

Interaktive Co-Registrierung für multimodale Krebs-Bilddaten basierend auf Segmentierungsmasken

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Medizinische Informatik

eingereicht von

Tanja Eichner, B.Sc.

Matrikelnummer 11911100

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller

Mitwirkung: Assoc.Prof. Noeska Smit, PhD

Dipl.-Ing. Dipl.-Ing. Eric Mörth

Wien, 17. Mai 2022

Tanja Eichner, B.Sc.

Eduard Gröller

Interactive Co-Registration for Multi-Modal Cancer Imaging Data based on Segmentation Masks

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Medical Informatics

by

Tanja Eichner, B.Sc.

Registration Number 11911100

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller

Assistance: Assoc.Prof. Noeska Smit, PhD
Dipl.-Ing. Dipl.-Ing. Eric Mörth

Vienna, 17th May, 2022

Tanja Eichner, B.Sc.

Eduard Gröller

Erklärung zur Verfassung der Arbeit

Tanja Eichner, B.Sc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 17. Mai 2022

Tanja Eichner, B.Sc.

Acknowledgements

First of all, I would like to thank my supervisors Eric Mörth, Noeska Smit and Eduard Gröller. Thank you so much for making my stay in Norway possible. I especially wish to thank Eric for his continuous support and enthusiasm throughout this whole thesis project, especially for helping me out and showing me new possible directions whenever needed. You are a great inspiration and I learned a lot working with you! I also want to thank Noeska Smit for mentoring my project in Bergen. Thank you for inviting me again to the research team, for sharing your passion for visualization, for your great and helpful ideas, and your understanding character. Furthermore, I am very thankful to Eduard Gröller for his support throughout my whole Master's studies in Vienna. Thank you for introducing me to the field of computer graphics and visualization, for sharing your inspirational stories, and for always supporting and helping me with my exchange plans.

I also want to thank Ingfrid Haldorsen and the whole MMIV team at the Haukeland University Hospital in Bergen. It was a pleasure to be part of this amazing team. Working in such an interdisciplinary environment was so motivating and broadened my horizon. A special thanks goes to Margherita and Lucia for not only being colleagues, but becoming real friends I can count on.

Furthermore, I wish to thank Ingrid for being such a wonderful roommate and for making me feel so much at home here. Thank you also to my friends that made my time in Bergen unforgettable. Thank you to Bruno for all the immense support, David (thanks for all the Gran Ca cheese), Iris, Ermes, my Harstad gang Ole and Torkjel, Katrine and Sofie, Anne Kari, Matteo and Erik. Thank you for the beautiful memories that I will always carry with me.

Even though COVID-19 restricted life during most of my Master's studies, I had a fantastic time studying in Vienna, especially thanks to my roommates Nati, Kathi and Kathi, as well as Elli, Alicia, Marian, Harry, Lena, Ambroise and Lorenzo. Despite the distance between us, I am very grateful to have my primary school friends Carina and Jana by my side, as well as Sophie, Julian and Mario who know how to cheer me up, especially in difficult times.

Most of all, I want to thank my parents Heidi and Werner and my sister Tina for always believing in me, supporting and encouraging me to pursue my dreams, and visiting me in every place I choose to stay. Thank you! Without you, I would have never come that far.

Danksagung

Zuallererst möchte ich meinen Betreuern Eric Mörth, Noeska Smit und Eduard Gröller danken. Vielen Dank, dass ihr mir den Aufenthalt in Norwegen ermöglicht habt. Ein besonders großes Dankeschön geht an Eric für die Betreuung und kontinuierliche Unterstützung während der gesamten Masterarbeit. Danke, dass du mir geholfen und immer wieder neue Wege aufgezeigt hast! Mein Dank gilt auch Noeska Smit für die Betreuung meiner Arbeit in Bergen. Danke, dass du mich in das Forschungsteam eingeladen und deine Leidenschaft für Visualisierung mit mir geteilt hast. Danke für deine großartigen und hilfreichen Ideen und deine verständnisvolle, empathische Art. Außerdem bin ich Eduard Gröller für seine Unterstützung während meines gesamten Masterstudiums in Wien sehr dankbar. Danke, dass du mich in das Gebiet der Computergrafik und Visualisierung eingeführt und mich bei meinen Austauschplänen immer unterstützt hast.

Ich möchte auch Ingrid Haldorsen und dem gesamten MMIV-Team am Haukeland University Hospital in Bergen danken. Es war eine Ehre, Teil dieses großartigen Teams sein zu dürfen. Die Arbeit in einem so interdisziplinären Umfeld war sehr motivierend und hat meinen Horizont erweitert. Mein besonderer Dank gilt Margherita und Lucia, die nicht nur Kolleginnen waren, sondern auch echte Freundinnen geworden sind.

Außerdem möchte ich Ingrid dafür danken, dass sie eine so wunderbare Mitbewohnerin ist und ich mich hier so zuhause fühlen kann. Vielen Dank auch an meine Freunde, die meine Zeit in Bergen unvergesslich gemacht haben. Danke an Bruno, David (danke für den ganzen Gran Ca Käse), Iris, Ermes, meine Harstad-Gang Ole und Torkjel, Katrine und Sofie, Anne Kari, Matteo und Erik. Danke für die vielen schönen Erinnerungen, die mich immer begleiten werden.

Obwohl COVID-19 das Leben während des Großteils meines Masterstudiums stark beeinflusst hat, hatte ich eine wundervolle Zeit in Wien. Der Grund dafür waren vor allem meine Mitbewohnerinnen Nati, Kathi und Kathi, sowie Elli, Alicia, Marian, Harry, Lena, Ambroise und Lorenzo. Trotz der Entfernung zwischen uns bin ich sehr dankbar, auch meine Grundschulfreundinnen Carina und Jana an meiner Seite zu haben. Danke auch an Sophie, Julian und Mario, die mich besonders in schwierigen Zeiten immer aufmuntern.

Am allermeisten möchte ich meinen Eltern Heidi und Werner und meiner Schwester Tina danken. Danke, dass ihr immer an mich glaubt, mich unterstützt und ermutigt, meine Träume zu verfolgen. Danke, dass ihr mich an jedem möglichen Ort besucht, egal wo es mich hinverschlägt. Ohne euch wäre ich nie so weit gekommen.

Abstract

Endometrial cancer is the most common and most lethal gynecologic malignancy worldwide. Multiple MRI sequences are acquired per patient in gynecologic cancer research because they reveal different tissue characteristics. Radiomic tumor profiling extracts features from medical imaging data aiming to find new tumor imaging biomarkers. Co-registration and tumor segmentation of multi-sequential MRI data build the base for radiomic tumor profiling. Many approaches exist that aim to automate these time-consuming manual processes. After automatic co-registration, volumes are often still misaligned. This lack of registration quality has an impact on the results of radiomic tumor profiling, since we cannot ensure voxel integrity.

We distinguish between rigid and deformable registration. Rigid registration transforms a volume using only translation and rotation parameters, while deformable registration can include local deformations. Tumors are rigid structures compared to the tissue around them. Therefore, rigid co-registration can be sufficient to align tumors. However, to analyze also surrounding structures, deformable registration is necessary. Even though tumors are rigid structures, they can appear slightly different in the varying sequences due to imaging physics. Applying deformable registration to the whole image can result in tumor deformations that do not resemble the underlying biological tissue characteristics and can alter important information about tumor tissue characteristics.

To address these two problems, we propose the web-based application MuSIC (Multi-Sequential Interactive Co-registration). The tool allows medical experts to co-register multiple sequences simultaneously based on a pre-defined segmentation mask that has been generated for one of the sequences. In our workflow, a simulated-annealing-based shape matching algorithm searches for the tumor position in each sequence that can vary in translation and rotation parameters. We present the updated segmentation positions to the user, who can interactively adapt the positions if needed. We include multi-modal visualization techniques for visual quality assessment during this procedure. Based on the positioning of the segmentation masks, we register the volumes. We allow for both rigid and deformable co-registration. Due to our approach based on segmentation masks, we apply local transformations mainly outside the tumor tissue in deformable registration.

We evaluate our approach in a usability analysis with medical and machine learning experts. They find the tool very intuitive and especially the medical experts clearly see themselves using MuSIC in the future.

Kurzfassung

Endometriumkarzinom ist weltweit die häufigste und tödlichste gynäkologische Malignität. In der gynäkologischen Krebsforschung werden pro Patientin mehrere MRT-Sequenzen aufgenommen, da sie unterschiedliche Gewebemerkmale erkennen lassen. Bei der radiomischen Tumorprofilierung werden Merkmale aus medizinischen Bildgebungsdaten extrahiert, um neue bildgebende Biomarker zu finden. Die Co-Registrierung und Tumorsegmentierung von multisequenziellen MRT-Daten bilden die Grundlage für die radiomische Tumorprofilierung. Es gibt viele Ansätze, die darauf abzielen, diese zeitaufwändigen manuellen Prozesse zu automatisieren. Auch nach der automatischen Co-Registrierung sind die Volumina oft nicht ausreichend gut ausgerichtet. Die mangelhafte Qualität der Registrierung wirkt sich auf die Ergebnisse der radiologischen Tumorprofilierung aus, da die Integrität der Voxel nicht sichergestellt werden kann.

Wir unterscheiden zwischen rigider und deformierbarer Registrierung. Bei der rigiden Registrierung wird ein Volumen nur mit Hilfe von Translations- und Rotationsparametern transformiert, während bei der deformierbaren Registrierung auch lokale Verformungen möglich sind. Im Vergleich zu umliegendem Gewebe sind Tumore rigide Strukturen. Wenn ausschließlich der Tumor im Fokus der Registrierung steht, kann daher eine rigide Co-Registrierung ausreichend sein. Um jedoch auch die umgebenden Strukturen zu analysieren, ist eine verformbare Registrierung erforderlich. Obwohl Tumore starre Strukturen sind, können sie in den verschiedenen Sequenzen aufgrund der Bildgebungsphysik leicht unterschiedlich aussehen. Daher kann die Anwendung der deformierbaren Registrierung auf das gesamte Bild zu Tumorverformungen führen, die nicht den zugrunde liegenden biologischen Gewebecharakteristika entsprechen und wichtige Informationen über die Eigenschaften des Tumorgewebes verändern können.

Um diese beiden Probleme zu lösen, schlagen wir die webbasierte Anwendung MuSIC (Multi-Sequential Interactive Co-Registration) vor. Die Applikation ermöglicht die gleichzeitige Co-Registrierung mehrerer Sequenzen auf der Grundlage einer vordefinierten Segmentierungsmaske, die für eine der Sequenzen generiert wurde. In unserem Arbeitsablauf sucht ein auf Simulated Annealing basierender Shape-Matching-Algorithmus in jeder Sequenz nach der Tumorposition, die sich in Translation und Rotation unterscheiden kann. Anschließend präsentieren wir die gefundenen Positionen der Segmentierungsmasken für jede Sequenz dem Nutzer, welcher die Position der Segmentierungsmaske, falls nötig, interaktiv adaptieren kann. Wir inkludieren multimodale Visualisierungstechniken zur vi-

suellen Qualitätsbeurteilung während des interaktiven Registrierungsprozesses. Basierend auf der Platzierung der Segmentierungsmasken registrieren wir die Volumina. Wir ermöglichen sowohl eine rigide als auch eine deformierbare Co-Registrierung. Durch unseren Segmentierungsmasken-basierten Ansatz werden bei der deformierbaren Registrierung lokale Transformationen primär im Gewebe außerhalb des Tumors angewendet.

Wir evaluieren unseren Ansatz in einer Usability-Analyse mit Experten aus den Bereichen Medizin und maschinellem Lernen. Die Teilnehmer finden unsere Anwendung sehr intuitiv und vor allem die medizinischen Experten können sich gut vorstellen MuSIC in Zukunft zu nutzen.

Contents

Abstract	xi
Kurzfassung	xiii
Contents	xv
1 Introduction	1
1.1 Motivation & Problem Statement	1
1.2 Aim of the Work	4
1.3 Contributions	5
1.4 Outline	5
2 Medical and Technical Background	7
2.1 Endometrial Cancer	7
2.2 Data Set	11
2.3 Radiomic Tumor Profiling	13
2.4 Segmentation	14
3 Related Work	19
3.1 Registration Algorithms and Components	19
3.2 Trade-off between Automation and User Interaction	24
3.3 Evaluation	29
4 MuSIC	35
4.1 Automatic Shape Matching	36
4.2 Interactive Shape Matching	47
4.3 Co-Registration	54
5 User Interface	59
6 Implementation	69
6.1 Frontend	69
6.2 Backend	76
	xv

7 Case Studies	79
7.1 Shape Matching	79
7.2 Registration	82
8 Usability Analysis	93
8.1 Evaluation Procedure	95
8.2 Evaluation Results	96
9 Discussion & Conclusion	99
9.1 Summary	99
9.2 Discussion	100
9.3 Future Work	102
9.4 Conclusion	103
List of Figures	107
List of Tables	109
List of Algorithms	109
Acronyms	111
Bibliography	113

Introduction

1.1 Motivation & Problem Statement

Cancer is, after cardiovascular diseases, the second most common cause of death in the world, and millions of people are diagnosed with malignancies each year [BFS⁺18, SEJ⁺16]. Among gynecologic diseases, endometrial cancer is the most frequent malignancy, and it causes most cancer deaths worldwide [HST16, HSG⁺14, MLC⁺16]. Medical imaging aims to support diagnosis and treatment planning for malignant diseases and has led to significant advances in these two tasks. Therefore, imaging technology has evolved immensely in the last years. In case of cancer, medical imaging data often consists of scans of different modalities, like Computed Tomography (CT) or Magnetic Resonance Imaging (MRI), per patient and examination date, depending on the tissue and tumor type. This variety of medical images allows for the examination of varying tumor characteristics that become visible in the different modalities [LSBP18, RvdHD⁺15].

A relevant application of multiple MRI scans per patient in radiology research is radiomic tumor profiling. The field of radiomics aims to extract meaningful features from high-dimensional data. In the future, this could help to find new imaging biomarkers and improve patient-specific predictions by creating profiles for specific cancer types [MWLH⁺20]. Our collaborators at the Mohn Medical Imaging and Visualization Centre (MMIV) at the Haukeland University Hospital in Bergen, Norway, work, among other topics related to medical imaging, on radiomic tumor profiling to improve precision medicine [FBYH⁺18, HHD⁺21]. Figure 1.1 shows a pipeline in medical image processing for radiomic tumor profiling where, ideally, the steps are performed automatically. As we can see when following the gray arrows, radiomic tumor profiling relies on a previous segmentation that delineates the tumor area. In general, the process of segmentation deals with the division of an image into semantically meaningful regions. In contrast to object recognition, it does not include a description or definition of the segmented areas [APT06, MT96]. In case of tumor data, segmentation of the tissue belonging to

the malignant structure is necessary to quantitatively describe the tumor, examining its characteristics in comparison to surrounding tissue, and allowing for predictions or decisions for following treatments [EGSEB11].



Figure 1.1: A pipeline for radiomic tumor profiling in multi-modal medical imaging. In an ideal scenario, the steps are performed automatically following the gray arrows. In our case, segmentation and co-registration are interchangeable. We use the segmentation performed for one sequence to aid the co-registration.

Manual segmentation in medical imaging is still standard in research practice. This means that radiologists manually draw the outlines of a tumor in every slice of one modality. For example, our collaborating radiologists need up to three hours to contour a relatively big tumor of the endometrium in one MRI sequence. Therefore, in recent years, many approaches have been developed to automate this time-consuming manual process [EGSEB11, JAFGM+15, ZFC+17]. Also our collaborators Hodneland et al. [HDWL+21] use Deep Learning (DL) for automated image segmentation of endometrial cancer. In their approach, they predict tumor segmentation masks for one MRI sequence and achieve results that are close to the quality of manually delineated areas. However, one challenge they are currently facing is that the predictions are less accurate in the other six sequences that are available in our endometrial cancer data set [HSG+14, HS16].

As shown in Figure 1.1, a usually necessary previous step for automated multi-modal segmentation and radiomic tumor profiling is the registration of the different modalities or sequences of one patient. We will refer to this as co-registration. The volumes have to be registered after the acquisition to bring them to a common reference space and align corresponding structures in order to exploit the potential of multiple modalities in automated segmentation [MWLH+20, MBSK16, OT14]. Co-registration can account for organ movements, like bladder filling in the pelvic area, and the possible variation in the patient’s position between the scans [MWLH+20]. It is also beneficial in manual segmentation since otherwise the radiologists would have to delineate the tumor in every single sequence [RvdHD+15]. Registration and segmentation belong to the biggest challenges in medical imaging, both with substantial potential clinical impact [SDP13]. For many registration and segmentation problems, Machine Learning (ML) algorithms currently present the state of the art and often lead to remarkable results. Still, scanning artifacts like noise due to patient movement or low resolution, lacking training data due to anatomical variation, and physiological changes make these tasks very challenging [RS14]. These factors often lead to insufficient registration and thus segmentation. Co-registration is an issue in several data sets. For example, Gousias et al. [GER+12] aimed to align T2- to T1-weighted MRI sequences of newborns. The authors report inaccuracies in the

co-registration they could not resolve. They assume that this could be due to the T2 sequence being a pseudo volume since, in their work, the T2 sequence has been acquired with overlapping slices, partial volume effects, or varying distortion fields.

Our collaborators encounter similar problems of unsatisfying co-registration results in endometrial and cervical cancer imaging data. Their work [HDWL⁺21] for automated segmentation yields good results for one modality. However, recently, they have been working on multi-modal segmentation, where the predictions are less accurate, most probably due to insufficient registration accuracy. In fact, when investigating the data set of endometrial cancer, many cases are not well aligned after a registration based on the DICOM header data that contains information regarding the patient’s position in the MRI scanner. Furthermore, available command-line tools that aim to align several sequences automatically did not yield sufficient registration results and need specific settings depending on the subject and sequence. Existing tools for segmentation or co-registration with a Graphical User Interface (GUI) are usually able to process only one modality at a time and often require algorithmic knowledge to find appropriate settings for the specific use case. Therefore, they are currently not used by our radiologists to perform the registration (semi-)manually.

Registration algorithms are either rigid or deformable. Rigid registration transforms a volume by only translation and rotation parameters, while deformable registration can apply local deformations. Tumors are rigid structures compared to the tissue around them [ZRC⁺20, AGM17]. Therefore, rigid co-registration can be sufficient to align the tumor. However, to analyze also surrounding structures, deformable registration is necessary. Even though tumors are rigid structures, they can appear slightly different in the varying sequences due to imaging physics. Therefore, applying deformable registration to the whole image can result in tumor deformations that do not resemble the underlying biological tissue characteristics and can alter important information about the tumor. It would be desirable to be able to apply deformable transformations mainly to surrounding soft tissue to avoid unnatural tumor deformations.

In 2020, we proposed a tool for interactive clustering exploration called ICEVis [MEH⁺20], which supports the investigation of potential tumor sub-regions and features related to the different areas. We computed the sub-regions by hierarchical clustering based on a previously generated segmentation mask. Also in this project, we observe a problem with poorly registered multi-sequential data since the outcomes are different if a voxel in different sequences does not correspond to the same biological location. However, with this application, we can potentially filter out misaligned sequences. Figure 1.2 shows the clustering outcome with ICEVis of an example case where the results indicate a bad registration outcome. In the presented modality (b1000), the bright structure represents the tumor. The segmentation mask does not match the structure, and the algorithm identifies the difference as two distinct areas. In this case, the red part represents an area outside the tumor. Furthermore, the parallel coordinates plot, which indicates the difference in gray values in the sequences, shows a significant divergence in gray values even though the tumor itself is relatively homogeneous.

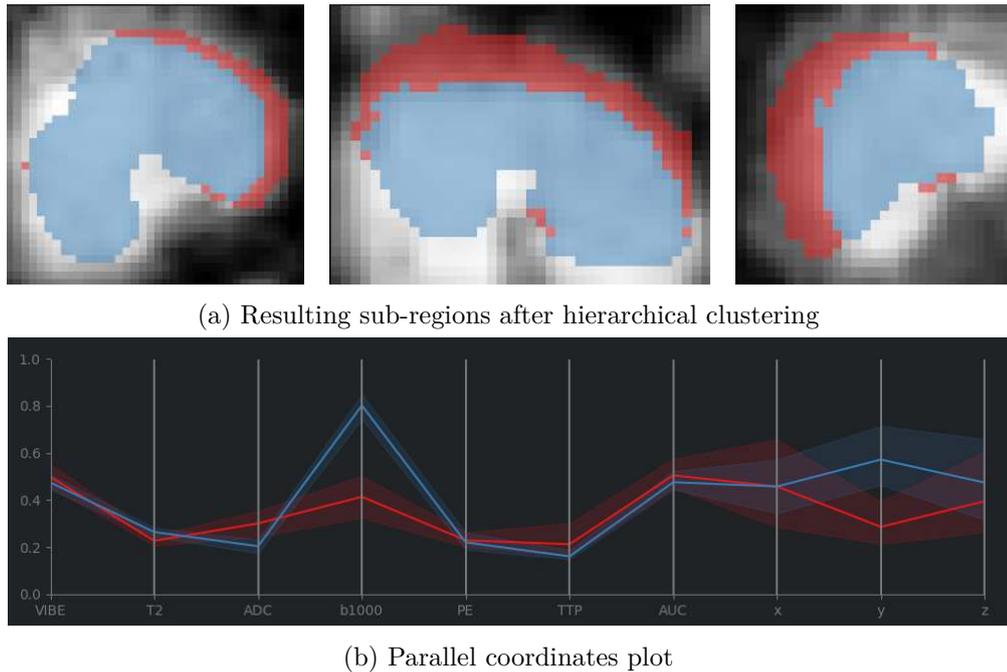


Figure 1.2: Example outcome of the tool ICEVis we developed in collaboration with the [MMIV](#) for exploring tumor subregions in multi-modal medical imaging. This example shows that we can identify badly registered sequences. (a) shows the resulting two sub-regions of the segmentation mask in three slicing directions. The segmentation does not match the tumor in the underlying sequence (b1000). The red part belongs to an area outside the tumor, while the blue region is part of the malignant structure. In the parallel coordinates plot (b), we can observe a high difference in gray values for this sequence.

1.2 Aim of the Work

Automated segmentation algorithms and, as the next step, radiomic tumor profiling could benefit from a better co-registration for complex cases. Therefore, the aim is to facilitate interactive co-registration for endometrial cancer data, especially in cases where automatic registration fails. More specifically, we want to allow for both rigid and deformable registration. In the case of deformable registration, we want to apply local transformations mainly to the areas outside the tumor and not deform the tumor itself. In this way, we maintain the tumor’s shape in the varying sequences, which resembles the underlying biological processes better and could aid research in radiomic tumor profiling in the future.

Furthermore, our medical collaborators at the [MMIV](#) wish to process multiple sequences simultaneously to take different imaging characteristics into account.

Another goal is to make the application easily portable and extendable, so it can possibly

be used in other institutions as well, which encounter similar problems.

1.3 Contributions

To fulfill the aims mentioned above, we propose the tool MuSIC (Multi-Sequential Interactive Co-Registration). The application is targeted towards radiologists in gynecological cancer research to perform co-registration of cancer imaging data with a pre-defined segmentation mask in cases where automatic registration methods do not lead to satisfying results. We have one segmentation mask per patient available that matches one of the sequences. This mask has either been generated manually or by the [ML](#) algorithm of Hodneland et al. [\[HDWL⁺21\]](#).

The main contributions of this master's thesis consist in the single steps of a workflow we defined for the registration of multi-modal cancer images based on a pre-defined segmentation mask:

- An initial automatic **shape matching approach** finds a given 3D tumor segmentation mask in the available sequences. The proposed transformations of the segmentation is shown to the user for each sequence.
- We provide **intuitive user interaction** that allows the user to adapt the positions of the segmentation masks in cases where the shape could not be detected automatically, and **multi-modal visualization techniques** to assess the quality of alignment during the interactive process and after the registration.
- We allow for rigid and deformable **registration** of the data based on the positioning of the segmentation masks.

MuSIC is a **web-based application** that includes these steps to make the approach accessible to our collaborating radiologists in research. The tool allows to process multiple sequences in parallel. No algorithmic background knowledge is necessary to make the workflow suitable for the use by radiologists in research.

We evaluate the proposed work in a qualitative usability study at [MMIV](#) with domain experts in the field of radiology that represent the target user group. We also include [ML](#) experts in the evaluation as they also see potential in the application and would like to use it for better co-registration without having to involve clinical experts.

1.4 Outline

In [Chapter 2](#) we present necessary medical knowledge regarding the pathology of endometrial cancer to understand the use case better. Furthermore, we describe the data set in greater detail and provide examples of the available sequences and badly registered cases, which represent the target cases for our application. This chapter also includes

research work in radiomic tumor profiling and segmentation to emphasize the importance of medical image registration.

[Chapter 3](#) will introduce related work regarding medical image registration based on the different algorithmic registration components. We further differentiate between automated and interactive approaches, and result evaluation.

In [Chapter 4](#), we present our main contributions. We define a workflow and describe the single steps in detail together with the used techniques. The workflow comprises the shape matching component for initialization, as well as web-based visualization, interaction methods, and insights into the registration procedure.

[Chapter 5](#) shows the design of the [GUI](#) and its functionality, while [Chapter 6](#) covers the implementation and structure of our web-based application. In [Chapter 7](#), we describe the processing of two example cases in greater detail, including the outcome of the shape matching and final registration in comparison to other automated tools. In [Chapter 8](#), we present the results of our usability analysis with domain experts.

Finally, in [Chapter 9](#) we discuss and conclude the proposed work. We furthermore present ideas for possible extensions and subsequent work on the tool.

Medical and Technical Background

The pipeline of medical image processing for radiomic tumor profiling starts with the acquisition of MRI cancer images (see Figure 1.1). This chapter introduces the medical and technical knowledge needed to understand the data and tasks of this master's thesis. It includes information about endometrial cancer, the female pelvic anatomy, and the role of medical imaging in cancer research. We will also describe the differences between the MRI sequences of our data set.

To emphasize the impact of registration on tasks like segmentation and, in turn, radiomic tumor profiling, we will then touch upon published work on these two topics before presenting related work in the field of image registration in Chapter 3.

