FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

ASSIGNMENT OF MASTER’S THESIS

Title:	Software for Stereotactic Navigation System in Epileptosurgery
Student:	Bc. Tomáš Michalík
Supervisor:	Ing. Petr Ježdík, Ph.D.
Study Programme:	Informatics
Study Branch:	Web and Software Engineering
Department:	Department of Software Engineering
Validity:	Until the end of winter semester 2020/21

Instructions

1. Study the flow of stereotactic planning for invasive electroencephalography, get familiar with possible risks with veins penetration as well as brain eloquent regions.
2. Design and implement the plug-in module in 3D Slicer platform for planning the invasive monitoring of patients with refractory epilepsy, describe your implementation.
The plugin should be able to import standard 3Tesla MRI T1, T2 or Flair modalities to represent the electrode planning results as well as binary or intensity coding of restricted areas. The final electrodes positioning also should be possible to export in human readable spreadsheets.

References

Will be provided by the supervisor.

Ing. Michal Valenta, Ph.D.
Head of Department

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
Dean

Prague February 28, 2019

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Master’s thesis

Software for Stereotactic Navigation System in Epileptosurgery

Bc. Tomáš Michalík

Department of Software Engineering
Supervisor: Ing. Petr Ježdík, Ph.D.

May 9, 2019

Acknowledgements

I would like to thank my supervisor Ing. Petr Ježdík, Ph.D. for guidance in this project

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46(6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the “Work”), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work in any way (including for-profit purposes) that does not detract from its value. This authorization is not limited in terms of time, location and quantity.

In Prague on May 9, 2019 ....

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Michalík, Tomáš. *Software for Stereotactic Navigation System in Epileptosurgery*. Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2019.

Abstrakt

Tato práce je zaměřena na návrh a implementaci poloautomatického nástroje pro plánování implantace hloubkových elektrod v oboru epileptochirurgie. Navržené řešení je implementováno jako rozšíření do platformy 3D Slicer. Volumetrická data znázorňující omezující podmínky a seznam cílů navigace jsou použity jako vstupy pro automatické zpracování. Systém navrhuje seznam kandidátů na řešení, která mohou být interaktivně ohodnoceny. Toto je první iterace implementace, která obsahuje základní funkcionalitu zahrnující dávkové zpracování vstupních dat, automatické ohodnocení navigační trajektorie a uživatelské rozhraní pro interaktivní ohodnocení navrhovaných řešení.

Klíčová slova Počítačem podporované plánování, epilepsie, obrazem řízená neurochirurgie, volumetrická data

Abstract

The thesis is focused on the design and implementation of semi-automatic navigation planner for implantation of deep brain electrodes in epileptosurgery. The solution is implemented as an extension into 3D Slicer platform. It uses volumetric data model of restricted areas and list of navigation targets as inputs to automatic processing. The system suggests the list of solution candidates which can be interactively evaluated. This is the first iteration of the implementation which incorporates basic functionality including batch processing of the input data, automatic evaluation of navigation paths and user interface for interactive evaluation.

Keywords Computer-assisted planning, Epilepsy, Image-guided neurosurgery, volumetric data

Contents

List of Figures

Introduction

The goal of the thesis is to design and implement semi-automatic navigation planner for invasive deep brain electrodes implantation in epileptosurgery. This tool is implemented as an extension to the 3D Slicer platform.

Epilepsy is the most common group of neurological diseases. Despite great advancement treating epilepsy with specialized medication, about one-third of patients with focal epilepsy are pharmacoresistant. These patients can be potentially treated by methods of epileptosurgery which aims to remove or disconnect epileptogenic zone that is responsible for epileptic seizures. In some cases, it is required to precisely analyse epileptogenic zone by invasive EEG electrodes. The planning of this surgery may be supported by an automatic or a semi-automatic tool for finding optimal navigation paths.

The assisted solution may improve the planning process by lowering risks of an incorrect path design, shorten the planning length and minimizing processing errors caused by a human factor.

In the first chapter, there is provided an overview of invasive EEG surgery planning. The second chapter presents the goals of this thesis. In the third chapter, there are described current approaches of surgery planning in epileptosurgery. And finally, the fourth and the fifth chapters are devoted to the analysis, the design and realisation of the semi-automatic navigation planner.

Chapter 1

Problem description

1.1 Context overview

1.1.1 Epilepsy

Epilepsy is the most common group of neurological diseases. It is characterized by epileptic seizures which may vary from brief and nearly unnotable to long periods of vigorous shaking [7]. It is estimated there are around 100 000 people affected in the Czech Republic [8].

In most cases is epilepsy successfully treated with special medication. If this fails patients with some types of epilepsy may be treated with a surgical intervention (around 25-30% of all patients affected by epilepsy are pharmacoresistant [8]). And around 5% of patients from this group may benefit from surgical intervention.

1.1.2 Epileptosurgery

Epileptosurgery is a neurology discipline which aims to eliminate epilepsy genesis by performing a resection of a tissue which is responsible for epileptic seizures. In uncomplicated situations seizures onset zone can be clearly located by evaluating medical screening. The most important for analysing epilepsy are MRI and EEG [8].

This analysis requires additional steps in cases when the onset zone is not clear, has complex boundaries or interferes with critical brain centers. In these situations, invasive EEG may be used to gather more data about location and characteristic of the onset zone.

1.1.3 Electroencephalography

Electroencephalography (EEG) is a method which records a electrical brain activity. The most basic EEG is measured with electrodes placed along the scalp. This type of measurement has its limitations because of the distance

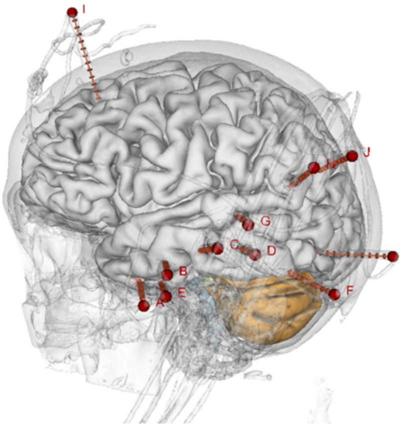


Figure 1.1: Example of SEEG in non-lesional temporal epilepsy [1]

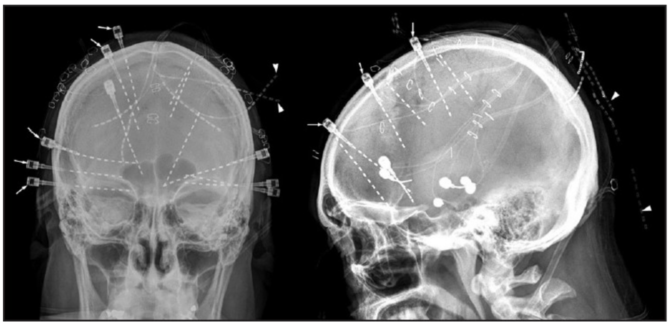


Figure 1.2: A post-implant skull X-ray showing multiple depth electrodes [2]

from actual brain tissue and it is not possible to measure activity of all areas of the brain cortex. Invasive EEG can be used to overcome these limitations and analyze particular brain regions. There are two types of an invasive EEG. The first approach uses electrodes formed to stripes or grids and they are placed directly on the surface of the brain cortex. The second approach uses intercerebral electrodes implanted using stereotactic surgical intervention, called stereoelectroencephalography (SEEG). This type enables analysis deep brain structures and collects data directly from the epileptogenic zone. Examples of SEEG electrodes implantation can be seen in Figure 1.1 and Figure 1.2.

1.2 Clinical view

In this thesis we are focusing on the stereotactic surgery planning aimed to implant invasive EEG electrodes to broaden analysis of a brain dysfunction

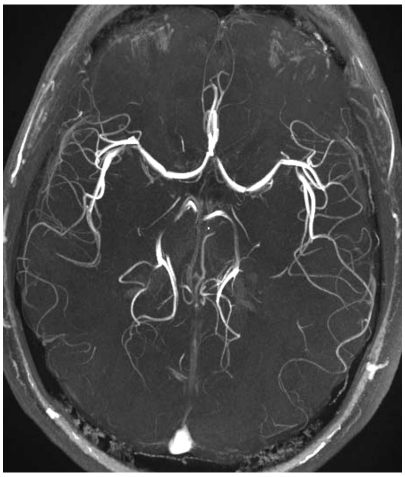


Figure 1.3: Magnetic resonance angiography [3]

causing epileptic seizures. This procedure is a standalone surgical operation which aims to position EEG electrodes at locations which cannot be examined by using scalp EEG.

