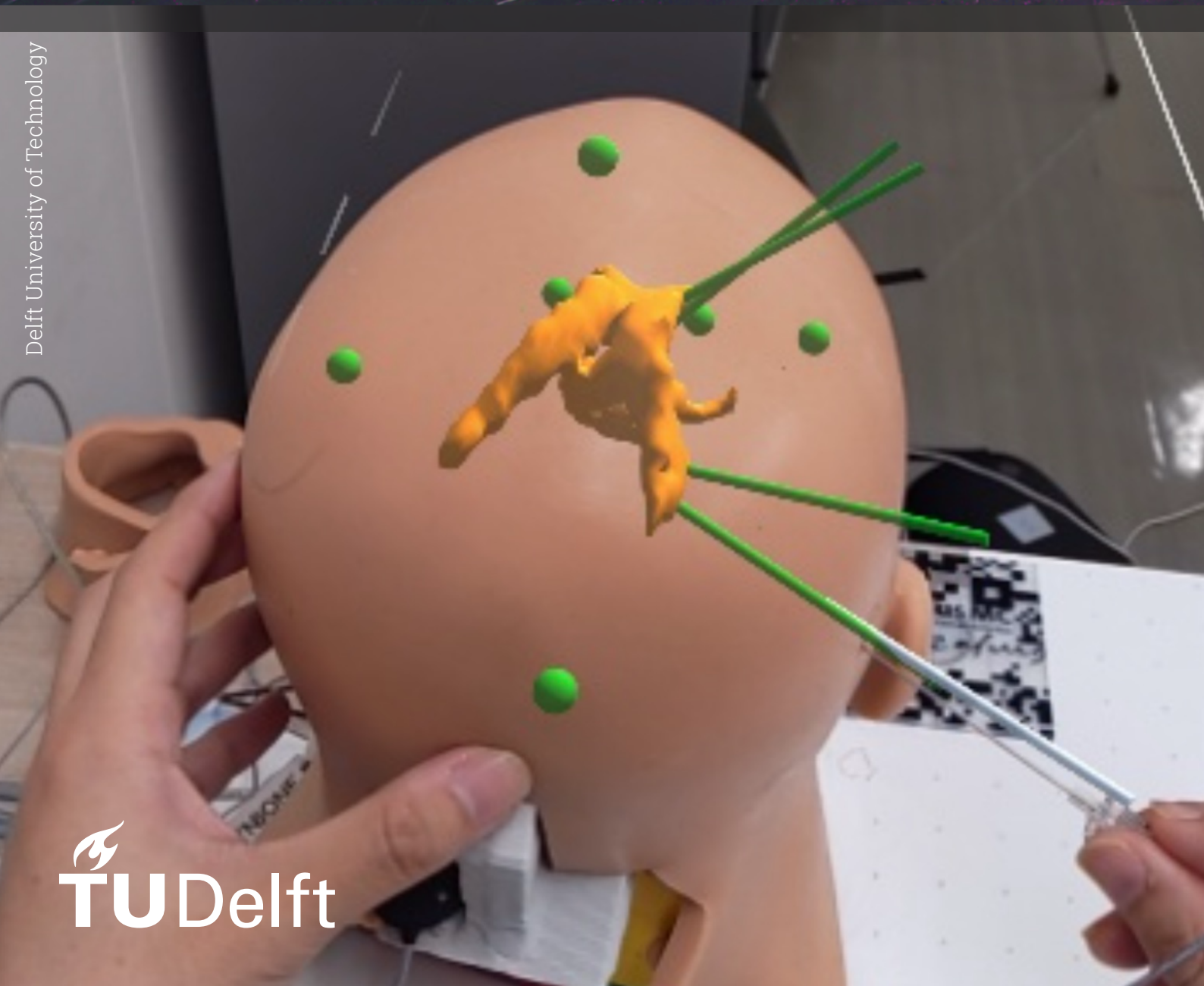
An AR Registration Approach for Ventricle Insertion Navigation Using the HoloLens2

Biomedical Engineering Master Thesis

Xiang Gao



Delft University of Technology

TU Delft

An Automatic Registration Approach for Ventricle Insertion Navigation Using the HoloLens2

by

Xiang Gao

Xiang Gao

5718589

Supervisor:	Ricardo Marroquim (TUDelft)
Daily Supervisor:	Mohamed Benmahdjoub (Erasmus MC)
	Abdullah Thabit (Erasmus MC)
Project Duration:	12, 2023 - 10, 2024
Faculty:	Faculty of Mechanical Engineering, Delft

TU Delft

Preface

This thesis marks the conclusion of my two-year journey in the Biomedical Engineering master's program at TU Delft and Erasmus MC. These past years have not only deepened my technical expertise but also honed my critical thinking and problem-solving skills, which I believe will be crucial in my future career.

I would like to express my sincere gratitude to Professor Ricardo, whose patience and insightful guidance were instrumental throughout my research. Additionally, I am deeply thankful for the support of Mohamed and Abdullah, who meticulously guided my experiments and helped me navigate the challenges of my project. Their expertise was invaluable.

I want to thank TU Delft and Erasmus MC for providing the essential resources and academic environment that enabled the success of my research. The access to advanced technology, cutting-edge facilities, and an enriching collaborative atmosphere were key factors in making this project a success.

Lastly, to my family and friends, thank you for being my source of strength and emotional support during this demanding process. Your encouragement has been instrumental in helping me achieve this milestone.

Xiang Gao
Delft, November 2024

Abstract

The application of Augmented Reality (AR) technology in neurosurgery is becoming increasingly widespread, especially in External Ventricular Drainage (EVD) procedures, where achieving efficient and precise registration remains a key challenge. Traditional marker-based registration methods are often complex and time-consuming, making them unsuitable for emergency surgical situations. This paper presents a markerless automatic registration method based on HoloLens 2, using the Dlib library to automatically detect facial landmarks, and performing point cloud registration based on these features. We designed four sets of experiments: first, to verify the feasibility of the triangle-based registration algorithm; second, to assess the accuracy and stability of Dlib's feature point extraction on HoloLens; third, to validate the registration accuracy, including the impact of skin displacement; and finally, to evaluate the accuracy of the insertion path in a simulated EVD procedure. The experimental results show that this method simplifies the registration process and demonstrates advantages in terms of registration accuracy and speed, effectively meeting the real-time and precision requirements of emergency surgeries. This study provides an efficient and reliable AR navigation solution for EVD procedures, with promising prospects for further research.

Contents

1 Introduction

In recent years, the application of augmented reality (AR) technology in surgical procedures has developed rapidly, gradually becoming an important part of modern medical technology. Microsoft's HoloLens mixed reality glasses, with its advanced cameras, depth sensors, and computer vision technology, can help surgeons visualize critical anatomical structures in real-time during complex surgeries, significantly improving the accuracy and safety of the operations. Registration technology, as the core of this process, precisely aligns digital images with the patient's actual anatomy. This not only enhances the visual experience for doctors but also greatly reduces the risk of errors during surgery, thereby improving the success rate of surgeries and the overall outcomes for patients.

Among various surgical procedures, external ventricular drainage (EVD) is a particularly challenging field. EVD surgery is commonly used to treat hydrocephalus, intracranial hypertension, or intracranial hemorrhage, aiming to insert a catheter into the ventricular system to drain excess cerebrospinal fluid. The main indications for EVD include subarachnoid hemorrhage, intraventricular hemorrhage, and traumatic brain injury [13]. These conditions often cause the drainage pathways of cerebrospinal fluid to be blocked by blood, leading to acute hydrocephalus and increased intracranial pressure [14]. Although EVD can effectively alleviate these problems, its operation is complex and prone to complications such as catheter path bleeding, improper catheter placement, and cerebrospinal fluid infections [17]. In traditional surgeries, doctors usually rely on anatomical landmarks and preoperative images for catheter insertion, or perform blind insertion based on experience. However, due to individual differences in patients' anatomy and positional changes during surgery, manual catheter insertion often carries errors, which may lead to misplacement or ineffective drainage [1, 26].

AR technology provides significant potential improvements for EVD procedures. Through systems like HoloLens, preoperative images can be overlaid with the patient's actual anatomy in real-time, providing surgeons with three-dimensional visual guidance. AR can not only accurately display the insertion location of the catheter but also show the real-time position of surgical tools, reducing frequent visual switching during surgery. Furthermore, AR can help doctors better understand hidden anatomical structures, avoiding damage to key areas and further improving the accuracy and safety of surgeries.

Recently, more and more research has focused on registration techniques for the HoloLens in EVD surgeries [20, 7, 25, 22, 23]. However, most of these techniques require external markers and auxiliary devices. For example, studies by Li et al. [20] and Wang et al. [25] propose methods that combine reflective markers with point cloud registration to achieve more precise and stable head tracking. This method uses reflective markers attached to the patient's head and utilizes the HoloLens 2's depth sensors to capture multiple frames of data, reconstruct the head surface, and perform registration using the point cloud-based ICP algorithm [7]. Another system [22] uses optical marker tracking through the OptiTrack cameras and the HoloLens 2, while some approaches [23] employ QR codes for registration. Despite the progress in HoloLens registration for EVD surgeries, these methods often require substantial preparation time for equipment calibration and marker placement. This is not ideal for emergency surgeries requiring a fast response, especially when dealing with acute intracranial pressure elevations.

Time is of the essence, as excessive intracranial pressure can lead to brain hypoxia or other irreversible damage. Doctors need to quickly insert the drainage tube to relieve pressure, but traditional marker-assisted registration methods take too long, failing to meet this need. Therefore, optimizing the registration process to increase speed while maintaining high accuracy and stability has become a key issue to address. In this context, markerless and external-device-free registration methods have emerged as a promising area of research.

In recent years, markerless registration methods based on depth sensors have gained popularity among researchers due to their simplicity and ability to perform surgical tasks more quickly. However, these methods also face challenges. For instance, the environmental data captured by the HoloLens sensors tend to be inaccurate, which can affect registration accuracy. Xie et al. [15] used the HoloLens's spatial mapping function to align the virtual model with the patient's skin surface, but the coarseness of the surface mesh data limited registration accuracy. Additionally, point cloud registration methods, while offering higher precision, often capture irrelevant information, requiring manual cropping of the point cloud. Enzo's study [27] proposed a depth-sensing-based HoloLens registration technique that relies on complete point cloud data. The method requires manual initialization, such as coarse registration or visual center cropping, to process the point cloud data. However, this approach often causes the ICP (Iterative Closest Point) algorithm to fall into local minima [21], especially when the point cloud data is incomplete or cropped improperly, affecting registration accuracy and increasing time and complexity due to manual cropping. Gsaxner et al. [9] propose a deep learning-based solution that uses a pre-trained SSD network to automatically detect regions of interest (ROI), effectively removing irrelevant noise points and improving registration accuracy. However, this method relies on pre-trained models and cannot accurately segment regions of interest in the point cloud, still requiring the ICP algorithm for fine registration. Similarly, the Vuforia engine was used for feature-based target tracking in temporomandibular joint (TMJ) surgeries [28]. Although this method creates multiple views to assist in surgical registration, it requires pre-uploading models and has limited tracking accuracy.

Given the challenges faced by existing markerless registration methods, as well as the high demands for operational efficiency and simplicity in EVD catheter insertion surgeries, this work proposes a new landmark point cloud-based automatic registration method that utilizes the Dlib computer vision library to optimize the process. First, we use Dlib to detect and extract regions of interest (ROI) from the point cloud data, typically facial landmarks. By focusing on these landmarks, we aim to improve registration accuracy while reducing the computational load required to process the entire point cloud.

After extracting the regions of interest, we employ a triangle-based registration algorithm to align the point cloud precisely. This algorithm leverages the geometric properties of triangular meshes to effectively handle noise and irregularities in the point cloud, resulting in more stable and accurate registration outcomes. Moreover, the triangle-based registration method demonstrates high computational efficiency when handling large-scale datasets, making it suitable for real-time surgical environments.

This study aims to improve the application of the HoloLens technology in EVD surgeries by proposing a reliable and efficient markerless, external-device-free automatic registration method. With the continuous optimization of these technologies, AR may play an increasingly important role in future surgical procedures, driving the advancement of medical imaging and surgical intervention techniques. The following sections will introduce the proposed method in detail and validate its effectiveness through experiments.

2 Related Data processing technologies and libraries

2.1. Data Preparation

2.1.1. Preoperative Data Preparation

In a HoloLens-assisted EVD surgery, the initial step is to acquire the original head model. The registration method used is landmark-based, where the virtual head CT image data, along with the corresponding guided trajectories and other visualized images, are aligned with the real patient's head model. The head model is provided by the CT slices scan imaging of the actual patient, as shown in Figure 2.1.