2.1 Endometrial Cancer

Among gynecological malignancies in the pelvic area, endometrial cancer is the most common one and causes the most deaths in countries with elevated income, such as European and North-American countries [MWLH⁺20, HSI6, HSG⁺14]. Reasons related to risk factors could be the higher occurrence of obesity and the increasing age of the population. Therefore, one focus of our collaborators of the MMIV at Haukeland University Hospital in Bergen, Norway, is on researching better strategies to foresee and treat gynecological malignancies.

The recurrence probability of these tumors after surgery is between 15 and 20 % in the pelvic/vaginal area or locations further away from that region. Despite the relatively high five-years survival rate of 80 % after primary surgery, survival expectancy after metastasis detection is only between seven and twelve months [HSI6].

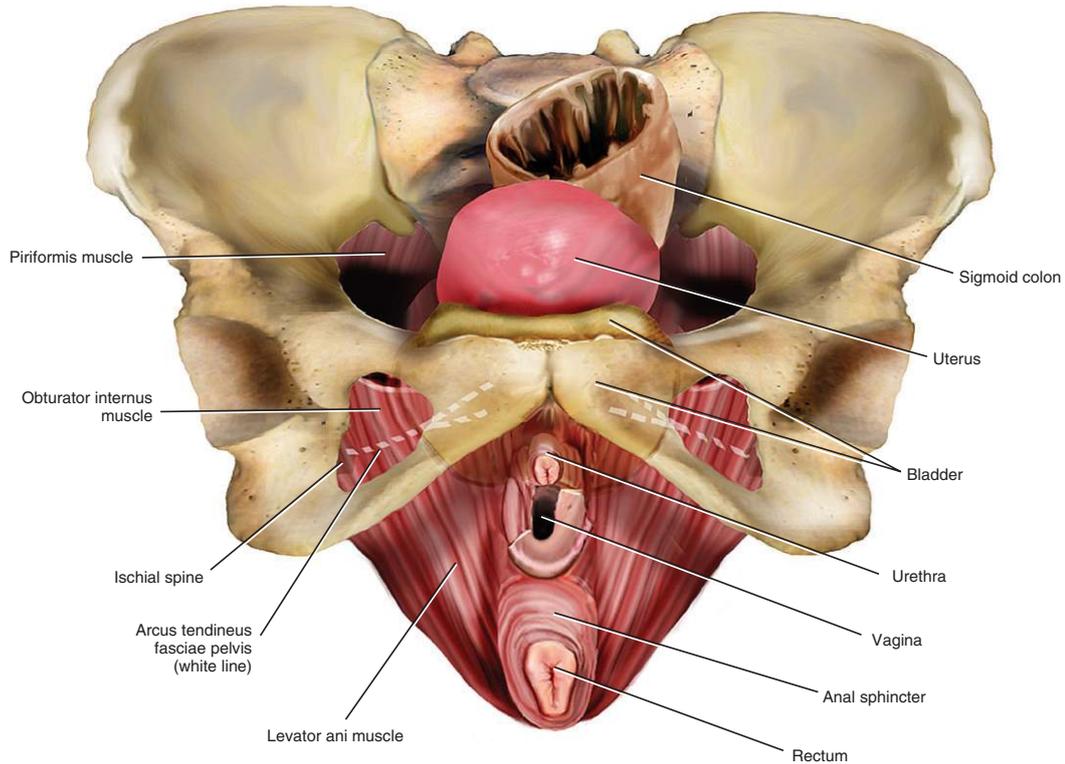


Figure 2.1: Illustration of the basic pelvic anatomy from Baggish et al. [BK15].

Female Pelvic Anatomy Figure 2.1 shows an illustration of the female pelvic area. Bone structures surround soft tissue, including the uterus, colon, bladder, and several muscles. The uterus lies between the bladder and rectum and has a cavity. More caudal, downwards away from the head, the bladder transitions in the urethra, the colon in the rectum, and the uterus connects to the narrower cervix and more caudal in the vagina. The endometrium is the innermost tissue layer of the uterus wall, which is depicted in the T2-weighted MRI image of the female pelvis in Figure 2.2. The figure shows the female pelvic area in a sagittal, which means parallel to a plane dividing the body into a left and right part, and a coronal view, which slices the body from back to front or vice versa. The endometrium itself is subdivided into a functional layer, which lines the uterus cavity and sheds during monthly menstruation, and a deeper basal layer [BK15, BME99]. The pelvic area undergoes inner organ movements, for example caused by the peristaltic movements of the colon or the varying bladder filling. These movements can affect the outcome of acquiring medical images, like CT or MRI, in this area.

Diagnostics and Tumor Staging When showing typical symptoms of endometrial cancer, like vaginal bleeding, an initial endometrial biopsy can confirm the suspicion of an existing tumor. Endometrial tumors are then classified based on the International Federation of Gynecology and Obstetrics (FIGO) staging systems [Pec09]. The classifi-

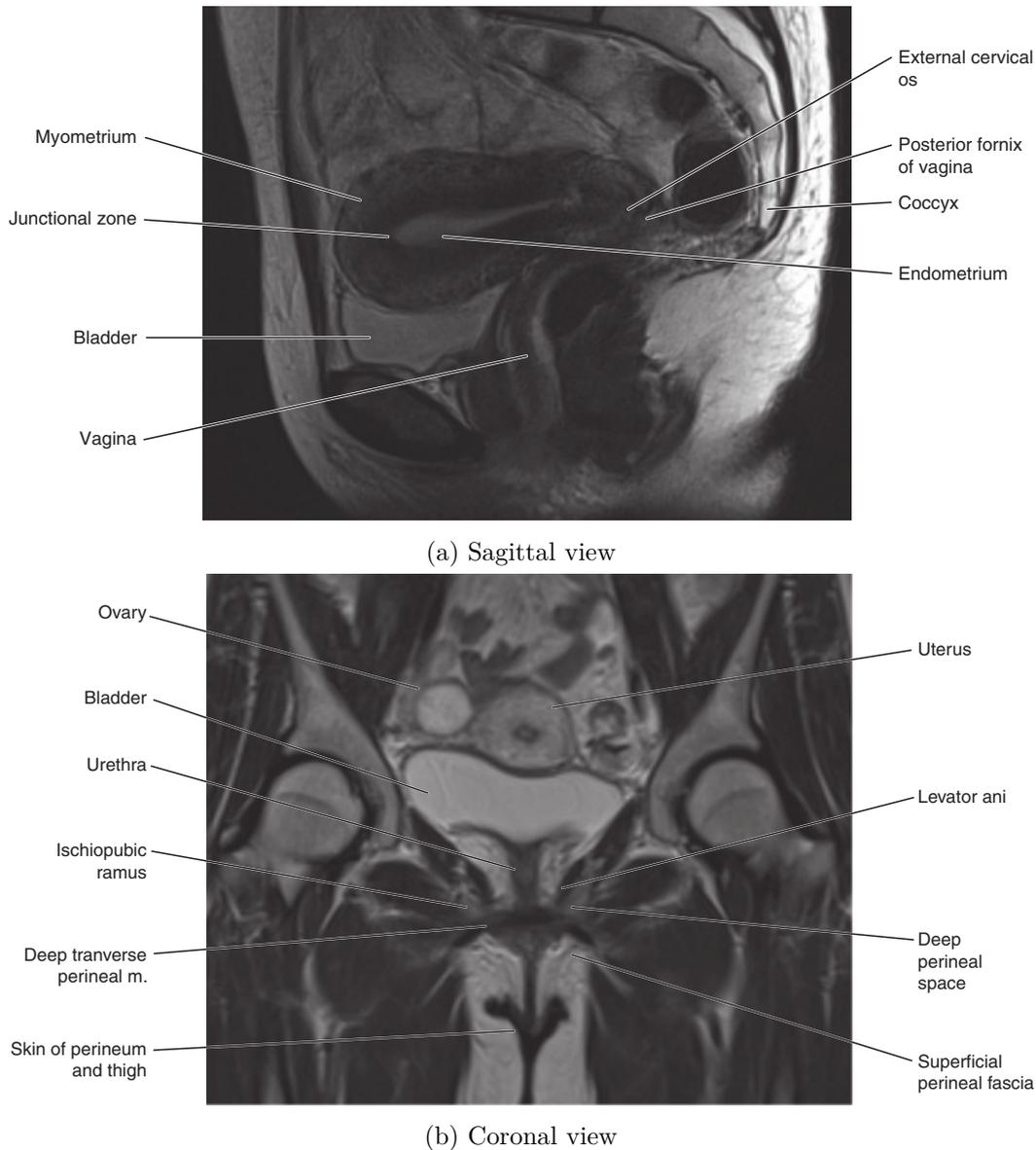


Figure 2.2: T2-weighted sagittal (a) and coronal (b) view on the female pelvic area. From Butler et al. [BME99].

cation scheme for endometrial cancer is depicted in Figure 2.3. If the tumor border is within the upper part of the uterus, i.e., the corpus uteri, it is classified as Stage I. If it also covers parts of the cervix but it is still within the borders of the uterus, Stage II applies. The tumor invades different local areas of the uterus' surroundings in Stage III. Stage IV affects the bladder or comes with metastases in distant locations. The survival probability decreases with increasing stage [BME99].

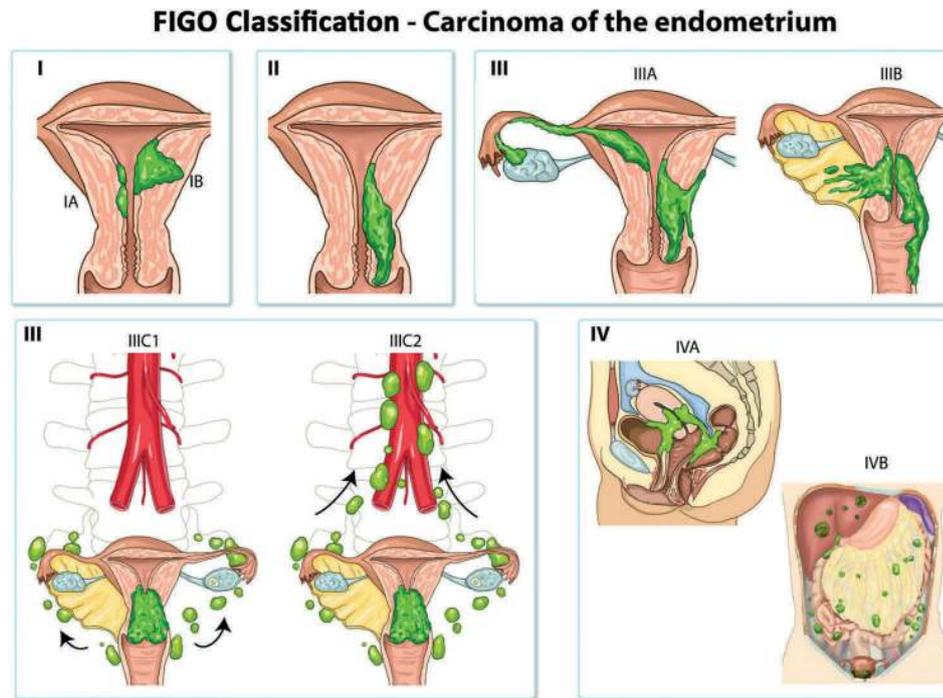


Figure 2.3: FIGO Staging of endometrial cancer, from Pundir et al. [PCI16].

The Role of Medical Imaging The acquisition of several medical imaging modalities or sequences provides deeper insights into human tissues than only considering one modality. The different image physics capture different tissue characteristics regarding anatomy and function [SDP13]. In case of endometrial cancer suspicion, pelvic multi-parametric contrast-enhanced MRI sequences are acquired in radiology routine. In several centers, also CT, Positron Emission Tomography (PET)-CT or Transvaginal Ultrasound (TVU) data is used to aid risk assessment. The resulting scans support the decision of further treatment planning, which usually results in surgery or adjuvant therapy [HS16].

TVU, CT, T2- and T1-weighted MRI and PET-CT are considered as "conventional" imaging for pre-surgical endometrial cancer staging. TVU, while having the advantage of good availability and low costs, comes with the drawbacks of depending on the experience of the physician to get representative images, low quality in deeper tissues, and a small field of view. Especially the latter makes it unsuitable for metastasis detection. CT and PET-CT provide a lower spatial resolution and soft tissue contrast than, for example, MRI and are hence less suitable for primary tumor staging. However, PET-CT provide higher accuracy for whole body metastasis detection. MRI is regarded as one of the most valuable imaging methods for tumor staging. Still, sensitivity, specificity, and accuracy can vary widely. "Novel" imaging methods, which are the focus of research, include 3D Ultrasound and Diffusion Weighted Imaging (DWI) with a lymph node-specific contrast agent [HS16].

In the future, the mentioned image modalities could be used to identify image biomarkers, which means objective indicators for specific medical phenomena, for tumor risk evaluation and personalized treatment [HS16].

2.2 Data Set

Our collaborators of the MMIV at Haukeland University Hospital in Bergen, Norway, provided the data set used during the development phase of this thesis. It includes MRI data of endometrial cancer patients in different sequences, as well as manual and ML-generated segmentations of the tumor [HSG⁺14, HS16].

As mentioned above, the scope of acquiring endometrial cancer imaging data for research is the examination and exploration of tumor biomarkers to improve predictions and treatment strategies on a personalized level. Therefore, researchers aim to analyze the collection of tumor textures in imaging data. The tumor has to be segmented before the analysis phase, which is currently done manually for only one of the seven sequences described below. A limiting factor to segment all modalities is the high amount of time needed for segmenting all slices in one sequence. The other sequences are co-registered. Radiologists in research extend this data set frequently. ML experts we collaborate with [HSG⁺14, FBYH⁺18, YHDL⁺18] as well as many other researchers [ZFC⁺17, JAFGGFGG19, Thr18] work on the improvement of automated algorithms to enable fast segmentation of various sequences.

The data includes information about clinical parameters and the mentioned segmentations. In addition, the data contains multi-sequential axial oblique scans of 92 patients with endometrial cancer. Axial oblique means that the images are acquired in a direction perpendicular to an organ axis. In our case, this is the long axis of the uterus [HSG⁺14].

Available sequences for this work are the Volumetric Interpolated Breath-Hold Examination (VIBE), Peak Enhancement (PE), Time To Peak (TTP), Area Under the Peak Enhancement Curve (AUC), T2, Apparent Diffusion Coefficient (ADC), and b1000. While VIBE and T2 support the assessment of the tumor extent, the other methods reveal tumor characteristics on a microstructural scale. Acquisition of the VIBE modality results in T1 weighted sequences with the advantage of a higher resolution along the z axis [RLL⁺99, KWW18]. DWI measures the Brownian motion of molecules in one voxel, resulting in a lower diffusion coefficient for highly cellular tissue [DCS⁺00, KM04]. ADC allows for quantitative diffusion measurement at different b-values, which determines the applied gradient strength and time point and thus represents the degree of diffusion [Sen01, KP06]. With an intravenous contrast agent, Dynamic Contrast-Enhanced (DCE) MR perfusion enables measurements like PE for the relative contrast enhancement, TTP, and AUC [GPME⁺14]. The sequences in the data set are aligned by performing a co-registration based on their header data. The header data is saved with an image during the acquisition process. Among other general information, it contains the location and position of the patient in the scanner, which can be used for alignment. An example slice depicted in the different sequences can be seen in Figure 2.4.

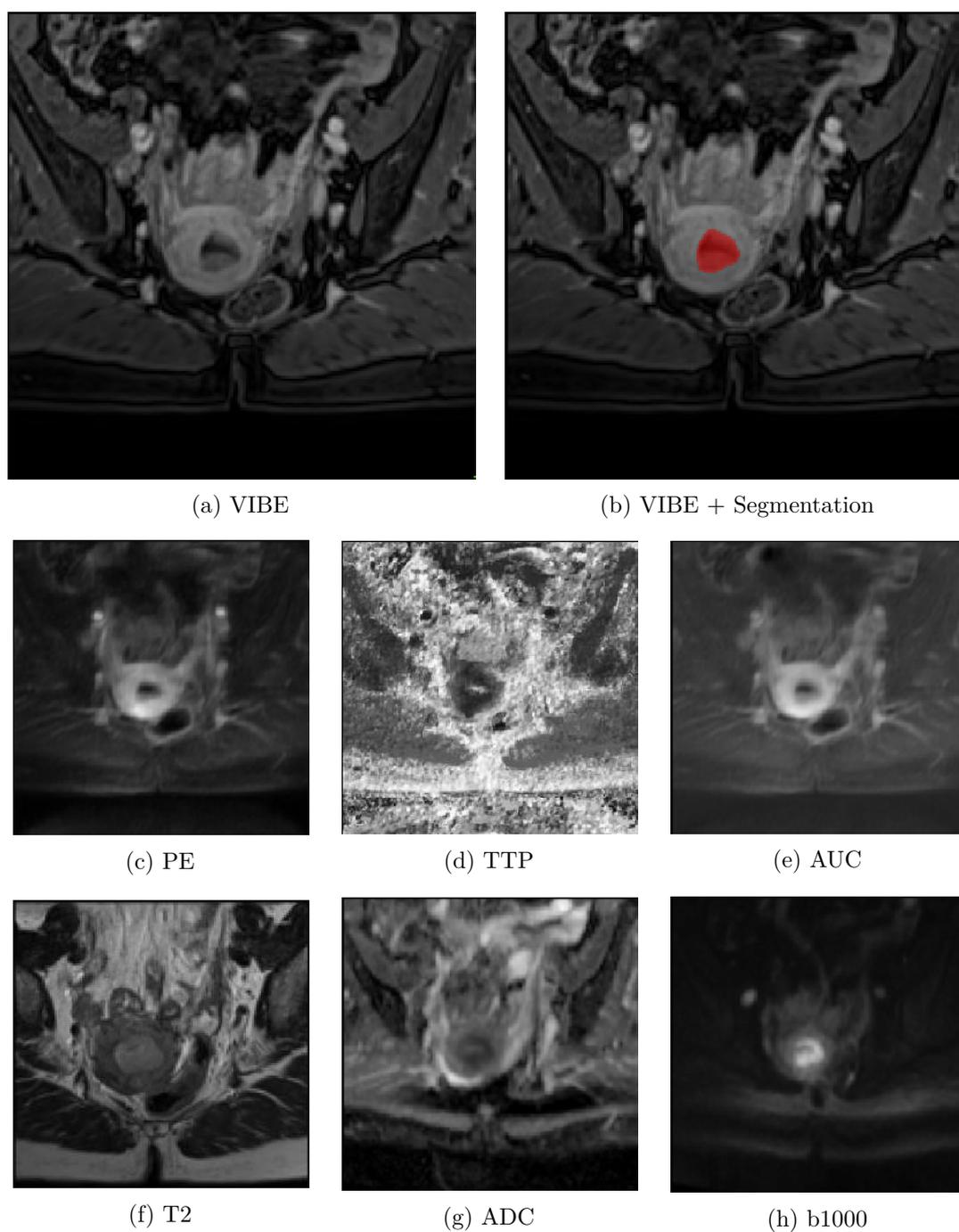


Figure 2.4: Depiction of an example slice in the different modalities from the endometrial cancer data set used for this work. (a) shows the VIBE sequence, which radiologists use for manual annotations. (b) depicts VIBE together with a manual segmentation mask as an overlay, (c)-(h) provide a view on the other six co-registered sequences.

As mentioned in [Chapter 1](#), several sequences of the data are still misaligned after performing registration based on the patient's position in the scanner, as provided in the DICOM image header. An example case is shown in [Figure 2.5](#), where the tumor mask that has been generated for the [VIBE](#) sequence on the top left does not match the tumor boundaries in the other three sequences.

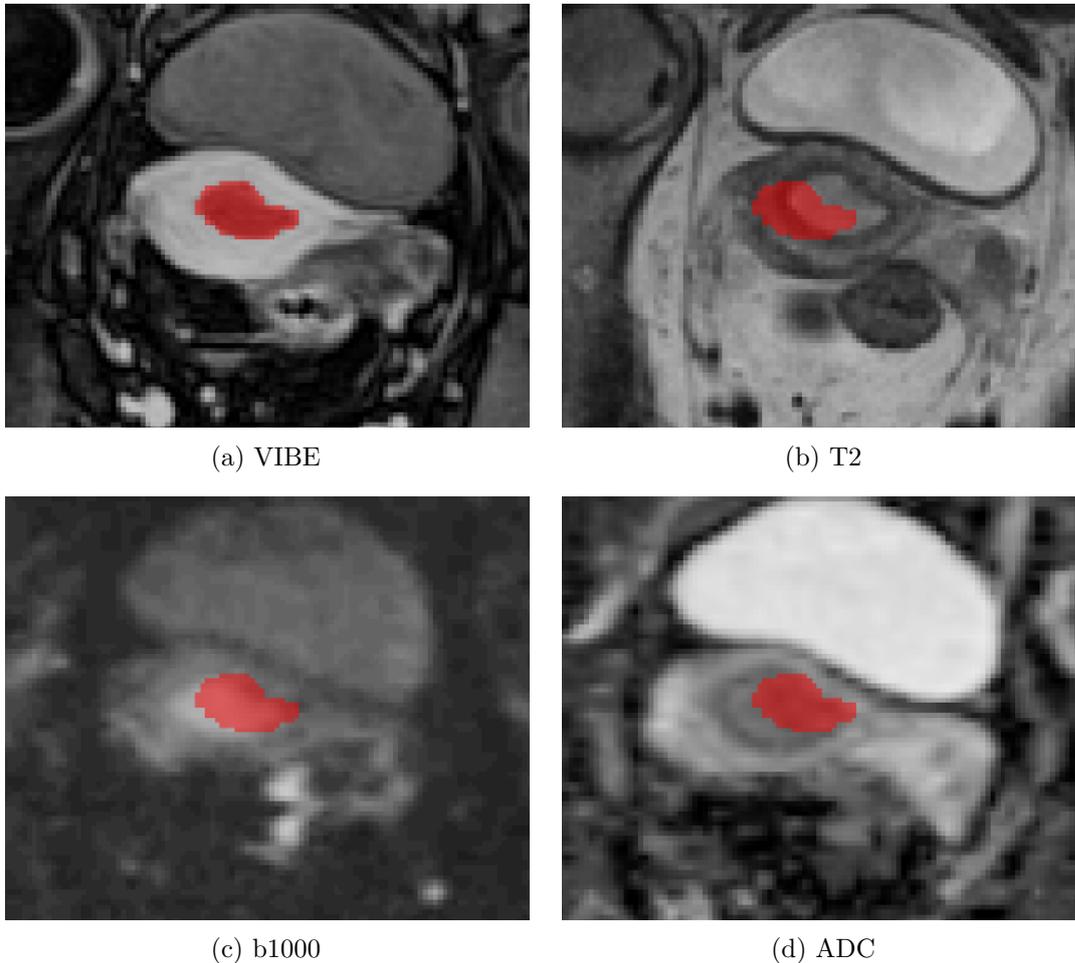


Figure 2.5: Example slices of a case in the data set with poorly aligned sequences (b)-(d). The segmentation mask has bin generated manually for VIBE (a) and is shown semi-transparently in red.

2.3 Radiomic Tumor Profiling

The field of radiomics covers feature selection and extraction in order to perform tumor profiling, which has evolved in recent years to improve precision medicine. The goal of radiomic tumor profiling is to derive new expressive tumor markers by yielding feature

vectors of high dimensions from medical imaging and clinical data. The input for radiomics feature extraction is usually the multi-modal volumetric image and the segmentation of the region to be analyzed. Therefore, the multi-modal segmentation strongly influences the results of radiomic tumor profiling [RBR⁺18, MWLH⁺20]. If multiple modalities are not aligned well enough so that the segmentation covers the actual tumor area, the outcomes are not reliable.

In radiomic tumor profiling, semantic features that are extracted from the region of interest can be classified into two types. While *qualitative* semantic features usually deal with lesion descriptions, *quantitative* features of different complexity can be derived using mathematical methods. The latter describe shape features together with geometric characteristics, such as size or volume, and first-order statistics features, like intensity histograms without considering the spatial position. Second-order statistics features are associated with textual properties, establishing inter-voxel relationships. Higher-order statistics features can be identified after the application of filters and transformations, for example for pattern detection [RBR⁺18]. Current research tries to involve [ML] methods in the feature selection process, as the selection by a user can be problematic. For example, each user might choose a different feature set, which has an impact on the calculated results. Instead, [ML] algorithms can learn the selection of appropriate features based on the data [RBR⁺18].

Our collaborators at [MMIV] work on radiomic tumor profiling in endometrial cancer. Fasmer et al. [FBYH⁺18] use radiomics to assess risk before surgery and aid decisions about upcoming treatment strategies. They extract features like tumor volume, volume-to-surface ratio, and several gray value features. The authors use those features to predict stage, metastasis, myometrial invasion, and other characteristics. Also, Hoivik et al. [HHD⁺21] worked on the identification of malignant cancer features with radiomic tumor profiling. An unsupervised clustering approach results in the division of patients into risk groups and survival prognoses. This division yields a specific signature for high-risk patients that has been evaluated in a larger cohort. iVAR [YJY⁺17] is a visual analytics system presented by Yu et al. to offer interactive analysis of radiomics features in medical images for a high number of patients. They use lung cancer CT data as an example use case. Also, the application RadEx [MWLH⁺20] by Mörth et al. supports radiologists in the exploration of new potential tumor biomarkers in a large cohort. In addition to multiparametric medical imaging data, the authors allow for the inclusion of clinical data in different formats. Their tool provides several linked views that facilitate hypothesis generation for potential tumor markers.

2.4 Segmentation

The process of image segmentation divides the input into two or several homogeneous areas based on a specific criterion [Rog00]. (Semi-) automated approaches can be grouped into supervised or unsupervised [Artificial Intelligence (AI)] algorithms and not-[AI]-based classical mathematical methods, as shown in [Figure 2.6].

