REAL-TIME TRACKING OF ELECTRODE DURING DEEP-BRAIN SURGERY

POLITECNICO DI MILANO

Department of Electronics, Information and Bioengineering
Master of Science in Telecommunication Engineering

POLITECNICO
MILANO 1863

REAL-TIME TRACKING OF ELECTRODE DURING DEEP-BRAIN SURGERY

Supervisors: Prof. Stefano Tubaro
Prof. Dimitri Van De Ville
Co-Supervisors: Dott.ssa Elvira Pirondini

Master Degree Thesis of:
Valerio Dilauro
Student ID: 878443

Academic Year 2017/2018

Abstract

The system developed in this thesis is part of a research project that aims to define and characterize a new model of subcortical stroke relative to primate brain. In particular it has been decided to reproduce a subcortical stroke, by using a thermocoagulation electrode to generate a lesion of the posterior limb of the internal capsule. Before the surgery, a surgical plan is used to evaluate how the electrode must be introduced in the brain to reach the desired brain part. The surgical plan is based on a structural Magnetic Resonance Imaging (MRI) acquired before the surgery, having as reference a stereotactic frame in which the monkey's head is fixed. The stereotactic frame is a neuronavigation guidance that helps the surgeon during the surgery giving him a reference. However, despite this, the surgeon is completely blind during the surgery.

The system developed in thesis aims to help the surgeon confirming in real-time the trajectory of the electrode inside the brain. To do that, an optical tracker device, the Vicon Motion Capture System, has been used to obtain, in real time, the 3D position of the electrode. In particular the Vicon System tracks the metallic stick that has on one end the electrode. The estimated electrode position is sent to the 3D Slicer software, which is usually used to visualize the inner part of the brain during the surgical intervention by means of the available primate's MRI. The system has been tested in a series of simulated surgery interventions and, thanks to the information provided by the developed system, the internal capsule has been every time reached with a significant precision. The system has shown to have an overall latency in the order of 15 ms that does not create problems to the surgeon during the intervention.

Sommario

Il sistema sviluppato per questa tesi è parte di un progetto di ricerca che mira a definire e caratterizzare un nuovo modello di ictus subcorticale relativo al cervello nei primati. In particolare è stato deciso di riprodurre l'ictus subcorticale, usando un elettrodo termocoagulatore per generare una lesione del braccio posteriore della capsula interna. Prima della chirurgia, viene ideato un piano chirurgico per valutare come l'elettrodo debba essere introdotto all'interno del cervello, in modo da raggiungere correttamente la parte desiderata. Il piano chirurgico è basato su un Imaging a Risonanza Magnetica strutturale (MRI) acquisito prima della chirurgia, avendo come referenza un casco stereotassico nel quale è fissata la testa del primate. Tuttavia, nonostante questo, il chirurgo non può vedere la posizione dell'elettrodo all'interno del cervello durante l'intervento.

Il sistema sviluppato in questa tesi ha l'obiettivo di aiutare il chirurgo, confermando in tempo reale la traiettoria dell'elettrodo all'interno del cervello. Per fare questo, è stato usato un dispositivo di tracciamento ottico, il Vicon Motion Capture System, per ottenere in tempo reale la posizione 3D dell'elettrodo tracciando la sua punta metallica. La stima della posizione dell'elettrodo è inviata al software 3D Slicer, che è usualmente usato per visualizzare la parte interna del cervello durante gli interventi chirurgici, attraverso la risonanza magnetica del primate. Il sistema è stato provato in una serie di interventi chirurgici simulati e, grazie alle informazioni ottenute dal sistema sviluppato, la capsula interna è stata raggiunta ogni volta con precisione millimetrica. Il sistema ha mostrato di avere una latenza complessiva nell'ordine dei 15 ms, che non creano problema alcuno al chirurgo durante l'intervento.

Ringraziamenti

Sono stati anni intensi, quelli che mi hanno accompagnato fino a qui. Ho vissuto esperienze bellissime, conosciuto persone fantastiche da ogni parte del mondo che mi hanno trasmesso tanto. Il tutto accompagnato da una città, Milano, e da un'università, il Politecnico, che mi hanno formato sia professionalmente che umanamente, senza dimenticare la mia amata Puglia in cui sono ben salde le mie radici.

Vorrei ringraziare i miei relatori, il professor Tubaro e il professor Van De Ville, per avermi dato la possibilità di svolgere questa tesi e per la loro continua disponibilità nei miei confronti. Ringrazio Elvira e Marco per avermi accompagnato e guidato durante questa tesi, farò sempre tesoro di quello che ho appreso da loro.

Un ringraziamento speciale va alla mia famiglia. Ai miei genitori, che hanno sempre creduto in me, sostenendomi nei momenti più difficili e incitandomi a dare sempre il meglio: *"If I have seen further, it is by standings on the shoulders of giants."*. A Stefano, Adriana e suo marito Adelmo, per avermi sempre accolto a casa come se non fossi mai mancato. Alla nonna Tina e alla nonna Maria, senza i loro insegnamenti non sarei arrivato fin qui. A tutti i miei zii, che mi hanno sempre sostenuto e aiutato. In particolare, a mia zia Marlena, che mi ha sempre accolto e fatto sentire meno la mancanza dei sapori di casa quando ero via. A tutti i miei cugini più piccoli, spero di essere un punto di riferimento per loro e ai miei cugini più grandi per esserlo stati per me.

Un ringraziamento affettuoso va ai miei ex compagni di classe e agli amici di una vita, Nico, Giacomo, Marco L., Cosimo, Lorenzo, Gianmarco e Marco S., sempre uniti a stimolarci e migliorarci l'un l'altro. Ringrazio i miei compagni d'università, compagni di mille avventure, sempre a sostenerci e farci forza a vicenda. Ringrazio Simone e Fabio, compagni di studio. Ringrazio Luca, Shahin e Alessio, per le tante risate nelle serate passate in compagnia.

In ultimo ringrazio Mariachiara, la donna che mi è sempre stata accanto in questi due magnifici anni. La donna che ha gioito insieme a me nei momenti più felici e che mi ha dato forza nei momenti di difficoltà. La ringrazio per aver dato un senso a quello che facevo, nei momenti in cui un senso credevo di averlo perso.

Contents

List of Figures

List of Tables

1 INTRODUCTION

This project was thought in order to help the surgeon during deep brain surgeries on primate. The surgeon utilizes a thermocoagulation electrode, in order to perform a lesion of the internal capsule of a deep brain structure, following a trajectory granted by a previous surgical plan. However, during the surgery, the surgeon is completely blind. Due to this, this project aims to confirm the trajectory with real-time monitoring.

Real-time intraoperative neuronavigation guidance is used more and more nowadays in neurosurgery, with different techniques. This project is based on a combination of a classical neuronavigation guidance (stereotactic frame) with a passive-optical real-time tracking system (Vicon Motion Capture System), using a preoperative imaging (MRI).

With this project, we wanted to investigate what were, if there were, the sources of errors during the surgery. Moreover, we wanted to understand if the system, which has been developed to precisely track in real-time the electrode position during the surgery, could really give an help to the surgeon, improving the precision and the accuracy of the lesion.

To explain how the system has been developed, first it will be introduced the surgical procedure, in order to describe the surgical environments and tools. Then, it will be described the Vicon Motion Capture System, used for the real-time tracking, and how it has been set up and combined with the stereotactic frame and with the MRI. Following, it will be explained how the data have been acquired and processed, in order to visualize the electrode position on the MRI. At the end, it will be showed the results of the tests that have been done, in order to quantify the precision, the accuracy and the overall real-time latency of the system.

2 PROJECT ARCHITECTURE

The objective of this project was to develop a system to precisely track in real-time the electrode's position during a deep brain surgery. Before starting to explain how this system has been developed, the surgical procedure will be described together with the tools used to made the surgery. The latter are: *i)*surgical plan, *ii)*stereotactic frame, *iii)*thermocoagulation electrode. These tools are very important, because they have been the starting points of this project and everything else has been developed around them.

After describing the steps of the surgical procedure, it will be introduced the Vicon Motion Capture System used to track, in real-time, the electrode's tip position. It will be described the system architecture and so the working area, the markers tracked by the infrared cameras of the Vicon and the importance of the stereotactic frame as reference system.

2.1 STEPS OF THE SURGICAL PROCEDURE

This project is a part of a more wide project that aims to engineering and characterizing a new primate model of subcortical stroke. The 25% of stroke cases in clinics are of sub-cortical origin. Therefore, they decided to design a model reproducing this case by performing a lesion of the posterior limb of the internal capsule, which carries corticospinal tracks [5]. In this model, in order to ensure consistency and reproducibility of the effects of the lesion, the latter is produced with the same thermocoagulation electrodes usually used in humans for thalamotomy (TCBA011, Fisher). This system is designed to perform reproducible lesions by closed-loop control of the temperature and current of the electrode. The location of the capsule and the related neurosurgical plan is performed using anatomical magnetic resonance imaging (MRI) of the animals head-fixed in a MRI-compatible stereotactic frame. Then the neurosurgical planning software (Stealth station Medtronic) is used to plan the angle depth of the electro-coagulator in order to reach the internal capsule from the frontal lobe.

Stereotactic frame Stereotactic, from greek "stereos" = "solid" and latin "taxis" = "to arrange", is used for surgical intervention that makes use of three-dimensional coordinate system to locate small targets inside the brain and to perform on them some actions. The mechanical device has head-holding clamps and bars which puts the head in a fixed position in reference to the coordinate system (the so-called zero or origin). A teeth mold is also added to the stereotactic frame in order to improve the model (fig.1).

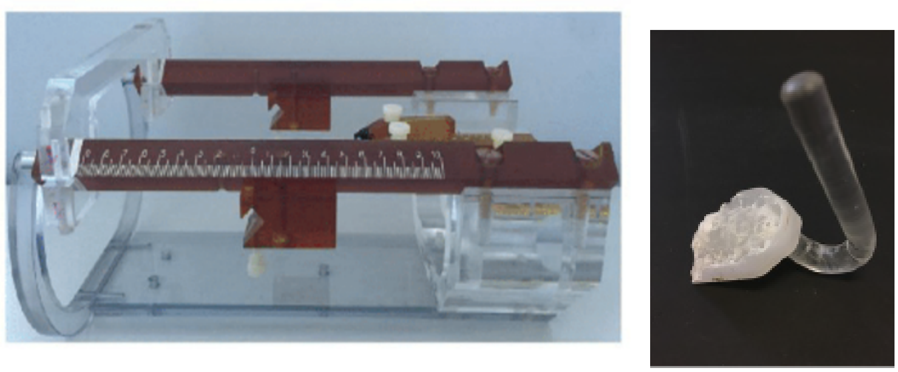


Figure 1: a) Standard stereotactic frame; b) Teeth mold used to keep more fixed the monkey's head

The stereotactic frame is a robust and reliable device that allows to head-fix the primate in a standardize way. This allows to be used during the MRI and in the same way during the surgery. This because the surgical plan, as well as all the measures taken to reach the target, are based on that MRI. In others words, the stereotactic frame is a reference coordinate system for the MRI (and so for the surgical plan) and for the surgery.