During the planning process, a range of aspects must be evaluated in order to lead to successful surgery. The most important requirement for a designed solution is to avoid all critical functional structures (e.g. blood vessels, lymph vessels). In general, it must disqualify certain areas which should be avoided because of high risk of damaging important functional centers. The key aspect of the planning is a human factor, a domain expert, who is responsible for combining general knowledge of brain anatomy with analysis of concrete brain imagines.

1.3 Technical view

From the technical perspective we are considering a range of imaging methods which give us a partial insight about the structure and the function of the brain. We need to consider limitations of every method, e.g. resolution and contrast, and combine these data to decide optimal path for the stereotactic intervention and minimize potential risks.

The actual implantation of an EEG electrode consists of the automated drilling of the entry on the skull surface and then manual insertion of the electrode. This procedure constitutes another restriction which is the limited entry angle.

The challenging part of the planning lies in the combination of huge amount of possible solution paths with human expertise which eliminates them based on empirical knowledge of brain functions and comparison with similar past surgical procedures. These requirements guide us to design a semi-automatic tool for assisted planning which combines both automatic image processing and manual human evaluation of suggested solution candidates.

1.4 Target users

A request for designing an alternative tool for the assisted stereotactic planing in epileptosurgery comes from Intracranial Signal Analysis Research Group (ISARG) [9]. It is a research group which consists of medical doctors from the Charles University in Prague and technically educated researchers from the Czech Technical University in Prague. The main focus of the group lies in the analysis of EEG signals of epileptic patients aimed to improve diagnostics and broaden the understanding of seizures genesis.

A close cooperation of medical and technical experts creates an opportunity to combine domain expertise and state-of-the-art algorithms and procedures from technical fields of signal and image processing.

Currently, the stereotactic navigation planning for implanting EEG electrodes at University Hospital Motol, Prague is done fully manually (see Chapter 3). This planning process can be potentially improved by introducing a tool for assisted planning of this specific surgery.

Chapter 2

Goals

The goal of this thesis is to design and implement a first iteration of a semi-automatic navigation planner for implanting invasive EEG electrodes using stereotactic surgical intervention. It should offer an assisted planning utilizing image processing of input volumetric data and become an alternative to a fully manual planning of this surgery.

A designed solution should be implemented as an extension into the open source platform 3D Slicer [10], take advantages of libraries used by this platform and create an open design of the extension to support interaction with built-in features of 3D Slicer. The solution should design a processing flow, an algorithmic solution and user interface for interaction with the tool. A designed solution of this extension should be focused on usage by target users - medical doctors and technical researchers studying epilepsy.

This work is focused on the design of a processing pipeline - including input data filtering, finding the optimal trajectories based on an input volumetric mask, presenting suggested data to the user and offer controllers to evaluate and manipulate suggested solutions. This work does not focus on data preparation, nor brain structures detection.

Chapter **3**

Current state

3.1 Current solutions for assisted stereotactic navigation planning in epileptosurgery

In this section is provided an overview of solutions dealing with automatic or assisted deep brain electrodes implantation planning.

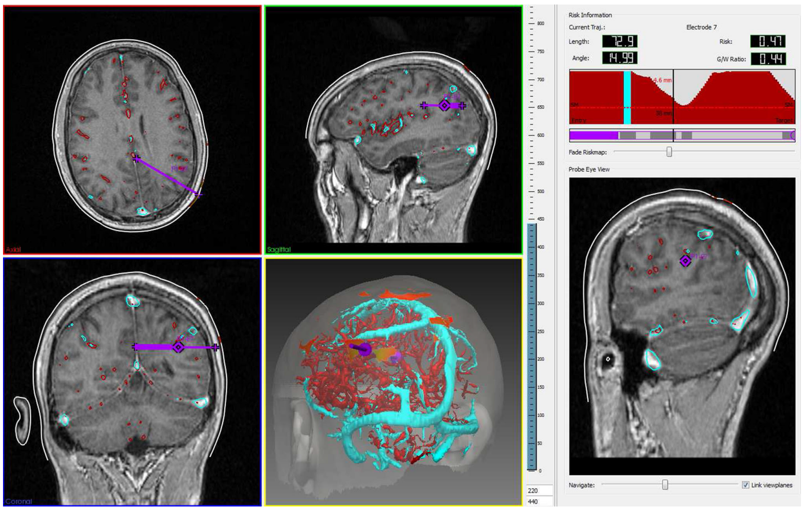
3.1.1 Enhanced data visualizations supporting manual planning

As described in [11], the assistance to a domain expert planning surgery may be provided by clear visualization of blood vessels which have to be avoided by navigation paths. In this article is presented the approach of constructing polygonal mesh of vascular tree based on multiple modalities. Compared to a single modality segmentation, the combination of data from different imaging techniques lead to an improvement of this segmentation thus it can lead to improvement of navigation paths planning.

3.1.2 Assisted single navigation path planning

The article [12] describes assisted operation planning of deep brain stimulation electrodes implantation. Compared to SEEG surgery, this surgery is done with a different goal but it shares similar requirements. It uses linear navigation paths and it has to avoid critical brain structures. In the mentioned article, there are considered two types of constrains which are incorporated in the automatic navigation paths planning. The first ones are rules describing strict constrains (e.g. the solution path has to avoid vessels) which has to fulfill. The second set of rules are soft constrains and they describe parameters which have to be optimized (e.g. maximize the distance from critical functional centers).

The automatic planning algorithm uses set of strict and soft rules plus segmented structures in a form of polygonal meshes. It utilizes numeric methods



Risk Information

Current Traj.: Electrode 7

Length: 72.9

Risk: 0.47

Angle: 14.99

G/W Ratio: 0.44

Figure 3.1: EpiNav$^{\mathrm{TM}}$ software [4]

for searching optimal paths by maximizing distance alongside navigation path and risky structures.

In this article is also discussed the decision of using polygonal meshes of segmented data compared to the use of volumetric data. The authors highlight advantages of using polygonal surface data because of its easy visual evaluation and also its possible usage in neuronavigation for augmented reality.

3.1.3 Assisted multi navigation paths planning

The article [4] presents automatic planning algorithm for searching possible navigation paths of SEEG electrodes implantation with consideration of multiple targets. This addition ensures that proposed navigation paths are not colliding. The solution is implemented in EpiNav$^{\mathrm{TM}}$ software, the user interface can be seen in Figure 3.1.

3.2 Stereotactic navigation planning software

In this section, we provide an overview of currently used software used for stereotactic navigation planning. Note that descriptions of the following software highlight only subset of their features relevant in this context. Generally, it is software fulfilling requirements to be used for planning stereotactitc surgeries.

3.2.1 StealthStation^{TM} S8

The StealthStation^{TM} S8 and StealthViz^{TM} are all-in-one surgical navigation and planning software solutions developed by Medtronic [13]. They support advanced visualizations, review and merge scans, trajectories planning and DTI tractography.

3.2.2 neuroinspire^{TM}

The neuroinspire^{TM}, developed by Renishaw, is surgical planning software specifically targeted on a planning of stereotactic procedures [14]. It focuses on targets and trajectories planning, defining safety corridors alongside trajectories and delineating anatomical features.

3.2.3 Brainlab Elements

Brainlab Elements, developed by Brainlab, is planning software which includes features such as segmentation, fiber tracking, and it uses automatic patient-specific anatomical mapping to detect abnormalities [15].

3.2.4 OsiriX

OsiriX is an open source DICOM viewer which can also be used for surgery planning [16] [17]. It is mainly focused on various data visualization and it is limited to Mac OS platform.