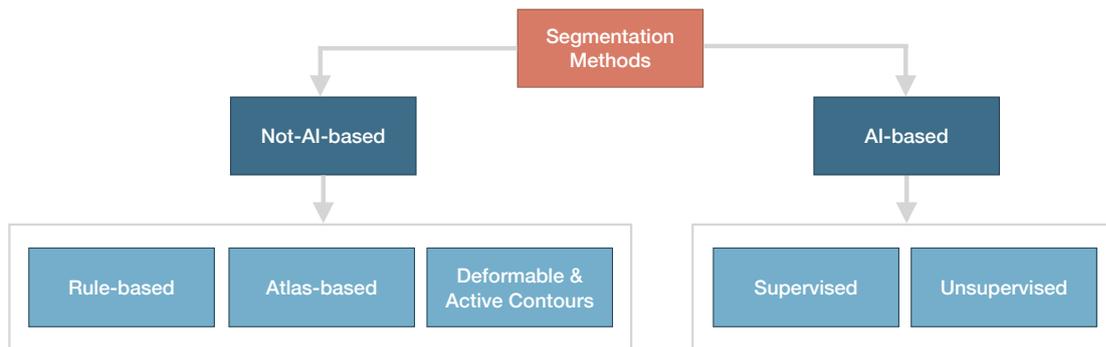


Figure 2.6: Classification of medical image segmentation methods. We followed Elnakib et al. [EGSEB11] for the grouping of not-AI-based methods.

Not-AI-based methods In *not-AI-based* techniques, we can distinguish between methods based on mathematical rules, anatomical atlases, or deformable models [EGSEB11]. Common heuristic *rule-based* algorithms are simple thresholds, often used for a fast initial segmentation. Also, region-growing algorithms belong to this category. These methods are usually not suitable for multi-modal segmentation, as they need to be adjusted for a specific sequence because several tumor tissue characteristics may appear only in a part of the modalities. *Anatomical atlases* contain ordinary organ locations and shapes together with their spatial relations. They offer an easy segmentation method for well-defined organ structures, but not for individually varying tumor characteristics [EGSEB11]. *Deformable or Active Contours (AC)* try to address this issue. These approaches work with minimizing an energy function based on image content and contour characteristics. The energy function of the traditional Snakes algorithm [KWT88] is composed of two components, one handles the smoothness of the curve, and another one pulls the curve towards the boundary. Its minimum is supposed to be on the boundary of an object of interest. Several improved versions of these algorithms have been published and successfully applied to medical imaging segmentation problems [CCCD93, CKS97, MS89].

AI-based methods Emerging technologies in the domain of ML led to remarkable results in a variety of medical applications, especially when dealing with classification problems. A high focus is set on automated tumor segmentation, also working with multiple modalities [HDWL+21, ZFC+17]. In the field of DL, supervised and unsupervised methods can be distinguished that come along with different advantages and disadvantages. Lundervold et al. [LL19] give an overview of DL methods for MRI analysis. A commonly used approach in *supervised segmentation* nowadays is a U-Net architecture, introduced by Ronneberger et al. in 2015 [RFB15]. It consists of a contracting and an expansive path, resulting in an almost symmetric, U-like architecture. This approach was extended to a 3D U-Net [CAL+16] for dense volumetric segmentation, which has been successfully applied for many medical purposes. Among those, Juan-Albarracín et al. [JAFGGFGG19] employed a U-Net for high-grade glioma segmentation, which are brain tumors in the

central nervous system. As mentioned in [Chapter 1](#), our collaborators [\[HDWL+21\]](#) also employed a 3D U-Net to automatically segment endometrial cancer on [MRI](#) images of one sequence. However, supervised approaches are limited to the amount and quality of available training data and easily tend to overfit the training data, which makes them less generalizable for unknown data [\[JAFGM+15\]](#). An advantage of *unsupervised methods* in comparison to supervised ones is that no explicit training data set is required. Instead of learning the class models, natural and meaningful groupings are inferred from the underlying data with the aim to maximize the intra-cluster similarity and minimize the inter-cluster relatedness [\[TSK05, Thr18\]](#). However, the advantage of not needing a training set comes along with a drawback. The segmentation itself becomes more challenging, and such models' performance is often worse compared to supervised models. This is due to the fact that no manual segmentation is present that guides the segmentation process [\[JAFGM+15\]](#).

Clustering methods are commonly used in unsupervised segmentation approaches. These approaches can be grouped into two categories: hierarchical and partitional clustering. In partitional clustering, the data set is divided into non-overlapping clusters in a way that each data object belongs to exactly one subset. Hierarchical clustering includes sub-clusters with a structure organized in a tree. The root of such a diagram is the cluster containing all data objects [\[TSK05, Thr18\]](#). k-means [\[Lo82\]](#) is a well-known partitional clustering method based on a distance minimization criterion. The aim of k-means is to divide the data space into k clusters. In an iterative manner, each data point is assigned to the cluster with the nearest centroid by computing the Euclidean distance [\[JAFGM+15\]](#).

Multi-modal segmentation Several of the above-mentioned methods have been successfully applied to the medical field [\[HDWL+21, JAFGGFGG19\]](#). When segmenting structures using multiple modalities, the fusion of the varying sequences plays an important role, as exploiting the features contained in different modalities can improve the segmentation results [\[ZRC19\]](#). [DL](#) has become a key method for this task in recent years [\[ZRC19\]](#). According to the review of Zhou et al. [\[ZRC19\]](#), the number of publications regarding multi-modal segmentation has been rising constantly. Guo et al. [\[GLH+19\]](#), propose a neural network architecture for segmentation that fuses several modalities (in their example [CT](#), [MRI](#), and [PET](#)) at distinct layers in the network and compare the results. Yuan et al. [\[YWW+20\]](#) use Generative Adversarial Networks (GANs) for multi-modal segmentation of unpaired and unregistered 3D medical images. Zhou et al. [\[ZRC19\]](#) apply correlation constraints to model the relationship between different modalities in a neural network. Also, methods for multi-modal active models have been proposed. The work of Fang et al. [\[FWW20\]](#) includes region information of multiple modalities in the energy minimization function. Hao et al. [\[HQY18\]](#) also work with multiple modalities for the segmentation of juxta-vascular pulmonary nodules of [CT](#) and [PET](#) images. In contrast to the work of Fang et al. [\[FWW20\]](#), they process the different modalities individually. Yang et al. [\[YZN+09\]](#) developed a tool for multi-modal image segmentation using an [AC](#) approach. They follow and extend the "Active Contours

without edges" algorithm by Chan and Vese [CV01] to multiple modalities. In contrast to the traditional method based on the image gradient, average intensity values are considered to make the algorithm more resistant to noise.

To emphasize the importance of registration, it remains to say that the described algorithms rely on a reliable registration. The way features are extracted from multiple sequences by automated algorithms assumes correctly aligned images.

Related Work

Registration is an important topic in medical imaging research. Therefore, a lot of work has been published in this area. This chapter first introduces basic concepts of image registration by describing commonly used algorithms and components. Afterwards, we present more recent work. We focus on the distinction between automatic and interactive approaches. In the end, we describe how to evaluate registration algorithms. For a visual evaluation by the user, we include visualization methods to assess registration results.

3.1 Registration Algorithms and Components

Medical image registration deals with the alignment of two or more medical images that represent the same body area. The aim hereby is to ensure voxel integrity, which means that one voxel coordinate in aligned sequences represents the same location in the human body [OT14]. Image registration is usually applied before segmentation, especially when working with data of multiple patients and modalities. It can have immense benefits for manual segmentation, as the contours drawn in one sequence can be adopted in other acquisitions as well [SRM⁺19]. Since it is often not manageable to acquire multiple sequences at the same time, they need to be fused afterwards [SDP13]. The goal is to align two or more medical images or volumes in the best possible way by optimizing a transformation function [KBD17]. The fixed image is the reference image that is not deformed, while one or multiple moving images are aligned to the fixed image. In general, medical image registration methods require the following components: a transformation, a similarity measure, and an optimization method [RS10]. Since discrete image processing steps might cause artifacts, interpolation is used to improve the resulting visual image quality [LGS99]. The components and the registration process are depicted in Figure 3.1. To achieve a satisfying registration result, choosing an approach suitable for the use case is crucial. After registration, it is essential to evaluate the methods accordingly.

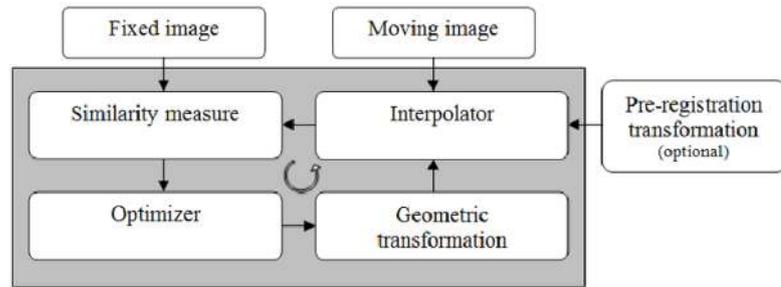


Figure 3.1: Components of an intensity-based registration process. The moving image is deformed to align with the fixed target image. From Oliveira et al. [OT14].

Medical image registration belongs, next to segmentation, to the biggest challenges in medical imaging [SDP13]. The given data complexity results in difficulties, especially when it comes to noisy images, organ movement, and anatomical variability, as well as variations of intensity distributions due to scanner and workflow characteristics [RS14]. Therefore, a lot of work has been published in this area. The review of Maintz et al. [MV98] provides a broad classification scheme for registration methods. Moreover, they define the classification criteria which we depict in Figure 3.2. After describing the different aspects of dimensionality, modalities, subject, object and task, registration basis, nature of transformation, transformation domain and optimization procedure, we focus on the distinction in terms of interaction in Section 3.2.

Dimensionality Considering spatial dimensions, we often have to deal with the registration between two 3D images in medical imaging. However, it might be the case that 2D slices need to be registered to 3D volumes or vice versa. Registration from 2D to 3D is necessary, for example, in image-guided surgery. Also the registration of time series with time intervals of varying length finds application in medical imaging. Examples are to monitor the injection of a bolus moving through vessels, or the long-term observation of tumor or bone growth [MV98, SDP13].

Modalities involved Mono-modal registration processes images from the same modality, for example, two CT sequences. If, in contrast, multiple modalities are involved, the registration procedure is called multi-modal, for example the registration between PET and MRI [MV98]. Even though different MRI images are strictly speaking of one modality, we refer to this problem as multi-modal, as described in Chapter 2, the images differ strongly and cannot be considered mono-modal in a mathematical sense. Also patient-atlas registration is possible, which can be mono- or multi-modal [AURU⁺16].

Subject If several scans of one patient are registered, we refer to this process as intra-subject registration or co-registration. The alignment of medical images of different patients is called inter-subject registration. If the fixed image has been generated from a

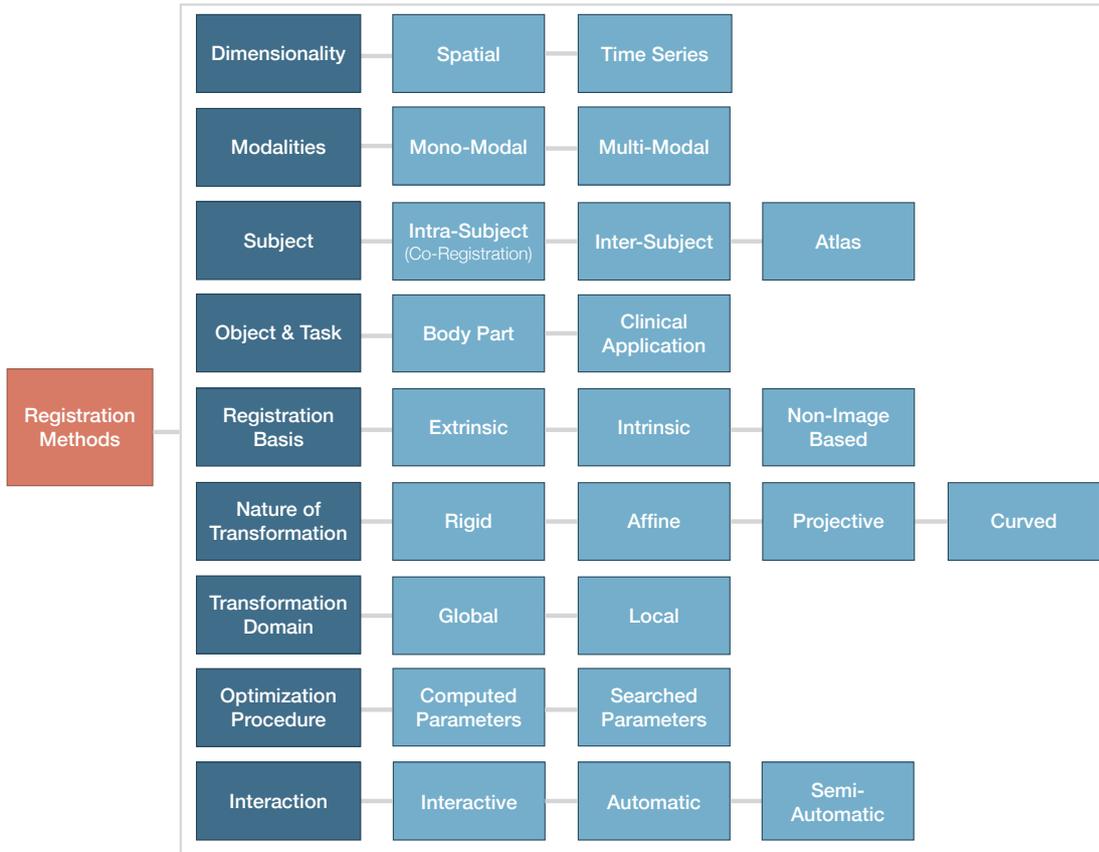


Figure 3.2: Classification scheme for image registration methods. Following the grouping of Maintz et al. [MV98].

large set of subjects, we talk about atlas registration [MV98]. We address the problem of intra-subject registration in this thesis.

Object and task Published approaches often focus on the use for specific body areas, like cardiac or brain registration [KBD17]. Furthermore, approaches differ by the task they support. A substantial amount of work has been published in the area of treatment planning, for example radiotherapy planning [SDP13, SRM⁺19, SFF06]. Also, surgery scenarios have to be considered, as in image-guided therapy approaches, the pre-treatment images have to be registered in real-time to the current state of the patient, including position and movement artifacts [SDP13].

Registration basis The information used for image registration can be extrinsic, intrinsic, or non-image-based [MV98]. The latter can be applied in the same acquisition session of one patient, assuming that the patient does not move and the scanner is in the same position. The former is based on extrinsic markers, which means from outside

the image space, usually attached to the patient. In this case, the registration procedure is fast-forward. However, preparation and planning steps are necessary before image acquisition. Those markers can be invasive, like screwed frames or other markers, or non-invasive [AURU+16]. Intrinsic methods can be based on intensities only or involve previously determined landmarks or segmentations. The advantage of the latter two methods is that the search space is limited, so it is less probable to get stuck in local optima during the optimization procedure, and it is less time-consuming. An expert user typically defines corresponding anatomical landmarks at easily detectable locations in every image [MV98].

If the registration is based on **segmentations**, different procedures are possible. The desired structure can either be segmented separately in the different images, or the segmentation can be searched in the moving images. The model can be rigid or deformable. However, the problem of the latter case is the dependency on a good initialization and good image contrast [MV98, AURU+16]. Several approaches have been proposed in the field of image-guided surgery [CKMK19, KDSU+16] or cardiac imaging [TA13]. They require deformable registration to account for organ movements between pre-surgical and intra-surgical images. Various approaches focus on combining segmentation and registration and include automatic segmentation generation [CDPRE21, KDSU+16, SFF06, CKMK19]. Image masks can be used to define the region of interest for the registration. Ketcha et al. [KDSU+16] assign weights to a mask defining regions that are reliable for segmentation and areas containing content mismatch between the image series. A way to automatically generate masks is proposed by Schumacher et al. [SFF06]. They use [AC] to generate weighted masks of organs in the moving image, given the segmentations and weight information in the fixed image.

Nature of transformation As shown in [Figure 3.3], the nature of a transformation can be rigid (involving only translation and rotation), affine (with additional mapping of parallel lines onto parallel lines), projective (where shear is applied along non-parallel lines), or curved (where lines are mapped to curves) [MV98].

Transformation domain If the transformation is applied to the whole image, a global or parametric transformation domain is used. In a local transformation domain, different image regions can have diverse transformations. This is also referred to as non-parametric registration, as the model is parameterized locally, leading to a high number of variables and a higher complexity [Yan17].

The latter, however, may impose the challenge of local discontinuity and violated invertibility. Often, global rigid or affine registration techniques are used in medical imaging. Rigid transformations imply the determination of only a few transformation parameters for translation and rotation. Even if considering body and internal organ movement, they are often a satisfying approximation of the underlying problem. For more complex transformations, affine methods can be used that involve shear and scale, while parallel lines are preserved [MV98].

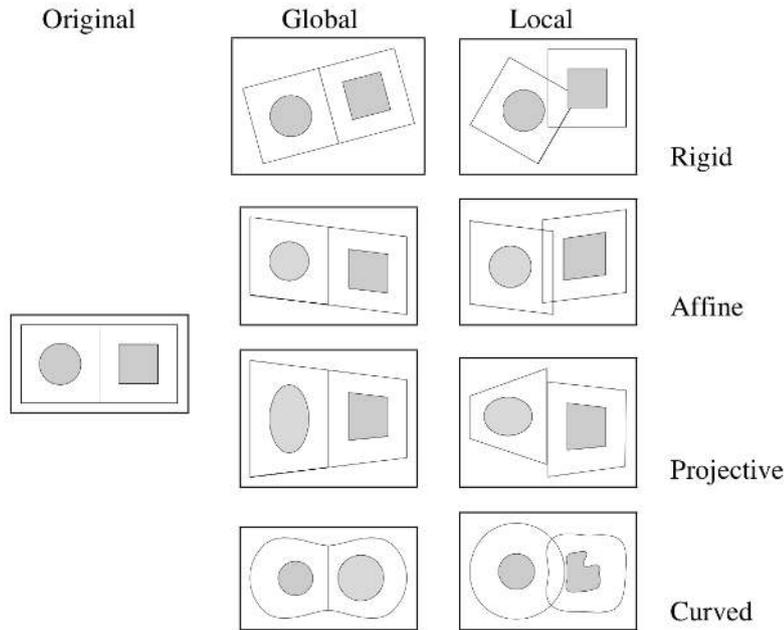


Figure 3.3: Examples of 2D transforms in image registration. From Maintz et al. [MV98].

Optimization procedure Transformation parameters can be directly derived from the underlying data or searched for with the help of an optimization function. The latter usually optimizes the similarity between two images given a mathematical function that captures the underlying problem. In mono-modal registration problems, such a function can be relatively simple, while more complex patterns are necessary for multi-modal applications [MV98]. Multi-modal problems can be solved by either changing them to a mono-modal problem or using information-theoretic approaches. For the first method, one of the modalities can be mapped or simulated by the other one, or both modalities are transferred to a standardized domain [SDP13]. In the work of Michel et al. [MP10], the conditional probability of a target voxel intensity is learned from a training data set of registered images through stochastic regression. Afterwards, they use a Markov Random Field for regularization. Information theory approaches are used more often in the literature to solve multi-modal problems than transforming the problem to a mono-modal case. Examples for information theory approaches are the use of gradient descent methods, like in the work of Klein et al. [KPSV09], stochastic search, for example mutual information maximization by Viola et al. [VW197], and optimization with simulated annealing by Kirkpatrick et al. [KGV83]. Gradient descent methods have been widely used for regularization purposes. The optimization function is minimized by going stepwise in the direction of the negative gradient [SDP13]. However, deterministic gradient computation can be expensive, especially in high-dimensional medical search spaces. Stochastic gradient descent methods use an approximation of the gradient for faster computation. Another drawback is the potential slow convergence. Therefore, conjugate gradient descent methods use the information of previous steps to

calculate the new step direction. As with other continuous methods, gradient descent searches for optima locally, so good initialization might be important to avoid ending up in local optima. On the other hand, discrete methods like Markov Random Fields or simulated annealing are often less precise, but also less sensitive to bad initializations. Therefore, they are less likely to end up in local minima. With increasing precision, also computational expenses rise [MV98, SDP13].

3.2 Trade-off between Automation and User Interaction

Maintz et al. [MV98] distinguish between fully automatic methods, semi-automatic ones considering initialization and adaptations, and manual interactive approaches with or without initialization. The latter, however, are not very present in the literature, which might be because completely manual registration depends on good visualizations and still requires a high amount of time from the medical expert.

As described in [Chapter 2], this work involves 3D medical images of multiple modalities that need to be co-registered within one subject. The degree of automation is an essential factor in this thesis, so we will examine related work in terms of this aspect and the corresponding user interaction.

3.2.1 Machine Learning Algorithms

Many publications exist that present approaches for automated algorithms [FLW⁺20, HKY20]. Complete automation is often considered as the main goal. Extrinsic methods with markers that are well visible in the medical scans can, in most cases, be automated. Intrinsic methods with anatomical landmarks or segmentations are usually semi-automatic, while methods relying on voxel properties are mainly automated [MV98, VLM⁺17, AURU⁺16].

In recent years, more and more [ML] and especially [DL] approaches for image registration have been published that overcome existing challenges [FLW⁺20, HKY20, CDPRE21, CW18]. Especially the difficulty of defining an appropriate similarity measure for intra-subject multi-modal registration is addressed with [DL]. Fu et al. [FLW⁺20], as well as Haskins et al. [HKY20] provide a review regarding the use of [DL] for medical image registration.

Supervised learning There are different possibilities to apply supervised [ML] to image registration problems. One way is to learn the similarity metric between two or more modalities. Cheng et al. [CZZ18], implement a stacked denoising autoencoder for rigid [MRI] to [CT] registration. With the help of a [Convolutional Neural Network (CNN)], Simonovsky et al. [SGBM⁺16] use gradient descent to optimize the dissimilarity between T1- and T2-weighted [MRI]. Instead of the similarity metric, another option is to let the network learn the transformation parameters. The work of Chee et al. [CW18], called Affine Image Registration Network (AIRNet), involves a [CNN] for rigid [MRI] brain

registration. They train the network by using a mean squared error minimization. With this method, they can improve the state of the art of uni- and multi-modal registration. For the actual registration step, the authors use the affine transformation function provided by SimpleITK [LCIB13] with mutual information as similarity metric, linear interpolation, as well as gradient descent optimization.

Some methods incorporate segmentations or landmarks to improve registration results in desired areas. For both segmentation or landmark registration, a combined loss that is optimized to align both labels and image intensities can be used. Hering et al. [HKHH19] use such a combined loss function for mono-modal deformable registration of cardiac images for tracking of heart motion based on segmentations. To assist brain tumor resection, Canalini et al. [CKMK19] provide a deformable registration method for Ultrasound (US) volumes based on a CNN trained on previously generated manual segmentations. They account for the brain shift phenomenon when opening the skull and further brain movements in different stages of the surgery. In their approach, they allow for the registration of several volumes acquired during the different surgery stages. Instead of using existing segmentations for the alignment, Xu et al. [XN19] combine the tasks of segmentation and mono-modal deformable registration in a joint training of two networks. If no segmentation is available, a segmentation network generates the segmentation before registration. Mahapatra [Mah18] trained a Generative Adversarial Network (GAN) for medical image registration with images containing corresponding landmarks. Vandaele et al. [VLM+17] work with automated landmark detection using extremely randomized trees and multi-resolution voxel windows. Then, they perform rigid registration by optimizing a least-squares algorithm between CT and cone-beam CT for radiotherapy preparation. In that specific case, other methods often fail, as cone-beam CT images usually contain a smaller body part, and user interaction is required for selecting regions of interest. They apply supervised ML to detect landmarks with a separate model for each landmark and modality. This method implies manual annotation in training data for each use case. Furthermore, they assume that the landmarks look very similar across patients. Certain bones are well visible in CT scans and can be used for radiotherapy patient-positioning. Tumorous tissues, however, are very heterogeneous and, additionally, might transform even surrounding soft tissue.

Motion correction might be desired to account for internal organ movement causing deformation in multi-modal images. Jun et al. [LYZW18] implemented a CNN to account for motions in the lung area caused by respiration. The survey of Tavakoli et al. [TA13] covers segmentation and registration methods for cardiac images, where motion correction can be required.

Unsupervised learning In contrast to the previously mentioned approaches, unsupervised learning does not require ground truth data for unregistered sequences. Here, most methods focus either on using a similarity metric (or a combination of several similarity metrics) or other image features to estimate the transformation [HKY20].

Stergios et al. [SMM+18] employed a CNN based on a similarity metric that combines

linear and non-rigid transformations of motion lung [CT](#) data. They use [Mean Squared Error \(MSE\)](#) and regularization for their loss function. Ferrante et al. [\[FOGM18\]](#) apply transfer learning to improve the generalization of algorithms to new medical domains. Their loss function consists of a normalized cross-correlation component and regularization terms. They evaluate their method on mono-modal cardiac X-ray and [MRI](#) cine images to include motion correction. The approach performs well on similar image domains by fine-tuning the parameters, but struggles with entirely different scenarios. For multi-modal imaging, Sun et al. [\[SZ18\]](#) use a [CNN](#) composed of a feature extraction part, a generator of deformation fields, and a spatial sampling component to register [MRI](#) and [US](#) brain images to be able to detect brain shifting intra-operatively. They combine information about pixel intensity and gradient values as a similarity function.

Kori et al. [\[KK19\]](#) present a 2D feature extraction approach for T1- and T2-weighted brain [MRI](#) data with affine transformation parameter regression. They use pre-trained networks and calculate the loss with the Dice score [\[Dic45\]](#) of the binary moving and fixed image.

Problems While it would be desirable to have thoroughly automated registration algorithms, the medical domain imposes several problems and challenges. The search space for complete automation of non-linear registration problems has a high dimensionality and involves many parameters. Computational cost and ending up in local minima due to non-convexity are limiting factors. For example, using a [CNN](#) to determine a similarity measure is often slower than using a pre-defined metric, as in the case of Simonovsky et al. [\[SGBM⁺16\]](#). Furthermore, solving a registration problem involves different parameters depending on the organ or body area shown in the images [\[MV98, SDP13\]](#). Especially in multi-modal registration, finding a robust similarity measure that is applicable for different scenarios is difficult and limits current research approaches [\[HKY20\]](#).

