UNIVERSITÀ POLITECNICA DELLE MARCHE

Department of Information Engineering (DII)

Master of Science in Biomedical Engineering

Development of an Augmented Reality system based on marker tracking for robotic-assisted minimally invasive spine surgery

Supervisor: Prof. Emanuele Frontoni

Co-Supervisor: Sara Moccia, PhD

Author:

Francesca Pia Villani

Academic Year 2019 - 2020

Ringraziamenti

Vorrei esprimere la mia gratitudine verso tutti coloro che mi hanno sostenuta durante questo percorso aiutandomi direttamente o indirettamente a sviluppare questo lavoro.

Ringrazio innanzitutto il Professor Frontoni, relatore di questa tesi, per la fiducia che mi ha riservato e per avermi offerto la possibilità di lavorare a questo stimolante progetto consentendomi di ampliare le conoscenze in un ambito quasi sconosciuto e all'inizio spaventoso!

Un enorme ringraziamento, dal più profondo del cuore va a Sara, per la pazienza, la disponibilità e l'inestimabile aiuto che mi ha dato in questi mesi e senza il quale difficilmente questa tesi esisterebbe.

Grazie a Manni per aver condiviso con me questo percorso dal primo all'ultimo giorno, posso affermare con assoluta certezza che senza di lei non sarebbe stato lo stesso e soprattutto ora non conoscerei la parola "Lasai"!

Grazie a Salvì per esserci sempre stato, per aver creduto in me ogni giorno e avermi aiutato a superare i momenti difficili.

Infine, il ringraziamento più speciale va ai miei genitori e alle mie sorelle, la consapevolezza del loro amore, della loro presenza e della loro fiducia in me è stata giorno per giorno spinta per affrontare questo percorso e arrivare fin qui.

Abstract

Spine surgery is performed nowadays for a great number of spine pathologies including degenerative disorders, neoplasms, infections, trauma, inflammatory arthropathies and congenital malformations.

The incidence of spinal disorders has undergone a drastic increase in recent years, reaching epidemic proportions. It is estimated that globally, 4.83 million spinal surgeries are performed annually. This prevalence led, over the past few decades, to an evolution of spine surgery into an extremely specialized field.

In this scenario, traditional open interventions to the spine have been integrated and often substituted by minimally invasive approaches. Minimally invasive surgeries (MIS) -emerged as an alternative to open spine procedures- are characterized by small surgical incisions, surrounding tissue spare and intraoperative monitoring. This approach has been associated with less surgical-related morbidity, lower complication rate and shorter recovery time and postoperative hospital stay. On the other hand, the main MIS disadvantages are loss of depth perception, reduced field of view and consequent difficulty in intra-operative identification of relevant anatomical structures. Together with MIS, surgical robots emerged during the '90s and since then they have been used in spinal surgeries to enhance and complement the surgeon's abilities, providing a 3D view, an increased accuracy of implant placement while decreasing invasiveness and complications, and reducing radiation exposition both for the patients and operating staff. However, robotic MIS suffers from the same drawbacks of MIS.

To overcome these drawbacks, recently, Augmented Reality (AR) has been introduced in surgical applications. AR refers to the superimposition of virtual elements on the intra-operative scene. AR can support the surgeon providing intra-operative guidance, information matching and identification of the relevant structures, allowing to reduce the amount of unnecessary damage to the patient. However, even though the irruption of AR has promised breakthrough changes in surgery, providing surgeons with unprecedented visualization capabilities of graphical information displayed in real time directly over the surgeons' field of view, its adoption has been slower than expected as there are still usability hurdles for which no appropriate solutions exist. These include the inability of wearable visualization devices to render large datasets, insufficient framerates and unnatural delays, lack of integration with surgical equipment and poor precision in patient motion tracking.

To overcome these problems, in this thesis a client-server architecture is proposed, on which computationally intensive tasks are offloaded on remote servers, leaving the wearable devices in charge of rendering the final frames transmitted as a video stream. This architecture will be coupled with high-precision tracking devices, capable of tracking the patient motion with respect to a fixed frame of reference.

This project mainly addresses the development of the client software, deployed on the Microsoft HoloLens 1 headset, which includes marker tracking capabilities and communication with the rendering servers via frames stream in real-time. The tracking algorithm has been implemented in C++ using open source computer vision libraries as OpenCV and ArUco. Once obtained a working algorithm several tests including overlap-based metrics, evaluation of robustness to lighting conditions and external noise have been performed to assess the stability of the system. Then the client software has been included in a communication architecture which makes use of the WebRTC (Web Real-Time Communications) protocols to enable real-time streaming over the network and provide desktop rendering power to HoloLens.

Results obtained are promising and although there is yet room for improvement and further research is needed, the current state of the application provides an evidence that this architecture may be implemented with positive outcomes in the field of spinal robotic assisted MIS.

Contents

Chapter 1

INTRODUCTION

The incidence of spinal disorders has undergone a drastic increase in recent years, reaching epidemic proportions. Common spine pathologies that require surgical intervention include degenerative disorders, neoplasms, infections, trauma, inflammatory arthropathies, and congenital malformations. Interventions performed to treat these conditions are often classified according to their level of invasiveness, as simple decompression, simple and complex spine fusion[1]. An example of spinal fusion can be seen in Figure 1.1. Spinal disorders have a high financial impact on society, causing pain and leading to a significant impairment of patients' quality of life and mobility [2].

Common spine disorders include:

- Degenerative diseases, which usually affect midcervical and lumbar regions, are the most common indication for spine surgeries [4]. These disorders progress as continuous changes, which start as a soft disk herniation, progress to spinal instability and/or spondylolisthesis, and eventually cause spinal stenosis and spondylotic myelopathy. Degenerative diseases of cervical spine are usually treated with decompression of affected nerve roots followed by stabilization and restoration of the column segments, while degenerative disorders of the lumbar spine are often treated with diskectomy.

- Conditions associated with spinal instability which can occur due to various reasons, such as trauma, tumors, infection, degenerative disorders, inflammatory

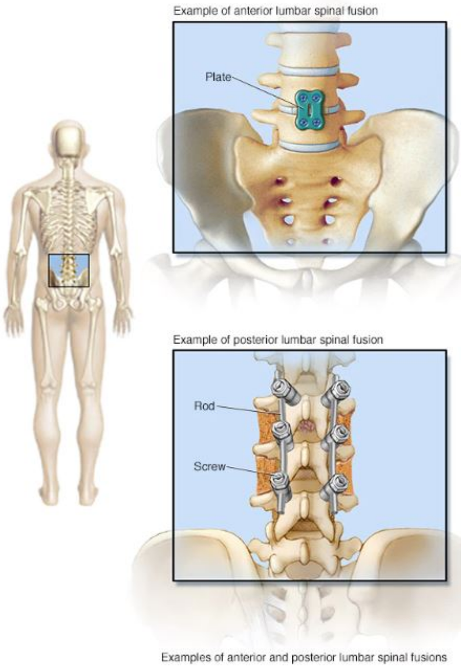

Example of anterior lumbar spinal fusion
Plate
Example of posterior lumbar spinal fusion
Rod
Screw
Examples of anterior and posterior lumbar spinal fusions

Figure 1.1: *Example of spinal fusion. Depending on the surgical approach, spinal fusion can be either anterior or posterior. With this procedure the vertebrae are fastened together with a metal plate or with rods and screws. Figure adapted from [3].*

diseases, or congenital conditions.

- Neoplasms of the spine such as metastatic extradural tumors (most common spine tumors) which usually involve the thoracic vertebral bodies; metastasis occurs from breast, lung, renal, gastric, or hematopoietic/lymphoid tissue.

- Infections of the spine: vertebral osteomyelitis and epidural abscesses usually affect thoracic and lumbar spine. Vertebral osteomyelitis typically interests anterior and middle columns and mostly occurs due to hematogenous spread or bacteria in vertebral bodies, if not treated it can spread further to involve adjacent vertebrae leading to destruction of vertebral bodies with vertebral collapse and spinal instability.

- Spinal deformity as scoliosis, a complex rib cage deformity characterized by increased lateral curvature of the spine (angle of curvature >10 degrees). Scoliosis can occur due to various causes: idiopathic, neuromuscular (NM), congenital, or secondary to spinal pathologies. Scoliosis correction surgery is often challenging, because of the risks of excessive blood loss and neurological injury [4].

In the United States, the surgical approach to treat lumbar spinal stenosis has been the fastest growing indication since the 1980s. In fact, the highest rates of spine surgery in the world have been reported by the United State despite similar incidence and prevalence rates of spine disorders are found when compared with other Countries [2]. It is estimated that globally, 4.83 million spinal operations are performed annually, while 1.34 million of these surgeries are performed in the United States [5].

This growing trend has been related to the aging population with greater disease prevalence, improved diagnostic modalities, development of new surgical techniques, and an increased number of spine surgeons [6].

In an aging society, the high demand for healthy life expectancy is likely to increase the need for surgeries for spinal degenerative diseases in the years to come [7].

1.1 Spine surgery

Spine surgeries are usually electively performed for a variety of conditions, which can vary in complexity from minimally invasive single level microdiskectomy to extensive multisegment instrumentations, often performed thorough combined approaches and in multiple stages. Over the past few decades, spine surgery has evolved into an extremely specialized field; highly complex procedures are increasingly being performed, across all age groups, and often through minimally invasive approaches [4].

Different kind of interventions to treat spinal disorders exist and can be differentiated according to the pathologies to treat and to the different level of invasiveness.

The aim of spinal decompression procedures is to increase the functional space for compressed neural elements. Spinal fusion involves joining of two vertebrae by insertion of graft material into the decompressed site. It has two major indications: first for the management of disorders that compromise the structural integrity of the spine (degenerative and inflammatory pathologies or vertebral fractures) and second, in combination with decompression procedures, to stabilize the spine when its native stability has been compromised [4].

Internal spine fixation is used to reconstruct compromised columns within a spinal motion segment (two adjacent vertebrae and their interconnecting tissues), to provide temporary immobilization and provisional stability, until osseous fusion occurs across the affected spinal levels. Various devices such as screws, wires, plates, and rods are used for this purpose. Lumbar interbody fusion procedures (fusion of at least two vertebrae) are usually indicated in patients with scoliosis, spondylolisthesis, spinal fractures, or multiple severe degenerative disk disease; to restore and stabilize the sagittal alignment of the spine, and to divert the neuroforaminal space. Lumbar interbody fusion procedures include: Posterior lumbar interbody fusion (PLIF), transforaminal lumbar interbody fusion (TLIF), Anterior lumbar interbody fusion (ALIF), lateral lumbar interbody fusion (LLIF) and axial lumbar interbody fusion (Ax-LIF) [4].

Newer techniques such as arthroplasty, nucleoplasty, dynamic stabilization, avoid fusion altogether and attempt to restore stability by dynamic internal fixation, in which the implanted hardware has the capacity to bear loads previously carried by disks, facets, and ligaments. Cervical disk arthroplasty, performed for single-level cervical disk degenerative disease, involves diskectomy and decompression of the epidural space, followed by insertion of an artificial disk into the disk space.

Spine surgeries can be performed through anterior, posterior, lateral, or combined anterior–posterior approaches. Anterior approach is used for exposure of ventral spine and spinal cord; anterior approaches for thoracic and lumbar spine may require invasion of thoracic and abdominal cavities, respectively. Posterior approach is used for dorsal spinal column surgeries; lateral approach is commonly used in thoracic spine surgeries. Combined anterior–posterior approaches are rarely used and are typically indicated for correction of multilevel collapse, unstable column injury, severe scoliosis, and infective or neoplastic conditions [4].

In recent years, MIS have emerged as an alternative to open spine procedures.

1.1.1 Minimally invasive spine surgery

Traditional open approaches to the spine, although familiar to surgeons, are associated with morbidity, increased blood loss, increased postoperative pain, longer recovery time, and impaired spinal function. Thus, less invasive techniques that can provide equivalent or superior outcomes compared with conventional open spine surgery, while limiting approach-related surgical morbidity, are desirable [8].

MIS procedures are characterized by small surgical incisions, minimal disruption of musculature compared with standard open approaches, intraoperative neurophysiologic monitoring and intraoperative imaging modalities including fluoroscopy and computerized navigation technologies. The use of small surgical incisions to approach pathology has been associated with less surgical-related morbidity, better long-term postoperative outcomes, and decreased costs largely due to shorter postoperative hospital stays [9]. Another key concept of MIS is to limit the amount of tissue resection to minimize postoperative spinal instability.

Spine surgeons are familiar with the patient anatomy when it can be directly visualized. However, minimally invasive exposures are generally limited to the area of surgical interest and certain key anatomic landmarks can be lost within this limited field of view [10]. Familiarity with the anatomy allows the surgeon to safely perform the surgery, for this reason MIS is more technically demanding, as surgeons must work through small channels and longer distances. Operative times and complications are reduced in MIS as the surgeon becomes more experienced with the technique [11]. MIS often requires the use of intraoperative fluoroscopy or image guidance. The surgeon needs to master the use of these systems to complete the surgery in a safe, effective manner.

Three surgical objectives have driven the evolution of MIS: limit tissue disruption and destabilization of the spinal column to leave the smallest operative footprint possible; achieve bilateral decompression via unilateral approach and achieve indirect neural decompression [12]. These techniques use smaller incisions and respect the anatomic planes, so they cause much less collateral damage as compared with open procedures. They are associated with a lower stress response, less postoperative pain with reduced anesthetic requirements, faster recovery, and a shorter hospital stay than open procedures; however, there are no differences in long-term outcome between the two techniques [4].

1.1.2 Robotic assisted spine surgery

Surgical robotics emerged during ‘90s even if robots were used in surgery since ’80s [13] and since then, progresses have been made to optimize the use of robotic technology. Surgical robots are designed to enhance and complement the surgeon’s abilities during surgery. Potential advantages of robotic-assisted surgery include: increased accuracy of implant placement and surgical procedures, improved clinical outcome, reduced operation time, reduced invasiveness of the procedure, and reduced radiation exposure to the patients, surgeons, and operating staff [12].

In general, robotic assisted spine surgery includes the following steps [12]:

1. Preoperative planning using a Computed Tomography (CT) image with 1 mm slices. On this scan, the surgeon plans the placement of the implants in a virtual 3-D model of the spine. The surgeon then transfers this preoperative plan to the workstation.

2. Mounting the stabilization platform to the spine, to which the robotic device is attached.

3. Automatic image registration using two intraoperative X-rays scans obtained with a fluoroscope and a reference frame. This step defines the position of each vertebra in a 3-dimensional space with respect to the mounting platform.

4. Implant placement by forwarding the robot to the various trajectory positions according to the preoperative plan, followed by drilling and positioning the appropriate device.

This technology offers the benefits of precise preoperative planning for the most suitable entry points and the most appropriate trajectories and intraoperative execution of the plan. All these parameters can be computed even in the presence of severe deformities and loss of anatomical landmarks [12].

The surgical robots can be divided into three categories characterized by different levels of assistance:

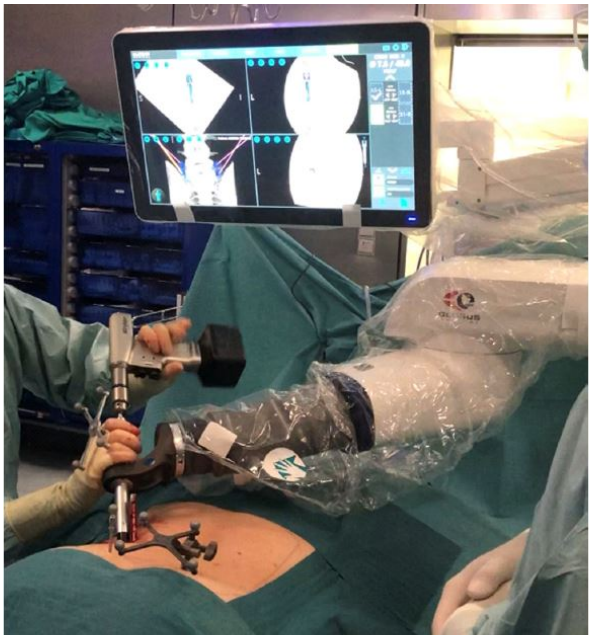


Figure 1.2: *The Excelsius GPS (Globus Medical, Inc., Audubon, PA, USA) surgical system allows to guide pedicle screw insertion using a patient-mounted reference array. Figure adapted from [21].*

- Tele-surgical systems which present remote command station from where the surgeon controls every motion of the machine, e.g. da Vinci Surgical System (Intuitive Surgical Inc., Sunnyvale, CA, USA);
- Supervisory controlled systems in which the machine is preprogrammed with actions which are autonomously performed by the robot itself under close supervision of the surgeon;
- Co-autonomy type shared-control models in which both the surgeon and the robot concurrently control motions [14], [15].

The improvement in quality of care provided by this technology can be measured by the precision of screw placement and the decreased number of surgical and long-term complications [5]. The precision of robotic guidance systems in accurately placing screws (Figure 1.2) has been demonstrated by several studies [16], [17], [18], [19], which have shown up to 99% of accuracy [20].

Clinical complications of traditional spine surgery include infections, neurological deficit, and patient complaints requiring revision or readmission [22]. Complication rates in robotic-guided MIS are significantly lower than traditional surgeries, moreover this technology allows to produce minimal muscle dissection, retraction and bleeding which leads to less intra-procedural and post-procedural complications with significantly less pain, strain, and stiffness [5] . Also, revision surgeries are reduced with robotic MIS as compared to freehand-based MIS [23].

Intraoperative fluoroscopy has always been a drawback, as it exposes both the surgeons and the operating staff to radiations, especially in MIS procedures. However, robotic assisted spine surgery minimizes reliance on intraoperative fluoroscopy, as preoperative CT scans are required for planning the robotic procedure, decreasing the use of fluoroscopy of 75% as reported by previous literature research [23].

