the textured M1 and M2 markers were modelled, and modifications were made to the mandible to create a unified structure with M3.

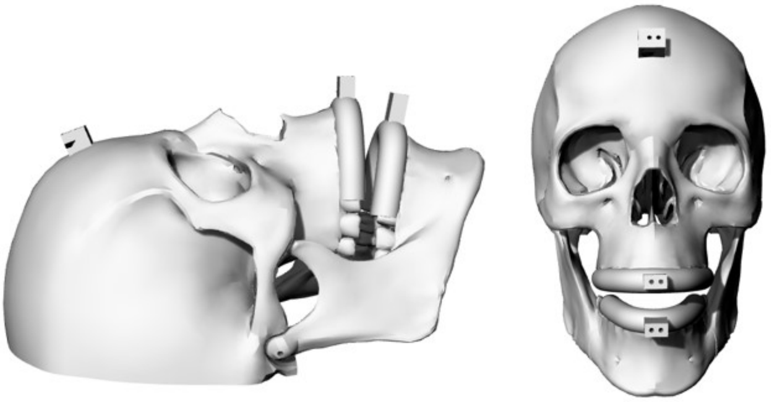


Figure 82 Initial CAD model.

5.1.1 M1 and M2

To construct the two markers, the first step is the extraction of the control points from the imported mesh using the ‘ExtractPoint’ command. Only the points that form the circumferences of the two holes on the bridge structure are retained. Using the ‘Circle: 3 Points’ command, the two circumferences are reconstructed. By activating the ‘Cen’ snap, a point at the center of each circumference is placed. Next, using ‘Single Line’, a line connecting the two centers is created, and using the ‘Mid’ snap, a point at the midpoint of this line is positioned. Then, a square surface measuring 32.5x32.5 mm is created, which is centered at the midpoint between the holes using the ‘Move’ command. At this point, the square surface was extruded by 32.5 mm, using the ‘ExtrudeSrf’ command with the ‘Solid: Yes’ option enabled. To create holes in the marker corresponding to the holes on the bridge structures, two cylinders were created from the centers of the circumferences using the ‘Cylinder’ command, with a diameter of 2 mm and a height of 10 mm. Once the cube structure and the two cylinders were obtained, these three elements were meshed using the ‘Mesh’ command, and then the ‘MeshBooleanDifference’ command was applied to obtain the marker with cylindrical holes. This procedure is followed for both M1 and M2.

The next step involves assigning a texture to each face of the markers. These textures correspond to the VarjoMarkers assigned within Unity to the faces of the virtual markers. To do this, the first step is to explode the faces of the newly created cubes using the “Explode” command so that a different texture can be assigned to each face. Otherwise, a single texture would be assigned to the entire structure. First, the textures need to be loaded by going to the materials panel and clicking the ‘+’ button to choose images from the corresponding folder, selecting ‘Type’ as ‘Picture.’ Then, select one of the cube faces, navigate to the “Properties” panel, and choose the ‘Texture Mapping’ tab. Here, set the ‘Type’ to ‘Box,’ select the center of the surface, adjust the image rotation as needed, and specify the image dimensions to match those of the surface. Now, assign one of the imported textures to the corresponding face and repeat the same process for all the faces of markers M1 and M2, excluding the ones with holes. In Figure 83 the sull with the two textured marker, M1 and M2 are reported.

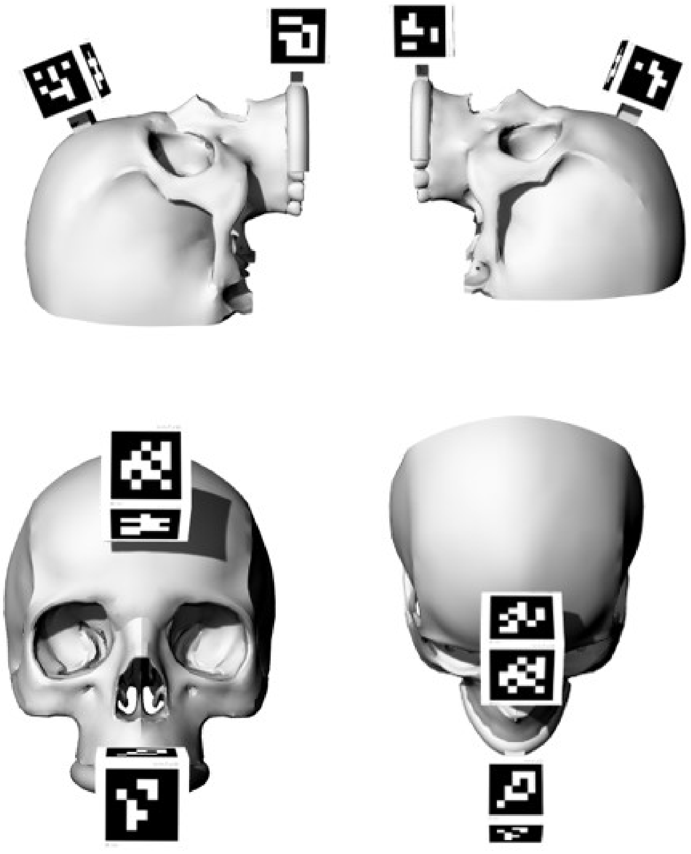


Figure 83 3D model of cranium and maxilla with M1 and M2.

5.1.2 Mandible with M3

The initial 3D model had the mandible divided into three parts in accordance with the previously modelled cutting lines. In this case, a single-piece mandible was desired, and to achieve this result, the 'Join' command was used. Now, instead of having the bridge structure on the splint for marker anchoring, it was preferred to create a single structure that included both the mandible and marker M3. To achieve this, the previous bridge structure on the mandible was cut. A plane was created using the 'Surface from 3 or 4 corner points' command, and then the mandible was separated from the bridge structure using the 'Mesh Split' command. Next, a new parallelepiped was created to form the new bridge structure and was joined to the mandible using 'MeshBooleanUnion.' Now, a cube with dimensions of 32.5x32.5x32.5 is created, which will serve as the new M3 marker. To reduce the weight of M3 and prevent it from potentially breaking the bridge structure due to its weight once printed, it was hollowed out using the 'Shell' command with a thickness of 4mm, and then meshed. Finally, the cube was united with the rest of the mandible using the ‘MeshBooleanUnion’ command. Using the ‘Shell’ command involves removing one face of the cube. The decision was made to remove the bottom face, as it would never be visible through the headset when the application is launched. Now, following the same procedure described in the previous section, textures have been assigned to the faces of marker M3, excluding the one that features the structure connecting it to the mandible and the bottom one due to the ‘Shell’ command. The final outlook of the mandible and the M3 is reported in Figure 84.

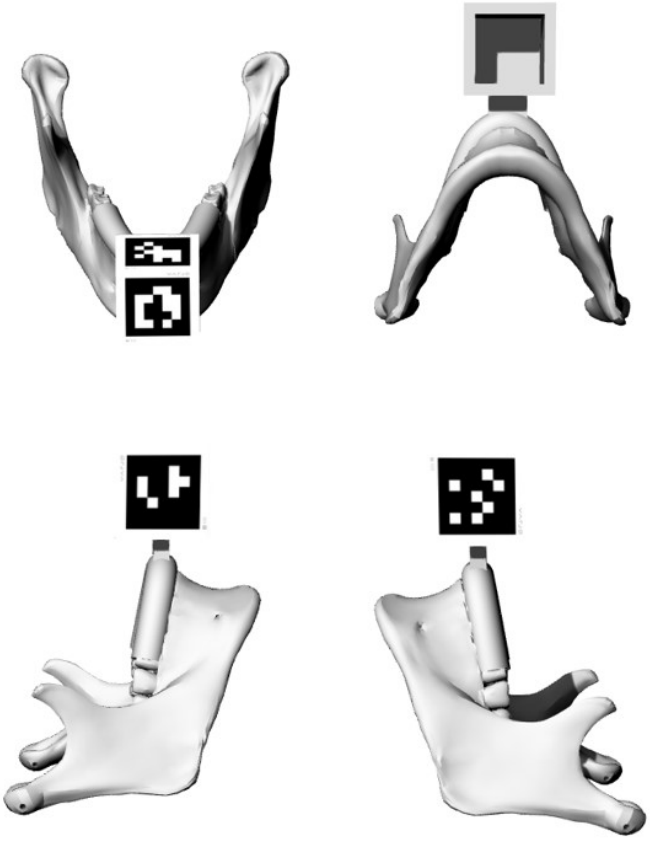


Figure 84 3D model of the mandible and M3.

5.2 3D Printed Components

The developed model, as anticipated, is exported in *.VRML to keep texture information and 3D printed using the ZPrinter450. The core material used for the printing of all the model components is a high-performance mixture powder containing calcium sulphate hemihydrate combined with a binder, a solution dispensed through the print head and applied to the powder. After the printing, a drying cycle occurs to add strength to the parts followed by a cleaning process, to remove all the un-sintered powder. In the post-processing phase, an infiltrant is used to permeate the printed powder and bond it together to give more durability. Since the VarjoMarkers directly printed on the faces of the cubes through 3D printing were not well-tracked by the device, the VarjoMarkers were printed on paper and then adhered to the corresponding faces. The skull assembly components are reported in Figure 85 and Figure 86.

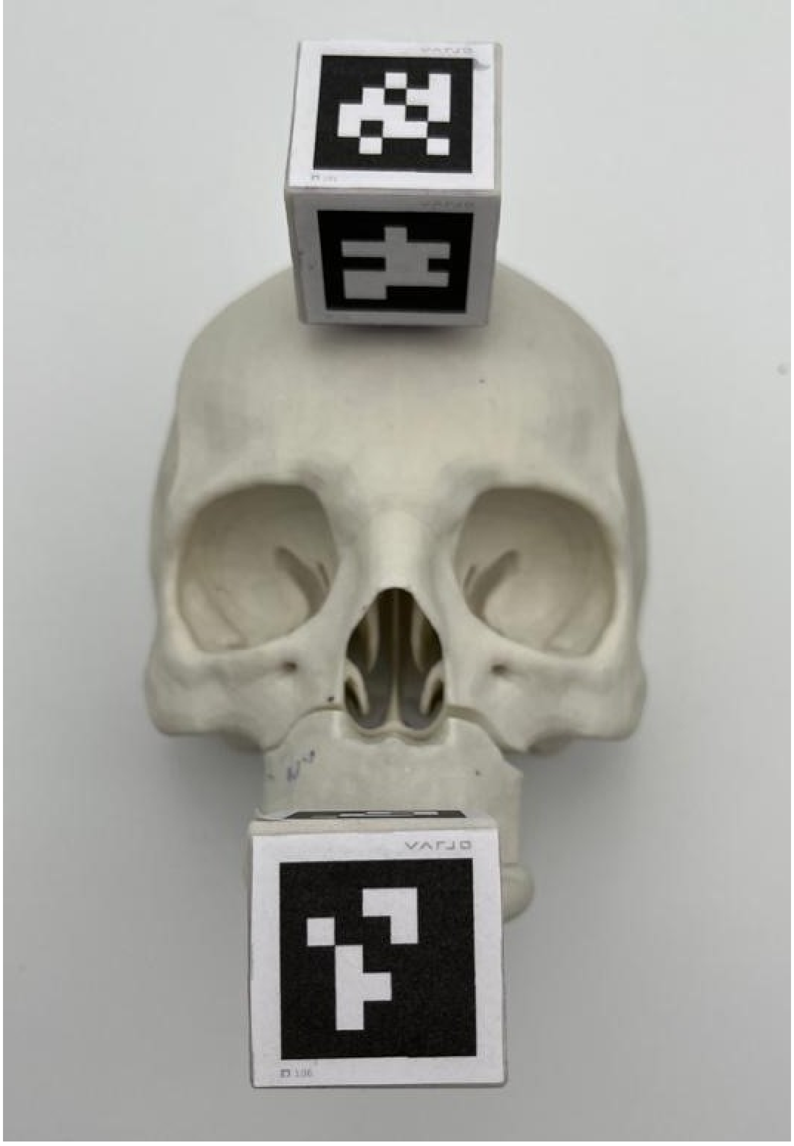


Figure 85 Printed model: assembled cranium and maxilla with M1 and M2 anchored.

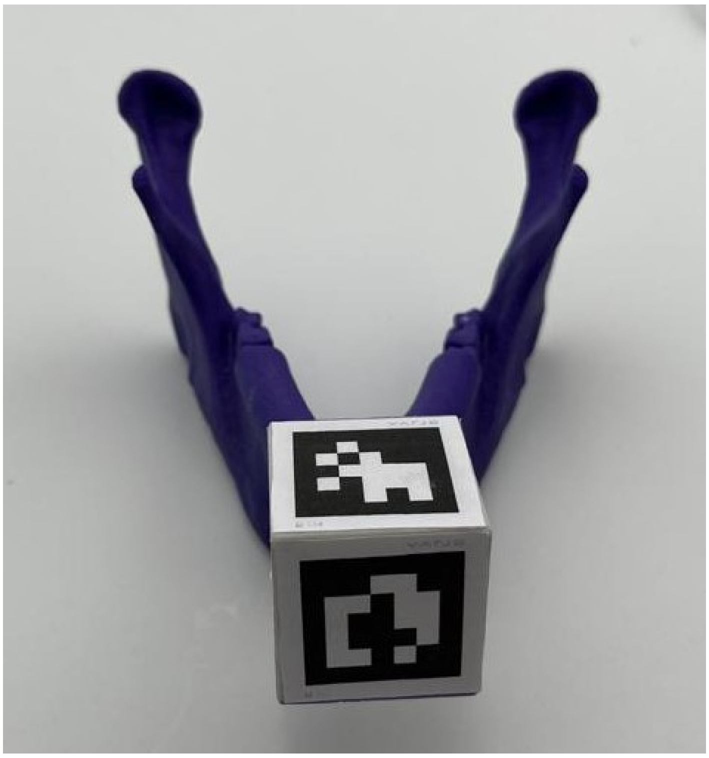


Figure 86 Printed model: mandible and M3.

5.3 Test

Once the application is launched on Varjo XR-3, clicking Play in Unity, and the phantom has been 3D-printed, the test phase can start. The application has been tested by 7 users. The first user (U01) and the fifth user (U05) are two researchers with familiarity with MR technology, the second user (U02) and the third user (U03) are two researchers with no familiarity with MR technology, the fourth user (U04) is a medical doctor with no familiarity with MR technology, the sixth user (U06) is a student just introduced on the MR technology and, the seventh user (U07) is a university professor with some familiarity with MR technology.

The entire process followed during the testing phase is reported in Figure 87.

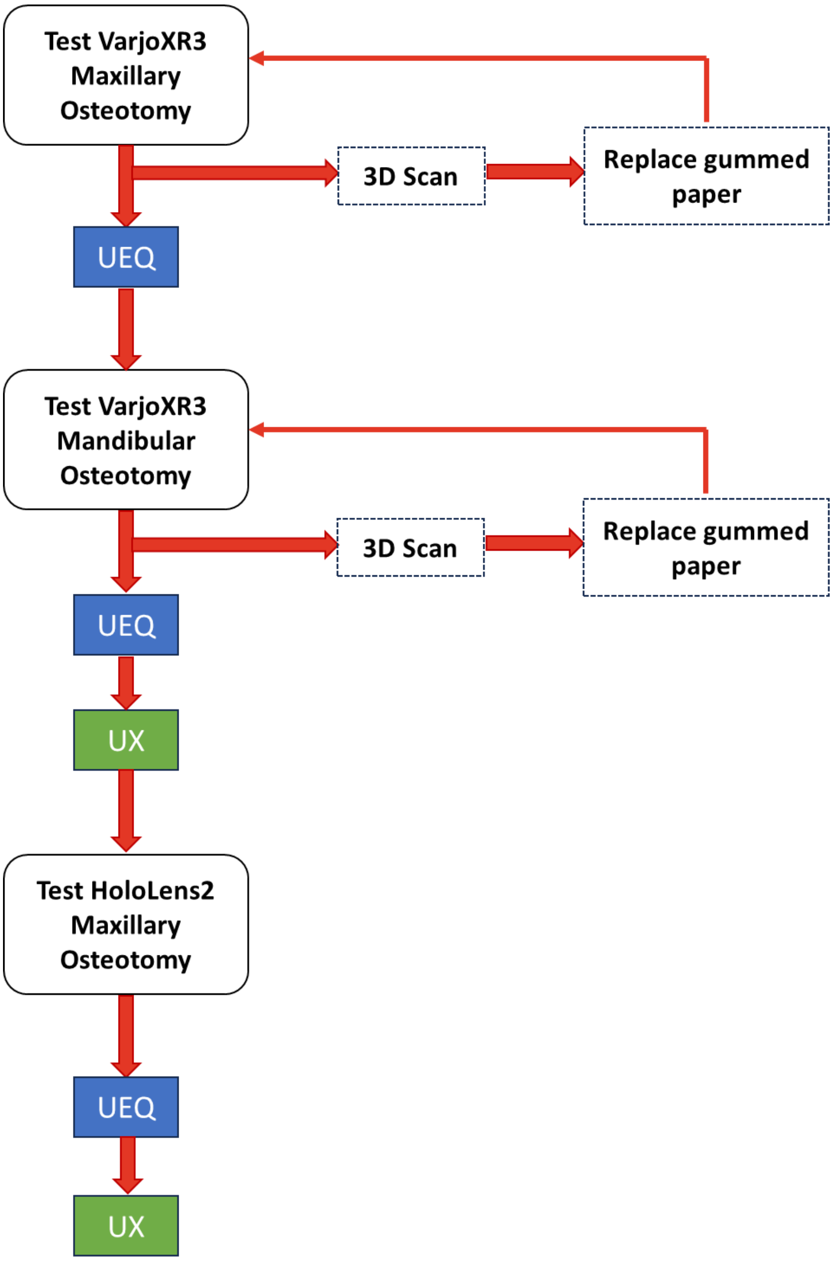

Test VarjoXR3 Maxillary Osteotomy
3D Scan
Replace gummed paper
UEQ
Test VarjoXR3 Mandibular Osteotomy
3D Scan
Replace gummed paper
UEQ
UX
Test HoloLens2 Maxillary Osteotomy
UEQ
UX

Figure 87 Flowchart followed during the testing phase.

The first task aims to perform maxillary osteotomy. Therefore, the skull and maxilla were assembled together, and their respective markers were anchored. Gummed paper tape was applied to the assembly interface. Then, the user, wearing the Varjo XR3, initiated the maxillary osteotomy by framing his hands, flipping the left palm, and selecting the "Maxilla Osteotomy" button from the displayed menu attached to the left hand. Then, by looking at marker M2, osteotomy lines and drilled holes appear in position. Now the user must trace the lines and holes on the skull with a pen, following the ones displayed. Figure 88.

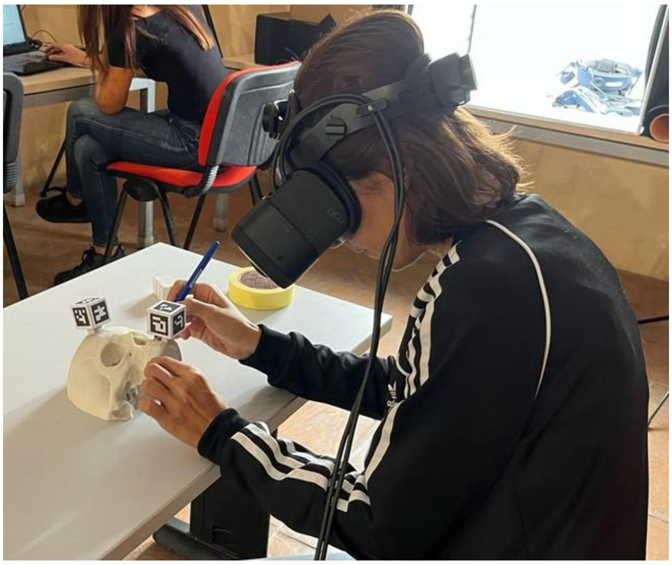


Figure 88 Maxillary osteotomy task.

Second task, instead, aims to perform mandibular osteotomy. In this case, only the mandible with its marker is required. The user, once again looking at his hands and flipping the left hand, must select the "Mandibular Osteotomy" button. Subsequently, by focusing on marker M3, the mandibular osteotomy cut lines positioned on the anatomy are visualized. Following these, he must trace the cutting lines on the skull with a pen. Figure 89.

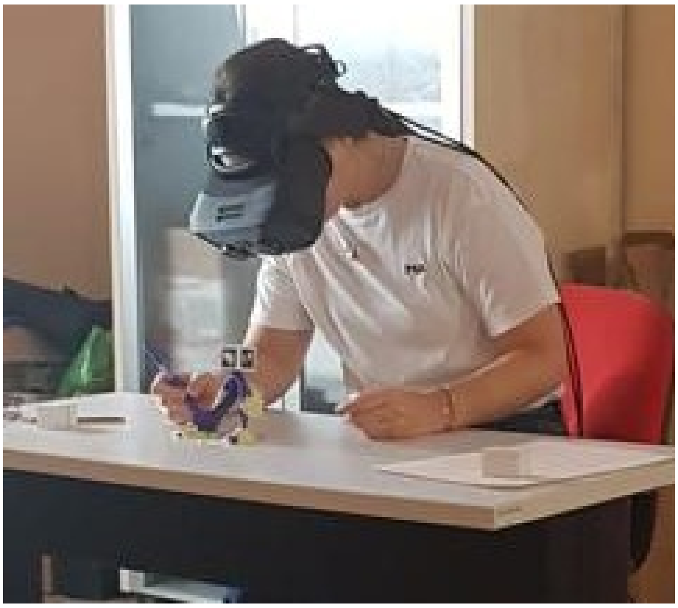


Figure 89 Mandibular osteotomy task.

Then, each user was asked to wear the HoloLens 2 and proceed with the ocular calibration by following the steps 'Menu → Impostazioni → Sistema → Calibrazione → Esegui Calibrazione Oculare' directly on the device. At this point, the maxillary osteotomy was tested without tracing the lines, simulating only the movement in order to make a qualitative comparison with the same mode tested with the Varjo XR3. Each user performed one trial for each test.

5.4 Quantitative Validation

After the testing phase, both quantitative and qualitative validation was carried out. The main steps are summarized in the Figure 90.

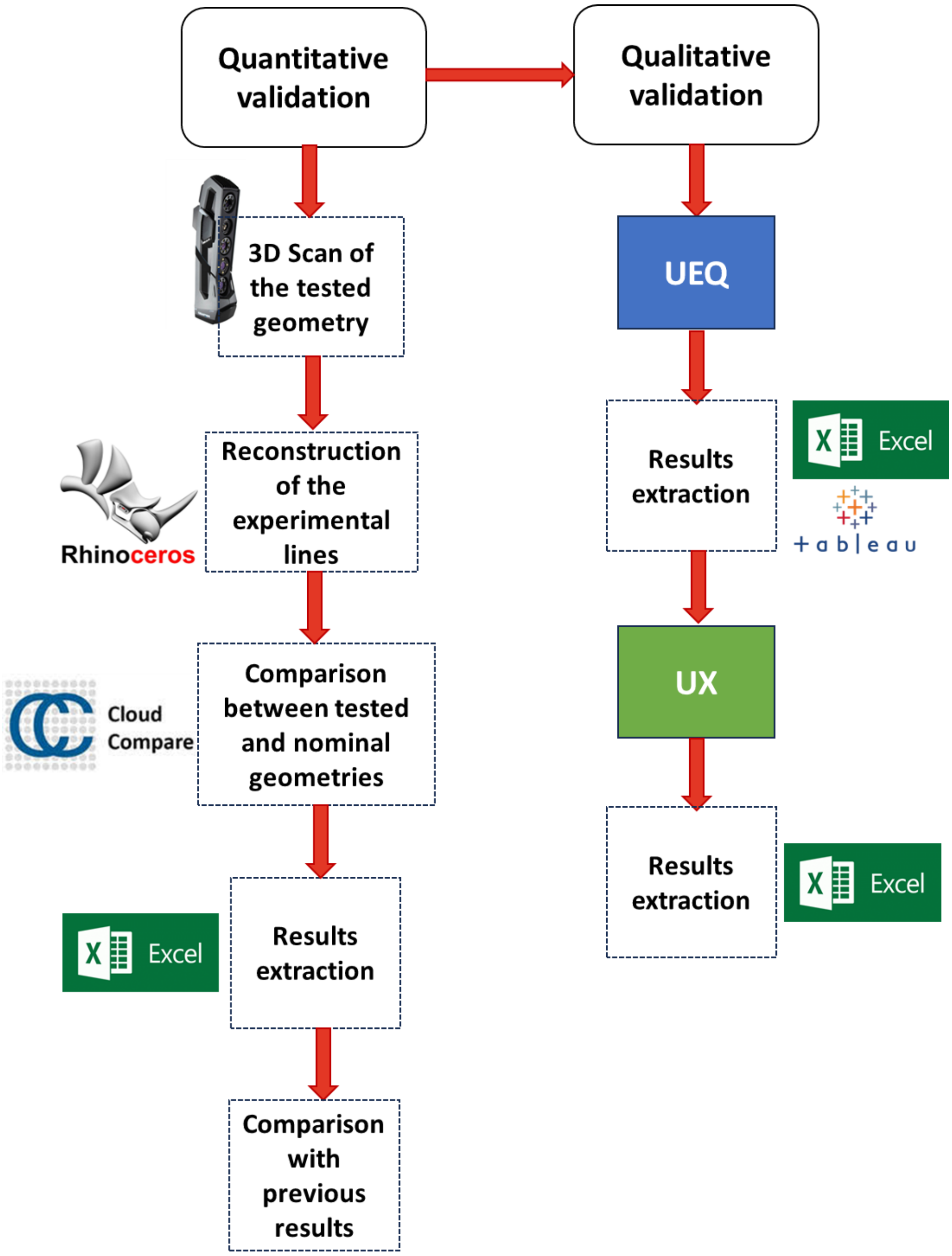


Figure 90 Flowchart of the validation process

To assess the accuracy of the application regarding osteotomy tasks, the curves drawn by each user were compared to the nominal curves to quantify fluctuations and standard deviations.

The quantitative evaluation was possible by scanning the tested phantom, which allows obtaining a textured CAD file from which it is possible to reconstruct the cutting lines for mandibular and maxillary osteotomies and the holes for fixation. The tested geometries were reconstructed in Rhinoceros 7, while the deviations were extracted using the CloudCompare software.