Thermocoagulation electrode Thermocoagulation electrodes are used to perform lesions utilizing high frequency currents. This current is delivered from the electrode tip and it is controlled at a desired target temperature, through a temperature sensor.

During the surgery, the electrode is fixed in a “holder arm”, which is fixed to the stereotactic frame (fig.2). The arm is able to move along the stereotactic frame side, up and down and transverse with respect to the stereotactic frame and in two angle: medio-lateral angle and frontal-posterior angle. Once that the stereotactic frame position and angle are fixed, the electrode will move down, in deep, in order to reach the internal capsule.

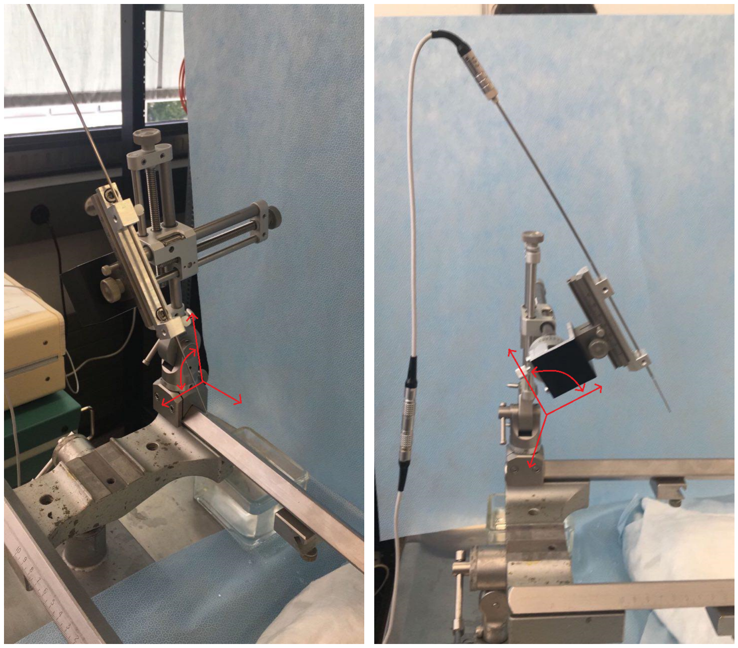


Figure 2: Thermocoagulation electrode fixed in its arm, which is fixed in the stereotactic frame. a)Medio-lateral angulation; b)Frontal-posterior angulation.

Surgical plan Surgical plan is the preoperative method of previsualizing a surgical intervention, in order to pre-define the surgical steps. Given a target to be reached, the surgical plan is done by a software (Stealth station Medtronic) based on a MRI previously taken.

In this specific case, in order to reach the internal capsule, the MRI is done with the stereotactic frame as reference. Then the surgical plan gives as output the trajectory of the surgery and the measures of how the electrode arm has to be set on the stereotactic frame in order to reproduce that trajectory. In this way the surgeon knows how to reach the target in a reliable way.

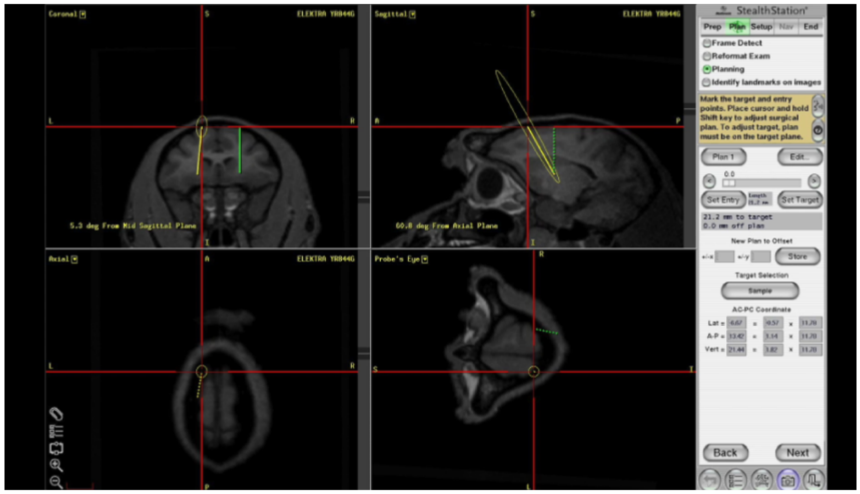


Figure 3: Surgical plan

2.2 VICON MOTION CAPTURE SYSTEM

Vicon is the leading developer of motion capture products and services for the Life Science. Motion capture is the process of recording the movement of objects or people. The device used for the acquisition of the movements can be optical or non-optical; for the optical device a photogrammetric system is used, which is a set of cameras that emit light (red, infra-red or near infra-red) and a set of markers (small spheres) that are made with reflective material. There are many different approaches to motion capture, the main of them are:

- **Optical-Passive**: this technique uses retroreflective markers that are tracked by infrared cameras. It is the most flexible and common method used in the industry.

- **Optical-Active**: this technique uses LED markers connected by wires to the motion capture suit. A battery or charger pack must also be worn by the subject.
- **Video/Markerless**: this technique does not require markers to be worn and instead relies on software to track the subjects' movement. Varying tracking methods yield different results, but real-time and final data error ranges tend to be larger than marker-based solutions.
- **Inertial**: this technique does not require cameras except as a localization tool. Inertial sensors are worn by the subject and the data from the sensors is transmitted wirelessly to a computer.

Passive optical motion capture is the most accurate, flexible and common type of motion capture and it is the area where Vicon specializes. The technology was originated in the life science market for gait analysis, but it is now used widely by VFX studios, sports therapists, neuroscientists and for validation and control of computer vision and robotics.

Vicon motion capture system tracks the position of infra-red reflective markers in space, using high resolution cameras that are equipped with their own infra-red circular strobes mounted on the camera itself. The reflective markers are placed on the object to be tracked and their position, in space, is captured by the cameras and displayed on the monitor in real-time. In this project, the markers were placed on the electrode arm, the cameras acquired their position and then, through a geometric translation, the electrode tip position was evaluated. A very good accuracy and real-time acquisition was required for this project and Vicon satisfied both of them with a mean positioning error of 0.15 mm [4] and a camera frame rate of 330 Hz [8].

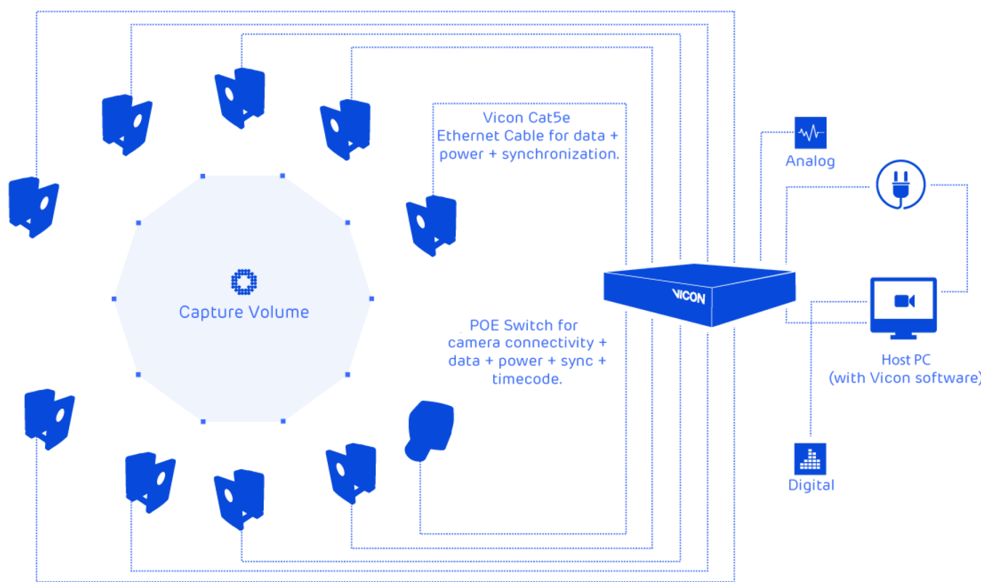


Figure 4: Vicon system standard architecture

The Fig. 4 shows the standard Vicon architecture. It includes a set of cameras, a set of markers, a captured volume, a switcher and a PC where the Vicon software is hosted:

- The **captured volume** represents the volume visible by the cameras. In order to be visible, this volume is previously calibrated with a special wand: this guaranteed precision and repeatability of the data. The wand is also used, after the calibration, to set the three axis and their center point.

- A set of **markers** (small spheres made with reflective material) is placed on the object to be tracked, which need to be in the captured volume in order to be visible by the cameras.

- A set of **cameras** is placed in order to capture a defined volume. A high camera sensor resolution is fundamental for the accuracy of a motion capture system, because the resolution is what "describes" the markers to the system. A high camera speed is also fundamental for any application that involves subjects moving quickly or in a very subtle, nuanced way. Vicon offers a camera resolution of 2.2 MP (2048x1088) and a frame rate of 330 Hz.

- A **POE switcher** provides a single communication point between cameras and PC. The switcher can be used also to synchronize third-part devices.

- A **PC**, with the Vicon software installed, allows displaying, process and manage the motion captured data. It is possible to manage the "preliminary phase" before starting recording and so to start the calibration phase and the masking phase, which is very important to delete and to "mask" all the reflective sources that can interfere with the markers.

Vicon Motion Capture System is widely used in a lot of areas, from object tracking to entertainment, from performance analysis in sport to biomechanics. Vicon is also the only world passive optical motion capture systems that are classed as Medical Devices, as certified by a notified body under ISO 13485 [8]; this allows the wide use, as in the case of this project, for clinical studies and research work. One of this work involved the Vicon in a “Mixed reality simulation of rasping procedure in artificial cervical disc replacement (ACDR) surgery” [2].