In the case of [DL](#), researchers have to face the following additional challenges: Supervised learning requires an extensive and high-quality ground truth data set for sufficient robustness [\[HKY20\]](#). However, ground truth data in registration is often lacking, especially since the requirements for applications differ strongly, depending on the use case, and manual registration is time-consuming and often subjective. Furthermore, similar to segmentation problems, with clinical ground truth data, the accuracy depends significantly on the physician carrying out the manual registration. Several authors, for example Eppenhof et al. [\[EP18\]](#), generate ground truth data by applying random transformations to a well-aligned data set. In this case, one has to make sure that the simulated data is similar to actual data retrieved in a clinical setting. Haskins et al. [\[HKY20\]](#) state that also the uncertainty measurement of model predictions remains an open challenge. Last but not least, the algorithms mentioned above are not (yet) included in tools, so they cannot be used by radiologists to register images and, at the same time, evaluate the registration outcome. Even with high accuracy in most cases, there will always be outliers where the registration fails because of noise or individual patient characteristics.

3.2.2 Command Line Tools

Several toolboxes have been provided to users as command line interfaces. Usually, such tools require parameter settings by the user. Therefore, the user needs algorithmic background knowledge.

One of them is the well-known open-source C++ framework [Insight Segmentation and Registration Toolkit \(ITK\)](#). [ITK](#) offers many software tools regarding image processing and analysis for developers. Its first version has been published in 1999 with the aim to aid reproducible research. Now, it is a popular platform for medical image analysis. In fact, several of the command line tools we will present below are based on [ITK](#).

[Advanced Normalization Tools \(ANTs\)](#) [\[ATS09\]](#) belongs to the most used software tools for biomedical image analysis [\[KBD17\]](#). They provide segmentation, registration, and geometrical quantification algorithms. For registration, several methods are available, among them rigid, affine, and transformable registration. R and Python interfaces are available. [Plastimatch](#) [\[SLW+10\]](#) is an open source software for image processing in radiotherapy. Besides functionality for cone-beam [CT](#) and medical image reconstruction, they provide a tool for fast deformable image registration. They support only the registration of two volumes at the time by a multi-stage approach. [Elastix](#) [\[KSM+09\]](#) is a command line tool for medical image registration based on image intensities. It uses the [ITK](#) registration framework as well as other data structures of [ITK](#). [SimpleElastix](#) [\[MBSK16\]](#) is a library to make [Elastix](#) available easily for multiple programming languages, among them Python. It extends the library [SimpleITK](#) [\[LCIB13\]](#) which aspires to the same purpose for several other [ITK](#) libraries. [FSL](#) [\[JBB+12\]](#) focuses on brain imaging data and provides several algorithms for anatomical, functional, and diffusion images. Their tools [FLIRT](#) and [FNIRT](#) offer linear and non-linear registration. Apart from that, tools for brain extraction, multi-subject analysis, subcortical segmentation, and many more are available.

3.2.3 Tools with Graphical User Interface

In order to overcome the mentioned limitations, user interaction is required. There are several tools that provide interactive registration approaches for medical imaging data.

[Li et al.](#) [\[LXN+05\]](#) assign a color to each individual volume during the interactive rigid registration process to make them easily identifiable. The color homogeneity is then used as a similarity measure, where they assume that homogeneous regions are well aligned. [Cheung et al.](#) [\[CK09\]](#) let the user set different landmarks during an iterative registration procedure. The user identifies landmark pairs based on visual similarity. [RegistrationShop](#) is a 3D registration tool by [Smit et al.](#) [\[SHS+14\]](#) that offers interactive manual, landmark-based, and automatic transformations. A screenshot of the tool is provided in [Figure 3.4](#)

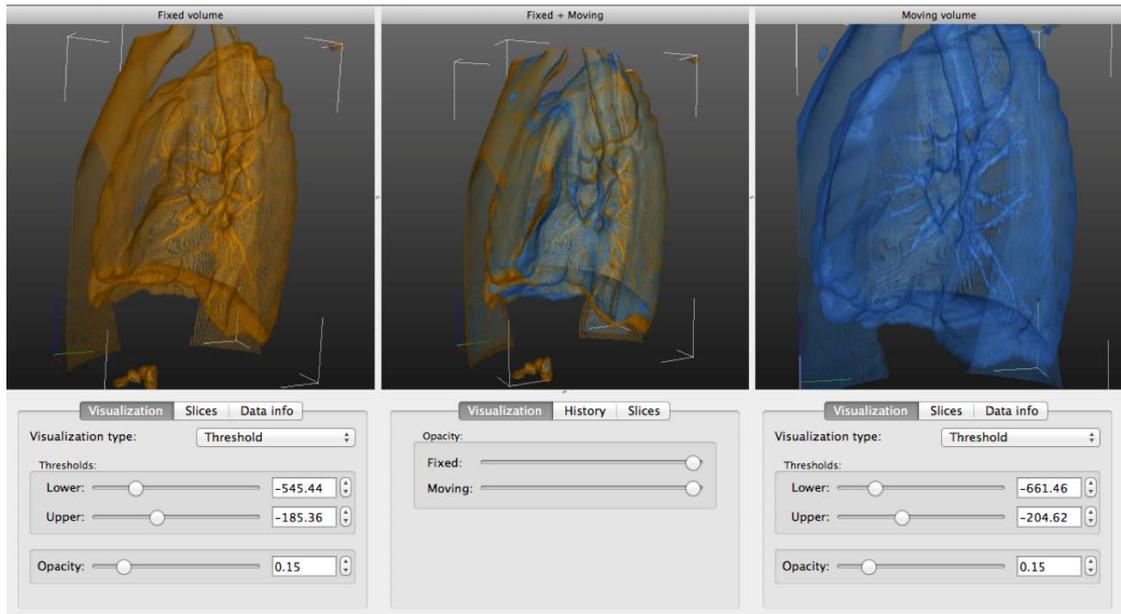


Figure 3.4: Main interface of the tool RegistrationShop for 3D interactive registration. The user can adapt the visualization style for both volumes and interact with the volumes in space. From Smit et al. [SHS⁺14].

Publicly Available Tools

Not all algorithms are provided in existing tools or software [KBD17]. This narrows the potential of them being used by medical professionals. The review of Keszei et al. [KBD17] focuses on how accessible these tools are to make them available for research regarding medical image processing. They focus on non-rigid registration methods and evaluate them based on specific criteria. In total, they evaluated 18 different tools. A large number has been developed for brain registration, for example, the widely used tool SPM [Ash12], or the software AIR [WGH⁺98]. However, some are also applicable to other body regions. In general, they observed an ongoing trend of registration based on intensity values rather than features. The evaluated criteria are: access and technology, interfaces and application, community support (i.e., the latest updates and how active the support is), support of standard medical image formats, and finally registration type and similarity measures.

We have mentioned the approaches presented by Keszei et al. [KBD17] which are relevant for this work in the description of existing command line tools (ANTs, FSL and Elastix). The authors include further software with graphical user interface in their survey. 3D Slicer [PHK04] is one of them. In contrast to other tools, it does not focus on one body region or application. It builds upon ITK and offers functionality for medical image analysis and visualization as a desktop or virtual reality application. Several modules are available for image registration based on manual, semi-automatic, or automatic

registration. For developers, most of the functions are also available in Python and C++. A Plastimatch [SLW⁺10] plugin is available for the user interface. FireVoxel [MR] is a software package with graphical user interface for multi-modal medical image analysis, available for Windows. For interactive co-registration, the user is able to choose a region of interest in which the registration will be more accurate than in outer areas. Furthermore, the user can provide additional algorithmic settings before the registration process starts. The images are shown in a side-by-side view, and the result is displayed as a color-encoded overlay.

Even when providing a GUI, tools often require the users to have algorithmic knowledge to set specific parameters. This fact makes them more flexible, but it limits their usage for medical professionals like radiologists that are not familiar with the backgrounds of the algorithms.

The presented approaches do not fulfill the requirements we specified together with medical experts. To the best of our knowledge, there is no approach aiming to find an optimal position for a given segmentation mask in different sequences that is then used for registration. In the case of deformable registration, the segmentation mask helps to apply local transformations mainly in structures surrounding the tumor which matches underlying tissue characteristics and avoids tumor deformations. Furthermore, the focus of published work in case of non-existing segmentation masks is to generate them in the moving images. In contrast, we focus on the search of the structure that fits a given mask. In addition, publicly available automated methods and command line tools require settings that can be patient-specific or even vary for each sequence. For cases where automated methods fail, we need a way to interactively correct the results. A drawback of existing tools with GUIs is that it is only possible to register exactly one volume to another one. Multi-modal support in graphical user interfaces is still lacking.

3.3 Evaluation

Medical image registration algorithms, especially non-rigid ones, are difficult to validate. This is often due to the lack of sufficient ground truth data. Therefore, there has not been much progress in the development of appropriate evaluation methods in the last years, and it can be considered as an unsolved problem [PMEM16]. To evaluate a registration method, we need to assess its outcome accuracy. This can be done with mathematical measures to quantify the error and by providing appropriate visualizations for a visual, usually more qualitative evaluation [PMEM16].

3.3.1 Numerical Evaluation

Most of the published literature focuses on mathematical measures for evaluation. We can distinguish evaluation methods using ground truth data and the ones using a gold standard. The former methods use an accurate transformation measure for evaluation, while the latter approaches work with an approximate measure [PMEM16].

Evaluation on ground truth data Most methods involving ground truth data rely on the synthetic transformation of aligned images. Apart from that, also phantoms that model human body parts can be used [PMEM16, OT14]. Maglogiannis et al. [MZ04] evaluated their work on registration using Simulated Annealing (SA) with different 2D images of dermatological cancer, US, brain MRI, and lung X-ray. They apply different translation, rotation and scaling to the images within defined ranges (scaling between 0.1 and 1.0, rotation between -30 and 30 degrees). Pluim et al. [PMEM16] propose a method for local error estimation using a CNN that is trained on synthetically transformed data of 2D digital subtraction angiography. The error is assessed locally by training the network on image patch pairs. Wang et al. [WDO⁺05] use several ways to evaluate their registration approach: synthetically transformed CT images, a physical phantom of the pelvic area, and generated segmentations.

The problem of synthetically generated transformations or phantoms is that they often deviate from realistic scenarios. Noise, appropriate organ deformations, and imaging variations are usually not taken into account [PMEM16]. Wang et al. [WDO⁺05] tried to produce more realistic evaluation data by generating transformations based on manual and automatic control points to specify the applied deformation. It is, however, very difficult and time-consuming to create phantoms that cover several realistic scenarios [SFJ⁺16].

Evaluation using a gold standard For gold standard validation, it is usually not recommended to use other similarity metrics than the one used for the registration itself. Another similarity measure will lead to invalid evaluation outcomes, as it does not measure the same information. Furthermore, it will lead to worse results, as it is probably less suitable for the use case (otherwise it would have been used for the registration procedure). However, it is also not a valid measure to use the same metric, since the algorithm already aims to find the best solution for it [PMEM16].

Another option is to measure the alignment of certain segmented anatomical tissues, for example by computing the overlap. In this case, a problem that might occur is the possibly undetected misalignment of regions that are not covered in the segmentation. Simonovsky et al. [SGBM⁺16] evaluate their T1- to T2-weighted MRI registration approach with a separate data set of 20 aligned brains of newborns, while their CNN has been trained on adult data. They compute the Dice [Dic45] and Jaccard coefficients [Jac12] of manually segmented regions in comparison before and after registration and report how much the Dice score improved. Furthermore, they compare between their metric and registration using mutual information with and without mask to show that their approach outperforms the other metrics. Chee et al. [CW18] use ground truth segmentation masks of an aligned T1- and T2-weighted brain image data set to evaluate their supervised neural network approach for inter-patient registration. The data set contains images of 12 subjects, including manual segmentations. They evaluate the method with the Jaccard index and the modified Hausdorff distance [HKR93] to assess the similarity between the image masks of two patients after registration.

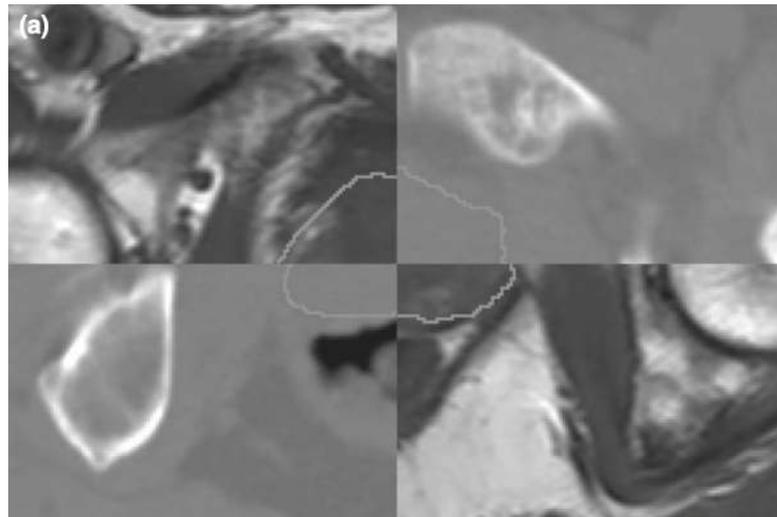
A generally accurate measure is to set manual landmarks and measure the distance between them after registration [OT14, PMEM16]. It is regarded as the method with the highest reliability to assess registration accuracy [PMEM16]. Pluim et al. [PMEM16] train a classifier to distinguish categories of registration errors based on features that were computed around manually set landmarks. The algorithm distinguishes between good, poor, and incorrect alignment. Fitzpatrick et al. [FW01] also described the use of physical landmarks and the computation of the target registration error. With these approaches, it is only possible to assess local deformations. Depending on a use case, this can be sufficient, especially if landmarks are set in the specific area of interest. However, especially for non-rigid registration, many landmarks are necessary, which involves a lot of manual work. Furthermore, it can be challenging to find landmarks in data that shows low contrast or noise in areas of interest. Even though manual work is often used as a gold standard, it adds uncertainty, as anatomical landmarks in medical imaging can be hard to identify [PMEM16, FW01].

For a numerical evaluation of our approach, there are no internal clinical or public data sets available containing unprocessed and ground truth data that represent our problem of intra-patient co-registration. As mentioned, setting landmarks in our MRI data, where we often observe low contrast, is prone to introduce inaccuracies. Transforming a publicly available registered data set from another medical domain has a high risk to not resemble real scenarios well. Therefore, we include visual evaluation methods in our approach.

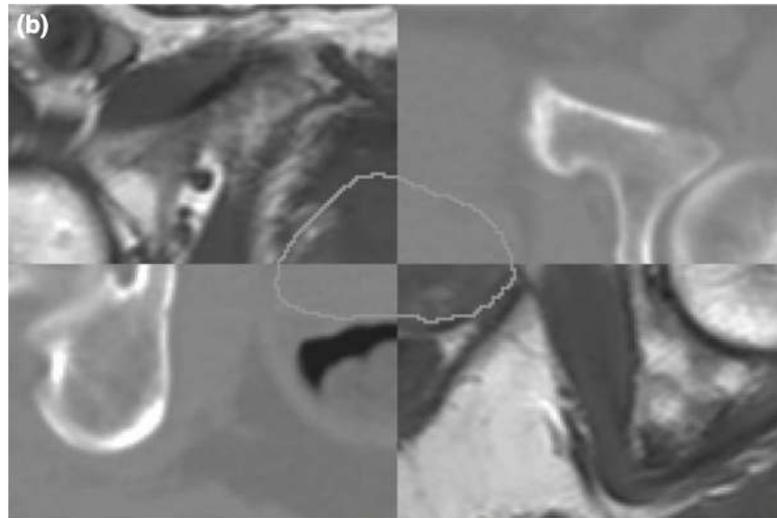
3.3.2 Visual Evaluation

To assess the outcome of image registration visually, several visualization methods exist. Schlachter et al. [SRM⁺19] reviewed published work related to visualization in radiotherapy planning. Image registration is an important part in several steps of radiotherapy planning, where dose calculation per volume unit is crucial for the patient’s health and therapy success. According to the authors, objective evaluation measures for the registration processes are missing, and manual registration leads to inter-rater variability. Visual approaches try to target these challenges and let the user explore and correct noisy registration outcomes interactively.

For their work in deformable prostate MRI and CT registration, Hamdan et al. [HBR⁺17] use checkerboards to visually evaluate the outcome of their method. Hereby, the sequences are shown in a slice-based manner with alternating image patches. The authors enhance those checkerboards with contours (see Figure 3.5). In contrast to simple colored overlays, the advantage of checkerboards is that they are applicable to show similarity even if the gray value range of the modalities differs. However, it is more difficult to assess registration inaccuracies in homogeneous regions [SFJ⁺16]. Cheung et al. [CK09] also use checkerboards. Furthermore, they work with glyph-based 3D visualizations showing the deformation field, as well the deformation grid. To visualize the displacement vector field, Handels et al. [HWS⁺07] color-encode the magnitude of it and display the result on a 3D surface model. This gives information about the amount of deformation, but not about the registration accuracy. Schlachter et al. [SFJ⁺16] provide visualizations for



(a) Rigid Registration



(b) Deformable Registration

Figure 3.5: Checkerboard visualization after rigid (a) and deformable (b) registration, from Hamdan et al. [HBR⁺17].

the assessment of voxel-wise dissimilarity. Interaction and several visualizations allow for detailed exploration of registration accuracy. An example can be seen in Figure 3.6.

In interactive visualization techniques, several magic lens approaches have been proposed to facilitate the exploration of complex data. As introduced by Bier et al. [BSP⁺93], the view on the data is altered within a specific field. Often, this view appears near the current mouse position. Magic lenses have been applied to the medical field, for example in volume visualization in surgical applications [KYWH12] or flow visualization

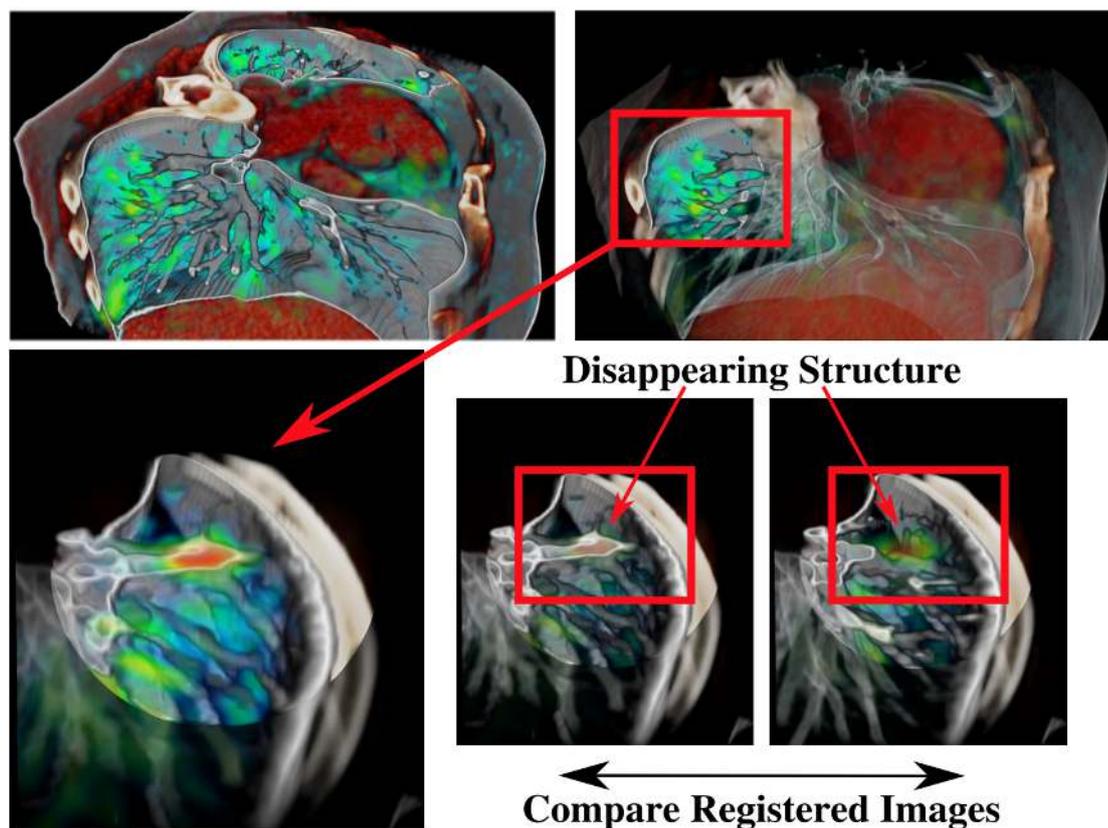
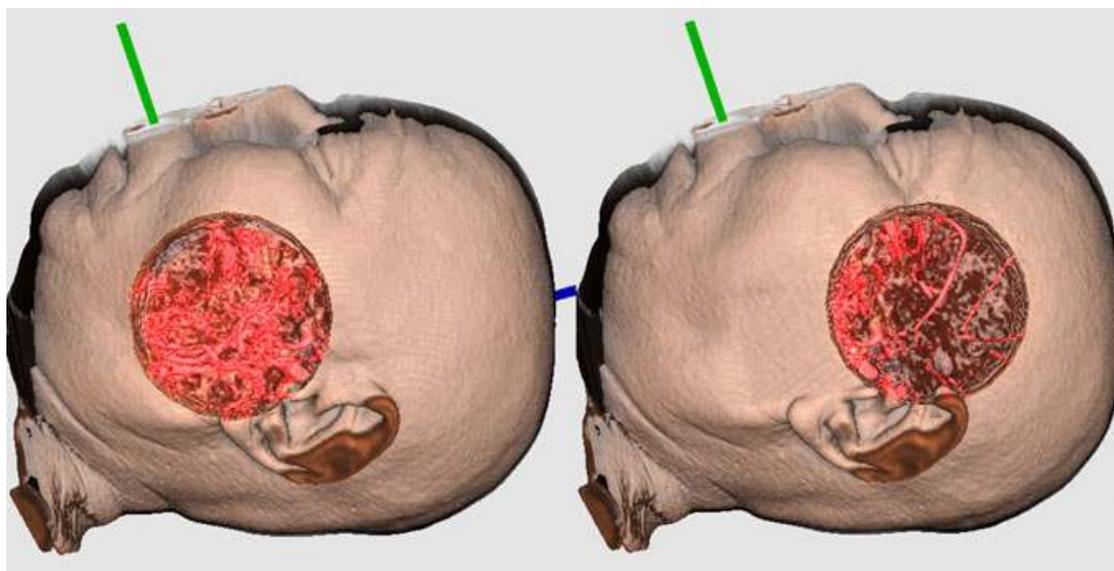


Figure 3.6: Visualizations to assess the accuracy of non-rigid medical image registration methods. Local dissimilarity is shown together with underlying data. The user is able to examine the data by various interaction techniques. Work from Schlachter et. al [SFJ⁺16], figure taken from a later review of Schlachter et al. [SRM⁺19].

for the exploration of blood flow in cerebral aneurysms [GNBPT11]. We show examples in Figure 3.7.

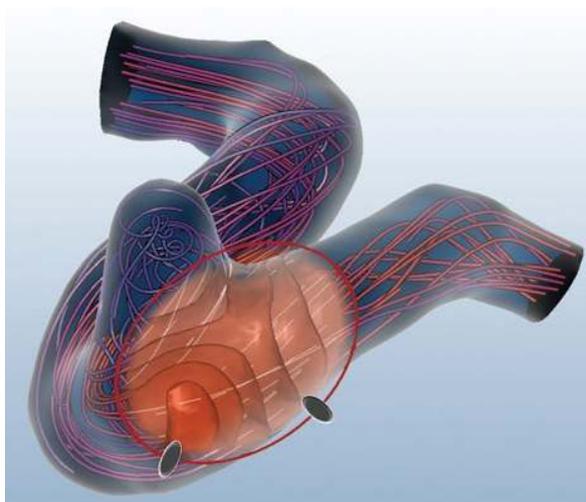
We are working with MRI data with a different histogram range for each sequence. Due to imaging physics, structures like the tumor can appear differently, for example slightly bigger or smaller, even though it represents the same tissue. Therefore, voxel-wise dissimilarity visualization would be difficult to assess. Checkerboards can be used for comparison of two volumes in a slice-based manner. A pre-defined grid can hide relevant details and makes it challenging to assess differences in small areas. For this reason, we combine the idea of a checkerboard with an interactive magic lens to check the registration of multiple volumes. In our literature research, we could not find a magic lens approach that compares two volumes in a slice-based manner to assess misalignments in registration.



(a) Lens with varying transfer function for volume data



(b) Zoom lens for volume data



(c) Contextual lens for flow data

Figure 3.7: Magic lens visualizations for medical data exploration. The position of the lenses changes with the cursor. Inside the lens in (a), a transfer function is used to visualize vessels and soft tissues [KYWH12]. In (b), a circular region is emphasized by a zoom lens [TGK+17]. (c) provides context information for flow data in cerebral aneurysms in form of isolines within the lens [GNBP11].

We introduce the web-based tool MuSIC (Multi-Sequential Interactive Co-registration) to provide radiologists with an intuitive and fast method for interactive multi-sequential co-registration. In this chapter, we present our contributions to the field of co-registration in medical imaging.

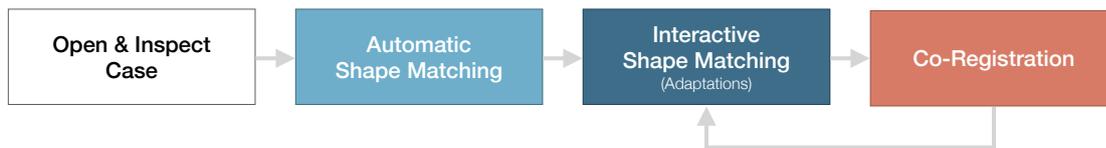


Figure 4.1: Workflow of the proposed tool including the main steps. The automated shape matching aims to find the shape of a given segmentation in all modalities. The user can make adaptations to the proposed position. Based on the potentially varying position of the segmentations in the various modalities, they are co-registered using a landmark-based approach.