Regarding the scanning phase, the Go!Scan3D device is used. The device first needs to be calibrated, and the reference plane needs to be established. The reference plane is a white board on which circular markers are specifically attached. Then, the cranium-maxilla complex and subsequently the mandible, after being tested, are positioned on the reference plane. Before starting the scanning session, the parameter 'Metodo di Posizionamento' must be set to 'Target/Geometria' and the 'Resolution' parameter must be set to '0.50 mm' on the scanner dedicated software. Then, the Go!Scan3D device is handled and moved around the tested geometries to acquire their geometry from multiple views. If the scanner is at the correct distance from the object, it provides feedback with a green light; otherwise, the feedback is a red light. Simultaneously on the PC display, using the scanner dedicated software, appears the scanned object with its texture. The acquired scan is then processed to remove noise and unnecessary points and finally it is exported in the *.VRML format to keep texture information. Then to proceed with the reconstruction, the *.VRML file is imported in Rhinoceros 7. Figure 91 and Figure 92. Here, to reconstruct cutting lines a cloud of points is created through the 'Multiple Points' command and using the 'Vertex' snap the points are positioned on the mesh vertexes of the visible experimental cutting lines, an interpolating line is created through the 'Polyline Through Points' command and it is rebuilt with the 'Rebuild Curve Non-Uniform', setting the tolerance to 0.1. Rebuilt lines are piped with the 'Pipe' command by setting the radius to 0.375 mm and finally meshed with the 'Mesh' command. To reconstruct drilled holes, points are created through the 'Multiple Points' command, using the 'Vertex' snap to position the points on the vertexes of the mesh where visible experimental holes are traced. Even if the users have traced the outline of each hole, to compare the results with the results obtained in the previous work [1], the center of the traced holes is reconstructed. To do this, the command 'Circle: fit Points' is used to obtain a circumference that approximates the traced hole, in this manner it is possible to find the center by placing a Point with the ‘Cen’ snap activated. Then, a sphere centered on this point and with a radius of 0.25 mm is generated with the ‘Sphere: Center Radius’ command and finally meshed with the ‘Mesh’ command. The same procedure is repeated for each hole. Figure 93 and Figure 94. Also the nominal drilled holes center need to be reconstructed in Rhinoceros to perform the comparison. The frontal and back circumferences of the designed cylinder are reconstructed using the ‘Circle: 3 Points’ command and then at the center of both a point is placed using the ‘Center’ snap and ‘Multiple Points’ commands. These two centers have been used to reconstruct the cylinder axis through the ‘Single Line’ command. Then, the ‘MeshToNURBS’ command has been used to convert the maxilla fragment from mesh to NURBS. This was necessary to place a point at the intersection of the cylinder axis with the maxilla, using the ‘Intersect’ command to determine the intersection point. Finally, the ‘Sphere: Center Radius’ has been used to create on the intersection point a sphere with a radius of 0.25 mm that is finally meshed with the ‘Mesh’ command. The same procedure is repeated for each drilled hole.

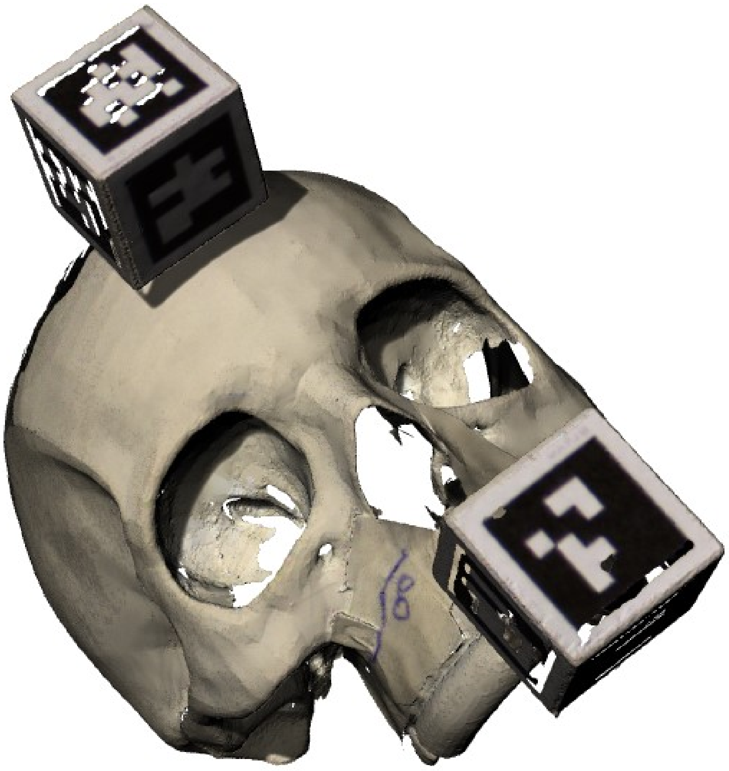


Figure 91 Scanned cranium and maxilla with drown cutting lines and drilled holes.

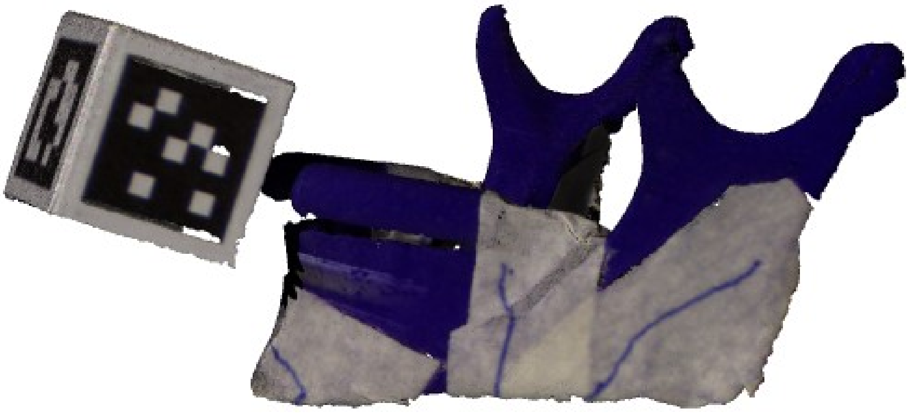


Figure 92 Scanned mandible with drawn osteotomy lines.

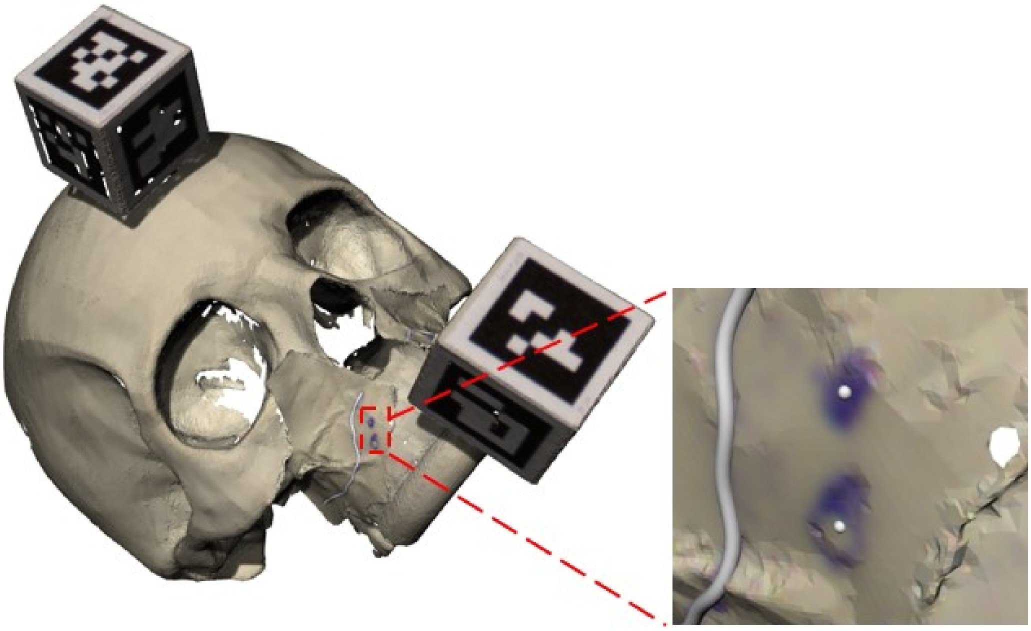


Figure 93 Reconstructed cutting lines and zoom on the spheres positioned at the center of the traced drilled holes for Maxillary Osteotomy.

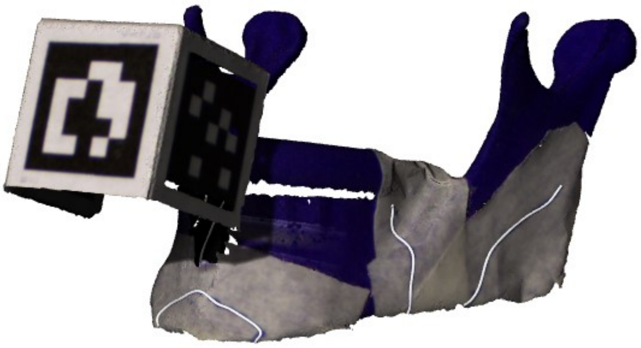


Figure 94 Reconstructed cutting lines for Mandibular Osteotomy.

Experimental cutting lines and drilled holes and tested geometry are separately exported in *.STL format to be then imported in CloudCompare.

All the *.STL files of the nominal and experimental geometries have been imported in CloudCompare. To distinguish nominal geometries from experimental ones, following the path, 'Edit → Colors → Set Unique' , yellow color has been assigned to nominal geometries, and red color to experimental geometries. Figure 95. To compare nominal and experimental geometries, they must be aligned. This can be achieved selecting the geometries that must be aligned and following the path 'Align Two Clouds by Picking Equivalent Points'. Once stated which is the geometry to be aligned and which is the geometry that must be aligned, by picking some reference points of the nominal geometry and then selecting the same points with the same order in the experimental geometry, an automatic algorithm aligns the two. Figure 96. Figure 97. In addition to nominal and experimental geometries, the experimental and nominal cutting lines of interest for each specific case must also be imported. The nominal cutting lines will already be aligned with the nominal geometry, while a transformation matrix must be applied to the experimental cutting lines. The transformation matrix to be applied is the same one used to rotate and translate the experimental geometry to match the nominal one, that can be found in the CloudCompare Console. To apply the transformation matrix, select the lines that need alignment and proceed with 'Edit → Apply Transformation.' In the window that appears, replace the transformation matrix with the one from the Console. Now, the comparison can be performed by selecting the experimental lines and the nominal lines and proceed with 'Compute Cloud/Mesh Distance'. The 'Max Distance' has been set to 10 mm, the 'Signed Distance' has been unchecked and the 'Compute' button is clicked to start the comparison. At this point, all the results that will be described in the following section are available by selecting the compared experimental geometry and can be easily exported as images or files to be used and processed in specific environments such as Microsoft Excel.

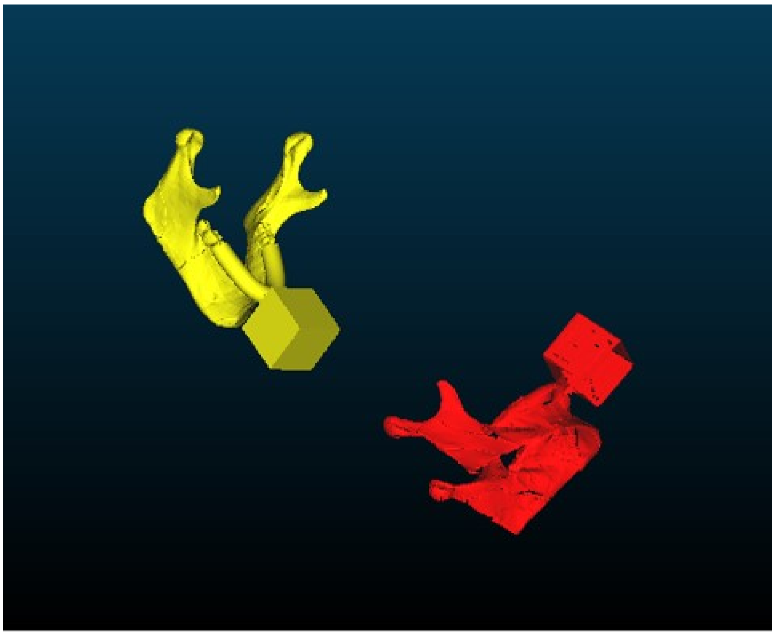


Figure 95 Nominal and experimental mandibles imported in CloudCompare.

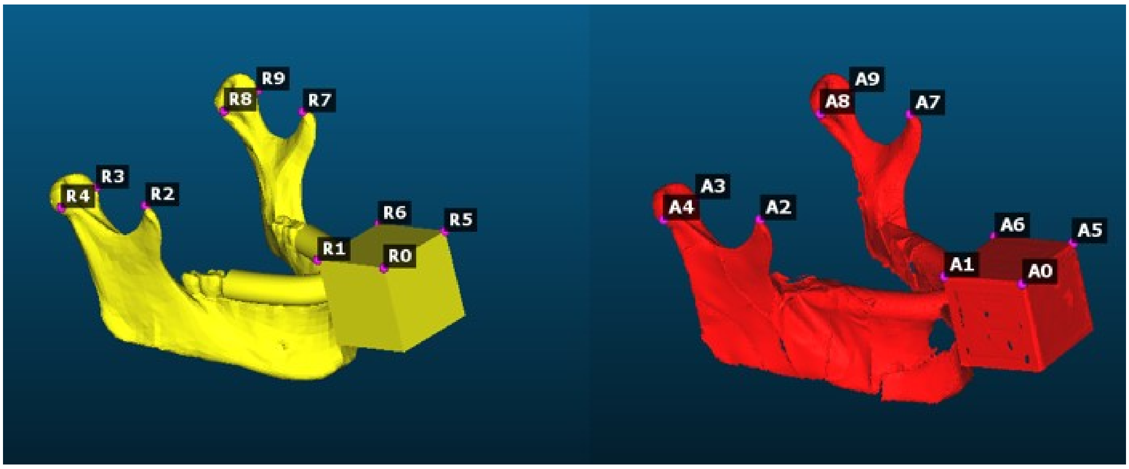


Figure 96 Points picked in the nominal and experimental geometry for the alignment.

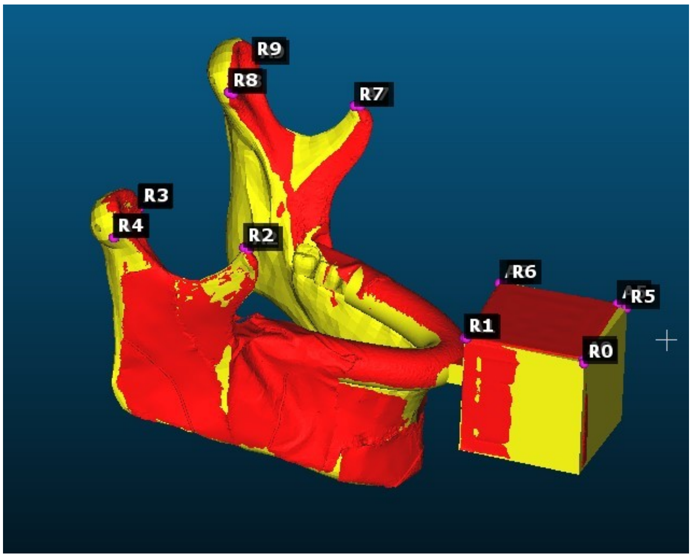


Figure 97 Superimposed mandibles.

5.5 Qualitative Validation

In addition to quantitative validation, qualitative validation is also conducted to compare the two applications developed for the Varjo XR3 and the HoloLens2 and to compare the two headsets through questionnaires administered to users. The questionnaires chosen are the User Experience Questionnaire (UEQ) and the User Experience (UX) survey.

The UEQ is a questionnaire that presents a series of items to the user, each item have the form of a semantic differential, this means that each item consists of two terms with opposite meanings. Users are required to assign a number from 1 to 7, where 1 corresponds to the evaluation that mainly agree with the first term, 4 represents a neutral response, and 7 is the evaluation more aligned with the second term. [76] In this case, the 12 items presented to the users are: Not understandable/Understandable, Difficult to learn/Easy to learn, Obstructive/Supportive, Complicated/Easy, Does not meet expectations/Meets expectations, Inefficient/Efficient, Confusing/Clear, Good/Bad, Organized/Cluttered, Motivating/Demotivating, Conventional/innovative, Useful/Useless. The UEQ was administered to users at the end of each task to assess their experience regarding the usability of the application for the mode just performed.

The UX survey, instead, is a survey in which users are asked to provide a rating ranging from 1 to 7 for a series of specially designed statements. In this case, users were presented with the following 5 statements:

- Q1: I believe that the device is invasive.
- Q2: After a while I was wearing the device, I forgot I was wearing it.
- Q3: I would gladly wear the device having perceived its usefulness.
- Q4: Wearing the device caused me discomfort (nausea, headache, etc.).
- Q5: The use of the device promotes concentration during the activity.

This survey was administered to each user twice, once after completing all the tasks with the Varjo XR3 and the other time after finishing the tasks with the HoloLens. Indeed, the goal of this questionnaire is to compare the two headsets through the responses provided by users regarding their personal experience.

6. Results

In this section, the data extracted from CloudCompare are presented to complete the testing and validation phase. These data are then compared with the data obtained from tests conducted with HoloLens in the previous work related to maxillary osteotomy [1] and with the results obtained by Gioacchini [77], who tested mandibular osteotomy using HoloLens. Furthermore, here the results of the UEQ and UX questionnaires to which users were subjected during the testing phase are presented.

First are reported the images for the first user resulting from the cloud-to-mesh absolute distance with a color scale appositely created within CloudCompare. It is composed by 4 levels of absolute distance: the first level is blue and correspond to the lowest values; the second is green and corresponds to low values; the third level is yellow and corresponds to middle values; the last level is red and correspond to the highest values. Figure 98. Figure 101. Figure 104. Figure 107. Figure 110 Then, figures reporting cloud-to-mesh absolute distance histogram with superimposed the Weibull distribution function Figure 99. Figure 102. Figure 105. Figure 108. Figure 111. Lastly, the graphs of the cumulative distribution functions are reported. Figure 100. Figure 103. Figure 106. Figure 109. Figure 112. The figures with the results for all the other users are reported in the Appendix.

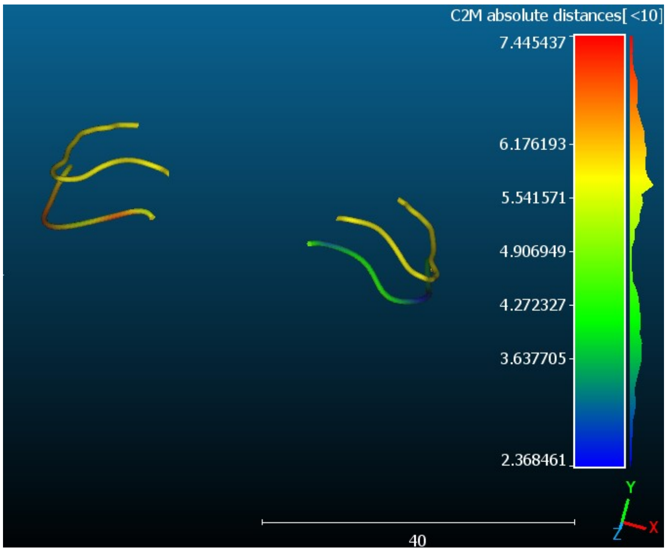


Figure 98 User 01 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines maxillary cutting lines.

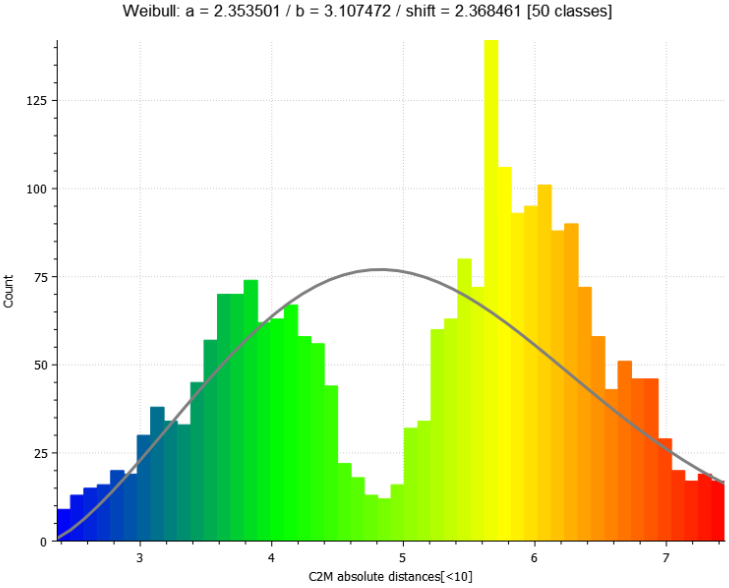


Figure 99 User 01 cloud-to-mesh absolute distance histogram and Weibull function for maxillary cutting lines.

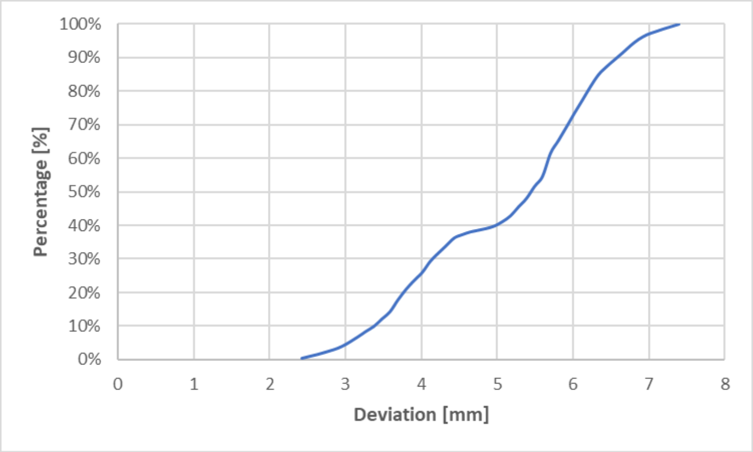


Figure 100 User 01 cumulative distribution function for maxillary cutting lines.

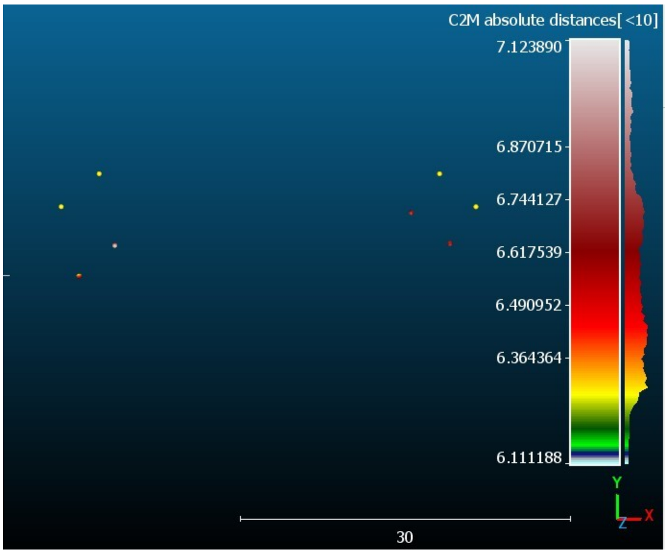


Figure 101 User 01 mapping of cloud-to-mesh absolute distance and nominal (yellow) drilled holes .

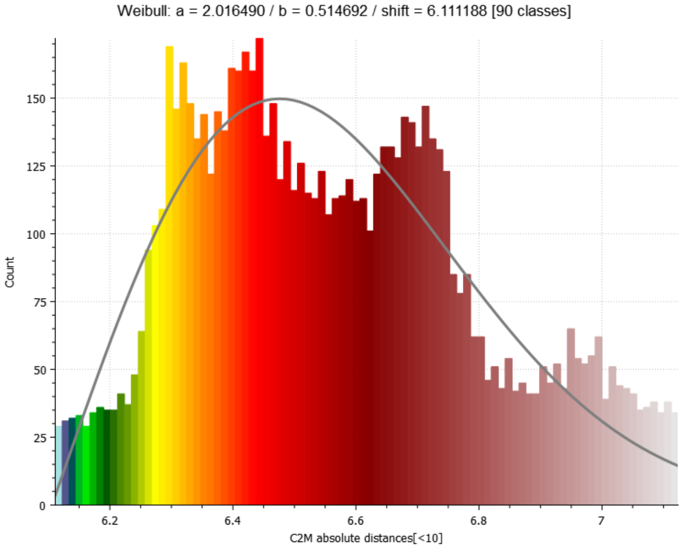


Figure 102 User 01 cloud-to-mesh absolute distance histogram and Weibull function for drilled holes.

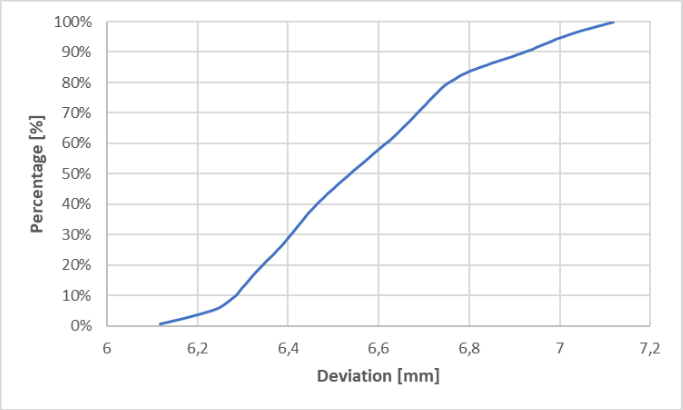


Figure 103 User 01 cumulative distribution function for drilled holes.

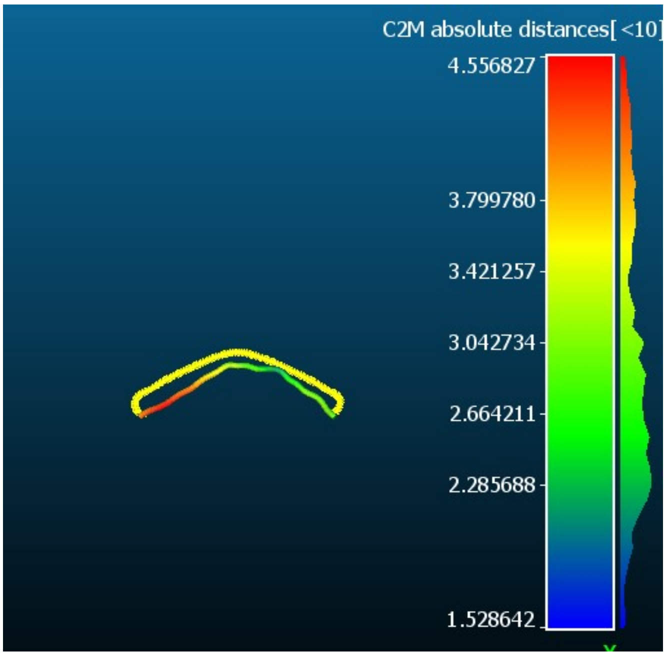


Figure 104 User 01 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for genioplasty.

