Visualization, Comparison, and Selection of Brachytherapy Plans
TI3806 Bachelor Project Final Report

J.M. van Esveld
M.L.R. Prananta
A.J. van Schendel
M.H. Tavasszy

Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science

TU Delft Supervisor(s): K.A. Hildebrandt, N.H.L.C de Hoon
Client Advisor: P.A.N. Bosman
Project Coordinator(s): O.W. Visser, H. Wang

2016/2017 Q4

Preface

This reports concludes the project for the course TI3606 Bachelorsproject, a project we have worked on in the last ten weeks. The assignment was commissioned by Peter Bosman from Centrum Wiskunde & Informatica (CWI), with behind him a dedicated team of researchers from both CWI and Academisch Medisch Centrum (AMC).
The opportunity to work on project like this was for us purely coincidental. We were late with picking a project, and all interesting projects were already taken. Only through an unplanned phone call with Peter Bosman did we become aware of this assignment. Of course, when the opportunity arose, we did not hesitate. We canceled the project we had initially chosen and agreed to join Peter Bosman and his project. For all of us, this was the first opportunity we had to make a difference in the battle against cancer with the skill set we currently posses. We are grateful to have partaken in this project, and we are honored to have worked with a group of experienced and skilled individuals. We would like to say a special thanks to some notable people who helped us during the project:

- Peter Bosman, for giving us the opportunity to work on this project.
- Anton Bouter, Stef Maree and Marjolein van der Meer, for helping us with specific problems we encountered and providing feedback and resources to us.
- Klaus Hildebrandt, our coach from TUDelft who guided us and advised us during the project.
- Niels de Hoon, a member of the coaching team who provided valuable feedback to us during the project.

Delft, June 2017

J.M. van Esveld
M.L.R. Prananta
A.J. van Schendel
M.H. Tavasszy

Abstract

Cancer is a deadly disease which occurs when cells divide uncontrollably. In the battle against cancer, many types of treatments exist to preferably cure the disease. Brachytherapy is one such type of treatment, a form of radiotherapy where the radiation source is internally placed through implanted catheters.

A research team consisting of researchers from Centrum Wiskunde & Informatica (CWI) and Academisch Medisch Centrum (AMC) have developed a solution that generates a wide range of near-optimal brachytherapy treatment plans for treating prostate cancer, which they have named the brachy automated treatment-planning system (BATS). However, the generated plans can currently not be viewed in any tangible way. Physicians must be able to oversee and look into the generated plans before they can act upon it. In addition, the physician must be able to compare multiple treatment plans and must be able to select the most optimal treatment plan.

We have developed a front-end solution that works in tandem with BATS that provides the necessary features to create a working environment suited to the needs of the research team. Multiple plans can be viewed and compared through visualization and plan properties. Additionally, the application has DICOM compatibility and can operate on Windows and Linux operating systems.

While not all requirements are met, we are satisfied with the resulting application. Recommendations for future iterations include improved navigation through icon-based buttons, and more control options through hotkeys and the like. Additional 3D treatment plan visualizations and multi-plane views would provide opportunities for 3D planning.

Contents

1. Introduction

Cancer is a deadly disease which occurs when cells divide uncontrollably. Cancer can occur in many organs and many different types of treatments exist to preferably cure the disease. Brachytherapy is a form of radiotherapy treatment effective against certain types of cancer, such as prostate cancer. It is a form of radiotherapy where the radioactive source is internally placed, often next to the organ itself through implanted catheters [2]. A research team consisting of researchers from Centraal Wiskunde & Informatica (CWI) and Academisch Medisch Centrum (AMC) have developed a solution that generates a wide range of near-optimal brachytherapy treatment plans for treating prostate cancer, which they have named the brachy automated treatment-planning System (BATS). However, the generated plans can currently not be viewed in any tangible way. Physicians must be able to oversee and look into the generated plans before they can act upon it. In addition, the physician must be able to compare multiple treatment plans and must be able to select the most optimal treatment plan, which is currently not possible with BATS alone.

The goal of this project is to streamline this process by developing a front-end solution to BATS; a graphical user interface (GUI) that can visualize multiple treatment plans and highlight the differences between them. For the development of our program, we set to research the following research question:

- How can we aid a doctor in choosing an as optimal plan from multiple brachytherapy treatment plans using a front-end GUI?

We derived the following sub-questions to help us answer the main research question:

- How can we relay information about the patient and brachytherapy plans within our software?
- How can we streamline the general procedure of selecting brachytherapy treatment plans with the use of our software?

This report will describe our research and the development of the program. We first provide a description of the problem and the requirements that were set. We follow up with a look into the current state-of-the-art in GUI design, medical visualization and brachytherapy treatment planning systems. The chapter following will describe our program in greater detail, as well as our justifications for the design choices we made. In the discussion chapter, we evaluate our results and provide recommendations for the application. It ends with the conclusion, where we sum up our thoughts and results and formulate our answer to the research question.

1.1 Background

This section will explain the background knowledge needed to understand the context of this project and the definitions used in this project. It focuses on explaining the method of brachytherapy, the creation of treatments plans, and the difficulties that arise.

Brachytherapy is a type of radiotherapy effective against certain types of cancer, where the source of radiation is internally implanted, as opposed to methods like external beam radiotherapy (EBRT) [2]. Radiotherapy is a often-used method, where ionizing radiation is used to destroy cancer cells. Like most treatment methods, adverse effects are inevitable due to real-world constraints. As such, brachytherapy becomes a delicate balancing act between providing enough dose for cancerous tissue and minimizing the dose for normal tissue. In our research we will focus on high-doserate (HDR) brachytherapy for treating prostate cancer.

When performing this procedure, a radioactive source is pulled through a catheter which has been implanted within the body. It stops at so called dwell positions for a predetermined amount of dwell time. By tweaking the dwell times, certain regions receive bigger doses or smaller doses of radiation [12]. Certain guidelines for the maximum and minimum dose per region are determined per patient. For some patients it might be beneficial to protect normal tissue or organs at risk (OARs), while other patients might benefit more by blasting the cancerous tissue with a high dose of radiation. These constraints are derived from dose volume histograms (DVH). A tabular representation of a DVH can be found in Table 1.1. This example table shows that at least 95% of the planning target volume (PTV) should receive 100% of the prescribed dose. However, no more than 50% of the PTV should receive 150% of the prescribed dose.

ROI	**Dose [%]**	**Volume [%]**	**Volume [cc]**
Bladder	74.00		≤ 2
Bladder	86.00		≤ 1
PTV	100.00	≥ 95	
PTV	150.00	≤ 50	
PTV	200.00	≤ 20	
Rectum	74.00		≤ 2
Rectum	78.00		≤ 1
Urethra	110.00		≤ 0.1
Vesicles	80.00	≥ 95	

Table 1.1: An example of DVH Indices[22]

There are a few steps to the brachytherapy treatment planning procedure. The first step determines the relevant organs for which the volumes are contoured. Next, the implants and applicators are reconstructed through medical images. Based on this information the dwell times and the dose distribution can be calculated and optimized. This optimization process is based on two objectives: Dose to cancerous tissue and dose to normal tissue or OARs [22]. This process is done through specially made medical applications and overseen by a physician, who in the end makes the final call for the viability of a treatment plan.

2. Problem Description

Our client and his team of researchers developed a back-end solution to generate near-optimal plans, aptly named the brachy automated treatment-planning system (BATS). This system generates a range of near-optimal treatment plans constrained within the given objectives. A downside to this process is that the generated plans need to be manually reviewed by a physician, while the amount of plans for one patient can reach well into the thousands. No solution currently exists that provides the ability to automatically oversee the plans, view the plans and compare the plans with each other. Thus, the client has requested us to develop this solution, a program that works in tandem with BATS which can be used to view multiple plans and compare them if necessary. A full description of the project can be found in Appendix A

The client also requested DICOM compatibility in the front-end application. The DICOM file format, which stands for Digital Imaging and Communications in Medicine, is the international standard for patient data and medical images [7]. Before the start of this project, the research team had to juggle with multiple programs and file formats in order to retrieve and export patient data from one program to another. To improve the workflow for the client and his team, the program must be compatible with both patient (.pat) and DICOM (.dcm) files.

2.1 Requirements

This section will list all the requirements for the application, as precise and concise as possible. The requirements are split into categories according to their importance to the success of the project according to the MoSCoW method. The requirements were verified by the client and by our supervisor.

Must Haves

- Run on both Windows and Linux.
- Provide an overview of all existing treatment plans.
- Provide the user of a more detailed overview of a specific treatment plan, including:
    A slice-by-slice graphical view of the involved organs.
    A slice-by-slice graphical view of the different radiation dose levels
    Provide a preview of these plans when the user hovers over, in terms of a tabular representation of the dose-volume histogram and other numeric indices.
- Compare at least 2 different treatment plans:
    Side-by-side
    Overlayed, where differences are concretized.
- Read patient .pat files, including imaging.

Should Haves

- Show the treatment plans in the form of a pareto front from which the user can select plans.
- Mark and unmark potentially interesting treatment plans, e.g. adding them to a "Favorites" list, that can be compared side-by-side.
- Revert actions, by providing a "undo"-button.
- Navigate between different windows, by providing home, back, and forward buttons.
- Open the plans location in the file system by providing a button to the user.
- Mark the chosen plan within the list of selected plans.
- Show a 3D visualization of treatment plan data.
- Read .dicom files directly.

Could Haves

- Record user actions for analysis.
- Support a customizable UI layout.
- Add notes to treatment plans.
- Record recent files for easy access.
- Utilize hotkeys for GUI elements.
- Save the state of the program for each patient, meaning the favorites, notes, and ultimately chosen plan.
- Clear aforementioned state and reset, providing a confirmation window before resetting.
- Revert a state reset.
- Move comparison windows to separate windows (pop-out)

Won't Haves

- Adjust or change treatment plan data
- Adjust program colors or fonts

3. State-of-the-art

In this chapter we review our findings of our research into state-of-the-art GUI design, medical visualization and brachytherapy treatment planning systems. These are all relevant subjects that play a role in the design of our application and it gives insight into how we should approach certain elements of our program.

3.1 User Interface

Many researchers and designers have looked into designing graphical user interfaces (GUIs) that provides the best user experience (UX). In general, there are a few points that most researchers agree upon, that can be summarized to a list of design principles. The most important of these are listed below [11, 14, 13].

Firstly, it is important to understand what the user wants to achieve, and what the user wants to know. One should assume that the user is uninformed about the application. Preparing for the worst-case scenario will leave you with an easy to use program. In the case of this project, the users are clinical staff, such as physicians. They are used to current state-of-the-art medical software, and most likely have little to no knowledge about computer science. The application will be used to assist them in their work, and should therefore be as simple as possible to use, while still maintaining functionality. We have talked to our client, who have worked with physicians, and gained some insights about the prospective users. Physicians prefer methods they are used to, an important factor to consider. Modern clinical applications often show 2D, slice-by-slice visualizations of the dose distribution around the organs. We will therefore provide such a representation as well in our application.

Secondly, it is also important not to scare or overwhelm the user. Programs that show a tenfold of buttons in the main screen might indicate complexity for users. It is important to understand that users want to pick up a new program as soon as possible without the hassle of learning everything about it. Adobe, with programs like Photoshop, InDesign, and Illustrator, has done a good job at maintaining a clean look without hiding all the functionality. They put advanced functions in a menu that requires a couple of clicks to get to, leaving the most basic and most used functions at the surface. The program is capable of performing advanced image editing, but looks simplistic on the surface.

Thidrly, messages should not be long, rather short and to the point. The user want to use a program as efficiently as possible. The user will most likely not read messages provided by the application for entertainment. The use of icons is a good way of reducing clutter. However, these icons should be thoroughly tested to make sure they are understood correctly. A good example of this is shown by Hobart [14]. An icon was designed to resemble so-called "rolled up" totals, a function in accounting software. The designer decided to use a cinnamon roll as icon. While clever from the designers perspective, the users could initially not understand the function of the cinnamon bun button at all.

At last, always select the right container for certain elements. This is especially important for pop-up windows. While they grab the users attention immediately, they are often perceived as annoying. As a designer, always strive for subtle yet clear messages. Some windows might have to be able to move around. Coming back to the Adobe example, in Photoshop different files and menu bars can be rearranged within the application. Certain other elements cannot by design, as there is no use for this or they would get lost. As a designer it is important to give the user space for personalization, but it is also important to set limitations as to not allow them to clutter the program.

3.2 GUI Libraries

When developing an application that uses a GUI it is common to use an external GUI library. This prevents developers from having to reinvent the wheel for every application. These state-of-the-art libraries all provide different features and are specialized in different fields. Some are lightweight while others are packed with extra features. There are few features that the GUI library is required to have for an application that satisfies the requirements described in Section 2.1. The most important ones are: cross-platform support, OpenGL support and C++ support. This narrowed down the list to four commonly used GUI libraries: wxWidgets, FLTK, GTK+ and QT. This section will go into detail why we made the decision to use wxWidgets for our application.