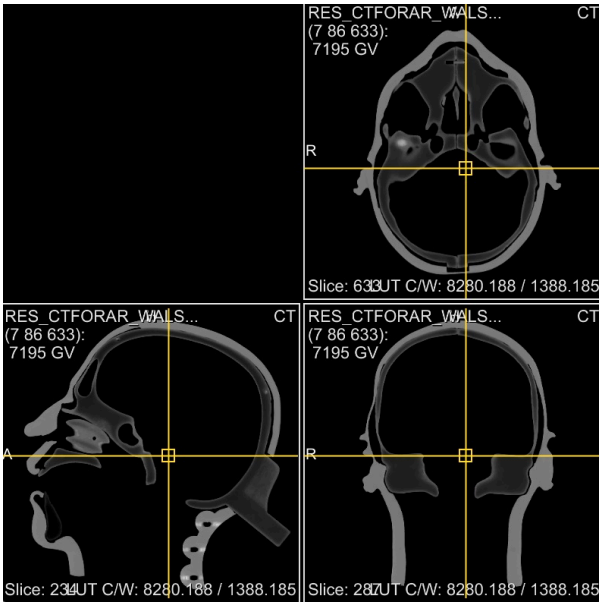


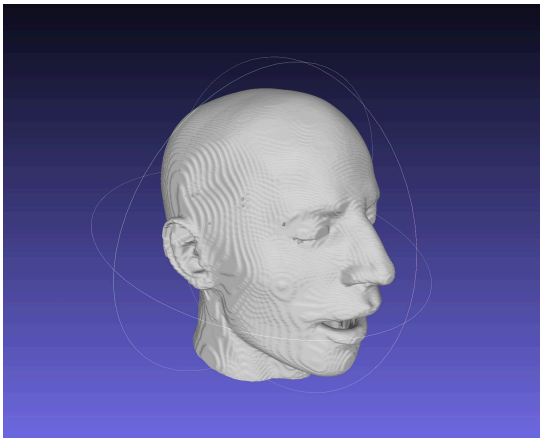
Figure 2.1: Dicom CT slices of head model

The geometry representation of the HoloLens is the mesh format, which must be converted from the original medical data. Typically, the medical imaging data obtained from medical imaging devices are in DICOM format. DICOM (Digital Imaging and Communications in Medicine) CT data and Mesh models represent two distinct types of data representations. DICOM is a medical imaging standard containing both images and patient data, commonly used for CT scans that represent 3D anatomy as a series of 2D grayscale slices. In contrast, Mesh models are used in graphics and AR applications like HoloLens, representing 3D shapes by connecting vertices, edges, and faces to create a surface.

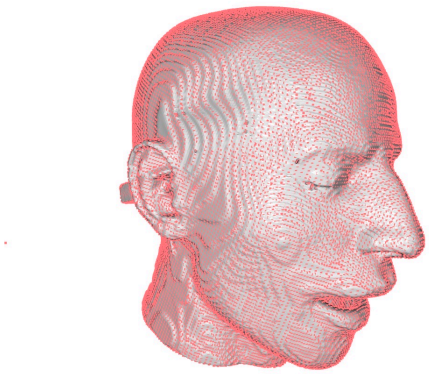
Converting DICOM CT data to a mesh model involves multiple steps. First, the DICOM images are pre-processed, such as for noise reduction and contrast adjustment, to improve the image quality. Then, a 3D voxel model is generated based on these images. Following this, the conversion from the voxel model to a mesh model is accomplished through surface reconstruction. This process involves surface extraction, mesh smoothing, and simplification

to meet rendering and visualization needs. This series of steps can be completed using various software tools such as MeVisLab or 3D Slicer. Ultimately, the mesh model is converted into formats supported by HoloLens, such as OBJ or FBX, for further use in applications and visualization.

When converting the 3D model into point cloud format, Open3D is first used to load the generated mesh model. Open3D provides sampling functions to uniformly sample the mesh and convert it into point cloud data format, which is shown in Figure 2.2(a). Open3D supports various sampling strategies, such as area-based uniform sampling and Poisson sampling, which can effectively capture the geometric details of the model. After sampling, the generated point cloud can be further used for tasks such as 3D registration, visualization, and analysis.



(a) Mesh visualization



(b) Mesh with point cloud (pink color)

Figure 2.2: Mesh to point cloud data

2.1.2. Preoperative landmarks

For landmark-based registration, it is essential to mark landmarks on the head model before surgery. This process involves annotating specific landmarks on the virtual model, which are used to highlight key points on the face that are relatively stable and less prone to movement. This enhances the accuracy of rigid registration between the virtual model and the actual patient during surgery. Typically, landmarks such as the corners of the eyes and the tip of the nose are chosen. Since this is a point cloud-based registration task, the head's CT model must first be converted into a point cloud format, as shown in Figure 2.3.

The facial landmarks are manually selected from the point cloud model. As shown in Figure 2.4(a), a partial point cloud of the head was generated using a 3D scanning mobile application[1]. Because this point cloud data contains color information, it is easier to identify facial features compared to a point cloud generated from a CT scan converted into an OBJ file. Therefore, this point cloud model is used during the initial testing and simulation of the registration algorithm (manually selecting points based on facial features).

Landmarks are selected using the Open3D library, which offers a point selection feature. By scripting, Open3D allows users to interactively select a point on the point cloud model using the mouse, as shown in Figure 2.4. The selected point's index in the overall point cloud is then registered, allowing the marked point to be referenced in subsequent registration processes.

For point clouds sampled from mesh models, Open3D is typically used to extract the vertices

[1]3D Scanner App by Laan Labs. Available on the App Store.

Figure 2.3: Point cloud of head model

(a) point cloud used for test

(b) Choosing points on test point cloud

Figure 2.4: Choosing landmarks in Open3d

of the mesh and save them as point cloud data. However, it is often difficult to accurately identify facial features for landmark selection, as shown in Figure 2.3. Therefore, an alternative approach is to first process the mesh model in Unity, as it provides a clearer representation of facial features. Landmarks can be annotated in the Unity scene. Then, based on the coordinate information of the landmarks, the nearest points in the point cloud at the corresponding locations (landmarks in the point cloud) can be calculated. To improve computational efficiency, a KD-tree can be built for efficient nearest-neighbor search [6].

2.2. Dlib Toolkit auto-detects facial landmarks

Dlib has been widely used in the fields of machine learning and computer vision, known for its efficient and accurate algorithms. The object detection and facial recognition algorithms in Dlib have performed excellently in multiple studies. For example, King (2009) [19] detailed the design and implementation of Dlib, emphasizing its efficiency and accuracy in the C++ environment.

In terms of facial expression recognition, the Dlib model also demonstrated outstanding performance. A recent study using the Dlib model for facial expression recognition showed an accuracy of 97 of 100 [2]. This high precision makes Dlib highly reliable in practical applications. In summary, Dlib has gained widespread application and recognition across multiple fields due to its efficient and accurate algorithms.

In this study, we employed the Dlib library to facilitate the automatic detection of landmarks, streamlining the process of marking points within the point cloud data. Dlib is a widely-used open-source machine learning library, particularly renowned for its robust applications in computer vision, including facial detection and landmark recognition. The facial detection algorithm in Dlib leverages Histograms of Oriented Gradients (HOG) features combined with a linear classifier, using image pyramids and sliding window techniques for efficient face detection [19]. Moreover, Dlib includes a pre-trained model capable of detecting 68 facial landmarks, which allows for precise identification of facial features [17]. By deploying Dlib for real-time landmark detection within the Hololens camera system, we ensure reliable support for augmented reality (AR) applications, enabling both facial recognition and expression analysis [3].

To implement real-time facial landmark detection, Dlib was set up on the server side. The frames captured by the Hololens front camera are transmitted to a PC via TCP protocol, using the HL2SS client. Each frame is processed by Dlib for face detection, and once a face is identified, the detection process moves on to identifying 68 landmark points. These detected landmarks are then visualized, typically marked in blue, highlighting each identified key point, as illustrated in Figure 2.5. This approach efficiently automates the landmark detection process, simplifying data handling and providing a stable and accurate foundation for subsequent AR applications.

During the process of detecting facial landmarks using Dlib, the system not only identifies key points in the 2D image but also matches these points with depth information to generate a 3D point cloud. Specifically, the 2D target points detected by Dlib are remapped and integrated with depth information and applied to the 3D point cloud reconstruction process through TSDF (Truncated Signed Distance Function). In this process, the detected facial landmarks (such as the inner corners of the eyes and the tip of the nose) are automatically located in the 3D point cloud and aligned with other structures. Figure 2.6 shows the overlap of the red landmarks detected by Dlib with the actual 3D point cloud.

This method of combining depth information with facial landmark detection not only enables precise landmark identification but also functions as a filter during the point cloud reconstruc-

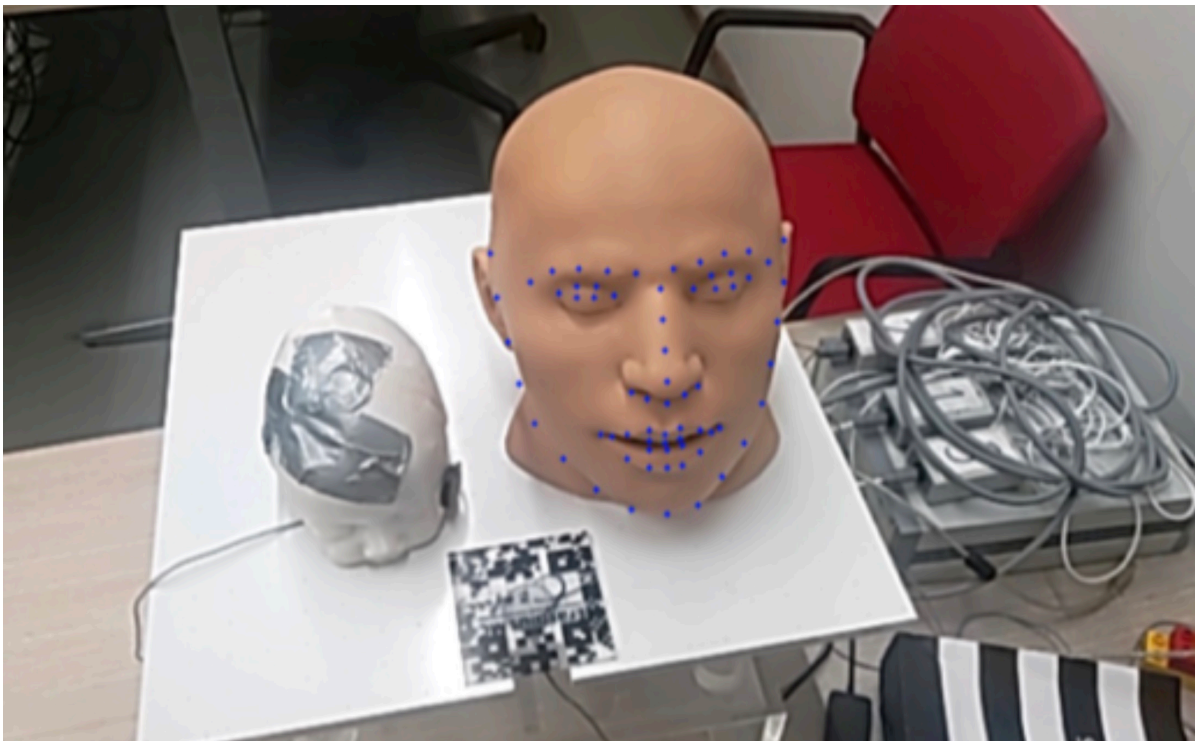


Figure 2.5: Dlib detecting landmarks

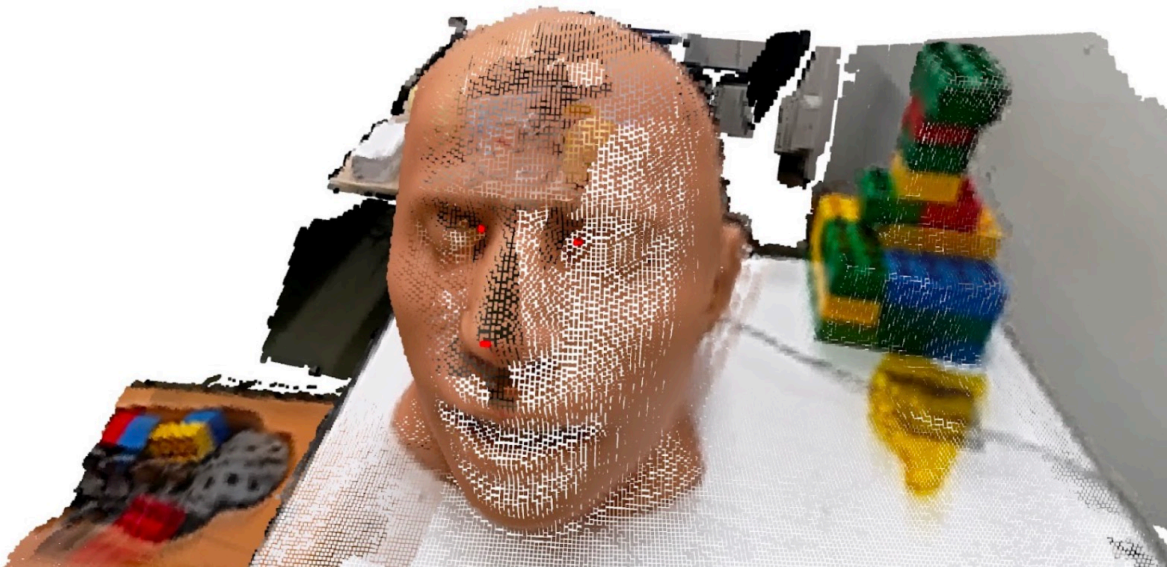


Figure 2.6: Dlib extracted landmarks (red) with point cloud of environment visualization

tion process. Although the TSDF can generate a complete point cloud of the face, the system retains only the 3D point cloud data related to the landmarks detected by Dlib, as these landmarks are identified in advance. This approach reduces computational load and enhances the accuracy of point cloud registration. As shown in Figure 2.7, Dlib first processes the image data from the HoloLens camera, detecting key landmarks in the 2D image and matching these points with the depth map. When Dlib applies these coordinates to the point cloud reconstruction process, it effectively filters out all irrelevant point cloud data that does not correspond to the detected landmarks, ensuring that the final point cloud reconstruction only includes the essential facial landmark points. This efficient preprocessing mechanism contributes to the precision of the registration algorithm between the virtual model and the actual anatomical structure.