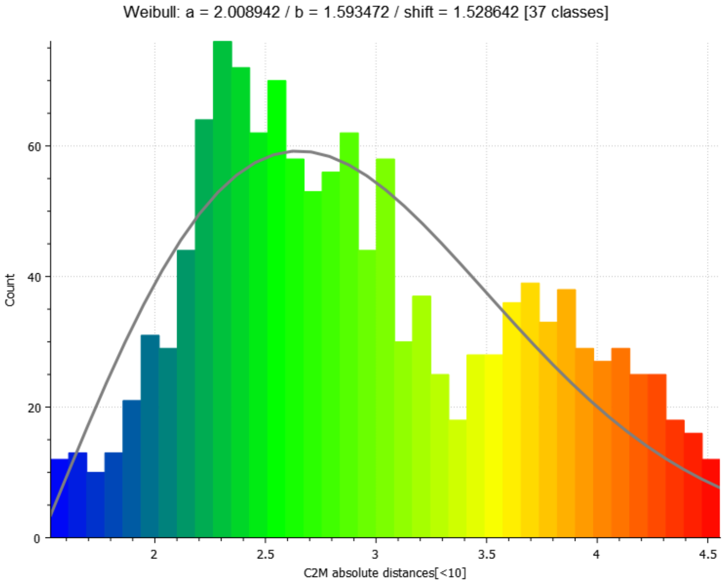


Figure 105 User 01 cloud-to-mesh absolute distance histogram and Weibull function for genioplasty.

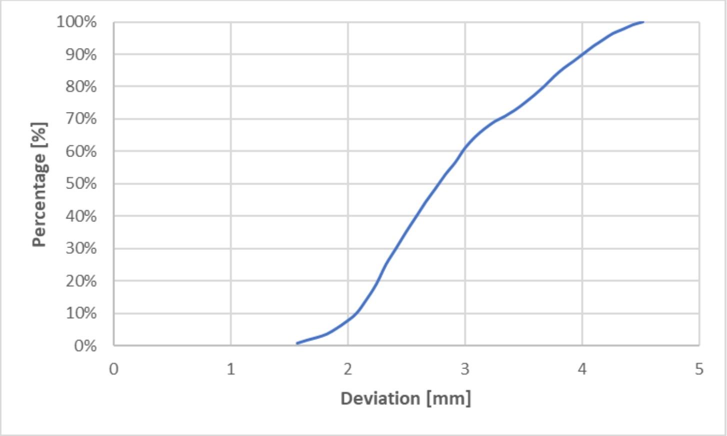


Figure 106 User 01 cumulative distribution function for genioplasty.

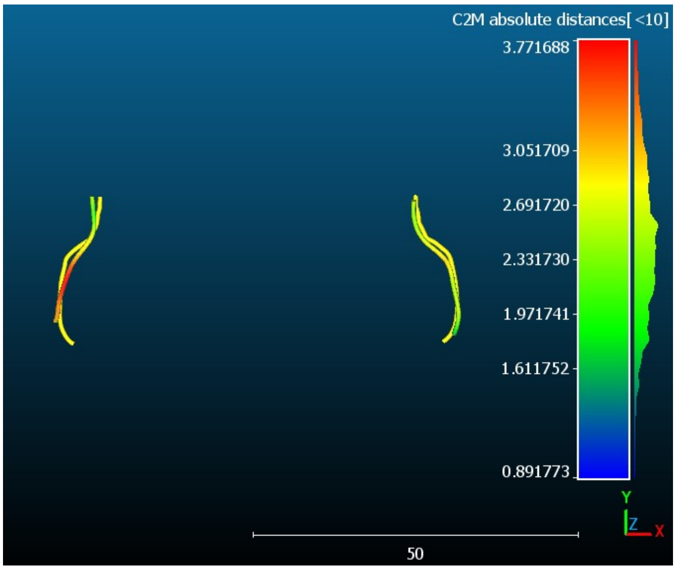


Figure 107 User 01 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for mandibular advancement.

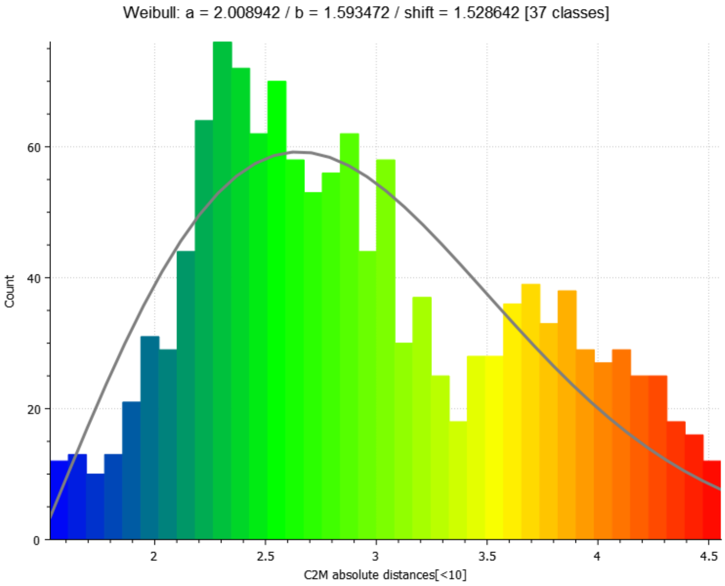


Figure 108 User 01 cloud-to-mesh absolute distance histogram and Weibull function for mandibular advancement.

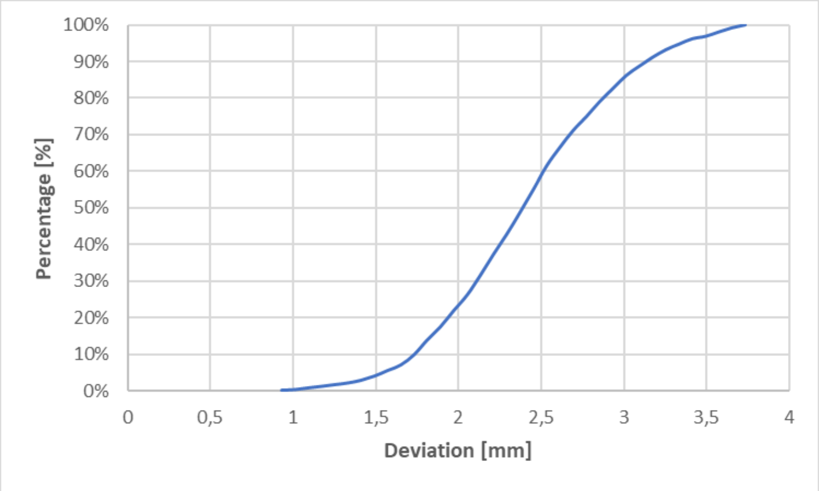


Figure 109 User 01 cumulative distribution function for mandibular advancement.

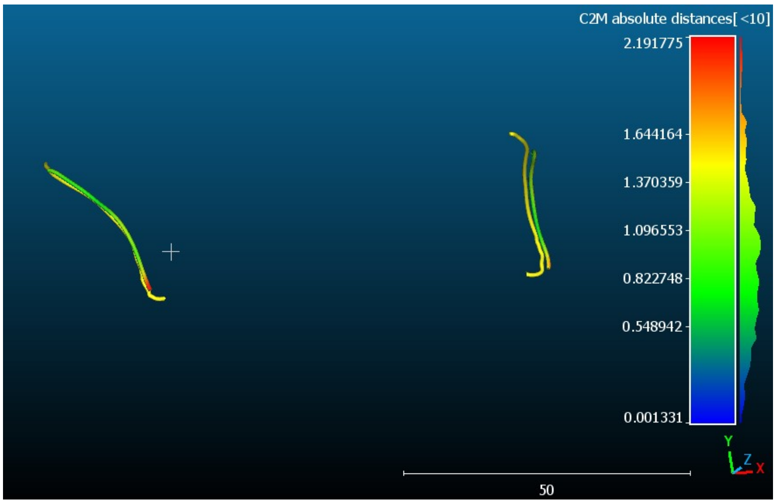


Figure 110 User 01 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for mandibular angle osteotomy.

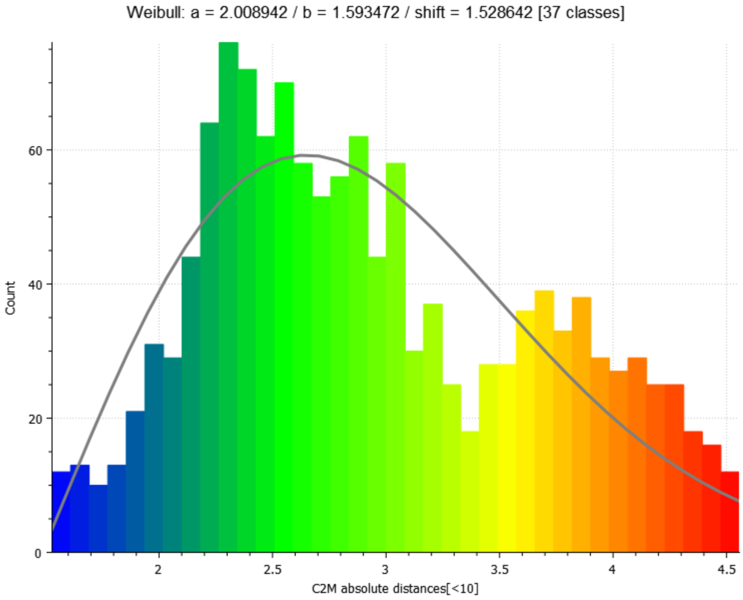


Figure 111 User 01 cloud-to-mesh absolute distance histogram and Weibull function for mandibular angles osteotomy.

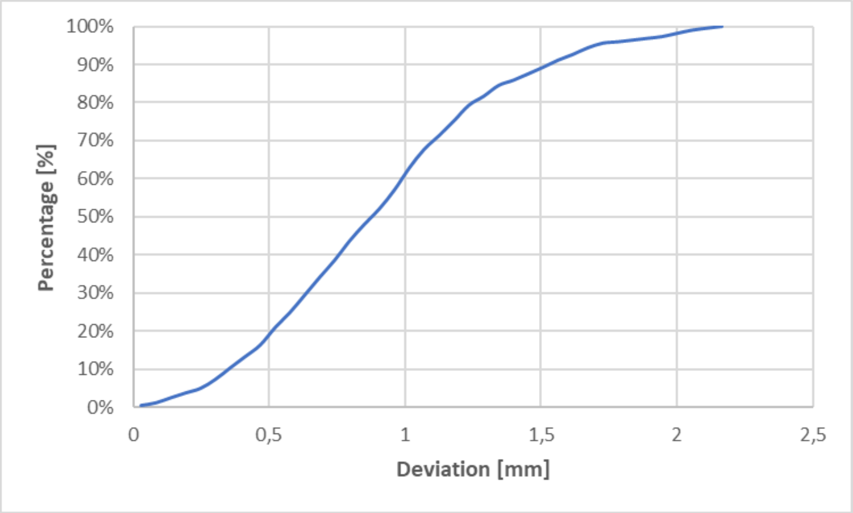


Figure 112 User 01 cumulative distribution function for mandibular angle osteotomy.

From these data exported from CloudCompare, the weighted average and the standard deviation for each curve is computed. From the data exported from CloudCompare, the weighted average and standard deviation for each test have been calculated, separately evaluating the mandibular cutting lines and drilled holes for maxillary osteotomy, as well as the cutting lines for genioplasty, mandibular advancement, and mandibular angle in the context of mandibular osteotomy. Additionally, the mean of the weighted averages and standard deviations for each type of test has been calculated among the 7 users. The results are reported in Table 6, Table 7, Table8, Table 9, Table 10, where CL stands for cutting lines, DH stands for drilled holes, G stands for genioplasty, AVM stands for mandibular advancement, and ANGM stands for mandibular angle.

Table 6 Results for each user maxillary cutting lines trial.

	Weighted Average [mm]	**Standard Deviation [mm]**
U01_CL	5,138797	1,480785
U02_CL	5,614289	2,665241
U03_CL	4,748228	2,59081
U04_CL	5,522905	3,576801
U05_CL	4,796546	1,747074
U06_CL	5,580783	1,23063
U07_CL	3,99729	2,217225
Mean	5,056977	2,215509

Table 7 Results for each user drilled holes trial.

	Weighted Average [mm]	**Standard Deviation [mm]**
U01_DH	6,557821	0,293962
U02_DH	5,719502	0,601258
U03_DH	5,080735	0,37196
U04_DH	4,531855	0,71111
U05_DH	6,122405	0,596219
U06_DH	6,564268	0,417361
U07_DH	6,429828	0,334269
Mean	5,858059	0,475163

Table 8 Results for each user genioplasty trial.

	Weighted Average [mm]	**Standard Deviation [mm]**
U01_G	2,946543	0,88622
U02_G	2,228392	0,944758
U03_G	1,782253	0,510356
U04_G	2,571448	0,729064
U05_G	1,668911	0,584978
U06_G	3,06839	0,672407
U07_G	2,193424	0,523026
Mean	2,351337	0,692973

Table 9 Results for each user mandibular advancement trial.

	Weighted Average [mm]	**Standard Deviation [mm]**
U01_AVM	2,44234	0,842827
U02_AVM	3,439663	1,025979
U03_AVM	3,15603	1,235276
U04_AVM	2,9471	1,106765
U05_AVM	4,619632	1,276209
U06_AVM	5,718141	1,705715
U07_AVM	4,4259	1,021565
Mean	3,821258	1,173477

Table 10 Results for each user mandibular angle osteotomy trial.

	Weighted Average [mm]	**Standard Deviation [mm]**
U01_ANGM	0,932111	0,640182
U02_ANGM	2,551753	1,903928
U03_ANGM	3,051941	2,605878
U04_ANGM	1,57337	1,430438
U05_ANGM	3,953231	2,356139
U06_ANGM	4,59906	2,916492
U07_ANGM	2,309942	1,398684
Mean	2,710201	1,893106

At this point, the results obtained are compared with those previously obtained using HoloLens by Danieli [1] for maxillary osteotomy and Gioacchini [77] for mandibular osteotomy. Specifically, in Table 11, the average standard deviations for the maxillary cutting lines and drilled holes obtained by Danieli [1] and those obtained in this project are reported. In Table 12, on the other hand, the average standard deviations for genioplasty, mandibular advancement, and mandibular angle obtained by Gioacchini [77] and those obtained now are reported.

Table 11 Comparison between average standard deviations about maxillary osteotomy tasks

	Average Standard Deviations	**Average Standard Deviations [mm] Danieli [1]**
Drilled Holes	0,475163	0,69485
Maxilla Cutting Lines	2,215509	0,535377

Table 12 Comparison between average standard deviations about mandibular osteotomy tasks.

	Average Standard Deviations [mm]	**Average Standard Deviations [mm] Gioacchini [77]**
Genioplasty	0,692973	0,754355
Mandibular Advancement	1,173477	0,774581
Mandibular Angle	1,893106	1,229851

Finally, the results concerning qualitative validation are reported. Figure 213 shows the mean results of the UEQ for the maxillary osteotomy performed with Varjo XR3 and those for maxillary osteotomy performed with HoloLens2. In Figure 214 and Figure 215 the results of user experiences regarding the usability of Varjo XR3 and HoloLens2 are reported respectively.

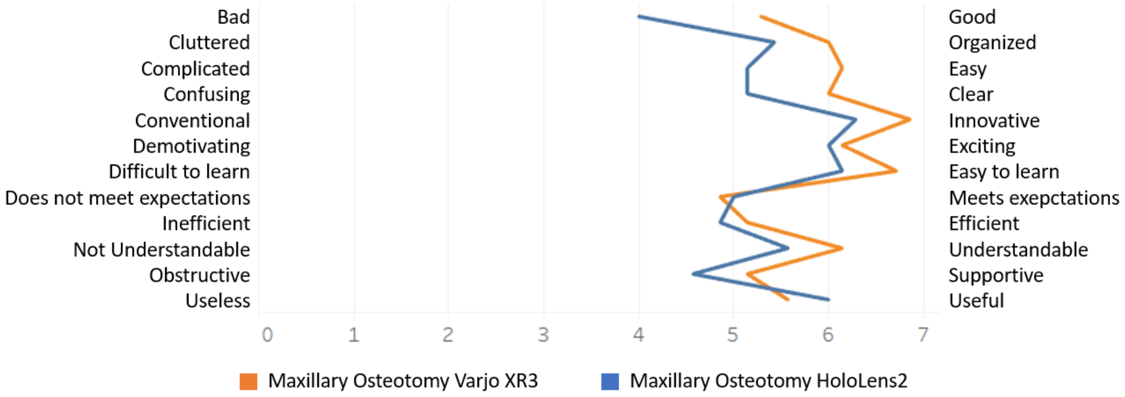


Figure 113 UEQ results for the comparison between maxillary osteotomies performed with Varjo XR3 and HoloLens.

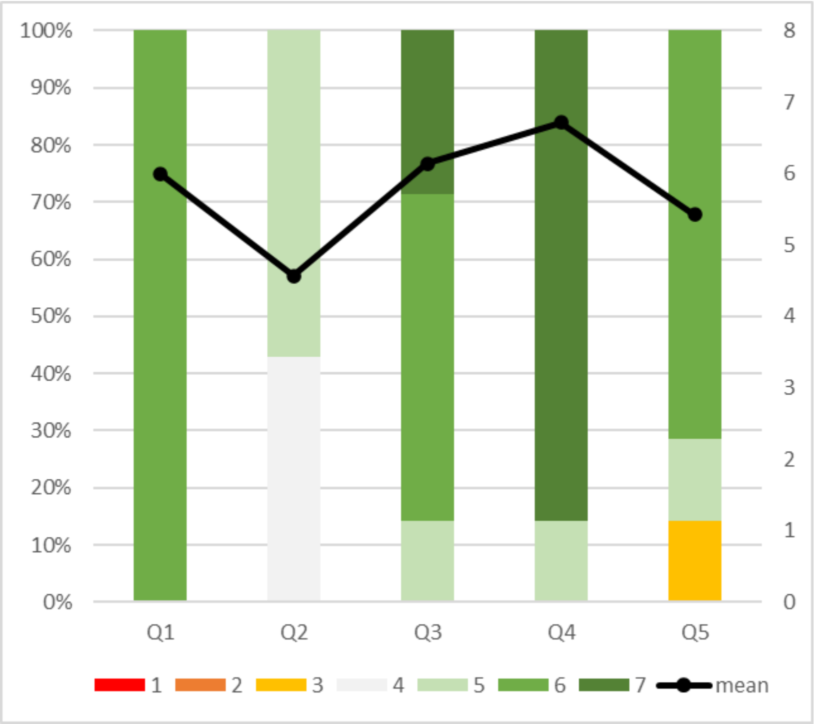


Figure 114 UX survey results for HoloLens2.

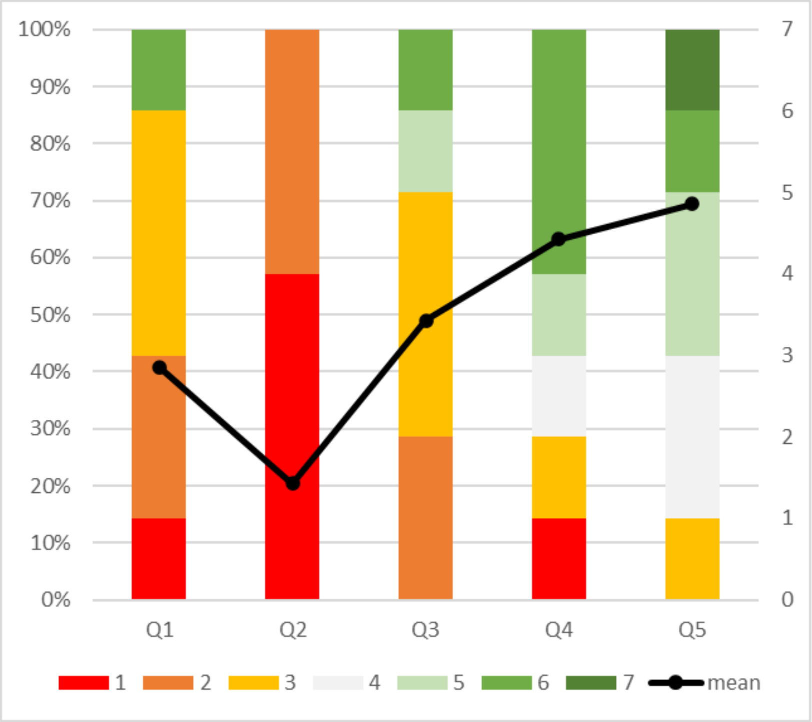


Figure 115 UX survey results for Varjo XR3

7. Discussion and Conclusion

In this section, the obtained results are discussed, and some reflections are provided. Starting from the results extracted from CloudCompare, it is possible to observe significant deviations between the nominal lines and those traced by each user. This was already quite evident during the testing phase; in fact, when using the mixed reality application developed for the Varjo XR3, it was clear that the lines that were displayed on the 3D-printed phantom were not in the correct position. This was particularly noticeable for the mandibular osteotomy cutting lines, as each user knew that they should be visualized at the assembly interface between the cranium and the maxilla since those same lines were used in Rhinoceros to divide the skull model downloaded from Sketchfab into the two cranial and maxilla components. While the deviation of the mandibular cutting lines was less apparent during testing, in reality, the average distances are more or less similar, as reported in Tables 7-10. In fact, for the maxilla cutting lines and drilled holes, the average distance is approximately 5mm and 3mm, respectively, and for the genioplasty, mandibular advancement, and mandibular angle cutting lines, the average distances are approximately 2mm, 4mm, and 3mm, respectively. The reasons for this misalignment of the cutting lines and fixation holes may be attributed to several factors. Firstly, the placement of holograms is entirely dependent on the 3D digital model because the distances calculated in Rhinoceros between the curves and their respective markers were put as input into the 'Hologram_Positioning' script assigned to the holograms to determine their placement. However, there are differences between the digital model and the 3D printed physical model in terms of size, due to print tolerances, deformations, and shrinkage of the material during printing or hardening, printer calibration, and the choice of printing orientation. Additionally, there were printing issues concerning the mandible and its marker, which theoretically should have been printed as a single piece but, in practice, were printed as two separate pieces. Marker M3 had a recess that corresponded to the bridge structure in the mandible, so these two pieces were fitted together for testing, introducing an additional source of uncertainty. Furthermore, the size of the printed markers was about 1mm larger on each side, causing inaccuracies in the center calculation using the 'CalcolaCentroCubo' script. The calculation became even less accurate when considering that the VarjoMarkers were manually glued on a larger surface than the VarjoMarker itself, affecting the precision of marker placement and orientation on the faces of the three cubes. Additionally, the holes on the back faces of M1 and M2 were enlarged using a drill since, despite being designed with a 2mm diameter, they were not large enough to accommodate the screws for anchoring to the cranium and to the maxilla. Enlarging the holes made them slightly larger than originally designed, so the orientation of the cubes during the testing phase was not exactly the same as in the 3D model. Lastly, the position of the hologram was not entirely stable when the user moved to perform tasks. This instability could be eliminated by improving the algorithm that positions the holograms based on the tracked marker, and it might be preferable to position the holograms using the positions of two markers rather than one. This way, if one marker's tracking is lost, the hologram's position remains stable due to the tracking of the other marker.

Since the average distance of the experimental lines from the nominal ones is a value that appears to be of little significance when compared to the average distances observed in the studies [1] and [77], it was decided to conduct a comparison of standard deviations to assess whether the misalignment of the lines could be attributed to a constant offset but the overall morphology of the traced lines were close to the nominal morphology. The quantitative results were compared with the findings presented in the works of Danieli [1] and Gioacchini [77] pertaining to maxillary osteotomy and mandibular osteotomy, respectively, as reported in Table 11 and Table 12. From these results and the comparison of the distribution images presented in this thesis and in [1] and [77], it becomes apparent that the distribution of points is broader compared to the previous works. It's worth noting that only the results of the cutting curves for genioplasty appear similar to the previously obtained results, with a standard deviation on the order of 0.7 mm. This implies that, from a morphological standpoint, there is greater uncertainty experienced by users in accurately tracing the cutting lines according to the nominal ones displayed on the skull. Consequently, it is important, to investigate the underlying causes. Certainly, the instability of the hologram does not aid in accurately tracing the cutting lines, but the primary issue encountered by all users during testing lies in the difficulty of perceiving the depth correctly when placing the pen in contact with the geometry and following the cutting lines, as the tip of the pen was obscured by the presence of the hologram.

Now, let's analyze the qualitative results obtained from the questionnaires. Regarding the comparison between Varjo XR3 and HoloLens2 applications for assessing maxillary osteotomies, as indicated by the UEQ mentioned in the previous section (Figure 213), the results are quite similar, with slightly better overall results in terms of task performance with Varjo XR3. In fact, the latter is found to be more understandable, supportive, exciting, innovative, clear, more organized, and efficient, even though it is perceived as less useful and less responsive to meeting user needs. As for the user experience related to the use of the headsets, the difference between the two is quite evident. The results of the two user experiences are presented in Figure 214 and Figure 215. In fact, while everyone agrees that HoloLens is non-invasive, over 80% of users consider Varjo XR3 to be invasive. This judgment aligns with the results of the second question where users were asked if they forgot they were wearing the device after a while. For HoloLens, the majority agrees with this statement, while the rest remain neutral on the subject. However, for Varjo, all users strongly disagree with the statement. This is primarily because Varjo XR3 is a much heavier device due to the cameras that enable the visualization of the external environment, compared to HoloLens2. HoloLens2 weighs about half a kilogram, while Varjo XR3 weighs nearly a kilogram, and this weight is partially increased by the cables required for connection to the workstation. This difference limits the execution of tasks and overall comfort during use. Invasiveness could also refer to the fact that in Varjo XR3, external world vision is only possible through the cameras, so any malfunction of the headset would render the user blind. In contrast, HoloLens2 provides a direct view of the real world through its lenses, making an abrupt interruption of the view during tasks impossible. These considerations also seem valid when looking at the results of the third statement regarding the will in the usage the headset for various activities. For HoloLens2, all the responses range from agreement to strong agreement, whereas for Varjo XR3, only 30% of the users agree, while the remaining 70% disagree. No users experienced discomfort like nausea or headaches, etc. while wearing HoloLens2. In contrast, when using Varjo XR3, the majority of users did not report similar discomfort, but it's essential to note that nearly 30% of users encountered these issues. This result may be attributed to the greater weight of the device and the altered depth perception through the use of external cameras, which can destabilize users, making them feel uncertain in their movements and experience side effects with the use of Varjo XR3. Finally, concerning considerations about the use of these devices with respect to concentration on tasks, over 80% agree that the use of HoloLens enhances concentration, while for Varjo XR3, the percentage of users in agreement is lower, at around 60%.

Therefore, based on the results obtained, the overall experience is better with HoloLens despite the notable qualities of Varjo XR3. In fact, Varjo XR3 enables better integration of the real with the virtual, providing holograms with higher contrast, making them more visible in the real world, thus enhancing the user's perception of these objects in their environment. Furthermore, the use of Varjo XR3 and its specific markers, VarjoMarkers, allows the user to work at a closer distance compared to HoloLens2. For instance, in this thesis, "small" markers were chosen, for which tracking, and the consequent hologram display are optimal up to half a meter and can be tracked up to one meter away. In contrast, HoloLens2 requires an optimal working distance of 2 meters for a clear hologram display, a condition clearly impractical in the context of a surgical operation.