3.2.5 Other software

Other representatives of stereotactic navigation planning systems are Leksell SurgiPlan® [18], inomed planning software - iPS [19], iPlan® Stereotactic Planning Software [20] and MNPS [21].

3.3 Fully manual planning

Surgical planning can be done fully manually, as it is used at University Hospital Motol, Prague. This approach uses some software only as a platform which supports basic tasks - e.g. displaying volumetric data and placing markers. The planning itself is done by domain specialist and relies solely on his expertise.

The process consists of the following steps:

- Evaluating relevant image data, such as contrast enhanced MRI
- Selecting target and source markers locations

- Evaluating selected solution path in 2D view slice by slice and validating the intersection of designed path with brain structures
- In case that selected solution interferes with blood vessels or other structures which have to be avoided - solution or target coordinates have to be customized and the whole solution path has to be reevaluated

3.4 Discussion

We can divide mentioned software into proprietary and open source. Proprietary software for surgical planning is often offered as a complete package including the planning software, planning workstation and surgery navigation system. On one hand, the advantages of this software is that it is maintained by companies providing professional support, they ensure compatibility with navigation tools and provide robust environment for using in predefined scenarios. On the other hand, they are typically very closed systems not possible to customize or extend for custom tasks.

On the other side of the spectrum lies open source software, e.g. OsiriX, which typically does not offer such robust connection to navigation and surgery devices but their open design allows them to be customized and used as a platform for implementing novel techniques and procedures.

Chapter 4

Analysis and design

4.1 Platform

Although the platform is set by the thesis' task, in this section presents a brief discussion of possible alternatives.

The previous chapter explored current solutions used in the field of surgery planning. These software tools supporting key tasks in a surgery are, as the nature of the medical field requires, robust solutions designed for very specific tasks. On the other side, they typically do not support any form of extension or customization. This rigid design disqualifies them from being used in research environments where they might be the point of interest a testing of novel techniques and algorithms.

Other option is implementing a new tool from scratch. This enables us to design optimal software architecture for our task, use up-to-date software frameworks and programming languages and have a full control over data processing and the system itself. On the other side, initial requirements for such a system might be quite exhausting.

A brief overview of necessary components:

- 2D and 3D (and 4D) medical data rendering
  - Polygonal models
  - Volumetric data sets
- Support of a reasonably wide spectrum of input file formats (e.g. DICOM, NIFTI)
- Data exporting
- Basic tools for data manipulation

The last option is to use some existing open source platform and implement a custom extension. If this software is used by a target community, the chances that the new extension will be accepted by medical experts are much higher then with a completely new tool.

3D Slicer is an open source platform which is widely used by medical and research community. It includes wide range of modules supporting various features for data processing and visualizations, which makes it a good candidate for using in this use case.

4.2 Domain model

In this section, objects from the target domain, which are considered during the design process, are described.

- Medical screenings
- Restricted areas data model
- Navigation target
- Navigation source
- Navigation solution path

Medical screenings

There are various types of possible medical screening, in this context we are focusing on any arbitrary data which can be represented visually in 2D or 3D space. In general, there are two categories of neuroimaging - structural and functional imaging. The data can be in a form of volumetric data sets or polygonal models.

The data are used by a domain expert during the navigation planning process to decide which solution paths are better then the others.

Restricted areas data model

The main input for assisted navigation planning is a set of data which represents restricted areas that have to be avoided. There are two approaches - the first one considers two values restricted and non-restricted. This model can be saved as a polygonal model which follows the boundaries between those areas, or a volumetric model containing only two different values. The second approach considers a probabilistic model where every point can have a value which represents the probability of how much should be avoided. This information can be stored as a volumetric data set.

Navigation target

Navigation target is a target point or a target area which should be reached during the surgery. The target is located by its RAS coordinates, global coordinate system of 3D Slicer.

This is the input to the assisted navigation planning but if necessary it can be relocated manually during the manual phase of the planning process.

Navigation source

A navigation source is a source point which determines (together with navigation target) the vector of the solution. It is defined in RAS coordinates.

Navigation solution path

A navigation solution path (sometimes referred to as the solution or navigation path) is defined by the target and the source points in RAS coordinates and represents a single plan for stereotactic navigation.

4.3 Requirements

Functional requirements

- Automatic generation of navigation solutions
- Evaluate a solution candidate
- Cluster similar solution candidates
- Solution candidate customization
- Solution candidate validation
- Selection of navigation solution from solution candidates
- Results export

Non-functional requirements

- Plug-in module in 3D Slicer platform
- 3Tesla MRI T1, T2 and Flair input volumes
- Binary or intensity coding of restricted areas
- Volumetric representation of restricted areas

4.3.1 Requirements detailed description

Automatic generation of navigation solutions

The system generates all possible navigation solution paths for every navigation target. A generated solution is described as solution candidate.

Evaluate a solution candidate

The system evaluates navigation path using input mask (restricted areas data model) and assign relevant score which can be used for comparing independent solutions.

Cluster similar solution candidates

Use some clustering algorithm to reduce count of suggested solutions and select relevant representative of a group of similar solutions.

Solution candidate customization

It should be possible to customize the suggested solution manually and recalculate solution score.

Solution candidate validation

The system should provide UI widget for navigation solution validation. The automatic part of the processing suggests list of solution candidates which are presented to the user. After this, the domain expert browse all solutions, validate solution candidates one by one and mark them as rejected or accepted.

Selection of navigation solution from solution candidates

The manual process of solution candidates evaluation ends with selection of the best solution from the list of accepted solution candidates.

Results export

It should be possible to export the final electrode positioning into human readable spreadsheets.

Plug-in module in 3D Slicer platform

The solution should be implemented as a plug-in module in 3D Slicer platform [10]. It should utilize the open architecture of 3D Slicer and allow the user to interact with the data from other built-in or external extensions.

3Tesla MRI T1, T2 and Flair input volumes

The system should support input volumetric data from 3Tesla MRI scans including T1, T2 and Flair sequence types.

Binary or intensity coding of restricted areas

The system should support binary or intensity coding of restricted area.

Volumetric representation of restricted areas

Use volumetric representation of restricted areas. Compared to the other option, the polygonal models, this approach does not rely on a segmentation algorithm which may introduce an error into the planning process.

4.4 Use case

The following section is dedicated to the description of the basic use of the designed extension. This simple usage may be extended by incorporating 3D Slicer built-in or external modules for data modification, visualizations, etc.

4.4.1 UC1 Create stereotactic navigation plan

1. Open the application (3D Slicer)
2. Load restricted area data model
3. Load additional medical screenings
4. Use build-in module Markups to define list of navigation targets
5. Switch to NavigationPlanner module (implemented module)
6. Select inputs - restricted area data model, list of navigation targets
7. Run the processing
8. Browse targets list one by one
    a) Browse list of suggested solution candidates one by one
        i. Browse a single solution candidate navigation path in 2D views to check the validity of suggested path
        ii. (optional) If suggested solution is not acceptable customize solution by moving navigation target or navigation source
        iii. Mark solution candidate as rejected or accepted
    b) Select the best solution candidate
9. Export the data

4.5 Architecture model

4.5.1 3D Slicer architecture

3D Slicer [10] is a free open source cross-platform application for medical image computing. A summary of this article [5] is provided in the following

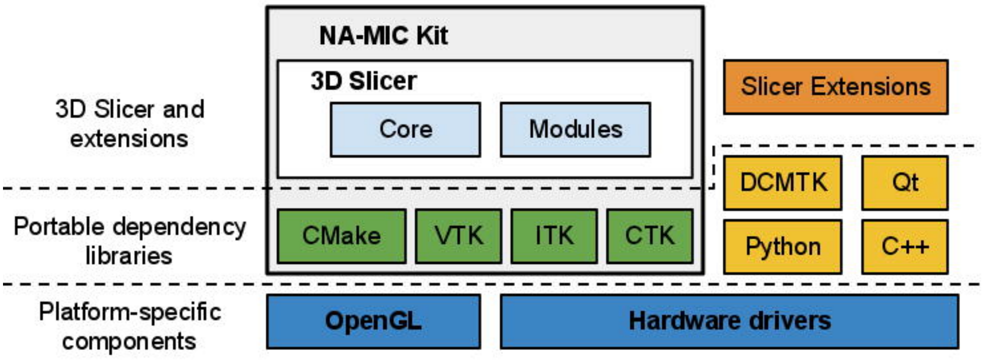


Figure 4.1: 3D Slicer architecture [5]

paragraphs. 3D Slicer is a layered and modular software (see Figure 4.1) which takes advantages of existing libraries and encapsulate them into a single package. The application consists of core and built-in modules plus it can be extended by developing external extensions.