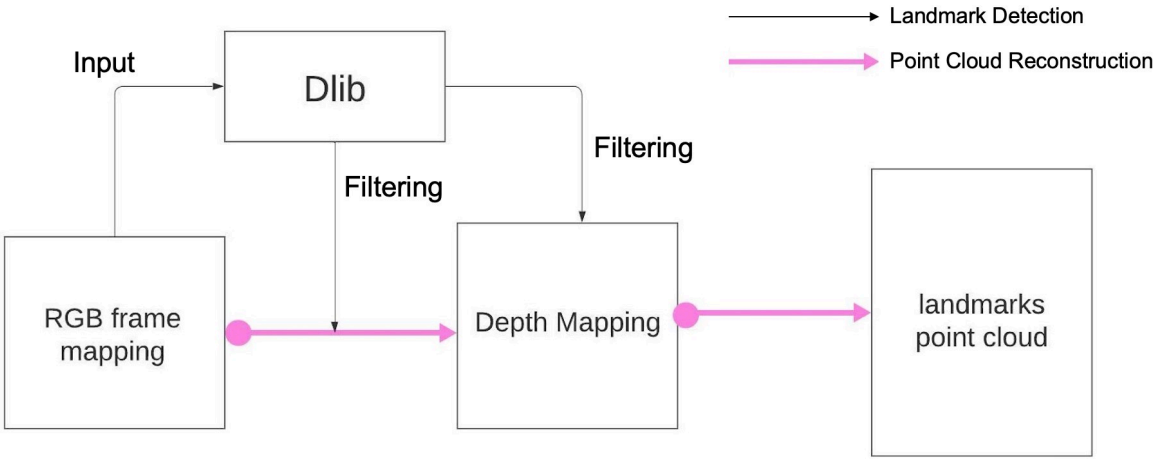


Figure 2.7: Dlib in pipeline

2.3. AR Visualization in Unity

In this study, a navigation program based on Unity is designed to assist in guiding EVD surgeries using HoloLens. In Unity, we create a new project and import the pre-prepared OBJ head model for visualizing the surgical process. The model is carefully adjusted with specific color and transparency settings to allow users to clearly see the relationship between the virtual head model and the actual head model, as shown in Figure 2.8. Visual feedback is essential for surgeons during the operation, improving the accuracy and safety of the procedure.

The visualization of the ventricles is very important. We ensure that the ventricles are displayed with appropriate visualization modes and colors so they are well-represented within the program. Additionally, the surgical planning path is visualized in the program to guide the surgeon in following the pre-planned route during the surgery. These visualization elements can be shown or hidden dynamically using voice command features in Unity, allowing the surgeon to adjust the content as needed during the procedure.

While traditional navigation systems display enhanced information such as annotations, tracking tools, and anatomical segmentation superimposed on preoperative data, AR technology takes this a step further by overlaying these enhancements into the user's real-world environment. Microsoft's HoloLens 2 is particularly suitable, as it can function as a standalone navigation system. When using HoloLens 2 for AR visualization, the models must be optimized based on the device's computing capabilities, display features, transparency, and other

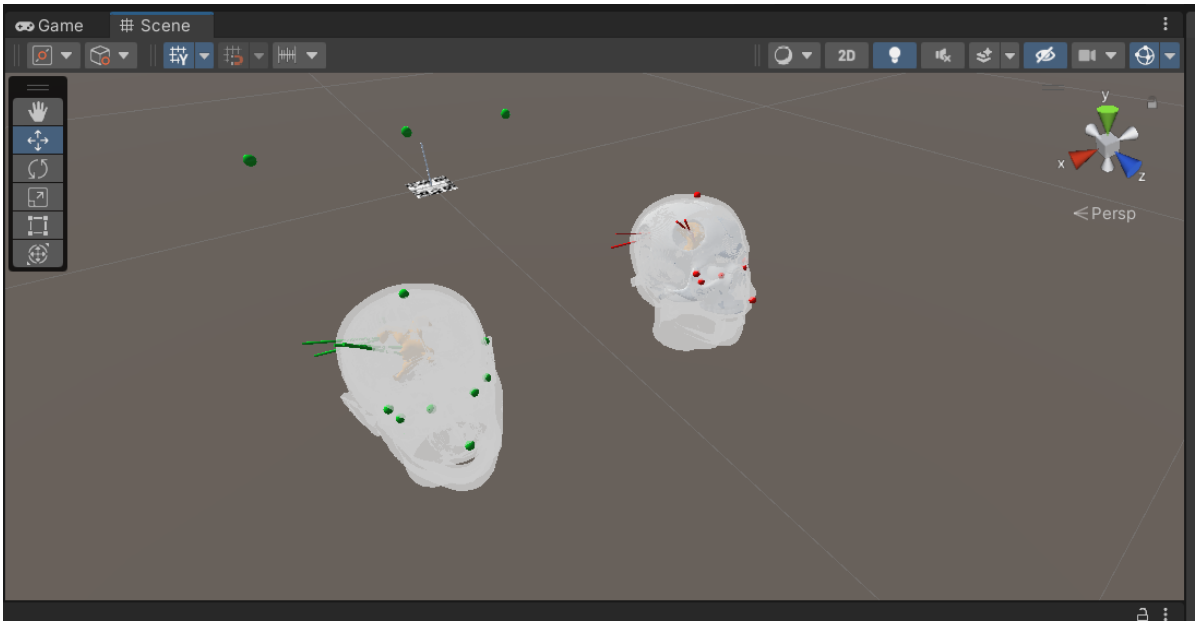


Figure 2.8: Unity scene for AR visualization

factors. This ensures smooth rendering and interaction on the HoloLens 2.

In some cases, more detailed models may be used during registration than those used for visualization. For example, models with many vertices and faces can be remeshed into simpler versions for visualization, reducing rendering complexity while maintaining functionality. This helps ensure smooth performance on the HoloLens without overloading the device.

The AR visualization workflow begins by displaying surface models, enabling users to assess the accuracy of the registration. Once the alignment is deemed satisfactory, voice commands can be used to deactivate the surface model, allowing the user to focus on the projected internal models. These internal models, such as the ventricle in the case of EVD surgery, are expected to guide the surgeon throughout the procedure. Pre-planned incisions and drilling paths can also be visualized on demand, improving the precision and accuracy of the surgical intervention.

Through this design and implementation, our navigation program aims to provide intuitive and efficient support for EVD surgeries, helping surgeons make more precise decisions in complex surgical environments. The ultimate goal of this research is to enhance the practical application of HoloLens technology in real-world surgeries, offering patients a safer surgical experience.

3 Methodology

Multiple steps are involved in the External Ventricular Drainage (EVD) insertion surgical navigation process. This includes the initial preoperative patient imaging scans, the creation of the virtual model, the mid-stage preoperative planning and preparation of the AR program, as well as the later stages involving registration and AR visualization for surgical guidance. The entire workflow from preoperative to postoperative stages is illustrated in Figure 3.1. The ultimate goal is the augmented visualization of the target area, represented by the red circular region.

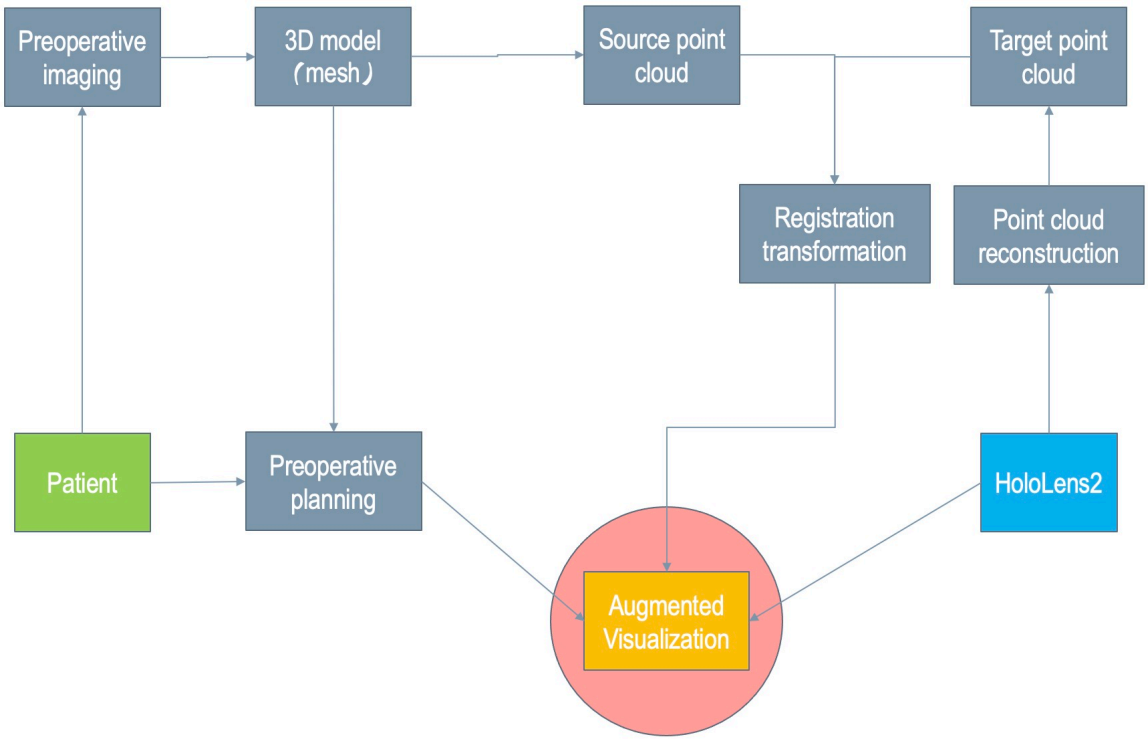


Figure 3.1: Flow chart of depth-based registration

3.1. Registration Algorithm

The registration algorithm aligns data from different sensor sources within a unified reference frame, enabling precise alignment between virtual models and real-world environments. In augmented reality (AR) systems, particularly in surgical navigation, registration algorithms are crucial as they determine whether the virtual model can be accurately aligned with the patient's anatomical structures. This section introduces a landmark-based point cloud registration algorithm. This method aligns data by identifying and matching prominent feature points within the point cloud. Building on this foundation, we will discuss the theoretical principles, key technical components, and how this technique can be effectively implemented in the HoloLens system to ensure efficient and stable registration in complex real-world scenarios.

The landmark-based point cloud registration algorithm relies on three landmark points for

alignment, as illustrated in Figure 3.2. This method leverages the geometric principle that "three points define a plane." By registering the triangles formed by landmark points in both the source and target point clouds, we can achieve data alignment. Commonly used registration algorithms include the Iterative Closest Point (ICP) algorithm [12], and Random Sample Consensus (RANSAC) [11]. ICP is highly sensitive to the initial position, which significantly impacts the final registration result [16], while RANSAC relies on large datasets and a sufficient number of iterations to improve accuracy [4, 8]. A key evaluation metric for these algorithms is the point-to-point distance error. In this experiment, the three-landmark method serves as the basis for registration, simplifying the alignment of point cloud geometries by reducing it to an affine transformation between triangles.

The specific steps for triangle-based registration are as follows, which are also shown in Figure 3.2:

- Initial Alignment: First, align one corresponding pair of points from the source and target triangles;
- Rotation Alignment: Next, rotate the source triangle around the aligned point, ensuring one edge of the source triangle coincides with the corresponding edge of the target triangle;
- Final Adjustment: Finally, rotate around the aligned edge until the distance between the third pair of points is minimized, completing the registration process.

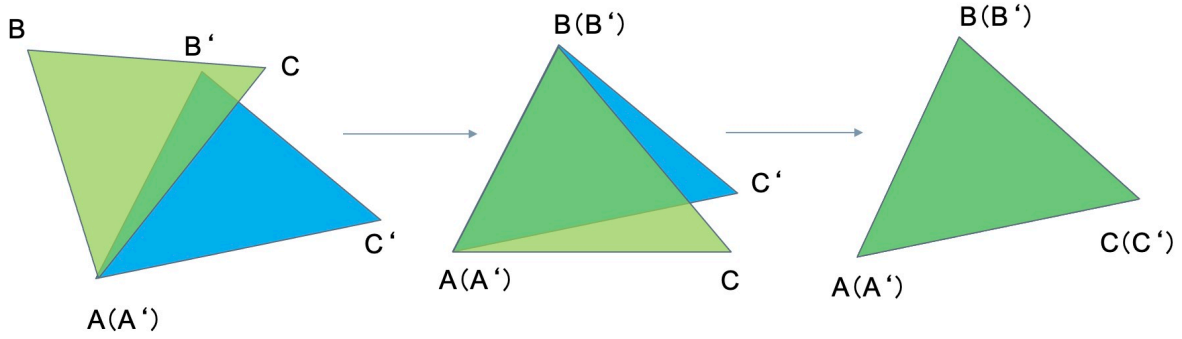


Figure 3.2: Triangle based registration

To implement this approach, the CT scan of the head is transformed into a point cloud, and three marked points are used to construct the source triangle for registration. The target point cloud, on the other hand, is captured using the HoloLens and processed in conjunction with the Open3D library on the PC server. After acquiring the two point clouds, the registration is calculated through the Open3D library, generating the final transformation matrix.

In practical applications, to achieve the correct transformation and registration of models in the Unity scene, a global transformation matrix needs to be sent to the HoloLens. However, in the registration method, local transformations are performed step by step, as shown in Figure 3.2. In each registration step, the local transformation matrix is applied relative to a local reference point (the vertices of the triangle). To ensure that the transformation in each step is correctly applied in the global coordinate system, we must base each registration step on the global coordinate system. Specifically, for each local transformation, we first translate the reference point to the origin of the global coordinate system, perform the necessary rotation or translation, and then apply another translation matrix to return the reference point to its

original position. In this way, each local transformation in the registration process is converted to a global transformation, and each step of the registration can be correctly executed in the global coordinate system. Ultimately, these transformation matrices, converted to the global coordinate system, are multiplied together to produce a complete global transformation matrix, achieving precise alignment between the source and target point clouds.

By using this landmark-based point cloud registration method, we can achieve high-precision alignment between virtual models and real-world structures within augmented reality environments.