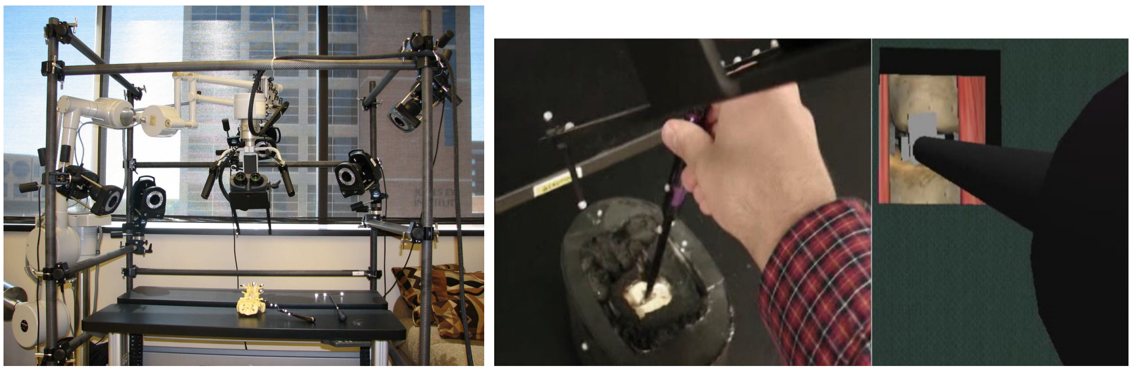


Figure 5: a)Vicon cameras and markers set up for the simulation; b) Simulation of the ACDR surgery using Vicon fro tracking and Mixed Reality for visualizing [8]

This application of the Vicon is similar to the one of this project: the aim was to simulate the ACDR surgery by tracking the scalpel used for the surgery, placing on it some markers and visualizing the procedure using the mixed reality technology. In our project mixed reality is not used, but it is very important to underline that Vicon and its technologies can be successfully used for tricky surgeries that requires high accuracy, improving the precision and minimizing the error sources, which in some cases are independent from the surgeon.

2.3 SYSTEM SET UP

Before discussing about the acquisition and the visualization process, we will see how the whole system was set up, which devices were used and what is the correct procedure in order to acquire and to visualize correctly the data.

There are three main steps involving the set up of the system. The first step is to set up the working area and to initialize the Vicon by calibrating and masking the captured area and setting the axis from the Vicon software. Then the markers have to be placed on the electrode arm and on the stereotactic frame: the marker on the stereotactic frame is set as center of the Vicon axis. This last step is fundamental for the whole procedure, because, by doing this, the stereotactic frame becomes the reference system both for the Vicon global coordinate system and for the local MRI coordinate system; in other words the stereotactic frame is the only common point that links the Vicon system to the co-registered MRI and enable the correct visualization and real-time tracking of the electrode tip position. The third and last step, is to set up the laptop and to launch the code that will start the acquisition and the visualization of the position in real-time on the MRI.

Working area All the developments and tests of the project were done at the “Translational Neurosciences Platform lab” of the University of Fribourg. The lab was equipped with all the Vicon devices and tools that have been introduced in the chapter 2.2: a captured volume where the acquisition were taken, a set of 4 markers and a set of 8 cameras were needed, an active wand to calibrate the captured volume and to set the axis, a POE switcher and a Pc that hosted the Vicon software. In addition, a laptop has been used (the red one in fig. 6), linked by ethernet cable to the POE switcher, in order to acquire, process and visualize the data.

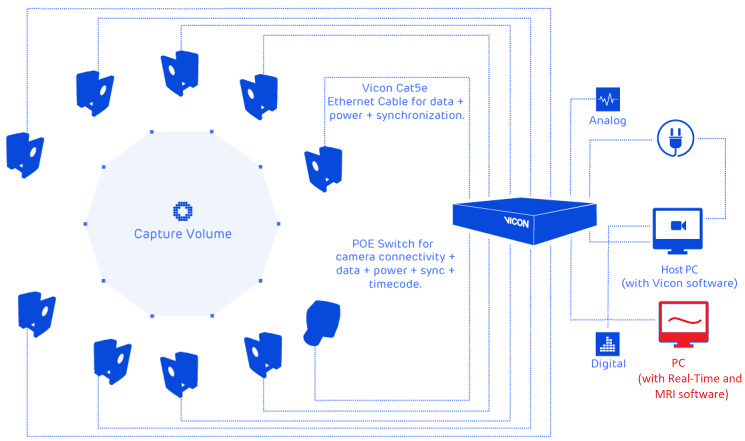


Figure 6: Modified Vicon architecture, with the laptop (red-one) containing the softwares for the acquisition and for the visualization

The area dedicated for the Vicon experiments is approsimately 6m x 4m, but the area captured by the cameras, where the markers are clearly visible, is less then 2m x 2m. This area has been sufficient for all the test needed, because the stereotactic frame needed only a flat surface to be the more stable as possible. Of course, during the surgery the area in this laboratory can't be used; in this case the whole Vicon system has to be integrated in the surgery room and the process applied as well.

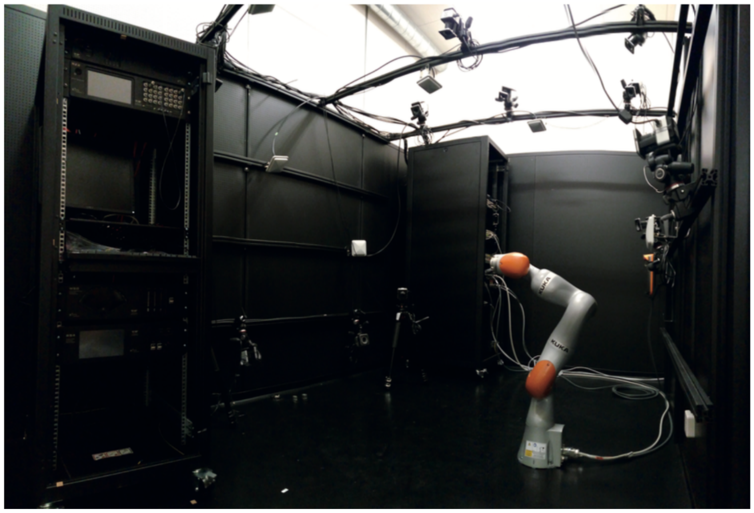


Figure 7: Working room of the "Translational Neurosciences Platform lab" of Fribourg, with Vicon cameras set up.

Before starting the acquisition and the visualization process, three operations have to be applied to the captured area:

- **Masking**: the area has to be masked from all the reflective sources that could interfere with the markers; this operation is done by the Vicon software and after an analysis of all the reflective sources captured by the cameras, it removes and literally masks them.
- **Calibration**: it enables the software to determine the positions, orientations and lens properties of all the Vicon cameras and to produce accurate 3D data. This operation increases the global system accuracy and it has to be done manually, moving in all the directions an active wand until all the cameras are calibrated.

- **Axis setting**: after the calibration phase, the active wand has to be placed on a flat surface and its position is set as center of the three axis by the Vicon software. The center of the axis can be changed later, but the orientation of the axis will remain the same.

These steps end the setting up of the working area and they have to be done each time, before starting the acquisition and visualization of the electrode tip position.

Markers The markers are small spheres covered with reflective material that is captured by the infra-red cameras and they may have different sizes, generally, between 4 mm and 10 mm (fig. 8). The two main problems starting managing the markers have been: how many markers were needed, in order to track successfully the electrode position and where to place them, so that the operation of placing-removing-replacing the markers, in a defined position, would have introduced to the smallest placement error possible. So, the problem was to choose positions that were unequivocal.

Figure 8: Reflective markers in different dimensions

In total, four markers have been used; one for the stereotactic frame to be set as center of the coordinate system and three for the electrode arm to evaluate the tip position. Actually, two markers on the electrode arm were sufficient to evaluate the tip position, but during the tests with two markers, we noticed that they were not captured correctly by the cameras and this caused flickering during the acquisition process. For this "flickering" problems, Vicon offers a solution by creating a model (fig. 10b): from the Vicon software, the markers have been labeled and linked each other with virtual segments, in order to create a model that the software automatically recognize once it is loaded. This guaranteed a stable and a correct acquisition, but creating a model requires at least three markers and so one has been added to the electrode arm.

Once that the number of markers needed has been defined, where to place them has been the next step to be solved. The goal was to find a way to place, remove and re-place the marker exactly in the same position with less error as possible. The solution was to find a strategical position for the markers and to 3D-print them in order to fit exactly in that position. For the marker of the stereotactic frame there was only one possible position (I will explain why in the next paragraph) and so the marker has been 3D-printed with a diameter of 6 mm in order to wedge in there perfectly (fig.9). For the markers on the electrode arm, two positions have been chosen one that had a rectangular shape and one that had a trapezoidal shape. The dimensions of the three shapes have been taken and the markers have been 3D-printed with a diameter of 6 mm (for the rectangular shape) and of 8 mm (for the trapezoidal shape) and with a base that fits exactly on that shapes (fig. 10a).

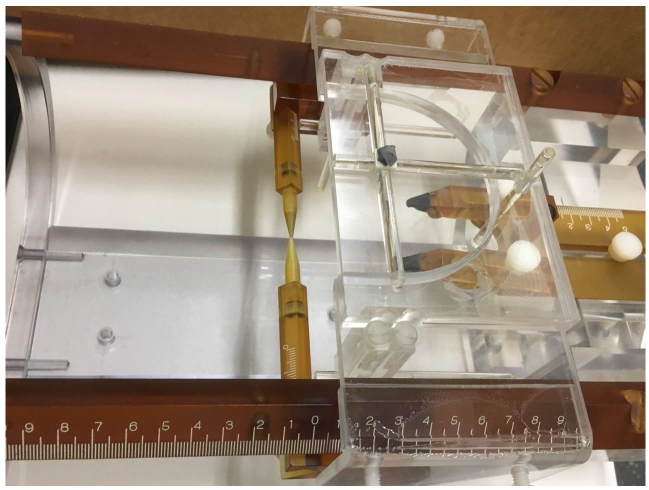


Figure 9: Marker placed in the center of the stereotactic frame

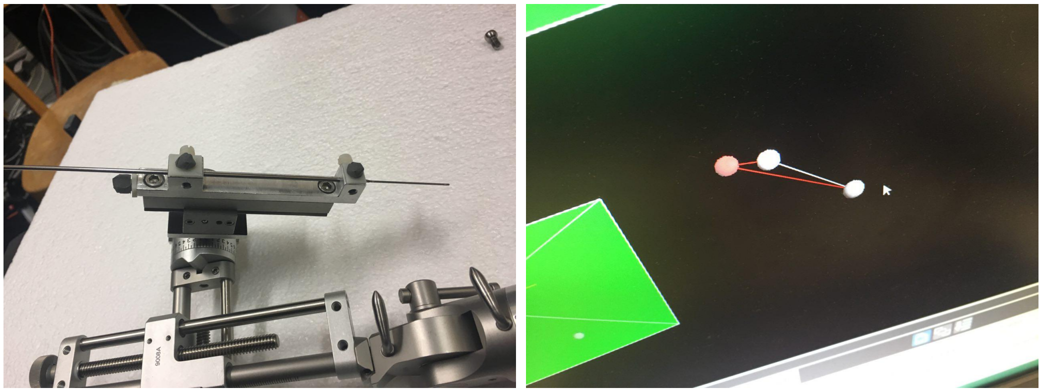


Figure 10: a)Markers placed on the electrode arm.; b)Model created with the three electrode markers, visualized in the Vicon software

Reference system Another big and fundamental step for the project was to find a way to link the two different coordinate systems: the Vicon global coordinate system and the MRI local coordinate system. This because the electrode tip position is visualized on the MRI, but it is tracked by the Vicon, so a common point is needed to be kept as reference during the tracking.