4.5.2 3D Slicer plug-in module types

There are two distinct ways how 3D Slicer can be extended - Slicer Execution Model (SEM) modules and loadable modules.

The first option allows developers to implement plug-ins which do not rely on 3D Slicer architecture and do not require any dependencies to Slicer core nor any linked libraries. These modules can use MVC architecture design while model and view components are managed by main application via SEM Module Manager (SEM-MM) (see Figure 4.2). Input and output data are specified using standardized XML schema which is also used to generate GUI widgets that controls input and output parameters. These modules can be implemented in any arbitrary language or platform as executable files, shared libraries or scripts. This approach suits for developing features which do not require user interaction or dynamic visualization, e.g. image resampling or filtering.

The second option, loadable modules, are closely tied to the 3D Slicer application and have direct access to its core features. This is suitable for interactive tools and it typically uses ITK [22] or VTK [23] libraries for constructing pipelines and lower level processing. They are suitable for interactive applications and have full control over communication with other parts of 3D Slicer. Loadable modules can be implemented in C++ or Python.

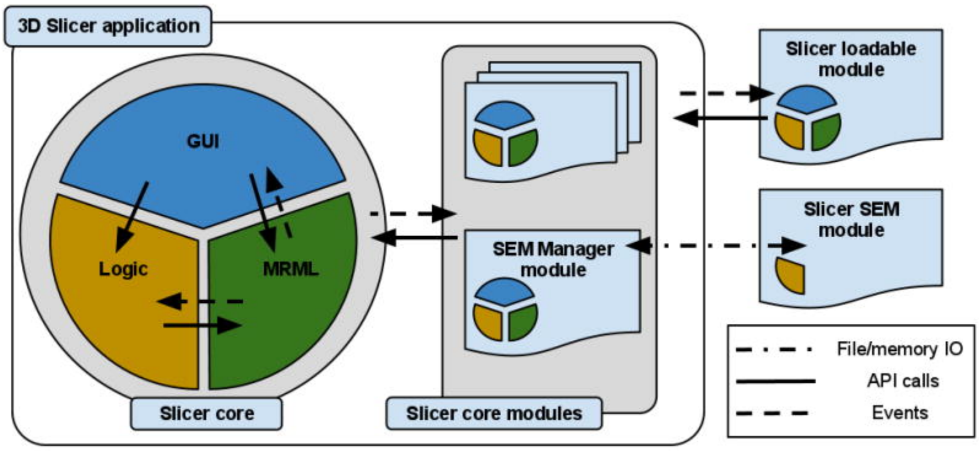


Figure 4.2: 3D Slicer external modules integration [5]

4.5.3 Solution design

The designed extension consists of the processing part and and the user interface for presenting calculated data and allowing suggested solutions to be explored and validated with dynamic visualization in 2D and 3D views.

A code of the extension is split into two modules. The first one is a Python loadable module which is responsible for creation and control of the user interface (referred as the main module). It does not require compilation and it is the most flexible way to extend 3D Slicer from the development perspective. It can be used for fast prototyping in combination with Python built-in command line interpreter which can take all advantages of run-time API access to linked libraries ,e.g. VTK [23], Qt [24], SimpleITK [25], which are fully wrapped to be used from Python environment.

The second module of this extension is implemented as a C++ CLI module using Slicer Execution Model which is suitable for low-level high performance processing. This module is responsible for batch processing of input data and returns list of solution candidates. This type of module is limited only to basic input-output types but for this use case it is not restrictive. Inputs are sent as command line parameters, input volume mask as a file, and results are returned as a serialized text via text file. It is used only for one-time batch processing thus communication overhead caused by using CLI interface is acceptable in this case.

This distribution of extension code should support easy prototyping of future changes in the loadable module and simple maintainability by technical researchers. The code requiring higher performance is separated and can be used as a separate tool out of 3D Slicer application.

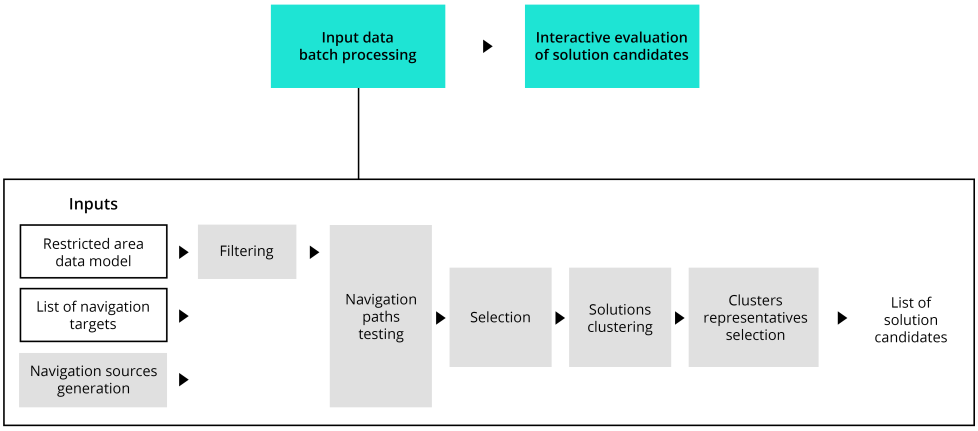

Input data batch processing
Interactive evaluation of solution candidates
Inputs
Restricted area data model
List of navigation targets
Navigation sources generation
Filtering
Navigation paths testing
Selection
Solutions clustering
Clusters representatives selection
List of solution candidates

Figure 4.3: Data processing pipeline

4.6 Processing pipeline

In the following section are discussed possible solutions of implementing separate steps of the processing pipeline. The whole processing design can be seen in Figure 4.3. The first part contains the automated batch processing of input data and the other part consists of an interactive evaluation of solution candidates.

4.6.1 Input data

Restricted area data model

This model includes information about which parts of the brain have to be avoided by navigation paths. This data can represented as a polygonal model or as a volumetric data set. In this phase of development we do not set any requirements on the input data nor we expect any specific content to be provided. It can be used with various data e.g. angiography - medical imaging of blood vessels, or coded regions which have to be avoided.

The representation in the form of a polygonal model is able to code binary information and distinguish between safe and restricted areas. This approach can be useful for using with automatic segmentation of brain regions to mark areas of critical brain centers. On the other hand, we have to rely on the segmentation algorithm and there is no way to model the uncertain boundary between safe and restricted regions which may be unclear.

The second form, representation as a volumetric dataset, overcomes the limitation of binary information and allows us to code probabilistic model into the data. Lower values indicate safer regions while higher values indicate more likely restricted area. This approach is more flexible and can be used without any intermediate steps with inherently volumetric data, e.g. MRI.

Navigation sources generation

In general, our goal is to generate all possible navigation paths for further evaluation. We recognize three basic approaches to the navigation sources generation.

The first way is compatible with restricted area data model represented by volumetric data set. It uses the resolution of the volumetric data and it generates source points at a position of every voxel which lies on the "surface" of the volume. By surface we mean first and last row of voxels alongside all sides of the volume box. Using this approach we get the best possible resolution of navigation paths which make sense to be tested in the context of input data resolution. The drawback of this method is the cumulation of generated points in the corners of the volume box so the source points are not generated uniformly.

The second way uses sphere to uniformly generate points of it's surface. This can be implemented using a gaussian random number generator to independently generate three coordinates which are then normalized to unit sized vector. This approach can be used to generate points online thus it may limit memory requirements during the processing. On the other hand, we have to decide how many points we need to have generated based on the resolution of the input data. It is possible to use the count of generated points from previous approach as a reference number for this issue.