1.2 Augmented reality

AR is a new promising technology consisting in dedicated software and hardware capable of showing images directly onto special lenses or monitors allowing the superimposition and combination of real-world scene into a person's field of view (the intraoperative scene), and some virtual content (the patient-specific anatomy) not visible in the real surroundings [24].

Virtual AR systems are able to show images directly on special visors and screens allowing the surgeon to visualize information about the patient and the procedure (i.e., anatomical landmarks, screw direction and inclination (Figure 1.3), distance from neurological and vascular structures etc) [21].

AR can avoid some drawbacks of MIS and provide opportunities for new medical treatments, in fact AR allows to reduce the amount of unnecessary damage to the patient, by enabling the physician to visualize aspects of the patient's anatomy and physiology without disrupting tissues. In addition, imaging methods such as CT, Magnetic Resonance Imaging (MRI), and Ultrasound (US) scan make possible the guidance of instruments through the body without direct sight by the physician [26].

Thanks to AR, sensitive structures placed in the surgical field can be identified in the pre-operative plan, and their intra-operative position can be retrieved in order for the robot to avoid the interaction with such structures. The patient-specific anatomy is obtained with high resolution anatomical imaging, and it can be acquired both in

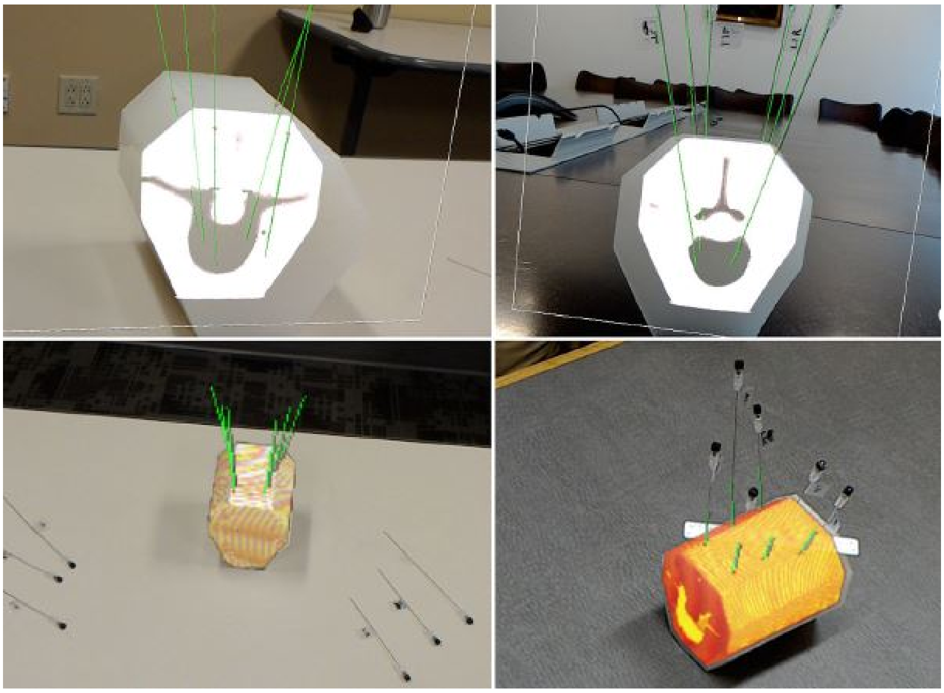


Figure 1.3: *Image showing a 3D model of the spine and interactive views using an AR HMD to visualize inner structures of the model. Through the HMD the virtual paths to insert the screws are shown (green lines) helping the positioning. Figure adapted from [25].*

the pre-operative (more common) or in the intra-operative phase (intra-operative view is often endoscopic).

In the past two decades, with the rapid progress of Graphics Processing Units (GPUs), the combined possibilities of AR and stronger GPUs have enabled the use of these technologies also in the medical area, where they can be used not only during analysis and planning, but also during surgical interventions. Thanks to the GPUs computational power, heavy processing can be offloaded to a dedicated backend in order to be light on the low-resource headsets, e.g the HoloLens headset [27], [28].

Advantages of using AR in surgical procedure include pre-operative and intra-operative guidance and decision-making, support to medical staff, match of information from different sources made by the system so the surgeon's cognitive load is reduced, increased procedure efficiency and reduced radiation exposure.

1.2.1 Pioneering AR Systems in MIS

AR systems were firstly introduced for trials in the neurosurgery field thanks to the presence of the skull, a rigid structure presents both in pre-operative and intra-operative phases, which allows the superimposition of frames acquired in the two phases according to the stereotactive approach. Stereotaxis is a branch of surgery which involves the 3D localization of the target, expressed with respect to rigid frames solid with the patient [29].

The brain shift, i.e. brain deformation occurring when the skull is open, is very limited (but still present) and usually is assumed null. Thus, it is relatively easy to superimpose pre-operative and intra-operative data under the assumption of rigidity.

AR appears in the medical field in the '80s with the superimposition of tumor boundaries obtained from CT to the microscopic view in the operating room (OR). Superimposition were possible thanks to the intra-operative localization of the microscope using US to determine position and orientation of the probe [30]. In the same years, superimposition of the anatomy extracted from pre-operative CT to the stereotaxic space were used to guide the laser resection of the tumor, guaranteeing control of the radiation, and spearing healthy tissues [31]. Another investigated AR application is in bypass surgery [32]. A big amount of studies and works are present on the topic of AR in neurosurgery, [33], [34], [35] so that nowadays it is considered a standard [36].

AR is used also in other fields as otolaryngology, maxillofacial surgery, ophthalmology, orthopedics, and dental surgery, in which a rigid supporting structure is present. On the contrary, in procedures in which such supports are not present, AR is not a standard procedure and presents big challenges because of soft tissue deformation, which is still an open issue.

One of the first study in which AR was applied on deformable tissues included the projection, through a display, of the anatomy extracted from US imaging on the abdomen of a pregnant woman [37]. First AR systems were used in laparoscopy for pancreaticoduodenectomy [38], liver segmentectomy [39] and urology [40].

1.2.2 AR in spine surgery

In the last decade, numerous AR systems have been described in literature regarding the treatment of degenerative cervical, thoracic and lumbar spine diseases, vertebroplasty, kyphoplasty, spine deformities, and biopsies [41]. An example of superimposition of the anatomy to the intra-operative scene is reported in Figure 1.4.

One promising field in which AR can provide intraoperative assistance to the spine surgeon is during pedicle screw fixation. AR, in fact, allows surgeons to not move their field of vision from the patient during the procedure and maintain their gaze while assessing the relevant trajectories and anatomy. Several studies have been conducted, both in vivo and on cadaveric models, to prove the advantage given by AR. Results show an average error of the needle insertion angle to be 2.09 degrees in the axial plane and 1.98 degrees in the sagittal plane with no pedicle breaches noted [43], [44]; increased accuracy and efficiency with thoracic pedicle screws. The additional benefits of using AR is that fluoroscopy usually is not used, sparing radiation exposure [45], [41].

AR was further used to automatically track instrument position to provide the surgeon with a real-time feedback of the instrument location. Such implementation led to an improved identification of the bone screw entry point and angulation, with a 97.4%–100% accuracy of the virtual screw placement as extracted from positional data [46].

In addition to the growing evidence related to the use of AR for pedicle screw placement, the use of AR has recently been investigated in the field of cervical spine. The first cervical spine AR application was shown in 2018 where it was combined with navigation to allow for necessary anatomical landmarks to be projected into the surgeon's visualized microscopic view [47]. Results from this study suggest that a wider implementation of AR for cervical spine application will be possible.

Spinal deformity is another subspecialty of spine surgery that can benefit from AR applications. In fact, spinal deformity surgeries are difficult to perform due to their deviation from standard anatomy, their 3-dimensional nature, and the high complication rates. In this field AR can be used to visualize resection planes of an intraoperative osteotomy, to help increase accuracy and patient safety [48].

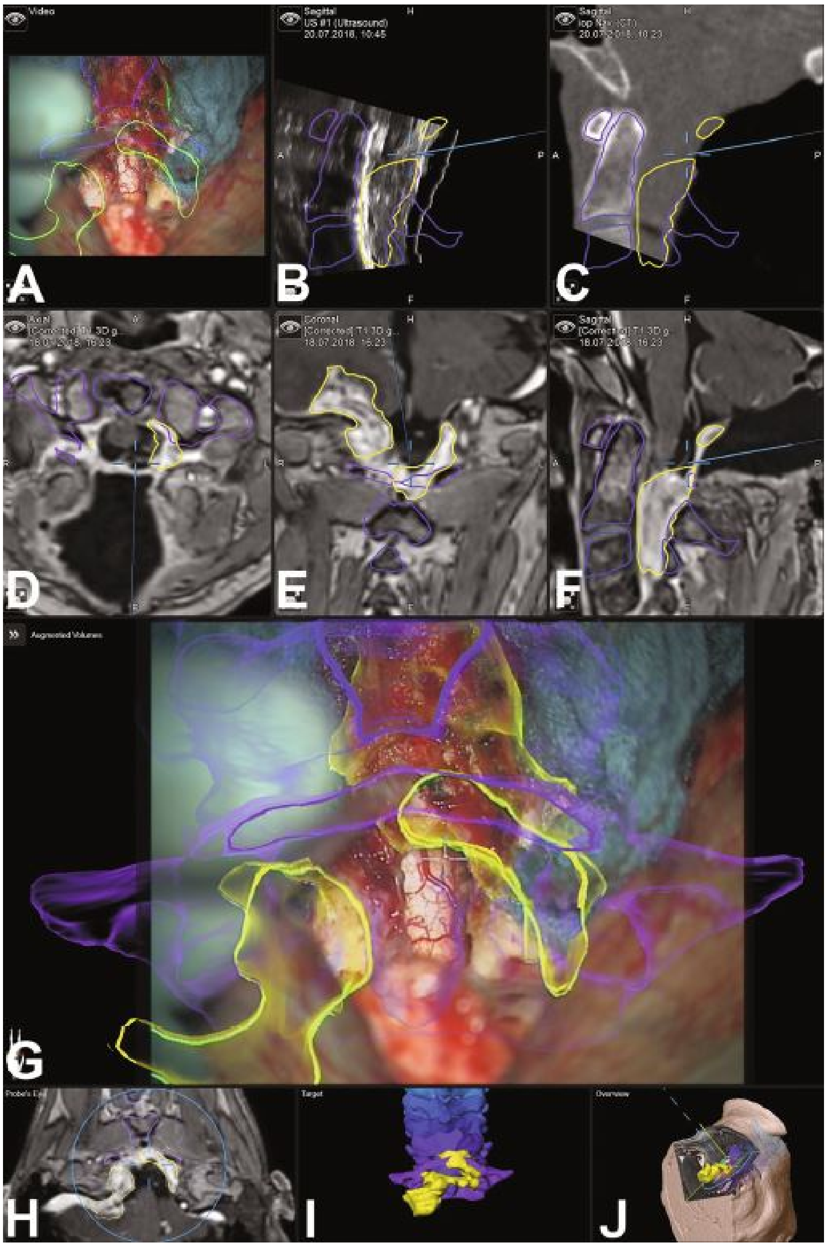


Figure 1.4: *Image showing intra and extradural squamous cell lung carcinoma during navigation with microscope video. From the figure it is possible to appreciate AR display with visualization of the AR volumes as semitransparent objects(G), the visualized 3-D objects (I), 3-D representation of how the video frame is positioned in relation to the imaging data (J). Figure adapted from [42].*

1.2.3 Limits of AR systems in spine surgery

Although AR systems for spine surgery represent a promising technology with evident benefits for both patient care and surgical performance, such devices still raise several questions regarding handling, feasibility, surgeon's learning curve and cost/benefit ratio [21].

In AR surgical navigation registration between the model image and the patient's spine may be an issue. This is due to slight changes in the patient's breathing or posture during the actual process. Moreover, due to the limited perspective of human eyes, the virtual model and the real model may appear completely matched during the experiment. This deficiency increases the error of virtual and real registration, which directly affects the precision of surgical navigation [49].

Technological limits of AR adoption in MIS include insufficient tracking precision, complex workflows, low framerates, and lack of visualization of intra-operative variables. As for the need of real-time patient motion tracking, set of markers attached to the patient body have been used. Multiple tracking technologies have been proposed (optical and electromagnetic tracking are the most widespread) but suffer from considerable drawbacks: optical tracking requires direct line of sight between a set of stereo cameras and the markers (difficult to achieve in a crowded OR) and electromagnetic tracking precision is greatly reduced by the proximity of metallic objects [50].

Other limits include hardware or software failure, cannula misplacement or skidding of the drilling tip on the pedicle surface due to peculiar bone anatomical configurations [51]. Moreover, this technology shows a demanding learning curve as it needs a moderate quantity of time to be completely experienced and understood (an average of 25 cases per surgeon are estimated to be necessary to acquire a high degree of accuracy and avoid mismatch during the screw placement) [52].

Another big limitation is the high cost of the instruments, which importantly reduces the possibility of a wider development of this surgery and consequently the average knowledge level for the single surgeon [21].

1.3 Disclosure

This thesis is part of the SUGAR - SUrgical Guidance using Augmented Reality - project, developed within the framework of Horizon 2020 research program and the Personalized Medicine megatrend identified by the ATTRACT consortium. SUGAR is an European project led in Donostia-San Sebastian, Spain, by Cyber Surgery, a start-up company focused in the development of a surgical robot for spine surgery, and Vicomtech, an applied research center specialized in advanced interaction technologies, computer vision and data analytics.

The aim of this thesis is the development of an AR system for robotic-assisted MIS. The system will provide real-time information to the surgeon during the intervention, without losing focus on the surgical field, to make the intervention safer and to reduce surgical time.

Successful adoption of the proposed technologies would change the way in which MIS are performed allowing safer, more efficient and more personalized procedures, in which surgical planning could be defined taking into account individual patients' anatomy and pathology.

Chapter 2

STATE OF THE ART

Medical images take a significant part in patient diagnostics at different level, including regular screening, diagnosis verification, preoperative planning, and follow-up checkup. Accordingly, great efforts are invested into this field to improve the quality of images and facilitate accurate scan interpretation and avoid medical errors.

In this Chapter a literature review about the use of AR in medical imaging will be presented. Starting from medical imaging equipment and conventional slice-by-slice techniques, the discussion will then pass to advanced 3D rendering methods including surface rendering and volume rendering and finally to the application and use of AR in the field of medical imaging. Then the limitations in the state of art and the thesis objective will be discussed.

2.1 Medical imaging

Medical imaging refers to the techniques and processes used to create images of the human body for clinical purposes, diagnosis or medical science including the study of normal anatomy and function [53].

Medical imaging is considered as a part of biological imaging, which has been developed from 19th century onwards. Advances in computer science, image technology, visualization technology and graphics workstation gave rise to many different processes and ways of medical imaging [54]. For clinical purposes, medical images of specific tissues or organs are obtained to assist in diagnosing a disease or specific pathology, as they provide precise anatomic and physiologic information to physicians.

A great variety of imaging equipment are available nowadays, including systems that generate 2D images (radiographs) as well as systems that generate volumetric images (CT, MRI, Positron Emission Tomography (PET), and Single Photon Emission Tomography (SPECT)). This latter technology is the most used nowadays as it provides a better visualization of physical structures with respect to a traditional 2D image [55]. Moreover, the selection of an appropriate medical imaging modality is important to obtain the target information for a successful investigation. Anatomical structures can be effectively imaged with CT, MRI, US and optical imaging methods; while information about physiological structures with respect to metabolic functions can be obtained through nuclear medicine SPECT and PET, US, optical fluorescence and several derivative protocols of MRI such as functional MRI.

To obtain a CT scan, multiple projection images are acquired as the X-ray tube and detectors assembly rotate around the patient. Image reconstruction algorithms, such as filtered back projection, are performed to generate cross-sectional images. Data are stored in Digital Imaging and Communication in Medicine (DICOM) format with a typical matrix of 512×512 pixels. A pixel is a 2D object with a discrete length in the x and y directions. A voxel is a 3D object created starting from a pixel by adding a third dimension to obtain volume. Each pixel has an associated gray-scale value called Hounsfield Unit (HU), which is a descriptor of the density and composition of the tissue.

MRI, similarly to CT, provides contiguous planar images of axial, sagittal and coronal projections. The imaging data for MRI scan are similar to that of a CT scan in matrix size and gray scale, but unlike CT, MRI do not employ ionizing radiation. In fact, a large magnetic field is directed through the patient transmitting a radiofrequency pulse into the body, then receiving coils process the returning electromagnetic signal from the body to create an image. Furthermore, MRI scans can perform exceptional contrast resolution between tissues of similar density [56].

2.1.1 Conventional medical data visualization methods

Conventional visualization method for volumetric datasets is a slice-by-slice viewing in the axial, sagittal and coronal imaging planes. Occasionally, a view in another plane other than the conventional ones is needed, in these cases oblique plane reformats can be used with the images still viewed in a conventional slice-by-slice approach. A standard viewing method includes a flat screen, high-resolution diagnostic imaging monitor with keyboard and mouse [55].

Challenges in visualizing conventional volumetric data include firstly information overload. In fact, the great improvements in spatial resolution (commonly smaller than 1 mm) in both CT and MRI have challenged radiologists with a great volume of the generated datasets. Radiologists have to review axial, coronal, and sagittal plane images, which for example can total over 1000 images for a single CT examination. They must go through each 2D slice and mentally create a 3D volume, which can be particularly challenging depending on the complexity of the anatomy [57].

A second challenge for radiologists to overcome is ensuring a very exhaustive look at each of these pixels to guarantee detection of small lesions. An example of this is the difficulty in the identification of tumors in an early stage, which is very important to improve patient survival and reduce cost of treatment. However, a very methodical slice-by-slice examination takes considerable time [58].