As we know from chapter 2.1, during the MRI the monkey is head-fixed in the stereotactic frame and so there is one point of the stereotactic frame that is clearly visible on the MRI. This point has been chosen as reference point and so that's why the marker of the stereotactic frame, has been placed in that position to be set as center of the coordinate system (fig. 11). Once that this point has been set as center, every electrode movement will refer to that point and the electrode tip position will be correctly visualized on the MRI. This because, during the surgery, the monkey is head-fixed exactly in the same way that it was head-fixed for the MRI, thanks to a personalize dental mold standardize for the stereotactic frame.

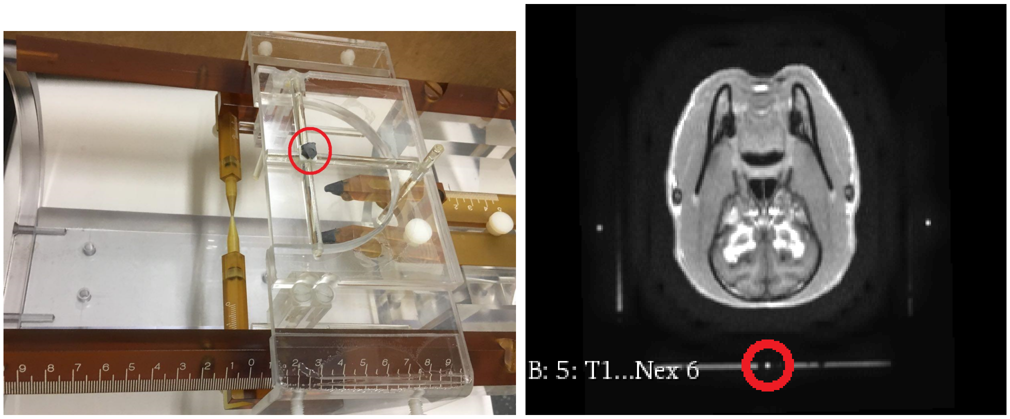


Figure 11: Point of the stereotactic frame visible in the MRI

It is important to underline that the stereotactic frame is the static reference of the whole system and thanks to this complex and standardize device it is possible to link a position in the global coordinate system of the Vicon to the one of the local (the monkey's head) coordinate system of the MRI.

Acquisition and visualization process Once that the working area has been set up and the markers have been placed, the acquisition and visualization process was ready to start. This step has been managed from the laptop, where a C++ code has been launched to acquire the position of the electrode and where the 3D Slicer software visualized that position on the MRI. This code, basically, launched a GUI that allowed to do two things: to connect to the Vicon, starting the acquisition process and to connect to the 3D Slicer software, starting the visualization process.

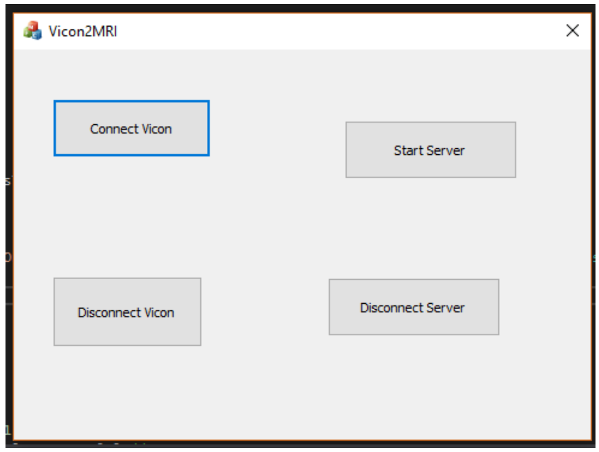


Figure 12: GUI of the application that manage the connection with the Vicon and with the 3D Slicer client

The first step has been to connect, by ethernet cable, the laptop to the POE switcher of the Vicon; this switcher has an IP address that had to be used in the code to start the communication between the Vicon and the laptop. The next step has been to set up the 3D Slicer software: this software is provided with the "OpenIGTlink extension" that enable the acquisition of the position to be visualized. In this software the MRI has been loaded in form of DICOM (Digital Imaging and COmmunications in Medicine) format. Once that the DICOM were loaded and 3D Slicer was "waiting" for the position, the C++ code was launched. Then, from the GUI (fig.12) that appeared, first was started the communication with the Vicon, allowing the acquisition of the markers position, then was started the communication with 3D Slicer, allowing the visualization of the electrode tip position on the MRI. After that the communication have been started , the acquisition and visualization processes continue automatically transmitting data in real-time.

3 PROJECT DEVELOPMENT

After describing all the tools needed for the development of the project, in this chapter it will be explained how the position of the electrode is acquired and processed thanks to a real-time application. Then, it will be explained how the position is visualized on the MRI thanks to the 3DSlicer software.

Two client-server connections have been used to acquire the positions from the Vicon and to send the position to the 3D Slicer software. The real-time acquisition has been done thanks to the ViconDataStream SDK, which guaranteed real-time streaming of data from Vicon switcher to the application. The real-time visualization has been done thanks to the OpenIGTLink extension, which guaranteed real-time streaming of data from the application to the 3D Slicer software.

3.1 ACQUIRING AND PROCESSING PROCEDURE

The acquiring and processing procedure concerned the receiving, the processing and the sending of data by the laptop. This has been done thanks to an MFC application (.exe) that made use of shared DLL. This application had three main roles:

- it acted as client, in order to acquire in real-time the two markers positions from the Vicon server;
- it processed these two positions through a geometrical translation, in order to obtain only the electrode tip position;

- it acted as server, in order to send the electrode tip position to the 3D Slicer software client.

A GUI, with four buttons (fig. 12), allowed to start and to stop the communication with the Vicon and with the 3D Slicer software. Once that both connection was on, the application automatically received, processed and sent the electrode tip position.

The receiving process has been done through a stream socket based on TCP. An ethernet cable has been used to link the client (the laptop with the application) to the server (the Vicon POE switcher), this ensured stability to the connection, avoiding loosing of signal. In order to acquire the data, Vicon provided a package, the Vicon DataStream SDK, allowed easy programmable access to the information contained in the Vicon DataStream. The function calls within the SDK enables users to connect to and request data from the Vicon DataStream: in particular, the SDK provided all the methods to set up the client and so all the requests and instructions that the client could do to the server, like "Connect", "SetStreamMode", "GetFrame", "GetFrameRate", "GetMarkerGlobalTranslation" ecc..., all methods that when called by the client returned a value or a status. In this way it has been possible to acquire in real-time the positions, in the space, of the two markers needed to evaluate the electrode tip position.

At the same time that the positions of the two markers were acquired, they were processed in order to obtain the electrode tip position. This processing involved a geometrical translation based on the fact that the two markers made a virtual line that is parallel to the electrode, so estimating the angle made by this line with the ground, it has been possible to estimate the electrode tip position.

The sending process, has been done through a stream socket based on TCP. In order to establish the connection, the 3D Slicer software (client) was launched with its extension, OpenIGTlink, and put on "waiting status" for a connection. When the application (server) was launched it connected to the port under which 3D Slicer was in "waiting status". In order to send the electrode tip position, a packet was made by the application and it contained the data in form of 4x4 transform matrix, where the last column represented the 3D position and the rest of the matrix represented the orientation in the space. Once that the data have been correctly packed and sent, the position has been visualized, in real-time, on the MRI loaded in the 3D Slicer software.

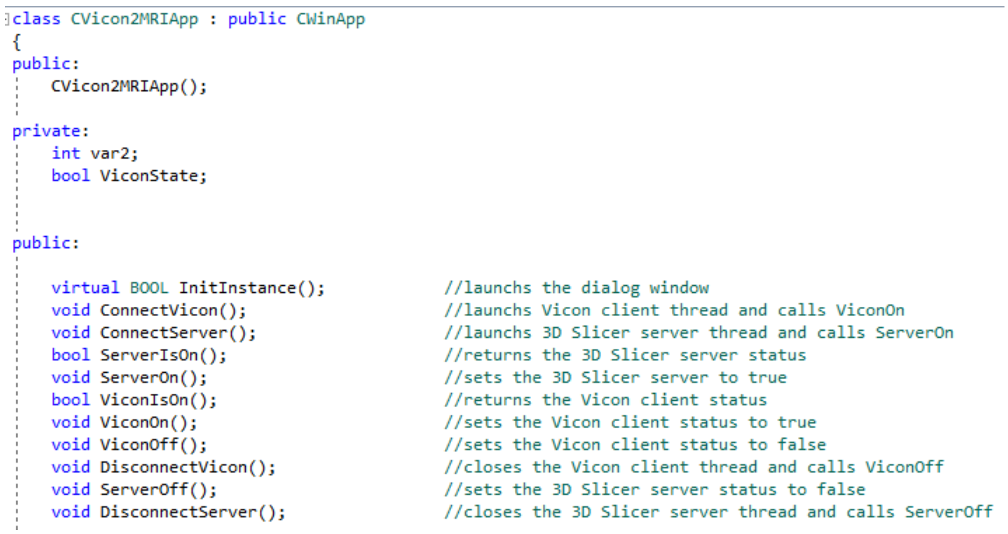
Vicon2MRI application The Vicon2MRI is a MFC application that made use of shared DLL. As its name say, the Vicon2MRI program transfer, in real-time, the Vicon data to the MRI of the primate, previously acquired and then visualized by the 3D Slicer software. An object-oriented programming and the C++ language has been used to made this application, which imported three main libraries:

- the ViconDataStreamSDK_CPP.lib, in order to use all the methods to set up the client for the acquisition of data from Vicon;
- the OpenIGTLink.lib, in order to use all the methods to set up the server for sending data to 3D Slicer;

- the pthreadVC2.lib, because a multithreading execution was required in order to manage the two different connections and the dialogue window.