Registration workflow We propose the following workflow for our tool (see [Figure 4.1](#)). First, the user loads the different sequences of one case and the corresponding segmentation. The segmentation fits one of the sequences which acts as a reference for the processing steps. Afterwards, the user can call the automatic shape matching, which aims to locate the segmentation automatically in the other sequences. The updated position of the segmentation is shown to the user for every sequence in the tool. In the step of interactive shape matching, the user can make adaptations to the segmentation position regarding translation and rotation. For this procedure, we provide multi-modal visualization support. Once the user has finished making adjustments, the sequences are co-registered. The user can still iteratively adapt the segmentation position if necessary. During this workflow, the user can group modalities that should be processed in the

same way. This is desirable because, as mentioned in [Chapter 2](#), several sequences can be derived from one modality. In our case, [PE](#), [TTP](#), and [AUC](#) are derived from one dynamically acquired sequence and therefore require the same processing steps. In [Figure 5.1](#) of [Chapter 5](#) we show a more detailed workflow together with related decisions for the user and how we represent them in our tool. In the subsequent sections, we present our approach for the following steps: automatic shape matching, interactive shape matching and the co-registration itself.

4.1 Automatic Shape Matching

The automatic shape matching aims to reduce the amount of user interaction by moving the segmentation to the boundaries of the tumor in each modality and, thereby, providing a first suggestion of the actual tumor location. It is based on the minimization of a cost function. Parameters that can be adapted to find the global optimum are the translation of the segmentation mask in x, y and z direction, as well as the rotation angles of the mask around each of the three axes. The user can afterwards adjust the proposed results.

For the automatic shape matching, we describe

- the *data preparation* and preprocessing steps,
- the *optimization procedure* based on simulated annealing, and
- our *cost function* used for optimization.

4.1.1 Data preparation

Data preprocessing is necessary to limit the search space and reduce computational cost. Furthermore, we need to enhance the contrast of tumor boundaries since [MRI](#) data is noisy and might not show a high contrast around the tumor itself, while other surrounding structures like the uterus or the bladder might be more prominent. This highly depends on the medical image characteristics of the individual sequences. We visualize the preprocessing steps in [Figure 4.2](#).

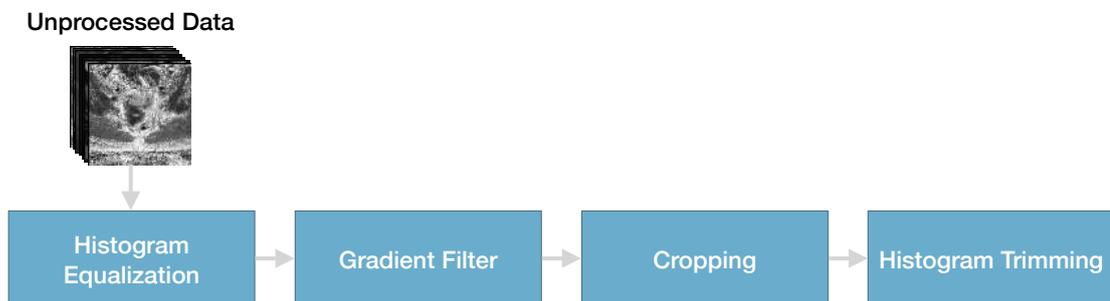


Figure 4.2: Preprocessing steps for the automatic shape matching.

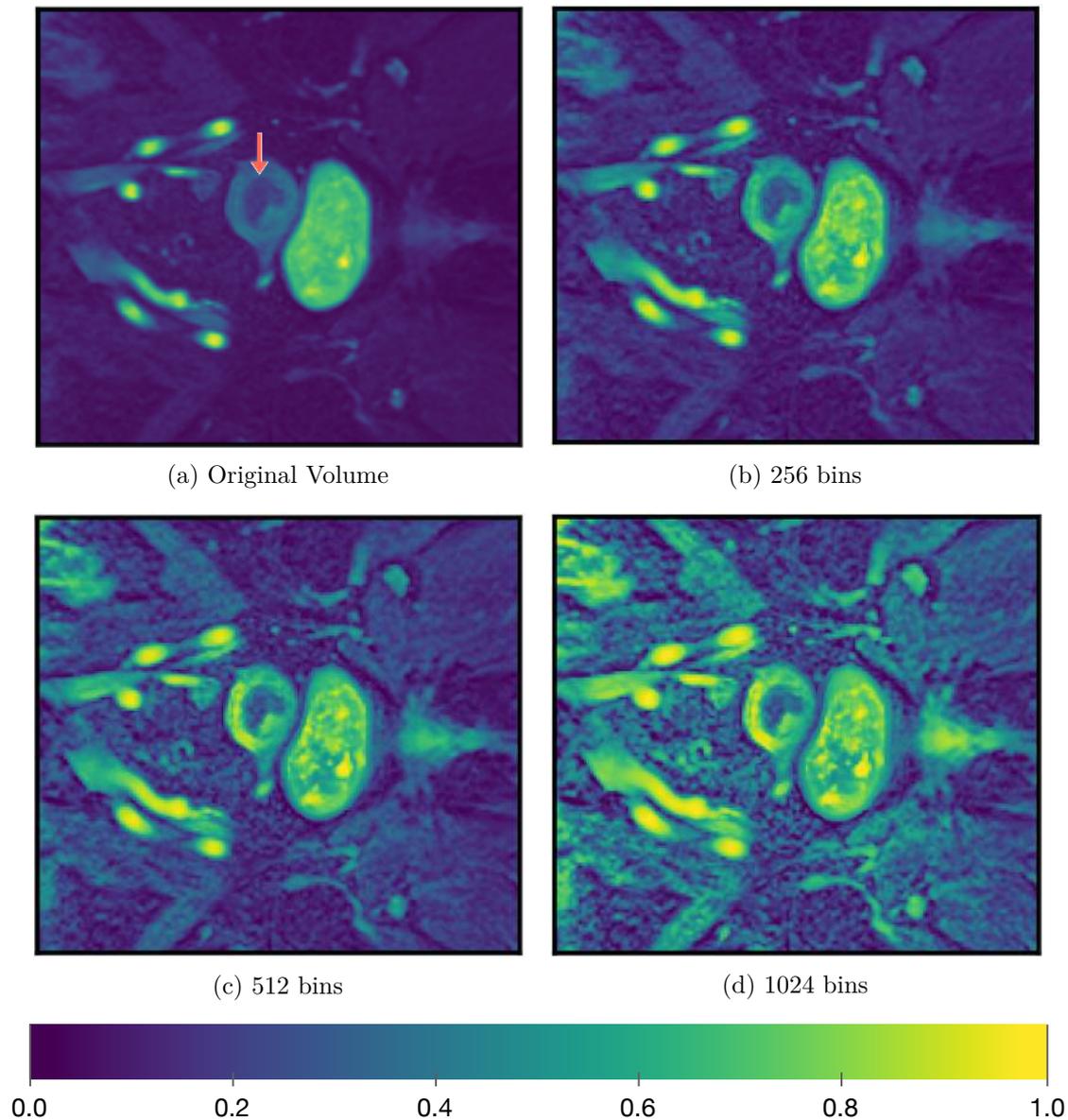


Figure 4.3: Comparison of adaptive histogram equalization with different bin sizes using scikit-image [vdWSN⁺14]. (a) shows the original PE sequence of a volume with an orange arrow indicating the tumor boundary. In (b), (c) and (d) we can see the processed volumes with a different number of bins used in the algorithm. The contrast is enhanced which also leads to a higher noise level.

Histogram equalization and normalization Adaptive histogram equalization can be used to locally enhance the contrast in medical imaging [PAA⁺87]. The algorithm acts in an area surrounding a voxel by transforming it to an intensity value that is related

to its rank relative to the surrounding gray values. This makes it possible to emphasize the contrast in regions even if they are darker or brighter than the main image area and to distribute the histogram more evenly over the whole gray value range. We apply the `equalize_adapthist` algorithm of scikit-image's [\[vdWSN+14\]](#) exposure package and normalize all modalities to a range between 0 and 1. Per default, the algorithm uses a kernel size of $k = \frac{1}{8}width \cdot \frac{1}{8}height$ of the image. As number of bins, we choose 512 over the default of 256 which results in a good trade-off between enough contrast in the tumor area and little additional noise. We show an example in [Figure 4.3](#).

Gradient filter Our aim is to find the tumor based on the gradient in the volume since the outline of a tumor determines its shape. Therefore, we apply SciPy's [\[VGO+20\]](#) [Gaussian Gradient Magnitude \(GGM\)](#) filter. It computes the gradient magnitude

$$\|\Delta f\| = \sqrt{\left(\frac{\delta f}{\delta x}\right)^2 + \left(\frac{\delta f}{\delta y}\right)^2 + \left(\frac{\delta f}{\delta z}\right)^2}$$

of a 3D image function f using Gaussian derivatives. We chose the parameter σ , which describes the width of the Gaussian filter, to be 1 voxel. Lower σ values lead to edgy results, while higher values make the image blurry and the tumor not recognizable.

Cropping The location of the tumor can deviate only in a relatively small range compared to the whole volume, since the patient position in the scanner does not change drastically during the scans and also the internal organ movements are within a few centimeters range. Therefore, we crop the volumes around the tumor position in the reference sequence. We do this by determining the minimum and maximum index belonging to the segmentation mask in each of the x, y and z direction and adding 15 voxels to the search space in each direction. We determined the size of this search space together with our collaborating radiologists. Larger displacements of the tumor due to organ movements are not common.

Histogram trimming In some of our sequences, the contrast between tumor and uterus can be very low. To optimize the workflow for our given data sets and to improve the automatic shape matching, we further emphasize the tumor boundary. Therefore, for each modality we define two values to trim the histogram of the equalized volume to a specific range in order to better distinguish tumor and endometrium. The values are set after a short visual analysis of the value range between tumor and endometrium.

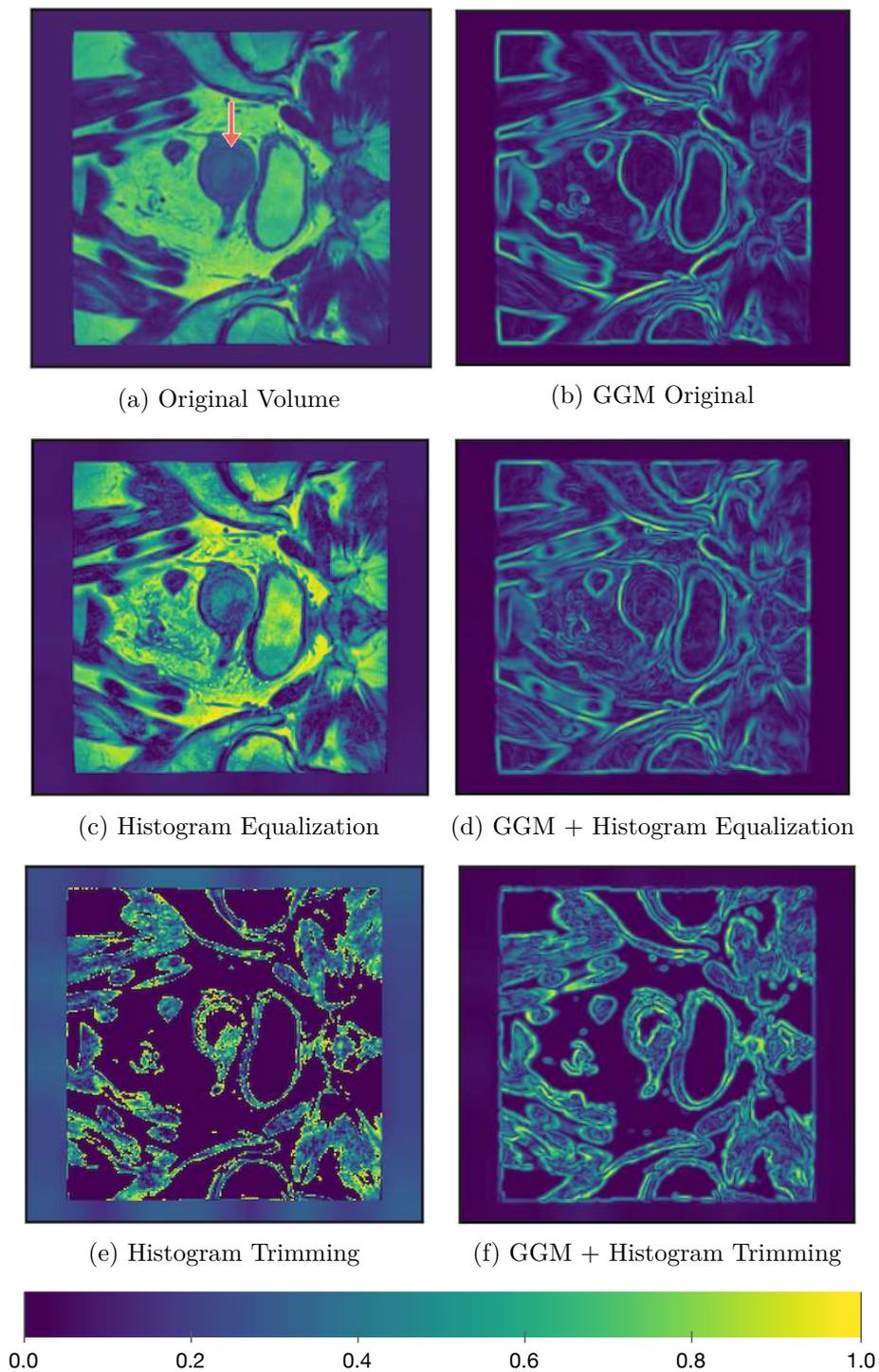


Figure 4.4: Comparison of preprocessing steps for the automatic shape matching. (a) shows a slice of the original volume from a T2 sequence. The orange arrow points to the tumor boundary which is hardly visible in this case. In (c) we can see the histogram equalized volume, where the contrast is higher also around the tumor border. (e) shows the volume after histogram trimming. (b), (d) and (f) depict slices of the volumes to the left processed with a Gaussian Gradient Magnitude (GGM) filter.

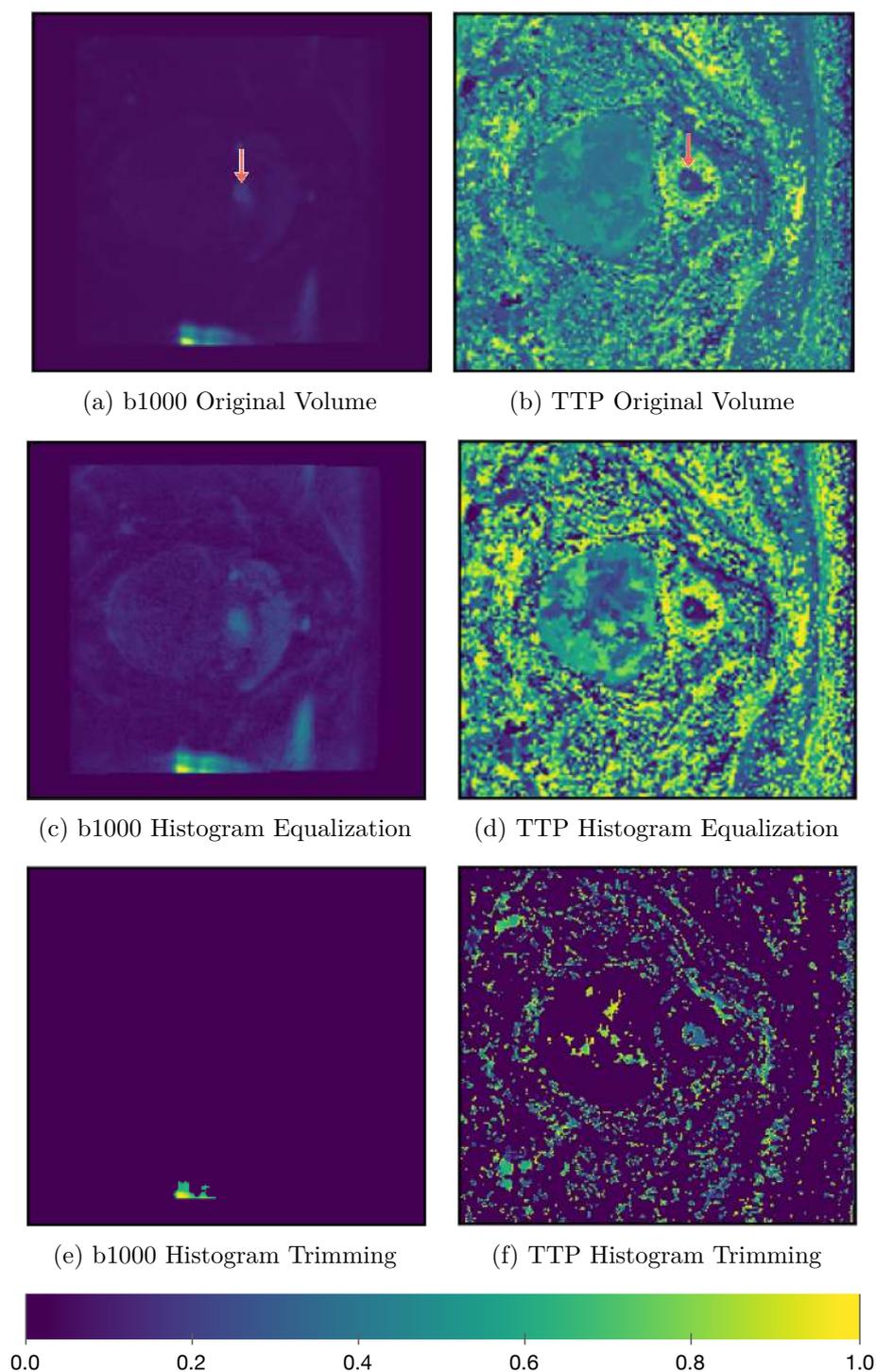


Figure 4.5: Difficulties of the pre-processing for automatic shape matching. (a), (c) and (e) show the original and processed b1000 volumes of one case. The orange arrow indicates the tumor boundary. The processing steps leading to good results for other cases have less effect here, since the sequence has a different histogram distribution than other b1000 sequences in the data set. (b), (d) and (f) visualize the TTP sequences of the same case. Here, higher imaging noise makes it difficult to emphasize the tumor.

Difficulties The previous steps emphasize the contour of the tumor in many cases of our data clearly. However, the scanning techniques of the modalities can vary a lot and if the acquired image deviates significantly from the medium range of image values, the preprocessing described above does not lead to the desired outcome. One of our available cases, for example, has a different histogram distribution and the volumes appear darker. Also if the volumes are very noisy, the procedure struggles with emphasizing the tumor boundary while making it distinguishable from other structures. We show examples of these cases and what the histogram equalized and histogram trimmed versions of the volumes look like in [Figure 4.5](#).

Combined processing of sequence groups To further enhance data quality for the shape matching and to provide the same solution for sequences with the same biological basis, we add those sequences to a combined volume. We equalize the histograms as described above and smooth the volumes with a Gaussian filter, for which we set $\sigma = 2$ voxels to sufficiently reduce noise. Higher σ values lead to a too strong blurring of the tumor border. Afterwards, we add the gray values at each position and normalize the newly created volume afterwards. This step has also the advantage that we exploit the contrast characteristics in the different modalities and achieve an overall bigger contrast. In these cases, we do not trim the histogram.

4.1.2 Optimization Procedure

To find the best fit for the location of the segmentation mask in a volume, we search for the optimum in a cost function. Two main directions for optimization algorithms are optimization based on the gradient and direct search methods. The former calculate the derivative (first or higher order) of a cost function in order to retrieve a direction for the next step. For complex cost functions, which are common in medical imaging, this can be computationally very expensive and often results in local optima. Direct search methods compute the next step based on the cost or energy function directly, without the need to calculate derivatives [\[MZ04\]](#).

One possibility to optimize cost functions is [SA](#). Introduced by Kirkpatrick [\[KGV83\]](#), the method is derived from metallurgic annealing. The cost or energy function is represented by the temperature of a material. In the beginning, the temperature is high, and the whole range of the search space can be explored. With repeated steps and depending on an acceptance criterion of the current parameter setting, the temperature decreases. The steps get smaller, supporting an exploitation of a local search space. In general, the exploration of the search space depends on a probabilistic component. Hereby, a *worse* solution than the current one might be accepted, making it possible to escape local minima. This is in contrast to the classical gradient descent approach.

We visualize the basic idea of the algorithm in [Figure 4.6](#). The following components need to be specified in advance [\[KGV83, MZ04\]](#):

Initialization The initialization involves the determination of an initial temperature T_0 , parameters and their bounds, as well as the number of iterations i_{max} . Since the initial temperature together with the cooling algorithm can have an impact on the results, Ben-Ameur [BA04] examined possible strategies to set the initial temperature based on the acceptance criterion. In practice, however, often an experimental setup is used, since the initialization is very use-case-specific. Initial parameters can be estimated or set to a random value. The number of iterations depends on the temperature, cooling schedule, and the size of the search space. It should be high enough to allow to possibly explore the whole space. For optimization, it can be reduced in an experimental setting or based on an evaluation scheme. In the approach depicted in Figure 4.6, i is initialized as 0 and increased iteratively until the maximum number of iterations is reached.

Computation of new solution We compute a new possible solution based on a cost function that we describe in Subsection 4.1.3. For this computation, we need a new set of parameters. To compute the initial solution, the parameters of the initialization are used. When entering the loop, the new parameters are determined based on the parameters of the current solution, a random component, and the current temperature T_i to further explore the search space. If the new solution is better than the current solution, the current solution is updated. If it is even better than the best solution, also the best solution so far is updated.

Acceptance criterion If the new solution is worse, which means that it has a higher cost than the current solution, it can still be accepted. The Metropolis criterion [MRR⁺53] is used to determine whether a new solution is accepted as current solution or not, based on the current cost J . If $-\Delta J \geq 0$, which means if the cost of the current solution minus the cost of the new solution is greater than or equal to zero, a random number ω from a uniform distribution is derived. If $\omega \leq e^{-\Delta J/T_i}$, the new solution is accepted and the current solution is updated. T_i is the temperature for the current solution [MZ04].

Cooling schedule A cooling schedule involves the calculation of a temperature update. Several update rules are possible, for example, based on a constant c between 0 and 1, where the temperature $T_{i+1} = cT_i$. Furthermore, the temperature does not have to be updated for each iteration step. In the algorithm in Figure 4.6, we update the temperature if the current solution is updated. As mentioned earlier, it can be quite complex to design an appropriate cooling schedule and it depends, among other criteria, on the size of the search space, the desired accuracy, and the computation of the cost function.

Stopping criterion In our scheme, a maximum number of iterations defines the stopping criterion. Instead, an early stopping criterion could be used. Common choices are, for example, the ratio of rejected vs. accepted solutions, or an initially calculated minimal temperature.

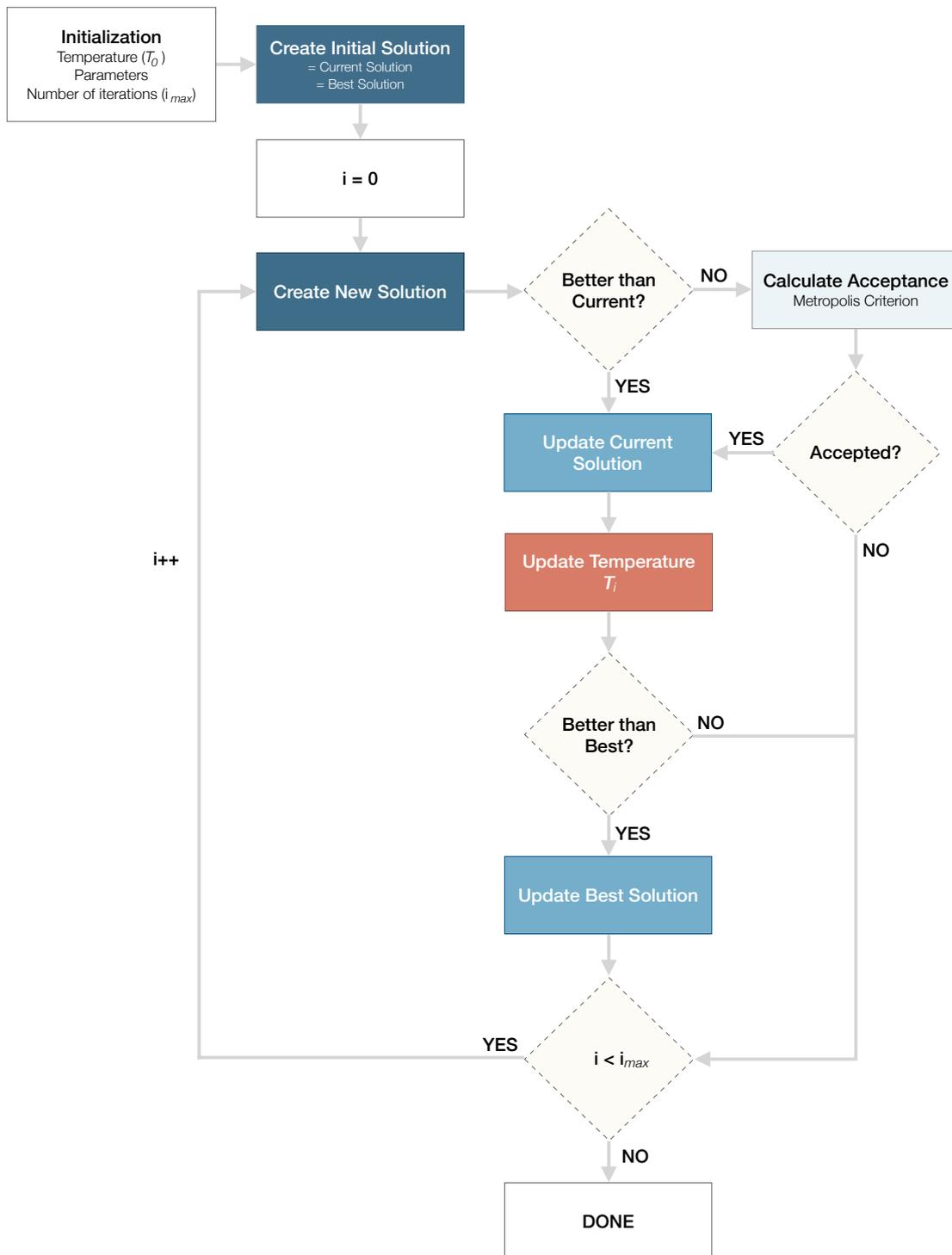


Figure 4.6: Simulated Annealing algorithm with the number of iterations as stopping criterion. Following [MZ04].