The look and feel of an application is an important factor when creating a UI which the user can easily understand and read. When a user does not recognize the look and feel than this could work against the first two design principles mentioned in the previous section. An example being that the user can become overwhelmed if it sees all kinds of new UI elements. Another problem, which goes against the first principle, occurs when the user cannot find an element because it has been moved or displayed differently than the user is used to. Because of this we wanted a library that could recreate the look and feel of the operating system (OS) it is running on.

Both GTK+ and QT try to emulate the look and feel of the OS without using the native controllers of that OS. This could allow for more customization options for the developer while still maintaining the native look and feel. WxWidgets does use the native controllers to create a truly native feel, this does however mean that there is less room for customization since all platforms need to support it. FLTK is the only library that does not aim to recreate the OS native look. This means that it will be harder for the user to get used to the program and that is the main reason why we did not use FLTK.

Another factor to consider is price, we were not planning on buying a license since this project would only last for a few months. QT is the only library which is not open source or free and requires a license to use. It has two models, one for commercial applications and another model that requires the project to be open source. Both were not an option because the project needed to stay private and we were not willing to pay for the library.

The experience of the user is the most important factor to consider, but the experience of and the ease of use for the developer also needs to be considered. The options that were left both have extensive documentation as well as active forums for questions. WxWidgets is available for multiple programming languages including C++, and supports OpenGL out of the box. GTK+ on the other hand requires two extensions to support C++ and OpenGL.

This leaves us with wxWidgets. It is free to use, requires no extensions or special licenses, and provides the native look and feel of the OS. The only thing that is missing is extensive customization due to the use of native SDKs. This was however not a major requirement, and in the end we chose wxWidgets as our GUI library.

3.3 Medical Visualization

In order to visualize the data needed for this program, several data sources need to be merged. There is input from medical images, such as CT and MRI images, radiation sources, and organ volumes. All these elements need to be visualized concurrently, without confusing the user. It should strive to accelerate decision making and improve the quality of the decisions made. According to Bühler [5], designing effective and efficient image fusion methods is a creative process and there are no standard recipes. This section will discuss the different technical methods of merging images, as well as methods that make the final image understandable, interactive, and as useful as possible.

Image Fusion Methods

A study by Dong et al. [8] states that there are four moments in a visualization process where merging can be performed: At signal level, pixel level, feature level and decision level.

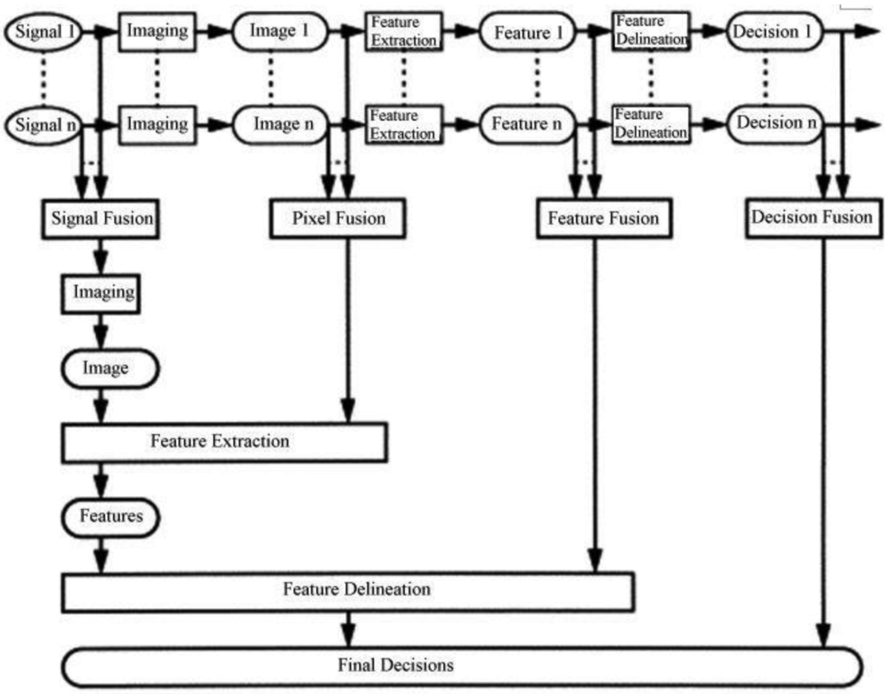


Figure 3.1: The different levels of image fusion.[6].

The different fusion methods are shown in Figure 3.1 on the third row. These are increasing in complexity, with signal level fusion being the least complex. Unfortunately, signal level fusion is impossible for this project. The signals are already processed and stored in different data types. The other three types of fusion will be discussed below.

First, there is pixel fusion. This fusion method is very straightforward. All input data is aligned and a merging algorithm will go over each pixel. For every pixel, the information from the corresponding pixels in the source images is analyzed. According to their importance, the source pixels are assigned more or less weight in the output pixel. An output pixel could be an (weighted) average of the input pixels, one pixel on top of another, or a mix between these two.

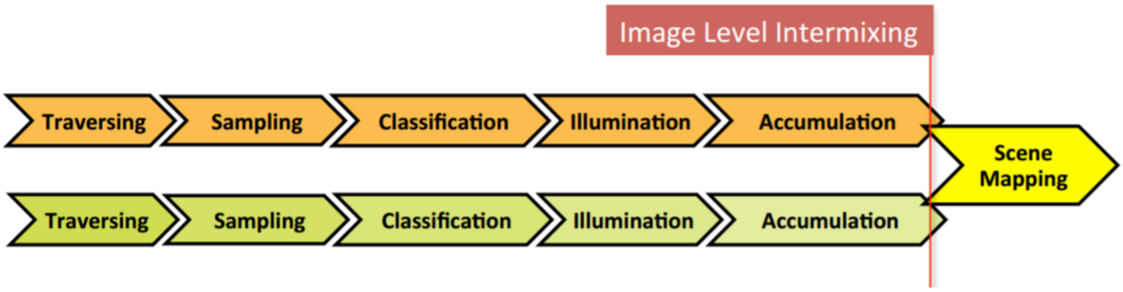


Figure 3.2: The graphic pipeline with the pixel level fusion point indicated [20].

The rendering pipeline is shown in Figure 3.2. Pixel level fusion can happen anywhere after sampling, until the point indicated in the figure. This figure was chosen, since the input data in this project is already fully processed. No classification has to be done, illumination would be done separately for each layer. The fusion would then be performed right after the accumulation of data for each layer.

Next up is is feature fusion. Feature based fusion is more complex than simple pixel fusion. It requires different objects to be recognized in all sources. Notable features, like edges, textures, and intensities from different objects are extracted. Similar features from different input images are then fused.

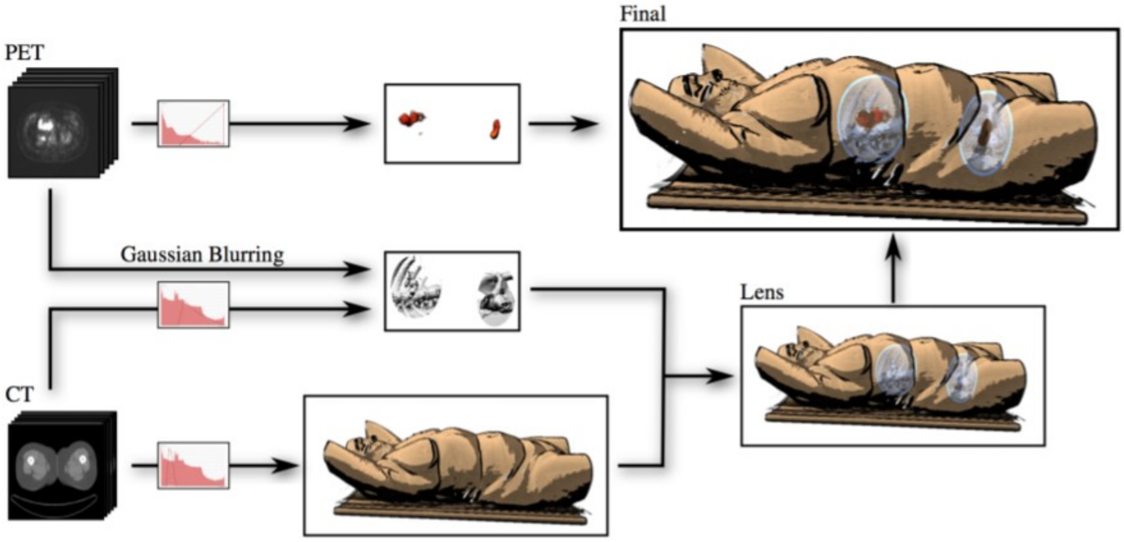


Figure 3.3: Feature Fusion: Illustrative Rendering[15].

A feature fusion example is shown in Figure 3.3. This process merges positron emission tomography (PET) data, used to detect and measure metabolic activity, with CT scans. The PET data is first blurred, then used in combination with the CT data to create a clear organ visualization. This is then merged with the CT scan. Finally, the PET data is added, and a final image is rendered.
Lastly, there is decision fusion, a more abstract way of merging images. It combines the results from a couple of different algorithms to produce a final fusion. All input images are processed individually to extract their information. Using decision rules, the final data is then extracted.

Smart Super Views

With a limited amount of screen area, and an image that has to be big to allow for as much detail as possible, one might run out of screen space. If options can be hidden behind different parts of the image, this might solve a big part of this problem. This is where smart super views come in. A smart super view is a view where the image acts as the user interface. Options appear when the user hovers over different parts of the image. These options differ depending on the location and type of object that the image represents on that location. An example is shown in figure 3.4.

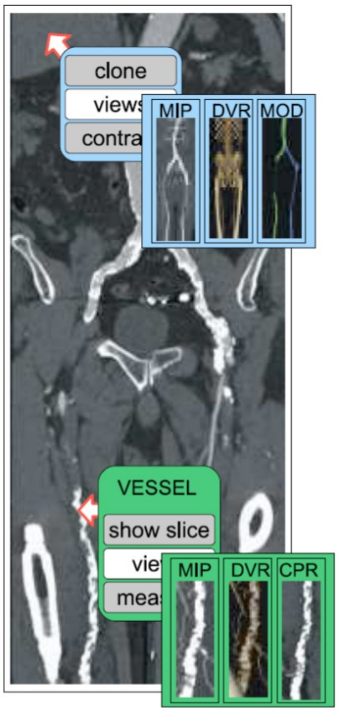


Figure 3.4: An example of a smart super view[17].

In this example, different visualizations such as maximum intensity projection (MIP), direct volume rendering (DVR), and maximum intensity difference accumulation (MIDA) are merged into one image. Depending on the type of tissue a different type of visualization is considered optimal. A decision is made when the image is clicked, based on the type of tissue below the cursor. If the type is bone, then DVR might be optimal. When the type tissue of is a vessel, CPR might be better. Based on this knowledge, different information is shown to the user.

Of course, the application will be different when used to visualize the dose distribution. The application would need to show the dose level when hovering over the image, hiding or showing organs when clicking on them, or showing more detailed differences in two plans when clicking at a certain point of the plan. We take this information into account when designing our program.

3.4 Brachytherapy Treatment Planning System

The procedure of brachytherapy treatment planning consists of the following steps [2]:

- Define the planning target volume (PTV) and organs at risk (OARs)
- Reconstruct the implanted sources or catheters
- Calculate and optimize the dose distribution
- Evaluate the dose distribution

For each step certain medical applications can be used to satisfy the step procedure. Multiple procedure steps can be satisfied with a single application. It is clear that the intention of our application is to evaluate the dose distribution of a treatment plan in greater detail.

Brachytherapy Treatment planning can be approached using 2D methods and 3D methods Only when all the above mentioned steps are processed using 3D methods can the planning be defined as 3D treatment planning [2].

3.4.1 Oncentra Brachy

Oncentra Brachy is a brachytherapy treatment planning system (TPS) developed by Elekta [9]. Oncentra Brachy is used to cover all the steps of the brachytherapy planning procedure and is currently being used at AMC for practical and research purposes.

Oncentra Brachy provides a variety of tools to ease the process of treatment planning. It can be used to map the anatomy of the patient by contouring regions of interests (ROIs), by scrolling through medical images slice-by-slice. Likewise, the same view can be used to reconstruct catheters. Oncentra Brachy also has the ability to plan in 3D, through a feature described as "contouring in arbitrary planes." Contours can be drawn in any plane and are then automatically translated to other planes, which increases the workflow [9].