In conclusion, the application presents several critical issues that must be overcome if the goal is to utilize mixed reality in a surgical environment. As previously suggested, the marker positioning should be improved by implementing the algorithm that track the markers, possibly utilizing the position of multiple markers so that both the placement and stability of holograms are enhanced and guaranteed when the user moves to find the best position to complete tasks. Additionally, it would be desirable to find a way to reduce the dependency of hologram positioning on the 3D model's geometry, as the physical model will inevitably have differences in size compared to the virtual model. These differences, when combined with other sources of uncertainty, can lead to less than optimal results, as in this case. Another interesting goal for future applications is to utilize facial features as reference points for hologram positioning. Although currently, tracking without using markers is less accurate than tracking with markers for proper registration of virtual content on real objects, this would be extremely useful in real scenarios to avoid using markers that may obstruct the surgeon during the operation and that can be difficult to stably attach to the patient. Therefore, technological advancements in marker-less tracking should be closely monitored, as they would be more practical in the context of a real surgical operation. As previously mentioned, even the Varjo XR3 device presents some issues, despite being the best mixed reality device currently available due to its wider field of view and incredible visual performance. The issue of the heaviness of the head-mounted display, which limits the user in performing tasks and can lead to discomfort for the wearer, can be improved by keeping the cables that connect the headset to the workstation elevated. These cables not only increase the perceived weight on the user's head and neck but also restrict their movements. In this context, the tasks are not particularly dynamic, but the problem of cable interference has been observed even with small user adjustments for task execution. Furthermore, as noted by all users, the perception of depth while wearing the headset is different from normal vision. This could be partially addressed by including an initial user adaptation phase in the testing protocol to help users adjust to this altered depth perception, ultimately improving task performance. The issue of difficulty in visualizing the pen tip when it overlaps with holograms, specifically the cutting lines, could be improved by increasing the transparency of the holograms. However, it's important to avoid making them too indistinct. Additionally, a valuable idea could be the tracking of the pen tip with feedback displayed to the user to ensure that the user is always aware of how he is moving and whether he is correctly following the cutting lines.

At present, it is not possible to test the application in a real-world context, but mixed reality technology remains highly attractive and holds immense potential, which keeps the interest in introducing this technology into operating rooms high. Future efforts aimed at improving the application and the overall experience based on the considerations and suggestions provided will bring closer to the goal of using mixed reality in maxillofacial surgery.

8. Appendix

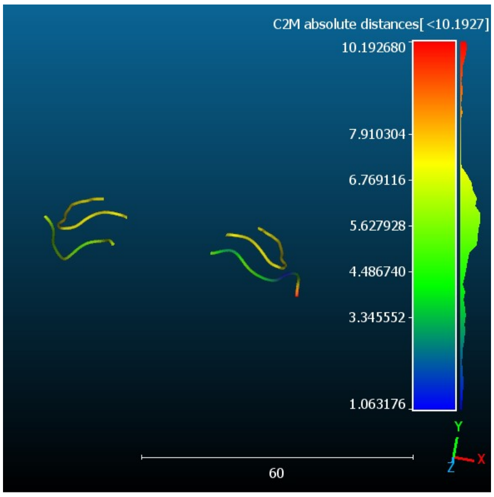


Figure 116 User 02 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines maxillary cutting lines.

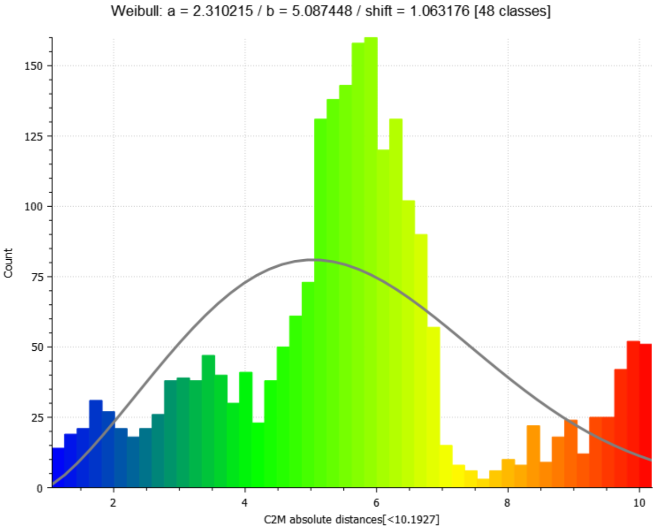


Figure 117 User 02 cloud-to-mesh absolute distance histogram and Weibull function for maxillary cutting lines.

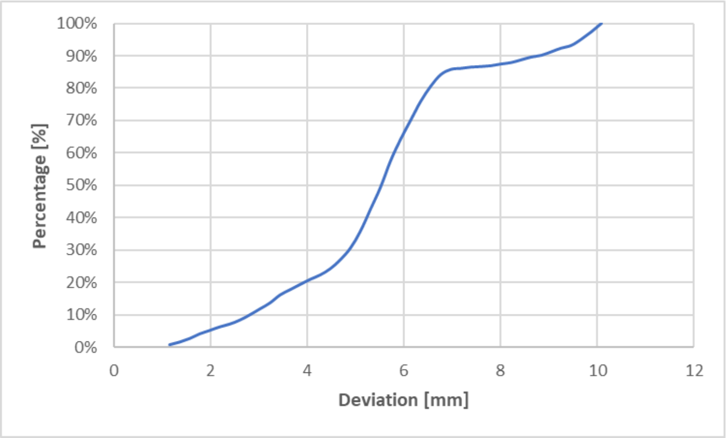


Figure 118 User 02 cumulative distribution function for maxillary cutting lines.

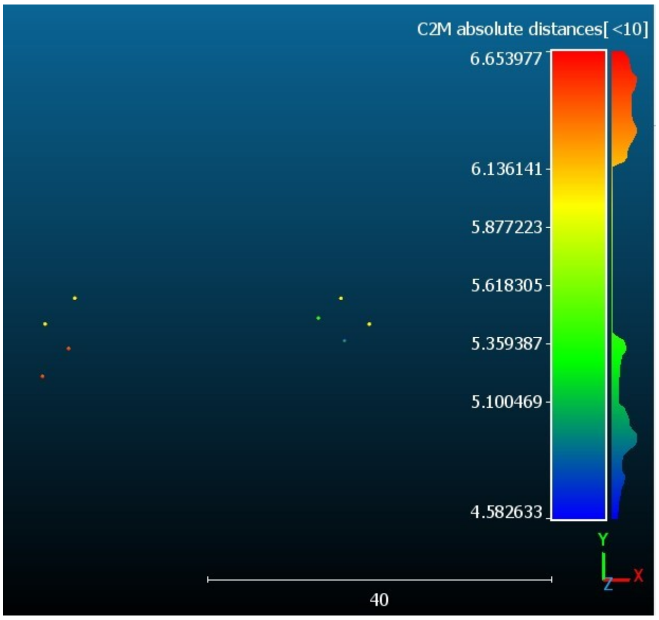


Figure 119 User 02 mapping of cloud-to-mesh absolute distance and nominal (yellow) drilled holes.

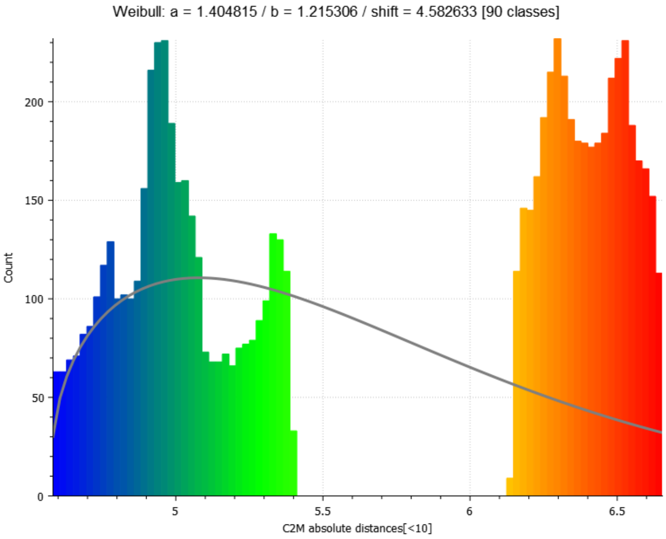


Figure 120 User 02 cloud-to-mesh absolute distance histogram and Weibull function for drilled holes.

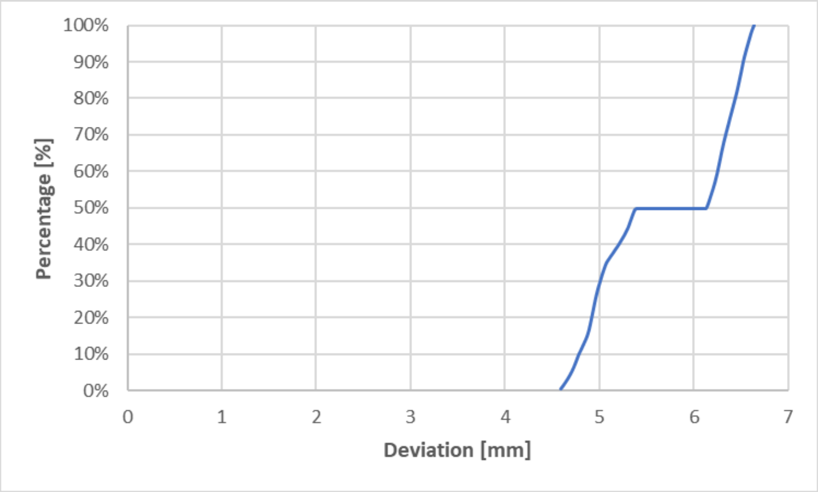


Figure 121 User 02 cumulative distribution function for drilled holes.

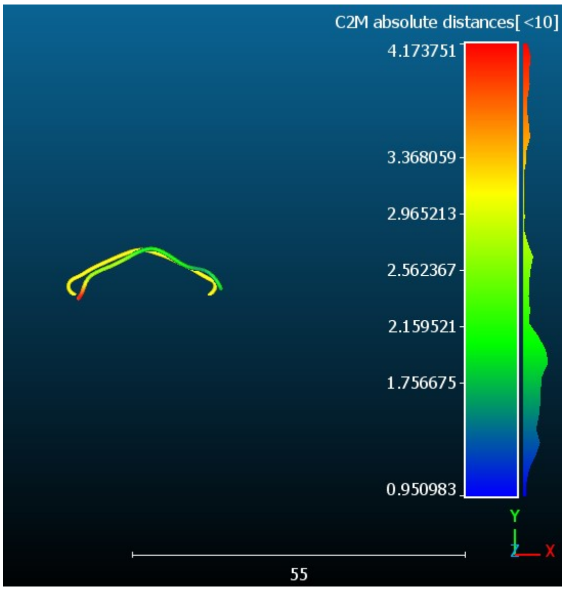


Figure 122 User 02 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for genioplasty.

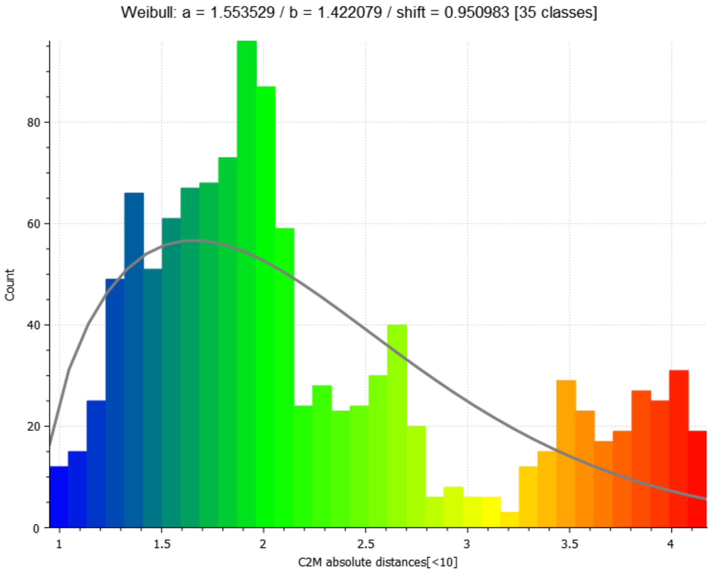


Figure 123 User 02 cloud-to-mesh absolute distance histogram and Weibull function for genioplasty.

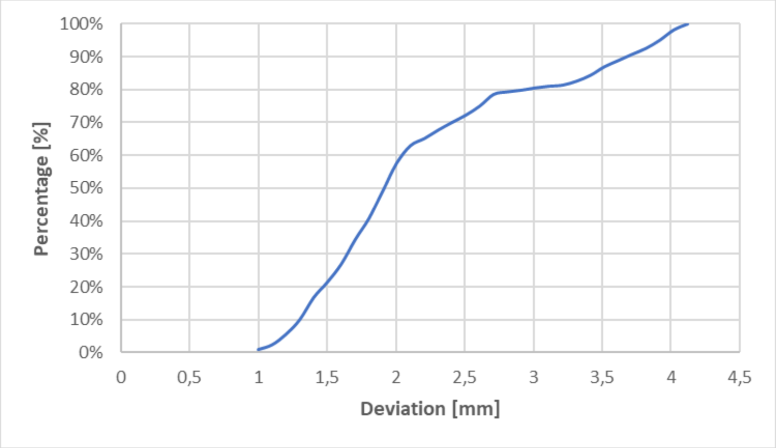


Figure 124 User 02 cumulative distribution function for genioplasty.

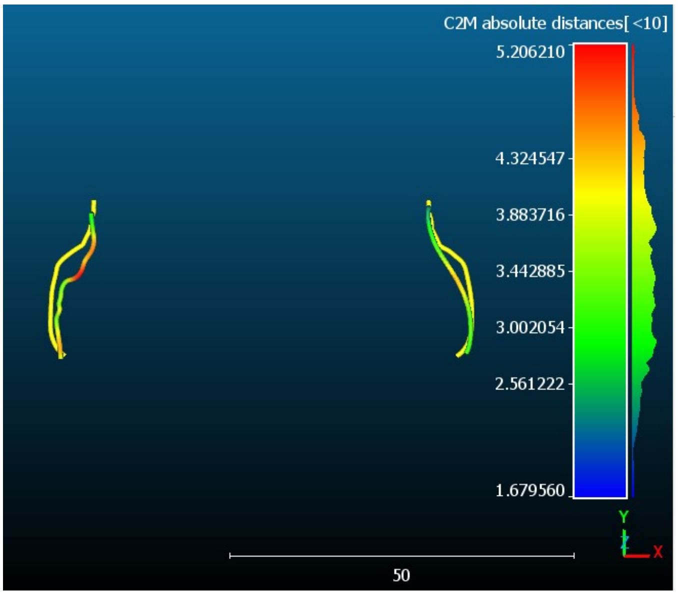


Figure 125 User 02 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for mandibular advancement.

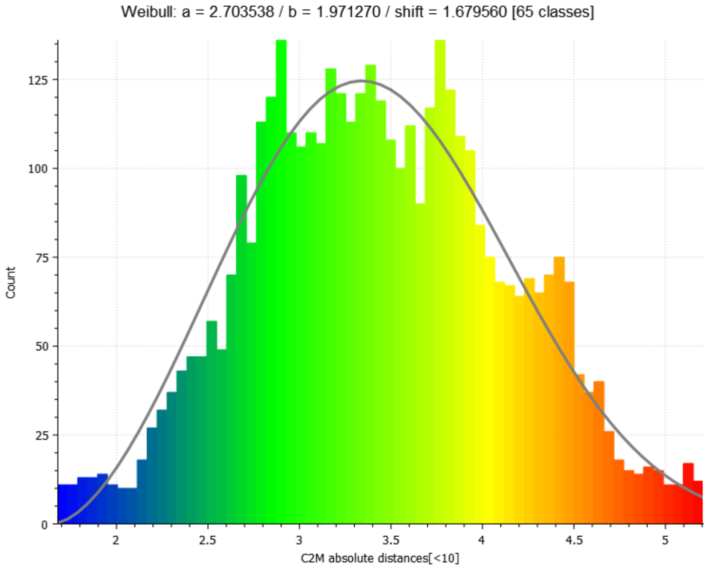


Figure 126 User 02 cloud-to-mesh absolute distance histogram and Weibull function for mandibular advancement.

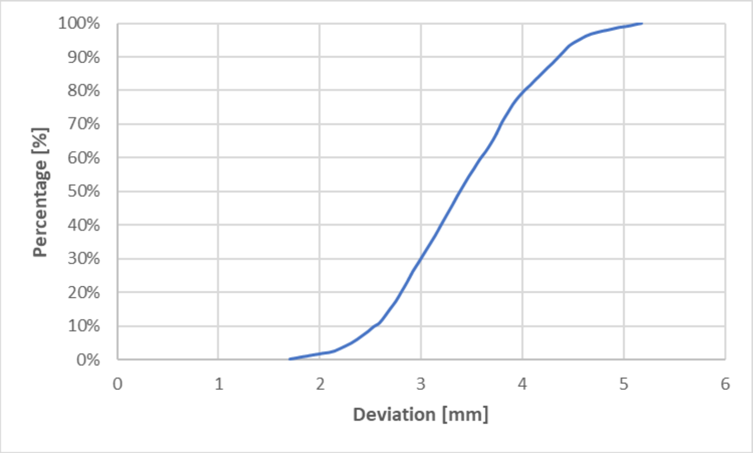


Figure 127 User 02 cumulative distribution function for mandibular advancement.

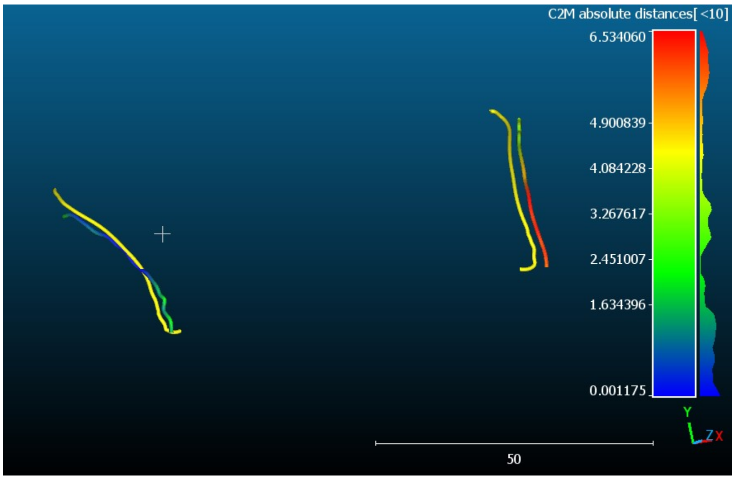


Figure 128 User 02 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for mandibular angle osteotomy.

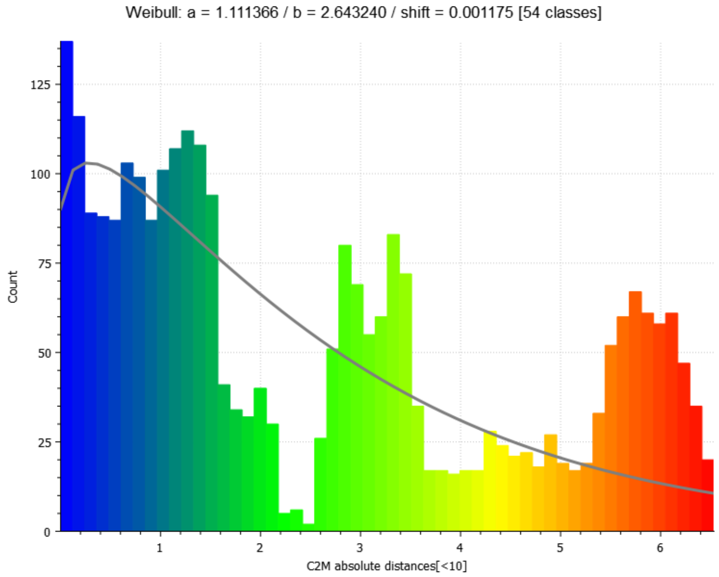


Figure 129 User 02 cloud-to-mesh absolute distance histogram and Weibull function for mandibular angles osteotomy.

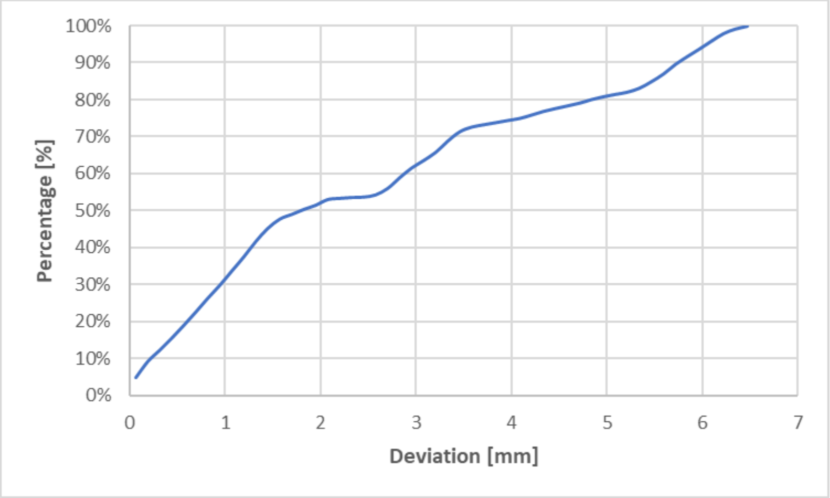


Figure 130 User 02 cumulative distribution function for mandibular angle osteotomy.

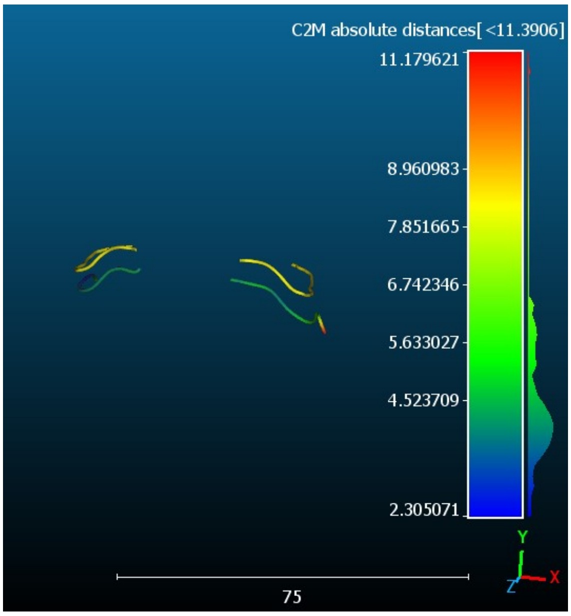


Figure 131 User 03 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines maxillary cutting lines.

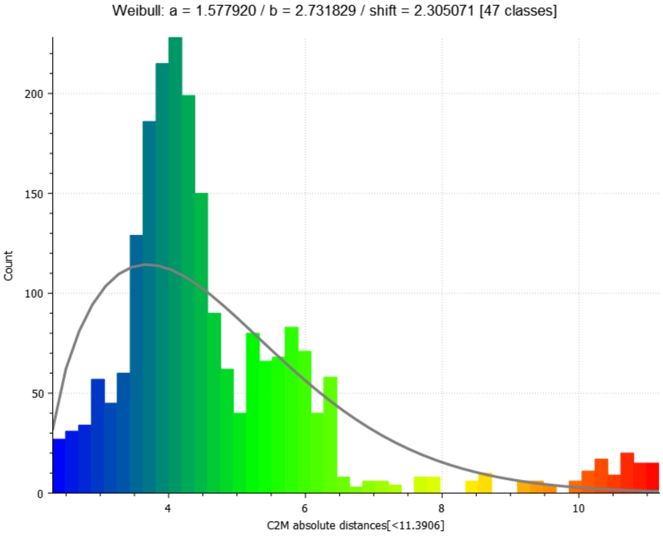


Figure 132 User 03 cloud-to-mesh absolute distance histogram and Weibull function for maxillary cutting lines.

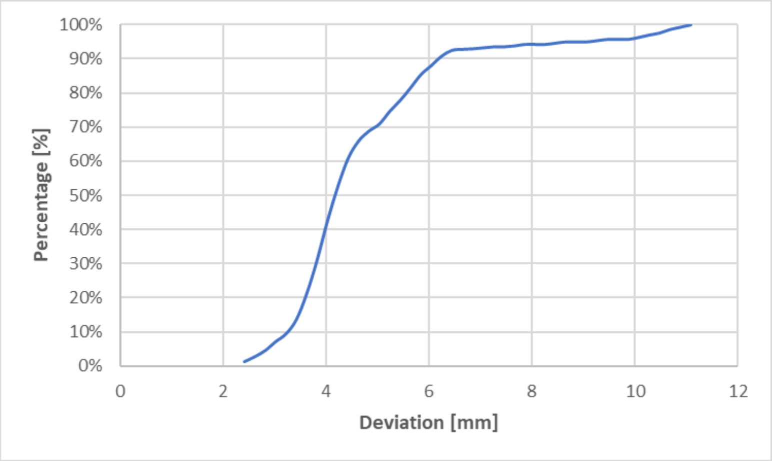


Figure 133 User 03 cumulative distribution function for maxillary cutting lines.

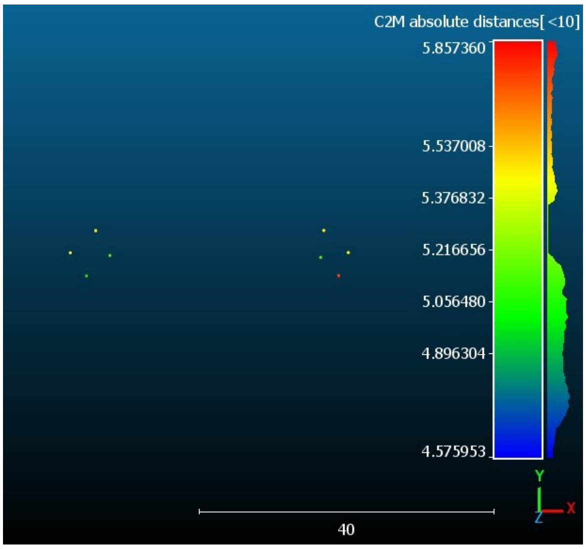


Figure 134 User 03 mapping of cloud-to-mesh absolute distance and nominal (yellow) drilled holes.

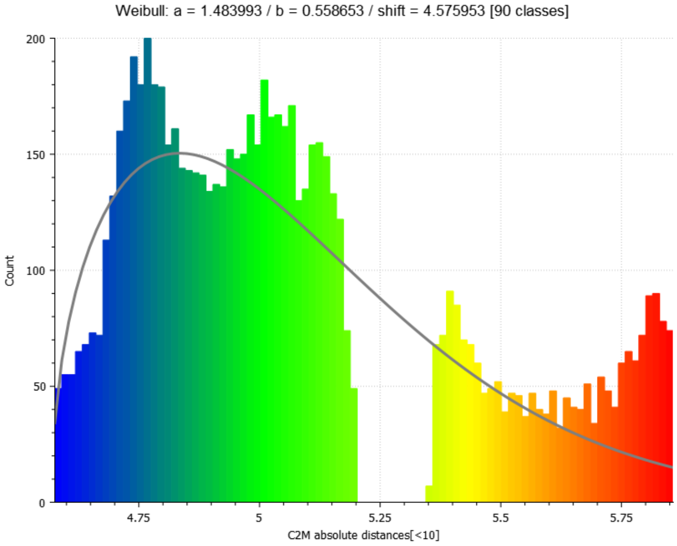


Figure 135 User 03 cloud-to-mesh absolute distance histogram and Weibull function for drilled holes.