2.1.2 Advanced medical data visualization methods

As a result of the large, complex datasets generated by volumetric data, innovative viewing methods have emerged. In this section, imaging techniques that can improve the visualization of the human body's complex 3D anatomy are described.

The first 3D rendering technique introduced to display the human body's anatomy is *surface rendering* (also known as shaded surface display) [55].

Surfaces are displayed using segmentation techniques such as thresholding which allows to select only the desired set of pixels and display surfaces within the body. A virtual light source is used to provide surface shading.

In surface rendering, only a single surface is used. This technique has both advantages and disadvantages. An advantage of displaying a single surface is the possibility to avoid overlapping of tissues within the human body. On the other side, this can become a limitation when trying to understand the relationships between multiple organ systems. In fact, thresholding is used for one tissue type at a time and it can be difficult to understand the anatomic relationship of different structures. Also, organs may have similar density to their surroundings, and it can be difficult to segment these structures out. Lastly, true depth perception cannot be achieved with surface rendering as images are displayed on flat screens.

Volume rendering is a technique that has been researched for many years in computer graphics and visualization community and has recently been applied to diagnostic medical imaging [59], [60].

In contrast to surface rendering which requires segmentation, volume rendering does not typically require segmentation; in case it is performed during volume rendering, the entire volume or subset of the volume can be preserved [59]. Values of interest are defined by using a transfer function to assign color and opacity to each intensity value. While this technology is more computationally demanding than surface rendering, it has several advantages.

The primary advantage is that it enables the radiologist to view the volume contiguously, in fact when the volume for 3D rendering is created, the slices are stacked up in the proper sequence and a non-overlapping volume of voxels is obtained. This can significantly help radiologists to visualize complex 3D structures, including vasculature [60], [61].

One of the key limitations of volume rendering is the overlapping of structure, as shown in Figure 2.1 and Figure 2.2. There have been considerable efforts to overcome this limitation including importance-driven volume rendering [62], smart visibility [63] and curved planar reformation [64]. A visual comparison between the volume rendering and surface rendering techniques is presented in Figure 2.3.

The above stated limitation of volume rendering can be minimized using Depth 3-Dimensional (D3D) Imaging. D3D is a system that can be used with either Virtual Reality (VR), AR or Mixed Reality (MR), depending on the selected head mounted display (HMD) [66].

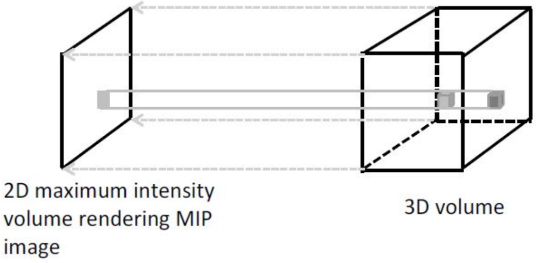


Figure 2.1: *Illustration of volume rendering with maximum intensity projection. During this process, a series of parallel rays are traced from the 3D volume to the 2D image. The projection image is created by displaying the voxel in a particular ray that has the highest brightness level (or HU in CT). The 3D volume contains a dark gray voxel and a light gray voxel, while on the 2D image only a light gray pixel is shown. Figure adapted from [58].*

.

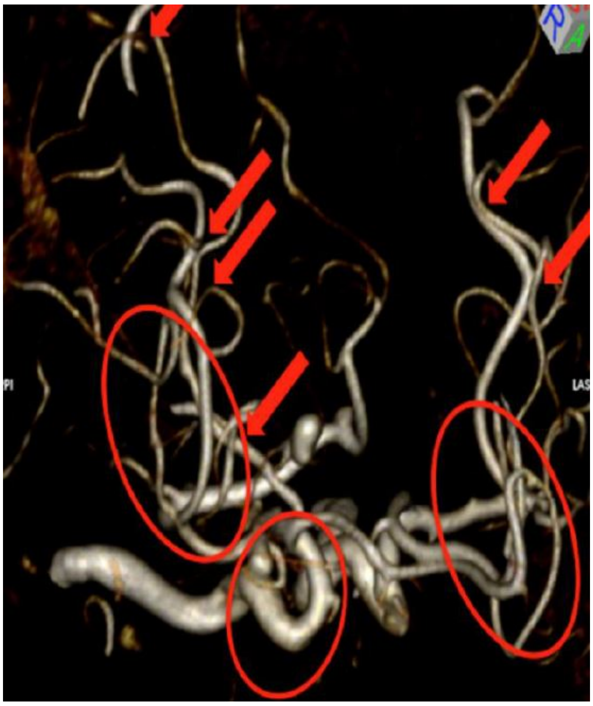


Figure 2.2: *Cerebral magnetic resonance angiography viewed with a volume rendering technique showing only the cerebral vasculature. The red arrows and ovals indicate areas of overlapping blood vessels which are seen from every angulation, limiting the evaluation. As images computer processing is performed including apparent light source and shadowing, it can further increase interpretation errors. Figure adapted from [58].*

.

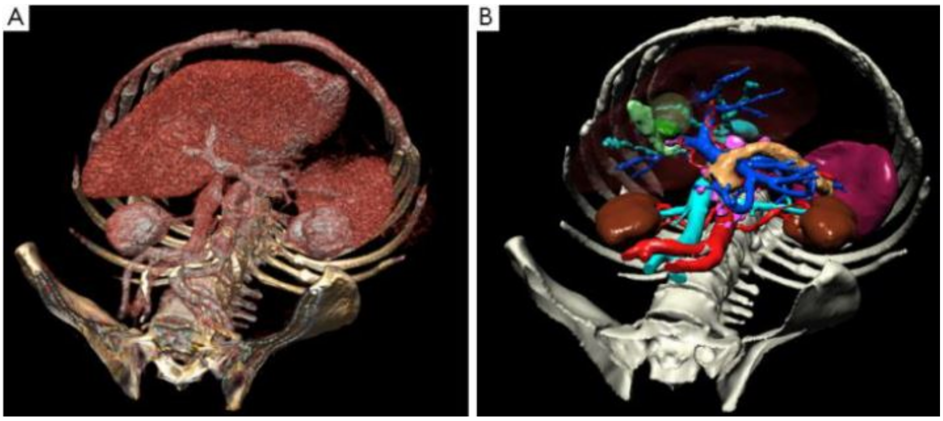


Figure 2.3: *Comparison between volume (A) and surface (B) renderings: volume rendering produce more realistic images and make better use of all available imaging data. Figure adapted from [65].*

In this system, the user wears an HMD on which a separate image is displayed to each eye to provide binocular disparity and depth perception. The left and right viewing perspectives are created through a rendering engine heavily reliant on the GPU. The rendering engine also provides some maneuverability possibilities to the user such as moving the viewing position, rotate or scale [67].

2.2 AR, VR and MR in medical imaging

While it is not possible to completely get rid of misinterpretation cases, a way to reduce their occurrence is found in the use of technologies, namely AR and VR. These can be used to educate and train radiologists, improving their expertise and skills by allowing them to review a bigger variety of medical cases as well as evaluate human anatomy variations and perform more accurate diagnosis.

AR and VR provide enhanced viewing including depth perception and improved human machine interface. AR, MR and VR HMDs present a unique image for each eye, thus achieving stereoscopy and depth perception.

Virtual reality technologies can be characterized as either non-immersive such as desktop computers, semi immersive or fully immersive VR [68], [69]. In fully immersive VR (as Oculus Rift and HTC Vive, illustrated in Figure 2.4), the HMD presents a virtual image and completely occludes the real-world from the user's field of view [70].

In semi-immersive VR, the HMD presents a virtual image and partially occludes

Figure 2.4: *Example of full immersive VR headsets, HTC Vive (left) and Oculus Rift (right). Figure adapted from [71], [72] .*

the real-world from the user's field of view. In VR, the user can navigate through the virtual world by head movements (via HMD tracking) or by walking (via external camera tracking). The user can interact with the virtual environment through handheld devices with haptic feedback or voice gestures. One of the challenges of VR is the lack of accurate head-tracking and motion sickness [73].

Augmented reality technologies mix a real-world environment with computer-generated information in real-time or non-real time for different purposes. The generated information that is used in AR can be video, audio, text, even smell and touch sensations, or a combination of them that can enhance real world experiences around the users [74]. Nowadays, the main focus of AR research is on visual AR, which is the most common type.

AR technologies can be subdivided into AR and MR. Both of them provide simultaneous display of a virtual image and a real-world image allowing the user to simultaneously interact with the real-world and the virtual image [75]. In both technologies, the user wears an HMD to display the virtual image together with the real-world image. In AR, the virtual image is transparent like a hologram as is seen in Meta and DAQRI systems, while in MR, the virtual image appears solid as is seen in HoloLens (Figure 2.5). The user receives simultaneous display of a virtual image from the patient's imaging examination and the real-world image of the surroundings, which would vary based on the task being accomplished. The real-world image would be the patient's anatomy in case of an integrated physical exam and medical imaging assessment, pre-operative planning assessment or intra-operative procedure.

Figure 2.5: *Example of full immersive VR headsets. From left: HoloLens, Meta, Daqri Helmet and Daqri Glasses. Figure adapted from [76], [77], [78].*

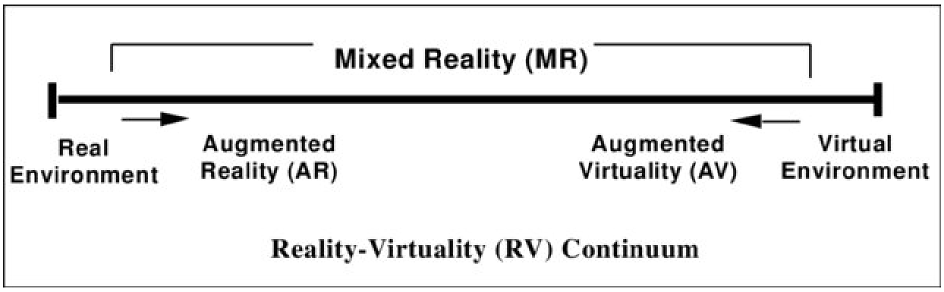


Figure 2.6: *In the reality-virtuality continuum, MR refers to the area between real environment and virtual environment; real environment refers to real-world objects which can be observed directly or by conventional display; virtual environment refers to an environment which is completely computer-generated. Any combinations of the real and virtual world can be placed between these two. Augmented Virtuality is a virtual environment in which physical objects are augmented by computer-generated information. Figure adapted from [79].*

In 1994 AR has been described [79] as part of MR, a single environment in which real-world and virtual objects are displayed together on a screen. All technologies that mix real world and digital information in any form are part of MR. To help the understanding and discernment about AR, MR and VR, the Reality-Virtuality Continuum concept (also known as Mixed Reality spectrum) which connect real-world to the virtual world has been introduced (Figure 2.6).

As the AR and MR domains are relatively new and the features specter is relatively continuous, it is not possible to find a formal definition of where AR stops, and MR starts. Generally, AR systems "augment" the reality by overlaying an image. The overlaid image is not spatially anchored (changes position as we maneuver in the real world) and it can be two dimensional. In MR, the real and virtual worlds are blended and mixed together in order to blur as much as possible the boundaries between the real-word and the virtual world: the virtual image is generally three dimensional and carries properties of real objects (position, rotation, speed, interaction with real environment).

2.2.1 AR systems

In general, AR systems can be divided in two groups, portable and stationary [80].

Stationary AR systems (personal computers, video game consoles, projectors) usually are equipped with powerful hardware that allow the use of more computationally expensive computer vision algorithms to get better understanding from the real-world environment and provide high quality augmented content.

On the other hand, portable devices (mobile phones, tablets, AR helmets, smart glasses) do not limit their users to specific location enabling the use of AR for a much wider range of purposes. Usually they are integrated with different sensors such as Global Positioning System (GPS), Inertial Measurement Unit (IMU) and digital compass which are used for more accurate and robust tracking. Mobile augmented reality systems can be classified as wearable, like smart glasses and AR helmets, and non-wearable, like smart phones and tablets. AR wearable devices give a better perception of the surrounding environment and being hand-free enable their users doing other tasks while receiving required information [81]. Usually, the users interact with the system using voice commands, gesture or gaze. However, the portability of mobile devices comes at the expense of hardware limitations as limited processing power and memory.

To overcome this problem distributed AR systems can be used, where mobile devices capture required data and send them to a more powerful server, where data are analyzed, and the required augmented information is generated for processing. The generated content is then sent back to the mobile devices for visualization [82]. Figure 2.7 displays the architecture of distributed AR systems. In this system, mobile devices send data through internet or wireless network, the quality of the network in this case plays an important role. In fact, data transmission latency has a big impact on the performance of AR systems as high latency can result in missed synchrony of augmented content with real objects in the scene.

A basic architecture of AR systems has three core components [83], which are:

1. Input device (sensors) to determine the state of the physical world where the application will be deployed; different devices such as cameras (color and RGB-

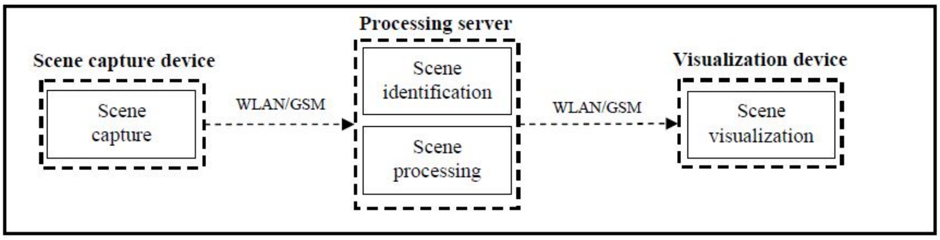


Figure 2.7: *Architecture of distributed AR systems. Figure adapted from [82].*

D), GPS, IMU and mechanical tracking device are employed, depending on the AR system's objectives.

2. A processor (AR Engine) to evaluate the sensor data and generate the signals required to drive the display. This component is formed by a tracker and a content generator unit. The tracker ensures that the augmented content are aligned (registered) properly in the scene; the content generator unit coordinates and analyzes sensor inputs, stores and retrieves data, carries out the tasks of the AR application program, and generates the appropriate signals to display. This component must have enough computational ability to perform tasks in real time and update the scene smoothly and at a rate that the user perceives as a constant stream of informations.

3. Output device (display) suitable for creating the impression that the virtual world and the real world are coexistent. AR systems use different types of display (visual, audio, haptic, stereoscopic, and stereophonic) to present the visual augmented content. In particular, visual displays can be categorized into three groups which are video-see-through, optical-see-through and video-projector. The *video-see-through displays* are used in handheld devices, closed-view HMD or monitor-based AR systems and provide an indirect view of the real-word as the image of the physical world and the generated augmented view are combined before being presented to the user. The *optical-see-through displays* are used in HMD and head-up displays. They are half-mirror displays which provide a direct view of the surrounding physical world; the augmented content is projected on the half-mirror display in order to present, at athe same time, both augmented view and real-world view to the user. *Video-projectors* are used in

some AR systems to project digital information directly on the physical objects.

2.3 Limitations in the state of art

From the literature review about the use of AR in medical imaging, some limitations have arisen. In particular, AR suffers from technical problems (insufficient tracking precision, complex workflows, low framerates and lack of visualization of intra-operative variables) for which currently no suitable solutions seem to exist.

One of the biggest limitations of AR is the low framerate achieved by HMDs. Although multiple HMDs have entered the market in the past years, their performance is still limited by the onboard hardware, which limits the rendering framerate of heavy datasets. This situation worsens when using volume rendering, that although allows more realistic visualizations, its computational requirements are much heavier and not suitable for embedded imaging hardware. To solve this problem, it would be possible to offload rendering operations to a remote machine, which transmits results to the AR HMD. This solution requires the use of an appropriate streaming technology to transmit rendered images to the HMD. But in this case another issue arises as common streaming technologies can have delays in the order of seconds, which are insufficient for a real-time AR application [84].

Another problem for surgical AR is the need of real-time patient motion tracking. Commonly, this has been solved by positioning markers on the patient body, whose position in space is measured by special tracking devices. Multiple tracking technologies have been proposed but suffer from considerable drawbacks: optical tracking requires direct line of sight between a set of stereo cameras and the markers -which is difficult to achieve in a crowded OR- while electromagnetic tracking precision is greatly reduced by the proximity of metallic objects. Optical tracking normally achieves tracking errors of several mm [50], which is not sufficient for high-precision interventions such as spinal surgery. However, previous studies with electromechanical tracking show the potential to achieve tracking errors < 1 mm, which is a considerable improvement over existing technology [85].

Further challenges faced by AR and VR in the medical imaging area include the still present perception of overlapping structures in the image, difficulty in carrying HMD during a surgical procedure as they can be considered bulky and heavy by the surgeon and also they can provide motion sickness which can hamper the medical staff's capacity to best performing the medical procedure [58].

2.4 Thesis objective

In this work a client-server architecture (Figure 2.8) is proposed, on which computationally intensive tasks are performed on remote servers, leaving the wearable devices in charge of rendering the final frames transmitted as a video stream.

In particular, the proposed system will have the following characteristics:

- A remote rendering server that offloads the heavy work from the AR glasses and renders the scene with sufficient framerate and low latencies.
- A marker tracking algorithm to track the patient motion with high precision and provide asynchronous communication with the remote rendering server.
- A client software architecture -deployed on a set of HMD- to achieve integrate visualization of intra-operative data and to prevent surgeons to look away from the surgical area to consult a remote screen.
- In-scene integration of intra-operative variables.