Implant reconstruction time is reduced with applicator modeling and implant modeling. Both automate some of the processes involved during applicator and implant reconstruction. Again, Oncentra Brachy implements 3D planning by using 3D geometric models that can be inserted in the image set [9]. For the final steps of the procedure, Oncentra Brachy provides tools to automate the dose distribution optimization using novel algorithms. Arguably the most important step of the treatment planning, Oncentra Brachy optimizes the workflow to reach this stage of the procedure as soon as possible [9].

The program we are developing does not cover all the steps like Oncentra Brachy. Oncentra Brachy works perfectly as a TPS, but our program will handle the evaluation of treatment plans differently by visualizing dose distribution of multiple plans for comparison. Despite this, Oncentra Brachy provide functionalities and features which would also benefit users of our program.

First, Oncentra Brachy supports full DICOM compatibility, meaning import and export of DICOM RT files. This feature became a requirement from the clients, due to the difficulties they had juggling with multiple file formats in their back-end program. By implementing DICOM compatibility, all other file formats could be discarded, which would increase the workflow for the research team.

For 2D visualizations, Oncentra Brachy uses 2D medical images and overlays them with isodose lines and colorwash. Medical images like CT, PET and MRI are fused and can be freely navigated and scrolled through. Furthermore, a feature referred to as live isodose was requested by the physicians we interviewed. This feature originated from the software they used, Oncentra Brachy. Live isodose generates real-time isodose lines on the spot over medical images. Oncentra Brachy also provides control over DVH parameters using a specially made DVH dashboard. An overview of the DVH parameters let physicians check if the objectives of the treatment plan are met.

4. Program Description

4.1 General Overview

In this section we give an overview of the features we developed for this program. These features satisfy most of the requirements set by the client. We will discuss some features in more detail in follow-up sections where we will also provide a small history of the development of the feature and justifications for our design decisions.

- The program can operate on Windows and Linux operating systems.
- The program has a customizable UI in which elements of the interface can be moved around. UI elements can be docked into the main window or set to float as a separate window.
- The program can visualize brachytherapy treatment plans in 2D by rendering pre-set isodose lines and dose distribution heatmaps, overlayed on the contours of the organs, dwell points and additional medical images on a slice-by-slice basis. These visualizations are shown in a screen within the visualization window when a plan is selected and loaded. The plan can be navigated with the mouse and the user can zoom in and out for more detail. In addition, all plan properties including DVH parameters are shown in a table below the visualization. The parameterse are marked either green or red to indicate if the DVH parameter is within the medical constraints.
- We implemented a live isodose line feature. In addition to the preset isodose lines, an isodose line is generated and moved to the location of the mouse within the plan visualization. The isodose line is generated using the dose level retrieved from the plan at where the mouse is currently hovering.
- We implemented an interactive visualization of the Pareto front. The Pareto front shows all plans plotted against two target values, dosage to target organs and dosage to non-target organs. The Pareto front shows a quick overview which the physician can use to select a region of plans which is in his interest. Each plotted plan can be selected by clicking on it and will appear in the visualization window.
- We implemented a method to select and view multiple plans at once. Once a plan is selected in either the list of plans or in the Pareto front, the plan visualization and the plan properties (such as DVH parameters) will appear in the visualization window. Any newly selected plan will be added to the right of the currently visualized plans.
- We implemented a dedicated comparison view to compare a reference plan with one other plan. This comparison view is available in a separate tab of the visualization window. To make use of the comparison window a plan can be set as a reference plan, followed by selecting any other plan. The comparison window has three plan visualization screens side-by-side; the left screen visualizes the reference plan. The right screen visualizes the plan of the other selected plan and the middle screen shows a comparison visualization. The comparison visualization shows the differences between two plans using a special colorwash heatmap and isodose lines.
- The program is DICOM compatible. This means the ability to import and export DICOM files. DICOM is the international file standard for patient data and medical images. As such, many medical applications implement DICOM compatibility. The patient data and medical images provided by DICOM are used in the visualization of plans.

4.2 Visualization

An important requirement for this project was to visualize the details of a brachytherapy treatment plan. We decided to visualize these plans in 2D. 2D visualization of medical plans is often the most clear cut way to show the situation. According to some physicians we interviewed, they only view plans in 2D and almost never in 3D. The main advantage of 3D visualization is improved context of the situation. It can provide a better view of the affected organs, but we expect the physicians to be experienced and used to 2D visualizations of medical plans.

The main purpose of visualizing the brachytherapy treatment plan is to visualize the dose distribution. The visualization of the dose distribution is then layered over visualizations related to the patient, such as contours of the organs, dwell points and CT or MRI images related to the patient. The visualizations related to the patient provide context for the situation. Implementing the visualization of the dose distribution provided the greatest challenge, and in the end we implemented two methods of viewing dose levels in a plan. One is through isodose lines, and the other through colorwash heatmaps.

4.2.1 Isodose Lines

Isodose lines show the contour of a certain dose level layered over a slice of an organ or the human anatomy [1]. Isodose lines are useful to quickly perceive an area of a certain dose level, without having to see the rest of the dose distribution. For example, one can easily perceive if areas of an organ are completely encompassed within a dose level, using isodose lines. Physicians often work with isodose lines for this reason. They often put more value into isodose lines than other methods of dose levels visualization according to the physicians we spoke to.

We have implemented isodose lines for the visualization of brachytherapy treatment plans. On default, the program shows isodose lines for 20%, 100% and 150% of the prescribed dose. Through UI control elements you can add new isodose lines at custom dose levels and remove dose lines as well. For the comparison screen we show the 100% isodose line on default.

The implementation of the isodose lines consists of several parts. To retrieve the dose levels needed to generate the isodose line, we calculate for each slice an grid of dose points. The grid contains 120 by 120 dose points and covers an area of $14\,400\,\mathrm{mm}^2$ positioned over the most relevant organs, the prostate and the organs surrounding it.

Using this dose grid, we apply a method called the Marching Squares method to calculate the lines used for the contour [16]. The Marching Squares method goes through each square of the grid, with a square being a collection of four neighboring points that form a square. Given the dose level for the isodose line, for each corner we determine if the value is above or below the dose level. When each corner has been processed, we assign a value to the square that corresponds to one of the $2^4 = 16$ possible outcomes. Afterwards each square is processed and the contour is rendered.

In each outcome, the line starts and ends between a pair of dose points. In the initial implementation, the start and end points are set exactly halfway between two corners. This results in rather blocky contours however. While increasing the density of dose points improves the result, the blocky nature of the contour is still apparent when zoomed in. To create more smooth contours, a form of interpolation was applied.

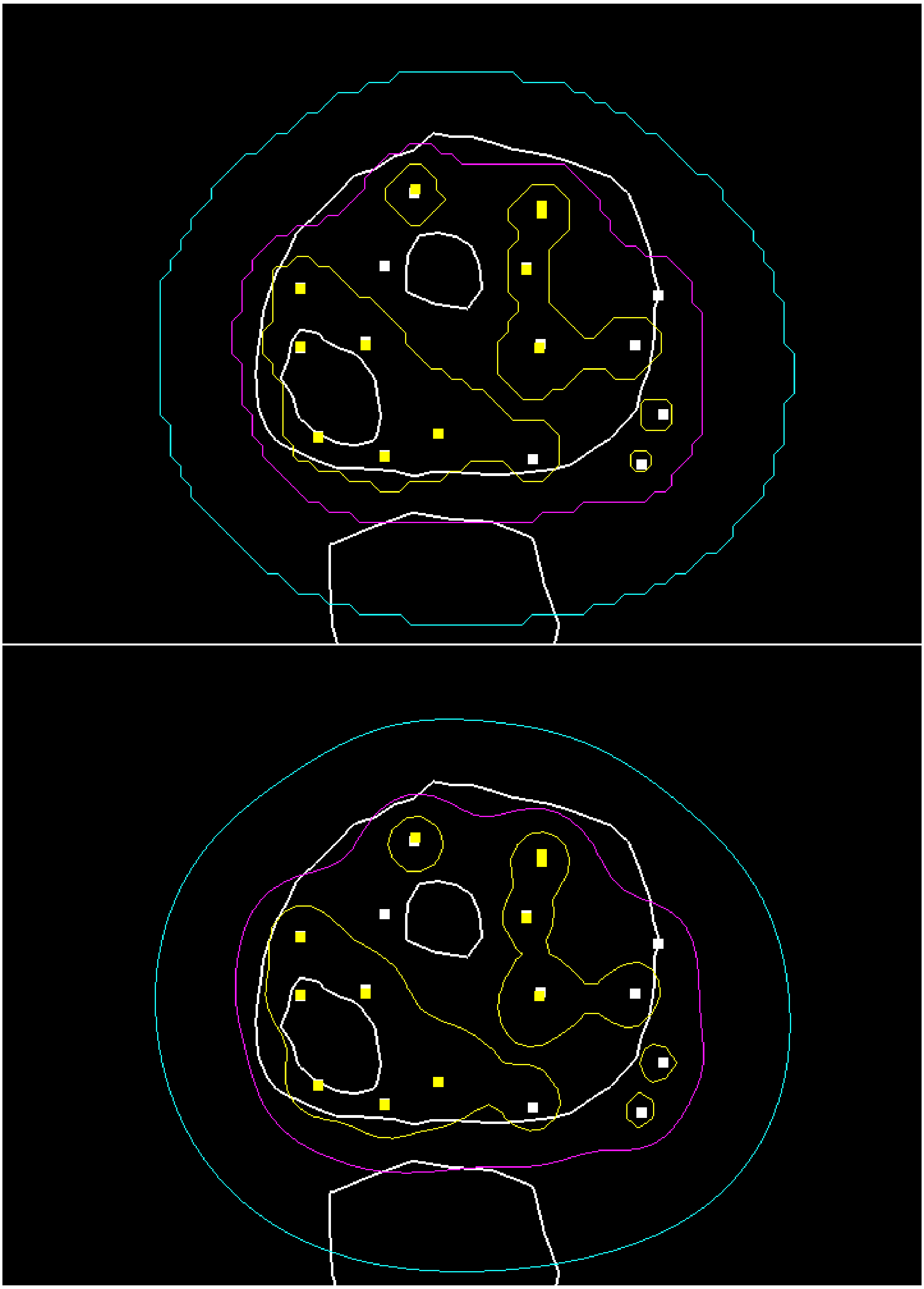


Figure 4.1: Effects of interpolation on contour smoothness. The top image shows contours rendered without interpolation and the bottom image shows contours rendered with interpolation.

The initial idea for interpolation was to add more points between the points that were established with the Marching Squares method. In the end we went with a more straightforward method, by simply shifting the start and end points of the line, interpolated using the values of the corners and the given dose level. The contours lose the blocky look by applying this interpolation, but the precision and smoothness of the contours are still largely dependent on the density of the dose points within in the area.

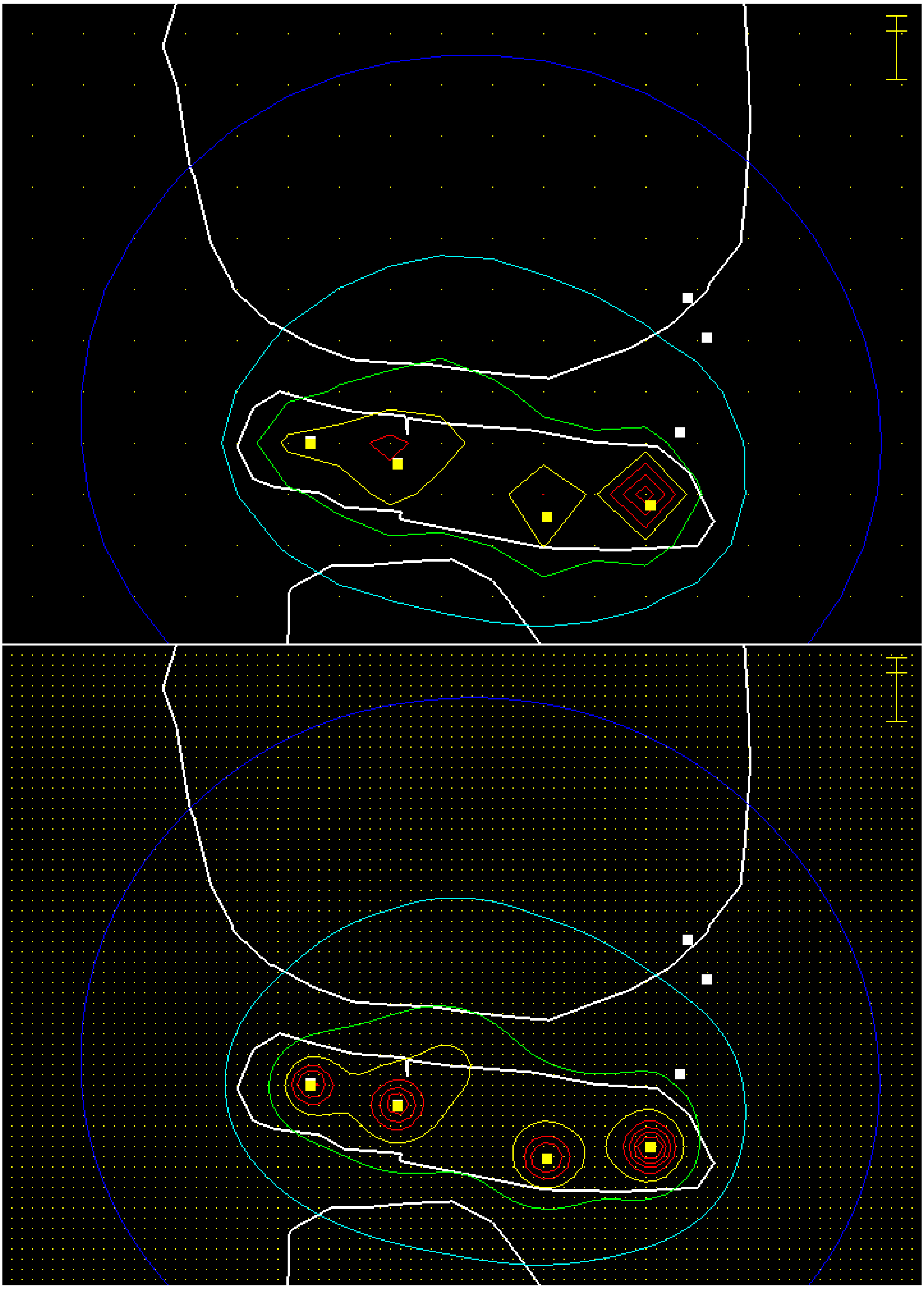


Figure 4.2: Effects of dose points density on contour smoothness. The top image shows contours rendered with a 20 by 20 dose points grid in a 10 000 mm^2 area. The bottom image shows contours rendered with a 100 by 100 dose points grid in the same area as the top image.