The last way requires using additional modality - a model of a head or a skull surface. This symbolizes a surface where we want to generate navigation source points. For this task An Intrinsic Algorithm for Parallel Poisson Disk Sampling on Arbitrary Surfaces can be used [26]. This approach eliminates the length of the navigation path which needs to be tested.

4.6.2 Processing

Filtering

Filtering can be incorporated in the processing pipeline if we consider that the restricted area data model is represented by volumetric data set. For this use case, it can be beneficial to use filters for edge detection and data pre-processing to simulate implanted EEG electrode thickness.

Edge detection can be implemented by modification of the original input data - replacing original voxel values with their fist or second derivation. This can be useful, for example, for an input data containing angiography to strengthen processing sensitivity to smaller vessels.

Other input filtering may be done to simulate a required thickness of EEG electrode or a safety region around navigation path. This can be implemented by replacing the original voxel value with aggregated value representing its neighbourhood, e.g. maximal value of neighbouring voxels.

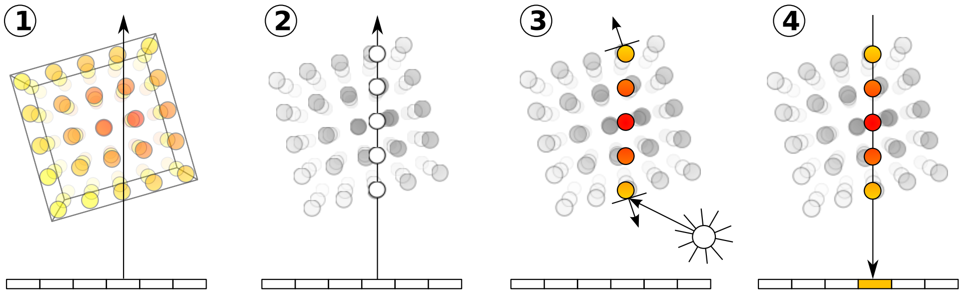


Figure 4.4: Volume ray casting [6]

Navigation paths testing

The principal part of the processing is the batch testing of generated navigation paths.

If the polygonal model is used as a restricted area data model we have to test if there are any intersections with the model between the source and the target navigation points.

For volumetric data, this becomes a relatively performance exhausting task. We are facing an opposite challenge compared to volumetric data rendering via ray casting algorithm (sometimes called volume ray tracing or volume ray marching), see Figure 4.4, we just do not need to incorporate any lighting model. In this situation we need to cast rays from a navigation source to a navigation target and accumulate a value representing navigation path quality.

The design of an accumulation function requires further discussion. The basic approach can calculate the resulting value as a maximum or a minimum of samples alongside the navigation path. In a data dependent scenarios, we can consider more specific calculation of this function. Let's assume the input data contains a detailed angiography and we may ask the following questions. Do we want to put more focus on larger vessels? Is crossing two smaller vessels better then crossing a single larger vessel? These questions are out of scope of this thesis and require additional analysis.

Selection

This is a simple step to eliminate the count of suggested solutions. It can be done by skipping solutions which have a calculated score / penalty mark beyond some threshold value. Or, the selection can be made to return fixed count of solutions.

Solution clustering

To reduce total count of suggested solutions it is necessary to include some kind of clustering. For this task we consider two basic classification algorithms k-means and mean shift.

K-means is a classification algorithm which classifies n observations into k clusters based on its distances [27]. In this situation the possibility to set the count of needed results may be handy, on the other hand we cannot estimate relevant count of output clusters since we do not know anything about the input data beforehand.

Mean shift algorithm frees us from deciding how many results do we expects [28]. Its advantages are that is does not assume any shape of clusters and internal parameter bandwidth has a physical meaning.

Clusters representatives selection

In the previous step, there are the navigation paths are split into disjunctive clusters. In this step it is selected how each individual cluster is represented. It can be done by selecting the best rated navigation path, or we can use a mean of all source positions' coordinates to construct a new solution representing the cluster.

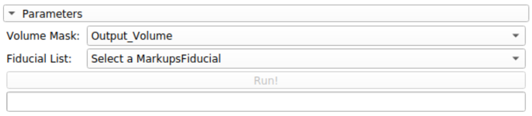
4.6.3 Interactive evaluation of solution candidates

In the previous steps, the input data are processed and solution candidates are then suggested as possible navigation paths. After this step, the responsibility passes to the user who should evaluate all navigation paths and decide the best navigation path for every navigation target. This should be main part where are combined human domain expertise and a support from automated processing system. The goal of this interaction is a selection of the best navigation paths for every target.

The solution candidate can be marked as accepted or rejected. Other option should be the possibility to modify the suggested path by changing position of the navigation target or the navigation source. This process should be interactive and indicate the quality of newly introduced navigation path to the user. Evaluation of the navigation path is mainly done in 2D views and any modality which is relevant to the evaluation can be used as a background (possible different modality to input of automatic processing) .

4.7 User interface design

User interface is designed to fit into 3D Slicer platform. The UI of designed extension consists of three parts - input parameters widget, results widget, and



Volume Mask: Output_Volume

Fiducial List: Select a MarkupsFiducial

Figure 4.5: Input parameters widget

standard 2D and 3D view of 3D Slicer. The complete view of the Navigation Planner extension displaying results can be seen in Figure 4.8.

4.7.1 Input parameters widget

The first part is very basic and contains input fields for input parameters setup. This view can be seen in the Figure 4.5.

4.7.2 Results widget

Once the results are calculated they are presented to the user, as in Figure 4.6. The view consists of three regions. The first one is for switching navigation targets. Then, the results table shows the list suggested solution candidates. And, at the very bottom, there are widgets for controlling validation of solution candidates.

4.7.3 2D and 3D view

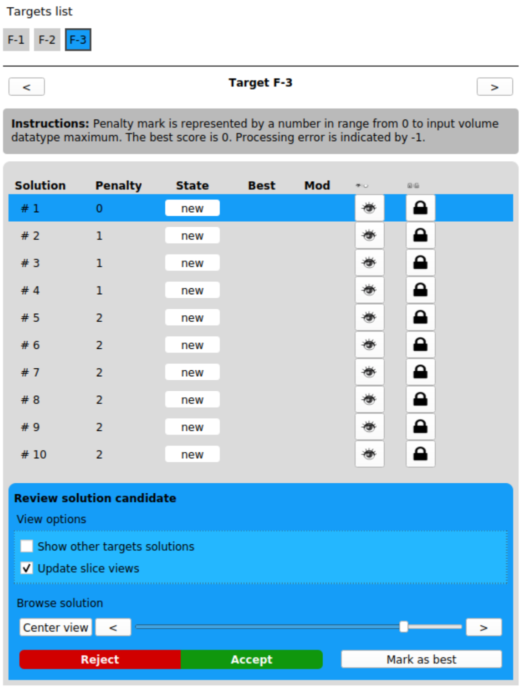
The last part of UI is displayed in the standard 2D and 3D views of the 3D Slicer, see Figure 4.7. In these views, every solution candidate is represented by the source and the target markup fiducial and line model which indicates the solution path.

4.8 Class diagram

Navigation Planner extension consists of two modules into 3D Slicer platform. The whole overview of the structure can be seen in Figure 4.9.

4.8.1 Navigation Planner module

Navigation Planner is a loadable scripted module implemented in Python and incorporates user interface for interaction with the provided content.



Instructions: Penalty mark is represented by a number in range from 0 to input volume datatype maximum. The best score is 0. Processing error is indicated by -1.

Solution	Penalty	State	Best	Mod		
# 1	0	new			👁	🔒
# 2	1	new			👁	🔒
# 3	1	new			👁	🔒
# 4	1	new			👁	🔒
# 5	2	new			👁	🔒
# 6	2	new			👁	🔒
# 7	2	new			👁	🔒
# 8	2	new			👁	🔒
# 9	2	new			👁	🔒
# 10	2	new			👁	🔒

☐ Show other targets solutions
☑ Update slice views

Figure 4.6: Results widget

NavigationPlanner

NavigationPlanner class is the root of this module and it is responsible for initiation of the UI widgets structure. It invokes data processing and handles the returned results.