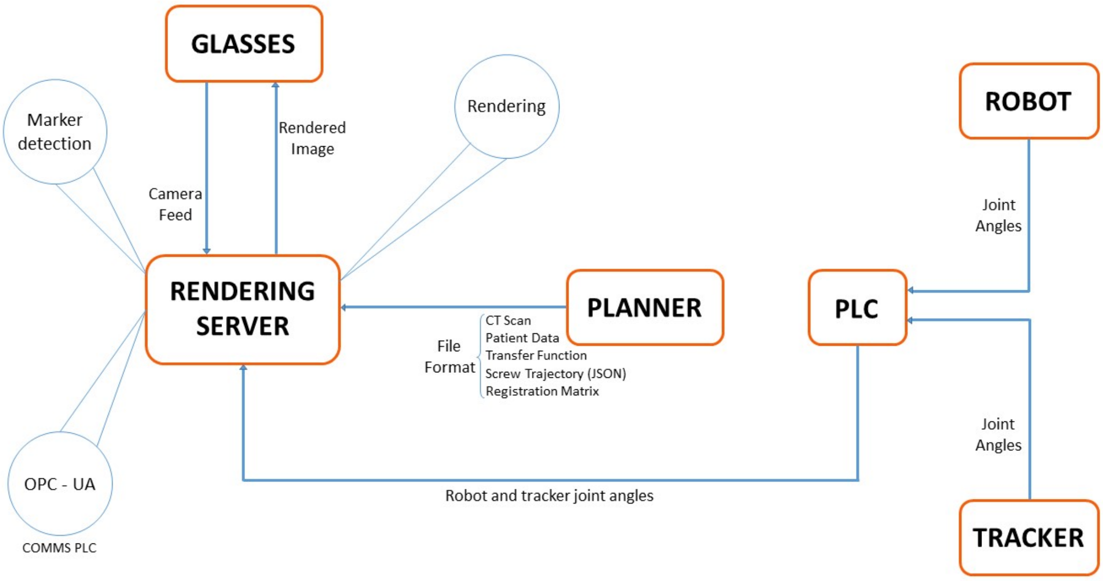


Figure 2.8: *General system overview.*

Chapter 3

MATERIALS AND METHODS

In this chapter, the architecture implemented in this work is presented. In Section 3.2.1, the developed algorithm of marker tracking is described. In 3.2.2 the external libraries used to develop the algorithm and adapt it to HoloLens are reported while in 3.2.3 the architecture developed to provide a communication between the HoloLens client and the desktop rendering server is treated. The materials used in this work and the evaluation protocol to assess the algorithm performances are presented respectively in Section 3.1 and Section 3.2.

3.1 Materials

For this project the HoloLens 1 HMD has been chosen over other available displays first of all because it is a MR headset, so it allows to "keep" the user in the real world, which is particularly important in case of medical staff. Moreover, HoloLens provides most of the images with almost zero latency allowing the solution to be used in dynamic situations. A second reason why HoloLens has been preferred over other MR devices is more practical. In fact, HoloLens has a lightweight (compared to other options), offers multiple user input possibilities, has an integrated Wi-Fi connection which is a very important characteristics as no additional cables or connected devices are needed. Technical reasons like features and support also led to the choice of this device. The GPU computing power limitation is addressed by using a dedicated Desktop Windows

PC rendering server. As regards the software implementation, in this work several open-source libraries have been used as a starting point for the development of a marker detection and tracking algorithm and for the integration of this algorithm on the HoloLens 1. The algorithm of marker tracking has been developed taking as reference OpenCV [86], an open source Computer Vision library, which offers infrastructures for real time computer vision application. In OpenCV the ArUco module, based on the ArUco library [87], [88], has been used to extend and implement the tracker application. ArUco library has been chosen among the many fiducial marker detection systems available, as it is the most popular and reliable one. In fact, ArUco is inherently robust and able to detect and correct binary code errors, it is characterized by a good performance at a wide range of marker orientations and great adaptability to non-uniform illumination conditions [88], [89]. The algorithm and the libraries mainly have been implemented in C++.

3.2 Methods

3.2.1 Tracking algorithm

The marker tracking algorithm has been created based on the use of ArUco markers. This algorithm has been firstly tested on a desktop application acquiring video of markers with a smartphone, later Adobe After Effect [90] has been used to simulate an intraoperative scenario and finally it has been adapted to be used as Universal Windows Platform (UWP) application for HoloLens in the context of server-client communication. The diagram in Figure 3.1 shows the phases of a tracking algorithm based on fiducial bitonal markers which will be better described in the following sections.

3.2.1.1 Detection of ArUco markers

This section describes the process of fiducial marker detection.

ArUco markers are synthetic square markers composed by a wide black border and an inner binary matrix which determines their identifier (id). The marker black border facilitates its fast detection in the image and the binary codification allows its

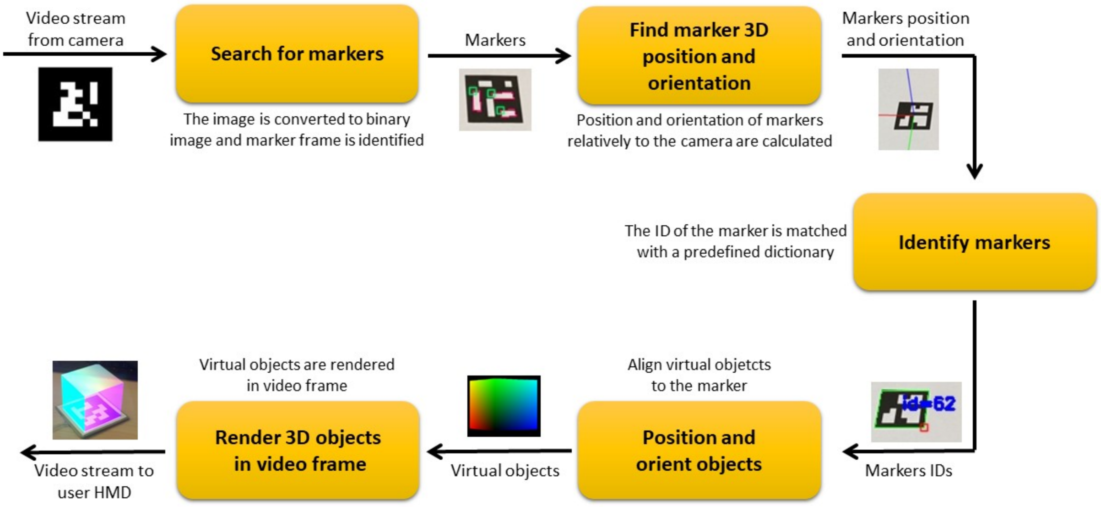

Video stream from camera
Search for markers
The image is converted to binary image and marker frame is identified
Markers
Find marker 3D position and orientation
Position and orientation of markers relatively to the camera are calculated
Markers position and orientation
Identify markers
The ID of the marker is matched with a predefined dictionary
Markers IDs
id=62
Position and orient objects
Align virtual objetcts to the marker
Virtual objects
Render 3D objects in video frame
Virtual objects are rendered in video frame
Video stream to user HMD

Figure 3.1: *Phases of ArUco marker tracking.*

identification and the application of error detection and correction techniques [87]. The marker size determines the size of the internal matrix (for instance, a marker size of 4x4 is composed by 16 bits); markers can be found rotated in the environment, but the detection process is able to determine its original rotation, so that each corner is identified unequivocally. This is also done based on the binary codification.

A dictionary of markers is the set of markers that are considered in a specific application. It is the list of binary codifications of each of its markers. The main properties of a dictionary are the dictionary size (the number of markers that compose the dictionary) and the marker size (number of bits of those markers). The marker id is the marker index within the dictionary it belongs to [91].

Given an image containing ArUco markers, the detection process must return a list of detected markers. Each detected marker includes the position of its four corners in the image (in their original order) and the id of the marker. In this work a 4x4 dictionary has been used as the number of markers needed in the scene was limited.

The detection process can be split in two main parts. The first is the candidate search, in which the image is analyzed in order to find square shapes that are candidates to be markers; the second is the identification stage, where the inner codification of the candidates is analyzed to determine whether they really are markers, and if they belong to a valid dictionary.

The algorithm of marker detection includes the following steps (values assigned to each parameter are listed in Table 3.1):

1. Apply an **Adaptive Thresholding** to the input image to obtain borders. During the Thresholding process each pixel is marked as an object pixel if its value is higher than a certain threshold, while those around with lower values are labeled as background pixels (this procedure is known as threshold above). The key point in this procedure is the choice of the threshold. In adaptive thresholding an initial threshold (T) is randomly chosen, the image is then segmented according to this threshold into object ($G1 = f(m; n) : f(m; n) > T$) and and background pixels ($G2 = f(m; n) : f(m; n) \leqslant T$), $f(m, n)$ represent the gray-level value of the pixel in m,n position. Then the average gray value of each set is computed and a new threshold value is generated as $T0 = (m1 + m2) = 2$ .The process is repeated until convergence is reached. Thresholding is adaptive when different values of T are used in different regions of the image [92].

   In the algorithm the thresholding is customized adapting the parameters that represent the interval where the thresholding window sizes (in pixels) are selected, namely "adaptiveThreshWinSizeMin", "adaptiveThreshWinSizeMax" and "adaptiveThreshWinSizeStep" which indicates the increments of the window size from min to max.

   Low values of window size can 'break' the marker border if the marker size is too large, causing it to not be detected. On the other hand, values that are too high can produce the same effect if the markers are too small, and it can also reduce the performance. Moreover, the process would tend to a global thresholding, losing the adaptive benefits.

2. **Find contours**. After the thresholding process, not only the real markers are detected but also other undesired borders. They are filtered out in different steps so that contours that are very unlikely to be markers are discarded.

Borders with a small number of points are removed through the parameters “minMarkerPerimeterRate” and “maxMarkerPerimeterRate”, which determine the minimum and maximum size of a marker and are specified relative to the maximum dimension of the input image. If the ”minMarkerPerimeterRate” is too low, it can penalize the detection performance since more contours would be considered in successive stages while this penalization usually does not occur for the ”maxMarkerPerimeterRate”, as usually more small contours than big contours are present in the image.

A second filtering step include a polygonal approximation of contours to extract and keep only the concave contours with 4 corners. This approximation is done through the parameter “polygonalApproxAccuracyRate” whose value determines the maximum error that the polygonal approximation can produce, and higher values are necessary for highly distorted images.

Corners are then sorted in an anti-clockwise direction and rectangles too close are removed. This is required because the adaptive threshold normally detects the internal and external part of the marker’s border and usually the external border is the desired one. Parameters used to do this operation are “minCornerDistanceRate” which describes the minimum distance between any pair of corners in the same marker relative to the marker perimeter; “minMarkerDistanceRate” which describes the minimum distance between any pair of corners from two different markers, it is expressed relatively to the minimum marker perimeter of the two markers and if two candidates are too close, the smaller one is ignored; “minDistanceToBorder” which describes the minimum distance of any of the marker corners to the image border (in pixels). In this case, as the position of marker corners is important to perform pose estimation, it is better to discard any markers whose corners are too close to the image border, setting a higher value for this parameter.

3. **Marker Identification**. Once marker contours have been extracted it can be identified analyzing bits of each candidate. Firs of all the projection perspective is removed to obtain a frontal view of the rectangle area using a homography

function, then the Otsu threshold algorithm [93] is used to separate black and white pixels. Otsu's algorithm is a method used to obtain binary images from images in gray levels, which assumes that the image to be processed contains two pixel classes (pixel background and foreground pixel) and calculates the best threshold value to separate the two classes. There are several parameters that can be set to customize this process: "markerBorderBits" indicates the width of the marker border relative to the size of each bit; "perspectiveRemoveIgnoredMarginPerCell" is needed because when extracting the bits of each cell, the numbers of black and white pixels are counted and in general it is better to ignore some pixels in the margins of the cells as after removing the perspective distortion, the cells' colors are not perfectly separated and white cells can invade some pixels of black cells (and vice-versa);

Once the marker has been identified, its internal code need to be extracted. To do so the marker is divided into a grid with the same number of cells as the number of bits in the marker, the internal cells of the grid contain the marker id information while the rest corresponds to the external black border. If the internal cells provide a valid id code, the marker is considered and its corners are refined using subpixel interpolation.

Finally, if camera parameters are provided, the extrinsic parameters of the markers to the camera are computed.

3.2.1.2 Calibration

Camera calibration is the process of obtaining intrinsics and extrinsics parameters of a camera which allows to determine where a 3D point in the space projects in the camera sensor (pose estimation).

Several libraries provide calibration algorithms, in this work the OpenCV routine has been used [94]. OpenCV uses the pinhole camera model in which a scene view is formed by projecting 3D points into the image plane using a perspective transformation. The camera parameters can be divided into intrinsics and extrinsics. Intrinsic parameters are focal length of the camera lens in both axes normally expressed in pix-

Table 3.1: *Values of the parameters set in the algorithm of marker detection.*

Parameter	Assigned Value
adaptiveThreshWinSizeMin	5
adaptiveThreshWinSizeMax	25
adaptiveThreshWinSizeStep	10
minMarkerPerimeterRate	0.05
maxMarkerPerimeterRate	3.5
polygonalApproxAccuracyRate	0.03
minCornerDistanceRate	0.05
minMarkerDistanceRate	0.05
minDistanceToBorder	5
markerBorderBits	1
perspectiveRemoveIgnoredMarginPerCell	0.15

els, optical center of the sensor expressed in pixels and distortion coefficients. Equation 3.1 shows the pinhole camera model.

$$s\,m' = A\,[R \mid t]\,M' \tag{3.1}$$

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

In an ideal camera, when capturing an image, a mapping of 3D points in the scene to the 2D plane on the image is performed, and a relationship between the coordinates of the 3D point and its projection is defined. However, camera lenses normally distort the scene, therefore when considering a pixel's projection also distortion components need to be evaluated. There are two type of distortions, radial and tangential, and are represented by the parameters p_1, p_2, k_1, k_2, k_3.

In the pinhole camera model (Figure 3.2) the camera is placed at the origin, the center of projection. The point P represents a point in the real world. The image plane

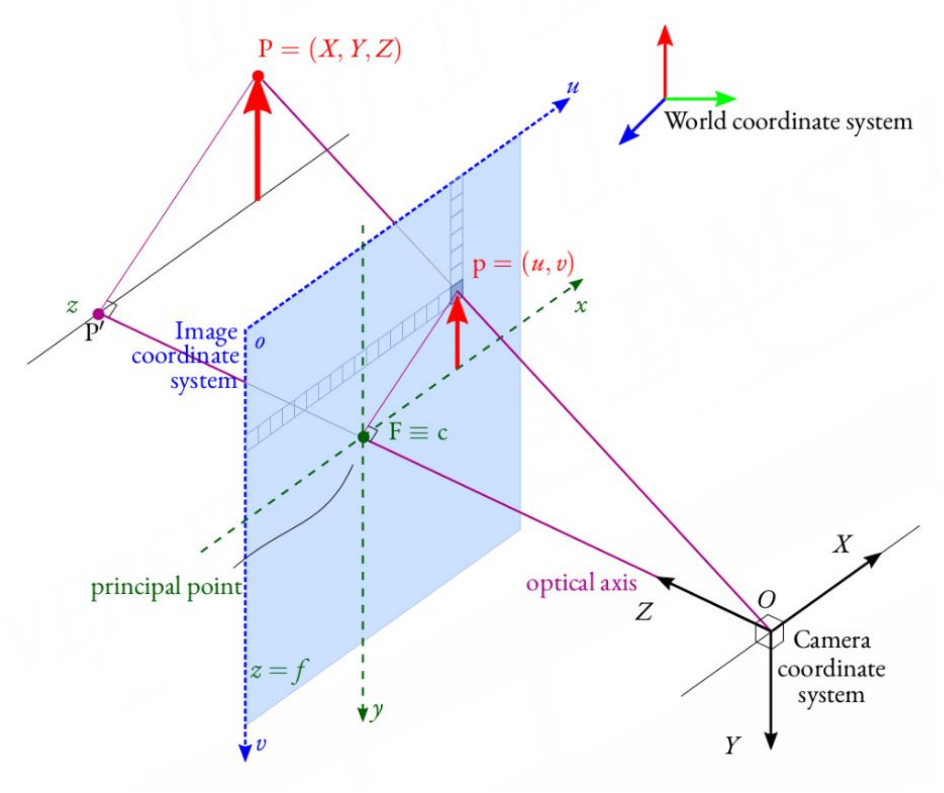


Figure 3.2: *Pinhole camera model. Figure adapted from [95].*

(or projective plane) represents the 2D plane obtained after capturing the image and contains the visible image itself. The point P gets mapped to the principal point (p), which is the point at the intersection of the image plane and the optical axis. The distance between the center of projection and the image plane is the focal length (f) of the camera. However, the principle point and the center of the image are not perfectly coincident as the center of the sensor is usually not on the optical axis. For this reason, two parameters, c_x and c_y, are used to model a possible displacement (away from the optic axis) of the center of coordinates on the projection screen. In this way it is possible to model the projection of a point P in the physical world, whose coordinates are (X, Y, Z), into the screen at some pixel location with the following equations: u = f_x(X/Z)+c_x and v = f_y(Y/Z)+c_y. Two different focal lengths are used as pixels on a typical imager are rectangular rather than square.

The relation that maps points in the physical world to the points on the projection screen is called a *projective transform* and usually makes use of homogeneous coordinates. The homogeneous coordinates associated with a point in a projective space (image plane) of dimension n are typically expressed with an (n + 1)-dimensional vector; as the image plane has two dimensions, points on that plane will be represented as three dimensional vectors. So, the parameters that define the camera (i.e. fx, fy, cx, and cy) can be arranged into a single 3-by-3 matrix, which is the **camera intrinsics matrix** and the projection of physical world points into the camera can be summarized as:

$$q \ = MQ \tag{3.2}$$

where

$$q = \begin{bmatrix} x \\ y \\ v \end{bmatrix}, M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, Q = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

For each image the camera takes of an object, it is possible to describe the pose of the object relative to the camera coordinate system in terms of a rotation and a translation. In general, a rotation can be described in terms of multiplication of a coordinate vector by a square matrix of the appropriate size. The rotation matrix R has the property that its inverse is its transpose; hence RTR = RRT = I, where I is the identity matrix.