A feature related to the isdose line is the live isodose feature. With this feature, the user can generate a real-time isodose line anywhere on the plan. When hovering the mouse over the visualized plan, an isodose line is generated at the mouse position using the dose level retrieved from the mouse position translated to the plan position. In addition, a tooltip appears that shows the dose level percentage on the position of the mouse. This feature was inspired by a similar feature from Oncentra Brachy treatment planning system [9].

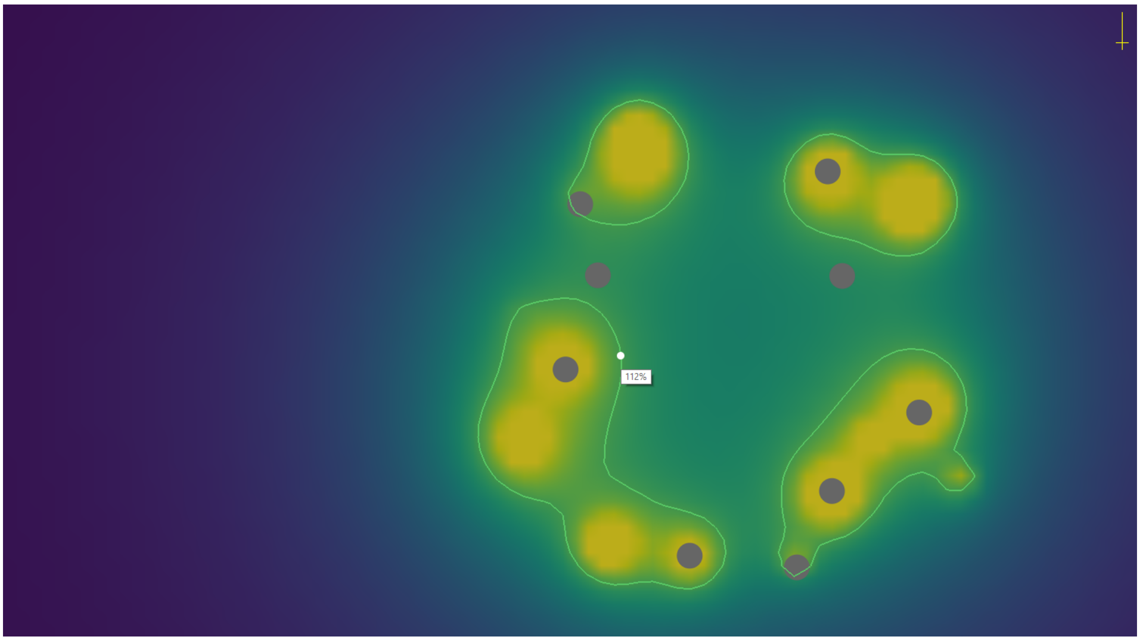


Figure 4.3: The live isodose feature in action. A isodose line is generated at the position of the mouse and a tooltip displays the dose level percentage.

The colors for the isodose lines used in the final build are mapped to the same colormap used for the colorwash heatmaps, which we will discuss in the next section.

4.2.2 Colorwash

Colorwash in medical context is often used to visualize a heatmap of the dose distribution over an area within the body. Dose levels as a percentage of the prescribed dose are mapped to a colormap to indicate with colors how high the dose is in an area. This provides a different and more complete way to show the distribution of the dose over certain areas. For the colorwash, It's important to choose a colormap that is clear and concise, as it has to convey important information to physicians. Through research, it has been determined which colormap properties are important to consider when clarity and compatibility are desired.

We initially implemented our colorwas with the JET colormap, which has for a long time been a default colormap for many applications, like MATLAB. The colormap contains a wide range of colors, basically passing all the colors of the rainbow. It is easy to implement, but it is suboptimal in certain circumstances. For example, when the colormap is converted to grayscale, it loses the information it holds over the data. The converted colors won't map properly anymore to the data, as certain colors become the same. This is as a result of a property that is not being upheld by the colormap, i.e., the color map is not perceptually uniform.
A perceptually uniform colormap is defined as having a horizontal line when the perceptual deltas of the colormap are plotted [23]. When colormaps are perceptually uniform, it improves the clarity of a colormap by making it more visible to colorblind users and by retaining as much information as possible when converted to grayscale. JET is not perceptually uniform in regular form and when converted to grayscale. Thus, JET does not suffice as a colormap for the dose distribution colorwash.

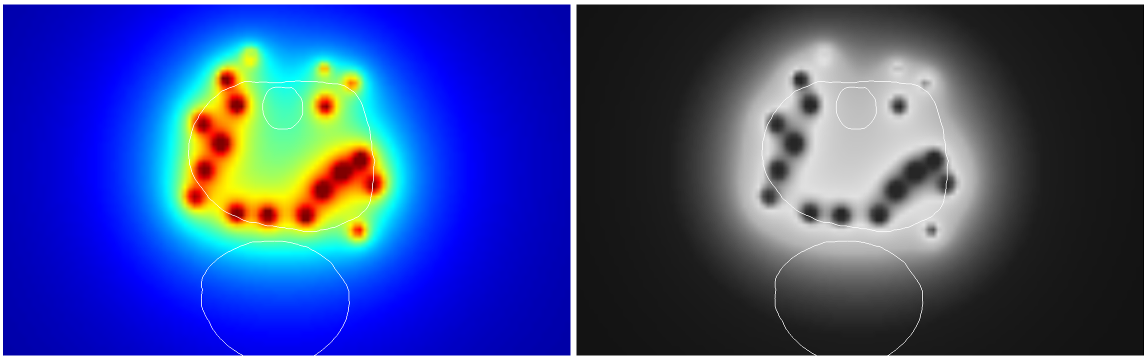
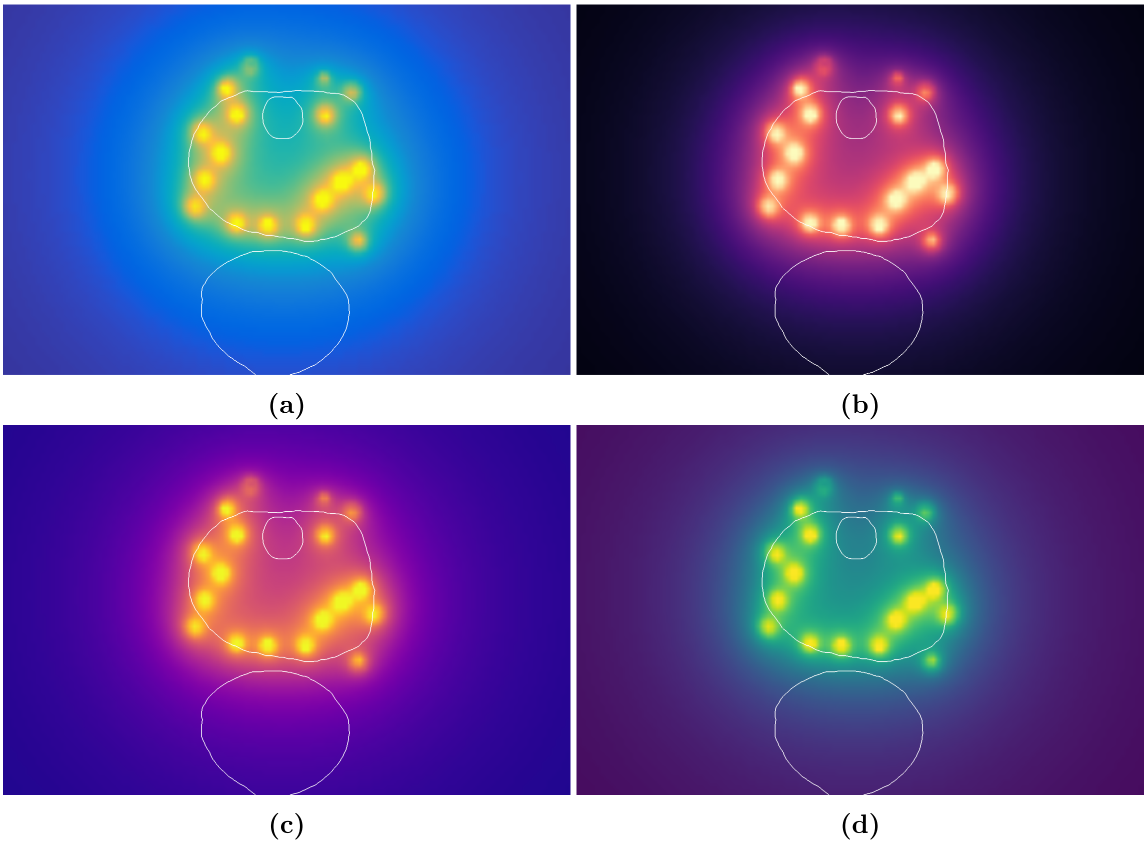


Figure 4.4: The JET colormap and the effects of converting it to grayscale. Dark blue and dark red tints become similar after grayscale conversion, it is impossible to retrieve the original information.

Several new colormaps have been developed that have been analytically designed to be perceptually uniform, both in regular form and grayscale form [23]. The differences between the colormaps lies in the range of colors used by the colormap. When all colormaps are perceptually uniform, the choice for a colormap is mostly based on preference or aesthetic reasons. We will discuss some of the choices we considered next.



(a) **(b)** **(c)** **(d)**

Figure 4.5: A collection of implemented color maps. (a) The new MATLAB default colormap, Parula. (b) The Magma colormap. (c) The Plasma colormap. (d) The Viridis colormap developed by Eric Firing [23]. The latter three are all perceptually uniform in regular form and grayscale form.

MATLAB has introduced a new default colormap named Parula, which replaces the old default colormap JET. This colormap improves upon jet with perceptual deltas that are more uniform, but the colormap is not perceptually uniform. We implemented this colorwash to see how it would perform, but we quickly noticed that some of the problems we encountered with JET were still present. For example, a bright blue halo can be seen in Figure 4.5(a) formed around higher dose levels. A similar effect occurs with the JET colormap.

This left us with three perceptually uniform colormaps, named Magma, Plasma and Viridis, displayed in Figure 4.5. There are no odd artifacts that are similar to what occured with Parula and JET. Our choice was based on which colors were most fitting for the context. In the end we went with Viridis because of the neutral use of the green and yellow tints. Red-green colorblind users, the most common type of color blindness, also benefit from this colormap [21]. Figure 4.6 shows the effects of grayscale conversion. The colormap retains much of the original information despite losing its colors.