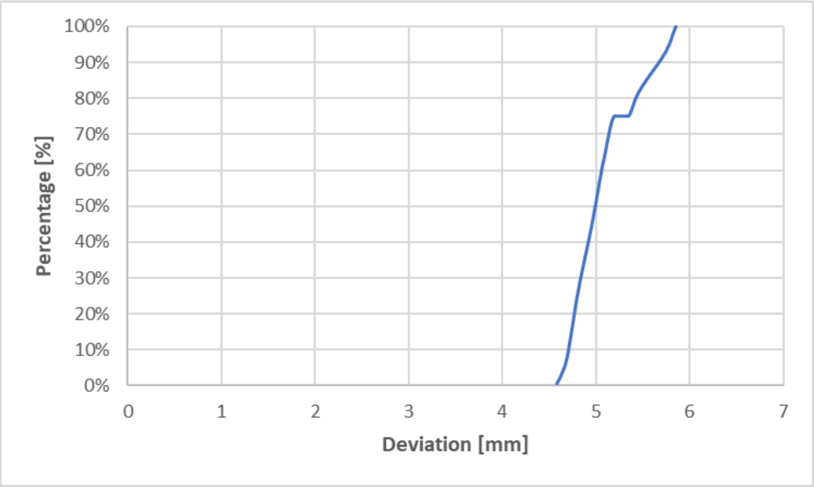


Figure 136 User 03 cumulative distribution function for drilled holes.

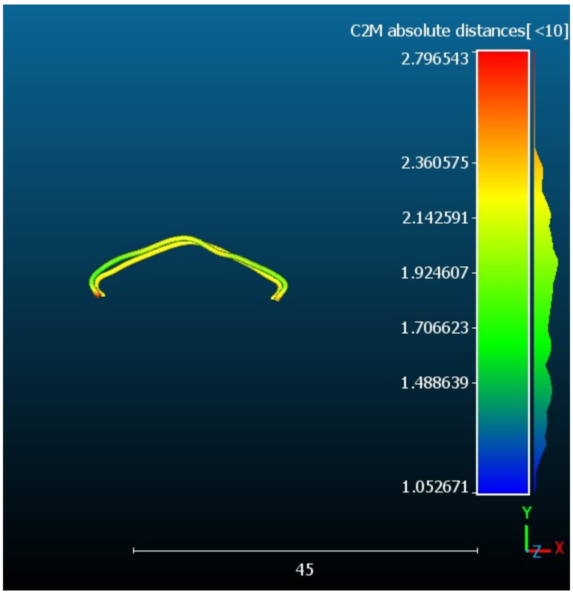


Figure 137 User 03 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for genioplasty.

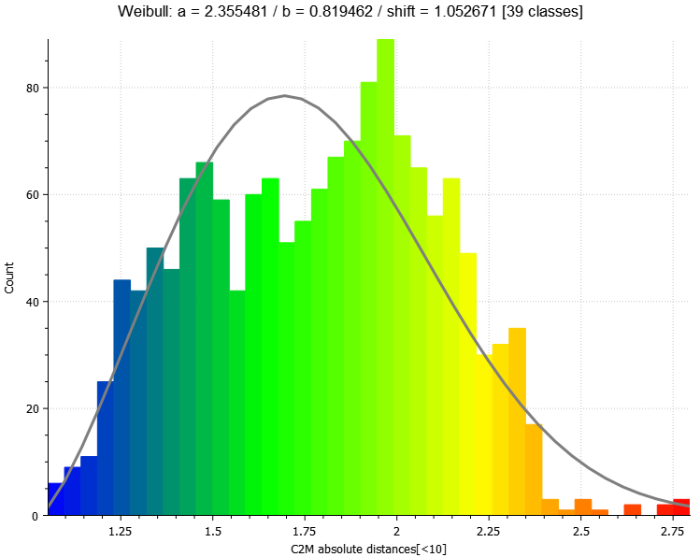


Figure 138 User 03 cloud-to-mesh absolute distance histogram and Weibull function for genioplasty.

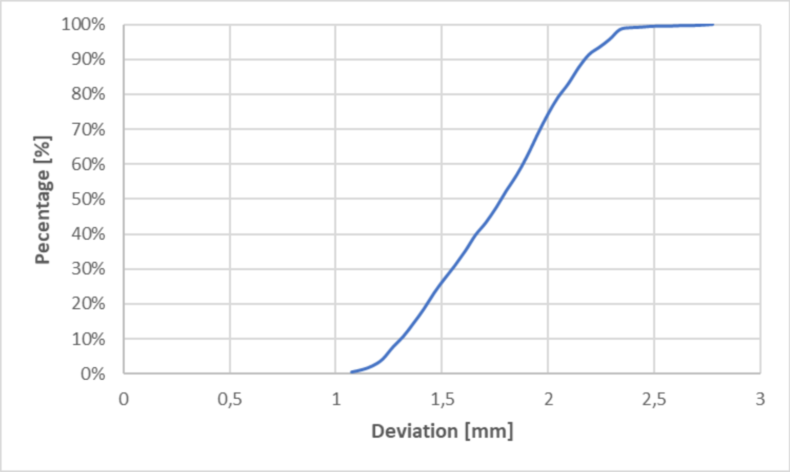


Figure 139 User 03 cumulative distribution function for genioplasty.

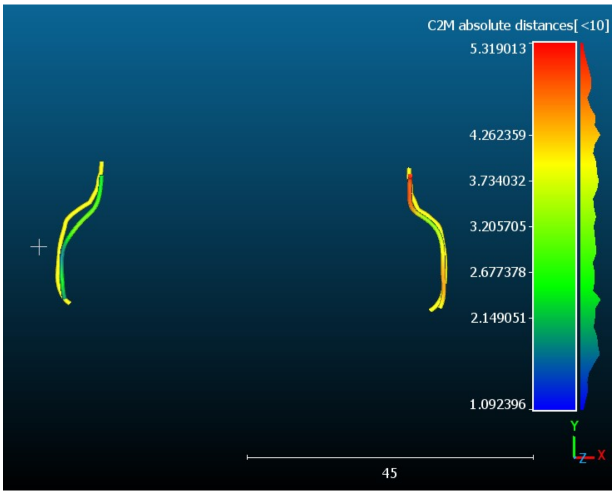


Figure 140 User 03 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for mandibular advancement.

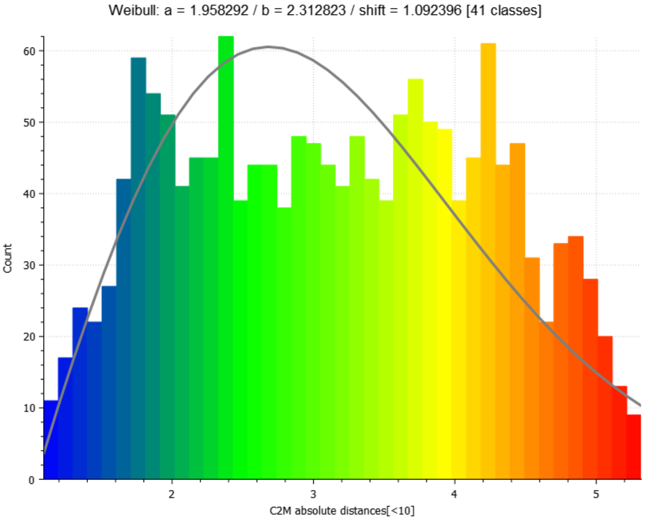


Figure 141 User 03 cloud-to-mesh absolute distance histogram and Weibull function for mandibular advancement.

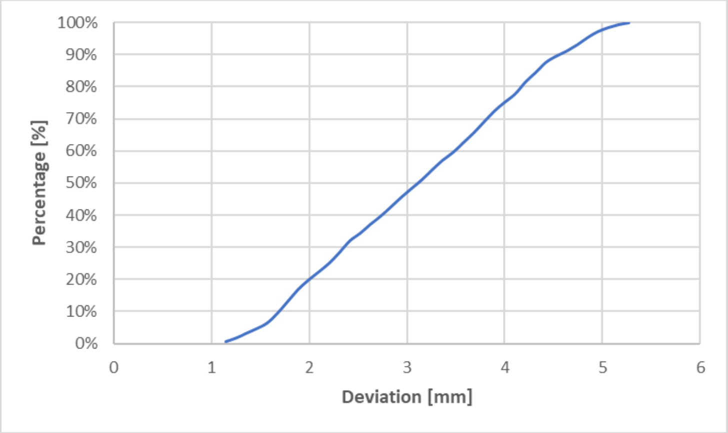


Figure 142 User 03 cumulative distribution function for mandibular advancement.

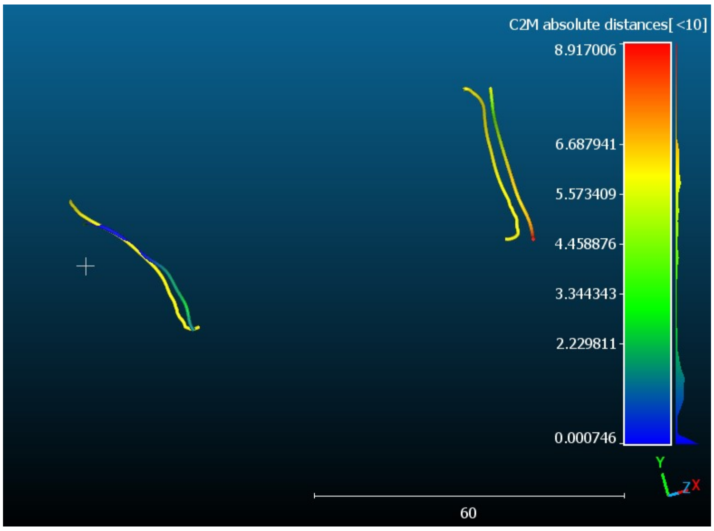


Figure 143 User 03 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for mandibular angle osteotomy.

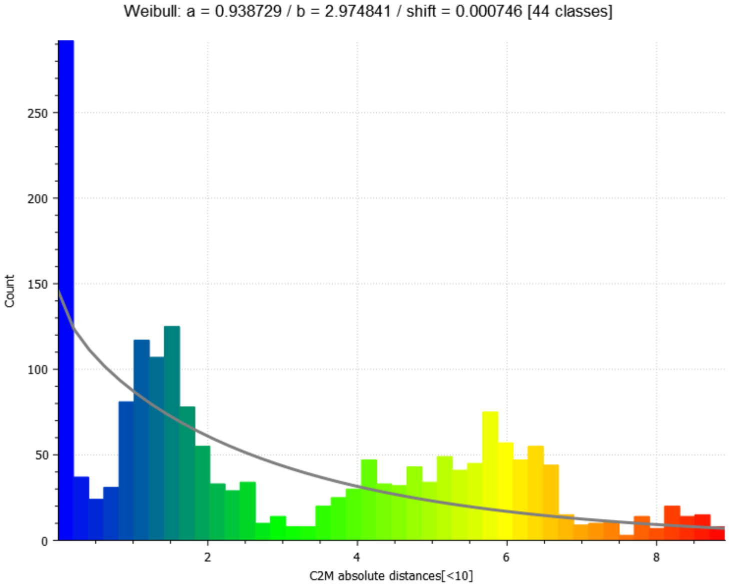


Figure 144 User 03 cloud-to-mesh absolute distance histogram and Weibull function for mandibular angles osteotomy.

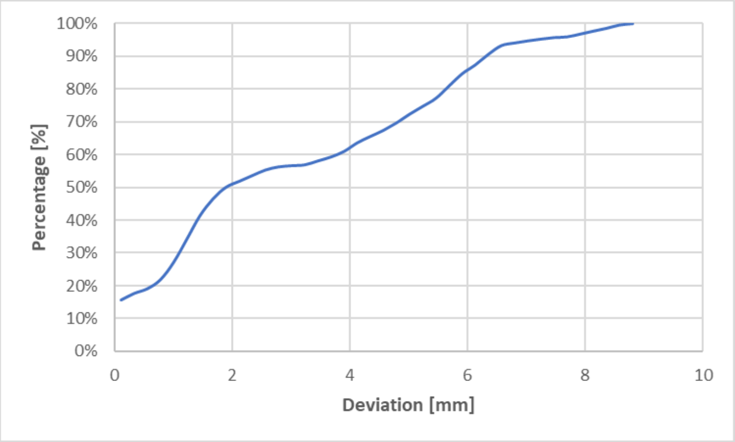


Figure 145 User 03 cumulative distribution function for mandibular angle osteotomy.

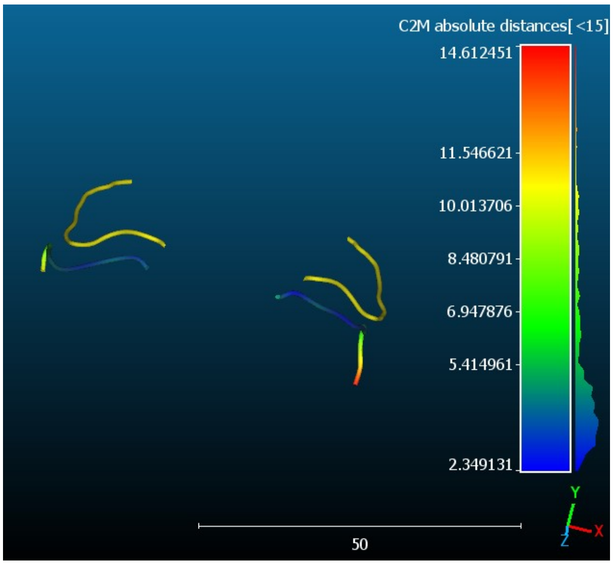


Figure 146 User 04 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines maxillary cutting lines.

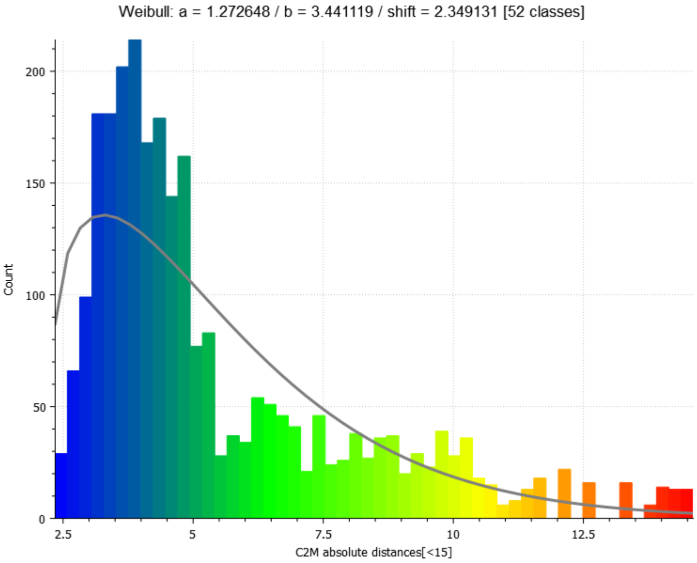


Figure 147 User 04 cloud-to-mesh absolute distance histogram and Weibull function for maxillary cutting lines.

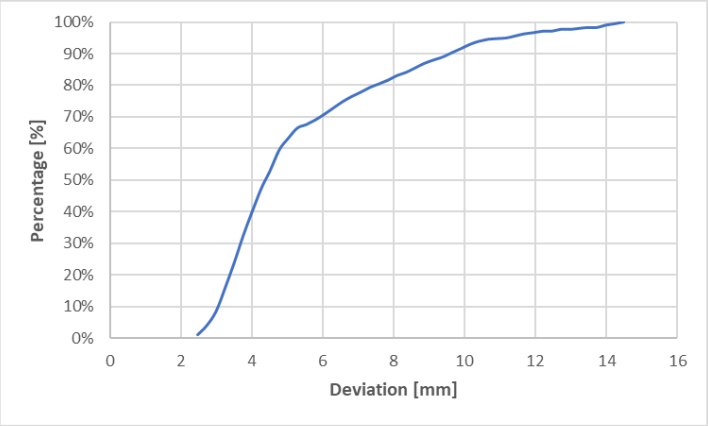


Figure 148 User 04 cumulative distribution function for maxillary cutting lines.

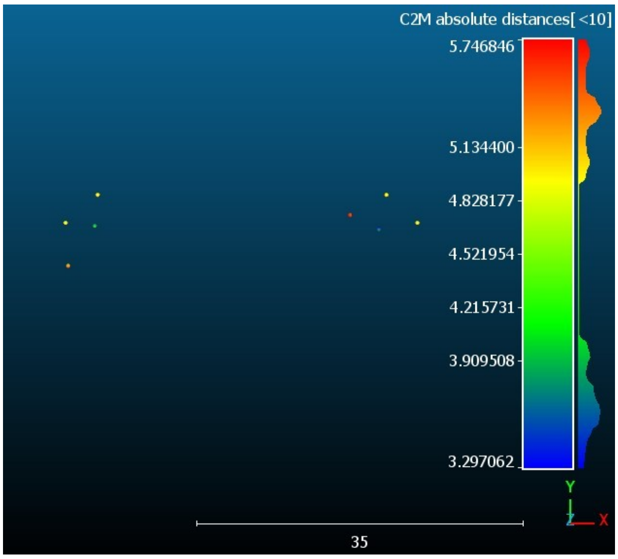


Figure 149 User 04 mapping of cloud-to-mesh absolute distance and nominal (yellow) drilled holes.

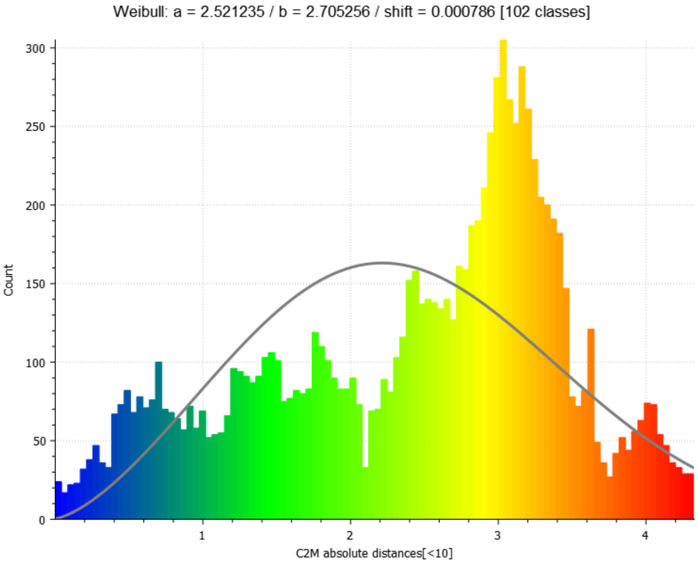


Figure 150 User 04 cloud-to-mesh absolute distance histogram and Weibull function for drilled holes.

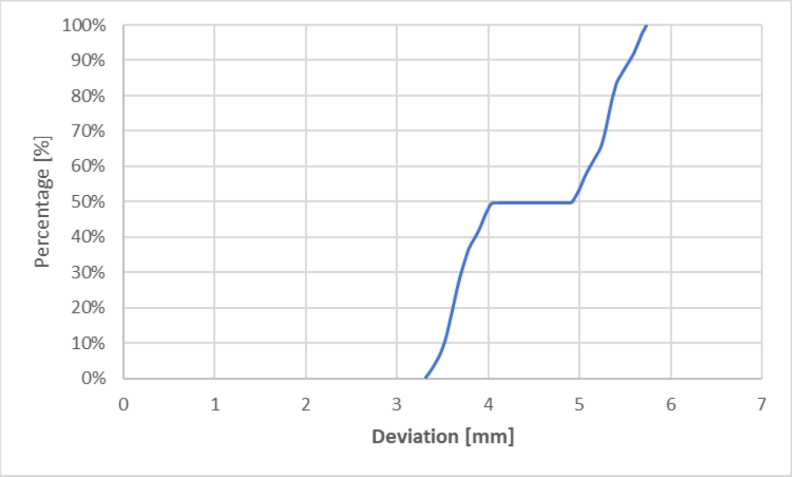


Figure 151 User 04 cumulative distribution function for drilled holes.

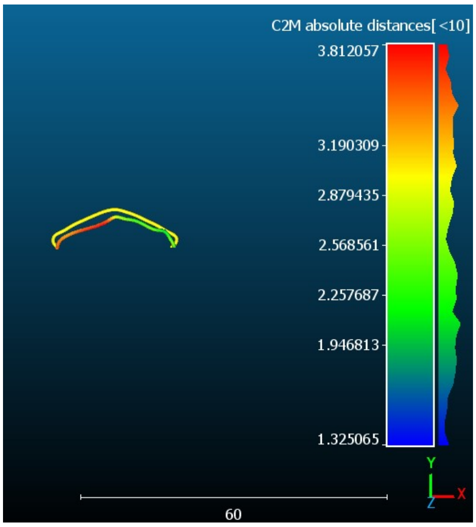


Figure 152 User 04 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for genioplasty.

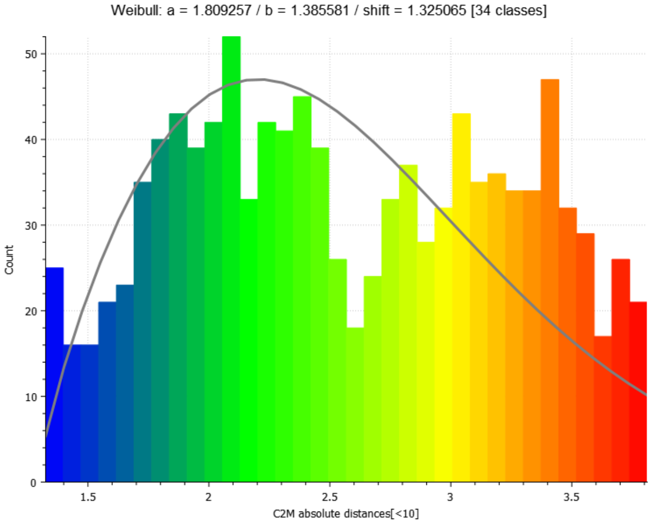


Figure 153 User 04 cloud-to-mesh absolute distance histogram and Weibull function for genioplasty.

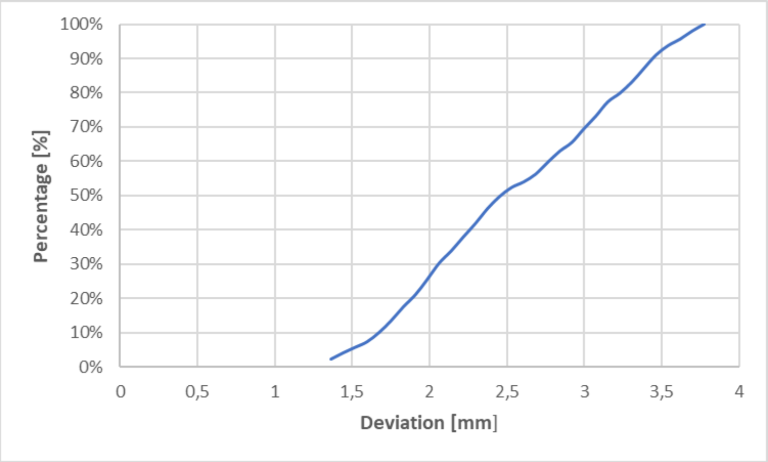


Figure 154 User 04 cumulative distribution function for genioplasty.

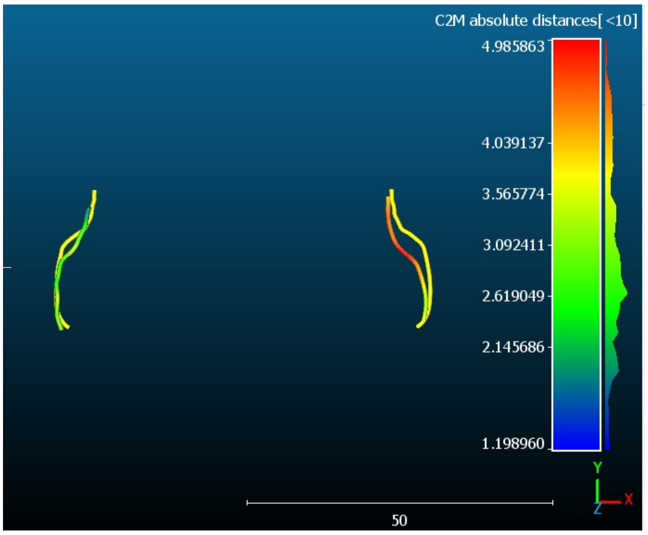


Figure 155 User 04 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for mandibular advancement.

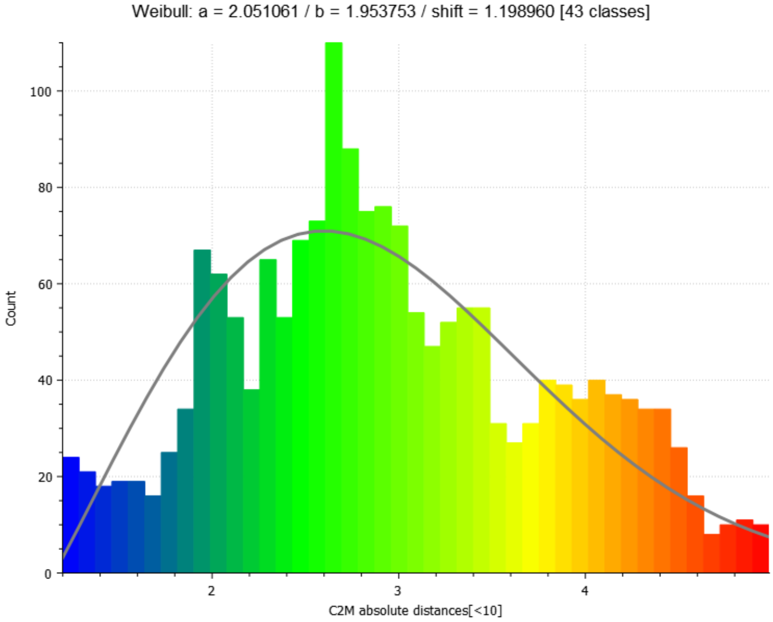


Figure 156 User 04 cloud-to-mesh absolute distance histogram and Weibull function for mandibular advancement.