Dual Annealing We used the approach described above to determine the initialization components and to find out how the algorithm behaves in terms of the given cost function. For the integration in our application, however, we use SciPy’s [VGO⁺20] Dual Annealing implementation. It is based on the work of Xiang et al. [XG00, XSFG97] and integrates the fast SA [Tsa88] with the classical SA [TS96]. With this approach, they allow for a more detailed exploitation of local search spaces by introducing a local search at accepted solutions of the global search. Initially, we set all the transformation parameters, which means both translation and rotation parameters of the segmentation mask, to zero to compute an initial solution. We work with data that has been pre-registered based on the image header, which is saved with the image during the acquisition and contains information regarding the positioning of the patient in the scanner. Therefore, the tumor in the different sequences should be at least in the closer surroundings with respect to the reference sequence (VIBE). The bounds contain one minimal and one maximal possible value for each parameter that is used to compute new solutions. The minimum and maximum translation parameters are computed based on the cropped bounding box with the 15 voxels margin around the segmentation. The minimum and maximum rotation parameters are set to -30 and 30 degrees, respectively. The size of the search steps is characterized by a distorted Cauchy-Lorentz visiting distribution. A parameter (in SciPy’s approach called the *visit* parameter) defines the height of the peak of the distribution, where high values allow for larger steps. Large values cause a heavier tail of the function. Also the temperature update rule depends on the *visit* parameter. We set the initial temperature to 1000 after trying different values, and SciPy’s *visit* parameter to 1.5. Both values are higher by default (5230 for the initial temperature and 2.62 for the *visit* parameter), which allows for bigger steps in the search space. Since for our use case it is beneficial to have small steps, we decrease the temperature and *visit* parameter. However, with too small values, the algorithm gets trapped in local minima.

4.1.3 Cost Function

The cost function should have its optimum where the segmentation mask best fits the image content in terms of rotation and translation parameters. We define a good fit as a high and homogeneous gradient in the sequence at the borders of the segmentation mask. In order to find this optimal position of the segmentation, we need to calculate the gradient magnitude at the borders of the segmentation. To calculate the cost for a given set of transformation parameters, we compute the following steps:

1. **Transform the segmentation mask** with a given transformation parameter set \mathbf{p} determined by the optimization algorithm. We can adapt six transformation parameters: Three for translations along each of the x, y and z axis, and three for rotations around each of the axes.
2. **Compute the segmentation hull indices** from the transformed segmentation mask. We get the binary segmentation hull by adding an outer and inner hull. Subtracting the transformed segmentation mask from a dilated version of it (kernel

size is set to 3x3x3) results in the outer part. Applying erosion with the same kernel size to the mask and subtracting the original transformed version yields the inner part. It is important to include both parts in the hull as, due to imaging physics and pre-processing, the tumor can appear slightly smaller or larger in the different modalities.

3. **Extract the gray values of the GGM filtered sequence** at the positions of the binary segmentation hull indices. This step corresponds to setting the nonzero values of the segmentation hull to the corresponding values of the GGM filtered sequence.
4. **Smooth the resulting volume** with a Gaussian filter to allow for some inaccuracies. σ is set to 1.5 voxels. We will refer to the smoothed volume as $G_{\mathbf{p}}$, where \mathbf{p} corresponds to the translation and rotation parameters used to transform the segmentation for the hull computation. The volume represents a smoothed version of the segmentation hull filled with values from the GGM filtered sequence at the corresponding indices.
5. **Weigh the smoothed volume $G_{\mathbf{p}}$** with the gray values of the GGM filtered sequence data. This step increases the difference between low and high magnitude values and, therefore, increases the difference between the cost of a well-fitted segmentation (mainly high values) and a segmentation that fits only on some edges, as the lower values are even more down-weighted. We will refer to the weighted volume as $W_{\mathbf{p}}$, which represents a weighted version of the segmentation hull filled with values from the GGM filtered sequence.
6. **Compute cost function** as described below.

We define the cost function as follows:

$$J(\mathbf{p}) = -2 \cdot \frac{\sum W_{\mathbf{p}}}{n} + \sigma(W_{\mathbf{p}}),$$

where n is the number of nonzero voxels of the weighted hull $W_{\mathbf{p}}$. \mathbf{p} is the parameter vector, consisting of $t_x, t_y, t_z, \theta_x, \theta_y, \theta_z$ as translation and rotation parameters for each axis. In the first term, we compute the sum of the gray values in the weighted segmentation hull $W_{\mathbf{p}}$ divided by n . We want the first term to be high to find a minimum of the cost function in a region with high gradient magnitude at the segmentation border. σ is the standard deviation of the weighted segmentation hull $W_{\mathbf{p}}$. This term of the cost function is responsible for the homogeneity of the voxels inside the weighted segmentation hull. It should prevent the optimization procedure to get stuck in a minimum where only some parts of the hull have very high values. This is often the case at the border of the endometrium and the bladder, as those two structures show a high difference in gray values and, therefore, also a higher gradient magnitude than between tumor and endometrium. Without this term, the segmentation tends to wander to that border. For this reason, high σ values act as a penalty for non-homogeneous segmentation hulls. In

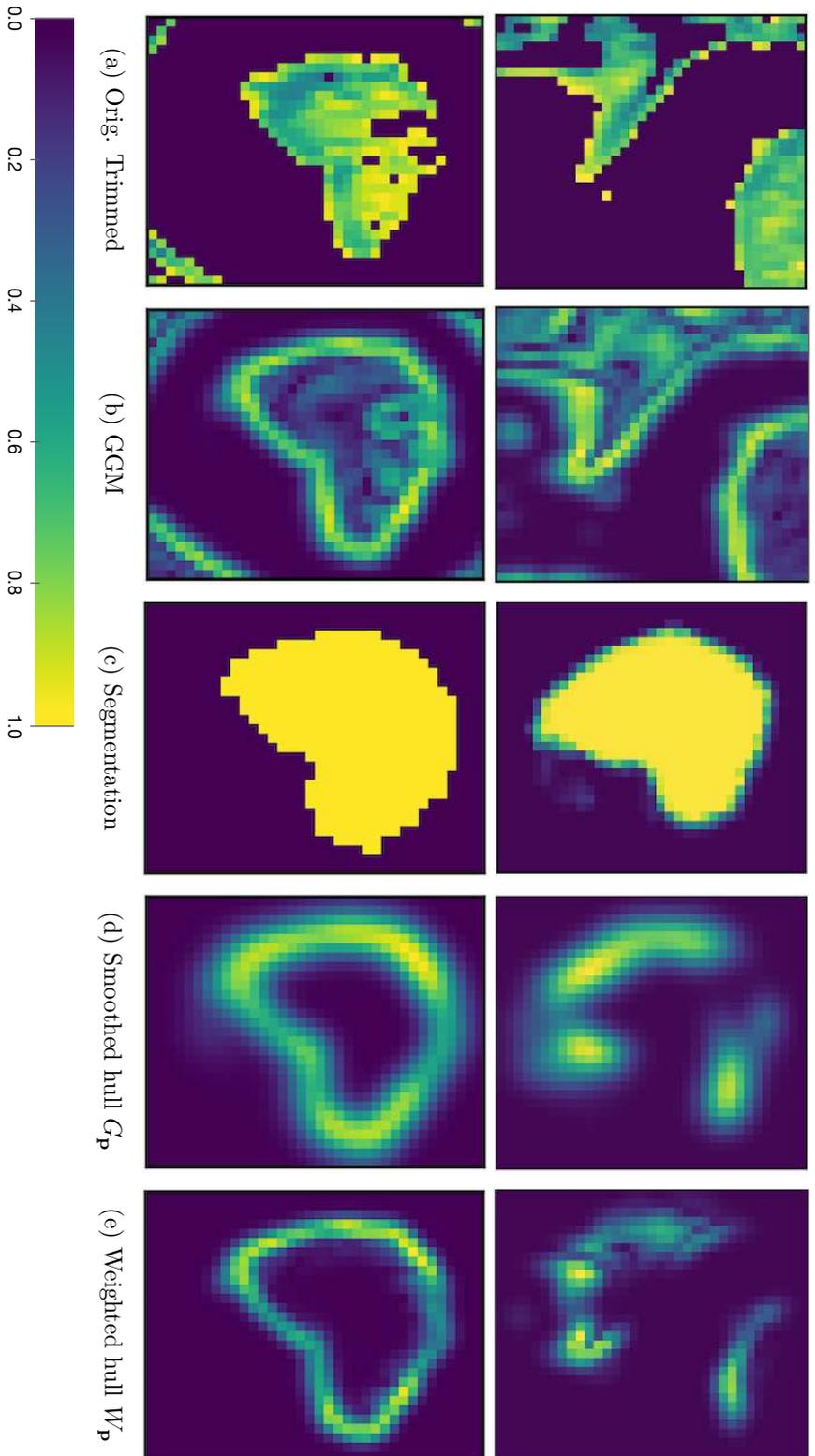


Figure 4.7: Example steps of the cost function computation in two scenarios. In the example in the top row, the segmentation is poorly aligned, which becomes visible in the discontinuous weighted hull W_p (e). In the second row, the segmentation is well aligned and the weighted hull W_p contains high and continuous values. (a) and (b) show the trimmed original and **GGM** filtered trimmed original in the area of the transformed segmentation. (c) depicts the segmentation mask. (d) visualizes the smoothed segmentation hull G_p computed from the segmentation in (c) and the image values of (b). (e) shows the segmentation hull W_p weighted by the gradient magnitude values of the volume in (b). The computed cost for the scenario in the upper row is $0.3869e^{-2}$. For the second row, it is $-0.1782e^{-2}$.

theory, both the first and second term are in the range between 0 and 1. We weigh the first term with a factor of two, since it is usually smaller than the value of the standard deviation. This is probably due to the fact that many voxels of the segmentation hull can have relatively small values (since we smooth it with a Gaussian filter and therefore make it greater to account for inaccuracies). The goal is to minimize the cost function. The resulting cost values can be both positive or negative. A better fit is only characterized by a smaller cost.

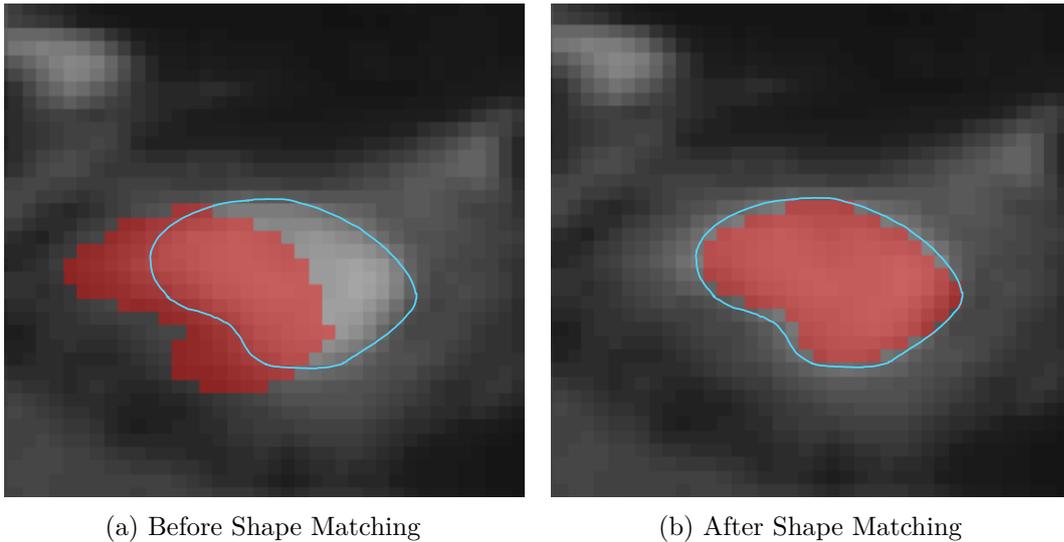


Figure 4.8: Example outcome of the shape matching in comparison to original segmentation mask position in a b1000 sequence. (a) shows the sequence with overlain segmentation before the shape matching. (b) visualizes the transformed segmentation mask with the same underlying sequence. The tumor in the volume is represented by the slightly brighter structure that we delineated with a bright blue outline. We can see that the segmentation mask fits the shape of the tumor better after the shape matching.

We depict example outcomes of the iterative steps in the cost function in [Figure 4.7](#). We show one scenario where the segmentation is not aligned well with the tumor boundaries, and one where the segmentation is in an optimal position. In [Figure 4.8](#), we show a comparison between the position of a segmentation mask in an example case before and after the automatic shape matching.

4.2 Interactive Shape Matching

After the automatic shape matching, the user can check the quality of the proposed position and eventually adapt it. For those adaptations, we provide simple user interaction to fit a segmentation mask to the tumor boundary. The user can translate and rotate the given segmentation to a desired location where the segmentation mask matches the tumor

boundary. Since we want to offer a web-based approach to allow for easy integration into hospital systems, we use WebGL for slice and segmentation mask rendering.

4.2.1 WebGL Shader

Slice shading In computer graphics, shaders are programs that define rendering effects for scenes to visualize them on the screen. In the beginning, we used a slice based shader [Thr19] provided by Three.js [Dan12] which allows to scroll through a volume and display one slice as a plane at the current depth in the data. We can use the shader twice: first for displaying the MRI volume, and second to display the segmentation mask as an overlay with user-defined opacity.

When it comes to user interaction, the shader works fine for only translation in x and y direction on the screen. However, the data is displayed as a plane in the 3D camera space. Rotating the 2D plane in the camera field results in a rotation of the current slice (see Figure 4.9) instead of a rotation of the segmentation volume. It would be possible to use a ray casting approach to render the volumes in 3D. However, radiologists commonly work with 2D slice views in the different slicing directions. Therefore, we decided to create a new shader to allow for translation and rotation of the segmentation in 3D while displaying a slice-based representation of the segmentation. To the best of our knowledge, there is no public web-based shader available that allows for the desired user interaction.

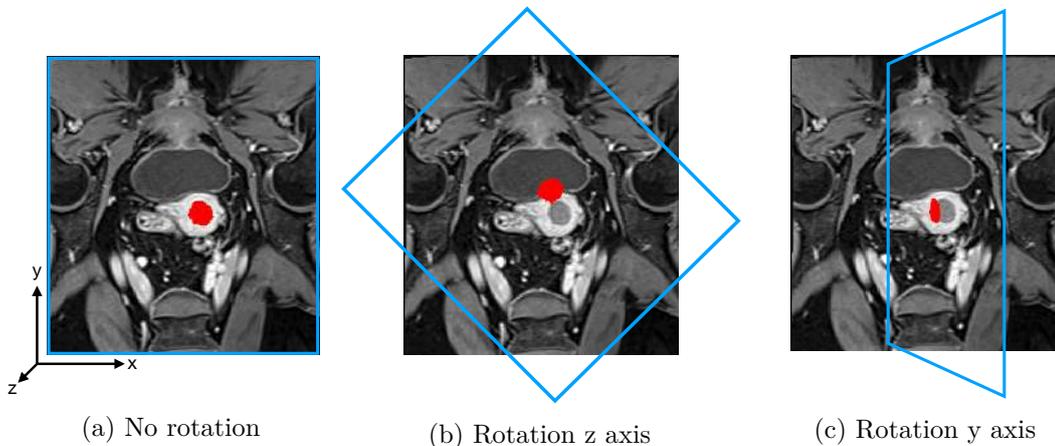


Figure 4.9: 3D rotation problem of a plane-based shader approach. We see a slice in a coronal view of the VIBE sequence in the background and the according tumor segmentation in red rendered on top. The blue boundaries indicate the plane on which the slice of the segmentation volume is rendered. Not rotating the plane (a) and the rotation around the z axis in the plane center (b) does not result in problems. Rotating around the y axis (c) rotates the plane with the current segmentation slice in the camera field instead of rotating the segmentation volume and showing the updated slice.

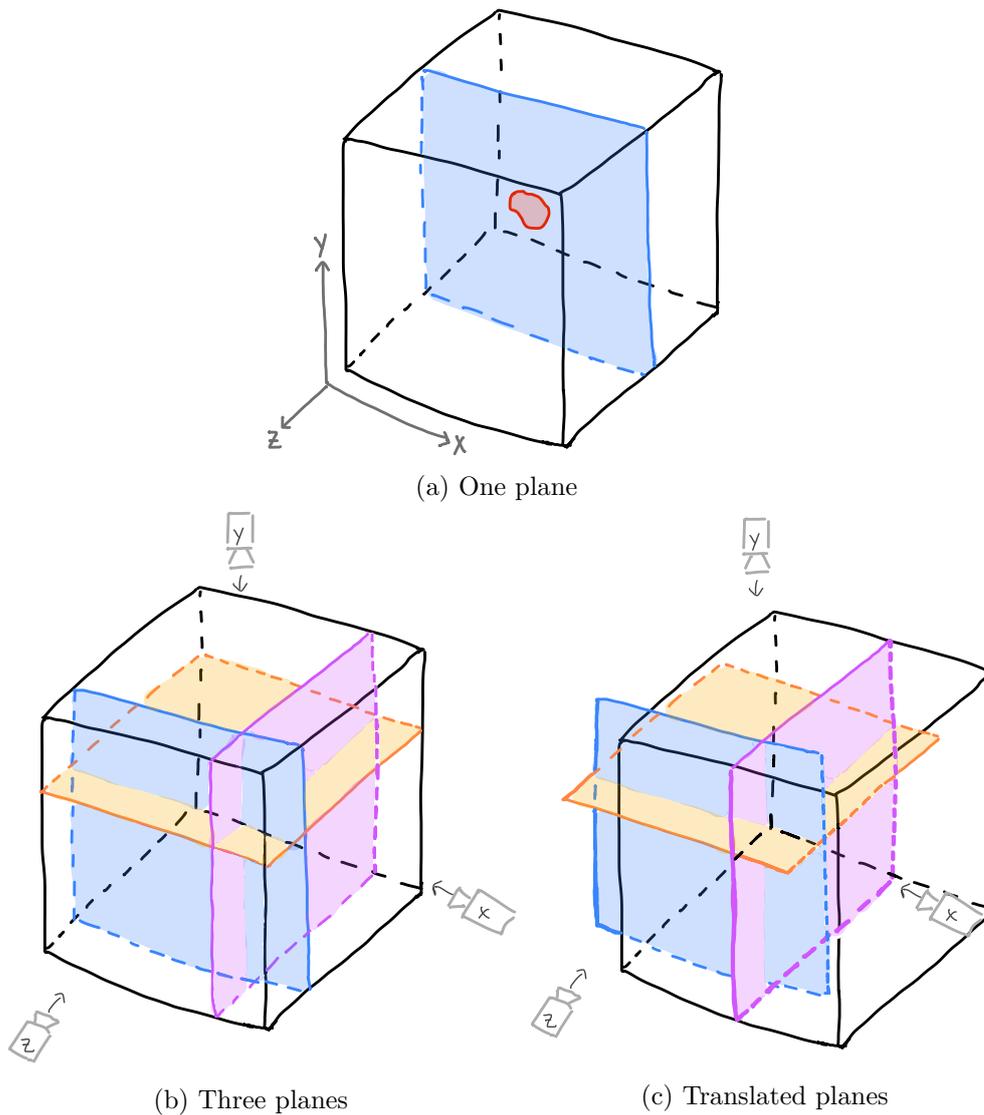


Figure 4.10: Sketch-based visualization of the intersections between the data volume, here represented as a cube, and (a) one plane and (b) three planes perpendicular to the three slicing directions. In (c), the planes are translated on the x axis to the left. The cameras in (b) and (c) visualize virtual cameras that define from which side we are looking on the data cube and what we show on the screen. Each camera is perpendicular its corresponding plane.

Custom shader In order to provide the desired functionality, we need to create a custom shader. We explain details about the implementation with WebGL in [Chapter 6](#). Here, we will focus on the main concept and user interaction to transform a segmentation mask and obtain a correct representation of the tumor volume in a 2D slice view. The

basic idea is to show the intersection between a plane and the data volume on the screen. We have three slice views per sequence: x, y and z, where z corresponds to the axial-oblique direction in which the MRI images in our data set have been acquired. Therefore, we need three different planes to show the three slicing directions respectively. The way the planes intersect with the volume is depicted in Figure 4.10. Since we want to show the intersection between the planes and the volume, we can imagine having three virtual cameras that define what we see on the screen. They are always perpendicular to their plane, also if the planes are translated or rotated.

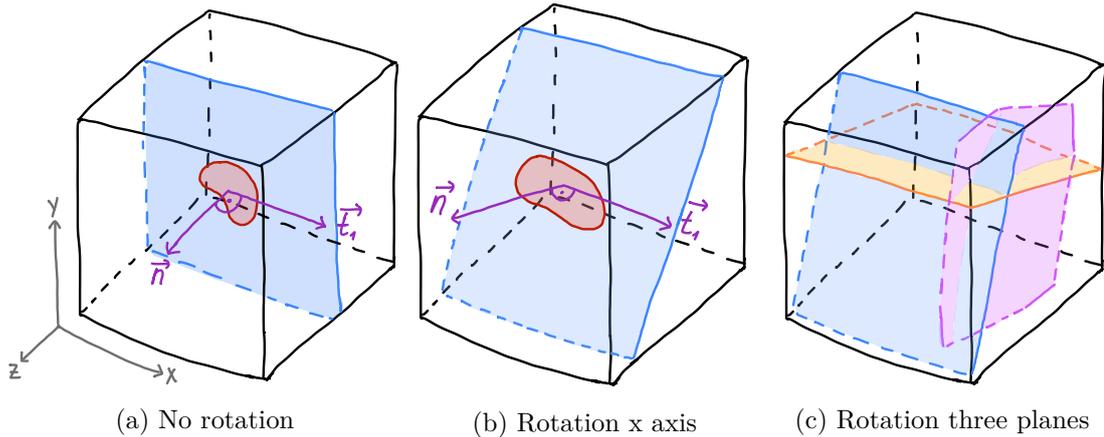


Figure 4.11: Sketch-based visualization of plane rotation as intersection with the data volume. (a) shows one plane perpendicular to the z axis and the intersection with the segmentation volume. \vec{n} is the plane normal, and \vec{t}_1 one of the two tangents laying in the plane, perpendicular to the plane normal. (b) shows the plane rotated around the x axis. (c) visualizes how the other two planes depend on this rotation.

4.2.2 User Interaction

The user can transform a segmentation on the screen by interacting with the planes. Here, we explain how the user interaction affects the positioning of the planes. In general, we allow for both translation and rotation along all different axes in each slice direction with the keyboard.

Translation The user can translate the mask with the four arrow keys on the keyboard. A translation in one slice view has different effects on the mask representations in the other slice directions. For example, if the user wants to move the segmentation 5 voxels to the right on the screen in the frontal view, the planes are moved accordingly to the left on the x axis. On the screen, this appears as the same change of the segmentation to the right for the view represented by the orange plane (looking from the top). However, the segmentation view represented by the pink plane changes the depth and we see another slice. We show how the position of the planes change on user translation in Figure 4.10. There is a difference between the translation of the segmentation mask, which changes

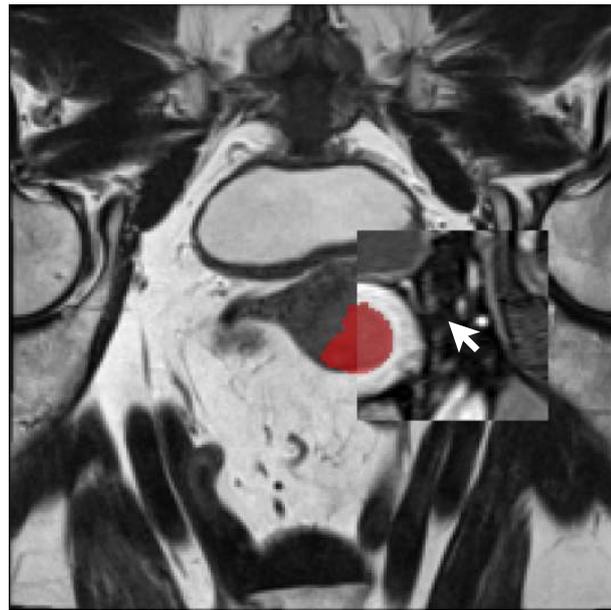
the position of all three planes, and scrolling through the slices, which moves only one of the planes in the corresponding depth.

Rotation The same concept regarding the behavior of the planes on user interaction applies to rotation. The inverse rotation of the plane reflects the transformation of the segmentation initiated by the user. The rotation of one of the planes is adopted by the others. We visualize the change of the planes in an example case of rotation in [Figure 4.11](#). We rotate the blue plane around the x axis with respect to the rotation center, which lies in the origin of the two vectors \vec{n} and \vec{t}_1 . Therefore, the rotation around the x axis corresponds to the rotation around \vec{t}_1 . This transformation is also applied to the pink and orange plane.