ResultsManager

ResultsManager holds all results and manages interaction for switching current navigation targets and updates results table widget.

SingleTargetResults

SingleTargetResults contains data of a single target. It manages solution candidates reviewing process and updates dependent parts of UI.

SolutionCandidate

An instance of this class represents a single solution candidate which is suggested by the system. It maintains state and manages its visualization in 2D and 3D views.

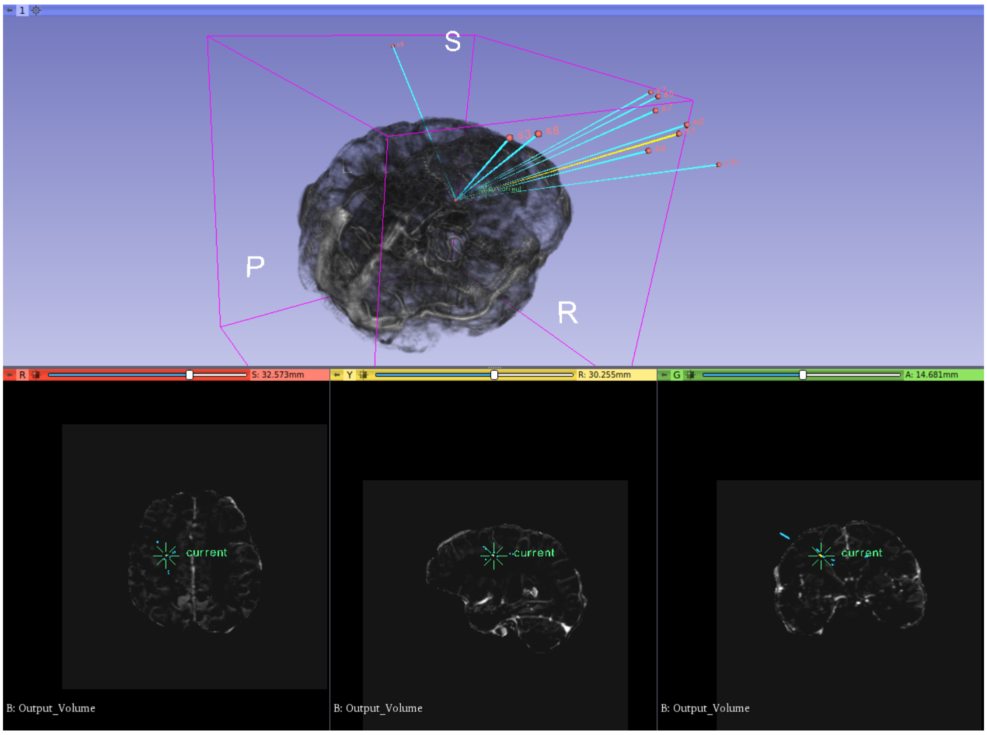
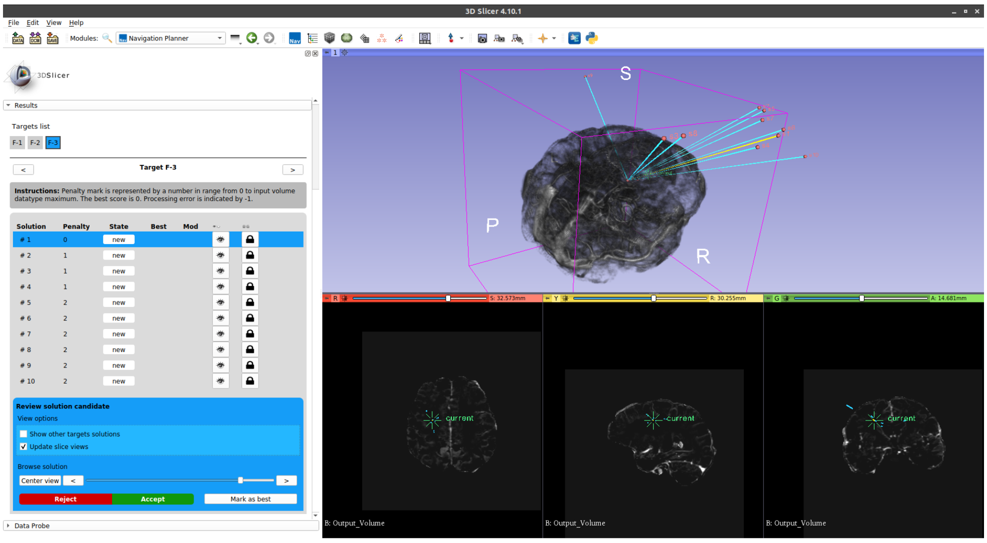


Figure 4.7: 2D and 3D views presenting solution candidates



Instructions: Penalty mark is represented by a number in range from 0 to input volume datatype maximum. The best score is 0. Processing error is indicated by -1.

Solution	Penalty	State	Best	Mod		
# 1	0	new			👁	🔒
# 2	1	new			👁	🔒
# 3	1	new			👁	🔒
# 4	1	new			👁	🔒
# 5	2	new			👁	🔒
# 6	2	new			👁	🔒
# 7	2	new			👁	🔒
# 8	2	new			👁	🔒
# 9	2	new			👁	🔒
# 10	2	new			👁	🔒

Review solution candidate

View options

□ Show other targets solutions

☑ Update slice views

Browse solution

Figure 4.8: GUI of the Navigation Planner extension

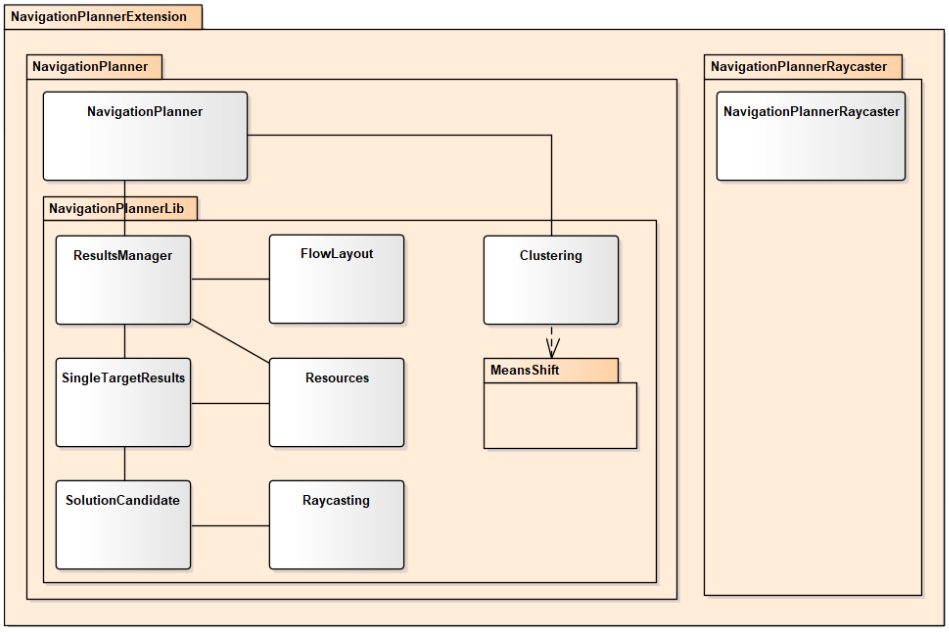


Figure 4.9: Navigation Extension class diagram

FlowLayout

FlowLayout extends QLayout class from Qt library and implements row based flow layout for organizing UI widgets.

Resources

Resourcess class encapsulates helper functions to access image resources.

Raycasting

Raycasting contains code for testing navigation paths in an interactive mode. This is a complement feature to batch processing of all data via external NavigationPlannerRaycasting module. This solution enables calculating solution score / penalty mark during modification of the navigation source position.

Clustering

Clustering class is responsible for division solution candidates into disjunctive clusters. It is depending on MeansShift package which contains implementation mean shift algorithm [29].