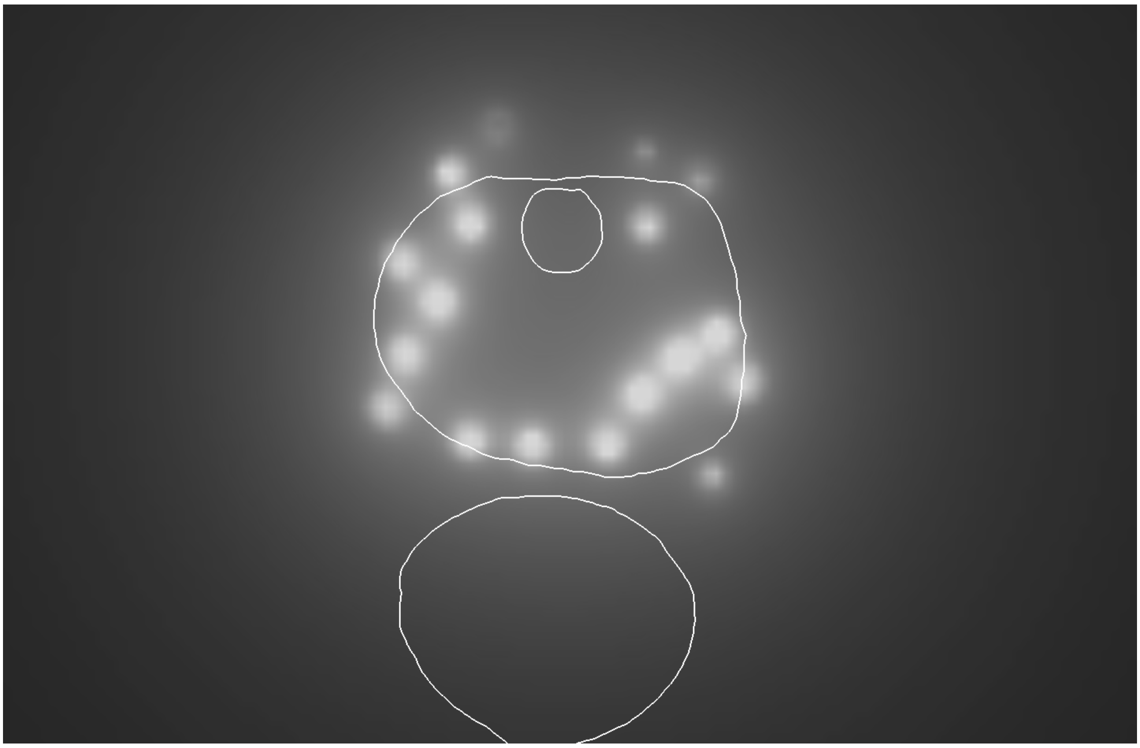


Figure 4.6: Viridis colormap converted to grayscale. Information about the dose levels is retained after conversion. Brighter spots indicate higher dose levels.

Lastly, for the comparison view we needed a special type of colormap to highlight the differences between the two treatment plans. For this purpose we chose a divergent colormap. Divergent colormaps start and end with contrasting or opposite colors, while it transitions to a neutral color halfway. We chose a red and blue divergent colormap through the website ColorBrewer and applied it for the comparison screen [4]. The red tints indicate where the dose levels of the other plan exceed the dose levels of the reference plan. Likewise, The blue tints indicate where the dose levels of the reference plan exceed the dose levels of the other plan. The neutral white color indicate that there is little to no difference in dose levels between the two plans.

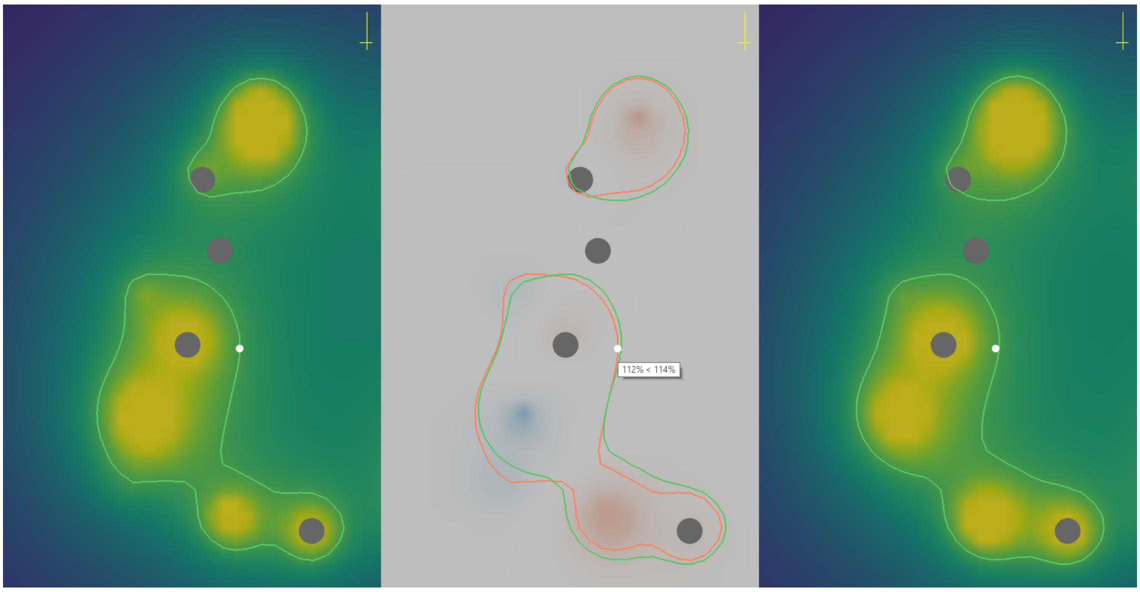


Figure 4.7: Red and blue tints indicate difference in dose levels, while the white color indicates no difference in dose levels.

4.2.3 Additional Elements

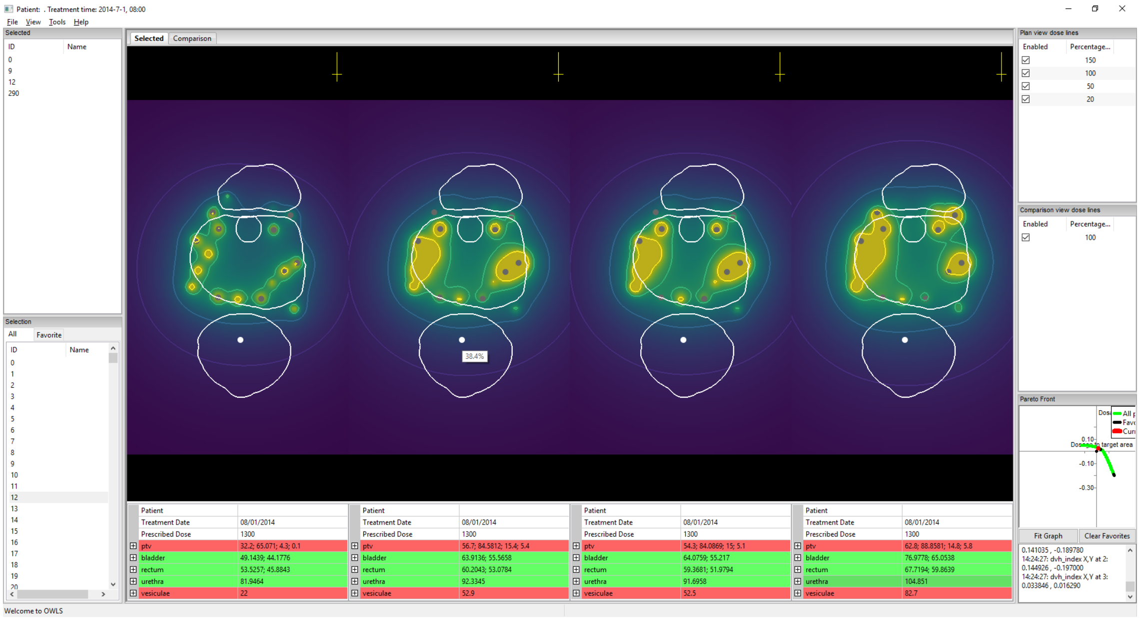
Other elements that are included in the visualization are elements related to the patient. These elements are:

- Visualization of organs, through white contours.
- Visualization of dwell points, indicated with round gray dots.
- Visualization of medical images, such as CT and MRI images.

For the final image, we use a form of pixel fusion. All elements are layered on top of each other in a certain order. Through this order the visualization maintains clarity while all elements are drawn. The first element to be rendered are the medical images. They provide the base for all other visualizations. Next are the colorwash heatmaps, followed by the organ contours and the isodose lines. The colorwash heatmaps are transparent, so that the medical images beneath remain visible. The final layer renders all the dwell points.

4.3 User Interface

The user interface was developed using wxWidgets, a C++ GUI library [25]. It provides the framework for our front-end implementation. The UI consists of several elements that provide the main functionalities of the program. The main window is the main frame that holds all other UI elements. Most of these elements can be moved and docked into different positions of the main window. They can also be detached from the main window and float in a separate window. We detail some of these elements in the following subsections.



Patient: , Treatment time: 2014-7-1, 08:00

Selected

ID	Name
0	
9	
12	
290	

Selection (All / Favorite)

ID	Name
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	

Plan view dose lines

Enabled	Percentage...
☑	150
☑	100
☑	50
☑	20

Comparison view dose lines

Enabled	Percentage...
☑	100

Patient	
Treatment Date	08/01/2014
Prescribed Dose	1300
⊞ ptv	32.2; 65.071; 4.3; 0.1
⊞ bladder	49.1439; 44.1776
⊞ rectum	53.5257; 45.8843
⊞ urethra	81.9464
⊞ vesiculae	22

Patient	
Treatment Date	08/01/2014
Prescribed Dose	1300
⊞ ptv	56.7; 84.5812; 15.4; 5.4
⊞ bladder	63.9136; 55.5658
⊞ rectum	60.2043; 53.0784
⊞ urethra	92.3345
⊞ vesiculae	52.9

Patient	
Treatment Date	08/01/2014
Prescribed Dose	1300
⊞ ptv	54.3; 84.0869; 15; 5.1
⊞ bladder	64.0759; 55.217
⊞ rectum	59.3681; 51.9794
⊞ urethra	91.6958
⊞ vesiculae	52.5

Patient	
Treatment Date	08/01/2014
Prescribed Dose	1300
⊞ ptv	62.8; 88.8581; 14.8; 5.8
⊞ bladder	76.9778; 65.0538
⊞ rectum	67.7194; 59.8639
⊞ urethra	104.851
⊞ vesiculae	82.7

```
0.141035 , -0.189780
14:24:27: dvh_index X,Y at 2:
0.144926 , -0.197000
14:24:27: dvh_index X,Y at 3:
0.033846 , 0.016290
```

Welcome to OWLS

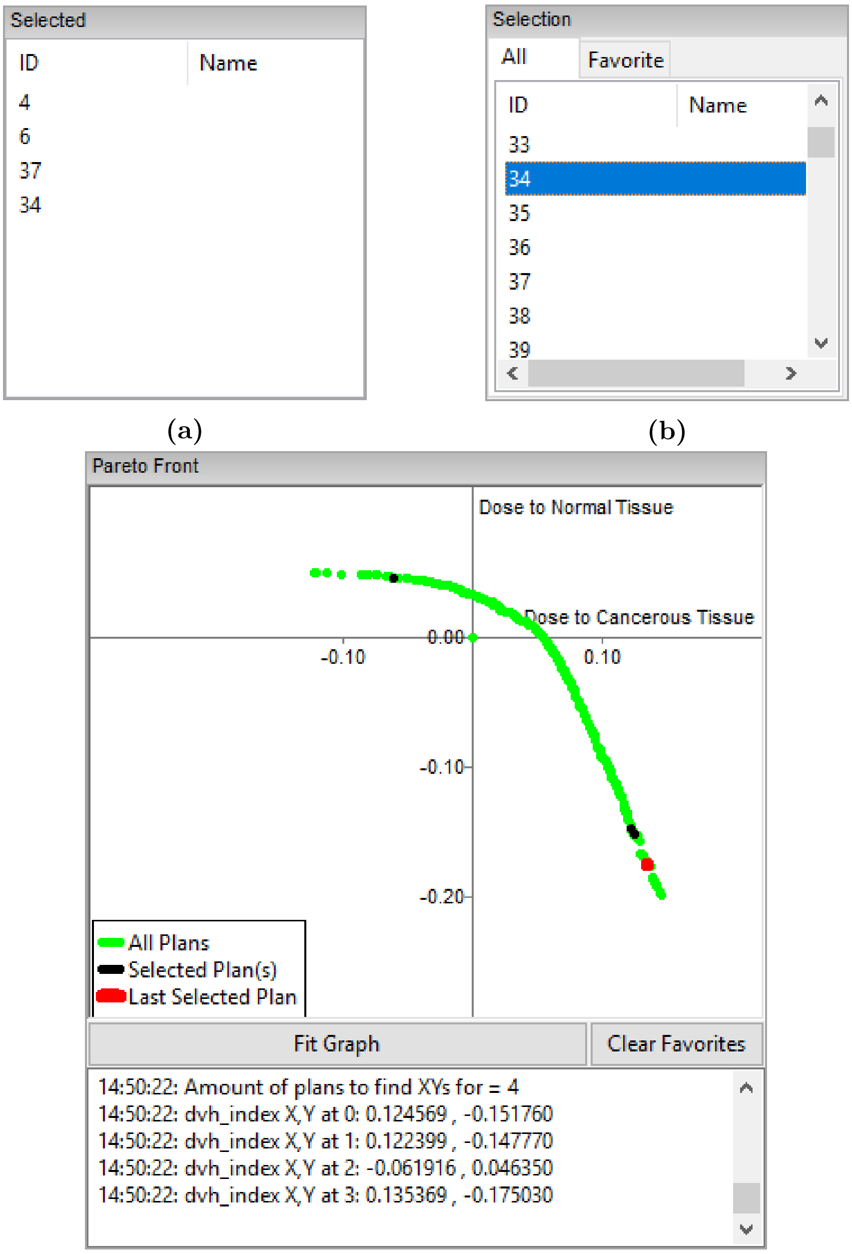
Figure 4.8: The main window contains all UI elements.