The main headers of the code was:

- “Client.h”, which imported the ViconDataStream library;
- “pthread.h”, which imported the POSIX Threads library;
- “ViconData.h”, by which the “KinData” class managed the acquisition of the data from Vicon, in particolar it was the class that called the “GetMarkerGlobalTranslation()” method acquiring constantly the markers positions;
- “Vicon2MRIDlg.h”, which set up the dialogue window;
- “Vicon2MRI.h”, which contained the main class of the code, “CVicon2MRIApp”, that managed all the communications and processes.



```
class CVicon2MRIApp : public CWinApp
{
public:
    CVicon2MRIApp();

private:
    int var2;
    bool ViconState;


public:

    virtual BOOL InitInstance();        //launchs the dialog window
    void ConnectVicon();                //launchs Vicon client thread and calls ViconOn
    void ConnectServer();               //launchs 3D Slicer server thread and calls ServerOn
    bool ServerIsOn();                  //returns the 3D Slicer server status
    void ServerOn();                    //sets the 3D Slicer server to true
    bool ViconIsOn();                   //returns the Vicon client status
    void ViconOn();                     //sets the Vicon client status to true
    void ViconOff();                    //sets the Vicon client status to false
    void DisconnectVicon();             //closes the Vicon client thread and calls ViconOff
    void ServerOff();                   //sets the 3D Slicer server status to false
    void DisconnectServer();            //closes the 3D Slicer server thread and calls ServerOff
```

Figure 13: Screen of the "CVicon2MRIApp" class, with all the methods used and explained

In fig. 13 is showed the constructor and the methods of the "CVicon2MRIApp" class, where the dialog window called directly the "ConnectVicon()", the "ConnectServer()", the "DisconnectVicon()" and the "DisconnectServer()" methods. When the "ConnectVicon()" was called, it initialized the Vicon thread and, using the IP address of the POE switcher, the communication with Vicon was started. Once that the Vicon status was changed to "true", the application started acquiring constantly the two markers positions, since the Vicon status was changed to "false" and the communication stopped. The communication with 3D Slicer has been managed in the same way: when the "ConnectServer()" was called, it initialized the 3D Slicer thread and, using the port 18944, the communication with 3D Slicer was started. Once that the server status was changed to "true", the application started processing the two markers positions, evaluating the electrode tip position. Then this position is packed (fig. 14) in form of 4x4 transformation matrix and sent to the 3D Slicer software. This process lasted since the server status was changed to "false" and the communication stopped.

```
if (my_socket.IsNotNull()) // if client connected
{
    //------------------------------------------------------------
    // Create a message buffer to receive header
    igtl::MessageHeader::Pointer headerMsg;
    headerMsg = igtl::MessageHeader::New();
    //------------------------------------------------------------
    // Allocate a time stamp
    igtl::TimeStamp::Pointer ts;
    ts = igtl::TimeStamp::New();
    // loop
    for (int i = 0; i < 100; i++)
    {

        igtl::Matrix4x4 matrix;
        GetTipPosition(matrix);

        transMsg = igtl::TransformMessage::New();
        transMsg->SetDeviceName("Tracker");
        transMsg->SetMatrix(matrix);
        transMsg->Pack();
        my_socket->Send(transMsg->GetPackPointer(), transMsg->GetPackSize());
        igtl::Sleep(interval); // wait

    }
}
```

Figure 14: Screen of the code that create and send, continuously, the packet containing the electrode position

Electrode tip position calculation The evaluation of the electrode tip position has been done using fundamentals of geometry in the three-dimensional space. In particular, the calculation has been made using the notions of a line passing for two points, of a parallel line to another line and of the angle made by a line and a plane.

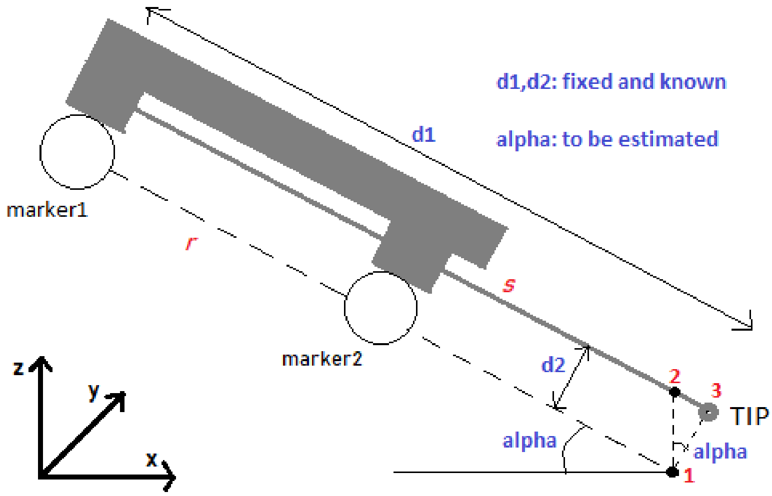


Figure 15: Model of the electrode arm that shows the steps of the geometrical translation for the tip position calculation

Taking as reference the fig.15, the goal was to estimate the point $TIP = (x_{tip}, y_{tip}, z_{tip})$, knowing the position $P_1 = (a, b, c)$ of marker1, the position $P_2 = (e, f, g)$ of marker2, the distance d_1 (between P_1 and TIP) fixed to 160 cm and the distance d_2 (between the line r, made by the two point, and the line s made by the electrode) fixed to 7 mm. Estimating the angle α made by the plane xy and the line r, the calculation reduced to a simple translation of points.

Given the parametric equation of the line r :

$$r : \begin{cases} x = a + v_x * t \\ y = b + v_y * t \\ z = c + vz * t \end{cases}$$

where $v_x = (e-a), v_y = (f-b), v_z = (g-c)$ are the versors of the line r, we need first to evaluate the point $1 = (x_1, y_1, z_1)$, which is the projection of the point TIP on the line r, estimating the angle α . Knowing that the angle between a line and a plane is equal to the complementary acute angle that forms between the direction vector of the line and the normal vector of the plane, if we considering the plane xy, its normal vector is $a_{xy} = [0, 0, 1]$ and the angle will be:

$$\alpha = \arcsin\left(\frac{v \cdot a_{xy}}{\|v\| \cdot \|a_{xy}\|}\right) =$$

$$= \arcsin\left(\frac{v_x * 0 + v_y * 0 + v_z * 1}{\sqrt{v_x^2 + v_y^2 + v_z^2} * \sqrt{0 + 0 + 1^2}}\right) =$$

$$= \arcsin\left(\frac{v_z}{\sqrt{v_x^2 + v_y^2 + v_z^2}}\right)$$

In order to find the point 1 we need to translate, of a distance d_1, the point P_1 by solving the equality:

$$v_x^2 * t + v_y^2 * t + v_z^2 * t = d_1^2$$

which gives:

$$t = \frac{d_1}{\sqrt{v_x^2 + v_y^2 + v_z^2}}$$

by substituting t in the parametric equation of the line r , we obtain the point 1.

Before finding the point TIP, we need to find the point $2 = (x_2, y_2, z_2)$ and the line s passing by TIP and parallel to the line r . Observing that the angle, made by the right-triangle 123, is exactly α and knowing the distance d_2, the point 2 is found by simply translating on the z-direction the point 1:

$$2 = \left(x_1, y_1, z_1 + \frac{d_2}{\cos\alpha}\right)$$

So the line s is:

$$s : \begin{cases} x = x_2 + v_x * t_1 \\ y = y_2 + v_y * t_1 \\ z = z_2 + v_z * t_1 \end{cases}$$

Finally, in order to find the point $TIP = (x_{tip}, y_{tip}, z_{tip})$ we need to translate, of a distance $d_2 * \tan\alpha$, the point 2 by solving the equality:

$$v_x^2 * t_1 + v_y^2 * t_1 + v_z^2 * t_1 = (d_2 * \tan\alpha)^2$$

which gives:

$$t_1 = -\frac{(d_2 * \tan\alpha)}{\sqrt{v_x^2 + v_y^2 + v_z^2}}$$

Substituting t_1 in the parametric equation of the line s, we obtain the coordinates of the electrode tip position.

All this calculations were done, meanwhile that the two markers positions were acquired, by the method "GetTipPosition(matrix)" called before creating and sending the packet to 3D Slicer (fig.14). As it was said before, 3D Slicer software allowed only 4x4 transformation matrix to be sent, so the electrode tip position, once evaluated,was put in the first three rows of the last column of a 4x4 matrix, where the others value represented the orientation of the electrode in the space.

3.2 VISUALIZATION PROCEDURE

The visualization process concerned the visualization of the primate's MRI and of the electrode tip position on it in real time and it was done thanks to the 3D Slicer software. 3D Slicer is an open source software platform for medical image informatics, image processing, and three-dimensional visualization. Built over two decades through support from the National Institutes of Health and a worldwide developer community, Slicer brings free, powerful cross-platform processing tools to physicians, researchers, and the general public. 3D Slicer provides a set of interactive tools and a stable platform that can quickly incorporate new analysis techniques and evolve to serve more sophisticated real-time applications, while remaining compatible with the latest hardware and software generations of host computer systems [3].

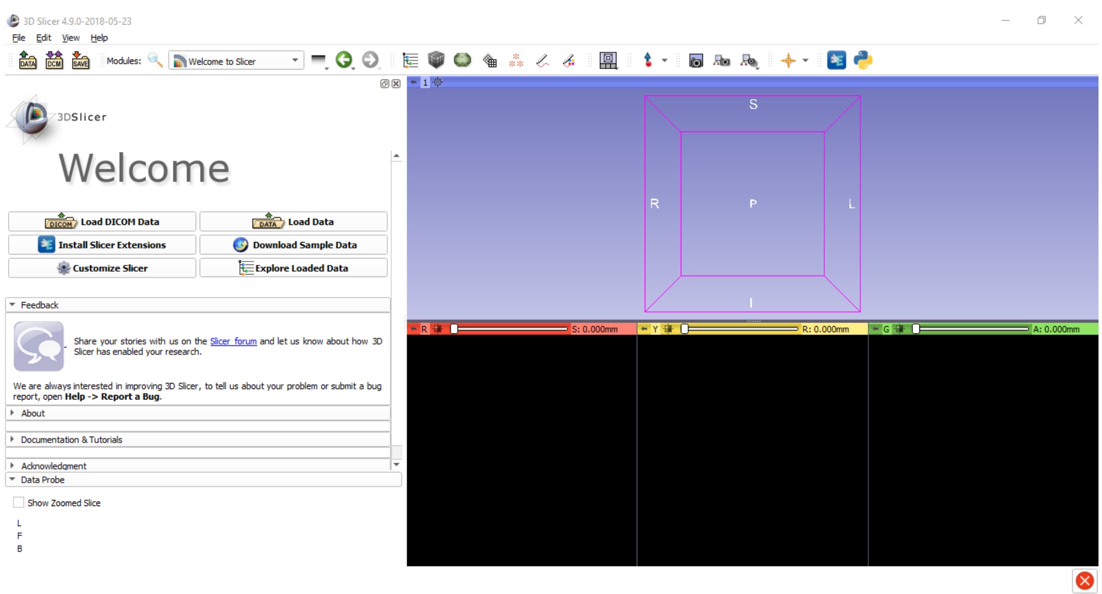


Figure 16: Home view of the 3D Slicer software

3D Slicer is used in a lot of medical applications, included neurosurgery because it allows to load and to process MRI in form of DICOM (Digital Imaging and COmmunications in Medicine) format visualizing it in the three different anatomical planes: axial, sagittal and coronal.