The translation vector represents a shift from one coordinate system to another system whose origin is displaced to another location; in other words, the translation vector is the offset from the origin of the first coordinate system to the origin of the second.

Combining the rotation matrix and the translation vector it is possible to obtain the **camera extrinsic matrix** (even though it does not exactly correspond to the camera's rotation and translation), which describe the camera motion around a static scene, or vice versa, rigid motion of an object in front of a still camera, that is it translates coordinates of a point (X, Y, Z) to a coordinate system, fixed with respect to the camera.

The extrinsic matrix takes the form of a rigid transformation matrix: a 3x3 rotation matrix in the left-block, and 3x1 translation column-vector in the right:

$$[R\,|\,t] = \left[\begin{array}{ccc|c} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \end{array}\right]$$

A further row is added at the bottom of the matrix to make it square and allow a further decomposition into a rotation followed by a translation:

$$\left[\begin{array}{c|c} R & t \\ \hline 0 & 1 \end{array}\right] = \left[\begin{array}{c|c} I & t \\ \hline 0 & 1 \end{array}\right] \times \left[\begin{array}{c|c} R & 0 \\ \hline 0 & 1 \end{array}\right] \tag{3.3}$$

$$= \left[\begin{array}{ccc|c} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \times \left[\begin{array}{ccc|c} r_{1,1} & r_{1,2} & r_{1,3} & 0 \\ r_{2,1} & r_{2,2} & r_{2,3} & 0 \\ r_{3,1} & r_{3,2} & r_{3,3} & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \tag{3.4}$$

When acquiring an image, the lens of the camera tends to distort the image. There are two main lens distortions, radial and tangential. Radial distortions arise as a result of the shape of lens, whereas tangential distortions arise from the assembly process of the camera as a whole.

Radial distortion is observed as lenses of real cameras distort the location of pixels near the edges of the imager. This phenomenon is the source of the “barrel” or “fish-eye” effect. The distortion is 0 at the (optical) center of the imager and increases toward the periphery. In practice, this distortion is small and can be characterized by the first few terms of a Taylor series expansion around r = 0:

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6); \qquad y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{3.5}$$

Here, (x, y) is the original location (on the imager) of the distorted point and $(x_{corrected}, y_{corrected})$ is the new location after the correction.

Tangential distortion is due to manufacturing defects resulting from the lens not being exactly parallel to the imaging plane. Tangential distortion is minimally characterized by two additional parameters, p1 and p2, such that:

$$x_{corrected} = x + [2p_1 y + p_2(r^2 + 2x^2)]; \qquad y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2 x] \quad (3.6)$$

Thus in total there are five distortion coefficients, they are all necessary in most of the OpenCV routines, they are typically bundled into one distortion vector, a 5-by-1 matrix containing k1, k2, p1, p2, and k3 (in this order). The distortion coefficients do not depend on the scene viewed. Thus, they also belong to the intrinsic camera parameters and they remain the same regardless of the captured image resolution.

In OpenCV the calibration has been done by targeting the camera on a known structure, a pattern of alternating black and white squares (chessboard) that has many individual and identifiable points. The use of a flat chessboard pattern comes from Zhang [96], and the advantage of using a chessboard to calibrate the camera is that being made of white and black squares of known size, it is difficult to confuse a point on the board with another, also taking it from different angles.

By viewing this structure from a variety of angles, it is possible to then compute the (relative) location and orientation of the camera at the time of each image as well as the intrinsic parameters of the camera. To provide multiple views, the board has been rotated and translated while keeping the camera fixed.

Given multiple images of the chessboard (in this work 80 images of a 9x6 chessboard with square dimension of 24mm from different views have been used), the OpenCV function cvFindChessboardCorners() has been used to locate the corners of the chessboard.

The input data needed for calibration of the camera are the set of 3D real world points and the corresponding 2D coordinates of these points in the image. 2D image points are found from the image (they are represented by the locations where two black squares touch each other in chessboards), while the 3D points from real world space are obtained by providing the algorithm with the size of the physical chessboard square and calculating the coordinate of the corners. These 3D points are called object points.

Once object points and image points have been calculated, the cv.calibrateCamera() function has been used to obtain the camera matrix, distortion coefficients, rotation and translation vectors.

At the end of the calibration process, the results have been saved into an XML file (which contains intrinsics and extrinsics parameters, camera matrix), and the file has been used in the routine for estimate the pose and track ArUco markers.

3.2.1.3 Camera pose estimation and tracking

Once the camera is calibrated ArUco markers can be used to estimate its pose (i.e. its 3D position in space with respect to the marker). The detection of the four corners of a marker allows to apply planar pose estimators. They estimate the relative pose of the camera with respect to the center of the marker.

At the beginning of the routine the marker size and the XML file from where read the calibration parameters have been specified to use the library to obtain the relative pose of the markers and the camera.

In particular, the function ”estimatePoseSingleMarkers()” receives the detected markers (through the process in Section 3.2.1.1) and returns their pose estimation respect to the camera individually. So, for each marker, one rotation and one translation vector are returned. The returned transformation is the one that transforms points from each marker coordinate system to the camera coordinate system. The marker coordinate system is placed on the center of the marker, with the Z axis perpendicular to the marker plane, pointing out. Axis-color correspondences are X: red, Y: green, Z: blue. Axis and marker contours have been drawn to provide a visual check of the pose estimation and make sure markers are found correctly.

With this function is also possible to track the position of a marker in a video or camera stream.

3.2.2 Libraries and UWP adaptation

After developing an application to track ArUco markers, it has been adapted to identify and track markers in simulation videos representing a robot which performs spinal MIS.

These scenes have been created by using Adobe After Effect 2020, a software used for animation, visual effects, and motion picture compositing. This software is usually used in the post-production to manipulate images and videos, and allows to combine layers of video and images into the same scene. An image representing an ArUco marker has been overlapped to a video of a spinal MIS simulation by means of the tracking function integrated in the Mocha [97] plug in, to evaluate the ability of the tracking algorithm to detect and follow the marker in a more crowded scene.

Finally, the tracking routine has been adapted to be used with HoloLens 1 and be included in the client-server architecture described in Section 3.2.3. An application, to be deployed on HoloLens needs to use the model provided by the UWP, which defines how apps are installed, updated, versioned and removed. It regulates the application life cycle - how apps execute, sleep, and terminate - and how they can preserve state. It also covers integration and interaction with the operating system, files, and other apps. A UWP app can run on HoloLens as on any other Window device. This is very useful as it allowed to use the same app both on the HoloLens and on the emulator.

To adapt the desktop application of marker tracking described in Section 3.2.1 different libraries have been used as a base. In particular the structure of HoloLensForCV [98] and HolographicFaceTracking [99] have been investigated.

HoloLensForCV uses OpenCV to obtain camera calibration and camera images. Then, the information is processed using OpenCV functions and visualized on HoloLens. This sample performs marker tracking and places spinning cubes on the corners of the detected marker. It has been investigated to access the photo/video camera of HoloLens and calibrate it. HolographicFaceTracking has been used to acquire video frames from the photo/video camera of HoloLens. Later the webcam app capability has been used in order to stream video images.

A “MarkerTracker” class has been added and used to process images and return a list of cubes inside the image where the markers are detected. This is a very intensive process to run in real-time and could reduce rendering performance.

For this reason, the application has been split in two parts using the architecture in Section 3.2.3. On the client side the camera frame has been accessed to perform marker tracking, and the matrix obtained has been passed to the server. The rendering part of the application has been offloaded and added in the server side.

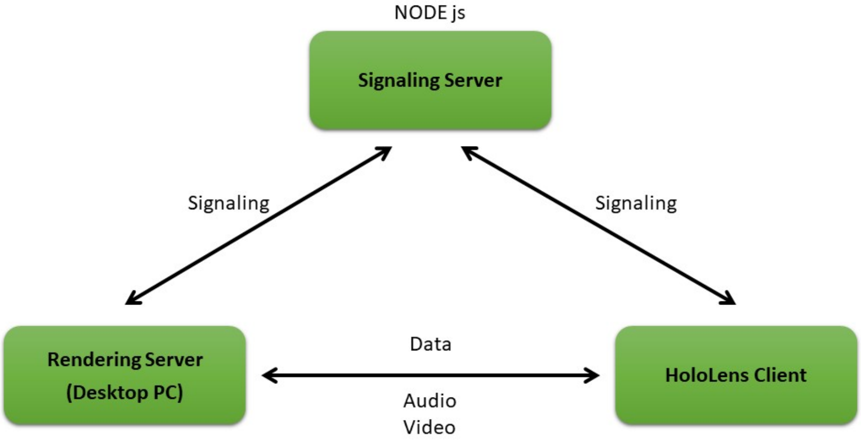


Figure 3.3: *System logical architecture.*

3.2.3 Server-Client communication

In this section will be presented the architecture used for the visualization of the 3D medical images using the Microsoft HoloLens 1 HMD. DirectX [100] and WebRTC [101] have been used to deliver desktop rendering power to HoloLens, so that the entire computation is done on the server side and HoloLens becomes a viewer. The proposed visualization architecture includes three interconnected applications, running at the same time: the HoloLens Client, the Windows Desktop Server, and a Signaling Server which manages the communication and connection between the first two, as shown in Figure 3.3.

This architecture has been built on the 3D Streaming Toolkit [102] which uses the WebRTC (Web Real-Time Communications) protocols [101], as well as the NVEncode hardware encoding library from NVIDIA. The 3D Streaming Toolkit system architecture is depicted in Figure 3.4.

The 3D Streaming Toolkit provides a server-side C++ libraries for remotely rendering and stream the 3D frames to the HoloLens, a client-side samples for receiving streamed 3D scenes, low-latency audio and video streams using WebRTC, as well as high-performance video encoding and decoding using NVEncode [102]. Moreover, the

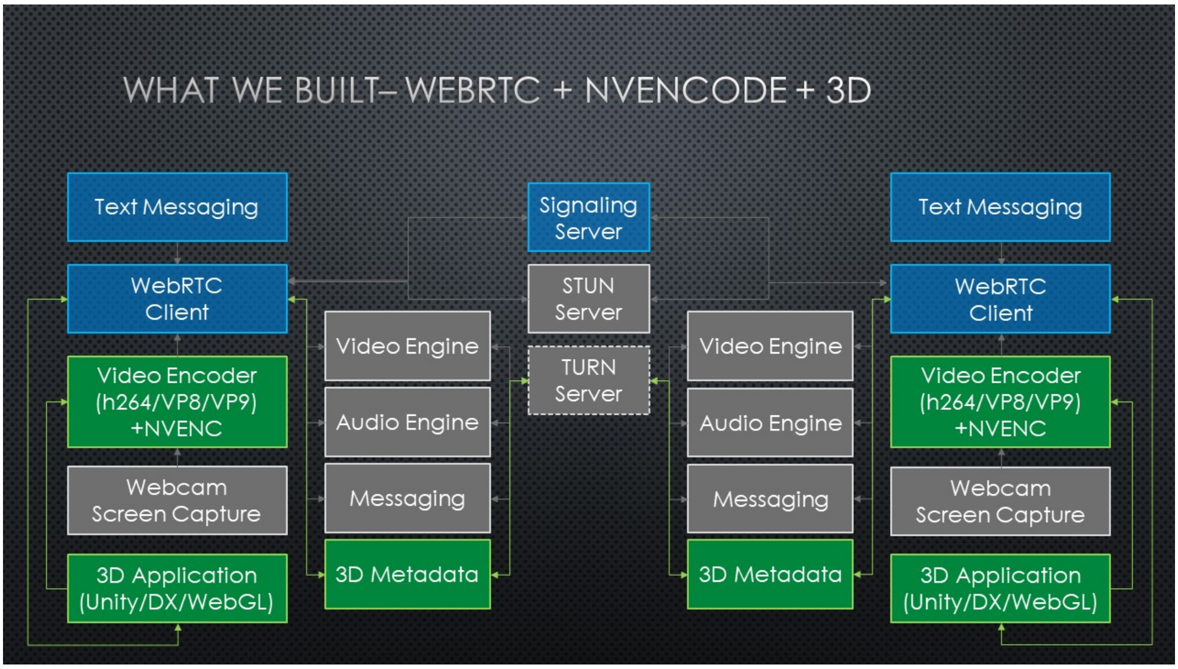


Figure 3.4: *Diagram of WebRTC and NVEncode technologies extended with 3DStreamingToolkit components (in green). Figure adapted from [102].*

toolkit provides addition to the typical WebRTC usage such as an NVIDIA NVEncode hardware encoder library for real-time encoding of 3D rendered content; and a dedicated data channel to manage the camera transforms and the user interaction events. This channel is used to update the HoloLens camera position in the rendering server when the user moves through the room.

Among others, some of the necessary prerequisites to use the toolkit are Windows 10, Visual Studio 2017, Windows 10 SDK - 10.0.14393.795, NVIDIA GPU with NVIDIA drivers CUDA Toolkit 9.1 (for NVEncode) and Node js [103] installed.

The hardware architecture used to establish the communication includes three components: a router, a desktop Windows server (hosting the rendering server app and the signaling server app), and the HoloLens 1 running the DirectX HoloLens Client.

3.2.3.1 Signaling server and networking

For reasons such as control, reliability, transmission speed and latency, a local network has been preferred for communication instead of internet. The peers interact with the signaling server to share the handshakes and start a direct peer-to-peer transmission.

After this point, the actual data are sent directly between client and server. While the traffic and computation load of the signaling server is low, it is still a core component of the WebRTC connection architecture. To simplify the overall architecture and improve communication speed, the signaling server has been deployed on the same windows desktop machine that runs the rendering server.

The signaling server has been cloned from [104] and installed using the Node Package Manager of Node.js [103].

Once installed, the signaling server is started with the simple command ”node ./server.js” on the command prompt.

3.2.3.2 Server

The server has been built from other researchers in Cyber Surgery using the Unity game engine and is designed to offload the heavy GPU rendering task from the HoloLens client. It is meant to run in a Windows OS and makes use of the following technologies:

- NVIDIA drivers and CUDA library to render and encode the scene frames which will be sent to the HoloLens client. Most NVIDIA graphics cards include dedicated hardware for video encoding, and NVIDIA’s NVEncode library provides complete offloading of video encoding without impacting the 3D rendering performance.

- The WebRTC open source project [101], released by Google in 2011 for the development of real-time communications between apps, including low latency audio and video applications. Communication between peers is managed through one or more data channels.

The server makes use of a native plugin of the 3DStreamingToolkit build pipeline. The plugin negotiates with clients to configure a stream, and for encoding and sending visual frame data from the server to the client.

Also on the server side the WebRTC configuration file needs to be adjusted according to the specification at [102]. Since a local network has been used, the “iceConfiguration” has been set to none, the “serverUri” to server system’s IP and “port” to 3000.

3.2.3.3 HoloLens client

The HoloLens client is a DirectX client which connects to the signaling server for handshaking, to finally establish a peer-to-peer connection with the Rendering Server via WiFi in order to receive the rendered frames as a stream, and send back to the Rendering Server updates concerning the HMD's position and rotation via the dedicated data channel. The Rendering Server at this point periodically updates the view according to the newly received coordinates of the HoloLens HMD in the world.

The HoloLens client application has been built in Visual Studio 2017. The core scripts employed by the client are the DirectX HoloLens client sample, to which a JSON file has been added to communicate with the surgical robot and transmit the pose of the robot, obtained by the detection of an ArUco marker (placed at the level of the end effector) to the server.

Once ready to be connected to the server, the configuration file (webrtcConfig.JSON) has been changed to provide the same settings as server side. In particular the IP address, the port and the heartbeat need to be set.

Finally, when successfully launched, the HoloLens client has been detected and selected on the server window and the communication has started.

3.3 Evaluation protocol

In this section the metrics used to describe the quality of the marker-based optical tracking system presented in Section 3.2.1 will be shown. The metrics analyzed are:

- Pose accuracy of the tracked marker: to test how accurately the position and orientation of the marker have been determined by the localization algorithm. This is particularly important in medical context.

- Runtime of the algorithm: to measure how long it takes to process a frame.

- Robustness of the algorithm: to test how the system reacts to different environment conditions, such as lighting and partial masking of the marker.

All these metrics have been evaluated by recording a set of videos containing a various number of markers (1 to 35), with a Huawei P20 camera, using an image resolution of 1080 x 1920 pixels. All tests have been performed using an Intel Core i5-7200U 2.50GHz x 4-core processor with 12GB RAM running Window10 (10.0.18363).

Correct detection of markers is a critical aspect that must be analyzed to verify that the proposed algorithm is able to obviate redundant information present in the scene, extracting exclusively marker information. To assess the quality of the tracked marker pose, it is necessary to know the marker pose as ground truth. The ground truth used has been extracted from the thresholding process described in Section 3.2.1.1. The indexes have been calculated in MATLAB 2018a [105]; multiple frames containing the marker in grayscale have been analyzed and compared with a mask. The mask has been created using the function roipoly, which creates an interactive polygon tool associated with the image displayed in the current figure and returns the mask as a binary image, setting pixels inside the Region Of Interest to 1 and pixels outside to 0. The segmented image is then compared with the ground truth.

In this thesis, the quality of the proposed algorithm has been assessed calculating the following spatial overlap based metrics:

1. **Accuracy**: measures how well a binary segmentation method correctly identifies or excludes a condition. It is defined by: $Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$, where TP, TN, FP, FN denote true positive, true negative, false positive and false negative, respectively.

2. **Sensitivity**: also called True Positive Rate (TPR) or Recall, measures the portion of positive voxels in the ground truth that are also identified as positive by the segmentation being evaluated. It is defined by: $Sensitivity = \frac{TP}{TP+FN}$.