It is worth mentioning that during the actual registration process, the source point cloud triangle is pre-defined by manually choosing the points, while the target point cloud triangle is captured by the HoloLens. In this process, Dlib automatically identifies facial landmarks and forms the target point cloud, which will be discussed in detail in Section 2.3. The front camera of the HoloLens continuously captures RGB and depth information, which is transmitted to the host computer for processing through data streaming protocols (such as the TCP scheme used in HL2SS). This process reconstructs the point cloud data, as shown in Figure 3.3, where the red points forming a triangle shape are gradually accumulated. The specific reconstruction methods and workflow will be introduced later. These points are processed to form the final target point cloud triangle. This data transmission process is implemented through a client-server architecture written in Python.

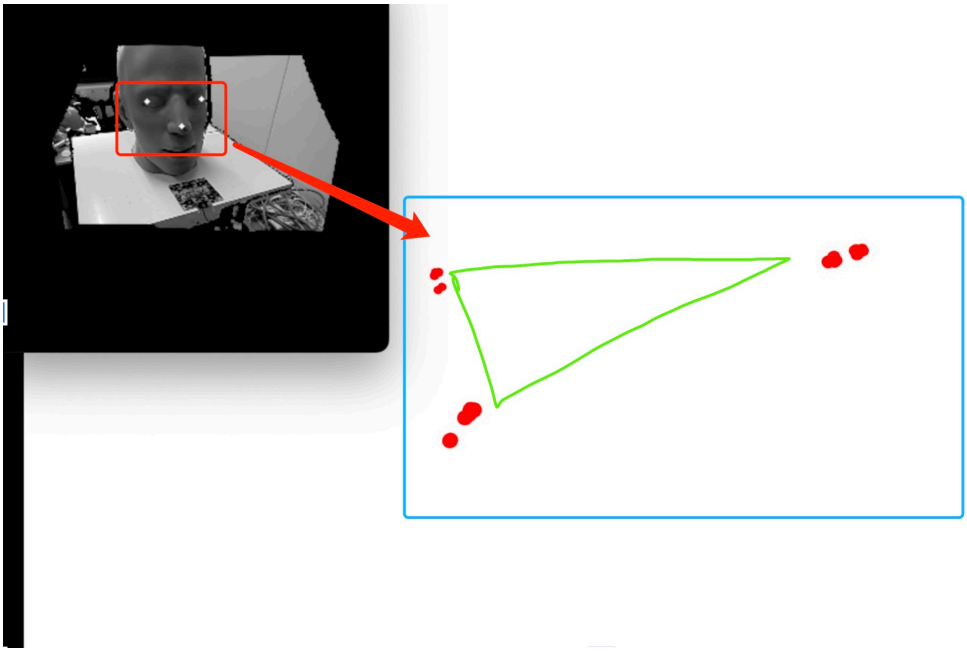


Figure 3.3: Extracting point cloud of landmarks

3.2. Experimental System Workflow

The point cloud registration system in this study consists of a HoloLens 2 and a PC connected to the same network. In this architecture, the HoloLens acts as the client, capturing depth data and providing real-time visualization, while the PC serves as the server, responsible for data processing, computation, and transmitting results back to the HoloLens. The system was developed using the Unity engine, integrated with the HoloLens 2 Sensor Streaming (HL2SS) plugin [10], ensuring efficient data transfer and interaction between devices. The pipeline is shown in Figure 3.4.

In practice, the HoloLens captures real-time point cloud data through its depth sensors and streams it to the PC for processing. The PC uses predefined source and target point clouds for registration, computes the transformation matrix, and aligns the point clouds accurately. Data processing is done using the Open3D Python library, and point cloud reconstruction is performed through the Triangular Signed Distance Function (TSDF) volume segmentation method. After the transformation matrix is calculated, it is transmitted back to the HoloLens, which then displays the real-time registration results, aligning the virtual model with the patient's anatomical structures in augmented reality. This system provides valuable AR feedback that supports surgeons in making more precise decisions during surgery.

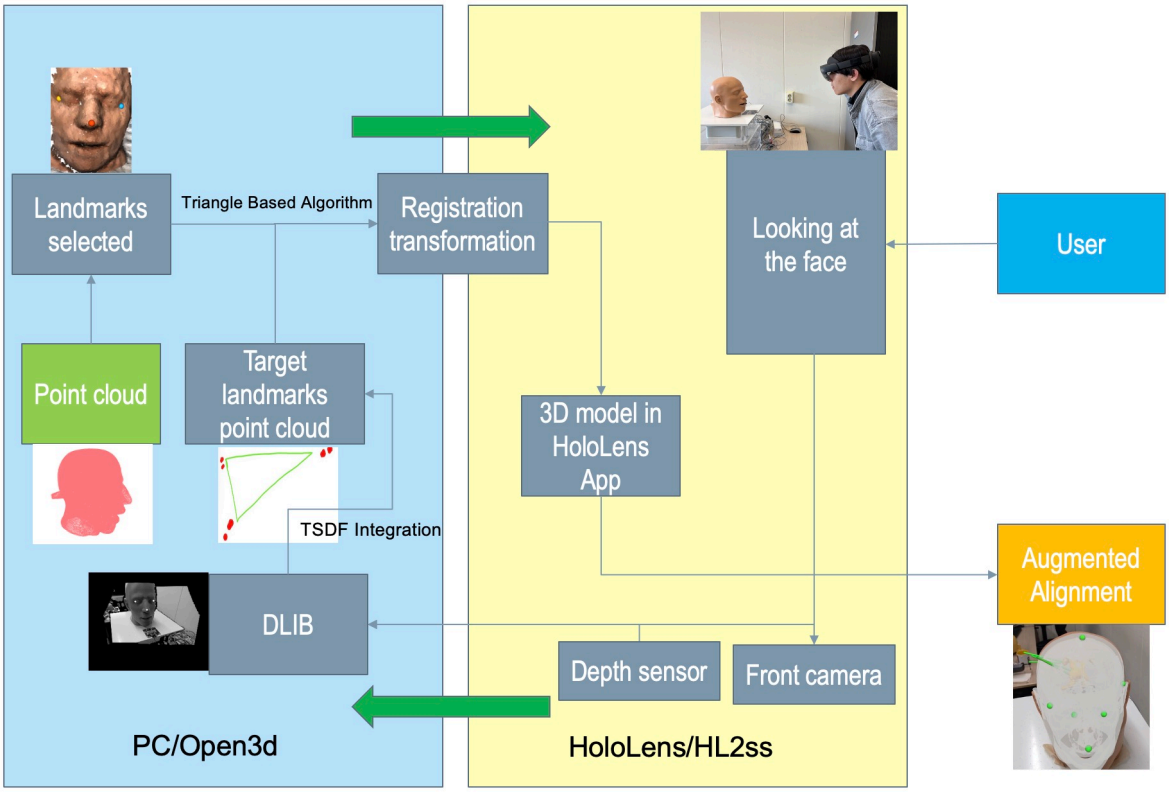


Figure 3.4: Framework of Registration system

4 Experiment and Results

We conducted four experiments aimed at evaluating the effectiveness of different techniques in surgical guidance. The first experiment focused on a simulation validation based on landmark point cloud registration, with the goal of verifying the accuracy and feasibility of the triangle-based registration algorithm. The results of this experiment will provide a reliable foundation for the subsequent ones. In the second experiment, Dlib was used to extract landmarks via the HoloLens, primarily to measure the precision and accuracy of the landmark point cloud extraction by the HoloLens, ensuring the quality of the collected data. The third experiment centered on the overall accuracy verification of the head model registration, assessing the system's stability and consistency through multiple registration tests. Lastly, the fourth experiment validated the precision of EVD insertion during a simulated surgery. This series of experiments is designed to offer critical insights for practical applications and further explore the potential of these technologies in complex surgical environments. Notably, in the following experiments, the left and right outer eye corners and the tip of the nose were chosen as landmarks.

4.1. Experiment 1: Simulation of Landmark-Based Triangle Registration

4.1.1. Experimental Setup

The aim of this experiment was to verify the effectiveness of the triangle-based point cloud registration method using landmarks. This method involves selecting three corresponding landmark points in both the source and target point clouds, which define two triangles. The registration process is achieved by calculating a geometric transformation matrix that aligns these two triangles. Since three points can define a triangle, the process of aligning the source and target point clouds can be viewed as registering two corresponding triangles. After applying the transformation matrix, the source point cloud is accurately aligned with the target point cloud. The selected facial feature points, including the left outer eye corner, the right outer eye corner, and the tip of the nose, are shown in Figure 4.1.

When selecting facial landmarks, it is important to account for slight displacements caused by skin and muscle movement. These landmarks are less likely to be affected by facial movement, ensuring rigid registration of the model during surgery. To ensure the accuracy of feature point selection, a mobile phone-scanned point cloud with color information was used in the experiment, allowing for a more intuitive selection of facial features compared to a monochromatic CT scan-generated point cloud. In addition, a comparison was made between the ICP (Iterative Closest Point) algorithm and the triangle-based registration method proposed in this study. The ICP algorithm, being sensitive to the initial position, usually requires a coarse registration as a foundation, while the proposed method reduces dependency on the initial position through affine transformation, addressing the sensitivity issue during the early stages of ICP registration.

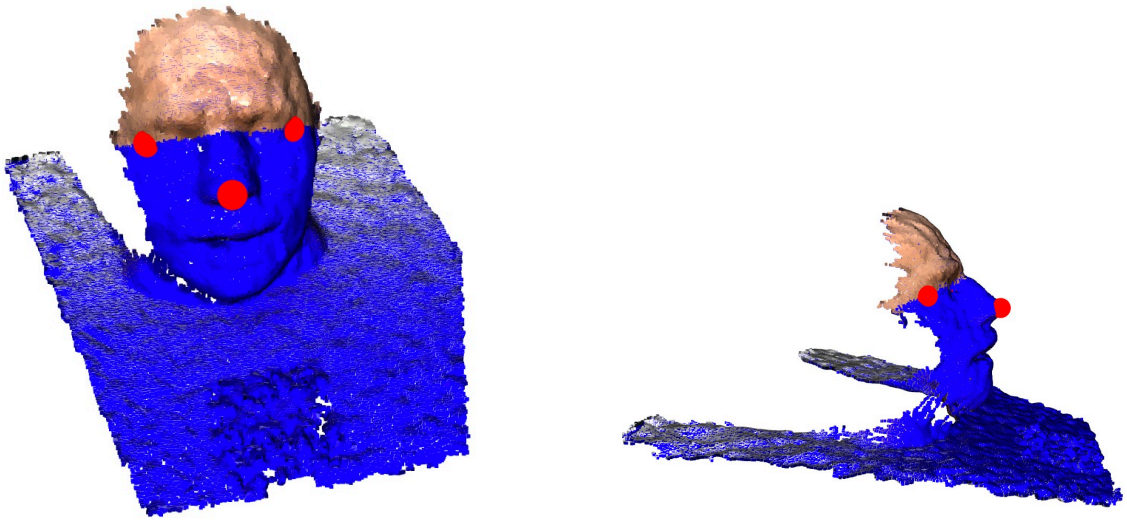
4.1.2. Experimental Results

This experiment was tested under two conditions: one with manually selected approximate landmark points and the other with exactly identical landmark points. Under the manual

Figure 4.1: Point cloud with chosen landmarks (colored spheres)

selection condition, the main purpose was visual validation, so no quantitative tests were conducted. Figure 4.3(a) shows the visualized registration result of the source point cloud (red) and the target point cloud (green). Due to larger errors in manual selection, registration under this condition was primarily used to validate the feasibility of the algorithm, with limited reference value for practical applications.

Under the condition of selecting identical landmark points, the system automatically calculated the transformation matrix between the source and target point clouds using the Open3D library, generating precise registration results. Quantitative test results indicated that even with the accurate selection of landmark points, a certain amount of error remained in the registration process. The registration TRE for the method of choosing the same points was 0.31 mm $\pm$ 0.04 mm.



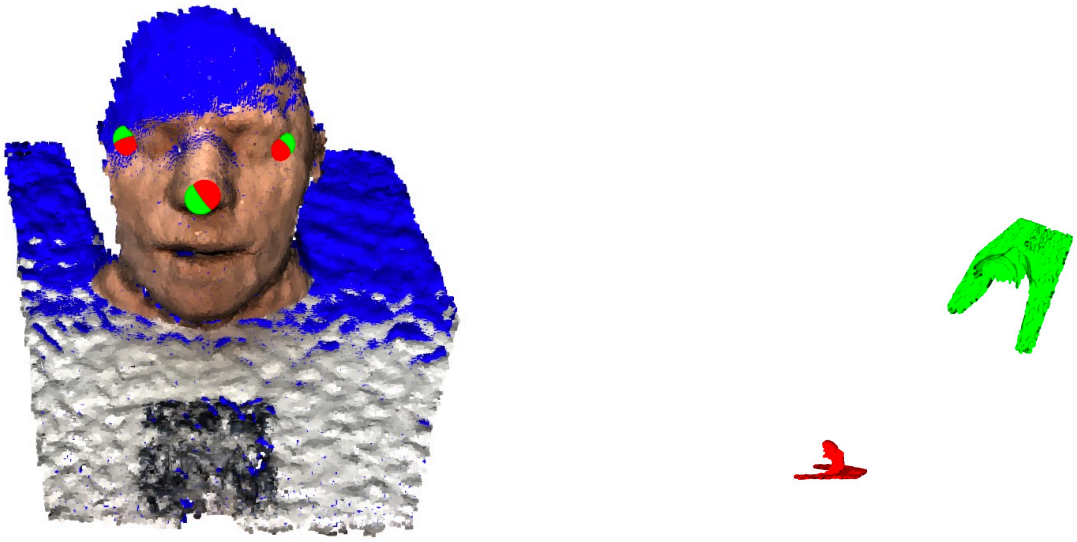
(a) Triangle registration front view

(b) Triangle registration side view

Figure 4.2: Triangle registration side view

In the comparison experiment with the ICP algorithm, only qualitative tests were performed,

specifically testing the advantages of the proposed registration algorithm compared to ICP under conditions where the source and target point clouds were far apart. The experimental results showed that when the source and target point clouds were at a significant distance, the initial registration of ICP often failed, as shown in Figure 4.3. This is because ICP is overly sensitive to the initial positional relationship. In contrast, the triangle-based registration algorithm proposed in this study was able to achieve satisfactory registration results even under distant conditions, successfully overcoming ICP's sensitivity to the initial position and validating the superiority of this method.