Once that the MRI has been loaded, in order to visualize on it the electrode tip position in real-time, the extension OpenIGTLink has been used. OpenIGTLink is an open-source network communication interface specifically designed for image-guided interventions. It aims to provide a plug-and-play unified real-time communications (URTC) in operating rooms (ORs) for image-guided interventions, where imagers, sensors, surgical robots and computers from different vendors work cooperatively.

Not only the electrode tip position has been tracked and displayed in real-time, but also the orientation of the electrode in the space has been tracked and displayed creating a needle from the "CreateModel" package of the OpenIGTLink extension of 3D Slicer. The needle orientation changed with respect to the electrode orientation and inclination; this because in the 4x4 transformation matrix, sent from the application to the 3D Slicer, also the orientation was contained and evaluated through the quaternion calculation.

DICOM DICOM (Digital Imaging and Communications in Medicine) is the international standard to transmit, store, retrieve, print, process, and display medical imaging information, as the MRI [1]. DICOM groups information into dataset, where each object consists of a number of attributes, including items such as name, ID, etc., and also one special attribute containing the image pixel data (i.e. logically, the main object has no “header” as such, being merely a list of attributes, including the pixel data) and it makes these medical information interoperable.

Once that the DICOM have been loaded in 3D Slicer, the software allowed to manage the slices of the three anatomical planes, sagittal, axial and coronal choosing the slices where the point of the stereotactic frame was visible and setting it as center of the 3D Slicer coordinates. In this way, every moves of the electrode have been tracked in the same way and direction as in the stereotactic frame (fig. 17). During the surgery, the surgeon will be able to choose the best slice in order to better visualized the electrode tip.

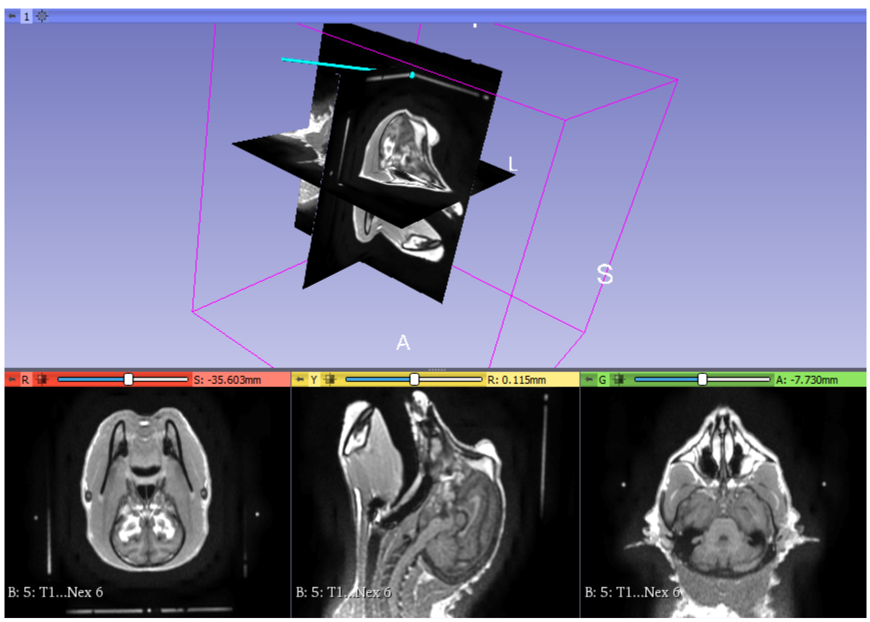


Figure 17: Visualization in 3D Slicer of the MRI, in axial, sagittal, coronal planes

OpenIGTLink extension This extension is an open, simple and extensible network communication protocol for IGT to transfer transform, image and status messages. This extension supports real-time image streaming and it allows to import intraoperative images from various imaging scanners to image-processing software as 3D Slicer. It aims to provide a plug-and-play unified real-time communications (URTC) in operating rooms (ORs) for image-guided interventions, like:

- Stereotactic surgical guidance using optical position sensor and medical image visualization software;
- Intraoperative image guidance using real-time MRI and medical image visualization software;
- Robot-assisted interventions using robotic devices and surgical plan software.

Some tests certified that the protocol was able to transfer position data with submillisecond latency up to 1024 fps and images with latency of $<$10 ms at 32 fps [7].

The setting up of OpenIGTLink has been very easy and user-friendly. It provided a lot of modules to enhance the interaction with the tracked device. In particular:

- with the module "OpenIGTLink IF" it has been possible to manage the connection with the server as client in order to start and to stop the communication (fig. 18a);
- with the module "CreateModel" it has been possible to create the needle that tracked in real-time the orientation and the position of the electrode on the MRI, as we can see in fig. 17;

- with the module “Transforms” it has been possible to manage the transform 4x4 matrix received by the software and also applied the transformation to the needle (fig. 18b);
- with the module “Model” it has been possible to make visible the needle position also in the axial, coronal and sagittal slices view, which is very useful for the surgeon during the surgery.

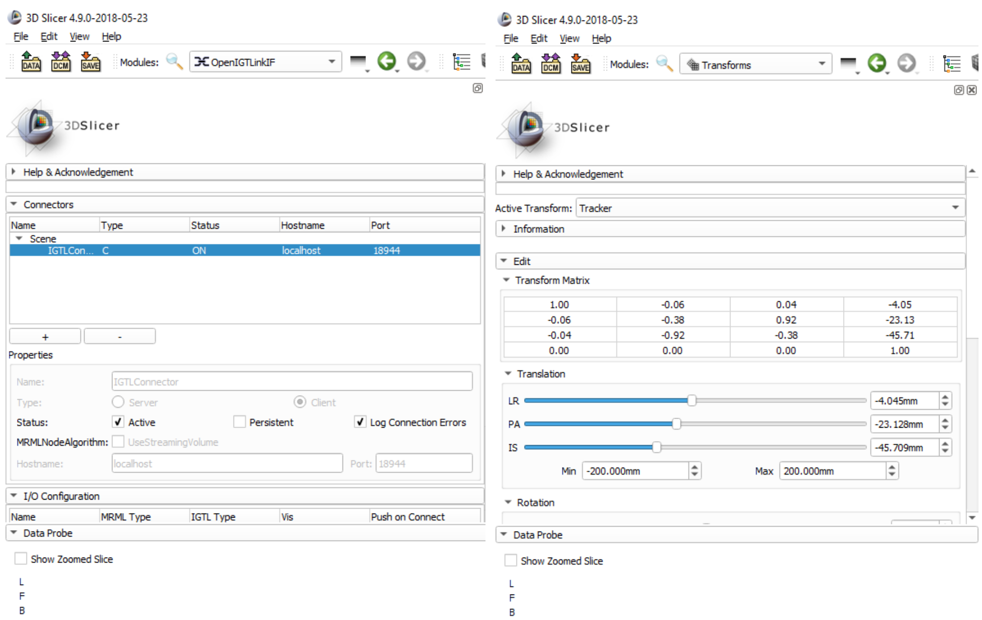


Figure 18: a)Screen of the “OpenIGTLink IF” home, which enables the connection; b)Screen of the “Transforms” module with the transformation 4x4 matrix in RAS coordinates

Once that all modules have been set up, the needle kept tracking the electrode position and orientation in the 3D view and in the slices view of the MRI.

Some image guided devices, already use the OpenIGTLink extension and they have a dedicated interface from where it's possible to manage directly the device. Vicon, actually, it is not one of that devices and this required a "starting from scratch" work, but the extension guaranteed a very feasible and reliable option.

Electrode orientation The tracking of the electrode orientation and rotation in the space has been done through the real-time quaternion calculation. A quaternion algebra is one possible way to represent 3 dimensional orientation, or other rotational quantity, associated with a solid 3D object. In 3-dimensional space, according to Euler's rotation theorem, any rotation or sequence of rotations of a rigid body or coordinate system about a fixed point is equivalent to a single rotation by a given angle θ about a fixed axis (called the Euler axis) that runs through the fixed point. A rotation through an angle of θ around the axis defined by a unit vector $u = (u_x, u_y, u_z) = u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k}$ can be represented by a quaternion using an extension of Euler's formula:

$$q = e^{\frac{\theta}{2}(u_x\mathbf{i}+u_y\mathbf{j}+u_z\mathbf{k})} = \cos\frac{\theta}{2} + (u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k})\sin\frac{\theta}{2}.$$

The calculation reduced to evaluating the angle of rotation and the axis around which rotate. The "starting" vector, from which 3D Slicer started to rotate the needle,$v_s = [0, 0, 1]$ and the "final" vector, made by the two markers position on the electrode, $v = [v_x, v_y, v_z]$ were both known; so the angle of rotation θ was the angle between them and it was given by:

$$\theta = \arccos\left(\left(\frac{v \cdot v_s}{\|v\| \cdot \|v_s\|}\right)\right) =$$

$$= \arccos\left(\frac{v_x * 0 + v_y * 0 + v_z * 1}{\sqrt{v_x^2 + v_y^2 + v_z^2} * \sqrt{0 + 0 + 1^2}}\right) =$$

$$= \arccos\left(\frac{v_z}{\sqrt{v_x^2 + v_y^2 + v_z^2}}\right)$$

The axis around which rotate, $e = [e_x, e_y, e_z]$, was given by the cross product of the two vector:

$$\begin{cases} e_x = (v_y * v_s[2]) + (v_z * v_s[1]) = (v_y * 1) + (v_z * 0) = v_y \\ e_y = -((v_x * v_s[2]) + (v_z * v_s[0])) = -((v_x * 1) + (v_z * 0)) = -v_x \\ ez = (v_x * v_s[1]) + (v_y * v_s[0]) = (v_x * 0) + (v_y * 0) = 0 \end{cases}$$

The quaternion was:

$$q = \cos\frac{\theta}{2} + \left(\frac{e_x}{e} + \frac{e_y}{e} + \frac{e_z}{e}\right)\sin\frac{\theta}{2}$$

where $e = \sqrt{e_x^2 + e_y^2 + e_z^2}$.