4.3.1 Plan Selection

There are several elements that can be used to select and manage plans. The first method is through a list view sorted by the plan ID, which we denote as the selection window. This window is on default docked on the bottom left.

The other method is through a Pareto front plot visualization, where each plan is plotted in the Pareto front. Each element of the Pareto front is a selectable plan that can be previewed or shown. In addition, the user can zoom in or select a region of the plot for further detail. This window is on default docked to the right. Interacting with the Pareto front visualization is how the client imagined the main way for this program to be used. As such, this feature was required to be included in the final build since the start of this project.

Another element related to plan management is a list view for selected plans. This list shows the plans currently selected for viewing. Plans can also be removed from view using this window. This window is on default docked above the selection window.



ID	Name
4	
6	
37	
34	

(a)

ID	Name
33	
34	
35	
36	
37	
38	
39	

(b)

```
14:50:22: Amount of plans to find XYs for = 4
14:50:22: dvh_index X,Y at 0: 0.124569 , -0.151760
14:50:22: dvh_index X,Y at 1: 0.122399 , -0.147770
14:50:22: dvh_index X,Y at 2: -0.061916 , 0.046350
14:50:22: dvh_index X,Y at 3: 0.135369 , -0.175030
```

(c)

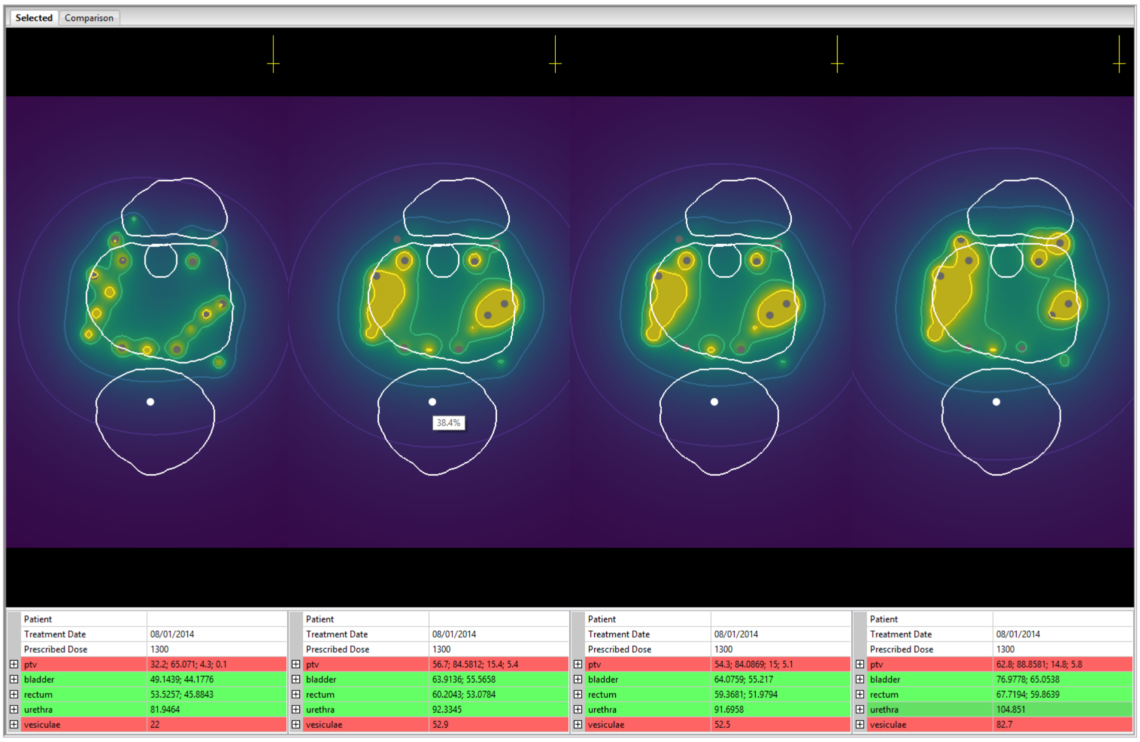
Figure 4.9: Overview of plan selection UI elements. (a) The list view for selected plans. (b) The list view for all loaded plans. (c) The pareto front visualization.

4.3.2 Multi-Plan View

The multi-plan view is used to show and visualize the details of the treatment plans. It is located in the visualization window, which is centered in the main window and cannot be moved. The multi-plan view is on display on default, but can later be accessed by clicking on the corresponding tab. The multi-plan view adds a visualization screen for each plan that has been selected. Newly selected plans will appear to the right of the plans that are already visualized. The corresponding plan properties are displayed beneath the visualization screen.

Each visualized plan are controlled concurrently, meaning that the user does not have to move or scroll through a plan one at a time. The position and location within the plans are all in sync. To improve the clarity of navigating through a plan, a reference mouse pointer is added for every visualized plan, indicated with a small white dot.

The multi-plan view provides physicians with a method for quick comparison without looking into much detail. The physician can observe the general details of each selected plan and can discard plans that are not to his liking. In addition, the DVH parameters shown in the table give an quick indication of the viability of the plan. When a parameter is highlighted with green, it means the parameter is within the given constraints. When the constraint is not met, the parameter is highlighted red.



Patient	
Treatment Date	08/01/2014
Prescribed Dose	1300
⊞ ptv	32.2; 65.071; 4.3; 0.1
⊞ bladder	49.1439; 44.1776
⊞ rectum	53.5257; 45.8843
⊞ urethra	81.9464
⊞ vesiculae	22

Patient	
Treatment Date	08/01/2014
Prescribed Dose	1300
⊞ ptv	56.7; 84.5812; 15.4; 5.4
⊞ bladder	63.9136; 55.5658
⊞ rectum	60.2043; 53.0784
⊞ urethra	92.3345
⊞ vesiculae	52.9

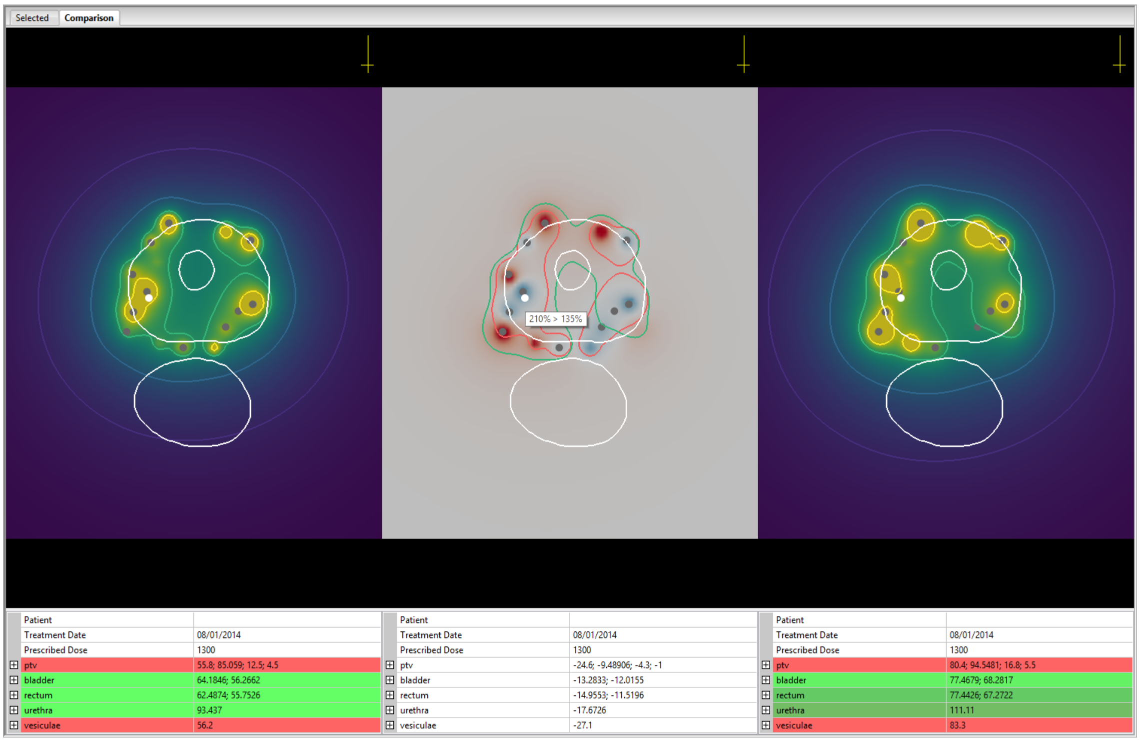
Patient	
Treatment Date	08/01/2014
Prescribed Dose	1300
⊞ ptv	54.3; 84.0869; 15; 5.1
⊞ bladder	64.0759; 55.217
⊞ rectum	59.3681; 51.9794
⊞ urethra	91.6958
⊞ vesiculae	52.5

Patient	
Treatment Date	08/01/2014
Prescribed Dose	1300
⊞ ptv	62.8; 88.8581; 14.8; 5.8
⊞ bladder	76.9778; 65.0538
⊞ rectum	67.7194; 59.8639
⊞ urethra	104.851
⊞ vesiculae	82.7

Figure 4.10: The multi-plan view shows all selected plans in a row and can be accessed by selecting the corresponding tab.

4.3.3 Comparison View

The comparison is used to compare two brachytherapy treatment plans in more detail. Like the multi-plan view, it is located in the visualization window and accessible through the corresponding tab. The comparison view takes a reference plan and compares it with the last selected plan. The reference plan is visualized and detailed in the left screen. The last selected plan will appear in the right screen. The corresponding plan properties are displayed beneath the visualization screen. The middle screen is a special comparison visualization screen, that shows the differences between the two plans through various elements, such as isodose lines and a specially made colorwash heatmap. Additionally, the table beneath the visualization screen that usually provides the properties of a plan, instead shows the difference between the two plan properties for each parameter.
The comparison view provides physicians with a method to compare two plans in greater detail. The comparison screen in the middle provides extra information that the physician can use to make a decision on the viability of the plan.



	Patient	
	Treatment Date	08/01/2014
	Prescribed Dose	1300
⊞	ptv	55.8; 85.059; 12.5; 4.5
⊞	bladder	64.1846; 56.2662
⊞	rectum	62.4874; 55.7526
⊞	urethra	93.437
⊞	vesiculae	56.2

	Patient	
	Treatment Date	08/01/2014
	Prescribed Dose	1300
⊞	ptv	-24.6; -9.48906; -4.3; -1
⊞	bladder	-13.2833; -12.0155
⊞	rectum	-14.9553; -11.5196
⊞	urethra	-17.6726
⊞	vesiculae	-27.1

	Patient	
	Treatment Date	08/01/2014
	Prescribed Dose	1300
⊞	ptv	80.4; 94.5481; 16.8; 5.5
⊞	bladder	77.4679; 68.2817
⊞	rectum	77.4426; 67.2722
⊞	urethra	111.11
⊞	vesiculae	83.3

Figure 4.11: The comparison view shows a reference plan on the left, the last selected plan to the right and a comparison screen that highlights the differences between the two plans.

4.3.4 Visualization Controls

Several elements provide control for the visualization aspect of a plan. Through the View menu, the user can enable or disable elements from the visualization. Elements like isodose lines, colorwash, organ contours and medical images can be configured in the view menu.