(a) Triangle algorithm by manually choosing landmarks

(b) Failure case for ICP Algorithm (source point cloud: red target point cloud : green)

Figure 4.3: registration Visualization of ICP algorithm and triangle algorithm

4.2. Experiment 2: Dlib Facial Landmark Extraction via HoloLens

4.2.1. Method

In the pre-prepared 3D model within the HoloLens software, manual marking of facial landmarks on the human model is required. Arbitrarily selecting these landmarks would lead to significant inaccuracies. Therefore, this study selected Dlib as the tool to detect facial landmarks in the real world to improve the accuracy of marking.

In the experiment, we used the front camera of the HoloLens2 as the camera for users to observe and detect facial features. Through a TCP communication system, each frame of the RGB images captured by the front camera was sent to the server host, where Dlib was used to detect facial landmarks.

It is important to note that the RGB images captured by the front camera undergo resampling and interpolation to match the spatial location of the depth map. This allows each pixel of the depth information to correspond to a pixel of RGB information. In this case, Dlib's facial detection on the resampled RGB image can simultaneously locate the depth information of the corresponding facial landmarks.

The purpose of this is to address the problem that facial landmarks cannot be directly extracted from depth information. Since the depth information is what we actually need, representing the positional coordinate information, Dlib's limitation is that it can only detect landmarks

based on RGB information.

4.2.2. Point Cloud Extraction Device

In the HoloLens, we used its built-in TOF depth sensors, Long Throw (LT), and the front camera to collect depth and RGB information. The AHAT and LT depth sensors in the HoloLens are responsible for short-range and long-range depth detection, respectively; AHAT is used for short-range detections, such as hand gesture recognition, while LT is mainly responsible for scene environment perception.

Referencing the study by Enzo et al. [18], which compared the two depth sensors of the HoloLens, it was found that the AHAT depth sensor offers higher accuracy at close range (20cm-60cm), while the LT sensor performs better at longer distances (greater than 60cm). The LT sensor generally shows higher error compared to the AHAT sensor, but this difference decreases for surfaces located beyond 50cm. Additionally, the standard deviation of the AHAT sensor increases with distance to the surface, and it exhibits more outliers compared to the LT sensor.

The distance for detecting facial landmarks with the HoloLens is approximately between 50cm and 80cm, falling within the suitable detection ranges for both AHAT and LT sensors, but more towards the long-range detection. Moreover, AHAT produces more outliers when detecting depth information, resulting in more outlying points in the point cloud, while the LT sensor generates fewer outliers. Since the registration method in this study is based on landmarks, having less noise helps reduce the acquisition and accumulation of outliers, which can potentially improve registration accuracy. Therefore, considering all factors, we chose the LT sensor for depth detection.

4.2.3. Stability of Dlib-Extracted Target Point Cloud

Experimental Setup

To evaluate the accuracy of extracting facial landmarks from the point cloud, this study conducted assessments from two dimensions: first, the stability of the extracted landmark point cloud, and second, the accuracy of the extracted landmark point cloud.

To verify the stability of extracting landmark point clouds, we set up a procedure where the user performed facial landmark point cloud extraction. During the extraction process, the user slowly moved left and right as well as forward and backward, aiming to keep the Hololens parallel to the frontal face of the head model. This allowed for dynamically collection of the landmark point clouds. As shown in Figure 4.4, subsequent experiments were also conducted following this standard to achieve consistent data collection.

Each landmark's position accumulates point clouds from different frames. After outlier removal and averaging, the final point is used in subsequent registration processes.

In the experiment, 14 sampling iterations were conducted, and the results were obtained. By recording and analyzing the final coordinates of the collected points across multiple iterations, we achieved the following results.

Results

The results are presented in a boxplot format. Since the absolute coordinates are related to the origin position of the HoloLens at the moment the program starts, we focus on the range of fluctuation of the detected coordinates rather than their absolute position.

As shown in Figures 4.5, 4.6 and 4.7, the results indicate that the fluctuation range of the target landmarks detected by the HoloLens is relatively large on the Y and Z axes. In the boxplot,

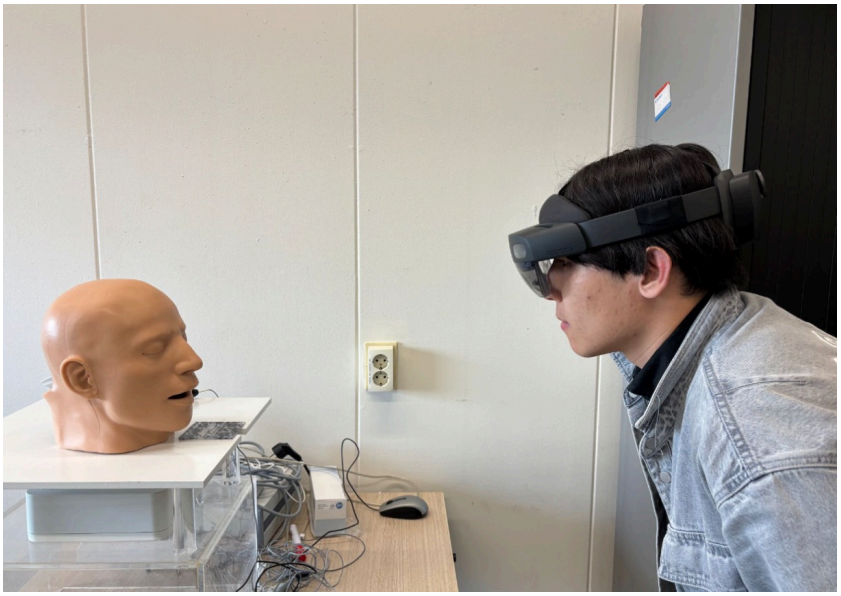


Figure 4.4: User collecting data wearing the HoloLens

the unit for the position coordinates is meters, and the maximum fluctuation range is around 1 cm, which may still be too large and potentially affect the accuracy of the registration.

4.2.4. Accuracy of Dlib-Extracted Target Landmarks

Experimental Setup

The landmarks in the extracted point cloud are the most critical part of the registration algorithm, as their positions determine the accuracy of the registration. To evaluate the accuracy of detecting and extracting landmarks from the point cloud using Dlib, we conducted the following experiment.

In the Open3D coordinate system, registration of the source point cloud is performed based on the detected landmarks and predefined landmarks. However, since the detected landmarks are extracted using Dlib from the HoloLens camera, the accuracy of their physical world positions depends heavily on the depth sensor, the RGB camera, and the overall detection accuracy of Dlib. Therefore, the discrepancy between the detected landmarks in the physical world and the pre-defined landmarks on the head model is particularly significant.

In an ideal scenario, the detected landmarks in the physical world should match the predefined locations on the real head model, enabling highly accurate registration in the simulation experiments. However, due to the limitations of manual calibration and detection accuracy, there may be differences between the detected landmarks and the predefined landmarks, leading to suboptimal registration results.

To address this, we designed an experiment. Small spheres were placed at the positions of the outer eye corners and the nose tip on the Ground Truth virtual model. Similarly, the detected landmarks from the HoloLens were visualized as small spheres projected in the real world, allowing us to observe the discrepancy between the detected landmarks and the head model's predefined positions.

The Ground Truth was chosen as the registration result from the EM tracking system. The EM tracking registration is a point-based registration process, where the coordinates of predefined landmarks on the head model are acquired using an EM-tracking pointer. The specific steps

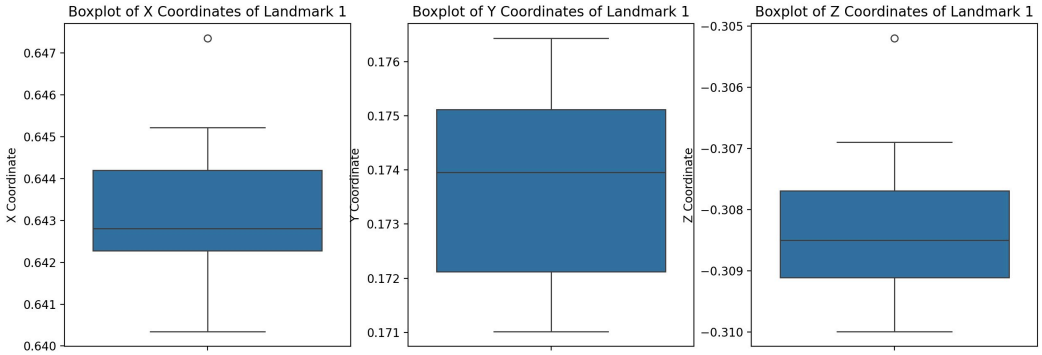


Figure 4.5: Boxplot of the first landmark.

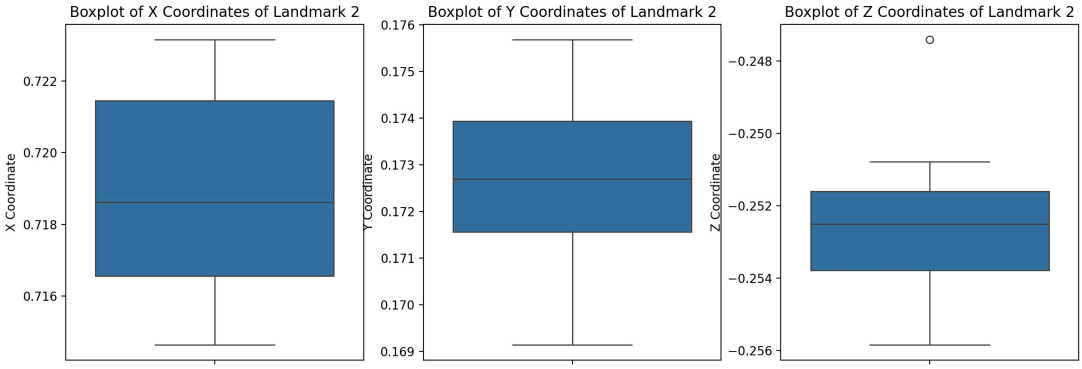


Figure 4.6: Boxplot of the second landmark.

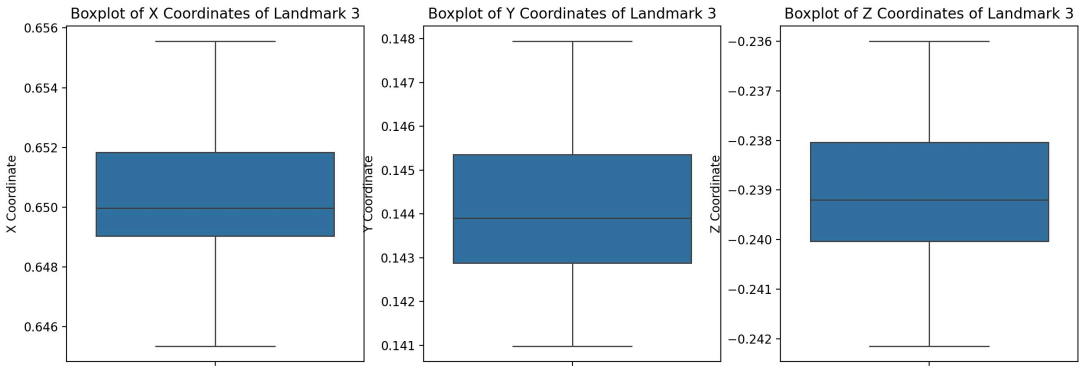


Figure 4.7: Boxplot of the third landmark.

are detailed in the validation section. This registration system, developed by Benmahdjoub et al. [5], offers high registration accuracy, achieving millimeter-level precision. In this experiment, we considered the EM tracking results to be highly accurate when the RMSE (Root Mean Square Error) was less than 1 millimeter The one we chose as the ground truth is 0.8284 mm).

Results

In the experiment, we measured the distances between the three landmarks (visualized as small spheres) detected by the HoloLens and the corresponding spheres on the Ground Truth virtual model. These distances represent the error between the landmarks detected by the HoloLens and the predefined landmarks.

Through multiple experiments (n=14), we calculated the average distance between each pair of landmarks. The results showed that the distances between the landmarks extracted by the HoloLens and the actual predefined landmarks on the Ground Truth were approximately 2 cm, as shown in Table 4.1. This error indicates that the accuracy of the landmarks detected by the HoloLens has some deviations, which may affect the final registration outcome.

Pair	Distance (m)
left outer corner of eye	0.0208 ± 0.0005
right outer corner of eye	0.0184 ± 0.0004
nose tip	0.0200 ± 0.0005

Table 4.1: Distances between ground truth and extracted landmarks

4.3. Experiment 3: Regsitration Validation

This section aims to validate the accuracy of the registration method, with a focus on the analysis of registration errors, including skin shift and skin-based registration errors.

4.3.1. Experimental Setup

To measure the registration accuracy, we designed a Unity program that includes two identical patient head models, each used for separate registration operations during the experiment. The goal of this experiment is to compare the developed registration method with the EM tracking-based registration method. The EM tracking-based method, which utilizes an EM tracking and QR code registration system [5], serves as the ground truth in this experiment.