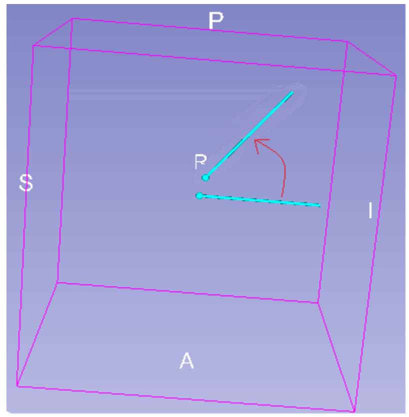


Figure 19: Starting position of the needle in 3D Slicer and relative translation

After that the quaternion has been evaluated, the method "QuaternionToMatrix(quaternion,matrix)" was called. This method, contained in the igtl library, converted the quaternion vector into a 4x4 transformation matrix in order to be sent to 3D Slicer; in particular the first three columns and rows contained the electrode orientation data, the first three rows of the last column contained the electrode tip position and the last row was made of zeros and a final one.

4 RESULTS

The tests made for this project focused on the accuracy and on the precision of the results obtained and on the latency of the data received, processed and sent. There were four main sources of error that could affect the system:

- the markers placement on the electrode and on the stereotactic frame;
- the electrode placement in the electrode arm;
- the electrode angle estimation;
- the electrode tip estimation.

All the tests have been done and repeated 25 times, in order to obtain the best error estimation possible and to evaluate the overall reliability of the system. In the same way, it has been tested the latency of the data received, processed and sent in order to evaluate if the system could be considered as real-time or not. All the results obtained from the tests were satisfactory both on the accuracy side and on the real-time side of the system.

4.1 ACCURACY AND PRECISION

All the source of errors, except for the electrode tip estimation one that intrinsically depended on the others, were manual errors and so they strongly depended on *i)*how the markers were placed, *ii)*how the electrode was placed and *iii)*how the electrode arm angles were set. For this reason, this tests have been repeated 25 times in order to be as much reliable as possible and all the acquisitions have been processed on Matlab, evaluating the mean error and the standard deviation. In particular it has been relevant to observe that there was a big source of error setting the frontal-posterior angle of the electrode arm; this because the device was not well calibrated for that angle and this led to a mean error of 2.06°. This discovery has been very important for the project, because this error is unavoidable also for the surgeon, during the surgery.

ESTIMATION	STD	MEAN ERROR
Marker placement (electrode)	0.44 mm	/
Marker placement (stereotactic)	0.15 mm	/
Electrode placement	0.48 mm	/
Medio-lateral angle	0.2°	0.22°
Frontal-posterior angle	0.52°	2.06°

Table 1: Standard deviation and mean error of the results obtained

Markers placement As it has been explained in chapter 2.3, the markers have been designed and 3D-printed in order to fit unequivocally on the center of the stereotactic frame and on the electrode arm and in order to reduce, as much as possible, the placement error. Although the markers were constantly tracked by the cameras, it was important to introduce less error as possible during the placement, because: *i)*the marker in the center of the stereotactic frame was the reference for the whole system and it needed to correspond to the point in the MRI, *ii)*the markers on electrode were the reference for the evaluation of the tip electrode and the distance, from the upper marker to the tip, required for the estimation was fixed to 16 cm.

The placement error was strictly manual and so the tests have been repeated 25 times following this procedure:

1. Immobilizing the electrode arm and the stereotactic frame;
2. Placing the markers;
3. Acquiring the position from Vicon;
4. Removing the markers.

This procedure has been repeated 25 times for each marker. When all the 25 acquisition have been obtained, they have been processed using Matlab, evaluating the standard deviation of the acquisitions, leading to:

- 0.15 mm, for the marker in the center of the stereotactic frame;
- 0.44 mm, for the markers on the electrode arm.

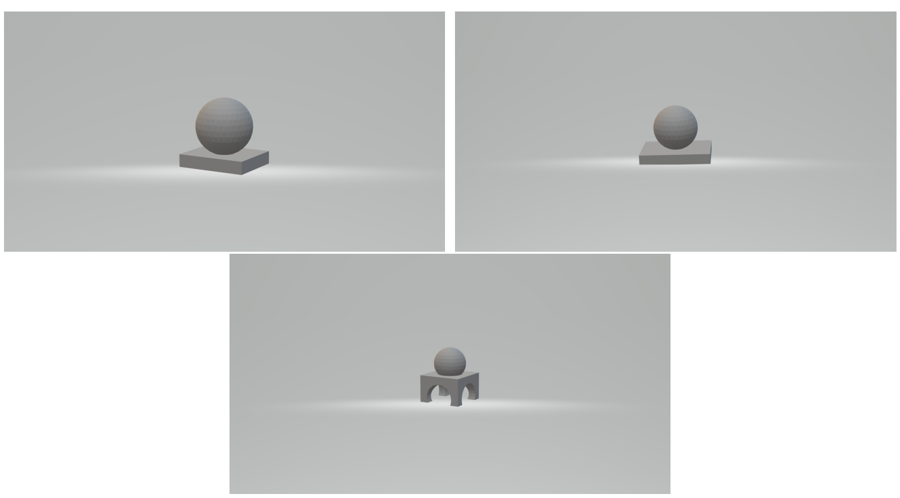


Figure 20: Markers models made with "freeCad" for 3D printing

Electrode placement One of the fixed data for the evaluation of the electrode tip position, was the distance d_1 from the upper marker to the electrode tip (fig. 15). The electrode has been inserted manually in the arm and it has been placed 16 cm far from the upper marker of the arm. So, also this placement introduced a strictly manual source of error and the test has been repeated 25 times following this procedure:

1. Immobilize the electrode arm and the stereotactic frame;

2. Inserting the electrode in the arm;

3. Acquiring the tip position from Vicon;

4. Removing the electrode from the arm.

This procedure has been repeated 25 times. When all the 25 acquisition have been obtained, they have been processed using Matlab, evaluating the standard deviation of the acquisitions, leading to 0.48 mm.

Angle estimation Two of the values that the surgical plan processed, for setting up of the stereotactic frame and of the electrode arm, was the angles of the lesion. In particular, the medio-lateral and the front-posterior angles had to be set up, manually, on the electrode arm and they gave the inclination of the electrode. The medio-lateral represented the angle traversal to the stereotactic frame, instead the front-posterior angle represented the angle parallel to the stereotactic frame. Also in this case, the setting up of this two angles was a strictly manual source of error and in order to estimate it, it was needed to take into account the fact that electrode inclination was also processed by the code.

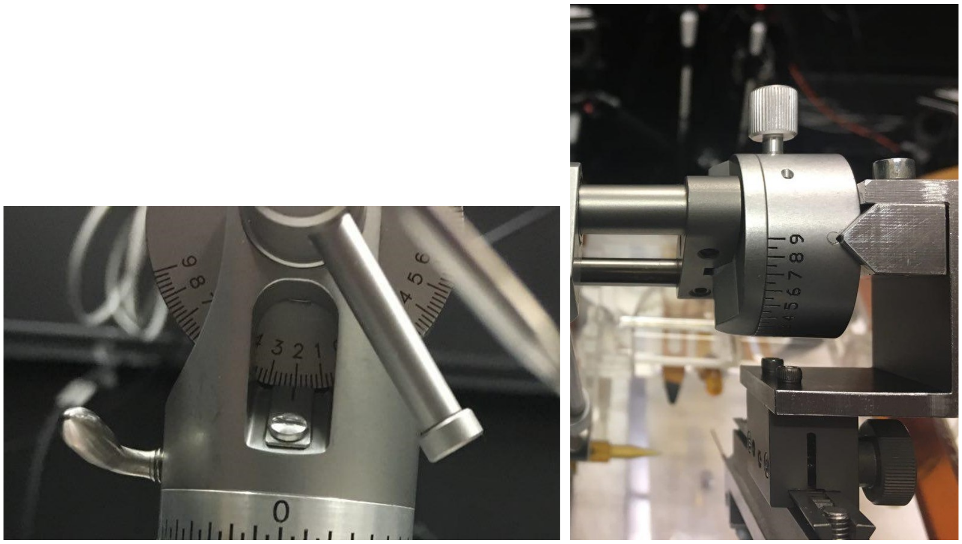


Figure 21: a)Calibration for the medio-lateral angle; b)Calibration for the frontal-posterior angle.

So, the procedure for estimating the medio-lateral angle has been this:

1. Setting to 0° the front-posterior angle;
2. Setting to a known value the medio-lateral angle;
3. Acquiring the angle estimated by the application.

This procedure has been repeated 10 times for each angle and for 10 different angles. Then, the acquisitions have been processed in order to obtain the standard deviation, between the acquisitions, and the mean error, between the known angle and the estimated angle. This led to a STD = 0.2° and to a Mean_Error = 0.22°.

The same procedure has been applied for estimating the frontal-posterior angle:

1. Setting to 0° the medio-lateral angle;
2. Setting to a known value the frontal-posterior angle;
3. Acquiring the angle estimated by the application.

This procedure has been repeated 10 times for each angle and for 10 different angles. Then, the acquisitions have been processed in order to obtain the standard deviation, between the acquisitions, and the mean error, between the known angle and the estimated angle. This led to a STD = 0.52° and to a Mean_Error = 2.06°.

This results have been very important for the project, because the difference between the two mean errors was due to a non-optimal calibration of the frontal-posterior inclination in the device. In fact, from the fig. 21, it's possible to see that the step, between different angles, is of 5° in the frontal-posterior case, instead of 2° in the medio-lateral case, which lead to a better calibration and to a better setting up. This discovery has been very important, because surgeons systematically do this error during the surgery, but, with this tracking system, it's possible to avoid this error.

Electrode tip estimation The error in the electrode tip estimation intrinsically depended on the other source of error, being the results all the manual errors saw previously. The electrode tip estimation depended on the electrode angle estimation, being the tip processed through the angle, and of course it depended on how the markers and the electrode were placed. So the only way to test the electrode tip estimation was visualizing it.

Two tests have been done in order to understand if the estimated tip position was reliable or not and both of them, were based on reproducing the lesion procedure. In the first test, having the setting up made by the surgical plan, the lesion procedure has been applied 10 times saving the final position of the electrode tip that it should had reached, hypothetically, the internal capsule. Knowing that the internal capsule of a monkey is approximately $1\,cm^3$, the saved position have been plotted with Matlab, holding on a sphere of the volume of $1\,cm^3$. In 9 cases over 10 the plotted positions were inside the sphere, and in the case where it was not, it was closed to the border (fig. 22).