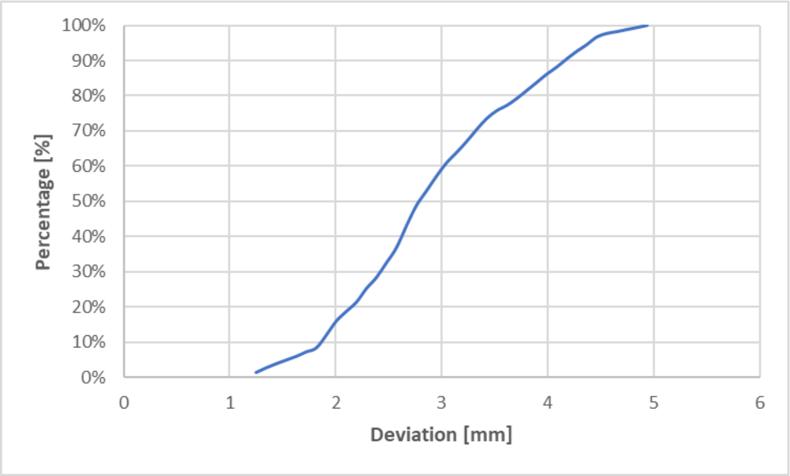


Figure 157 User 04 cumulative distribution function for mandibular advancement.

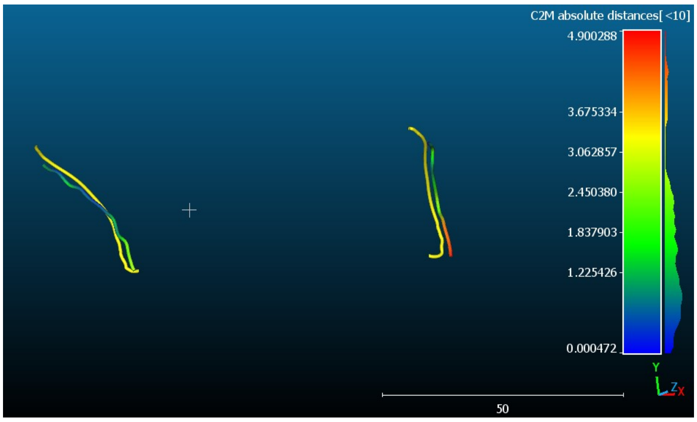


Figure 158 User 04 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for mandibular angle osteotomy.

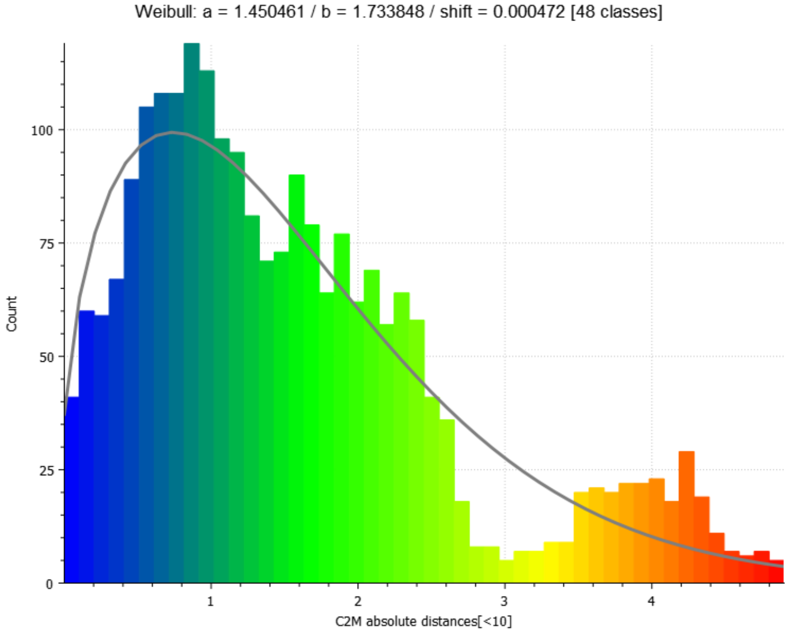


Figure 159 User 04 cloud-to-mesh absolute distance histogram and Weibull function for mandibular angles osteotomy.

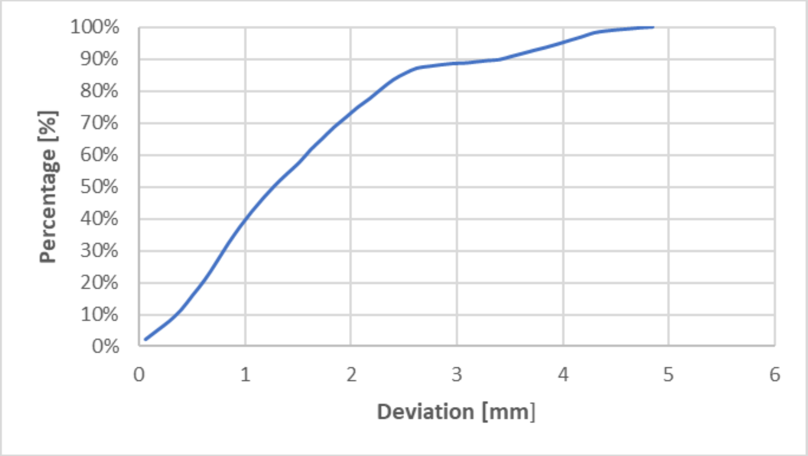


Figure 160 User 04 cumulative distribution function for mandibular angle osteotomy.

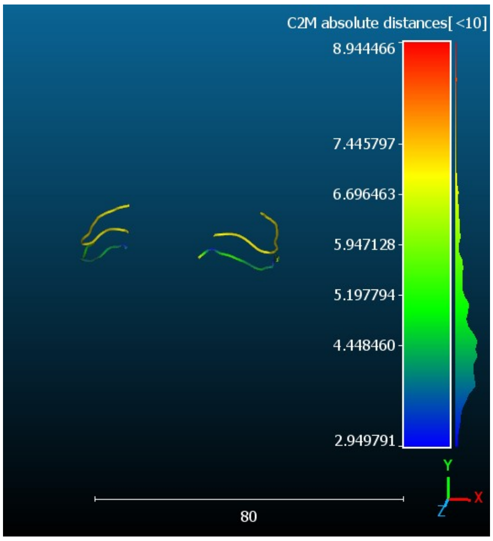


Figure 161 User 05 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines maxillary cutting lines.

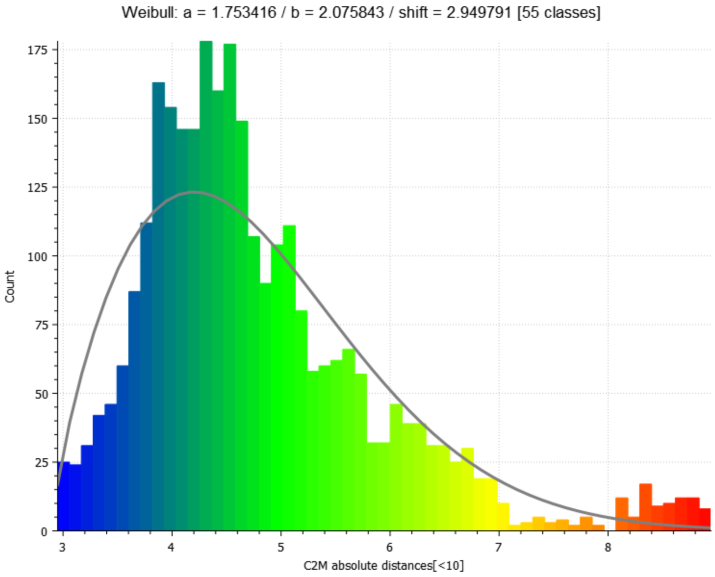


Figure 162 User 05 cloud-to-mesh absolute distance histogram and Weibull function for maxillary cutting lines.

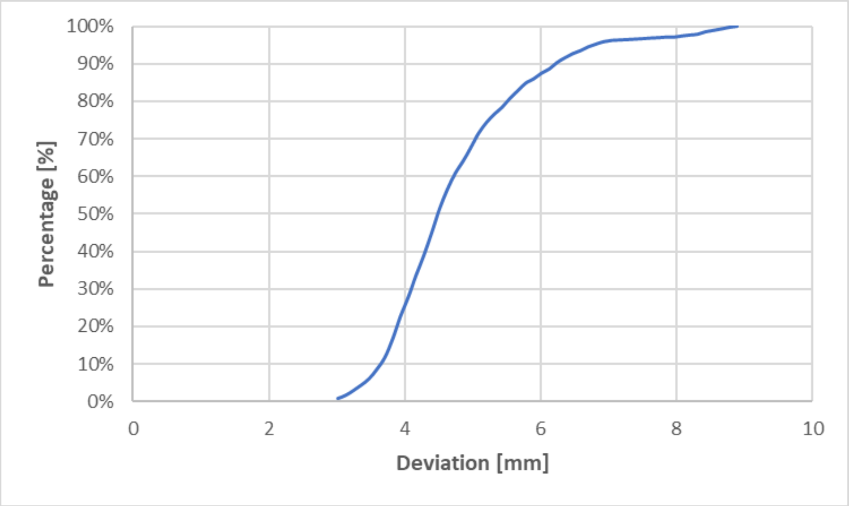


Figure 163 User 05 cumulative distribution function for maxillary cutting lines.

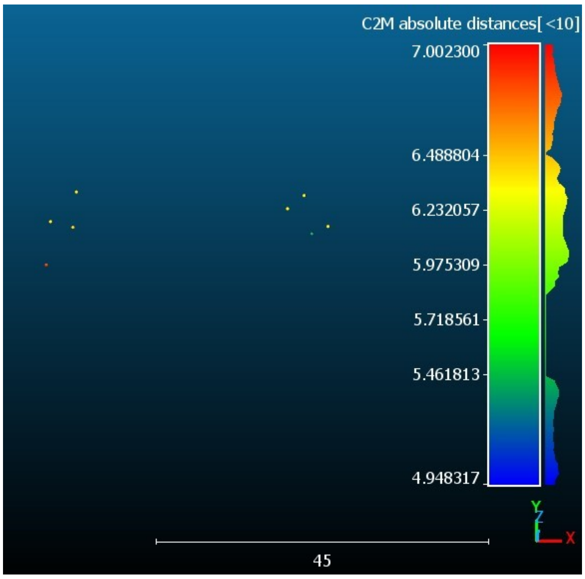


Figure 164 User 05 mapping of cloud-to-mesh absolute distance and nominal (yellow) drilled holes.

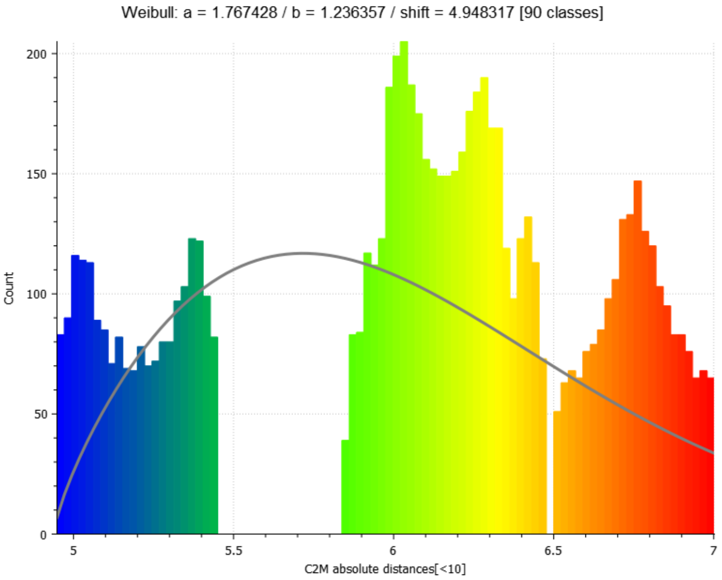


Figure 165 User 05 cloud-to-mesh absolute distance histogram and Weibull function for drilled holes.

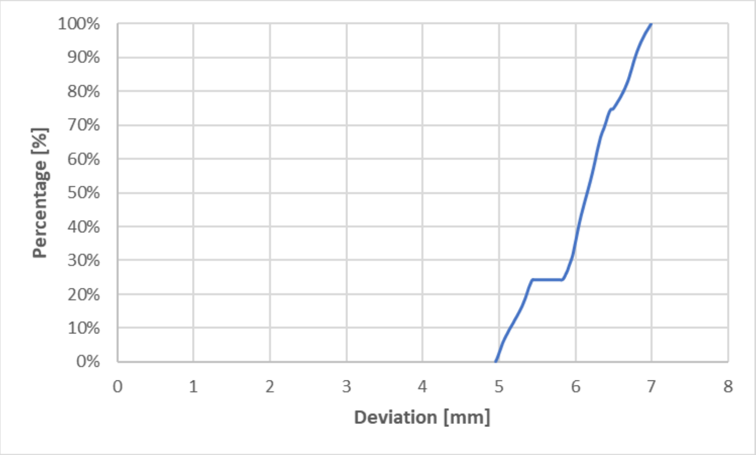


Figure 166 User 05 cumulative distribution function for drilled holes.

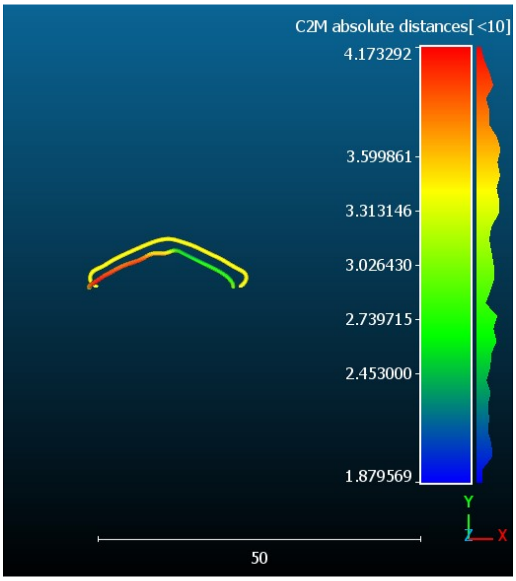


Figure 167 User 05 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for genioplasty.

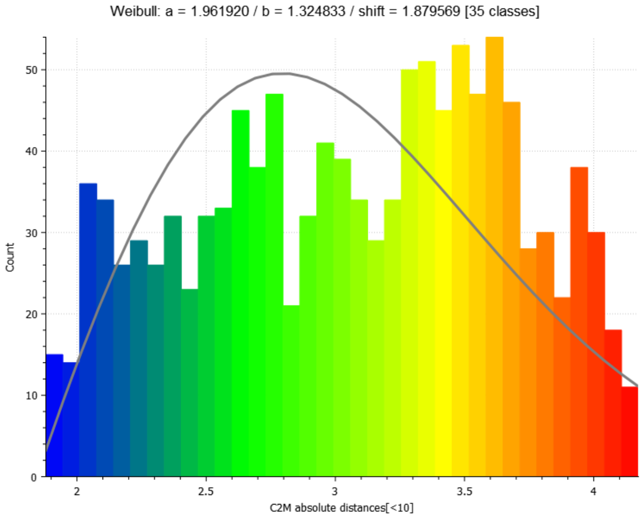


Figure 168 User 05 cloud-to-mesh absolute distance histogram and Weibull function for genioplasty.

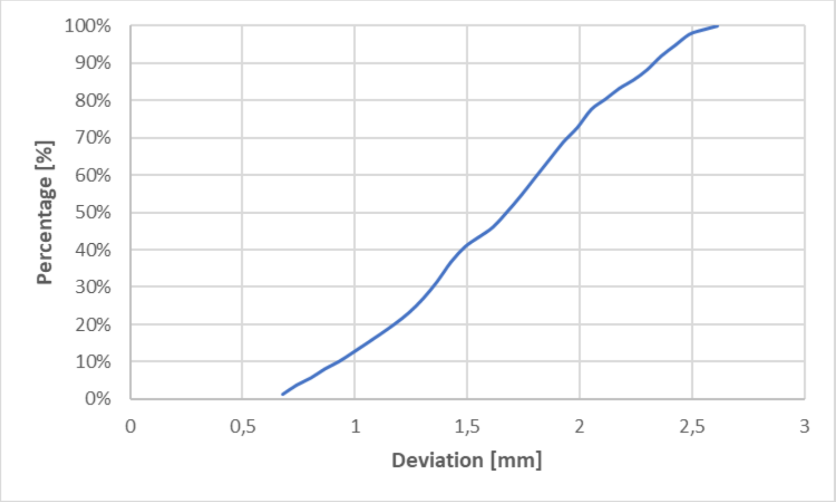


Figure 169 User 05 cumulative distribution function for genioplasty.

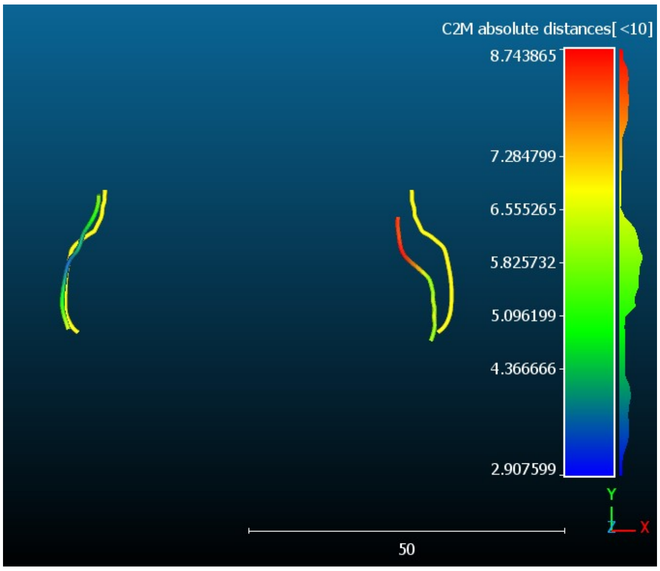


Figure 170 User 05 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for mandibular advancement.

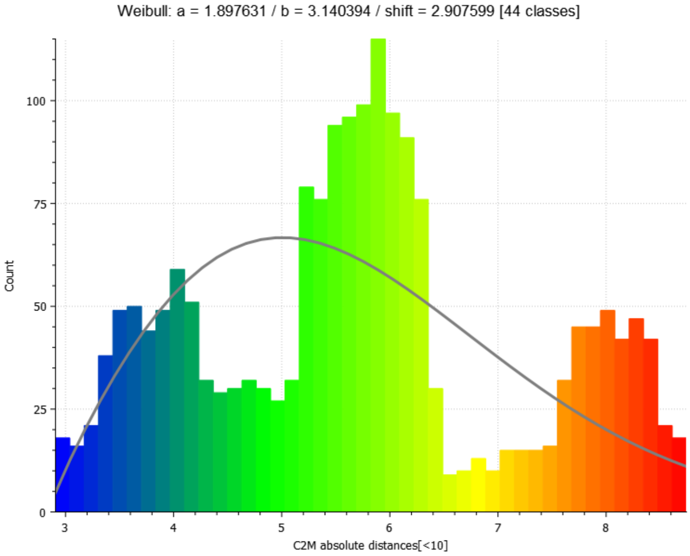


Figure 171 User 05 cloud-to-mesh absolute distance histogram and Weibull function for mandibular advancement.

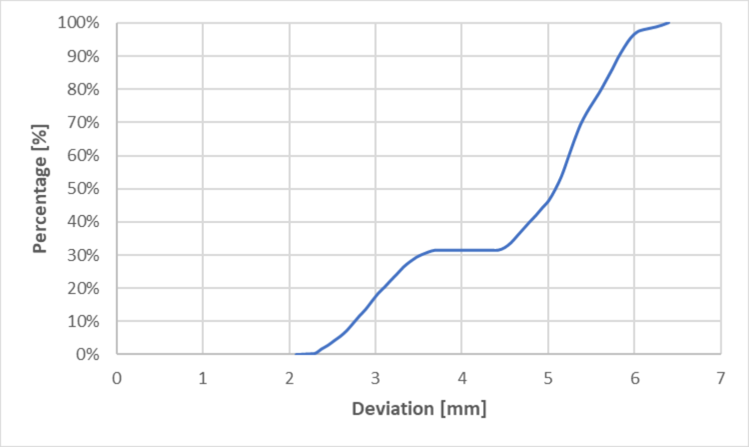


Figure 172 User 05 cumulative distribution function for mandibular advancement.

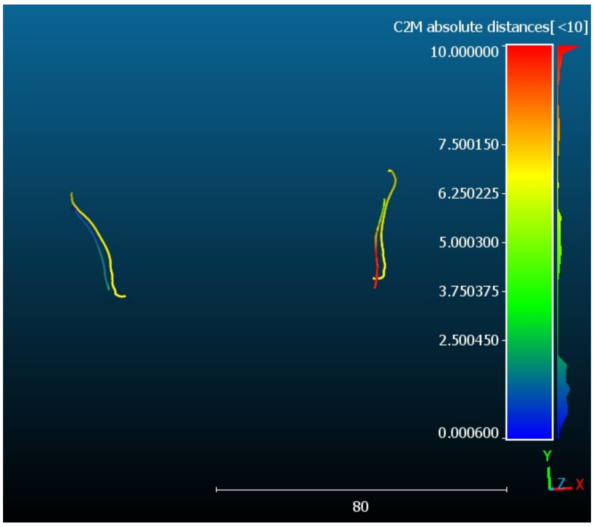


Figure 173 User 05 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for mandibular angle osteotomy.

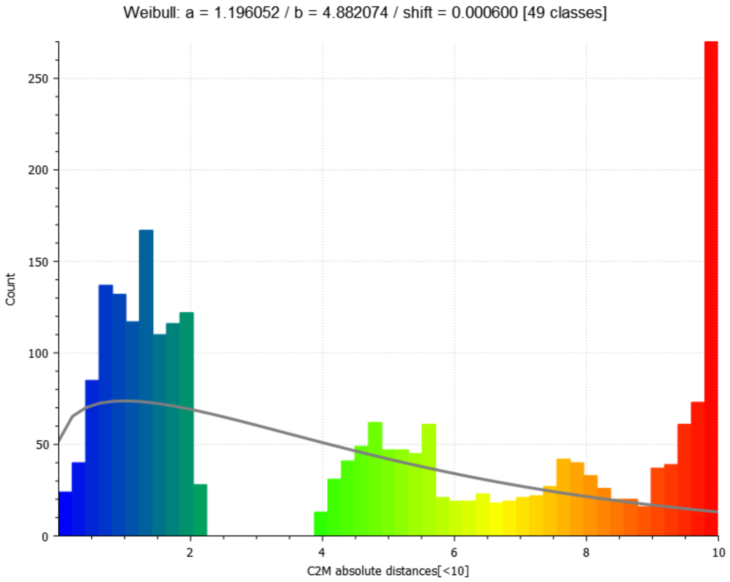


Figure 174 User 05 cloud-to-mesh absolute distance histogram and Weibull function for mandibular angles osteotomy.

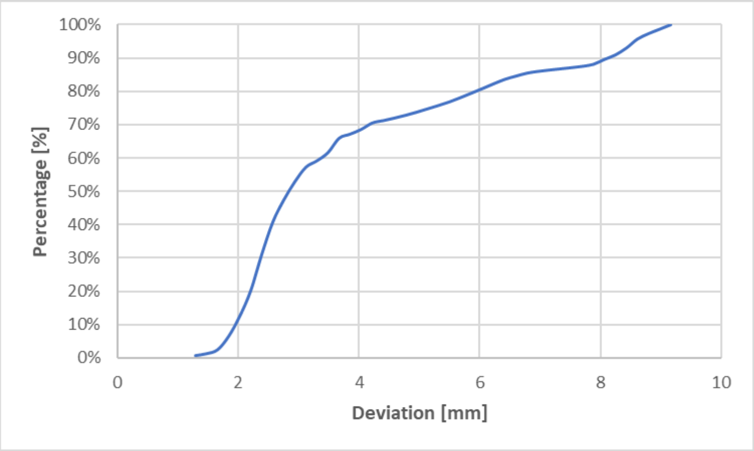


Figure 175 User 05 cumulative distribution function for mandibular angle osteotomy.

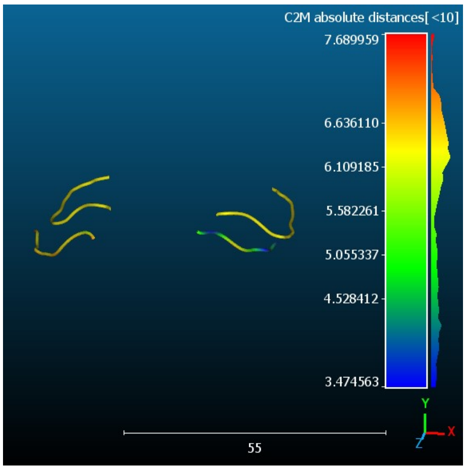


Figure 176 User 06 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines maxillary cutting lines.

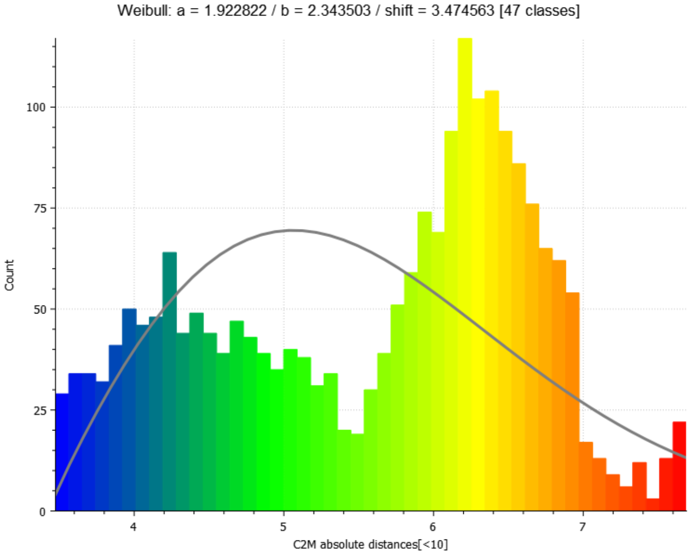


Figure 177 User 06 cloud-to-mesh absolute distance histogram and Weibull function for maxillary cutting lines.

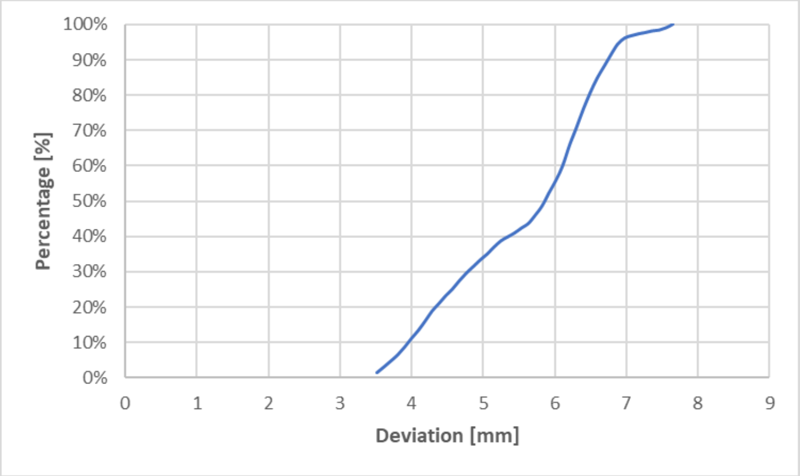


Figure 178 User 06 cumulative distribution function for maxillary cutting lines.