3. **Specificity**: also called True Negative Rate (TNR), measures the portion of negative voxels (background) in the ground truth segmentation that are also identified as negative by the segmentation being evaluated. It is defined by: $Specificity = \frac{TN}{TN+FP}$.

4. **Precision**: also called positive predictive value (PPV), it is not commonly used in validation of medical images, however it is used to calculate the F-Measure. It

is defined by: $Precision = \frac{TP}{TP+FP}$.

5. **F1-measure**: F-Measure is a trade-off between precision and recall. F-Measure is defined by: $FMS_{\beta} = \frac{(\beta^2+1)\ \cdot PPV\ \cdot TPR}{\beta^2\ \cdot\ PPV\ +\ TPR}$. When $\beta = 1.0$ (precision and recall are equally important), it becomes F1-Measure (FMS1). It is also called the harmonic mean, and it is defined by $FMS = \frac{2 \cdot PPV\ \cdot TPR}{PPV\ +\ TPR}$.

6. **MCC**: or Matthew Correlation Coefficient, is used as performance assessment and has a range of -1 (completely wrong binary classifier) to 1 (completely right binary classifier). It is defined by: $MCC = \frac{TP \cdot TN\ -\ FP \cdot FN}{\sqrt{(TP+FP)\ (TP+FN)\ (TN+FP)\ (TN+FN)}}$.

7. **Dice**: or Dice Similarity Index, measures how similar prediction and ground truth are, by measuring the TP found and penalizing the FP found. It is defined by: $Dice = \frac{2 \cdot TP}{2 \cdot TP\ +\ FP\ +\ FN}$.

8. **Jaccard**: the Jaccard index or Jaccard similarity coefficient, measures the similarity and diversity of sample sets. It has a relation with Dice, and it is defined by: $Jaccard = \frac{TP}{TP\ +\ FP\ +\ FN}$.

True Negative Rate and False Positive Rate as function of the marker size in an image have also been evaluated.

Runtime of the algorithm is another important parameter to consider, in fact the algorithm being part of a real time application needs to have a fast processing time. A timer has been integrated in the algorithms to get the time the system needs to process one frame. The running time has been measured at a distance of 40 cm, a marker size of 4 cm and a static scene with constant lighting conditions. Furthermore, to ensure consistent results, the measurements have been performed several times and finally the average of the detection rate per 100 frames has been determined.

As for the **robustness to external influences**, it is important to consider the environment in which a marker system is used. In fact, parameters as lighting conditions and partial masking of the marker play an important role. For these tests a marker grid has been used. First the detection rate has been determined with three different light conditions, starting with low light to direct strong light as often happens in OR. Moreover, in the OR it is often required to cover the marker with a protective transparent foil, which can lead to specular highlights and therefore to partial masking of the marker. To simulate this scenario the marker grid has been overlaid firstly with a transparent undamaged foil, and then with a strongly wrinkled foil.

Chapter 4

RESULTS

This section summarizes the results of the experiments conducted in Chapter 3. In particular, results of the statistical tests performed and metrics used to evaluate the quality of the tracking algorithm will be shown in Section 4.1, while results relative to the server client communication and visualization of 3D augmented information will be described in Section 4.2.

4.1 Evaluation protocol

In this section the results of the statistical tests performed to evaluate the quality of the marker-based optical tracking system will be presented.

4.1.1 Localization accuracy of the pose estimation

To determine how accurately the position and orientation of the marker is performed by the localization algorithm is particularly important in medical context. It has been found that with increasing distance between the sensor and the marker the accuracy decreases, while keeping the distance fixed accuracy improves with increasing marker size. For these reasons all the tests have been performed at a distance of 40-60 cm from the sensor to the marker, using a marker size of 3-5 cm.

The performance of the proposed algorithm has been assessed calculating several spatial overlap based metrics on individual frames from video sensor data. All the

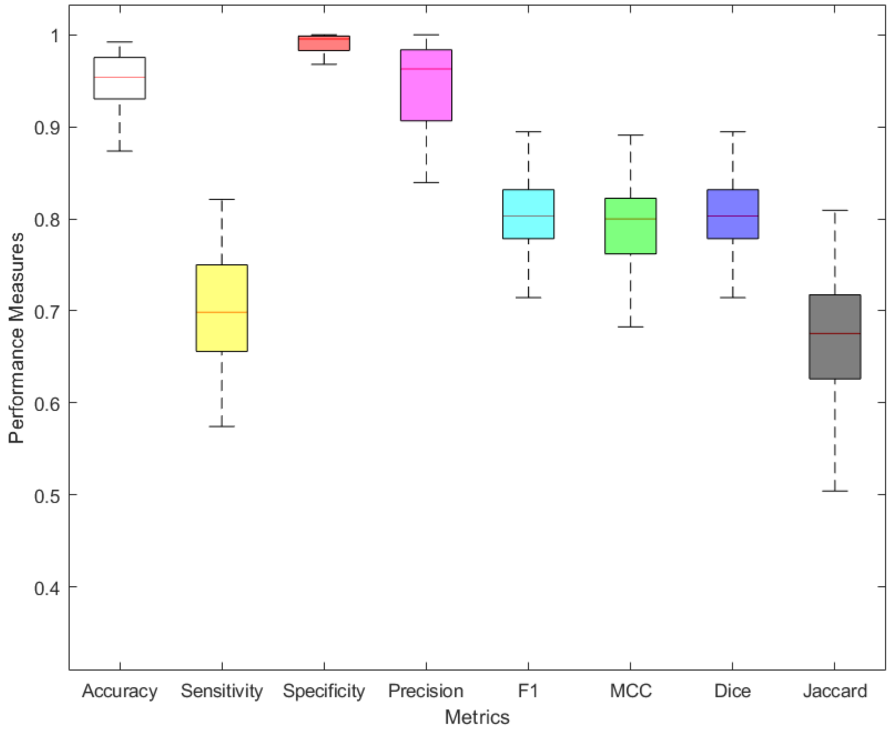


Figure 4.1: *Boxplot of the different metrics used to evaluate the localization accuracy of the pose estimation.*

items have been individually tested to verify the presence of a match between the frame under test and the ground truth system for each video frame. The performance on each individual frame has been then averaged over all the frames in the experiment to develop performance scores. The statistics of the computed performance measures are reported in Figure 4.1 which shows the relative boxplots; while Mean values and Standard Deviation (SD) are listed in Table 4.1.

Specificity has shown the highest value among the indexes. It indicates the ability to correctly generate a negative result when the marker is not present in the scene (high TNR), thus the algorithm rarely gives positive results in absence of marker.

Other than with overlap-based metrics, the TNR and FPR as function of the marker size in an image have also been evaluated. The TNR resulted equal to one in all cases tested. As it is a binary problem (FPR=1-TNR), the FPR resulted zero.

Table 4.1: *Mean value and SD of the metrics used to evaluate the localization accuracy of the pose estimation.*

Metric	Mean	SD
Accuracy	0.94	±0.04
Sensitivity	0.7	±0.1
Specificity	0.98	±0.03
Precision	0.94	±0.07
F1	0.8	±0.07
MCC	0.78	±0.08
Dice	0.8	±0.07
Jaccard	0.67	±0.1

4.1.2 Runtime

Results of the experiments done to determine the time required to process a frame to detect and to estimate the pose of a marker of size 4 cm at a camera distance of 40 cm in a static scene with constant lighting conditions, are presented in Table 4.2. In addition, the runtime of the same methods with several markers arranged on a 5x4 grid has been calculated (Table 4.3).

Table 4.2: *Average time of detection and pose estimation on a single marker.*

Process	Runtime ($\frac{ms}{frame}$)
Marker detection	224.74
Pose estimation	231.07

Table 4.3: *Average time of detection and pose estimation on a 5x4 marker grid.*

Process	Runtime ($\frac{ms}{frame}$)
Marker detection	502.68
Pose estimation	552.65

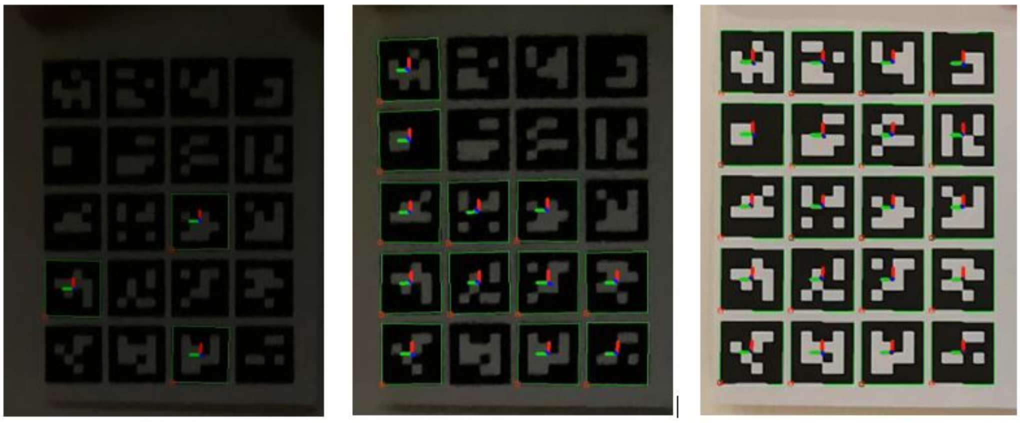


Figure 4.2: *Detection of an ArUco marker grid under three different light conditions. From left: low black light, increased black light and bright light.*

4.1.3 Robustness to external influences

The detection rates of the markers for each of the following conditions have been calculated dividing the number of markers correctly identified by the total number of markers in the grid.

The images in Figure 4.2 show a snapshot of the camera detecting the marker grid at the three selected light conditions (dark, medium and bright). Detection rates of each trial are shown in Table 4.4. Even with medium background light the detection of ArUco is possible with 42.7% in almost half of all images, while with a very low background light the rate decreases at 3.5% .

Table 4.4: *Detection rates under different lighting conditions.*

Light condition	Detection rate
Low black light	3.5 %
Increased black light	42.7%
Bright light	100%

When covering the marker grids with transparent foils to simulate the sterile covering in the operating room, the detection results were different depending on how strongly the foil was wrinkled and thus producing stronger light reflections.

Figure 4.3 shows the detection of an ArUco marker grid that has been covered with both a smooth foil and a wrinkle-rich foil. Image on the right clearly shows a resulting stronger reflection.

Table 4.5 shows the detection rates for these cases. In none of the cases it has been possible to achieve an optimal detection rate. Furthermore marker grids covered with a smooth film were better detected than the ones covered with a wrinkled foil in which only rates around 45% could be achieved.

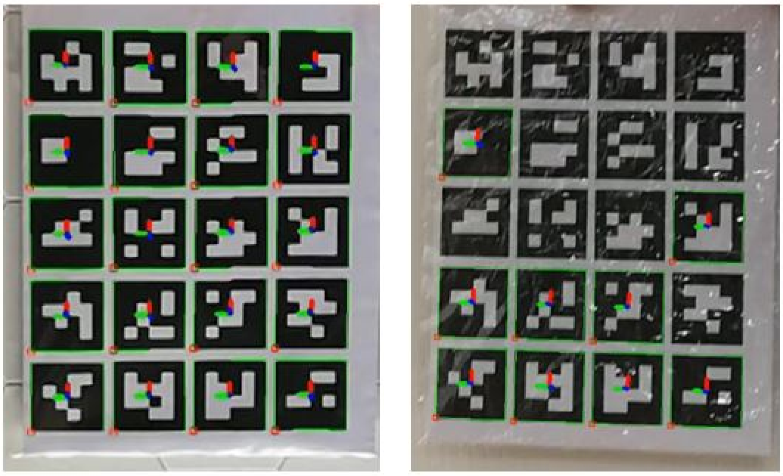


Figure 4.3: *Snapshots of the detection of ArUco marker grid covered with largely smooth foil (left) and covered with a wrinkle-rich foil (right).*

Table 4.5: *Detection rates with smooth and wrinkle-rich foil covers.*

Foil	Detection rate
Smooth foil	98 %
Wrinkled foil	47 %

In Figure 4.4 are shown the boxplot of metrics used to evaluate the localization accuracy in presence of black light and in presence of a wrinkled foil covering the marker. Significant differences have been observed in Dice index and the Sensitivity values if compared with the ones of the detection performance in optimal conditions (cfr. Section 4.1.1, Figure 4.1). Reduced marker detection rates in these conditions can be also observed in the reduction of these values. Accuracy and Specificity have similar values both in the optimal condition and in case of external disturbances.

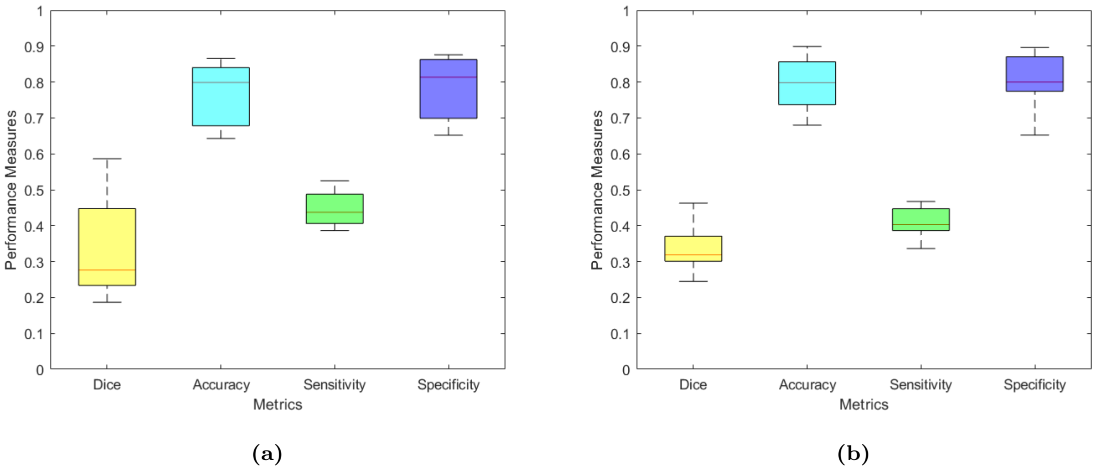


Figure 4.4: *Localization accuracy metrics. (a) Boxplot of the metrics used to evaluate the marker detection ability in condition of medium black light. (b) Boxplot of the metrics used to evaluate the marker detection ability in presence of a wrinkled covering foil.*

4.2 Communication and 3D visualization

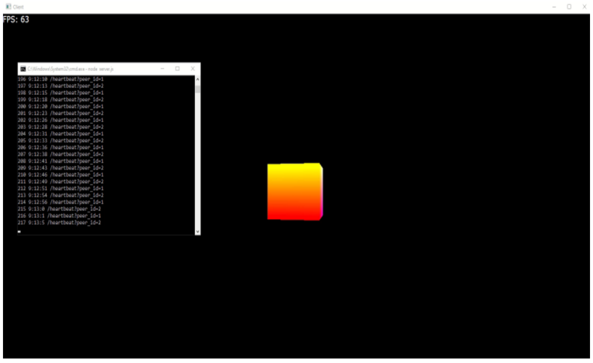
Results relative to server-client communication and visualization of 3D augmented information will be described in this section.

Communication has been successfully set between the rendering server and two different DirectX clients (desktop and HoloLens). Once launched the signaling server, the rendering server and the client have been connected to it and started to exchange information.

As a first trial to test the communication, a spinning cube has been remotely rendered and visualized on a desktop client in the 3S Streaming Toolkit Environment (Figure 4.5). Then the same desktop client has been used to receive a rendered frame implemented on a VTK server (Figure 4.6). These tests have been conducted on a computer receiving the frames from a remote host running both signaling and rendering servers; 60 fps have been obtained almost all the time.

Later, communication with the HoloLens client has been verified. The client has been developed in Visual Studio 2017 as UWP application and then deployed on HoloLens.

Firstly, an application to show spinning cube on the corners of the detected markers has been successfully received (Figure 4.7); then a second rendering showing the model



```
196 9:12:10 /heartbeat?peer_id=1
197 9:12:13 /heartbeat?peer_id=2
198 9:12:15 /heartbeat?peer_id=1
199 9:12:18 /heartbeat?peer_id=2
200 9:12:20 /heartbeat?peer_id=1
201 9:12:23 /heartbeat?peer_id=2
202 9:12:26 /heartbeat?peer_id=1
203 9:12:28 /heartbeat?peer_id=2
204 9:12:31 /heartbeat?peer_id=1
205 9:12:33 /heartbeat?peer_id=2
206 9:12:36 /heartbeat?peer_id=1
207 9:12:38 /heartbeat?peer_id=2
208 9:12:41 /heartbeat?peer_id=1
209 9:12:43 /heartbeat?peer_id=2
210 9:12:46 /heartbeat?peer_id=1
211 9:12:49 /heartbeat?peer_id=2
212 9:12:51 /heartbeat?peer_id=1
213 9:12:54 /heartbeat?peer_id=2
214 9:12:56 /heartbeat?peer_id=1
215 9:13:0 /heartbeat?peer_id=2
216 9:13:1 /heartbeat?peer_id=1
217 9:13:5 /heartbeat?peer_id=2
```

Figure 4.5: *Snapshots of the 3D Streaming Toolkit desktop client connected to the signaling server and receiving a spinning cube from the rendering server.*

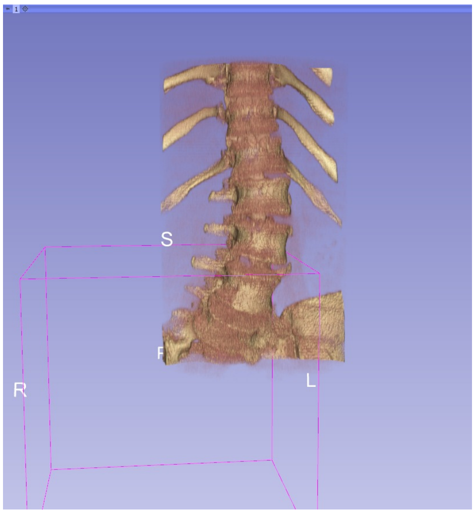


Figure 4.6: *Snapshots of a rendered frame representing a spinal cord implemented on a VTK server to be transmitted to the desktop client.*

of a vertebra developed on a Unity rendering server has been transmitted and effectively visualized on HoloLens as can be seen in Figure 4.8.