The EM tracking-based registration system consists of an EM tracking system, a QR code for Vuforia tracking [24], a Hololens, and a PC server. The EM-tracking system includes a hand-held pointer capable of tracking spatial coordinates (as shown in Figure 4.11(a)) and 4.12(a). After calibration, the coordinates of any point touched by the pointer tip are recorded on the PC server. Vuforia's QR code is used to enable visual tracking on Hololens. The process of obtaining ground truth registration based on EM tracking is as follows: First, the Unity server program is launched on the PC server, which sends the calculated transformation matrix to the Hololens client. The Hololens is worn, and the relevant client program is opened, ensuring that the Hololens is successfully connected to the server. Then, Vuforia's visual tracking function is activated using voice commands. Next, the EM tracking pointer is used to sequentially touch pre-selected landmarks in CT space (such as easily identifiable points on the skull), and the PC server records the spatial coordinates of these points. By collecting a sufficient number of landmarks, the registration matrix is calculated and transmitted to the Hololens, completing the registration of the ground truth model, as shown in Figure 4.11(b) and 4.12(b). To ensure that both the skull and head (including the skin) models are in the same CT coordinate system, the skull was segmented from the head model, as shown in Figure 4.8.

(a) Segmented region of skull

(b) 3D visualization of segmented skull

Figure 4.8: Visualization of skull segmentation

In this experiment, skin shift is a factor of particular concern. The landmark-based registration method developed in this study relies on landmarks on the skin surface. Although we aimed to select stable landmarks (such as the outer corners of the eyes and the tip of the nose) for registration, skin displacement relative to the skull can still occur, potentially affecting the registration results. In contrast, the EM tracking-based ground truth registration method is based on the skull, which offers higher accuracy. Since the skull is rigid, the coordinates of the pre-calibrated points relative to the skull in CT space remain unchanged, ensuring the precision of the EM tracking-based registration. However, the skin can shift slightly, especially when made of silicone, which is manually installed on the skull surface. This can lead to displacement, causing misalignment with the pre-calibrated coordinates on the model (as the model treats the skin as rigid, while in reality, it is not). Therefore, measuring skin shift and skin-based registration errors is crucial to understanding the impact of skin movement on registration accuracy.

We also recorded the time required for registration, including data collection time and registration computation time. The time taken to collect landmarks is recorded while using Hololens. During this process, the data collection time may vary significantly due to hardware limitations and differences in the observer's angle and method of operation. Therefore, during the experiment, it is important to ensure consistency in the observation method and angle, maintaining a uniform pattern of movement as in the previous experiment.

The experimental procedure and measurement methods include, first, measuring the skin shift. Using the registered skull and skin models, the displacement of the skin relative to the skull was evaluated. A set of reference points was evenly distributed across the anterior and posterior regions of the model to measure the coordinate differences between the points on the skull and skin registration. These reference points provided a consistent basis to objectively quantify skin displacement during surgery due to movement.

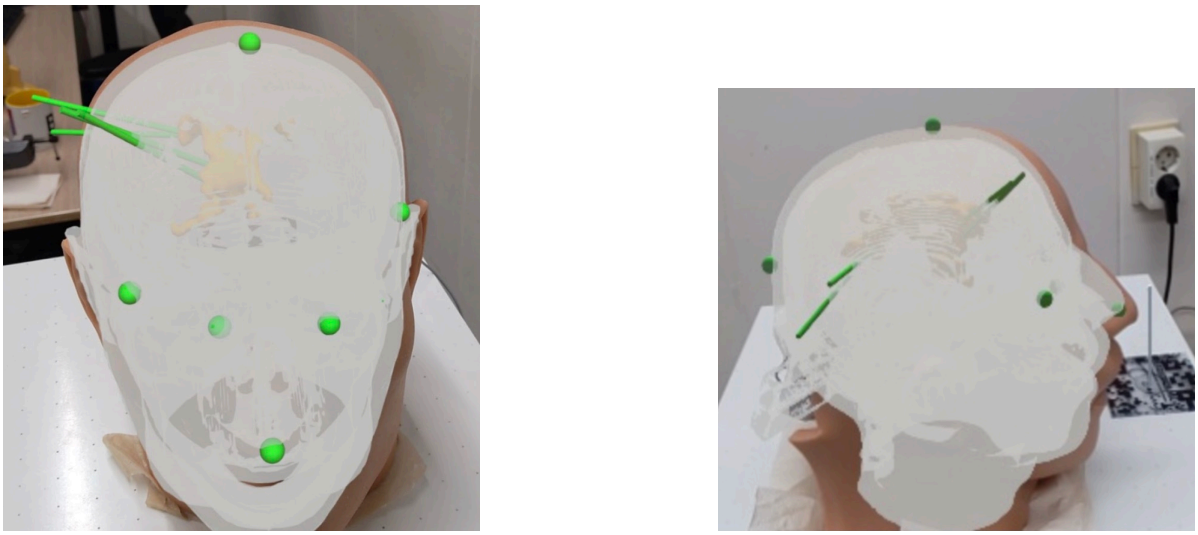
Second, we measured skin-based registration errors. On the Hololens, these reference points were visualized as spheres. The distance differences between the two registration methods were recorded: one based on skin registration using EM tracking and the other using the landmark-based registration method developed in this study. By comparing the two, we aimed

to understand the impact of skin deformation on registration accuracy, helping to optimize medical imaging and intervention techniques.

During the experiment, skull testing must be performed first, followed by skin testing. This arrangement is necessary because the skin tests must be conducted under the same skin displacement conditions to ensure data comparability and reliability. Through this experimental design, we hope to gain a deeper understanding of the role of skin deformation in the registration process, providing valuable insights for improving the accuracy of medical imaging and surgical interventions.

4.3.2. Results

After performing two separate EM tracking-based registrations for the skull and skin, we obtained data on skin shift and registration error. During the registration process, skull registration was performed first, using the EM tracking pointer to collect reference points on the skull and calculate the skull registration transformation matrix. Then, skin registration was performed, using the EM tracking pointer to collect the same reference points on the skin surface while maintaining the position of the skin relative to the skull. The transformation matrices from both registrations were applied to all the reference points, yielding the positions of the reference points in the EM tracking sensor coordinate system. The average distance between the skull and skin registration reference points was calculated, with the results repeated over 14 trials, as shown in Table 4.2.



(a) Registration front view

(b) Registration side view

Figure 4.9: Registration Result

In terms of registration validation, the data was collected by comparing the ground truth and the reference spheres from the depth-based automatic registration method within the same coordinate system (Unity). The displacement differences at each reference point were calculated, yielding data on skin shift. The same observation method from Experiment 2 was used for data collection, and the outcomes are displayed in Figure 4.9. The average distance between the reference spheres in the registration results and the reference spheres in the ground truth was calculated, and the target registration error (TRE) was determined. Figure 4.10 shows the overlap of the registration results and the ground truth. The red reference points represent ground truth-based registration, while the green reference points represent the depth-based automatic registration method.

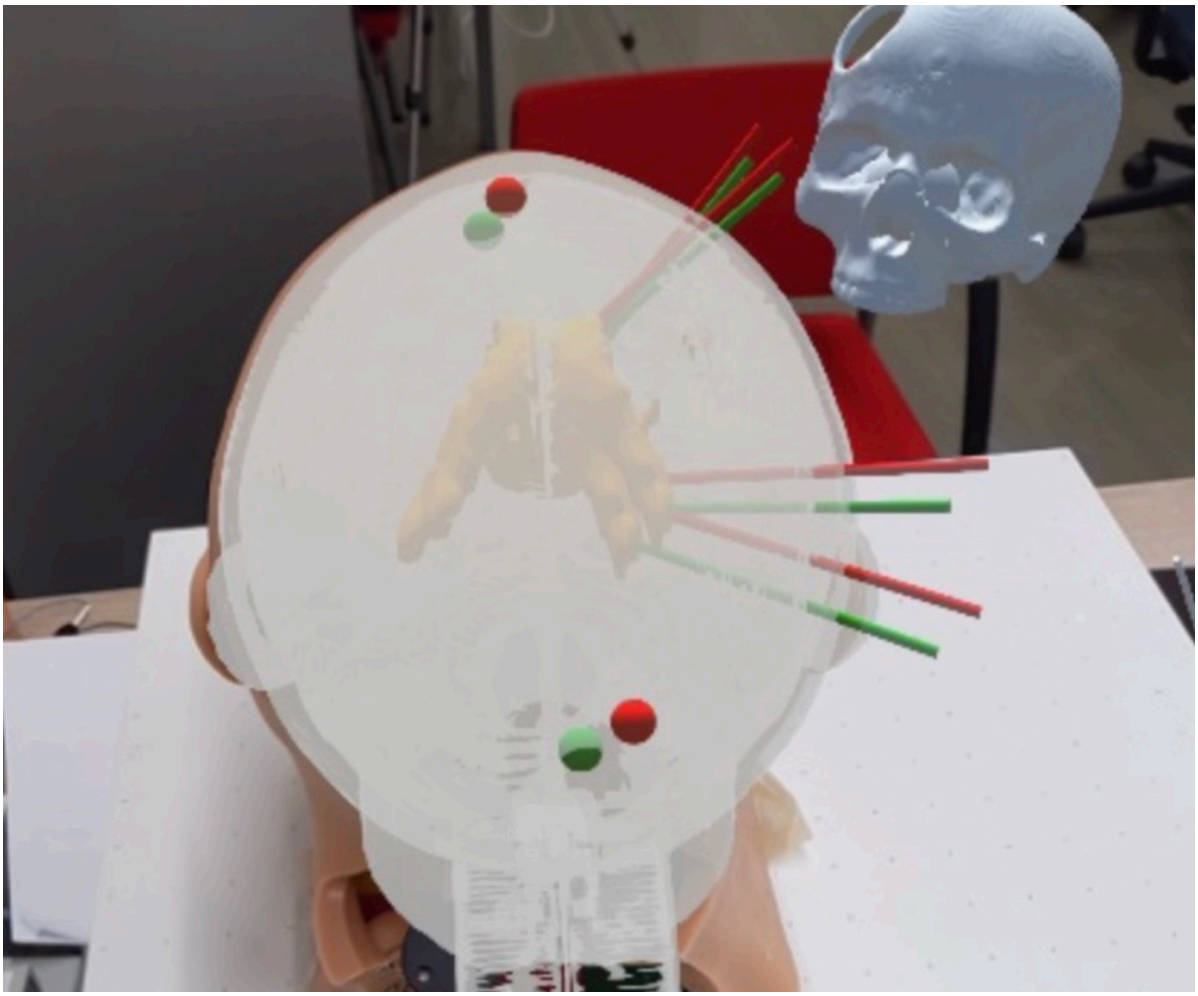
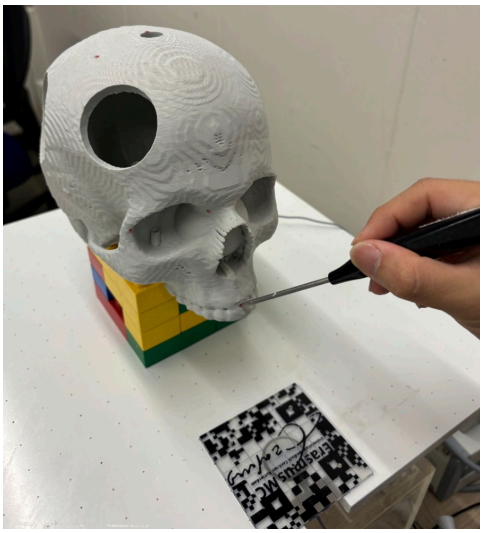
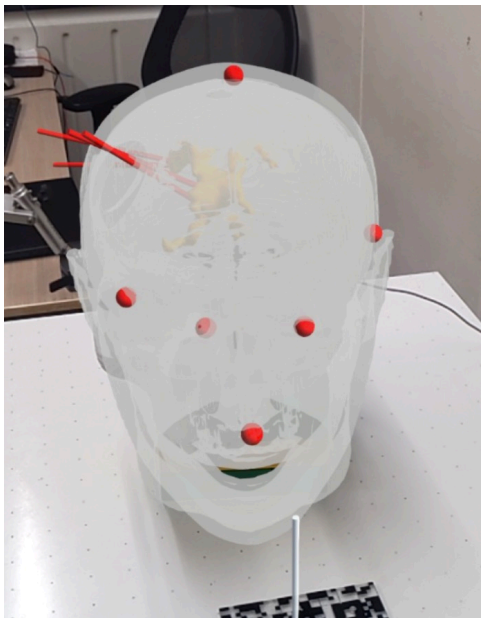


Figure 4.10: Visualization of the ground truth and registration result

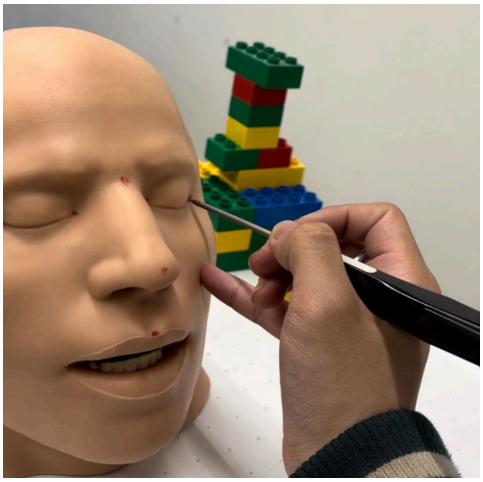


(a) EM-tracking collecting skull data



(b) EM-tracking registration visualization with real skull

Figure 4.11: Skull registration result



(a) EM-tracking collecting skin data

(b) EM-tracking registration visualization with skin

Figure 4.12: Skin registration result

The table below (Table 4.2) summarizes the experimental results, including the skin displacement relative to the skull, registration error, data collecting time, and registration time.

Skin Shift (mm)	Registration Error (TRE) (mm)	Data Collecting Time (s)	Registration Time (s)
6.14 ± 0.68	16.41 ± 3.06	30.21 ± 12.43	6.59 ± 0.73

Table 4.2: Skin shift, registration error, data collecting time, and registration time results.

These results indicate that skin movement has a potentially significant impact on registration accuracy. Since the skin is flexible and subject to external forces or deformation, the displacement of the skin relative to the skull reached around 6 mm. The high registration error reflects the negative effect of skin displacement on registration accuracy, validating the need to carefully consider skin deformability when performing landmark-based registration on the skin surface.