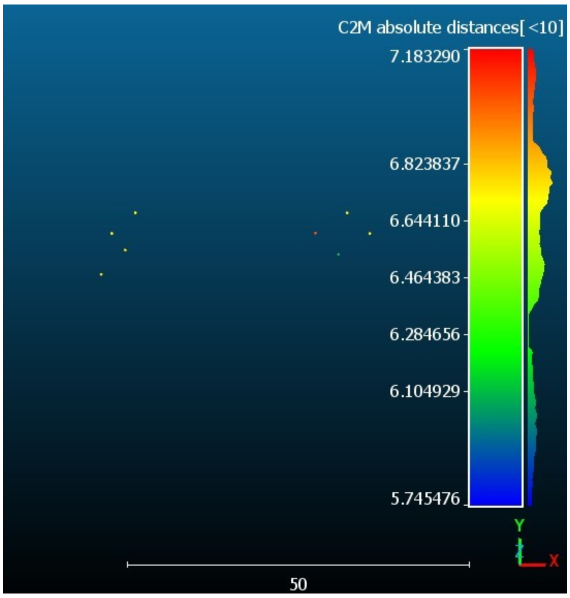


Figure 179 User 06 mapping of cloud-to-mesh absolute distance and nominal (yellow) drilled holes.

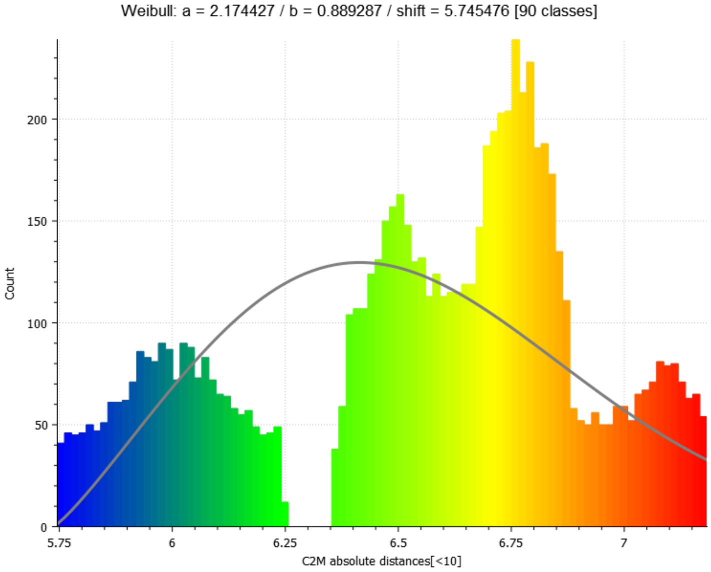


Figure 180 User 06 cloud-to-mesh absolute distance histogram and Weibull function for drilled holes.

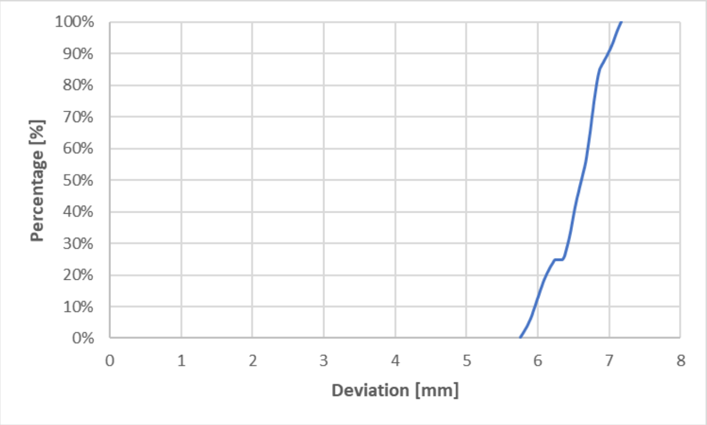


Figure 181 User 06 cumulative distribution function for drilled holes.

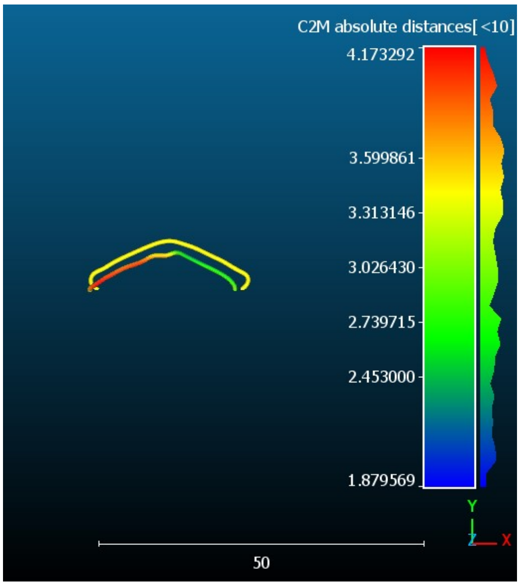


Figure 182 User 06 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for genioplasty.

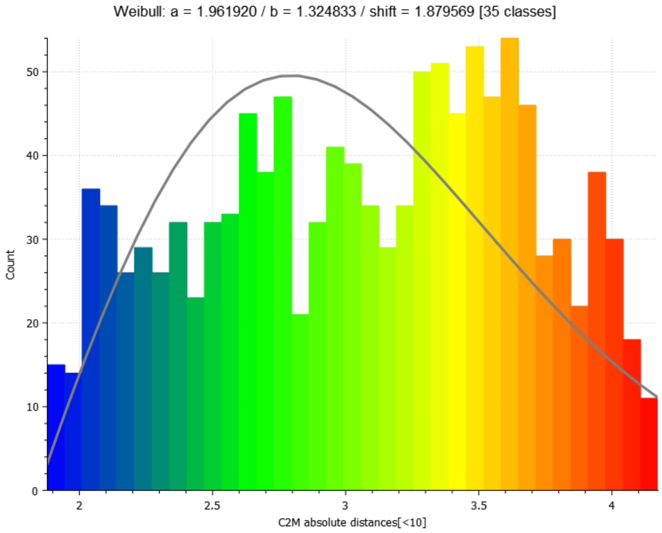


Figure 183 User 06 cloud-to-mesh absolute distance histogram and Weibull function for genioplasty.

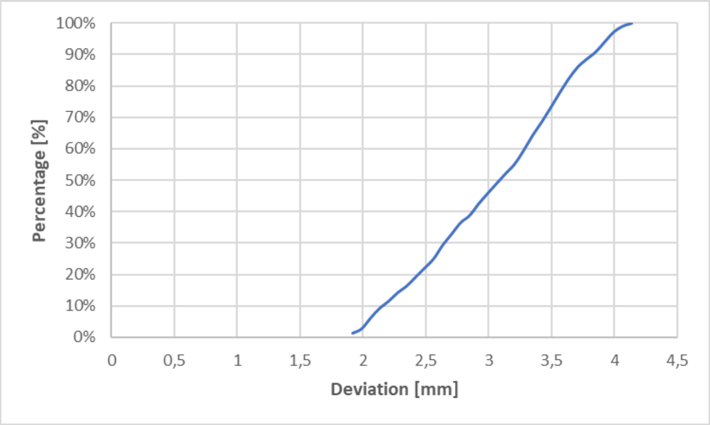


Figure 184 User 06 cumulative distribution function for genioplasty.

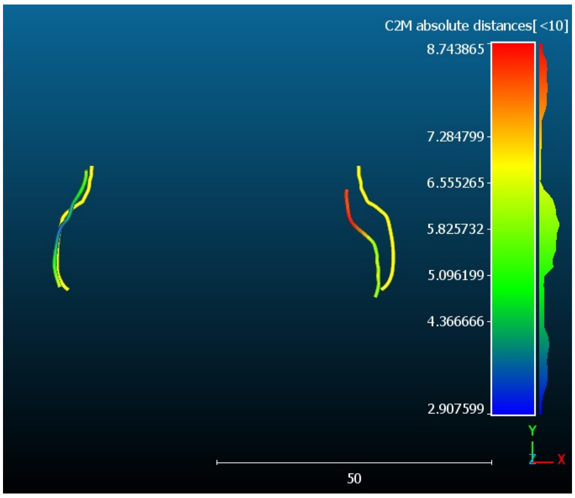


Figure 185 User 06 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for mandibular advancement.

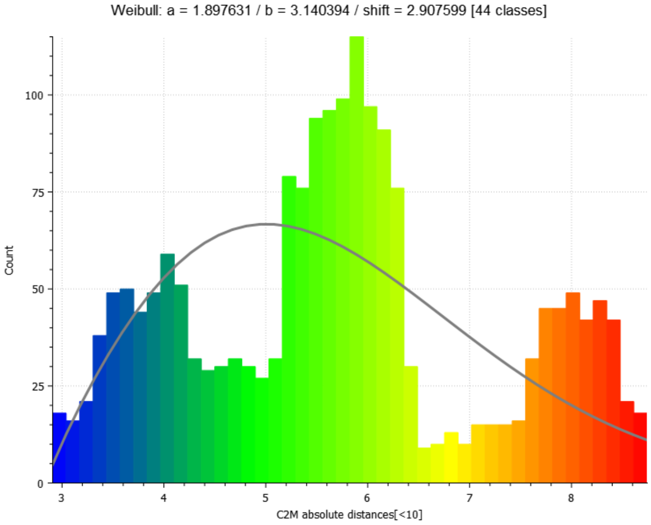


Figure 186 User 06 cloud-to-mesh absolute distance histogram and Weibull function for mandibular advancement.

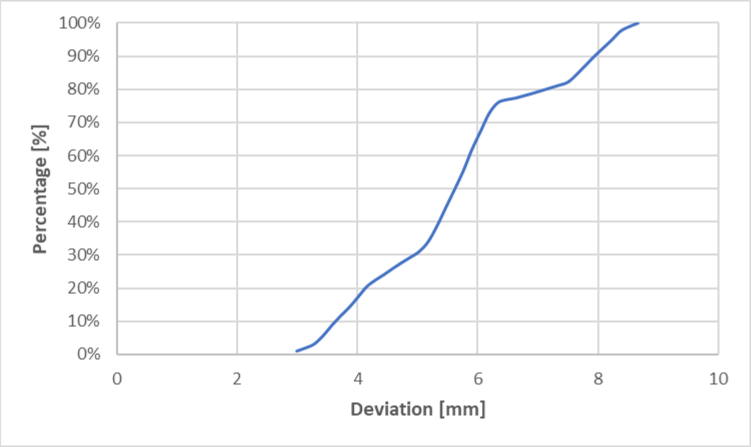


Figure 187 User 06 cumulative distribution function for mandibular advancement.

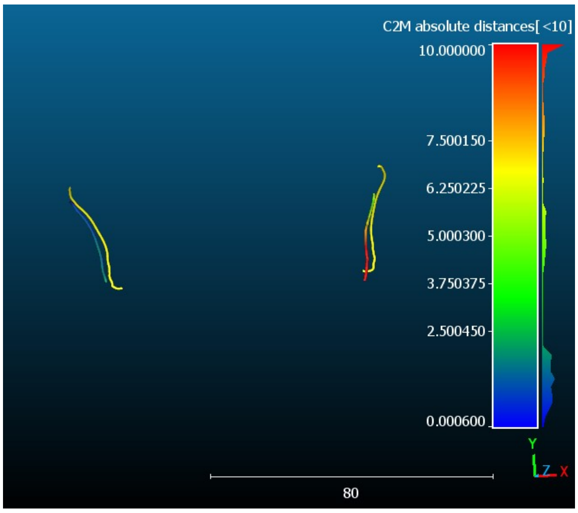


Figure 188 User 06 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for mandibular angle osteotomy.

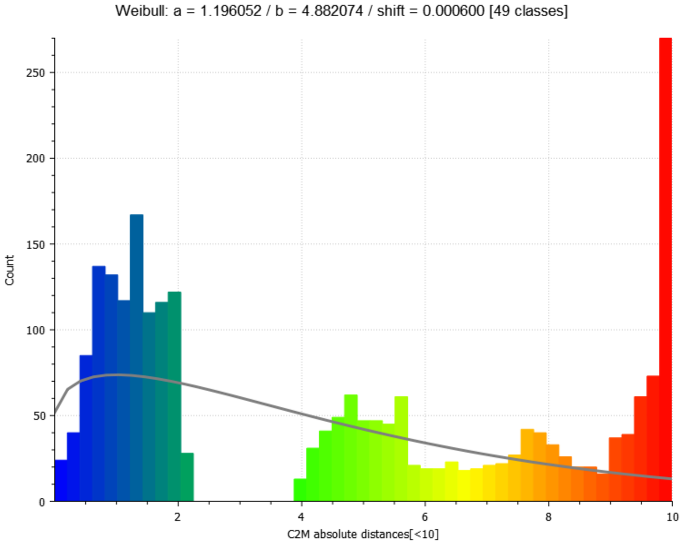


Figure 189 User 06 cloud-to-mesh absolute distance histogram and Weibull function for mandibular angles osteotomy.

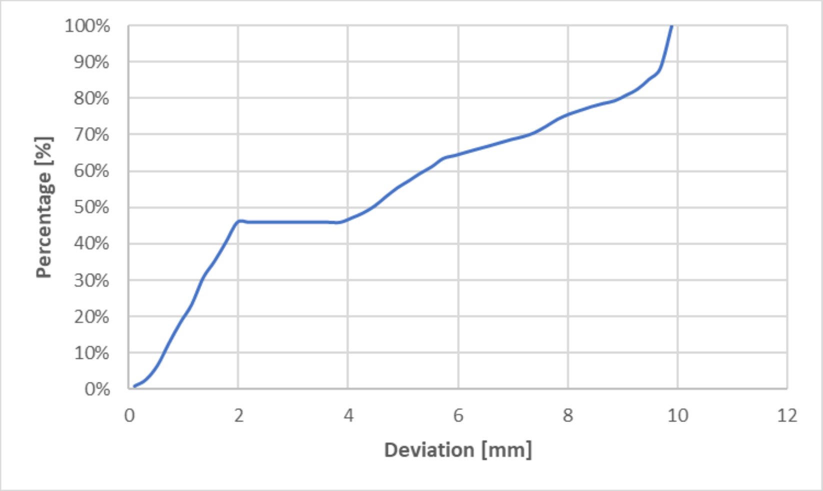


Figure 190 User 06 cumulative distribution function for mandibular angle osteotomy.

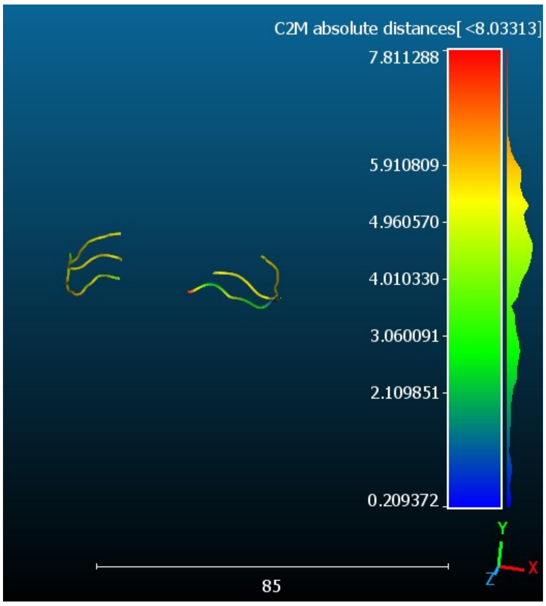


Figure 191 User 07 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines maxillary cutting lines.

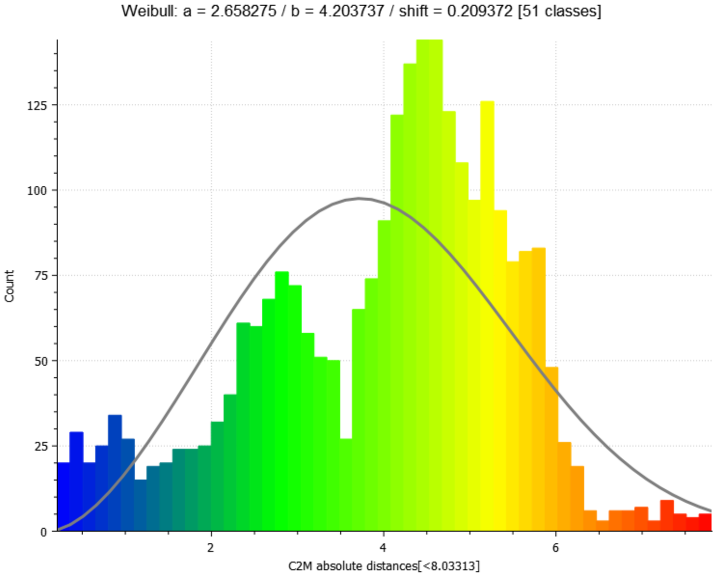


Figure 192 User 07 cloud-to-mesh absolute distance histogram and Weibull function for maxillary cutting lines.

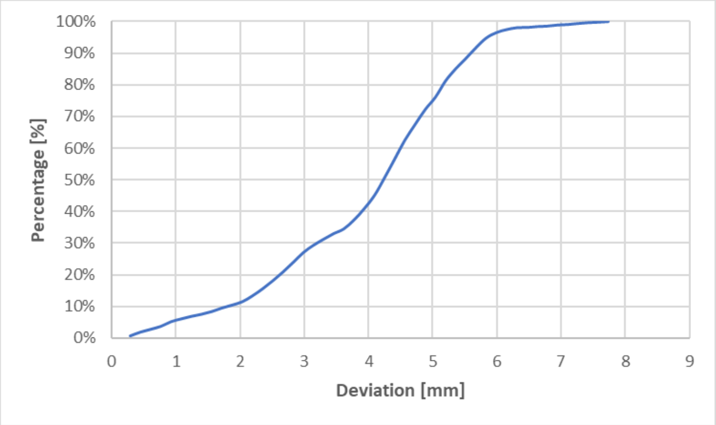


Figure 193 User 07 cumulative distribution function for maxillary cutting lines.

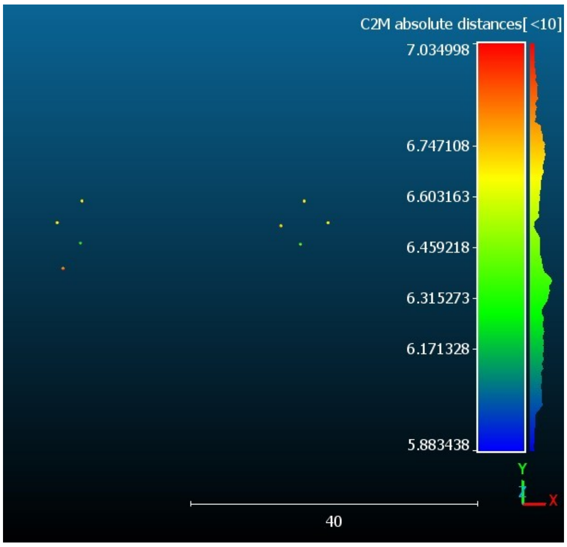


Figure 194 User 07 mapping of cloud-to-mesh absolute distance and nominal (yellow) drilled holes.

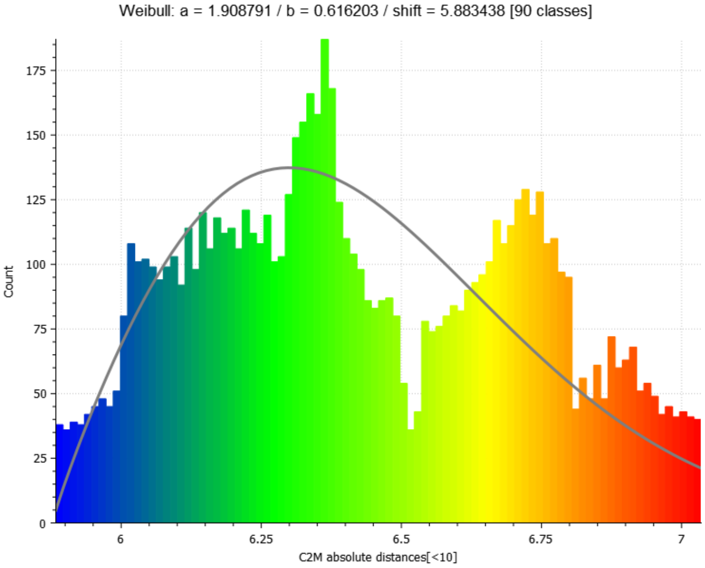


Figure 195 User 07 cloud-to-mesh absolute distance histogram and Weibull function for drilled holes.

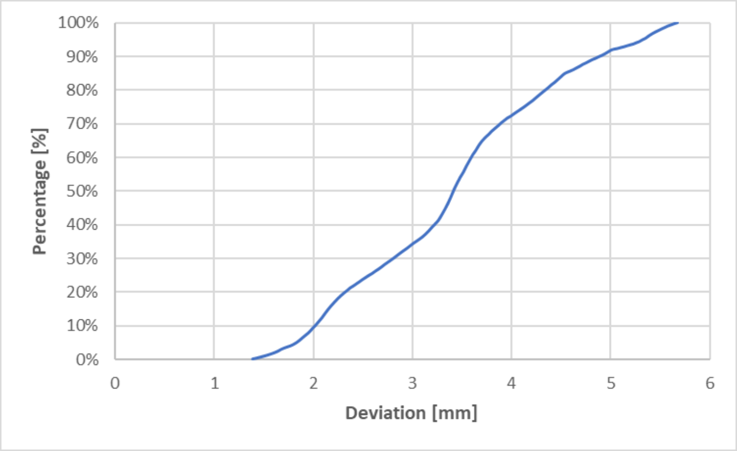


Figure 196 User 07 cumulative distribution function for drilled holes.

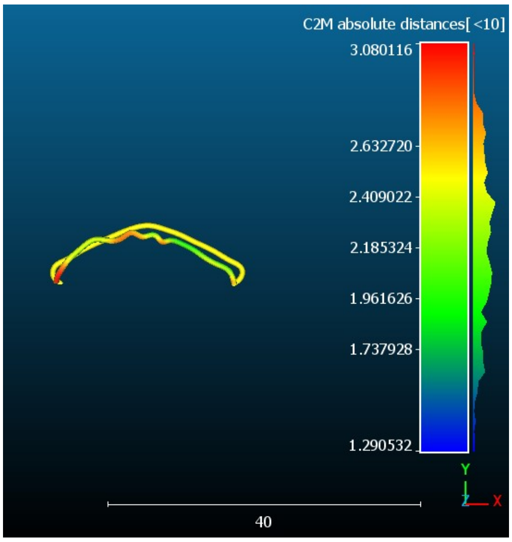


Figure 197 User 07 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for genioplasty.

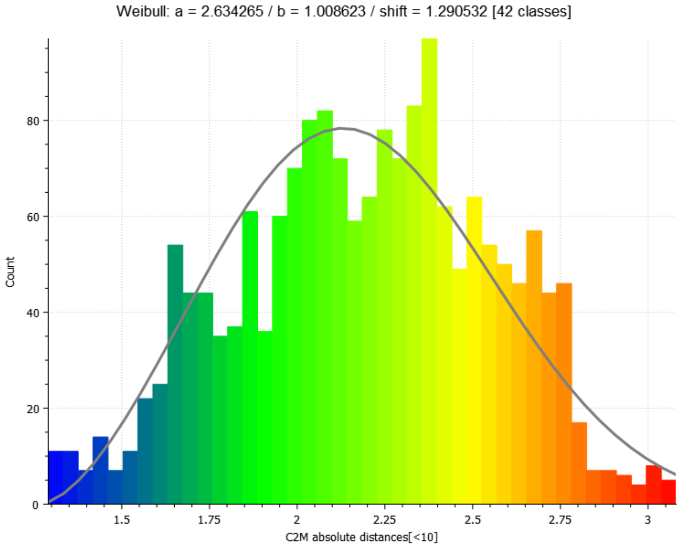


Figure 198 User 07 cloud-to-mesh absolute distance histogram and Weibull function for genioplasty.

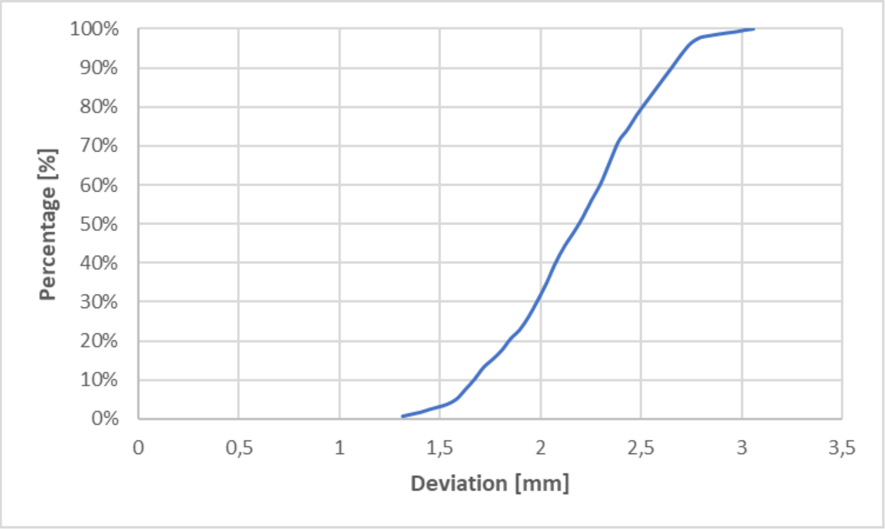


Figure 199 User 07 cumulative distribution function for genioplasty.

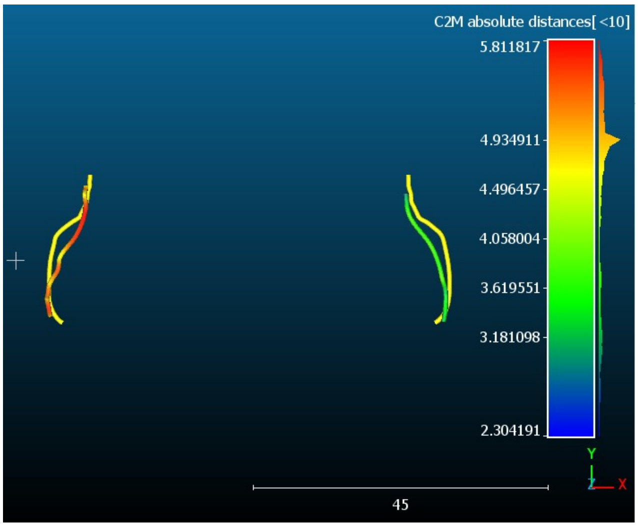


Figure 200 User 07 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for mandibular advancement.