Further results and videos used to evaluate the tracking algorithm can be found in this folder.

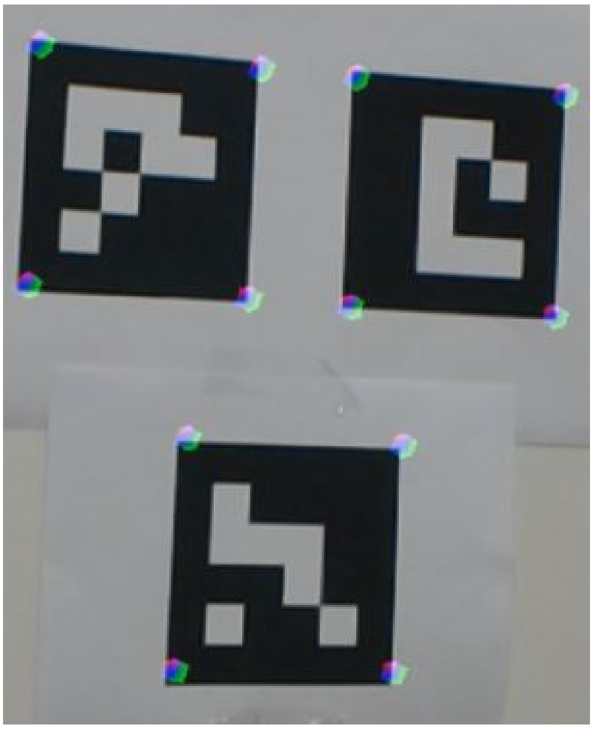


Figure 4.7: *Snapshots of the HoloLens client receiving remotely rendered frames showing spinning cubes positioned on the corners of the detected markers.*

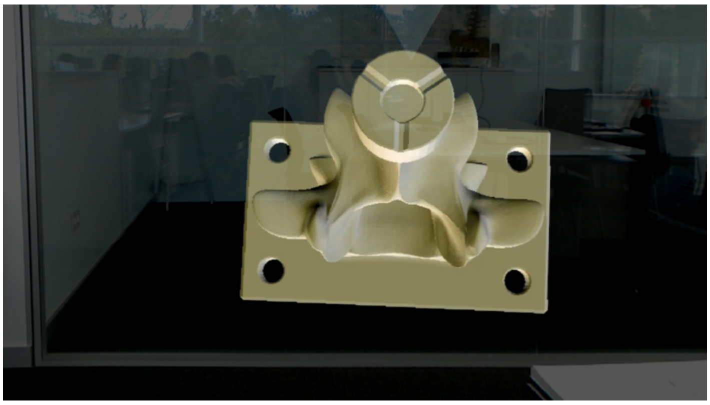


Figure 4.8: *Snapshots of the HoloLens client showing the model of a vertebra that had been previously remotely rendered on a Unity server and then transmitted to the headset through 3D Streaming Toolkit communication architecture.*

Chapter 5

DISCUSSION

In this Section results presented in Chapter 4 will be discussed.

Results of spatial overlap based metrics calculated on individual frames from video sensor data are promising. Among the indexes Specificity shows the highest value (mean=0,98 ±0.03), which is most likely due to the small dimension of the marker and so to the small portion occupied in the scene, which leads to the identification of a lot of TN. This finding is further strengthened by the result of the FPR as function of the marker size in the image, which resulted to be zero in all cases tested.

In Figure 4.1 all the metrics are shown in boxplots. Accuracy (mean=0,94 ±0.04) and Precision (mean=0,94 ±0.07) have the highest values after Specificity. Accuracy is a measure of the actual performance of the system with regard to both correctly detecting and correctly rejecting targets, so a high value can be interpreted as the capability of the algorithm to only consider TP, rejecting FP. This is also confirmed by the high resultant value of Precision, which is the fraction of detected items that are correct, and the high value of the F1-measure (mean=0,8 ±0.07) which gives an estimate of the accuracy of the system under test.

The Dice coefficient is the most used statistical metric in validating segmentations. In addition to the direct comparison between manual and ground truth segmentations, it is common to use the Dice to measure reproducibility (repeatability) and accuracy of manual segmentations and the spatial overlap accuracy [106]. Results show a high value of the Dice coefficient (mean=0,8 ±0.07), suggesting that the outcomes match the ground truth with a high extent thus the marker is detected in the correct position. The F1-measure is mathematically equivalent to Dice [107], in fact its resultant value is the same as Dice.

The MCC index shows how the manually segmented image is correlated with the annotated ground truth. The promising resultant value of the index (0,79 ±0.08) indicates the consistency and capability of the proposed algorithm in correctly identifying, tracking and estimating the pose of markers in the scene.

In this study the optimal size of the marker has been determined based on the detection rate, resulting in a distance of 40-60 cm from the sensor to the marker, using a marker size of 3-5 cm. It has also been observed that the dimension of the marker and the distance from the camera influence the computation speed, in particular the higher is the marker size and the smaller is the distance from the camera, the faster is the detection. This finding is in accordance with the literature [88].

Regarding the runtime, the obtained results (224.74 ms for single marker detection and 231.07 ms for pose estimation) have been found to be a bit higher than the literature [91]. However, the selection of the camera has also an influence on the runtime of the marker detection, hence the obtained runtime is supposed to be influenced by the used hardware (both camera and computer used to process the frames). It has also been observed that increasing the number of markers in the scene the runtime increases.

About findings on the capability of the algorithm to correctly identify markers in worse conditions like partial occlusion, results show that noise resilience decreases (detection rate = 47%) when covering the markers with a wrinkled foil. This is due to the higher reflection that is generated, thus the inability of the algorithm to find the corners of the markers and estimate their indexes. While for the illumination criteria it is difficult to get clear evaluation, however observations indicated that the ArUco markers were robust to heavy illumination in a smaller area. This can be thought of as connected to the partial occlusion experiment, where the heavy illumination almost entirely hides the image in that location. While illumination changes across the entire scene let the detection rate decrease up to 3.5% in low black light, suggesting the inability of the algorithm to detect markers in condition of reduced illumination.

Results from the boxplots of metrics used to evaluate the algorithm performance in these conditions show that the match with the ground truth is low (Dice), so the marker is often detected in the wrong position. On the other hand, Accuracy and Specificity have similar values both in the optimal condition and in case of external disturbances, and this is due to the fact that even with worse conditions a smaller percentage of markers is correctly identified.

The second part of the discussion is related to the communication and 3D visualization. In this case results are still partial as more work is needed to achieve the stated goal. If on one side the communication between the rendering server and the DirectX clients has been successfully set, on the other the client need to be completed integrating the tracking algorithm on the client to send information regarding the marker pose to the robot, in order for the server to read the last transform messages coming from the glasses and tracker, and upload its position. To do so, the tracking applications already adapted for this work need to be fused together to obtain a final application capable to accesses the camera frame and use the OpenCV libraries.

Nevertheless, the current state of the application provides an evidence that this concept and current material can support volumetric rendering with a dedicated server and a remote connection to the headset. In fact, with current tools, the tested application shows the 3DStreamingToolkit desktop client receiving the rendered frame, with 60 fps obtained almost all the time.

This result is promising as the final goal for the use of AR in the surgical world is the achievement of a real-time framerate of 30 fps at least. In fact, human visual system can process 10 to 12 images per second and perceive them individually, while 24 fps are perceived as motion. So, for AR real time applications, 30 fps allow a synchronized display of real and virtual images.

To overcome the problem of insufficient framerate, that impedes a natural interaction with the AR device, a cutting-edge volume rendering technique for remote or progressive rendering is under development for this project. In this way the solution will be able to provide a stereo medical visualization at around 60 fps (30 fps for each eye), and thanks to the use of the 3D Streaming Toolkit it will be possible to connect an unlimited number of peers to a single rendering server, allowing more than one medical operator to visualize the augmented scene.

Chapter 6

CONCLUSIONS AND FUTURE WORK

The project in which this thesis has been developed starts from the idea to provide the surgeon with proper assistance during spinal surgery, by visualizing the target area using an AR device.

To reach this goal a marker tracking algorithm has been developed to detect an ArUco marker system attached to the patient's skin or to the robot end effector to track movements. ArUco markers are reliable, robust, and able to detect and correct errors, they are characterized by a good performance at a wide range of marker orientations and great adaptability to non-uniform illumination conditions [88]. These characteristics led to the choice of this system among the many available.

The tracking algorithm has been developed using open source computer vision libraries and algorithms which have been adapted and extended to match the desired characteristics. The algorithm has been profusely studied and analyzed in terms of reliability and processing speed to verify the compliance with the required qualities. On this purpose several similarity metrics that consider all aspects of a marker system have been used to evaluate the performance of the software in correctly detect and track markers to estimate their pose.

The results on the evaluation of the selected fiducial marker show a high accuracy of the system in correctly estimating the pose. This is true until the light condition is stable, in fact in low light environments the detection rate decreases badly. The same can happen if markers in the scene are too close to one light source or if partial occlusion is present.

To use the developed algorithm directly on HoloLens and integrate it in an AR application, it has been merged on a system which allows real time communication between server and client through the WebRTC protocol. The HoloLens in this architecture has been used as client which receives frames remotely rendered on a desktop server, via a signaling server. The use of remote rendering allows the offloading of heavy GPU tasks from the HoloLens client, increasing the possibility to achieve a real-time framerate of 30 FPS.

At present the client-server system is still at an early stage, however the current state of the application provides an evidence that this architecture and current material may be implemented with positive outcomes. Further analysis will be necessary for performance improvements regarding the tracking algorithm stability for different light conditions and robustness to external environment variation. While, for the AR application the integration of tracking algorithm directly in the headset system is essential to transmit the marker pose to the robot allowing it to adjust its position accordingly. There is also the necessity to improve the rendering server to obtain stereoscopic volume rendering with a sufficient FPS rate to achieve realistic augmented visualization in real time.

It is acknowledged that additional research is required to improve the proposed algorithm and architecture so that, once attained the required adjustments, the presented system has the potential to be used in the medical field.

List of Abbreviations

MIS: Minimally Invasive Surgeries

AR: Augmented reality

PLIF: Posterior lumbar interbody fusion

TLIF: Transforaminal lumbar interbody fusion

ALIF: Anterior lumbar interbody fusion

LLIF: Lateral lumbar interbody fusion

Ax-LIF: Axial lumbar interbody fusion

CT: Computed Tomography

MRI: Magnetic Resonance Imaging

US: Ultrasound

GPUs: Graphics Processing Units

OR: Operating Room

PET: Positron Emission Tomography

SPECT: Single Photon Emission Tomography

DICOM: Digital Imaging and Communication in Medicine

HU: Hounsfield Unit

D3D: Depth 3-Dimensional

VR: Virtual Reality

MR: Mixed Reality

HMD: Head Mounted Display

GPS: Global Positioning System

IMU: Inertial Measurement Unit

UWP: Universal Windows Platform

TP: True Positive

TN: True Negative

FP: False Positive

FN: False Negative

TPR: True Positive Rate

TNR: True Negative Rate

PPV: positive predictive value

FMS1: F1-Measure

MCC: Matthew Correlation Coefficient

SD: Standard deviation

List of Figures

List of Tables

Bibliography

[1] R. A. Deyo, S. Mirza, B. Martin, W. Kreuter, D. Goodman, and J. Jarvik, “Trends, major medical complications, and charges associated with surgery for lumbar spinal stenosis in older adults,” *Jama*, vol. 303, no. 13, p. 1259, 2010.

[2] P. A. Cortesi, R. Assietti, F. Cuzzocrea, D. Prestamburgo, M. Pluderi, P. Cozzolino, P. Tito, R. Vanelli, D. Cecconi, S. Borsa, and et al., “Epidemiologic and economic burden attributable to first spinal fusion surgery,” *Spine*, vol. 42, no. 18, p. 1398–1404, 2017.

[3] A. Aggarwal, “Spinal fusion.” `https://www.mayoclinic.org/tests-procedures/spinal-fusion/about/pac-20384523`, Accessed: 30-05-2020.

[4] M. Tandon, *Chapter 24: Spinal Surgery*, p. 399–439. Hemanshu Prabhakar, 2017.

[5] B. Fiani, S. A. Quadri, M. Farooqui, A. Cathel, B. Berman, J. Noel, and J. Siddiqi, “Impact of robot-assisted spine surgery on health care quality and neurosurgical economics: A systemic review,” *Neurosurgical Review*, vol. 43, no. 1, p. 17–25, 2018.

[6] S. N. Salzmann, P. B. Derman, L. P. Lampe, J. Kueper, T. J. Pan, J. Yang, J. Shue, F. P. Girardi, S. Lyman, A. P. Hughes, and et al., “Cervical spinal fusion: 16-year trends in epidemiology, indications, and in-hospital outcomes by surgical approach,” *World Neurosurgery*, vol. 113, 2018.

[7] K. Kobayashi, K. Ando, Y. Nishida, N. Ishiguro, and S. Imagama, "Epidemiological trends in spine surgery over 10 years in a multicenter database," *European Spine Journal*, vol. 27, no. 8, p. 1698–1703, 2018.

[8] J. Schwender, L. Holly, and E. Transfeldt, *Minimally invasive posterior surgical approaches to the lumbar spine.* Saunders/Elsevier, 5th ed., 2006.

[9] O. Topcu, F. Karakayali, M. Kuzu, and N. Aras, "Comparison of long-term quality of life after laparoscopic and open cholecystectomy," *Surgical Endoscopy*, vol. 17, no. 2, p. 291–295, 2003.

[10] F. Phillips, I. Lieberman, and D. Polly, *Chapter 1: History and Evolution of Minimally Invasive Spine Surgery.* Springer, 2014.

[11] D. Lau, S. J. Han, J. G. Lee, D. C. Lu, and D. Chou, "Minimally invasive compared to open microdiscectomy for lumbar disc herniation," *Journal of Clinical Neuroscience*, vol. 18, no. 1, p. 81–84, 2011.

[12] F. M. Phillips, I. H. Lieberman, D. W. Polly, and M. Y. Wang, *Minimally invasive spine surgery: surgical techniques and disease management.* Springer, 2nd ed., 2019.

[13] Y. Kwoh, J. Hou, E. Jonckheere, and S. Hayati, "A robot with improved absolute positioning accuracy for ct guided stereotactic brain surgery," *IEEE Transactions on Biomedical Engineering*, vol. 35, no. 2, p. 153–160, 1988.

[14] N. Nathoo, M. Çavuşoğlu, M. Vogelbaum, and G. Barnett, "In touch with robotics: Neurosurgery for the future," *Neurosurgery*, vol. 56, no. 3, p. 421–433, 2005.

[15] R. Taylor and D. Stoianovici, "Medical robotics in computer-integrated surgery," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, p. 765–781, 2003.

[16] B. Fiani, S. A. Quadri, V. Ramakrishnan, B. Berman, Y. Khan, and J. Siddiqi, "Retrospective review on accuracy: A pilot study of robotically guided thoracolumbar/sacral pedicle screws versus fluoroscopy-guided and computerized tomography stealth-guided screws," *Cureus*, 2017.

[17] X. Hu, D. D. Ohnmeiss, and I. H. Lieberman, "Robotic-assisted pedicle screw placement: lessons learned from the first 102 patients," *European Spine Journal*, vol. 22, no. 3, p. 661–666, 2012.

[18] N. Keric, D. J. Eum, F. Afghanyar, I. Rachwal-Czyzewicz, M. Renovanz, J. Conrad, D. M. A. Wesp, S. R. Kantelhardt, and A. Giese, "Evaluation of surgical strategy of conventional vs. percutaneous robot-assisted spinal trans-pedicular instrumentation in spondylodiscitis," *Journal of Robotic Surgery*, vol. 11, no. 1, p. 17–25, 2016.

[19] K.-L. Kuo, Y.-F. Su, C.-H. Wu, C.-Y. Tsai, C.-H. Chang, C.-L. Lin, and T.-H. Tsai, "Assessing the intraoperative accuracy of pedicle screw placement by using a bone-mounted miniature robot system through secondary registration," *Plos One*, vol. 11, no. 4, 2016.

[20] D. P. Devito, L. Kaplan, R. Dietl, M. Pfeiffer, D. Horne, B. Silberstein, M. Hardenbrook, G. Kiriyanthan, Y. Barzilay, A. Bruskin, and et al., "Clinical acceptance and accuracy assessment of spinal implants guided with spineassist surgical robot," *Spine*, vol. 35, no. 24, p. 2109–2115, 2010.

[21] G. Vadalà, S. De Salvatore, L. Ambrosio, F. Russo, R. Papalia, and V. Denaro, "Robotic spine surgery and augmented reality systems: A state of the art," *Neurospine*, vol. 17, no. 1, p. 88–100, 2020.

[22] M. M. Mortazavi, S. A. Quadri, S. S. Suriya, S. A. Fard, S. Hadidchi, F. H. Adl, I. Armstrong, R. Goldman, and R. S. Tubbs, "Rare concurrent retroclival and pan-spinal subdural empyema: Review of literature with an uncommon illustrative case," *World Neurosurgery*, vol. 110, p. 326–335, 2018.