4.8.2 Navigation Planner Raycaster module

Navigation Planner Raycaster is a C++ CLI module which is using Slicer Execution Model. It contains an implementation batch data processing which evaluates all possible navigation paths and it returns list of solution candidates.

Chapter **5**

Realisation

5.1 Solution description

For the first iteration of the NavigationPlanner implementation we selected the solution described in the following sections. It is a subset of the designed processing pipeline but it incorporates all essential features supporting basic functionality and it creates structure prepared for a future extension.

5.1.1 Input data

Restricted area data model

We decided to use volumetric data sets as the model of restricted areas. Specifically, we use vtkMRMLScalarVolumeNode as the internal representation. It should be the most flexible solution to support both binary and intensity coding. To create a data combining information from multiple sources, we can utilize 3D Slicer built-in modules Add Scalar Volume, Subtract Scalar Volume and Multiply Scalar Volume. By using these modules, we are able to input data combining for example angiography and segmented critical regions into Navigation Planner.

List of navigation targets

To define navigation target, a representation from built-in module Markups - markup fiducials is used. It is a standard way of using a single point markers in 3D Slicer so that we can take advantage of its importing and exporting features.

Navigation sources generation

Navigation sources are generated with the simplest approach as points at the surface of input volumetric data. This utilizes the input data resolution potential and test the space of all possible navigation paths.

5.1.2 Filtering

Input data filtering was omitted in this implementation.

5.1.3 Navigation paths testing

Navigation paths are tested in two ways. This separation is necessary because data access via main module, implemented in Python, does not provide sufficient performance. The first way is used during batch processing of input data via module NavigationPlannerRaycasting. The second one, which is used during solution path modification, is implemented directly in Python module. In this case we are evaluating only a single navigation path so that the task manageable and this implementation provides sufficient reactivity of the GUI.

5.1.4 Solution clustering and solution candidates selection

After all possible navigation paths are tested, the 100 best solutions are selected and transferred to clustering. The clustering process uses mean shift algorithm to split navigation paths into disjunctive groups. After the clustering is processed, the best solution of each cluster is selected as its representative. Lists of these solutions are called solution candidates and together they create a list of suggested solutions which are presented to the user.

5.1.5 Interactive evaluation of solution candidates

Solution candidates presented in groups are split by targets and they can be marked as accepted or rejected. During the evaluation it is possible to modify suggested navigation path by changing a navigation source position. This action invokes recalculation of the navigation path score / penalty mark. At the end, a single solution candidate for each target can be marked as the best solution.

5.2 Implementation

The designed solution is implemented as an extension to 3D Slicer 4.10.1. The extension consists of two modules, the main one is implemented as a Python loadable module and the other one, independently, as a C++ CLI module using Slicer Execution Model.

The implementation tries to follow the same idea that is behind the 3D Slicer platform - to support interaction with data from other built-in or external modules. This creates a challenging development environment with a lot of variables which have to be controlled.

A reactivity of the extension seems to be sufficient. The input data batch processing for a single navigation target is finished in less then one second and the interactive evaluation of navigation paths maintains immediate response of GUI during modifications of the data (tested with an input volumetric data with a resolution 256x256x256). The performance bottleneck is a transition between batch processing, returning data from CLI module, and the phase of presenting data to the user. During this transition in performed files reading (returning data from CLI module back to the main thread) which introduce unwanted lag. On the other side, once the reading has proceeded, reactivity of the application is back to normal thus this is not a breaking issue.

5.2.1 Coordinate systems

There are used two coordinate systems in the designed modules. The first is 3D Slicer internal RAS coordinate system (Right, Anterior, Superior) and the second is volumetric data local coordinate system IJK - using volumetric data grid structure to address its elements.

5.2.2 User interface

The user interface is implemented using combination of Qt 5.11 to build UI widgets and a hierarchy structure of internal library MRML to represent constructed navigation paths. MRML nodes can be viewed in standard 2D and 3D views.

5.2.3 List of features

In this section provides a summary of implemented features.

- Loading input data coding restricted areas in the form of a volumetric scalar node
- Loading navigation targets provided as a list of markup fiducials
- Batch processing of the inputs
  - Generation of navigation paths
  - Automatic testing of navigation paths
  - Limiting an output count of the best navigation paths
  - Clustering similar solutions
  - Selecting cluster representative
- Interactive evaluation of solution candidates
  - Solution candidate rejection

    - Solution candidate acceptation
    - Indicating a state of solution candidate (new, accepted, rejected)
    - Show/hide a solution candidate
    - Lock/unlock a solution candidate
    - Navigation source modification
    - Indicating of a solution candidate modification
    - Browse navigation solution in 2D view
    - Browse navigation solution in 3D view
    - Mark solution candidate as the best for a navigation target

- Data export - navigation sources and targets export via standard markup fiducials list export

5.2.4 Limitations

The current implementation does not allow moving a navigation target during the interactive evaluation. If such action is needed, the user has to re-run the batch processing again with modified target position.

5.2.5 Known issues

The current implementation contains following issues:

- Results table widget - sometimes incorrectly re-render table view during a content update. This can be overcome by resizing module's widget column which forces the whole module's UI to repaint view.

- Build issue - a release build of the extension was made for 64bit Linux systems with 3D Slicer 4.10.1, with default modules included. The resulting build of the extension requires environment variable LD_LIBRARY_PATH to be set to ensure correct loading of the CLI module to the 3D Slicer application.

5.3 Testing

The testing was performed in a form of usability test with participants from the target user group.

5.3.1 Usability test

The testing was performed in a virtualized environment using Debian 9 OS with pre-configured 3D Slicer. Testers were given the following instructions with complementary questions about user experience.

Testing - part 1

The first part is devoted to the loading of the data and running the input data batch processing.

1. Load test volume data from /Documents/Testing-data/volume-mask.nii
2. Create 2 markup fiducials representing surgery targets
3. Switch to module ”CTU>NavigationPlanner”
4. Set ”volume-mask” (loaded volume) as Parameters > ”Volume Mask”
5. Set markup fiducials list containing targets as Parameters > ”Fiducial List”
6. Press ”Run!” to start processing
7. Once the results are calculated, you are able to analyze suggested solution candidates

Testing - part 2

The second part is dedicated to testing of the interactive evaluation of suggested solution candidates.

1. Set any solution candidate as ”rejected”

   (Hint: Select target, select solution candidate in the results table and press ”reject” at the bottom of the module’s widget pane)

2. Set ”the best” solution candidate
3. Analyze solution candidate slice by slice

   (Hint: Select target, select solution candidate, check the ”update slice views” view options. Browse solution path by pressing buttons ”¿” and ”¡”; or by moving with slider handle. Using these controllers will force to update ”current” fiducial marker which represents current position on the path of the solution.)

4. Customize position of the source markup fiducial

   (Hint: Unlock the solution candidate with the button ”locked/unlocked” in the results table. Use slider or arrow buttons to focus on the source point. Then, zoom-in closer to the source marker in 2D or 3D view and move the fiducial.)

   Note: Source markup fiducial sometimes cannot be moved because of overlapping with ”current” fiducial marker. To resolve this situation - uncheck the ”update slice views” view option, press ”¿” to move ”current” markup fiducial further on the solution path and now you should be able to move the source.

Questions for every step of the testing instruction

- Did you manage to complete Task x?
- Task x comments (optional)

User experience questions at the end of the testing

- Is the state of the application always clear (waiting for the user input, processing, ...)?
- Is the user interface intuitive and responds to the user inputs in the expected way?
- Would you suggest any change to the user interface?
- Any additional comments?

5.3.2 Usability test evaluation

The testing was done with three users from the group of target users. In this section is provided a summary of user responses.

Summary

All testers were able to follow the given instructions and complete all steps of the testing.

Following drawbacks were reported:

- Results table header has too small icons
- Objects generated by the extension should be clearly marked (e.g. custom object naming)
- Objects representing navigation paths should be named by its target ID and solution candidate ID
- Batch data processing causes lag in application reactivity

Testers' suggestions to improve user interface:

- Add option to revert back to the original navigation path after its modification
- Hide ”current” markup fiducial during the solution modification

Overall, the extension's user interface was evaluated as intuitive and suitable for the given task.