The registration method developed in this study is based on rigid body registration, aiming to align the virtual skull model with the real skull accurately, thus achieving precise localization of target tissues, such as the ventricles. However, the flexible nature of the skin introduces larger errors in skin-based registration methods.

Regarding registration time, multiple experiments revealed that collecting landmarks with the Hololens generally takes about 30 seconds. Although there were some fluctuations in time, the overall efficiency was high. After completing the landmark collection, the time required for registration computation was relatively stable, typically taking around 6 seconds.

4.4. Experiment 4: Evaluating EVD Insertion

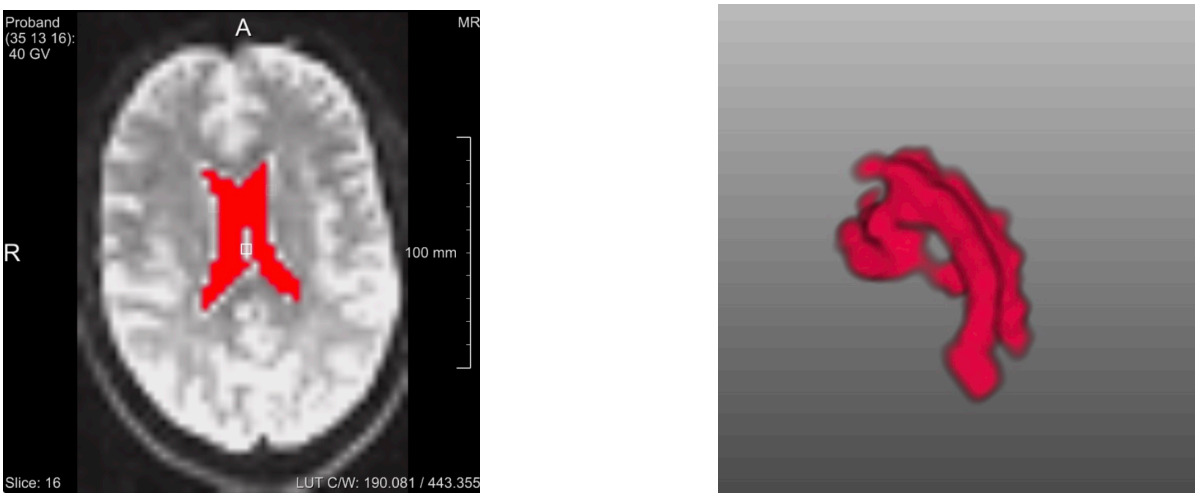
This section evaluates the accuracy of participants' performance in simulating an EVD (External Ventricular Drain) insertion surgery using the depth-based automatic registration method and discusses the results.

4.4.1. Experimental Setup

The entire workflow of alignment using the Hololens registration is applied to the EVD insertion surgery process in this experiment, serving as a navigation tool during the surgery. Preoperative skull and ventricle data are visualized, allowing a clear display of the head model and ventricles in the software for subsequent registration. To obtain an accurate ventricle model, the ventricles within the brain were segmented, as shown in Figure 4.13. This step is crucial for the surgical navigation process. Effective registration aligns the head model, including the skin, with the real-world head model from all angles. Ideally, when no tissue or skin movement occurs, the internal ventricles of the head should also align with the real-world ventricle tissue, aiding participants in clearly visualizing the internal ventricles during insertion and improving hand-eye coordination.

We simulated the placement of the EVD using a medical needle, which can be tracked by EM tracking, in a model head. The model head consists of two parts: an internally 3D-printed skull model and a silicone-based skin layer. The inside of the skull was injected with gelatin and frozen to form a soft structure simulating brain tissue. The role of the simulated brain tissue is to ensure that the needle path remains unchanged during insertion, similar to a real surgery, to prevent any deviation in the experimental results. As shown in Figure 4.14(a), the skin is cut open to reveal the gelatin inside the skull. In subsequent experiments, a new skin without holes will be used, as shown in Figure 4.14(b).

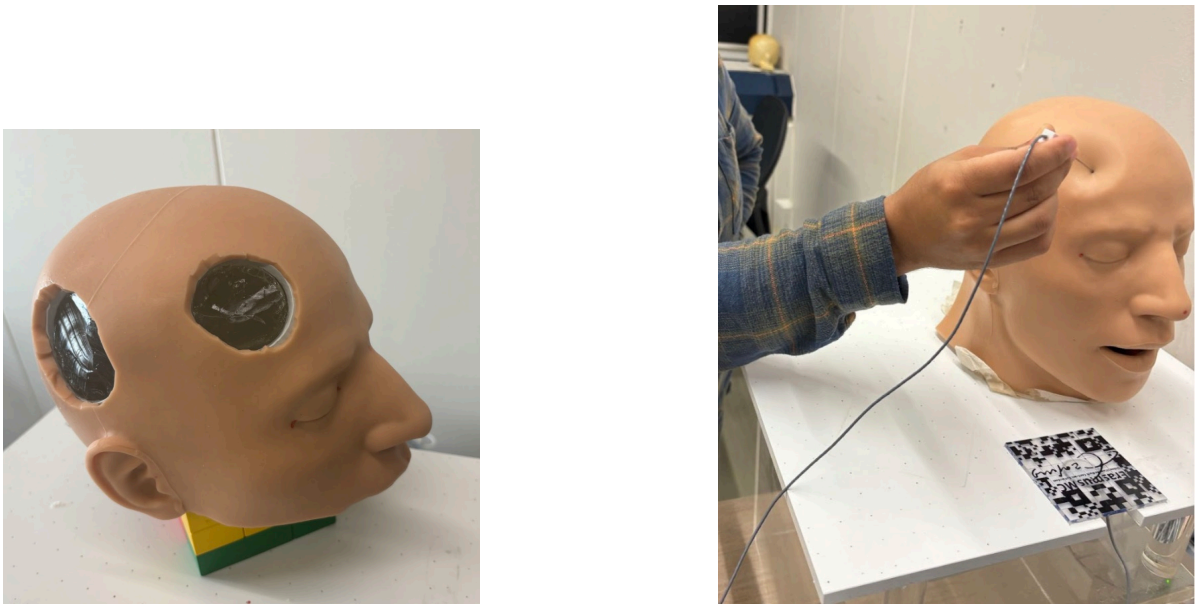
During the navigation process, after the head is aligned, a visualization guide further assists



(a) Segmented region of ventricle

(b) 3D visualization of segmented ventricle

Figure 4.13: Segmentation of ventricle



(a) Gelatin in hole

(b) User inserting needle

Figure 4.14: Simulation of EVD insertion

participants in navigating the surgery. Before the surgery, an insertion path is pre-planned on the virtual head model in the Hololens. The visualized insertion path extends outside the head and makes contact with the internal ventricles, simulating a realistic EVD insertion. Additionally, the EM tracking system tracks the medical needle used for the EVD insertion, and a real-time tracking pointer is visualized on the needle. This pointer is also visualized as a cylinder but with a different color to distinguish it from the guided insertion path. When participants simulate the EVD insertion, they simply need to align the tracked pointer with the green cylinder representing the guided path, as shown in Figure 4.15.

Experimental steps are shown as follows:

- Registration Process: Participants wear the Hololens and, through observation and adjustment, align the virtual head model with the real head model at all angles. Once participants are satisfied with the registration result, they proceed to the next step.
- Hide Head Model: Using voice commands, participants hide the registered head model, leaving only the visualized ventricle model, insertion path, and some reference points, making it easier to focus on the surgical target area and improve accuracy.
- Prepare Insertion Tool: The medical needle tracked by the EM tracking system is prepared, with a real-time visualized pointer (a white cylinder) displayed on the needle.
- Perform Insertion Simulation: Participants insert the needle into the head model according to the visualized insertion path. During insertion, they attempt to align the tracked pointer (white cylinder) with the pre-planned insertion path (green cylinder). Once the participants achieve a satisfactory alignment, they use voice commands to save the insertion experiment records, including translation and rotation information. This data will be used for subsequent error analysis and result evaluation.

4.4.2. Results

During the experiment, five different participants were selected, each performing four insertions at different positions, as shown in Figure 4.15. Among the five participants, three individuals had experience using the Hololens.

To comprehensively evaluate the accuracy of the insertion path in a controlled experimental environment, we used two models and a tracking cylinder. First, the Ground Truth model (EM Tracking method) includes a cylinder representing the reference insertion path, serving as a baseline for validating the accuracy of other methods. Second, the registration model (depth-based automatic registration method) includes a cylinder representing the insertion path, which is used to evaluate the effectiveness of the landmark-based registration method. Finally, the actual path indicator (tracked pointer cylinder) simulates the actual insertion path of the device, tracked in real-time by the EM tracking system, reflecting the participants' actual performance.

The criteria for evaluating the accuracy of the registration and insertion are as follows (see also Figure 4.16):

- Total Error:
  - Measures the distance between the Ground Truth model's cylinder(red) and the tracked pointer cylinder (white, actual insertion path).
  - Reflects the overall deviation of the actual insertion path from the expected path.
- Perception Error:

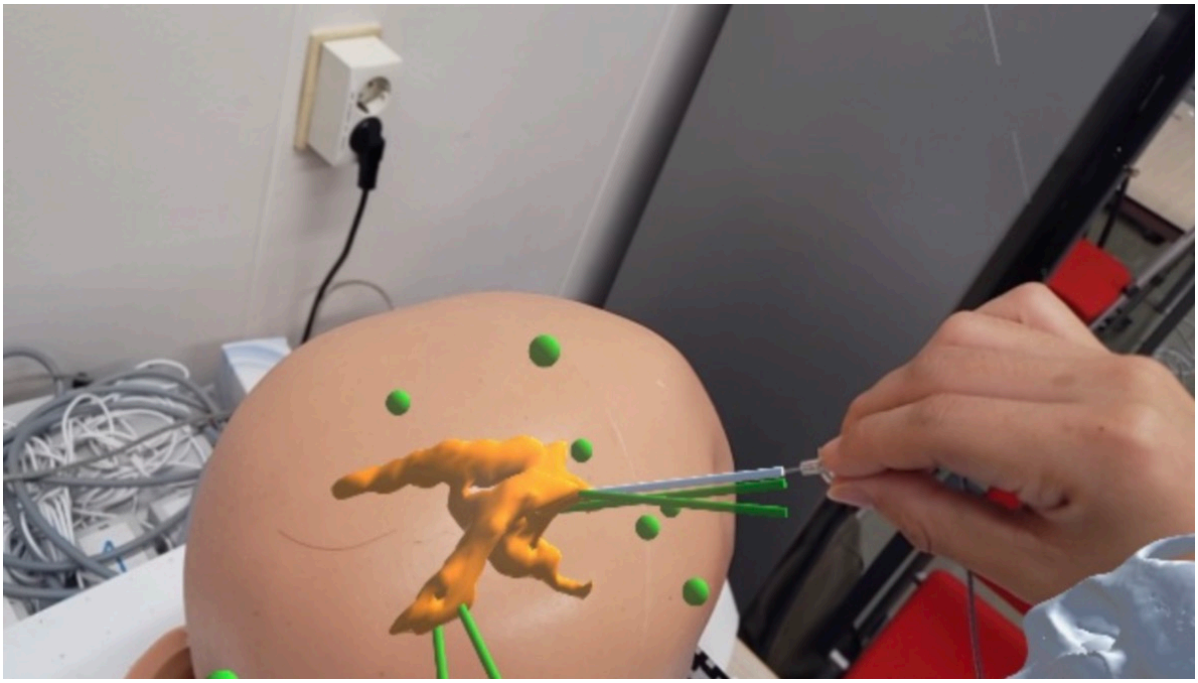


Figure 4.15: Insertion from user view

  - Measures the distance between the registration model's cylinder (green, landmark-based insertion path) and the tracked pointer cylinder (white, actual insertion path).
  - Indicates the discrepancy between the participants' perceived path and the actual insertion path when using the landmark-based registration method.
- Registration Error:
  - Measures the distance between the registration model's cylinder (green, landmark-based insertion path) and the Ground Truth model's cylinder (red, reference path).
  - Reflects the deviation of the landmark-based planned path from the reference path.

registration error		perception error		total error	
translation (mm)	rotation (°)	translation (mm)	rotation (°)	translation (mm)	rotation (°)
15.17 (± 3.24)	5.82 (± 2.71)	5.10 (± 3.94)	3.19 (± 2.59)	16.07 (± 4.33)	7.14 (± 2.67)

Table 4.3: Insertion error with translation (in mm) and rotation (in degrees)