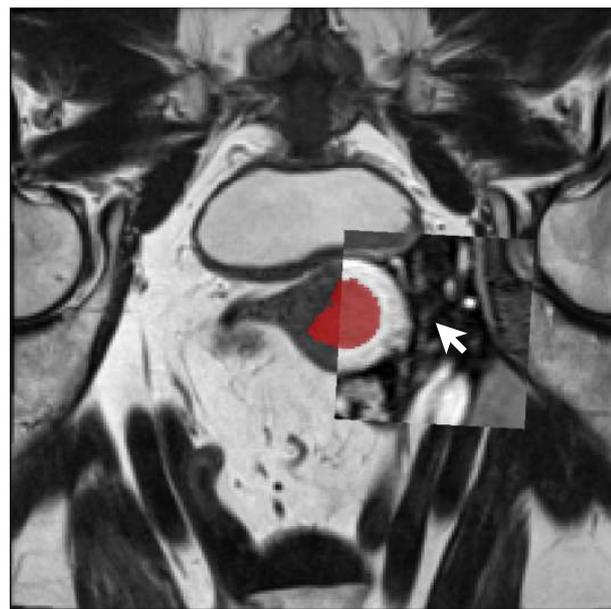
4.2.3 Assessment of Position Accuracy and Future Alignment

During the interactive shape matching, we want to provide a visualization method to assess the alignment based on the current segmentation position. Medical experts usually still have to compare different sequences by having them side by side, which requires good imagination skills and focus [\[RvdHD⁺15\]](#). To facilitate the comparison between two sequences, multi-modal visualization techniques can be used. We introduce an interactive magic lens [\[BSP⁺93\]](#) to allow for assessing the alignment of the tumor and surrounding structures. A square is rendered on top of a sequence showing the data of another specified sequence, for example the user-determined reference modality. It becomes active on keyboard interaction at the current cursor position. The user can change the position of the lens by moving the cursor to compare different structures while the lens is active. Furthermore, scrolling through different slices is still possible with an active lens visualization. We show the magic lens in [Figure 4.12](#). In contrast to a checkerboard or a transparent overlay of the whole volume, which are often applied to assess registration accuracy [\[HBR⁺17\]](#), [\[CK09\]](#), the magic lens offers an interactive and localized way to compare two volumes. Furthermore, we include the segmentation mask as overlay with the user-set opacity in the magic lens to assess the alignment of the segmentation in comparison to both sequences. The coordinates of the reference sequence that are displayed in the lens update based on the positioning of the segmentation mask. Before transforming the segmentation mask, the user can compare the original volumes. During the search for a good segmentation fit, the magic lens updates the translation and rotation of the displayed reference view. Thereby, it becomes possible to assess the future alignment of the two volumes after a full rigid co-registration already during the interactive transformation of the segmentation mask. We will describe the user interface and further possible user interactions in detail in [Chapter 5](#).

Magic lenses have been used in several medical applications [\[KYWH12\]](#), [\[TGK⁺17\]](#), [\[GNBPI1\]](#). However, there is no approach providing a magic lens for the assessment of misalignment in registration that already gives indications about the registration results during the interactive search for an appropriate alignment.



(a) Before transformation



(b) After transformation

Figure 4.12: Magic lens before and after segmentation transformations in a T2 sequence with VIBE as reference modality. The lens view updates based on the segmentation position, here with an x and y translation and a slight rotation around the z axis.

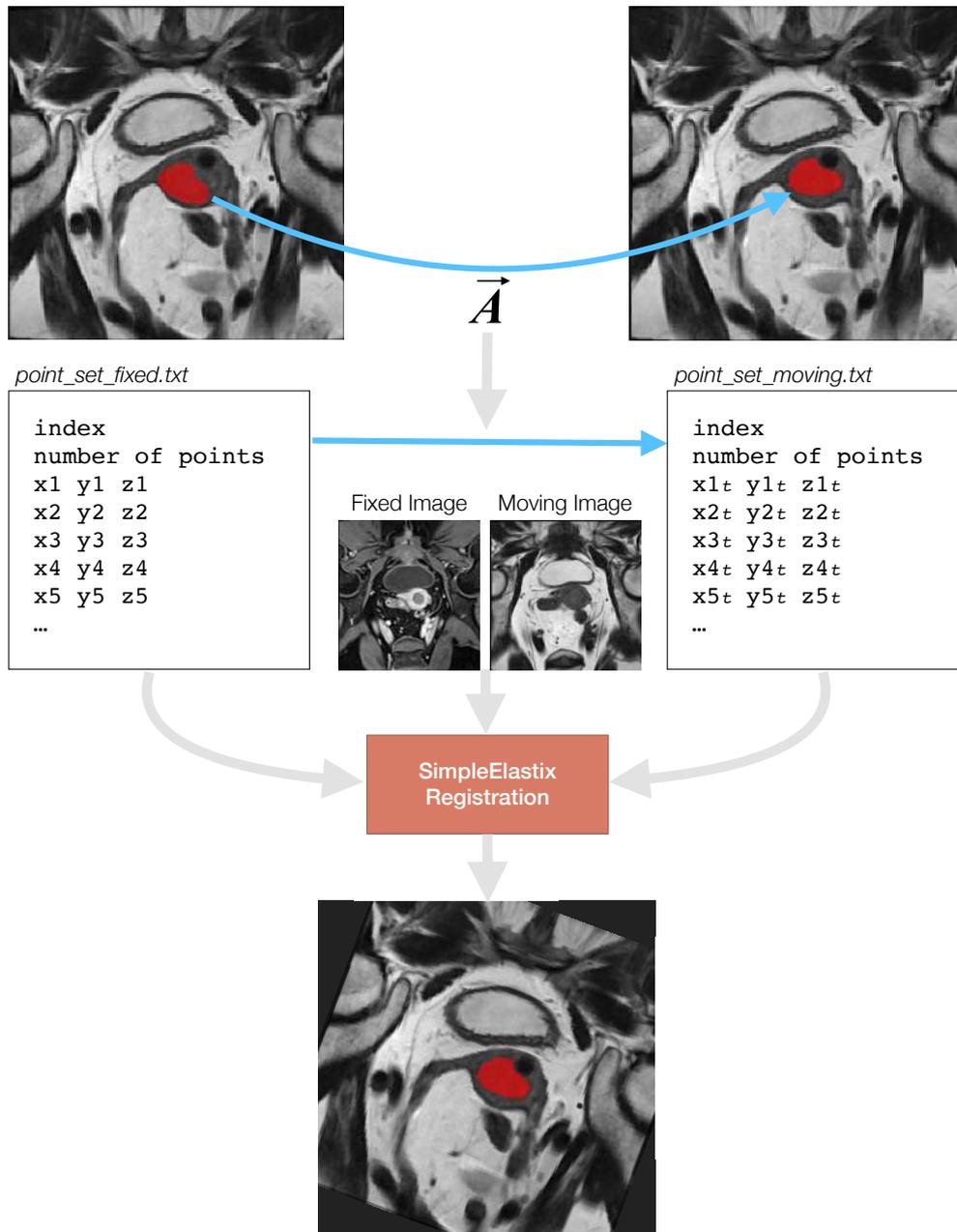


Figure 4.13: Process for the registration with Elastix [KSM⁺09]. The user transforms the segmentation to the desired position. The resulting transformation matrix \vec{A} , comprising the rotation and translation coefficients, is used to transform the landmarks of the fixed image to obtain the landmarks for the moving image. The landmarks of the fixed image are the indices where the values of the untransformed segmentation volume are equal to 1. We pass the two generated files containing the original and transformed landmarks to the Elastix module, which registers the currently processed sequence to the reference sequence based on the corresponding landmarks.

4.3 Co-Registration

To register the selected volumes in the web application, we use a landmark-based approach. The basic idea is to use the indices of the segmentation mask, namely the coordinates where the segmentation data equals 1, as landmarks for the registration process. The not transformed, original indices of the segmentation represent the landmarks in the fixed reference modality, while the user-adapted transformed segmentation indices are the corresponding landmarks of the moving modality. In this way, we avoid major deformations in the tumor area during registration. We use the Python library SimpleElastix [MBSK16] of the Elastix [KSM⁺09] framework for the actual registration and visualize the approach in Figure 4.13.

Elastix [KSM⁺09] is a command line tool that offers, among many other methods, landmark-based registration. After the user transforms the segmentation to the desired position that matches the underlying tumor border, we extract the transformation matrix \vec{A} that represents the transformation of the segmentation mask. We apply \vec{A} to the landmarks of the fixed modality. Both the fixed and corresponding transformed landmarks are passed to Elastix with the reference modality and the modality that should be transformed.

Components and settings Elastix is an elaborated framework that offers a large number of settings to customize the parametric registration process based on the specific use case. In general, it defines the registration problem including a cost function C depending on a parameterized transform \vec{T} with parameters μ as

$$\mathbf{T}_\mu = \underset{\mathbf{T}_\mu}{\operatorname{argmin}} C(\mathbf{T}_\mu; I_F, I_M), \quad \text{with}$$

$$C(\mathbf{T}_\mu; I_F, I_M) = -S(\mathbf{T}_\mu; I_F, I_M),$$

where I_F, I_M are the fixed and moving image, and S is the similarity measure. The parameter values μ depend on the registration approach. For example, there are three rotation and three translation parameters for 3D rigid registration. \mathbf{T}_μ is then the transform that maps the fixed image I_F to the moving image I_M . To define the settings, we followed the Elastix manual [KS18] describing the single steps and components. Figure 4.14 shows the components of the Elastix registration process.

Metrics Mutual Information (MI), `AdvancedMattesMutualInformation` in SimpleElastix, is a similarity metric that is suitable for multi-modal registration problems. SimpleElastix offers also Normalized Mutual Information (NMI), but it is not optimized in terms of performance. Other commonly used metrics are the Mean Squared Difference (MSD), which, however, assumes an equal intensity distribution from both images. This means the images have to be from the same modality. Normalized Correlation Coefficient (NCC) is recommended for images with linear intensity distributions. Kappa Statistics

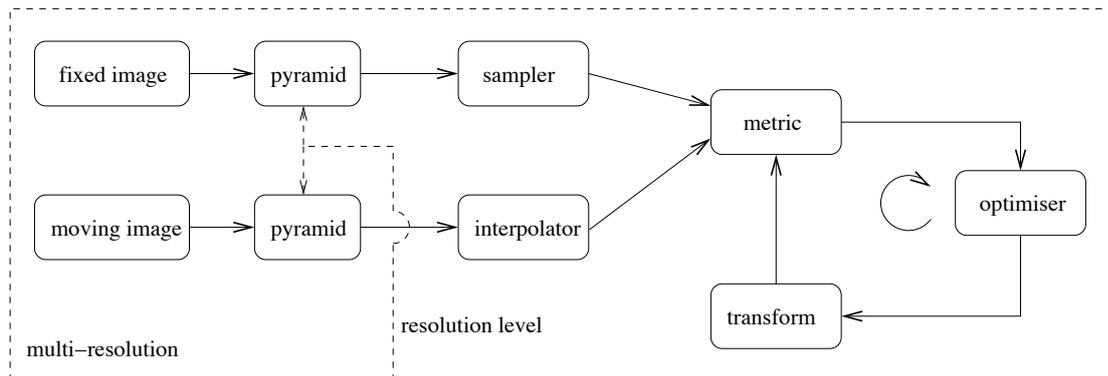


Figure 4.14: Registration components of Elastix [KSM⁺09]. From the Elastix manual [KS18].

(KS) can be used to measure the overlap between segmentations. A different measure, S_{CP} , is based on corresponding points, which minimizes the distance between two sets of corresponding points. It is defined as

$$S_{CP} = \frac{1}{P} \sum_{\mathbf{x}_{Fi}} \|\mathbf{x}_{Mi} - \mathbf{T}_{\mu}(\mathbf{x}_{Fi})\|,$$

with P as the number of points \vec{x}_i and \vec{x}_{Fi} , \vec{x}_{Mi} as corresponding points of the fixed and moving modality. We use a combination of the corresponding points metrics and Mutual Information.

Samplers Image samplers can avoid to calculate the similarity metrics for all voxel indices (full sampler) to reduce computational cost and, therefore, registration time. Elastix offers grid, random, and random coordinate sampling, next to the full sampler. Grid sampling is equivalent to downsampling. Both the random and the random coordinate sampler use a certain number of voxels that can be determined by the user in advance. The difference between them is that the random coordinate sampler is not constrained to real voxel positions, but also in-between-voxel positions can be used. We choose random coordinate sampling, as it can make the cost function smoother and easier to reach the optimum [MBSK16]. Additionally, we set the parameters `NewSamplesEveryIteration` and `UseRandomSampleRegion` to true. If leaving the default value 'false' for the latter parameter, the sampler uses samples from the whole data domain. Changing it to 'true' enables the function to choose samples in a neighbourhood around the currently processed voxel, according to a uniform distribution. We follow the recommendations and set the size of the neighbourhood (`SampleRegionSize`) to one third of the smallest value of the image size, resulting in a similarity measure that is computed locally.

Interpolators Elastix uses non-voxel positions to evaluate the transformations of the moving image in the optimization process. The options are nearest neighbour, linear,

and N-th order B-spline (`BSplineInterpolator`). While the first two options are fast, a B-spline interpolation often results in better quality outcomes. At least $N=3$ is recommended for good quality and reasonable speed, which is also our choice.

Transforms A transform maps the coordinates of the fixed image to the new coordinates in the moving image. We have the choice between translation, rigid, similarity, affine, B-splines, and thin-plate splines transform. Rigid (Euler) transforms include rotation parameters in addition to translation. Similarity transforms additionally scale, while affine transforms also apply shear to the moving image. B-splines are non-rigid transforms that use a control point grid to perform local transformations along the points, depending on another parameter that determines the number of control points in the surroundings influencing the current control point. Also thin-plate splines are a form of non-rigid transformations and are kernel-based. The transformation is composed of an affine and a non-rigid component and is computed with respect to corresponding landmarks in both images. In this approach, however, only the landmarks in the fixed image are provided by the user. The landmarks in the moving image initially are set to the fixed image's landmarks and adapted iteratively. We use a combination of rigid and B-splines transforms.

Optimizers Examples for possible optimizers to minimize the cost function determined by the similarity metric are quasi-Newton, nonlinear conjugate, gradient descent, and Robbins-Monro approaches. The latter two are recommended and suitable for most registration problems. The Robbins-Monro approach approximates the derivatives of the cost function, making it faster, but including potential errors due to the approximation. For those two methods, the setting of a gain factor α is necessary that weighs the negative gradient of the cost function to define the search direction. It can be difficult to tune and is very problem-specific. Since we want the tool to be used also for other data sets, we use `SimpleElastix AdaptiveStochasticGradientDescent`, which estimates an initialization for α .

Multi-resolution Two ways of initially reducing the complexity and computation time of a registration are mentioned in the Elastix manual [KS18]. One is to work with Gaussian image pyramids to smooth and/or downsample the images gradually. This option makes it possible to first match larger structures and refine the registration step by step for smaller structures. We use the default settings of the image pyramids. It can be also helpful to initially constrain the registration in the degrees of freedom. This is possible for example by using a rigid transformation before a B-spline registration to start the non-rigid part of the registration with better starting configuration. Also gradually adapting the B-spline grid complexity can be useful.

Considering the described setting options, we defined our registration approach based on segmentation indices as landmarks. We use multiple metrics per registration call and start with an initial rigid registration by only matching the corresponding landmarks with the `CorrespondingPointsEuclideanDistanceMetric`. For our data set, where

the largest images have a shape of (232, 256, 88) voxels, this step is very fast and can be computed in a few seconds per modality. Furthermore, for improving research in radiomic tumor profiling, our collaborators would like to ideally also match the close surrounding of the tumor, which in our case is the uterus. The uterus can slightly change its shape based on bladder filling and movement of the intestines. To account for small inaccuracies and a slightly different endometrial shape, we perform B-splines registration as a second step. To avoid deformation of the tumor, we also support this step with the same landmarks. We combine the two metrics `AdvancedMattesMutualInformation` and `CorrespondingPointsEuclideanDistanceMetric`. We can weigh the components for the cost function and set a lower weight to the B-splines (0.5) compared to a default value of 1.0. This ensures that the position of the points has a higher weight to avoid strong deformations.

Masking We have to assign a weight for the deformable registration metrics. Setting a weight implies that the Mutual Information metrics has an influence that theoretically could still cause deformations in the voxels of at landmark positions. Even though these deformations are not likely to occur, we further constrain the deformable registration to the area outside the tumor by using binary registration masks. We set the mask to 1 for voxels that are outside the tumor segmentation mask, and to zero otherwise. This mask limits the sampler to select voxels only in the defined area outside the tumor.

Grouping In our data set, several sequences are derived from one other sequence and represent the same biological basis. Next to a multi-metric strategy, `SimpleElastix` allows for registration with both multiple fixed or moving images. We use the feature of `SimpleElastix` to register the grouped modalities determined by the user as multiple moving images to the single reference modality. With a multi-metric and multi-image approach, the cost function changes to:

$$C(\mathbf{T}_\mu; I_F, I_M) = \frac{1}{\sum_{i=1}^N \omega_i} \sum_{i=1}^N \omega_i C_i(\mathbf{T}_\mu; I_F^i, I_M^i).$$

C_i defines the single sub-metrics and I_F^i, I_M^i the various single and moving images.

According to the Elastix manual [\[KS18\]](#), using multiple metrics and/or multiple images can help yielding better registration results and being more robust. Another advantage can be the reduced computation time of this approach.

User Interface

With MuSIC (Multi-Sequential Interactive Co-registration), we provide a web-based tool for multi-modal interactive co-registration based on a given segmentation mask that matches one of the sequences. Furthermore, we propose a workflow to perform co-registration for multi-sequence endometrial [MRI](#) data with our web application. This workflow is depicted in [Figure 5.1](#). We will describe and visualize the single steps in this chapter based on a use case of one subject of our data set. As described in [Chapter 2](#), we have seven sequences available with a segmentation mask for the [VIBE](#) sequence, created either manually or with an [ML](#) algorithm. The other sequences have been only aligned based on the information of the patient’s position in the scanner provided by the header information. This step transforms the sequences into the same space defined by the scanner. However, the alignment based on header data does not account for other factors, for example internal organ movement. Therefore, the quality of the alignment is poor.

[Figure 5.2](#) shows the tool after loading the sequences and the segmentation data. The user interface is composed of three different parts. The left toolbar contains different interactive components related to the workflow steps. The main view in the middle offers views for three different slicing directions of one volume. In our data set, the biggest view of the three slicing directions shows the sequence sliced in an axial oblique direction, in which the volumes have been acquired originally. The right grid view shows the other selected modalities in an axial oblique slicing direction.

Toolbar The toolbar (see [Figure 5.3](#)) contains several components. In the *Data* tab, we have buttons to upload and save data. The *Slice View* tab provides interaction possibilities with the slice views like zooming, panning, and contrast adjustment. Furthermore, it provides information regarding the current position in the volume represented by the slice numbers for each slicing direction. In the *Segmentation* tab, the user can adapt the segmentation opacity with a slider. When changing the position and rotation

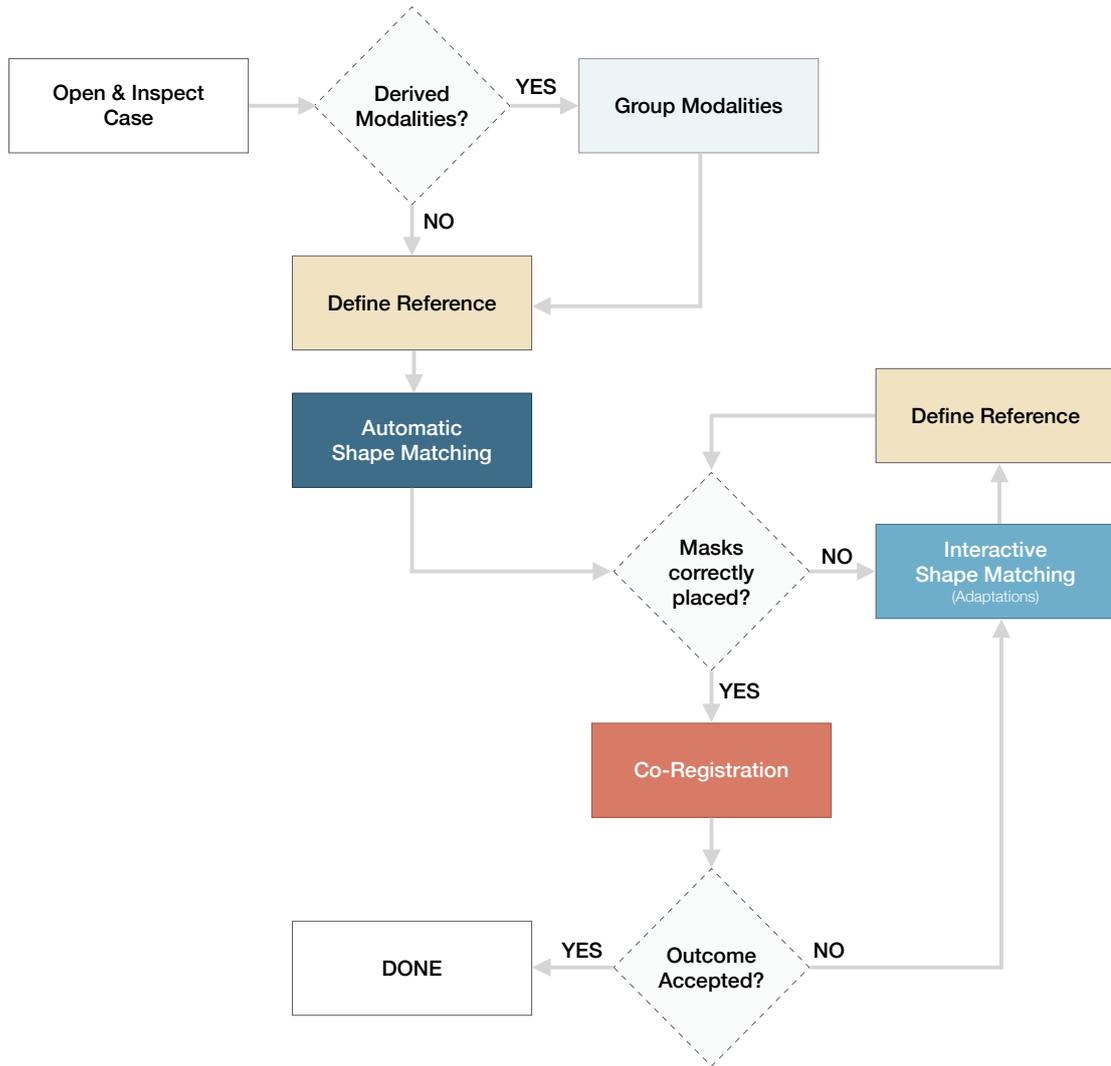
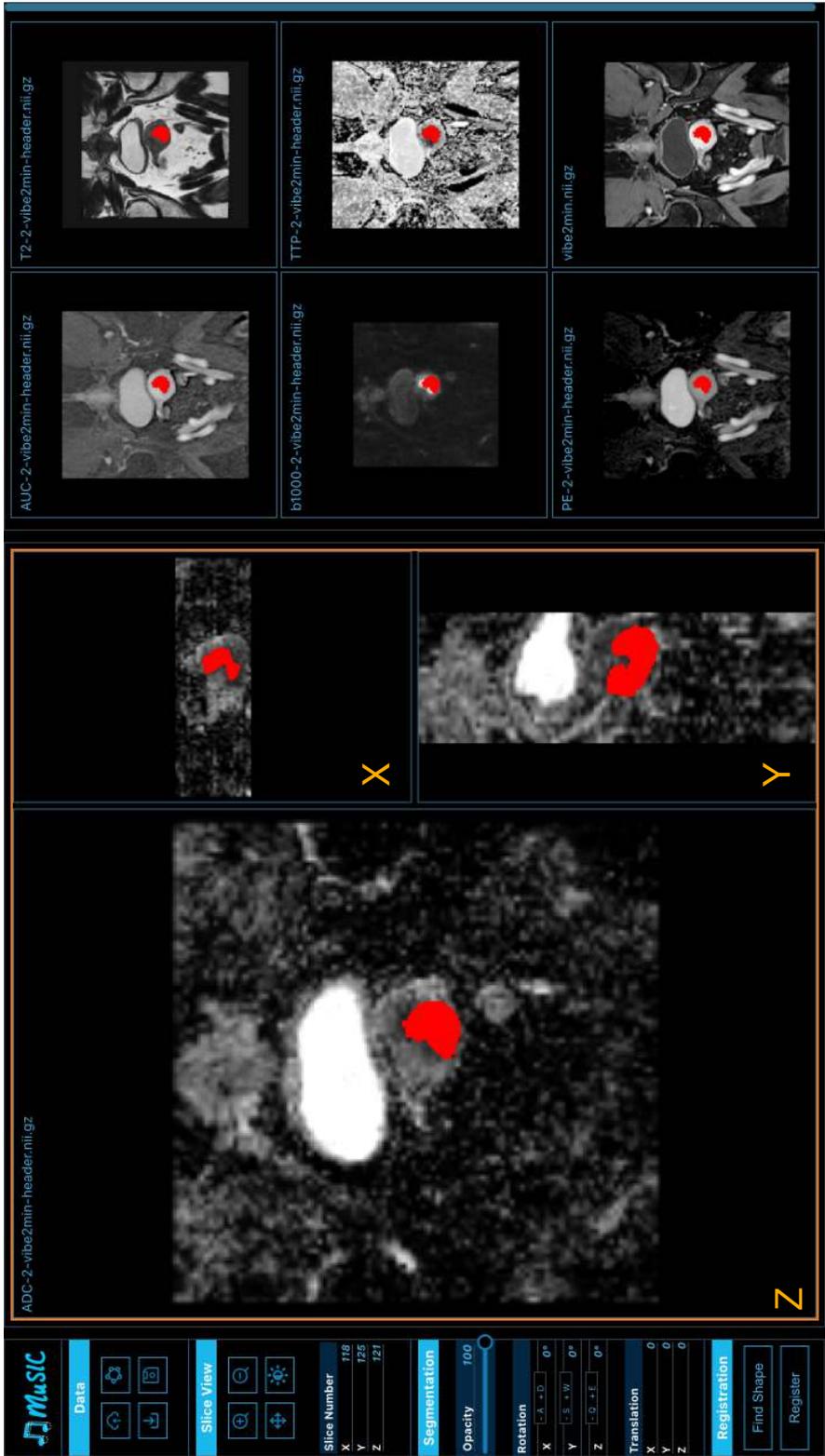


Figure 5.1: Detailed workflow diagram of the proposed tool.

of the segmentation, the two corresponding sub-tabs provide information about the transformation applied in the current main view. The information is depicted for the biggest view of the three slicing directions (axial oblique) and stored for each sequence, even when changing the main view. In case of rotation, we provide hints for the keys used for rotation around each axis in positive and negative direction. The *Registration* tab contains buttons to start the automatic shape matching and to perform the final registration after the adaptation of the segmentation masks.