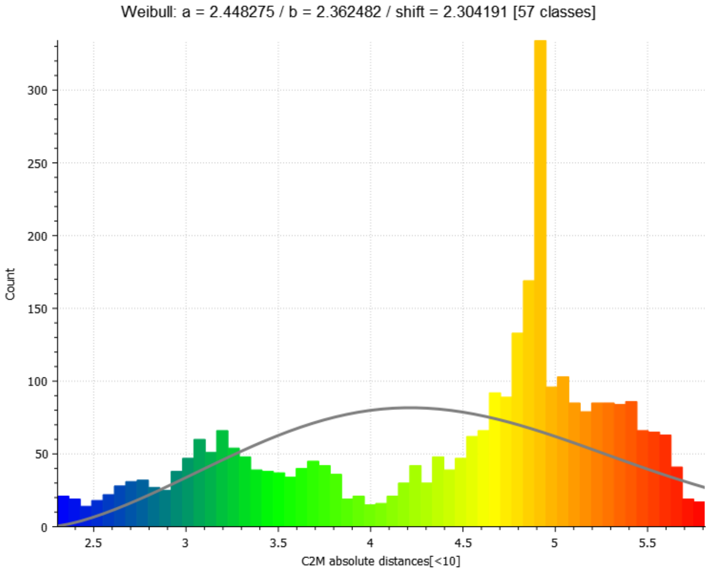


Figure 201 User 07 cloud-to-mesh absolute distance histogram and Weibull function for mandibular advancement.

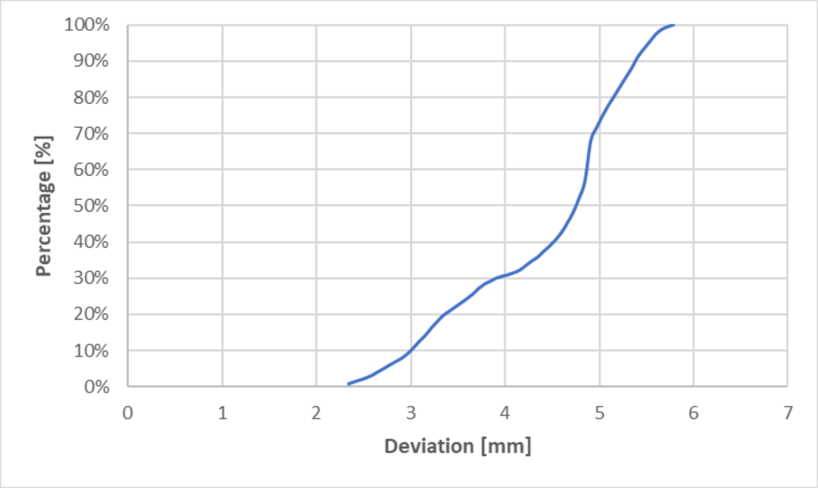


Figure 202 User 07 cumulative distribution function for mandibular advancement.

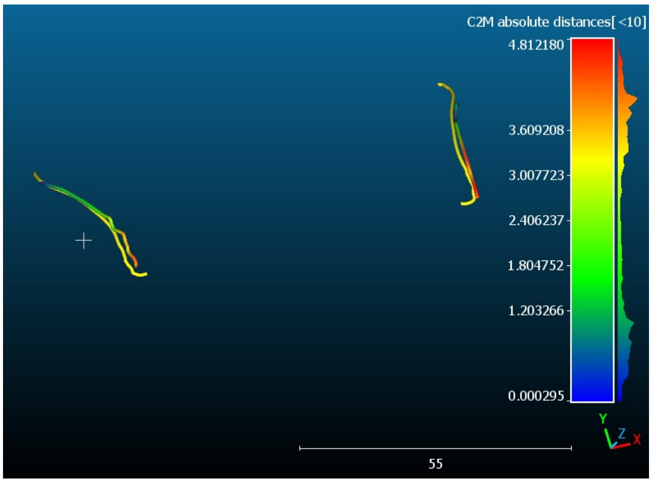


Figure 203 User 07 mapping of cloud-to-mesh absolute distance and nominal (yellow) lines for mandibular angle osteotomy.

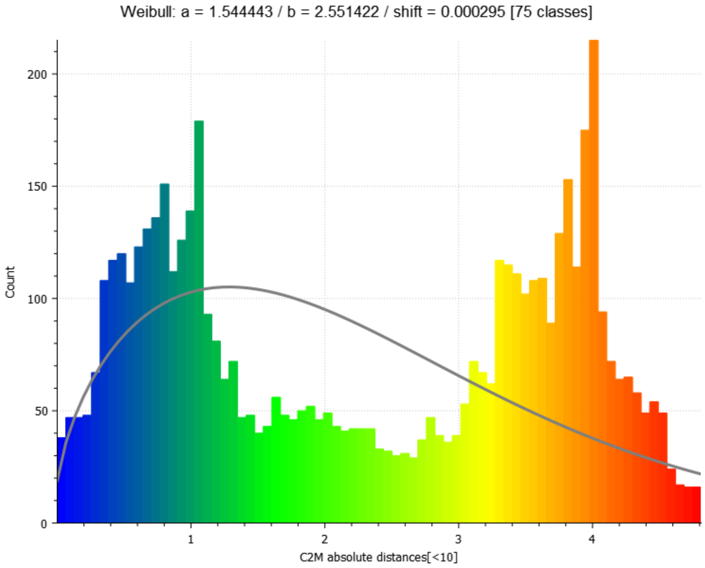


Figure 204 User 07 cloud-to-mesh absolute distance histogram and Weibull function for mandibular angles osteotomy.

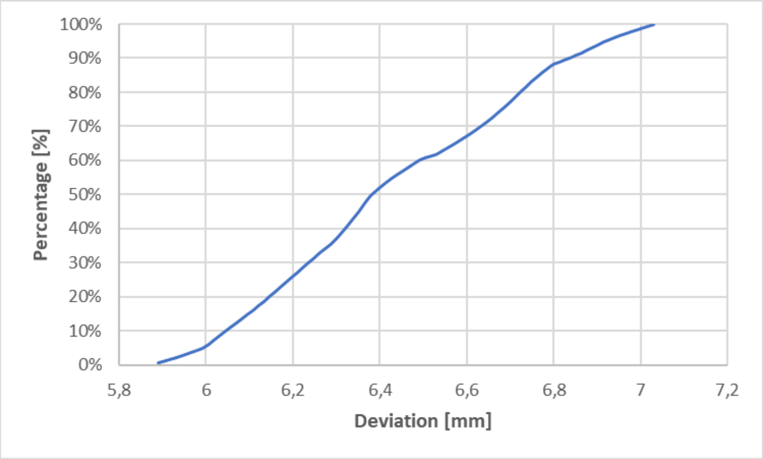


Figure 205 User 07 cumulative distribution function for mandibular angle osteotomy.

9. Bibliography

[1] Danieli Andrea, Design and Development of an Augmented Reality Application deployed in Microsoft HoloLens V2 for Maxillary and Mandibular Osteotomy and Repositioning

[2] Yu GY. Oral and maxillofacial surgery: Current and future. Ann Maxillofac Surg. 2013 Jul;3(2):111-2. doi: 10.4103/2231-0746.119209. PMID: 24205468; PMCID: PMC3814657.

[3] Dimitroulis G. Management of temporomandibular joint disorders: A surgeon's perspective. Aust Dent J. 2018 Mar;63 Suppl 1:S79-S90. doi: 10.1111/adj.12593. PMID: 29574810.

[4] Mohammad S. Dental implants. Natl J Maxillofac Surg. 2017 Jul-Dec;8(2):87-88. doi: 10.4103/njms.NJMS_79_17. PMID: 29386808; PMCID: PMC5773996.

[5] Ebenezer, S., Kumar, V.V., Thor, A. (2021) Basics of Dental Implantology for the Oral Surgeon. In: Bonanthaya, K., Panneerselvam, E., Manuel, S., Kumar, V.V., Rai, A. (eds) Oral and Maxillofacial Surgery for the Clinician. Springer, Singapore.

[6] Nahai FR, Williams JK, Burstein FD, Martin J, Thomas J. The Management of Cleft Lip and Palate: Pathways for Treatment and Longitudinal Assessment. Semin Plast Surg. 2005 Nov;19(4):275–85. doi: 10.1055/s-2005-925900. PMCID: PMC2884751.

[7] Cambridge University Hospitals.

[8] H. Kashani and L. Rasmusson, ‘Osteotomies in Orthognathic Surgery’, A Textbook of Advanced Oral and Maxillofacial Surgery Volume 3. InTech, Aug. 31, 2016. doi: 10.5772/63345.

[9] Zoabi, A.; Redenski, I.; Oren, D.; Kasem, A.; Zigron, A.; Daoud, S.; Moskovich, L.; Kablan, F.; Srouji, S. 3D Printing and Virtual Surgical Planning in Oral and Maxillofacial Surgery. J. Clin. Med. 2022, 11, 2385. https://doi.org/10.3390/jcm11092385

[10] Beek, DM., Baan, F., Liebregts, J. et al. A learning curve in 3D virtual surgical planned orthognathic surgery. Clin Oral Invest 27, 3907–3915 (2023). https://doi.org/10.1007/s00784-023-05013-2

[11] Goulart ME, Biegelmeyer TC, Moreira-Souza L, Adami CR, Deon F, Flores IL, Gamba TO. What is the accuracy of the surgical guide in the planning

of orthognathic surgeries? A systematic review. Med Oral Patol Oral Cir Bucal. 2022 Mar 1;27(2):e125-e134. doi: 10.4317/medoral.25042. PMID: 34564681; PMCID: PMC8898580.

[12] Vale F, Scherzberg J, Cavaleiro J, Sanz D, Caramelo F, Maló L, Marcelino JP. 3D virtual planning in orthognathic surgery and CAD/CAM surgical splints generation in one patient with craniofacial microsomia: a case report. Dental Press J Orthod. 2016 Jan-Feb;21(1):89-100. doi: 10.1590/2177-6709.21.1.089-100.oar. PMID: 27007767; PMCID: PMC4816591.

[13] S. H. Kang, H. J. Kim, H. W. Park, and S. H. Lee, "Maxillary cutting guide for executing a simulated osteotomy and removing the bony interference during orthognathic surgery", J. Med. Devices, Trans. ASME, vol. 9, no. 4, pp. 1–4, 2015, doi: 10.1115/1.4031162.

[14] T. Shirota et al., "CAD/CAM splint and surgical navigation allows accurate maxillary segment positioning in Le Fort I osteotomy", Heliyon, vol. 5, no. 7, p. e02123, 2019, doi: 10.1016/j.heliyon.2019.e02123.

[15] P. Pietruski et al., "Supporting mandibular resection with intraoperative navigation utilizing augmented reality technology – A proof of concept study", J. Cranio-Maxillofacial Surg., vol. 47, no. 6, pp. 854–859, 2019, doi: 10.1016/j.jcms.2019.03.004.

[16] Takahiro Kanno, Shintaro Sukegawa, Yoshihiko Furuki, Yoshiki Nariai, Joji Sekine, Overview of innovative advances in bioresorbable plate systems for oral and maxillofacial surgery, Japanese Dental Science Review, Volume 54, Issue 3, 2018, Pages 127-138, ISSN 1882-7616, https://doi.org/10.1016/j.jdsr.2018.03.003.

[17] Milgram, P. and Kishino, F. (1994) A Taxonomy of Mixed Reality Visual Displays. IEICE Transactions on Information and Systems, E77-D, 1321-1329. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.102.4646

[18] Rebbani, Z., Azougagh, D., Bahatti, L., & Bouattane, O. (2021). Definitions and applications of augmented/virtual reality: A survey. Int. J, 9.

[19] Nguyen Richard, Gouin-Vallerand Charles, Amiri Maryam. and interaction designs in mixed and augmented reality head mounted display: a scoping review and classification. Frontiers in Virtual Reality, Vol. 4, 2023. https://www.frontiersin.org/articles/10.3389/frvir.2023.1171230 doi: 10.3389/frvir.2023.1171230

[20] Rokhsaritalemi, S.; Sadeghi-Niaraki, A.; Choi, S.-M. A Review on Mixed Reality: Current Trends, Challenges and Prospects. Appl. Sci. 2020, 10, 636. https://doi.org/10.3390/app10020636.

[21] Ohlenburg, J., Broll, W., and Lindt, I. (2007). DEVALa device abstraction layer for VR/AR. In Universal Acess in Human Computer Interaction. Coping with Diversity, page 497-506. Springer.

[22] Nakamoto, T. and Minh, H. P. D. (2007). Improvement of olfactory display using solenoid valves. In Virtual Reality Conference, 2007. VR'07. IEEE, pages 179-186. IEEE.

[23] Maynes-Aminzade, D. (2005). Edible bits: Seamless interfaces between people, data and food. In Conference on Human Factors in Computing Systems (CHI'05)- Extended Abstracts, pages 22072210. Citeseer.

[24] Narumi, T., Nishizaka, S., Kajinami, T., Tanikawa, T., and Hirose, M. (2011). Augmented reality flavours: Gustatory display based on edible marker and cross-modal interaction. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11, pages 93102, New York, NY, USA. ACM.

[25] Simões, R.J., Raposo, C., Barreto, J., Edwards, P.J., & Stoyanov, D. (2018). Visual Tracking vs Optical Tracking in Computer-Assisted Intervention.

[26] P. Lang, A. Kusej, A. Pinz and G. Brasseur, "Inertial tracking for mobile augmented reality," IMTC/2002. Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference (IEEE Cat. No.00CH37276), Anchorage, AK, USA, 2002, pp. 1583-1587 vol.2, doi: 10.1109/IMTC.2002.1007196.

[27] Afif, Vision-based Tracking Technology for Augmented Reality: A Survey; Rabbi, A Survey on Augmented Reality Challenges and Tracking.

[28] Dongdong Weng, Dewen Cheng, Yongtian Wang, Yue Liu, Display systems and registration methods for augmented reality applications, Optik, Volume 123, Issue 9, 2012, Pages 769-774, ISSN 0030-4026, https://doi.org/10.1016/j.ijleo.2011.05.033

[29] Jihyung Kim, Seung-won Oh, Jinyoung Choi, Sungeun Park & Wooksung Kim (2022) Optical see-through head-mounted display including transmittance-variable display for high visibility, Journal of Information Display, 23:2, 121-127, DOI: 10.1080/15980316.2021.2018058.

[30] Birlo M, Edwards PJE, Clarkson M, Stoyanov D. Utility of optical see-through head mounted displays in augmented reality-assisted surgery: A systematic review. Med Image Anal. 2022 Apr;77:102361. doi: 10.1016/j.media.2022.102361. Epub 2022 Jan 12. PMID: 35168103; PMCID: PMC10466024.

[31] Carbone, M.; Cutolo, F.; Condino, S.; Cercenelli, L.; D'Amato, R.; Badiali, G.; Ferrari, V. Architecture of a Hybrid Video/Optical See-through Head-Mounted Display-Based Augmented Reality Surgical Navigation Platform. Information 2022, 13, 81. https://doi.org/10.3390/info13020081.

[32] Kiyokawa, Optical see-through HMDs make use of a transparent optical combiner to merge virtual content, projected into the field of view of the wearer, with a view of the real world.

[33] Doughty M, Ghugre NR, Wright GA. Augmenting Performance: A Systematic Review of Optical See-Through Head-Mounted Displays in Surgery. J Imaging. 2022 Jul 20;8(7):203. doi: 10.3390/jimaging8070203. PMID: 35877647; PMCID: PMC9318659.

[34] https://www.microsoft.com/it-it/hololens/hardware

[35] https://www.magicleap.com/magic-leap-2

[36] https://varjo.com/products/xr-3/

[37] https://support.hp.com/it-it/document/c06113255

[38] https://www.acer.com/ns/ja/JP/smart/ah101/

[39] https://www.samsung.com/us/support/computing/hmd/hmd-dyssey/hmd-odyssey-plus-mixed-reality/

[40] https://www.asus.com/it/news/lwa6edupmjxd7kji/

[41] Stackhouse AA, Rafi D, Walls R, Dodd RV, Badger K, Davies DJ, Brown CA, Cowell A, Meeran K, Halse O, Kinross J, Lupton M, Hughes EA, Sam AH. Knowledge Attainment and Engagement Among Medical Students: A Comparison of Three Forms of Online Learning. Adv Med Educ Pract. 2023 Apr 20;14:373-380. doi: 10.2147/AMEP.S391816. PMID: 37101695; PMCID: PMC10124742.

[42] Veer V, Phelps C, Moro C. Incorporating Mixed Reality for Knowledge Retention in Physiology, Anatomy, Pathology, and Pharmacology Interdisciplinary Education: A Randomized Controlled Trial. Med Sci Educ. 2022

Sep 23;32(6):1579-1586. doi: 10.1007/s40670-022-01635-5. PMID: 36532413; PMCID: PMC9755411.

[43] Pregowska, A.; Osial, M.; Dolega-Dolegowski, D.; Kolecki, R.; Proniewska, K. Information and Communication Technologies Combined with Mixed Reality as Supporting Tools in Medical Education. Electronics 2022, 11, 3778. https://doi.org/10.3390/electronics11223778.

[44] Otero-Varela L, Cintora AM, Espinosa S, Redondo M, Uzuriaga M, González M, García M, Naldrett J, Alonso J, Vazquez T, Blanco A, Cardós Alonso MDC. Extended reality as a training method for medical first responders in mass casualty incidents: A protocol for a systematic review. PLoS One. 2023 Mar 23;18(3):e0282698. doi: 10.1371/journal.pone.0282698. PMID: 36952495; PMCID: PMC10035843.

[45] Baetzner AS, Wespi R, Hill Y, Gyllencreutz L, Sauter TC, Saveman BI, Mohr S, Regal G, Wrzus C, Frenkel MO. Preparing medical first responders for crises: a systematic literature review of disaster training programs and their effectiveness. Scand J Trauma Resusc Emerg Med. 2022 Dec 24;30(1):76. doi: 10.1186/s13049-022-01056-8. PMID: 36566227; PMCID: PMC9789518.

[46] Pons Patricia, Navas-Medrano Samuel, Soler-Dominguez Jose L. Extended reality for mental health: Current trends and future challenges. Frontiers in Computer Science. Vol. 4. 2022. https://www.frontiersin.org/articles/10.3389/fcomp.2022.1034307 10.3389/fcomp.2022.1034307

[47] Annaswamy, T.M., Bahirat, K., Raval, G. et al. Clinical feasibility and preliminary outcomes of a novel mixed reality system to manage phantom pain: a pilot study. Pilot Feasibility Stud 8, 232 (2022). https://doi-org.ezproxy.cad.univpm.it/10.1186/s40814-022-01187-w

[48] Lau, I.; Gupta, A.; Ihdayhid, A.; Sun, Z. Clinical Applications of Mixed Reality and 3D Printing in Congenital Heart Disease. Biomolecules 2022, 12, 1548. https://doi.org/10.3390/biom12111548

[49] Wei P, Yao Q, Xu Y, Zhang H, Gu Y, Wang L. Percutaneous kyphoplasty assisted with/without mixed reality technology in treatment of OVCF with IVC: a prospective study. J Orthop Surg Res. 2019;14(1):1-9. https://doi.org/10.1186/s13018-019-1303-x

[50] Liu X, Sun J, Zheng M, Cui X. Application of mixed reality using optical see-through head-mounted displays in transforaminal percutaneous endoscopic lumbar discectomy. BioMed Res Int. 2021;2021: 9717184-9717188. https://doi.org/10.1155/2021/9717184

[51] Gu Y, Yao Q, Xu Y, Zhang H, Wei P, Wang L. A clinical application study of mixed reality technology assisted lumbar pedicle screws implantation. Med Sci Monit. 2020;26:e924982-1–e924982-12. https://doi.org/10.1002/central/CN-02193447/full

[52] Gregory TM, Gregory J, Sledge J, Allard R, Mir O. Surgery guided by mixed reality: presentation of a proof of concept. Acta Orthop. 2018;89(5):480-483. https://doi.org/10.1080/17453674. 2018.1506974.

[53] Lei PF, Su SL, Kong LY, Wang CG, Zhong D, Hu YH. Mixed Reality Combined with Three-Dimensional Printing Technology in Total Hip Arthroplasty: An Updated Review with a Preliminary Case Presentation. Orthop Surg. 2019 Oct;11(5):914-920. doi: 10.1111/os.12537. PMID: 31663276; PMCID: PMC6819179.

[54] Liu S, Wang H, Wang C, Zhang W, Chen G, Hu X. Intra-operative 3D hologram support with mixed reality technique based on CT- MRI fusion images: operation guidance for brain brachytherapy. J Contemp Brachytherapy. 2021;13(2):205-210. https://doi.org/10. 5114/jcb.2021.105289

[55] Li Y, Chen X, Wang N, et al. A wearable mixed-reality holographic computer for guiding external ventricular drain insertion at the bedside. J Neurosurg. 2018;1–8. [55] Iizuka K, Sato Y, Imaizumi Y, Mizutani T. Potential efficacy of multimodal mixed reality in epilepsy surgery. Oper Neurosurg (Hagerstown, Md). 2021;20(3):276-281. https://doi.org/10.1093/ ons/opaa341

[56] Iizuka K, Sato Y, Imaizumi Y, Mizutani T. Potential Efficacy of Multimodal Mixed Reality in Epilepsy Surgery. Oper Neurosurg (Hagerstown). 2021 Feb 16;20(3):276-281. doi: 10.1093/ons/opaa341. PMID: 33382064.

[57] Kitagawa M, Sugimoto M, Haruta H, Umezawa A, Kurokawa Y. Intraoperative holography navigation using a mixed-reality wearable computer during laparoscopic cholecystectomy. Surgery. 2021; 171(4):1006-1013. https://doi.org/10.1016/j.surg.2021.10.004

[58] Aoki T, Koizumi T, Sugimoto M, Murakami M. Holography-guided percutaneous puncture technique for selective near-infrared fluorescence-guided

laparoscopic liver resection using mixed- reality wearable spatial computer. Surg Oncol. 2020;35:476-477. https://doi.org/10.1016/j.suronc.2020.10.013

[59] Wierzbicki R, Pawłowicz M, Job J, et al. 3D mixed-reality visualization of medical imaging data as a supporting tool for innovative, minimally invasive surgery for gastrointestinal tumors and systemic treatment as a new path in personalized treatment of advanced cancer diseases. J Cancer Res Clin Oncol. 2021;148(1): 237-243. https://doi.org/10.1007/s00432-021-03680-w

[60] Li G, Dong J, Wang J, et al. The clinical application value of mixed-reality-assisted surgical navigation for laparoscopic nephrectomy. Cancer Med. 2020;9(15):5480-5489. https://doi.org/10.1002/cam4. 3189

[61] Li G, Cao Z, Wang J, et al. Mixed reality models based on low-dose computed tomography technology in nephron-sparing surgery are better than models based on normal-dose computed tomography. Quant Imaging Med Surg. 2021;11(6):2658-2668. https://doi.org/10. 21037/qims-20-956

[62] Porpiglia F, Checcucci E, Amparore D, et al. Percutaneous kidney puncture with three-dimensional mixed-reality hologram guidance: from preoperative planning to intraoperative navigation. Eur Urol. 2021;81(6):588-597.

[63] Nuri T, Mitsuno D, Iwanaga H, Otsuki Y, Ueda K. Application of augmented reality (AR) technology to locate the cutaneous perforator of anterolateral thigh perforator flap: a case report. Microsurgery. 2021;42(1):76-79. https://doi.org/10.1002/micr.30735

[64] Pepe A, Trotta GF, Mohr-Ziak P, Gsaxner C, Wallner J, Bevilacqua V, Egger J. A Marker-Less Registration Approach for Mixed Reality-Aided Maxillofacial Surgery: a Pilot Evaluation. J Digit Imaging. 2019 Dec;32(6):1008-1018. doi: 10.1007/s10278-019-00272-6. PMID: 31485953; PMCID: PMC6841997.

[65] Tang ZN, Hu LH, Soh HY, Yu Y, Zhang WB, Peng X. Accuracy of Mixed Reality Combined With Surgical Navigation Assisted Oral and Maxillofacial Tumor Resection. Front Oncol. 2022 Jan 14;11:715484. doi: 10.3389/fonc.2021.715484. PMID: 35096559; PMCID: PMC8795771.

[66] Sugahara K, Koyachi M, Koyama Y, Sugimoto M, Matsunaga S, Odaka K, Abe S, Katakura A. Mixed reality and three-dimensional printed models for resection of maxillary tumor: a case report. Quant Imaging Med Surg. 2021

May;11(5):2187-2194. doi: 10.21037/qims-20-597. PMID: 33936998; PMCID: PMC8047357.

[67] Liu L, Wang X, Guan M, Fan Y, Yang Z, Li D, Bai Y, Li H. A mixed reality-based navigation method for dental implant navigation method: A pilot study. Comput Biol Med. 2023 Mar;154:106568. doi: 10.1016/j.compbiomed.2023.106568. Epub 2023 Jan 24. PMID: 36739818.

[68] Rahimov CR, Aliyev DU, Rahimov NR, Farzaliyev IM. Mixed Reality in the Reconstruction of Orbital Floor: An Experimental and Clinical Evaluative Study. Ann Maxillofac Surg. 2022 Jan-Jun;12(1):46-53. doi: 10.4103/ams.ams_141_21. Epub 2022 Aug 16. PMID: 36199454; PMCID: PMC9527844.

[69] Han JJ, Sodnom-Ish B, Eo MY, Kim YJ, Oh JH, Yang HJ, Kim SM. Accurate Mandible Reconstruction by Mixed Reality, 3D Printing, and Robotic-Assisted Navigation Integration. J Craniofac Surg. 2022 Sep 1;33(6):e701-e706. doi: 10.1097/SCS.0000000000008603. Epub 2022 Mar 7. PMID: 35240669; PMCID: PMC9432809.

[70] Piramide, C.; Ulrich, L.; Piazzolla, P.; Vezzetti, E. Toward Supporting Maxillo-Facial Surgical Guides Positioning with Mixed Reality—A Preliminary Study. Appl. Sci. 2022, 12, 8154. https://doi.org/10.3390/app12168154

[71] Koyachi M, Sugahara K, Tachizawa K, Nishiyama A, Odaka K, Matsunaga S, Sugimoto M, Katakura A. Mixed-reality and computer-aided design/computer-aided manufacturing technology for mandibular reconstruction: a case description. Quant Imaging Med Surg. 2023 Jun 1;13(6):4050-4056. doi: 10.21037/qims-22-1118. Epub 2023 Apr 19. PMID: 37284085; PMCID: PMC10240039.

[72] Koyachi M, Sugahara K, Odaka K, Matsunaga S, Abe S, Sugimoto M, Katakura A. Accuracy of Le Fort I osteotomy with combined computer-aided design/computer-aided manufacturing technology and mixed reality. Int J Oral Maxillofac Surg. 2021 Jun;50(6):782-790. doi: 10.1016/j.ijom.2020.09.026. Epub 2020 Nov 3. PMID: 33158695.

[73] Narayan, K. Lalit (2008). Computer Aided Design and Manufacturing. New Delhi: Prentice Hall of India. ISBN 978-8120333420.

[74] R. McNeel, “Rhinoceros User Guide”, 2020, [Online]. Available: http://docs.mcneel.com/rhino/6/usersguide/en-us/index.htm.

[75] R. Mac, “Rhinoceros Guida all ’ uso”, 2019

[76] https://www.ueq-online.org/

[77] Gioacchini Melania, Augmented Reality App Development for Maxillofacial Surgery.