Comment to the testers' responses

In the implementation was changed the naming of generated objects. All objects are now prefixed with ”NavigationPlanner” string. Further, the objects of line models representing the navigation paths are now named in the proposed schema - including target ID and solution candidate ID. Other comments were added for processing in the next implementation iteration.

Conclusion

The main goal of the thesis was to design and implement the first iteration of a semi-automatic navigation planner for SEEG electrodes implantation. The solution should be implemented into the open source platform 3D Slicer, and it should create an alternative to a fully manual navigation planning. The designed extension successfully implements both input data batch processing and the interactive evaluation of suggested navigation paths. Even though the solution does not implement all processing steps presented in the chapter Analysis and design, it incorporates all essential features supporting basic functionality, and it presents structure prepared for a future extension.

The implementation is split into two modules. The main module implemented as a loadable module in Python is responsible for displaying the user interface, initialisation of the batch processing and controlling the interactive evaluation of suggested navigation paths. The second module is implemented as a C++ CLI module which uses the Slicer Execution Model, and it handles input data batch processing.

The key features of the presented planner are the following items:

- Batch processing of a volumetric input mask of restricted areas and a list of navigation targets
- Automatic evaluation of navigation paths based on the input mask
- Suggesting a set of solution candidates for each navigation target
- The user interface for interactive evaluation of navigation paths

Limitations and unimplemented features:

- The extension does not implement any input data filtration, and it does not consider the EEG electrode thickness
- The modification of a suggested navigation path is limited to the customisation of a navigation source

Future work

The current implementation represents a basic framework prepared for a future development. Further extension should include an implementation of following features.

- In the interactive evaluation phase - allow user to modify a navigation target
- Input data filtering
  - Introduce the input volumetric data filtering simulating an EEG electrode thickness / safety region around navigation path, as described in Chapter 4
  - Input data specific - edge detector of input volumetric data
- Explore more advanced approaches of evaluating a navigation paths
  - Consider data specific vessel detector
- Limitation of the entry angle with skull surface
- The option to disable cross hemisphere navigation paths

Bibliography

[1] Example of SEEG in non-lesional temporal epilepsy. Available from: https://www.parcdesalutmar.cat/en/epilepsia/diagnostic/estereoelectroencefalografia/

[2] Shah, A.; Mittal, S. Invasive electroencephalography monitoring: Indications and presurgical planning. *Annals of Indian Academy of Neurology*, volume 17, 03 2014: pp. S89–S94, doi:10.4103/0972-2327.128668.

[3] SBarnes. Maximum intensity projection of magnetic resonance angiography. 2009, [CC BY-SA 3.0 (http://creativecommons.org/licenses/by-sa/3.0/)]. Available from: https://upload.wikimedia.org/wikipedia/commons/6/66/Mra-mip.jpg

[4] Sparks, R.; Zombori, G.; et al. Automated multiple trajectory planning algorithm for the placement of stereo-electroencephalography (SEEG) electrodes in epilepsy treatment. *International Journal of Computer Assisted Radiology and Surgery*, volume 12, 07 2016, doi:10.1007/s11548-016-1452-x.

[5] Fedorov, A.; Beichel, R.; et al. 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magnetic Resonance Imaging*, volume 30, no. 9, 2012: pp. 1323–1341, ISSN 0730725X, doi:10.1016/j.mri.2012.05.001. Available from: https://linkinghub.elsevier.com/retrieve/pii/S0730725X12001816

[6] Thetawave. Volume ray casting. 2011, [CC BY-SA 3.0 (http://creativecommons.org/licenses/by-sa/3.0/)]. Available from: https://upload.wikimedia.org/wikipedia/commons/7/76/Volume_ray_casting.svg

[7] Janani, S.; Geetha, P.; et al. Diabetes-Epilepsy Symbiosis. *Research Journal of Pharmacy and Technology*, volume 10, no. 1, 01 2017: pp. 351–354.

[8] REKTOR, I.; OŠLEJŠKOVÁ, H. *Stručná epileptologie pro praxi.* Olomouc: Solen s.r.o., 2010, ISBN ISBN 978-80-87327-38-8.

[9] Intracranial Signal Analysis Research Group. Available from: `https://isarg.fel.cvut.cz`

[10] 3D Slicer. Available from: `https://www.slicer.org/`

[11] Zuluaga, M. A.; Rodionov, R.; et al. SEEG Trajectory Planning: Combining Stability, Structure and Scale in Vessel Extraction. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*, edited by P. Golland; N. Hata; C. Barillot; J. Hornegger; R. Howe, Cham: Springer International Publishing, 2014, pp. 651–658.

[12] Essert, C.; Haegelen, C.; et al. Automatic Computation of Electrodes Trajectory for Deep Brain Stimulation. In *Medical Imaging and Augmented Reality*, edited by H. Liao; P. J. E. Edwards; X. Pan; Y. Fan; G.-Z. Yang, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, ISBN 978-3-642-15699-1, pp. 149–158.

[13] The StealthStation$^{\mathrm{TM}}$ S8. Available from: `https://www.medtronic.com/us-en/healthcare-professionals/products/neurological/surgical-navigation-systems/stealthstation/cranial-neurosurgery-navigation.html`

[14] Neuroinspire software. Available from: `https://www.renishaw.cz/cs/neuroinspire-software--8244`

[15] Brainlab Elements. Available from: `https://www.brainlab.com/surgery-products/overview-neurosurgery-products/stereotactic-planning-software/`

[16] OsiriX. Available from: `https://www.osirix-viewer.com/`

[17] Spiriev, T.; Nakov, V.; et al. OsiriX software as a preoperative planning tool in cranial neurosurgery: A step-by-step guide for neurosurgical residents. *Surgical Neurology International*, volume 8, 10 2017: p. 241, doi:10.4103/sni.sni_419_16.

[18] Leksell SurgiPlan®. Available from: `https://www.elekta.com/neurosurgery/leksell-surgiplan.html`

[19] inomed planning software - iPS. Available from: `https://www.en.inomed.com/products/functional-neurosurgery/ips/`

[20] iPlan® Stereotactic Planning Software. Available from: `https://www.gheg.de/en/products/iplan-stereotactic-planning-software/`

[21] MNPS. Available from: http://www.mevis.com.br/index.php?mnps-101

[22] Insight Segmentation and Registration Toolkit. Available from: https://itk.org/

[23] Visualization Toolkit. Available from: https://vtk.org

[24] Qt. Available from: https://www.qt.io/

[25] SimpleITK. Available from: http://www.simpleitk.org/

[26] Ying, X.; Xin, S.; et al. An Intrinsic Algorithm for Parallel Poisson Disk Sampling on Arbitrary Surfaces. *IEEE Transactions on Visualization and Computer Graphics*, volume 19, no. 9, Sep. 2013: pp. 1425–1437, ISSN 1077-2626, doi:10.1109/TVCG.2013.63.

[27] Macqueen, J. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[28] Cheng, Y. Mean shift, mode seeking, and clustering. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 1995.

[29] MeanShift_py. Available from: https://github.com/mattnedrich/MeanShift_py

Appendix A

Acronyms

CLI Command line interface

DTI Diffusion tensor imaging

EEG Electroencephalography

ISARG Intracranial Signal Analysis Research Group

ITK Insight Segmentation and Registration Toolkit

MRI Magnetic resonance imaging

MVC Model-view-controller

SEEG Stereoelectroencephalography

SEM Slicer Execution Model

SEM-MM SEM Module Manager

UI User interface

VTK Visualization Toolkit

Appendix B

Contents of enclosed DVD

```
│ readme.txt ....................... the file with CD contents description
├─ exe ...........................................................
│  ├─ build.......................................the extension build
│  └─ testing-environment ...... VirtualBox image of testing environment
├─ src ....................................the directory of source codes
│  ├─ NavigationPlannerExtension..............implementation sources
│  └─ thesis .............the directory of LaTeX source codes of the thesis
└─ text .........................................the thesis text directory
   └─ thesis.pdf........................the thesis text in PDF format
```