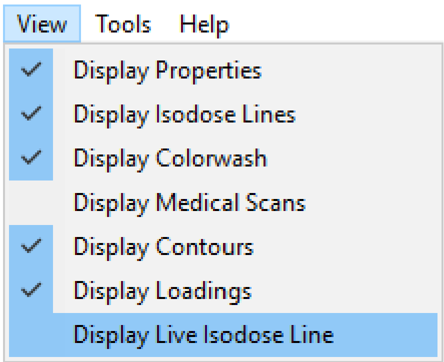
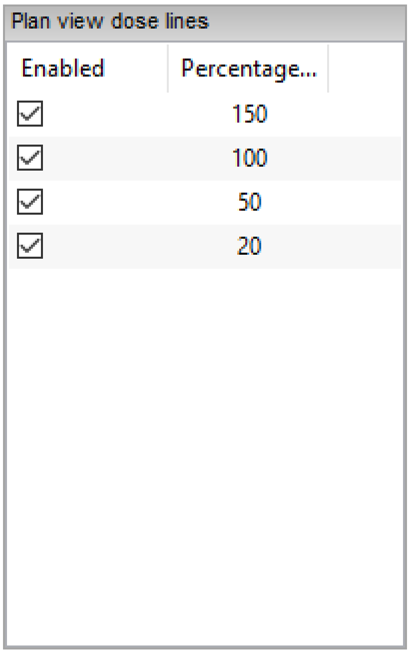


Figure 4.12: The view menu gives access to various elements of the visualization.

There are two UI elements that provide control for displaying and managing isodose lines. One controls the isodose lines in normal plan visualizations, while the other controls the isodose lines for the comparison screen in the comparison view, but otherwise function the same way. The element shows the list of isodose lines that are set to be visualized. Any newly added isodose line will be added to this list. In the left column, the isodose line can be enabled or disabled from the visualization. The isodose line can also be permanently removed if needed. The dose value of the isodose line can be edited by clicking the percentage number and typing in a new value.



Plan view dose lines

Enabled	Percentage...
☑	150
☑	100
☑	50
☑	20

Figure 4.13: The isodose lines control element. A seperate control element exists for isodose lines in the comparison screen.

4.4 Implementation Details

The program is implemented in C++. This decision was made because the BATS backend was written in C++ as well. Writing the program in C++ meant it was easier to connect the code and libraries together. The interface is implemented using the wxWidgets GUI library. The visualization of plans is implemented with the OpenGL graphics API. WxWidgets provided native OpenGL support, thus there were no additional difficulties in using the OpenGL libraries.
To draw the pareto front we used wxMathPlot, a library used to display 2D mathematical plots in wxWidgets [19]. For the compatibility with DICOM we used DCMTK, a collection of libraries that implements large parts of the DICOM standard. We were mostly interested in loading CT and MRI images stored using the DICOM standard. We used the testing framework “catch” to write unit tests for the classes that do not directly interact with the GUI.

5. Discussion

In this chapter we will discuss and evaluate the results of our program. While we are generally content with the application we delivered, there is still room for improvement in some areas of our program. We will list some points of contention and introduce solutions that could improve the application for future iterations. Lastly, we will discuss some of the ethical implications that come with the development of our program.

5.1 Evaluation

When we look at the requirements set at the start of the project, we can see that most of the requirements have been met. The plan selection system with a Pareto front visualization works as we intended, being able to quickly overview the different DVH parameters and selecting and previewing plans accordingly. The plan visualisation works according to expectations, including the isodose lines, colorwash heatmap functionality, MR images and live isodose lines. The comparison window includes a side-by-side overview and a comparison screen which shows the difference between the two plans. Reading patient files and DICOM files is also fully supported. Thus all the "must have" requirements are satisfied. We have furthermore satisfied most of the "should have" requirements.

Within the timeframe we failed to deliver some of the features we would like to have seen in the program. This is mostly in regards to navigation and other actions, such as undo and redo. We initially had a browser-like program in mind, where the user could use navigation buttons to navigate the program. These features didn't make it into the final build however, as we prioritized features that were of more importance.

Another major feature that we cut was visualizing a treatment plan in 3D. This happened for several reasons. Firstly, the complexity of this feature was beyond the scope of our project. We would like to have this feature implemented if we were given more time. Secondly, it also became clear over time that 3D visualization would be a nice to have feature, but not an essential feature. In our interviews with physicians and clinical staff, we noted that most of them never used the 3D plan visualizations provided by other medical applications. This feature would benefit some members of the research team however, who extensively use the 3D view.

The program also has features that are not fully implemented, or features that can be further optimized. For example, due to heavy calculations required for the dose distribution grid, the initial scrolling through the plan visualization is not a completely smooth experience. Also, due to this functionality not yet being available in the BATS back-end, the reading of a full DICOM folder, including the patient data and plan data, is not yet fully supported.

Since all of the essential features and most of the important features have been implemented we see the program as a success. Furthermore, the program has a modular structure meaning it is not difficult to implement new features or maintain current features. This enables the implementation of missing features in future iterations of this program.

5.2 Recommendations

During the development of this program some features came to light that would improve the program if it were implemented. In addition to all the "could haves" requirements that we recommend for future updates, we will discuss some additional recommendations next.

There is in general a lack of icon-based buttons in the application. Most control elements are delivered through text-based menus. The implementation of icon-based control buttons could provide users with an understanding of the button on a whim, as the saying ”a picture is worth a thousand words” often holds true. As mentioned earlier, the implementation of navigation buttons, such as a forward or a back button, could greatly benefit the workflow of the application. In addition to this, a history of visited plans would provide the user with a tool to quickly navigate to previously viewed plans.

While 3D visualizations were beyond the scope of our project, the idea of viewing a treatment plan from multiple planes (in 2D) would have been feasible to implement. As it is now, a plan can only be viewed from the top and can only be scrolled through the vertical axis. The addition of side and front views would benefit the visualization of a treatment plan by providing better context to the situation. This feature is also included in Oncentra Brachy, thus physicians that have worked with Oncentra Brachy should already be familiar with the idea of multi-plane views [9].

A physician we interviewed came up with the idea of a multi-comparison view. This feature would be an extension on the current comparison view. Instead of comparing two plans, the multi-comparison view would show multiple comparison screens and multiple plans for comparison. Our idea of this implementation would show the reference plan in full, while the comparison screens and the visualization of other plans stacked above each other. This feature would provide the user with detailed comparison of multiple plans instead of just two. We realize however that this feature could clutter the screen if too many plans are shown. Suffice to say a limit should be set as comparing more than four or five plans is a mentally impossible task even for physicians.

The last beneficial feature we would recommend is the support for multiple monitors. We know through interviews that physicians and researchers often have multiple screens at their disposal. If the program could adapt the use of multiple monitors, it would provide more workspace for the application and it could improve the workflow of the program. For example, the user could visualize multiple plans and spread them out on all the available monitors. To implement this, the visualization screens should be able to pop-out into a separate floating frames.

5.3 Ethical Implications

Ethical implications of medical software arise quickly. Since any mistake in this type of software could potentially lead to the hurt of an actual human being, a bug-free application is even more essential than in other fields. The European Union acknowledges this, and requires medical software that performs an action on data different from storage, archival, lossless compression, communication or simple search, to be CE-certified. The application also “administers or removes medicine, energy or other substance to or from the body”, namely radiation, thus would be classified as “medium to high-risk”. Therefore it has to be certified by a notified body to receive its CE mark [10, 18].

While dealing with sensitive information, the application has a couple of virtual safety nets behind it. First of all, it is an extension to existing software. The only thing it really does is visualize data that has been generated outside of its scope. All plans and all decisions a doctor makes within the application have to go through already certified medical software. This application is not designed to make decisions, the doctor still does. The application exists merely to assist a doctor, by highlighting differences, advantages, and disadvantages of pre-generated treatment plans.

A second ethical issue that might come up is the security of the patient data. This will not change from the current situation when this application would be used in a hospital. No internet connection is required, only a local connection to the hospitals database. This database should already exist and be secured up to high standards.

6. Conclusion

In this chapter we will answer the research question that we stated in the introduction, using the subquestions we formulated and the knowledge we gained during the research and development of the project.
We formulated the following research question: How can we aid a doctor in choosing an as optimal plan from multiple brachytherapy treatment plans using a front-end GUI? The following subquestions were formulated to find a proper answer to this question:

- How can we relay information about the patient and brachytherapy plans within our software?
- How can we streamline the general procedure of selecting brachytherapy treatment plans with the use of our software?

Physicians can perceive through isodose lines and DVH parameters whether a plan is suitable or optimal for a patient. Both are available in the application once a plan is selected and loaded. We added additional methods for communicating information through additional visualizations. We implemented colorwash heatmaps, to visualize dose distribution of a plan. This can communicate additional information about the dose distribution in a summarized manner. Another feature we added is the comparison view. The comparison view highlight differences through visualization and DVH parameter differences. This view can effectively communicate the differences between two plans.
To streamline the procedure we must look at the current procedure. The current process of plan selection is based on an individual plan-by-plan approach. Every plan must be individually looked at by the physician and discarded until a plan is found that is optimal. When the amount of plans reach in the thousands, this can be a time consuming task.
An important initial step for the selection between a huge number of plans would be a selection based on the set Pareto objectives. For example, if the dose level around normal tissue is too high, the plan is discarded. Likewise, if the dose levels on cancerous tissue are too low, the plan will also be discarded. In order to overview the Pareto objectives for all the plans concurrently, putting them into a graph where each plan is individually selectable would be the solution. The plans result in a Pareto front, a curve where the optimal mid-region of plans is easily perceivable. From this point on the number of selected plans can be severely cut down, thus making the next selection step easier.
For the next selection we have chosen for two types of plan comparison. The first type is an overview of multiple individual plans, where the plans are displayed next to each other and can be simultaneously scrolled through slice-by-slice. This approach works best when viewing more than two plans simultaneously. The second type of plan comparison is for comparing exactly two plans. It consists of a side-by-side view, and a comparison view in between. In this comparison view the relative differences between the two plans are visualized. After comparing multiple plans side-by-side and discarding suboptimal plans, a best plan can be chosen by the physician.

Bibliography

[1] R. Acosta. Computing Isodose Curves for Radiotherapy Treatment Plans. *Computing*, 2005. URL `http://digitalcommons.trinity.edu/compsci{_}honorshttp://digitalcommons.trinity.edu/compsci{_}honors/8`.

[2] D. Baltas and N. Zamboglou. 2D and 3D Planning in Brachytherapy. In *New Technologies in Radiation Oncology*, pages 237–254. doi: 10.1007/3-540-29999-8_19.

[3] P. Bosman. Visualisation, comparison, and selection of brachytherapy plans, 2017. URL `https://bepsys.herokuapp.com/projects/view/275`.

[4] C. Brewer and M. Harrower. ColorBrewer: Color Advice for Maps, 2016. URL `http://colorbrewer2.org/`. Accessed: June 2017.

[5] K. Bühler. From Many to One Medical Image Fusion, October 2015. URL `http://isgwww.cs.uni-magdeburg.de/miccai-vis/katja_buehler.pdf`. Accessed: 16 May 2017.

[6] X. Dai and S. Khorram. Data fusion using artificial neural networks: A case study on multitemporal change analysis. *Computers, Environment and Urban Systems*, 23 (1):19–31, 1999.

[7] DICOM Standards Committee. DICOM Brochure. Brochure, 1993. URL `http://dicom.nema.org/dicom/geninfo/Brochure.pdf`. Accessed: 10 May 2017.

[8] J. Dong, D. Zhuang, Y. Huang, and J. Fu. Advances in multi-sensor data fusion: Algorithms and applications. *Sensors*, 9(10):7771–7784, 2009.

[9] Elekta. Oncentra® brachy comprehensive treatment planning for brachytherapy. Brochure, 2014. URL `https://www.elekta.com/dam/jcr:0361c151-4eb5-4017-a5b6-6a35c4398671/Oncentra%C2%AE%20Brachy%20brochure.pdf`.

[10] European Union Compliance and Enforcement group. Guidance notes for manufacturers of class 1 medical devices, 2009. URL `http://ec.europa.eu/consumers/sectors/medical-devices/files/guide-stds-directives/notes-for-manufacturers-class1-09_en.pdf`.

[11] W. O. Galitz. *The Essential Guide to User Interface Design: an Introduction to GUI Design Principles and Techniques.* John Wiley & Sons, 2007.

[12] A. Gerbaulet. The GEC-ESTRO Handbook of Brachytherapy. 2002.

[13] C. R. Hildreth. The GUI OPAC: Approach with caution. *Public Access-Computer Systems Review*, 6(5), 1995.

[14] J. Hobart. Principals of Good GUI Design. *Classic System Solutions, Inc.*, October 1995.

[15] K. Lawonn, N. N. Smit, B. Preim, and A. Vilanova. Illustrative Multi-volume Rendering for PET/CT Scans. In *VCBM*, pages 103–112, 2015.

[16] C. Maple. Geometric design and space planning using the marching squares and marching cube algorithms. In *Proceedings - 2003 International Conference on Geometric Modeling and Graphics, GMAG 2003*, pages 90–95. IEEE Comput. Soc, 2003. ISBN 0769519857. doi: 10.1109/GMAG.2003.1219671.