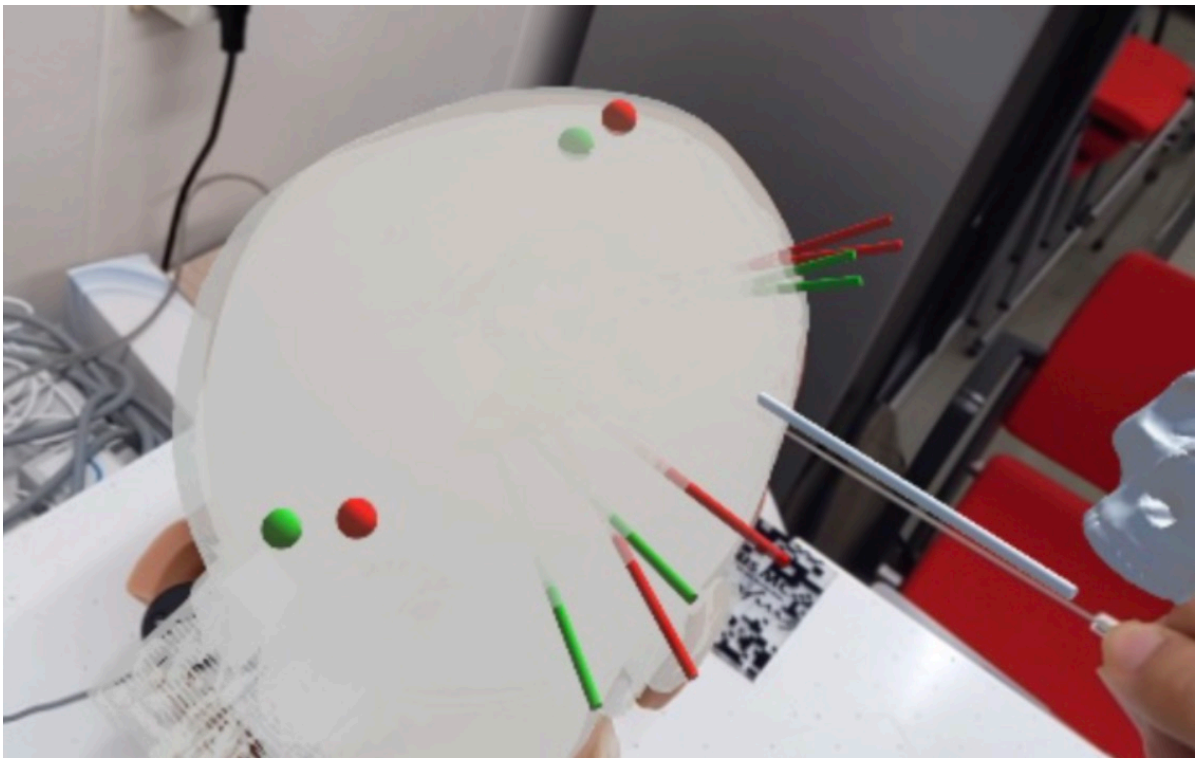


Figure 4.16: Ground truth, navigation result and tracking needle

5 Discussion

Currently, the robustness of the registration algorithm may be insufficient, and its stability is difficult to evaluate. The registration algorithm in use is relatively simple, deriving the registration transformation matrix through stepwise affine transformations between two triangles. Although the algorithm can demonstrate high accuracy when the three pairs of registration points coincide exactly, errors still persist. These errors may result from the limited precision during rotation when the algorithm is implemented in code, especially in the calculation of small angles. Additionally, during the registration process, the landmarks marked on the source point cloud and the point cloud obtained through the Hololens do not form identical triangles, leading to cumulative errors. For example, in the second step, the alignment of the lines is achieved by making them parallel, while the precision of the rotation in the third step further introduces errors. It is also worth noting that this triangle-based registration algorithm is sensitive to the order of points, meaning that different point orders can lead to different registration outcomes. In some cases, using a different point order might produce better registration results, which is another source of error.

When applying the Hololens for registration in practice, the accuracy is influenced by several factors at both the system and physical levels.

At the system level, for instance, the landmarks pre-marked on the head model may not perfectly match the positions of the landmarks detected in the actual point cloud. For the registration algorithm, as long as the virtual and real landmarks maintain the same relative position to the head, accurate registration can theoretically be achieved. However, in practice, the pre-marked points (source point cloud) and the actual points obtained (target point cloud) cannot match perfectly, resulting in registration errors. Furthermore, the visualization process in the Hololens is subject to SLAM drift issues, meaning that after prolonged use of the Hololens, the model's position may shift.

At the physical level, factors such as skin movement and real-time tissue displacement can affect both the registration process and the final surgical outcome. Skin movement may cause a "skin shift," where the skin moves relative to the skull, causing facial feature points to shift accordingly. Since registration is based on detected facial feature points, even if the predefined landmarks on the virtual head model align accurately with the detected landmarks, the actual registration between the skull and the brain's ventricles may not be as accurate.

Regarding registration time, it is currently unstable. During the experiments, instability in data collection caused significant fluctuations in the time required for data acquisition. Improving the stability and efficiency of point cloud data acquisition is an issue that needs to be addressed.

It is also important to note the limitations of Dlib detection. Dlib only enters the landmark detection loop when it detects a complete face. If facial features are not prominent or are missing for some reason, it is possible that Dlib will not detect a face, thereby failing to perform landmark detection. In the experiment, an attempt was made to use a head model without a jaw for registration testing, but since Dlib could not detect facial features, the registration failed.

In the EVD insertion experiment, the results varied among different participants due to their varying levels of experience with the Hololens and surgery. Participants with more experience were more familiar with operating Hololens and had a better understanding of the experimental procedures and instructions, which resulted in more accurate registration outcomes. Before performing EVD insertion, registration needs to be completed, and only when participants are satisfied with the registration results can they proceed with the insertion. The experiment revealed that participants with experience in simulated surgeries using the Hololens had higher standards for registration accuracy, resulting in better registration precision during the simulated surgery. The overall results showed that inexperienced participants had higher displacement and angle errors, while experienced participants had lower errors.

Additionally, besides registration accuracy, the Hololens' visualization and depth perception capabilities had a significant impact on the outcome of the EVD insertion. Participants reported after the experiment that the alignment they observed from different angles in the Hololens varied, and depth perception posed certain challenges, requiring multiple observations from different angles to achieve a satisfactory insertion. During the experiment, participants with varying levels of prior knowledge about Hololens exhibited different behaviors. For instance, participants with prior experience in using Hololens for EVD insertion showed smaller perception errors, indicating that experienced participants were better able to utilize the Hololens' visualization capabilities to find the correct position.

6 Conclusion

This study proposes a markerless, external-device-free automatic registration method based on the HoloLens 2, specifically designed for augmented reality (AR) navigation in External Ventricular Drainage (EVD) surgeries. The motivation for the study stems from the limitations of current registration methods used in EVD procedures. Traditional marker-based methods are complex and time-consuming, especially in emergency surgical scenarios where high efficiency and real-time performance are essential. Therefore, we developed a novel approach that utilizes the Dlib library to automatically detect facial landmarks, combined with point cloud registration technology, simplifying the registration process while enhancing the precision and efficiency of surgical navigation.

In the experimental section, we designed and conducted four experiments to validate the effectiveness of this method. First, we verified the feasibility of the algorithm through a point cloud registration experiment based on landmarks. Despite manual errors in marking the landmarks, the algorithm was able to achieve high registration accuracy, demonstrating the robustness of the method. Second, we tested the accuracy and stability of Dlib's detection and evaluated the performance of HoloLens cameras in extracting facial landmarks. Although small movements of facial features introduced minor errors, the overall detection accuracy met clinical requirements. The third experiment assessed registration accuracy by analyzing the impact of skin displacement on registration errors. Results showed that skin displacement of about 6mm caused registration errors of up to 16.41mm, indicating that skin flexibility significantly affects registration accuracy during surgery. Lastly, we validated the accuracy of the insertion path in a simulated EVD insertion experiment. Results showed that participants' experience with using HoloLens significantly influenced registration and insertion accuracy, with more experienced users achieving better precision.

The main contribution of this study is providing an efficient, markerless, external-device-free automatic registration method for EVD surgeries, significantly reducing dependence on external devices and simplifying the registration process. This method demonstrated reasonable accuracy and real-time performance in the experiments, making it suitable for emergency surgical scenarios with potential for broad clinical applications. Moreover, the experiments revealed the impact of skin displacement on registration accuracy, providing valuable insights for further optimization of the registration algorithm.

Future research will focus on optimizing the algorithm to reduce the effects of skin displacement and landmark detection on registration accuracy, further improving precision. Additionally, we will explore applications in other complex neurosurgical procedures. With continuous improvements in robustness, real-time performance, and accuracy, we believe this method will drive further exploration of depth-based automatic registration techniques.

References

[1] Abdullah AlAzri et al. “Placement Accuracy of External Ventricular Drain When Comparing Freehand Insertion to Neuronavigation Guidance in Severe Traumatic Brain Injury”. In: Acta Neurochirurgica 159.8 (2017), pp. 1399–1411. DOI: 10.1007/s00701-017-3201-5.

[2] Various authors. “NeuroLens: Augmented Reality-based Contextual Guidance through Surgical Tool Tracking in Neurosurgery”. In: Conference Proceedings. N/A.

[3] Tadas Baltrusaitis, Peter Robinson, and Louis-Philippe Morency. “OpenFace: An Open Source Facial Behavior Analysis Toolkit”. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV). 2016, pp. 1–10. DOI: 10.1109/WACV.2016.7477553.

[4] Daniel Barath, Luca Cavalli, and Marc Pollefeys. “Learning To Find Good Models in RANSAC”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022, pp. 15744–15753. URL: https://openaccess.thecvf.com/content/CVPR2022/papers/Barath_Learning_To_Find_Good_Models_in_RANSAC_CVPR_2022_paper.pdf.

[5] Mohamed Benmahdjoub et al. “Multimodal markers for technology-independent integration of augmented reality devices and surgical navigation systems”. In: Virtual Reality 26 (May 2022). DOI: 10.1007/s10055-022-00653-3.

[6] Jon L. Bentley. “Multidimensional Binary Search Trees Used for Associative Searching”. In: Communications of the ACM 18.9 (1975), pp. 509–517. DOI: 10.1145/361002.361007.

[7] Luca Cavalli et al. “Consensus-Adaptive RANSAC”. In: arXiv preprint arXiv:2307.14030 (2023). URL: https://doi.org/10.48550/arXiv.2307.14030.

[8] Luca Cavalli et al. “Consensus-Adaptive RANSAC”. In: arXiv preprint arXiv:2307.14030 (2023). URL: https://arxiv.org/abs/2307.14030.

[9] Laura Cercenelli et al. “Augmented Reality in Orthognathic Surgery: A Multi-Modality Tracking Approach to Assess the Temporomandibular Joint Motion”. In: International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS). 2023, pp. 379–394. DOI: 10.1007/978-3-031-43404-4_25.

[10] Juan Dibene and Enrique Dunn. HoloLens 2 Sensor Streaming. Nov. 2022. DOI: 10.48550/arXiv.2211.02648.

[11] “Face Emotion Expression Recognition Using DLIB Model and Deep Learning”. In: Springer (2023).

[12] Martin A. Fischler and Robert C. Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: Communications of the ACM 24.6 (1981), pp. 381–395. DOI: 10.1145/358669.358692.

[13] Herbert I. Fried et al. “The Insertion and Management of External Ventricular Drains: An Evidence-Based Consensus Statement”. In: Neurocritical Care 24.1 (2016), pp. 61–81. DOI: 10.1007/s12028-015-0224-8.

[14] Herbert I. Fried et al. “The Insertion and Management of External Ventricular Drains: An Evidence-Based Consensus Statement”. In: Neurocritical Care 24.1 (2016), pp. 61–81. DOI: 10.1007/s12028-015-0224-8.

[15] Christina Gsaxner, Alessandro Pepe, and Joachim Wallner. “Markerless Image-to-Face Registration for Untethered Augmented Reality in Head and Neck Surgery”. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI). 2019, pp. 236–244. DOI: 10.1007/978-3-030-32254-0_27.

[16] Dhrubodhi Guha, Naser Alotaibi, and Victor X. D. Yang. “Augmented Reality in Surgical Guidance for Neurosurgery”. In: Journal of Neurosurgery (2023). DOI: 10.3171/2023.9.JNS21510.

[17] Vahid Kazemi and Josephine Sullivan. “One Millisecond Face Alignment with an Ensemble of Regression Trees”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society. 2014, pp. 1867–1874. DOI: 10.1109/CVPR.2014.241.

[18] Enzo Kerkhof. “Depth based registration of 3D preoperative models to intraoperative patient anatomy using the HoloLens 2”. Master’s thesis. MA thesis. Delft University of Technology, 2023. URL: https://resolver.tudelft.nl/uuid:dde65ebe-eb0b-4747-b89f-b1af83020e09.

[19] Davis E. King. “Dlib-ml: A Machine Learning Toolkit”. In: Journal of Machine Learning Research 10 (2009), pp. 1755–1758. URL: https://jmlr.org/papers/v10/king09a.html.

[20] Haowei Li et al. “EVD Surgical Guidance with Retro-Reflective Tool Tracking and Spatial Reconstruction using Head-Mounted Augmented Reality Device”. In: (June 2023). DOI: 10.48550/arXiv.2306.15490.

[21] Valter Piedade and Pedro Miraldo. “BANSAC: A Dynamic Bayesian Network for Adaptive Sample Consensus”. In: arXiv preprint arXiv:2309.01530 (2023). URL: https://doi.org/10.48550/arXiv.2309.01530.

[22] Szymon Rusinkiewicz and Marc Levoy. “Efficient Variants of the ICP Algorithm”. In: Proceedings of the 3rd International Conference on 3D Digital Imaging and Modeling (3DIM) (2001). URL: https://graphics.stanford.edu/papers/fasticp/fasticp_paper.pdf.

[23] Max Schneider et al. “Augmented reality–assisted ventriculostomy”. In: Neurosurgical Focus 50 (Jan. 2021), E16. DOI: 10.3171/2020.10.FOCUS20779.

[24] Vuforia Engine: Augmented Reality SDK. Accessed: 2024-09-27. 2024. URL: https://developer.vuforia.com.

[25] Jih-fang Wang et al. “Tracking a Head-Mounted Display in a Room-Sized Environment With Head-Mounted Cameras”. In: Proceedings of SPIE - The International Society for Optical Engineering (Feb. 1990). DOI: 10.1117/12.20954.

[26] Tian Xie et al. “Holographic iRay: Exploring Augmentation for Medical Applications”. In: 2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR) Adjunct. 2017, pp. 220–222. DOI: 10.1109/ISMAR-Adjunct.2017.73.

[27] Q.-Y. Zhou, Jaesik Park, and Vladlen Koltun. “A Modern Library for 3D Data Processing”. In: 2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR). 2018, pp. 110–113. DOI: 10.1109/ISMAR.2018.00021.

[28] Q.-Y. Zhou, Jaesik Park, and Vladlen Koltun. “Open3D: A Modern Library for 3D Data Processing”. In: arXiv preprint arXiv:1801.09847 (2018). URL: https://doi.org/10.48550/arXiv.1801.09847.