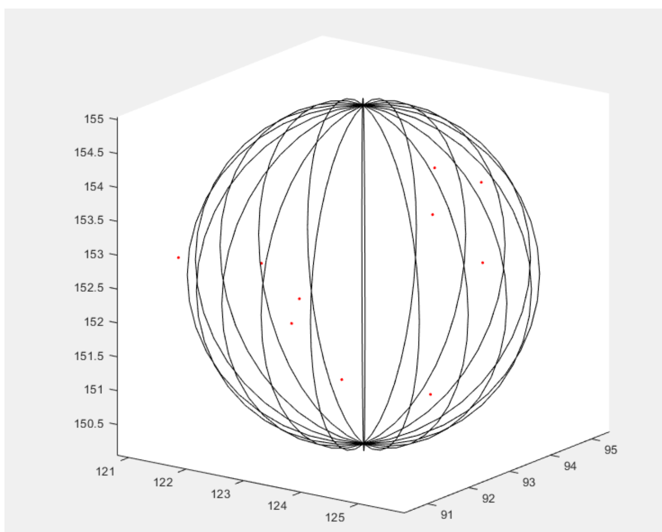


Figure 22: Plotting of the 10 position inside a 1 cm^3 sphere; only one was outside, but closed to the border

The second test has been done reproducing the lesion and keeping track of the electrode during the procedure. Six positions have been saved during the tracking: the starting position at 0 mm, the position at 5 mm in depth, the position at 10 mm in depth, the position at 15 mm in depth, the position at 20 mm in depth and the final position at 23 mm in depth, which correspond to the internal capsule. Then, these positions have been plotted with Matlab on the MRI, in order to see if the internal capsule was correctly reached. In order to do that, a conversion from the 2D format of the MRI (in 3D Slicer) to the DICOM format was required. This because in DICOM the images were acquired in the (i,j,k) image coordinate system, which describes how an image was acquired by the scanner device with respect to the anatomy, instead in 3D Slicer the images were visualized in the RAS anatomical coordinate system (RightAnteriorSuperior), which is a continuous three-dimensional space in which an image has been sampled. In other words, a conversion from the xyz position to the ijk voxel was required in order to visualize the tracked positions of the MRI. The conversion implied the knowledge of the transformation matrix (IJKtoRAS), which is the transformation matrix that also 3D Slicer used to visualize the xyz positions on the MRI in the software. Knowing that 4x4 transformation matrix, M, and the saved xyz position, the formula used to convert was:

$$\begin{pmatrix} i \\ j \\ k \\ 1 \end{pmatrix} = M^{-1} * \begin{pmatrix} x & y & z & 1 \end{pmatrix}'$$

After this conversion the positions have been plotted on the corresponding slice of the MRI. Analyzing the results, it has been possible to see that the internal capsule was correctly reached by the electrode tip and the trajectory, made to reach it, was the same of the one built by the surgical plan. The results, confirming the reaching of the internal capsule and confirming the trajectory, are shown in fig. 23.

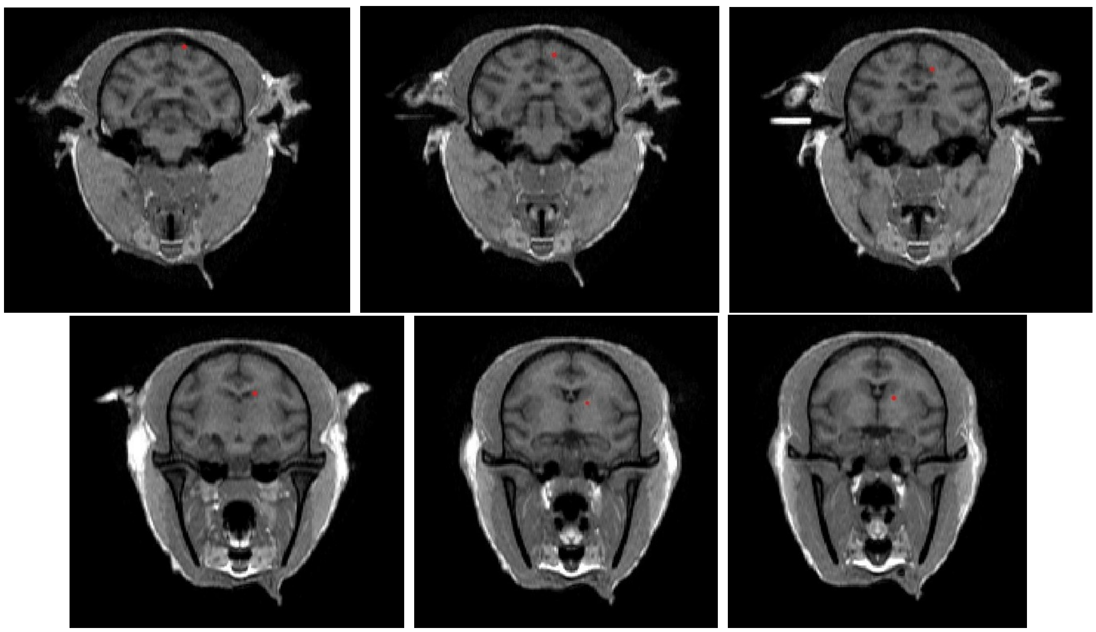


Figure 23: Steps of the simulation of the surgical procedure. The red points represent the position of the electrode at each acquisition and in the last image the internal capsule was correctly reached.

4.2 REAL-TIME ESTIMATION

Real-time systems are defined as those systems in which the correctness of the system depends not only on the logical result of computation, but also on the time to which the results are produced [6]. A real-time processing requires a continual input, constant processing and steady output of data and this system, satisfied all of these requirements. So, a real-time system should have logical and temporal constraints: for this system, the temporal constraints or deadline could be in the order of 50 ms and this led this project to be a “soft real-time system”. A soft real-time system is a system where temporal constraints cannot (sometimes) be met, without any dramatic impact; in fact the surgeon during the surgery could speed up or slow down with the movements, with respect to the rate of the data visualized. However, the overall real-time estimation led to approximately 15 ms of real-time latency.

For the real-time estimation, some timers have been used and put in strategical points of the code. The estimation involved:

- the acquisition of the frame from the Vicon and the extraction of the position from it;
- the processing of the electrode tip position and angle;
- the creation and the sending, to 3D Slicer, of the packet containing the position to be visualized.

The timers have been set up thanks to the “precision_timer” class, which encapsulate the hardware time stamp counter with an intuitive user friendly interface. In particular, in order to estimate the timer of a process, the procedure was this:

1. At the start of the processing: timer.start();
2. Just before the process to be estimated: timer.stop() –> t=timer.elapsed(); –> timer.restart();

3. Just after the process to be estimated: final_t=timer-elapsed()-t;

In this way, the procedure has been used for: *i)*the "GetFrame()" process, which acquired the frame containing the position of the markers, *ii)*the "GetMarkerGlobalTranslation()" process, which extracted the markers positions from the packet, *iii)*for the evaluation of electrode tip process, *iiii)*for the creation and sending to 3D Slicer of the packet containing the electrode position. The results are shown in the table 2 and led to an overall of 14.14 ms.

PROCESS	TIME (ms)
Frame acquisition	9.5
Position extraction	0.04
Position estimation	0.6
Packet sending	4

Table 2: Estimated latency of the four main step of the process

5 CONCLUSIONS

This project offers a real-time guidance during deep brain surgeries in primate, confirming the trajectory of the lesion. A thermocoagulation electrode is used to perform the lesion in this type of surgeries and this system aims to track in real-time the electrode position inside the brain, giving an help to the surgeon. During the surgery, the surgeon is blind and the lesion is based only on a previous surgical plan, which gives the angle depth of the electrode, and on a neuronavigation guidance (stereotactic frame), which is used as reference system. For this reason, combining these tools with a real-time guidance, which plots the electrode position on the MRI of the primate, could improve the precision and the accuracy of the lesion.

An optical motion capture system (Vicon system) has been used in order to track the electrode position, by placing on it two reflective markers. The stereotactic frame, used for the surgery, has been kept as static reference choosing one of its points, which is visible on the MRI, as center of the global coordinate system of the Vicon. In this way the coordinate system of the MRI has been linked with the one of the Vicon, making possible the tracking of the electrode position on the MRI, loaded on the 3D Slicer software. The real-time connection has been guaranteed thanks to the OpenIGTLink extension of 3D Slicer, which supports real-time streaming of transformation matrix containing the electrode position.

The tests confirmed the reaching in real-time of the internal capsule, during a simulation of the surgery, within a mm-precision and an overall real-time latency of 14.14 ms. During the tests, it has been discovered that the setting-up of the frontal-posterior angle, which carried the electrode used for the lesion, led to a mean error of 2.06°, due to a not-well calibrated device. From this, it has been possible to deduce that this error is systematically done also by the surgeon and so, this system could be used also in order to set up all the device needed, before the surgery, obtaining a better calibration of the devices and improving the accuracy.

Using this system in a surgical room, it could be tricky, because the working area needs to be set up in a specific way, placing the cameras in an optimal way in order to track correctly the data. Also the stereotactic frame needs to be as fixed as possible and this requires some accuracy. But, if this system could not be used during the surgery, it is possible to take advantages from it by setting up the device and validating the trajectory of the lesion just before the surgery.

The extension that guaranteed the real-time streaming of data for the visualization, OpenIGTLink, offers a dedicated and practical interface to some compatible tracking devices. Vicon, actually, is not one of them and so, in future, a more feasible and user-friendly interface could be integrated to the one used for this system.

References

[1] W. D. Bidgood and S. C. Horii. Introduction to the acr-nema dicom standard. *RadioGraphics*, 12(2):345–355, 1992. PMID: 1561424.

[2] T. Halic, S. Kockara, C. Bayrak, and R. Rowe. Mixed reality simulation of rasping procedure in artificial cervical disc replacement (acdr) surgery. In *BMC bioinformatics*, volume 11, page S11. BioMed Central, 2010.

[3] R. Kikinis, S. D. Pieper, and K. G. Vosburgh. 3d slicer: a platform for subject-specific image analysis, visualization, and clinical support. In *Intraoperative imaging and image-guided therapy*, pages 277–289. Springer, 2014.

[4] P. Merriaux, Y. Dupuis, R. Boutteau, P. Vasseur, and X. Savatier. A study of vicon system positioning performance. *Sensors*, 17(7):1591, 2017.

[5] R. J. Morecraft, J. L. Herrick, K. S. Stilwell-Morecraft, J. L. Louie, C. M. Schroeder, J. G. Ottenbacher, and M. W. Schoolfield. Localization of arm representation in the corona radiata and internal capsule in the non-human primate. *Brain*, 125(1):176–198, 2002.

[6] J. A. Stankovic. Misconceptions about real-time computing: A serious problem for next-generation systems. *Computer*, 21(10):10–19, 1988.

[7] J. Tokuda, G. Fischer, X. Papademetris, Z. Yaniv, L. Ibanez, P. Cheng, H. Liu, J. Blevins, J. Arata, A. Golby, T. Kapur, S. Pieper, E. Burdette, G. Fichtinger, C. Tempany, and N. Hata. Openigtlink: An open network protocol for image-guided therapy environment. *Int J Med Robot*, 5(4):423–434, 12 2009.

[8] I. Vicon Motion Systems. Vicon motion systems. http://www.vicon.com.