[23] S. R. Schroerlucke, M. Y. Wang, A. F. Cannestra, C. R. Good, J. Y. Lim, V. W. Hsu, and F. Zahrawi, "Revision rate in robotic-guided vs fluoro-guided minimally invasive spinal fusion surgery: Report from mis refresh prospective comparative study," *The Spine Journal*, vol. 17, no. 10, 2017.

[24] D. Yu, J. S. Jin, S. Luo, W. Lai, and Q. Huang, "A useful visualization technique: A literature review for augmented reality and its application, limitation and future direction," *Visual Information Communication*, p. 311–337, 2009.

[25] J. T. Gibby, S. A. Swenson, S. Cvetko, R. Rao, and R. Javan, "Head-mounted display augmented reality to guide pedicle screw placement utilizing computed tomography," *International Journal of Computer Assisted Radiology and Surgery*, vol. 14, no. 3, p. 525–535, 2018.

[26] L. T. De Paolis, *Augmented Reality in Minimally Invasive Surgery*, vol. 55, p. 305–320. Springer, 2010.

[27] M. Wieczorek, A. Aichert, O. Kutter, C. Bichlmeier, J. Landes, S. Heining, E. Euler, and N. Navab, "Gpu-accelerated rendering for medical augmented reality in minimally-invasive procedures," *Chair for Computer Aided Medical Procedures (CAMP)*, p. 102–106, 2010.

[28] L. Trestioreanu, "Holographic visualisation of radiology data and automated machine learningbased medical image segmentation," 2018.

[29] J. M. Fitzpatrick, "The role of registration in accurate surgical guidance," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 224, no. 5, p. 607–622, 2009.

[30] D. W. Roberts, J. W. Strohbehn, J. F. Hatch, W. Murray, and H. Kettenberger, "A frameless stereotaxic integration of computerized tomographic imaging and the operating microscope," *Journal of Neurosurgery*, vol. 65, no. 4, p. 545–549, 1986.

[31] P. J. Kelly, B. A. Kall, S. Goerss, and F. Earnest, "Computer-assisted stereotaxic laser resection of intra-axial brain neoplasms," *Journal of Neurosurgery*, vol. 64, no. 3, p. 427–439, 1986.

[32] I. Cabrilo, K. Schaller, and P. Bijlenga, "Augmented reality-assisted bypass surgery: Embracing minimal invasiveness," *World Neurosurgery*, vol. 83, no. 4, p. 596–602, 2015.

[33] W. Lorensen, H. Cline, C. Nafis, R. Kikinis, D. Altobelli, and L. Gleason, "Enhancing reality in the operating room," *Proceedings Visualization '93*, p. 410–415, 1993.

[34] W. E. L. Grimson, G. J. Ettinger, S. J. White, P. L. Gleason, T. Lozano-Pérez, W. M. Wells, and R. Kikinis, "Evaluating and validating an automated registration system for enhanced reality visualization in surgery," *Lecture Notes in Computer Science Computer Vision, Virtual Reality and Robotics in Medicine*, p. 3–12, 1995.

[35] F. Saucer, A. Khamene, B. Bascle, and G. J. Rubino, "A head-mounted display system for augmented reality image guidance: Towards clinical evaluation for imri-guided nuerosurgery," *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2001 Lecture Notes in Computer Science*, p. 707–716, 2001.

[36] S. Bernhardt, S. A. Nicolau, L. Soler, and C. Doignon, “The status of augmented reality in laparoscopic surgery as of 2016,” *Medical Image Analysis*, vol. 37, p. 66–90, 2017.

[37] M. Bajura, H. Fuchs, and R. Ohbuchi, “Merging virtual objects with the real world,” *ACM SIGGRAPH Computer Graphics*, vol. 26, no. 2, p. 203–210, 1992.

[38] E. Marzano, T. Piardi, L. Soler, M. Diana, D. Mutter, J. Marescaux, and P. Pessaux, “Augmented reality-guided artery-first pancreatico-duodenectomy,” *Journal of Gastrointestinal Surgery*, vol. 17, no. 11, p. 1980–1983, 2013.

[39] P. Pessaux, M. Diana, L. Soler, T. Piardi, D. Mutter, and J. Marescaux, “Towards cybernetic surgery: robotic and augmented reality-assisted liver segmentectomy,” *Langenbeck’s Archives of Surgery*, vol. 400, no. 3, p. 381–385, 2014.

[40] O. Ukimura and I. S. Gill, “Imaging-assisted endoscopic surgery: Cleveland clinic experience,” *Journal of Endourology*, vol. 22, no. 4, p. 803–810, 2008.

[41] J. S. Yoo, D. S. Patel, N. M. Hrynewycz, T. S. Brundage, and K. Singh, “The utility of virtual reality and augmented reality in spine surgery,” *Annals of Translational Medicine*, vol. 7, no. S5, 2019.

[42] B. Carl, M. Bopp, B. Saß, B. Voellger, and C. Nimsky, “Implementation of augmented reality support in spine surgery,” *European Spine Journal*, vol. 28, no. 7, p. 1697–1711, 2019.

[43] Y. Abe, S. Sato, K. Kato, T. Hyakumachi, Y. Yanagibashi, M. Ito, and K. Abumi, “A novel 3d guidance system using augmented reality for percutaneous vertebroplasty,” *Journal of Neurosurgery: Spine*, vol. 19, no. 4, p. 492–501, 2013.

[44] M. A. Kirkman, M. Ahmed, A. F. Albert, M. H. Wilson, D. Nandi, and N. Sevdalis, “The use of simulation in neurosurgical education and training,” *Journal of Neurosurgery*, vol. 121, no. 2, p. 228–246, 2014.

[45] A. Elmi-Terander, R. Nachabe, H. Skulason, K. Pedersen, M. Söderman, J. Racadio, D. Babic, P. Gerdhem, and E. Edström, “Feasibility and accuracy of thoracolumbar minimally invasive pedicle screw placement with augmented reality navigation technology,” *Spine*, vol. 43, no. 14, p. 1018–1023, 2018.

[46] G. Burstrom, R. Nachabe, O. Persson, E. Edström, and A. E. Terander, "Augmented and virtual reality instrument tracking for minimally invasive spine surgery," *Spine*, vol. 44, no. 15, p. 1097–1104, 2019.

[47] J. R. Mascitelli, L. Schlachter, A. G. Chartrain, H. Oemke, J. Gilligan, A. B. Costa, R. K. Shrivastava, and J. B. Bederson, "Navigation-linked heads-up display in intracranial surgery: Early experience," *Operative Neurosurgery*, vol. 15, no. 2, p. 184–193, 2017.

[48] M. Kosterhon, A. Gutenberg, S. R. Kantelhardt, E. Archavlis, and A. Giese, "Navigation and image injection for control of bone removal and osteotomy planes in spine surgery," *Operative Neurosurgery*, vol. 13, no. 2, p. 297–304, 2017.

[49] L. Chen, F. Zhang, W. Zhan, M. Gan, and L. Sun, "Optimization of virtual and real registration technology based on augmented reality in a surgical navigation system," *BioMedical Engineering OnLine*, vol. 19, no. 1, 2020.

[50] P. Vávra, P. Zonča, P. Ihnát, M. Němec, and J. Kumar, "Recent development of augmented reality in surgery: A review," *Journal of Healthcare Engineering*, 2017.

[51] F. Ringel, C. Stüer, A. Reinke, A. Preuss, M. Behr, F. Auer, M. Stoffel, and B. Meyer, "Accuracy of robot-assisted placement of lumbar and sacral pedicle screws," *Spine*, vol. 37, no. 8, 2012.

[52] B. Schatlo, R. Martinez, A. Alaid, K. V. Eckardstein, R. Akhavan-Sigari, A. Hahn, F. Stockhammer, and V. Rohde, "Unskilled unawareness and the learning curve in robotic spine surgery," *Acta Neurochirurgica*, vol. 157, no. 10, p. 1819–1823, 2015.

[53] A. P. Dhawan, H. K. Huang, and D. Kim, "Principles and advanced methods in medical imaging and image analysis," 2008.

[54] D. Ganguly, S. Chakraborty, M. Balitanas, and T. Kim, "Medical imaging: A review," *Communications in Computer and Information Science Security-Enriched Urban Computing and Smart Grid*, p. 504–516, 2010.

[55] D. B. Douglas, D. Venets, C. Wilke, D. Gibson, L. Liotta, E. Petricoin, B. Beck, and R. Douglas, "Augmented reality and virtual reality: Initial successes in diagnostic radiology," *State of the Art Virtual Reality and Augmented Reality Knowhow*, 2018.

[56] D. B. Douglas, M. Iv, P. K. Douglas, A. Anderson, S. B. Vos, R. Bammer, M. Zeineh, and M. Wintermark, “Diffusion tensor imaging of tbi,” *Topics in Magnetic Resonance Imaging*, vol. 24, no. 5, p. 241–251, 2015.

[57] P. Ferroli, G. Tringali, F. Acerbi, M. Schiariti, M. Broggi, D. Aquino, and G. Broggi, “Advanced 3-dimensional planning in neurosurgery,” *Neurosurgery*, vol. 72, no. 1, p. A54–A62, 2013.

[58] D. Douglas, C. Wilke, J. Gibson, J. Boone, and M. Wintermark, “Augmented reality: Advances in diagnostic imaging,” *Multimodal Technologies and Interaction*, vol. 1, no. 4, p. 29, 2017.

[59] K.-H. Hohne, M. Bomans, U. Tiede, and M. Riemer, “Display of multiple 3d-objects using the generalized voxel-model,” *Medical Imaging II*, 1988.

[60] P. S. Calhoun, B. S. Kuszyk, D. G. Heath, J. C. Carley, and E. K. Fishman, “Three-dimensional volume rendering of spiral ct data: Theory and method,” *RadioGraphics*, vol. 19, no. 3, p. 745–764, 1999.

[61] E. K. Fishman, D. R. Ney, D. G. Heath, F. M. Corl, K. M. Horton, and P. T. Johnson, “Volume rendering versus maximum intensity projection in ct angiography: What works best, when, and why,” *RadioGraphics*, vol. 26, no. 3, p. 905–922, 2006.

[62] I. Viola, A. Kanitsar, and M. Groller, “Importance-driven volume rendering,” *IEEE Visualization 2004*, p. 139–146, 2004.

[63] I. Viola and M. E. Gröller, “Smart visibility in visualization,” *Computational Aesthetics in Graphics, Visualization and Imaging*, 2005.

[64] A. Kanitsar, R. Wegenkittl, D. Fleischmann, and M. Groller, “Advanced curved planar reformation: flattening of vascular structures,” *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 2003.

[65] L. Soler, S. Nicolau, P. Pessaux, D. Mutter, and J. Marescaux, “Real-time 3d image reconstruction guidance in liver resection surgery,” *Hepatobiliary Surgery and Nutrition*, vol. 3, p. 73–81, Apr 2014.

[66] D. Douglas, E. Petricoin, L. Liotta, and E. Wilson, “D3d augmented reality imaging system: proof of concept in mammography,” *Medical Devices: Evidence and Research*, vol. Volume 9, p. 277–283, 2016.

[67] D. B. Douglas and R. E. Douglas, "Method and apparatus for three-dimensional viewing of images," Oct 2016.

[68] L. Chen, W. Tang, and N. W. John, "Real-time geometry-aware augmented reality in minimally invasive surgery," *Healthcare Technology Letters*, vol. 4, no. 5, p. 163–167, 2017.

[69] O. Baus and S. Bouchard, "Moving from virtual reality exposure-based therapy to augmented reality exposure-based therapy: A review," *Frontiers in Human Neuroscience*, vol. 8, 2014.

[70] F. Cutolo, A. Meola, M. Carbone, S. Sinceri, F. Cagnazzo, E. Denaro, N. Esposito, M. Ferrari, and V. Ferrari, "A new head-mounted display-based augmented reality system in neurosurgical oncology: a study on phantom," *Computer Assisted Surgery*, vol. 22, no. 1, p. 39–53, 2017.

[71] "Vr headsets and amp; equipment." `https://www.oculus.com/`, Accessed: 05-03-2020.

[72] "Vive, vr headsets series." `https://www.vive.com/eu/`, Accessed: 10-03-2020.

[73] W. Chen, J. Chao, Y. Zhang, J. Wang, X. Chen, and C. Tan, "Orientation preferences and motion sickness induced in a virtual reality environment," *Aerospace Medicine and Human Performance*, vol. 88, no. 10, p. 903–910, 2017.

[74] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani, and M. Ivkovic, "Augmented reality technologies, systems and applications," *Multimedia Tools and Applications*, vol. 51, no. 1, p. 341–377, 2010.

[75] E. E. Lovo, J. C. Quintana, M. C. Puebla, G. Torrealba, J. L. Santos, I. H. Lira, and P. Tagle, "A novel, inexpensive method of image coregistration for applications in image-guided surgery using augmented reality," *Operative Neurosurgery*, vol. 60, p. 366–372, 2007.

[76] "Hololens (1st gen) hardware." `https://docs.microsoft.com/it-it/hololens/hololens1-hardware`, Accessed: 07-08-2019.

[77] "Meta view." `https://www.metavision.com/`, Accessed: 28-10-2019.

[78] "Augmented reality devices and software for industrial tasks." `https://daqri.com/`, Accessed: 04-11-2019.

[79] P. Milgram and F. Kishino, “A taxonomy of mixed reality visual displays,” *IEICE Transactions on Information and Systems*, vol. E77-D, p. 1321–1329, Dec 1994.

[80] A. B. Craig, *Chapter 3 - Augmented Reality Hardware*, p. 69–124. Morgan Kaufmann, 2013.

[81] B. H. Thomas and C. Sandor, “What wearable augmented reality can do for you,” *IEEE Pervasive Computing*, vol. 8, no. 2, p. 8–11, 2009.

[82] H. Lopez, A. Navarro, and J. Relano, “An analysis of augmented reality systems,” *2010 Fifth International Multi-conference on Computing in the Global Information Technology*, p. 245–250, 2010.

[83] A. B. Craig, *Chapter 2 - Augmented Reality Concepts*, p. 39–67. Morgan Kaufmann, 2013.

[84] M. Zorrilla, A. Martin, J. R. Sanchez, I. Tamayo, and I. G. Olaizola, “Html5-based system for interoperable 3d digital home applications,” *2012 Fourth International Conference on Digital Home*, 2012.

[85] Bertelsen, D. Scorza, C. Cortés, J. Oñativia, Escudero, E. Sánchez, and J. Presa, “Collaborative robots for surgical applications,” *ROBOT 2017: Third Iberian Robotics Conference Advances in Intelligent Systems and Computing*, p. 524–535, 2017.

[86] OpenCV, “Open source computer vision library.” ://opencv.org/.

[87] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and R. Medina-Carnicer, “Generation of fiducial marker dictionaries using mixed integer linear programming,” *Pattern Recognition*, vol. 51, p. 481–491, 2016.

[88] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, “Speeded up detection of squared fiducial markers,” *Image and Vision Computing*, vol. 76, p. 38–47, 2018.

[89] A. Sagitov, K. Shabalina, R. Lavrenov, and E. Magid, “Comparing fiducial marker systems in the presence of occlusion,” *2017 International Conference on Mechanical, System and Control Engineering (ICMSC)*, 2017.

[90] M. Christiansen, *Adobe After Effects CC Visual Effects and Compositing Studio Techniques.* Adobe Press, 2013.

[91] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, p. 2280–2292, 2014.

[92] S. Arora and A. Kulkarni, *GPU Approach for Handwritten Davanagari Document Binarization*, vol. 2, p. 299–309. Springer, 2017.

[93] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, p. 62–66, 1979.

[94] G. Bradski and A. Kaehler, *Chapter 11. Camera Models and Calibration*, p. 370–403. O'REILLY Media, 2008.

[95] "Camera calibration and 3d reconstruction." `https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html`, Accessed: 19-08-2019.

[96] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, p. 1330–1334, 2000.

[97] B. FX, "Mocha ae - after effects 2020 edition." `https://borisfx.com/videos/welcome-to-mocha-ae-after-effects-2020-edition/?gclid=CjwKCAjwi_b3BRAGEiwAemPNU6qB5p3S-ahb9NCtjT6hmd2xv4fhn9UVmMRRCYf1ufqFYyeEK5MBXRoC9TAQAvD_BwE`, Accessed: 01-04-2020.

[98] "Hololens for cv." url=https://github.com/microsoft/HoloLensForCV, journal=HoloLensForCV, Accessed: 30-09-2019.

[99] "Holographic face tracking." `https://github.com/Microsoft/Windows-universal-samples/tree/master/Samples/HolographicFaceTracking`, Accessed: 2020-05-30.

[100] Microsoft, "Directx." `https://docs.microsoft.com/it-it/windows/mixed-reality/creating-a-holographic-directx-project`, Accessed: 30-07-2019.

[101] "Webrtc." `https://webrtc.org/`, Accessed: 12-10-2019.

[102] "3d streaming toolkit." `https://3dstreamingtoolkit.github.io/docs-3dstk/`, Accessed: 19-08-2019.

[103] "Node.js." https://nodejs.org/en/, Accessed: 13-11-2019.

[104] "Signaling server for webrtc communication." https://github.com/anastasiia-zolochevska/signaling-server, Accessed: 13-11-2019.

[105] "Matlab 2018a." https://it.mathworks.com/products/new_products/release2018a.html, Accessed: 05-04-2020, Aug. 2019.

[106] K. H. Zou, S. K. Warfield, A. Bharatha, C. M. Tempany, M. R. Kaus, S. J. Haker, W. M. Wells, F. A. Jolesz, and R. Kikinis, "Statistical validation of image segmentation quality based on a spatial overlap index," *Academic Radiology*, vol. 11, no. 2, p. 178–189, 2004.

[107] A. A. Taha and A. Hanbury, "Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool," *BMC Medical Imaging*, vol. 15, no. 1, 2015.