[17] G. Mistelbauer, A. Köchl, R. Schernthaner, I. Baclija, R. Schernthaner, S. Bruckner, M. Sramek, and M. E. Gröller. Smart super views—A knowledge-assisted interface for medical visualization. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 163–172. IEEE, 2012.

[18] Nictiz. Medical apps, is certification required? Brochure, 2013. URL `https://www.nictiz.nl/SiteCollectionDocuments/Infographics/Infographic%20medical%20apps.pdf`.

[19] D. Rondini. wxMathPlot, June 2007. URL `http://wxmathplot.sourceforge.net/`. Accessed: June 2017.

[20] N. Schubert and I. Scholl. Comparing GPU-based multi-volume ray casting techniques. *Computer Science-Research and Development*, 26(1):39–50, 2011.

[21] The National Eye Institute. Facts About Color Blindness, February 2015. URL `https://nei.nih.gov/health/color_blindness/facts_about`. Accessed: June 2017.

[22] M. van der Meer. Prostate HDR Brachytherapy. Presentation, 2017.

[23] S. van der Walt and N. Smith. mpl colormaps, 2015. URL `http://bids.github.io/colormap/`.

[24] D. van Heesch. Doxygen: Main page, 2017. URL `http://www.stack.nl/~dimitri/doxygen/`. Accessed: 10 May 2017.

[25] wxWidgets Team. wxWidgets: Cross-Platform GUI Library, 2017. URL `https://www.wxwidgets.org`. Accessed: 10 May 2017.

A. Project Information

The following project description was provided by Bosman [3].

A.1 Project Description

More than one in three people in The Netherlands will develop cancer. Being well-equipped to treat cancer effectively and ensure the highest quality of life after treatment, is therefore of extreme societal relevance and value. Radiation therapy, or radiotherapy, is an important and often-used modality for treating cancer whereby ionizing radiation is used to target cancer cells. To administer radiotherapy, a plan must be created that describes how radiation sources should be used to establish a favorable radiation dose distribution in the body. A question that naturally arises, is "What is the best radiation treatment plan for me?", much like the everyday question "What is the best route to my destination?" that we ask our car navigation systems. The answer to the latter question enabled the worldwide innovation of vehicle navigation systems. This answer is rooted in fundamental mathematics and computer science research on graph theory and is provided quickly and reliably by modern ICT hardware. Answering the former question is however much harder because a truly ideal radiation treatment plan that only affects cancer cells but not surrounding healthy cells, is not physically realizable due to real-world constraints. This leads to many difficulties when designing radiation plans using software, resulting in great variation between plans designed by different medical experts.

A novel multi-objective optimization-based approach has been proposed by a consortium of researchers of Centrum Wiskunde & Informatica (CWI) and the Academisch Medisch Centrum (AMC), both located in Amsterdam, The Netherlands. A multi-objective optimization approach aims to provide multiple solutions at once that represent high quality trade-offs between the objectives of importance (in this case, generally speaking, the irradiation of cancer cells and the irradiation of healthy cells). This novel approach has the potential to innovate clinical practice by providing important insight into the most important trade-offs to be considered when choosing a final treatment plan, ultimately allowing to design better plans faster than currently possible. The customer in this project is CWI, represented by Dr. Peter A.N. Bosman, who is the project leader on a currently running joint research project with AMC and the company Elekta. In this project, amongst others, the aforementioned multi-objective optimization approach is being developed.

For translational results (i.e., feedback to/from clinical practice), having high-quality, intuitive software tools are crucial. However, the vast majority of the focus of the involved personnel on the project is on the design of fundamental, novel algorithms and on obtaining novel scientific research results. There is therefore a clear need for a subproject aimed at the design and development of software tools that allow to present and navigate a collection of trade-off plans found by the optimization software to the users (i.e., medical experts of AMC) in an insightful manner. This includes visualizations of medical images (i.e., 3D CT/MRI scans), overlaid with insightful heatmaps of dose distributions pertaining to individual plans, as well as visualizations of differences between plans. It further requires efficient navigation of a set of plans as well as selection of plans supported by fast and easy access to different views on trade-offs between different indicators of interest (such as dose delivered to specific organs). Finally, being able to insightfully make additional small changes to selected treatment plans through interaction with a graphical user interface would be highly valuable.

Although medically approved commercial software exists to display treatment plans, the research team has no direct possibility to incorporate their own developments in such software, making crucial feedback interactions with clinical personnel difficult. Moreover, insightfully showing what the differences are between two (or more) plans is not supported in commercial software. Yet, this is of key importance, especially considered within the bigger picture of efficiently navigating a SET of multiple treatment plans from which ultimately one plan must be selected to be delivered to a patient.

A key challenge is to make a front-end / graphical user interface that achieves the aforementioned in a manner that makes such a final selection as easy and insightful as possible while being fast and efficient. Moreover, since the fundamental algorithmic research environment is Linux-based (coded in C and/or C++), but clinical practice is Microsoft Windows-based, it is key to develop such a front-end in a manner that is platform-independent. In addition, to support future research and future adaptations, understanding how users interact with the front-end system is key. To this end, a further design requirement is a "Big Brother is Remembering you" feature through which the actions of a user can be recorded and stored in a database in a fashion that is easy to analyse later. What the best choices are in meeting these requirements is an open question to be solved by the students.

Finally, accepting this project requires the students to agree that all developed software will remain with CWI. Moreover, students will agree that any and all (patient) data that students will have access to, must remain on computers at AMC and/or CWI and shall not be copied to outside these institutes, unless explicitly allowed by project leader Peter Bosman.

A.2 Company Description

CWI (Centrum Wiskunde & Informatica) is The Netherlands' national research institute for mathematics and computer science. At CWI, our strength is discovering and developing the new ideas that benefit society and the economy, as well as other scientific areas. Our research is rooted in practical, real-life questions and explores essential aspects of modern life, including transport and communication networks, internet security, medical imaging, and smart energy systems. Our innovations form an integral part of numerous software products, programming languages, and international standards. The AMC is more than just a hospital. It is a world on its own where it is always about the people. Patient care, education, and research are integrated. This results in a world of possibilities for all talented people that are looking for a place to excel.

B. **Project Plan**

B.1 Timeline

Week (Date)	**Task description**
Week 1 (24/4)	Looking for coach, project introduction at AMC on Friday
Week 2 (1/5)	Getting given software working, choosing framework, importing files
Week 3 (8/5)	Writing project plan, implement basic patient and plan overview, OpenGL canvas, DICOM research
Week 4 (15/5)	2D contour and heatmaps for plans, first comparison view, start on DICOM reading
Week 5 (22/5)	DICOM reading implementation, finish 2D overviews, implement favorites, back, forward, and home buttons
Week 6 (29/5)	DICOM writing implementation, overlayed comparison view
Week 7 (5/6)	Implement save state for application. Opened plans and favorites are saved when program is closed. Record recent files. Start on advanced plan selection method.
Week 8 (12/6)	Advanced plan selection method: pareto front? Also, research and start on 3D visualization.
Week 9 (19/6)	Implement 3D visualization, writing report.
Week 10 (26/6)	Finalizing project, writing report.
Week 11 (3/7)	Final presentations

Table B.1: Project timeline as set at the start of the project

B.2 Software and Implementation

The code provided to us by our client was written in C++. We decided to use the same language for our project to provide better synergy with their existing code. Another advantage of this language is that it compiles on both Windows and Linux. A logical choice for IDE was Microsofts own Visual Studio. Since the 2017 version was not supported by some libraries, and the client uses the 2015 version, we decided to use the 2015 version as well. Multi-platform support was leading in the choice for a GUI library. Several libraries were looked at, but the final choice was wxWidgets, that supports Windows and Linux [25]. It also has built-in OpenGL support, which makes drawing 2D and 3D images possible.

B.3 Documentation and User Manual

In order to document our code, a tool called DoxyGen is used [24]. This tool generates documentation from our comments, in the form of HTML. A general overview will be given by UML files, one for the GUI and one for our data structures. A user manual will be written in the form of a README.md on the Git repository. This way we ensure that our client will be able to actually use our code properly. We will not write a manual for user of the program, as it should be easy and self-explanatory enough for users to understand without the need for an external explanation.

B.4 Testing and Validation

The “catch” Testing Framework for C++ is used to write unit tests. Since GUI is impossible to test using automated testing tools, the classes implementing this will be ignored in unit testing. They will be manually checked for errors.

B.5 Non-Disclosure Agreement

Due to the sensitive nature of the code used (the BATS program is not officially released yet), and the (anonymized) patient data, public sharing services cannot be used for code. This includes but is not limited to all Google services, WhatsApp, Dropbox, and GitHub. Therefore, a private git server was set up to provide a service for version control. After the end of the project, all data and code is handed over to the client, and all data and code on the developers side is destroyed. This has been agreed upon by both the client and the developers, a contract has been signed by both parties.

C. SIG Evaluation

The following feedback was provided by Software Improvement Group (SIG).

C.1 First Feedback

De code van het systeem scoort 3 sterren op ons onderhoudbaarheidsmodel, wat betekent dat de code gemiddeld onderhoudbaar is. De hoogste score is niet behaald door een lagere scores voor Unit Size en Duplication.

Voor Unit Size wordt er gekeken naar het percentage code dat bovengemiddeld lang is. Het opsplitsen van dit soort methodes in kleinere stukken zorgt ervoor dat elk onderdeel makkelijker te begrijpen, te testen en daardoor eenvoudiger te onderhouden wordt. Binnen de langere methodes in dit systeem, zoals bijvoorbeeld de 'compute'-methode, zijn aparte stukken functionaliteit te vinden welke ge-refactored kunnen worden naar aparte methodes. Commentaarregels zoals bijvoorbeeld 'find the roi belonging to this dose object' zijn een goede indicatie dat er een autonoom stuk functionaliteit te ontdekken is. Het is aan te raden kritisch te kijken naar de langere methodes binnen dit systeem en deze waar mogelijk op te splitsen.

Voor Duplication wordt er gekeken naar het percentage van de code welke redundant is, oftewel de code die meerdere keren in het systeem voorkomt en in principe verwijderd zou kunnen worden. Vanuit het oogpunt van onderhoudbaarheid is het wenselijk om een laag percentage redundantie te hebben omdat aanpassingen aan deze stukken code doorgaans op meerdere plaatsen moet gebeuren.

Bij jullie project bestaat de duplicatie echter niet uit code, maar uit data in colorMap.cpp. Dit is natuurlijk geen typisch bestand, maar over het algemeen is het verstandig om code en data te scheiden. Je zou deze data beter in een apart bestand kunnen plaatsen en dit bestand vervolgens in de code in te laden.

Als laatste nog de opmerking dat er geen (unit)test-code is gevonden in de code-upload. Het is sterk aan te raden om in ieder geval voor de belangrijkste delen van de functionaliteit automatische tests gedefinieerd te hebben om ervoor te zorgen dat eventuele aanpassingen niet voor ongewenst gedrag zorgen.

Over het algemeen scoort de code dus gemiddeld, hopelijk lukt het om dit niveau nog wat te laten stijgen tijdens de rest van de ontwikkelfase.

D. **Info Sheet**

Title
Visualization, Comparison, and Selection of Brachytherapy Plans

Organization(s)
Centrum Wiskunde & Informatica
Academisch Medisch Centrum

Presentation Date
6 July 2017

Description
A research team consisting of researchers from Centraal Wiskunde & Informatica (CWI) and Academisch Medisch Centrum (AMC) have developed a solution that generates a wide range of near-optimal brachytherapy treatment plans for treating prostate cancer, which they have named the brachy automated treatment-planning system (BATS). However, the generated plans can currently not be viewed in any tangible way.

For this we have developed a front-end solution that works in tandem with BATS that provides the necessary requested features. Multiple plans can be viewed and compared through visualization and plan properties. Additionally, the application has DICOM compatibility and can operate on Windows and Linux operating systems.

While not all requirements are met, we are satisfied with the resulting application. Recommendations for future iterations include improved navigation and control through icon-based buttons, and more interaction options through hotkeys. Additional 3D treatment plan visualizations and multi-plane views would provide opportunities for 3D planning.

Team Members

- Name: Jasper van Esveld
  Role: Developer
  Contributions: System Design and All-round Contributor to various Features.
- Name: Luke Prananta
  Role: Developer and Administrative Position
  Contributions: User Interface and Visualization of Plans.
- Name: Matthias Tavasszy
  Role: Developer
  Contributions: DICOM Compatibility and various Visualization Features.
- Name: Arjan van Schendel
  Role: Developer
  Contributions: DICOM Compatibility and Pareto Front Visualization.

Contact Details

- Client Advisor:
  Dr. P.A.N. Bosman (Peter.Bosman@cwi.nl)
- TUDelft Supervisor(s):
  Dr. K.A. Hildebrandt (K.A.Hildebrandt@tudelft.nl)
  N.H.L.C de Hoon (N.H.L.C.deHoon@tudelft.nl)