Open and Inspect Case With the upper left button of the sidebar in the data tab (see [Figure 5.3](#)), the user can select and upload up to seven different volumes. With the



Toolbar

Main Slice View

Grid View

Figure 5.2: Screenshot of the **Multi-Sequential Interactive Co-registration (MuSIC)** application. The left side shows a toolbar with possibilities to load data, interact with it, and start the shape matching and registration algorithm. In the middle, we have a main slice view in three slicing directions. The grid view on the right shows the other modalities.

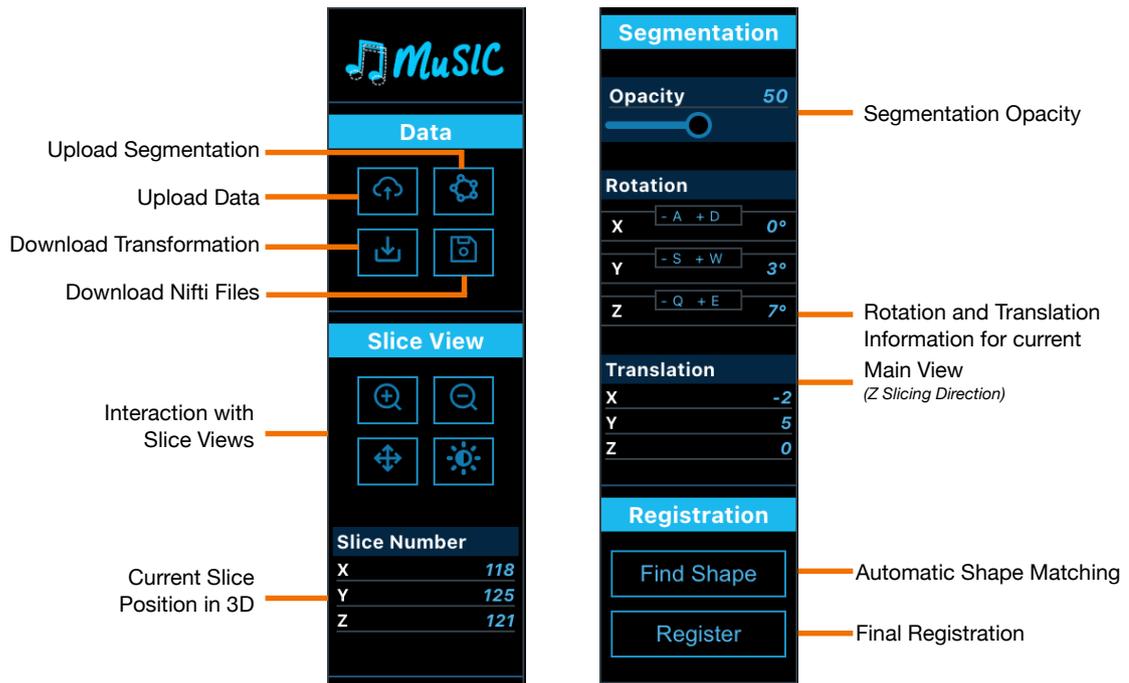
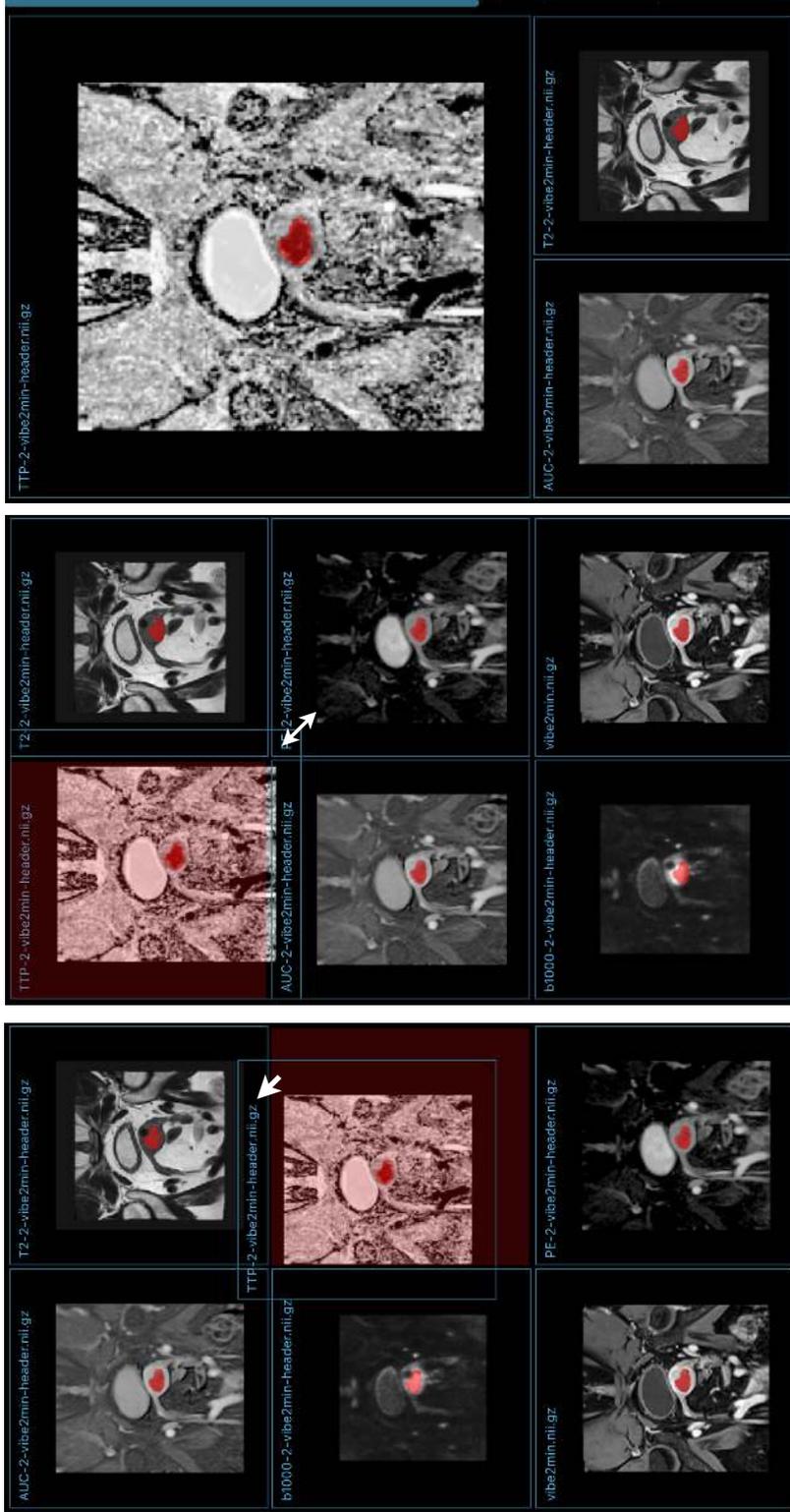


Figure 5.3: Elements of the MuSIC toolbar (here divided into two pieces).

button next to it on the right, one can upload also a segmentation volume. Loading the selected volumes and segmentation mask results in a view similar to [Figure 5.2](#). To inspect the case, the user can interact with the slice views either by the provided buttons, or with the mouse. Scrolling up or down changes the slice number of the slice direction where the cursor is currently located. Scrolling in one sequence adapts the slice depth of the other sequences as well. Pressing the right mouse button and moving the mouse up or down results in zooming in and out, respectively. When pressing the mouse button in the middle, the user can change the position of the slice in the view, which is especially helpful with a high zoom factor to find the region of interest. We designed these interactions based on the currently used tool in the research PACS (Picture Archiving and Communication System).

Custom Layout We allow the user to adapt and customize the layout of the grid view (see [Figure 5.4](#)). Single views can be changed in position and size by mouse interactions to make the user interface more flexible. To change the position of a view, users can simply interact with it by drag and drop interactions. To make a view bigger and smaller, the user can click and drag the bottom right corner of a view container. A dark red background always indicates position and size of the view when the mouse button is released. If the layout gets bigger than the pre-defined layout, a scrollbar is automatically added to the view.

Furthermore, the user can change the sequence that is shown in the big main slice view.



(a) Changing position

(b) Adapt size

(c) Position and size adapted

Figure 5.4: The customizable layout of MuSIC. In (a), the user begins to change the position of the TTP modality by dragging the viewport with the mouse to a desired position. The red background shows where the view would be placed if the user released the mouse. (b) shows the beginning of a resizing interaction with the same sequence, where the user drags the lower right corner to make the view bigger. (c) shows the result of the interaction.

When hovering over a view, a button with a symbol indicating expansion appears on the top left of the view, as shown in [Figure 5.5](#). Clicking on it switches the selected sequence with the sequence in the main view. This is advantageous during the adaptation of the segmentation location because it allows for a view in all three slicing directions.

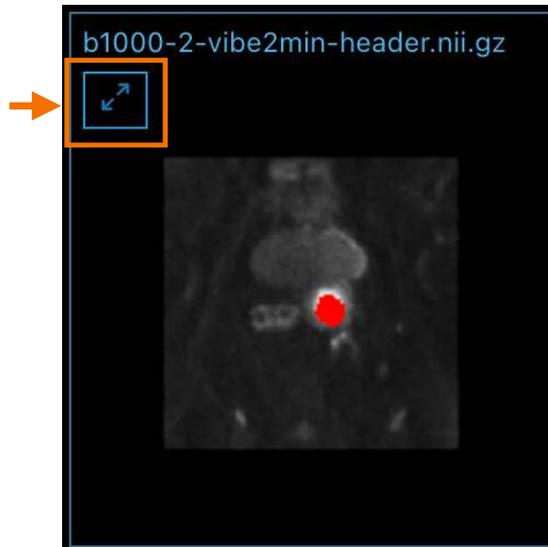


Figure 5.5: Slice view with button to switch sequence in the main view. The button appears when hovering over the view.

Grouping of Modalities In order to define grouped sequences, the user has to press the `shift` key and click on the desired modalities. The selected views are visualized with a bold, bright blue border. We show the difference between selected and not selected views in [Figure 5.6](#). All modifications made in one of the sequences are mirrored in the other modalities that are also part of the grouping. For example, moving the segmentation in one grouped sequence to the right, automatically transforms the masks in the other grouped sequences as well. The selection stays consistent when changing the layout or switching the reference view.

Reference Modality As depicted in [Figure 5.1](#), the user should define a reference sequence before the automatic shape matching and, in case of interactive modifications, before calling the final co-registration. The user can select the reference sequence by pressing the `Ctrl` key and clicking on the desired view. The view is highlighted with an orange border to emphasize the current reference.

Automatic Shape Matching For now, the automatic shape matching does not consider previous changes in the segmentation masks performed by the user. After the search is done, the updated position and rotation of the segmentation mask, as



Figure 5.6: Interactive grouping of sequences. The upper two and the lower left sequences are in a group as indicated by a bold, bright blue frame. Adaptations in one of them are applied to the other one as well. The sequence at the bottom is not part of the group.

determined by the algorithm, appears on the screen. The user can keep the changes or revert them for each modality individually.

In case the segmentation location is not satisfying and the user prefers to adapt it based on the initial position, a *Reset Changes* button appears at the current mouse position when pressing the `Alt` key (see [Figure 5.7](#)). It sets the position and rotation of the segmentation under the cursor to its original values, which are the same as in [VIBE](#). If the view is grouped with other modalities, their segmentation mask movements and rotations are reverted as well.

Interactive Shape Matching For transforming the segmentation mask, we provide several keyboard interactions. The transformations depend on the current mouse position

for both the sequence and slicing direction. The user can *translate* the mask in the current view with the four arrow keys (up, down, left, right) in x and y direction. The different slicing directions are linked. Therefore, translating the segmentation in one slice view can have different effects in the other slicing directions (as explained in [Chapter 4](#)).

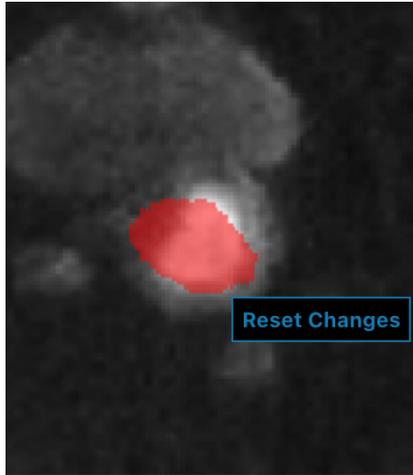


Figure 5.7: Reset button that appears at the current mouse position to set the position and rotation of the segmentation to the initial, unmodified state.

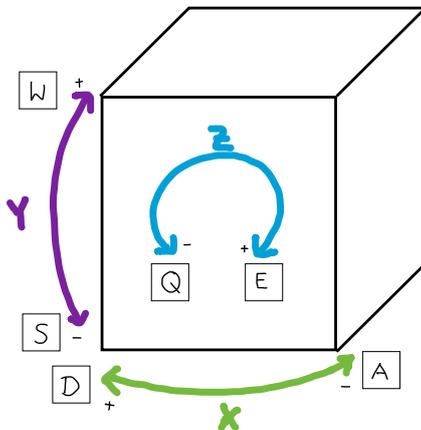


Figure 5.8: Rotation interaction with the segmentation mask indicating the corresponding keys.

To *rotate* the mask in different directions, several keys can be used. We visualize the possibilities for the frontal axial oblique view in [Figure 5.8](#) with the help of a 3D cube to facilitate orientation. The toolbar provides hints about which keys to use for rotation, next to the current state of translation and rotation.

Magic Lens We can activate the magic lens in the view where our mouse cursor is located at by pressing the key R. Around the cursor, a square is displayed which shows the reference sequence (we showed this feature in [Figure 4.12](#) of [Chapter 4](#)). Scrolling and other interactions are still possible if the lens is activated. When releasing the R key, the lens disappears. The lens can be used in all sequence views and slicing directions.

Co-Registration After adapting the masks, the tool looks like shown in [Figure 5.9](#). In an ideal case, all the masks fit the tumor border of the underlying sequence. When clicking on the *Register* button in the toolbar, the volumes are aligned to the reference sequence. After the registration is done, the views update and the new, transformed volume is shown. We reset the segmentation mask automatically to the original position in the reference sequence. The user can assess the results visually by analyzing the fit of the mask and by using the magic lens. Iteratively adapting the mask placement is possible if the user still wants to do so. Only those sequences where the user changes the segmentation mask are aligned. The other sequences stay the same and are not transformed. Currently, we save the registered images and transformation parameters automatically. However, also additional download with the help of the corresponding buttons in the toolbar (see [Figure 5.3](#)) is possible.

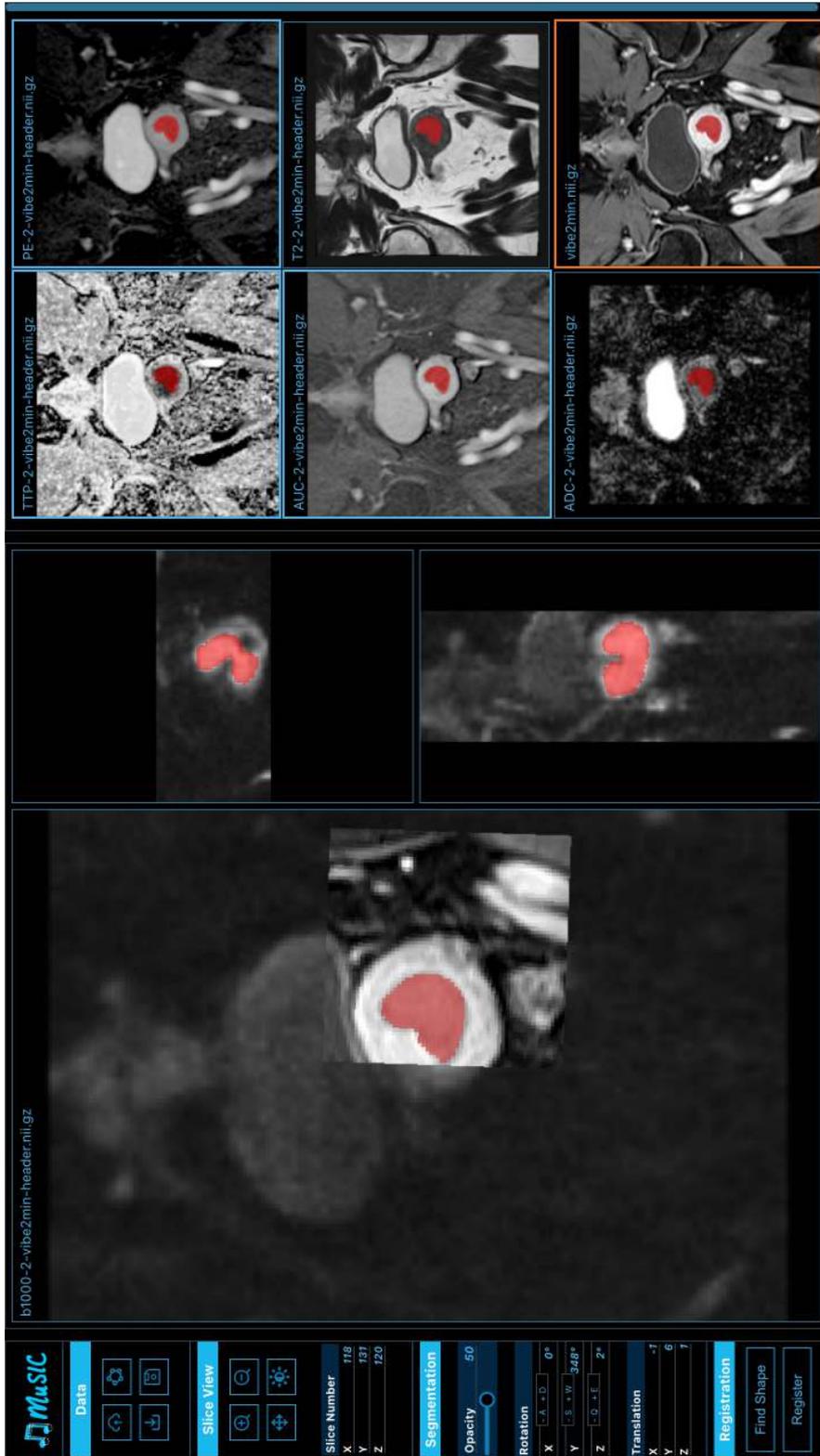


Figure 5.9: MuSIC after grouping and adapting the segmentation positions. VIBE is set as the reference sequence as depicted with the orange border. TTP, PE and AUC are grouped. When clicking on *Register*, the sequences are aligned according to the segmentation positioning.

Implementation

To make our tool easily accessible, we created a modular web-based approach. We use JavaScript as the frontend programming language and Python in the backend.

The project is structured mainly in two directories: `frontend` and `api`. In the `frontend` directory, there is a `package.json` file containing the necessary dependencies to run and open the web page in the browser. Using the tool's backend part requires a Python environment (we use Python 3.8). The Python requirements can be installed from a `requirements` file in the project's base directory. Furthermore, ITK [MLI⁺14] and Elastix [KSM⁺09] need to be installed to perform the registration. We use a Windows PC with a WSL2 Ubuntu Linux subsystem. However, Elastix and all other libraries are available for Windows, Linux, and macOS.

6.1 Frontend

For frontend development, we use the open-source JavaScript library **React** [Wal11]. It allows the creation of reusable components for responsive web-based user interfaces. Maintained by Meta (former Facebook) and other contributors, it has become one of the most used libraries for web development in the last few years. Among other features, it offers virtual DOM trees for real-time updates, JSX as a syntax extension of HTML for JavaScript, and state management features [Wal11]. Several third-party components based on React are open source, such as React-Grid-Layout [Rea], which we use for the interactive, draggable, and resizable layout in the multi-sequence grid view.

Layout and Design For the general layout and design of our application, we use Leaner Style Sheets (**LESS**) as a backward-compatible extension to Cascading Style Sheets (CSS). It makes writing CSS more readable and efficient, as it allows, for example, to create nested classes and variables.

6.1.1 Rendering

WebGL is an OpenGL ES-based API for interactive 3D rendering in the web. It can be used with JavaScript interfaces. In a 3D space, a camera defines the point of view, including position, orientation, and other camera characteristics. Therefore, it determines how we, as a user, look on an object in a scene. Such objects in WebGL are made out of vertices that represent coordinates (x, y, z) in the 3D coordinate system. The vertices together form a mesh. Textures or materials contain surface characteristics of a mesh in the form of bitmaps, such as color, light reflection properties, and more [Par12, ML13]. The open source library **Three.js** [Dan12] provides common objects for 3D WebGL renderings in a high-level API. It offers an object-oriented interface with pre-defined features and interaction possibilities [Par12]. We use Three.js to create and interact with the renderer, cameras, scenes, and objects, as well as with vectors and matrices. We have one canvas covering the whole application and, therefore, also only one WebGL renderer. Using one canvas and renderer for each view results in potential errors due to too many WebGL renderer contexts. Each view, which means each sequence and slice direction, represents one scene. For each scene, we have one camera. We define where to render the scene based on the element that contains the scene. With this information, we can set the current viewport of the renderer when looping over the views. Since the views are in a 2D space, and we do not need perspective projections, we use an orthographic camera setup for each view. We hide the elements containing the x and y slicing direction views for the slice directions that are currently not visible. We show the structure for rendering in [Figure 6.1](#).

6.1.2 Custom WebGL Shader

As described in [Section 4.2](#), we developed a WebGL shader to display the segmentation data. In order to render the representation of an object on the screen, shader programs implement how vertex coordinates get mapped and displayed on the screen by the GPU. They are written in the OpenGL ES shading language (GLSL ES) and consist of a *vertex* and *fragment* (or pixel) shader. The former is called for each vertex of the data. It is responsible for the calculation and manipulation of the vertex positions for the projection to the viewport. The latter mainly creates the final color output for each fragment [Par12, ML13].

We want to visualize three slicing directions. For each of the three slicing directions, we have to calculate the intersections between a plane and the segmentation volume to render the intersection onto the screen. Therefore, we send the following input parameters to the shader program:

- **Data array:** The shader receives the gray values of the data volume as a normalized Float32 array buffer in a range between 0 and 1.
- **Original volume size:** The original size of the data volume is specified in the data header which contains information from the scanner and acquisition process.

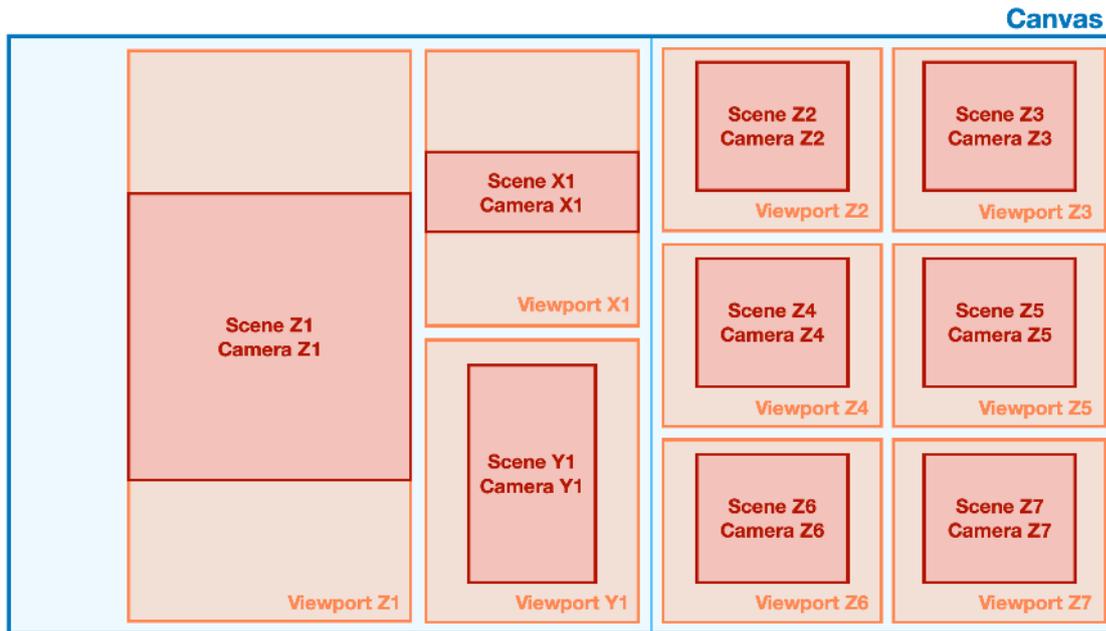


Figure 6.1: Structure of the tool for rendering of the different sequences. The canvas covers the whole application, so for all scenes we use the same renderer. The canvas is split into viewports, where each viewport is defined by a container element. In each viewport, the renderer displays one scene with one camera specific for the current viewport.

- **Desired output size:** The size for the 2D rendering. In case of the coronal view, for example, these are the first two dimensions of the original data volume ($x_m ax$ and $y_m ax$).
- **Virtual camera ID:** The ID of the virtual camera representing the camera orientation, as described in [Figure 4.10](#) of [Chapter 4](#). It can be seen also as the slicing direction of the data volume. The value can be 0 for the frontal z view (i.e., axial oblique), 1 for x (volume viewed from the top), and 2 for y (side view, corresponds originally to sagittal oblique). The computation of the coordinates that are rendered on the screen depends on the camera direction.
- **Depth:** A 3D vector defining the current depth of the plane for each slice direction, initialized as the middle slice of the volume, which means the rounded maximal slice number divided by two. The user can adapt the depth by scrolling through the volume.
- **Plane normal \vec{n} :** A 3D vector initialized as $(0, 0, 1)$, perpendicular to the frontal z plane.
- **Plane tangent1 \vec{t}_1 :** A 3D vector laying in the frontal z plane, perpendicular to the plane normal. Initialized as $(0, 1, 0)$.

- **Translation vector:** A 3D vector to allow for translation of the planes via user interaction in each slicing direction. Initialized as $(0, 0, 0)$.
- **Rotation center vector:** A 3D vector that specifies the center point of rotation, around which the planes are rotating in the 3D space. We set the rotation center vector to the normalized center of mass of the segmentation. If it is set to $(0.5, 0.5, 0.5)$, the planes rotate around the volume center.

In the shader, we process the intersection with one of the three planes and derive the correct coordinates for the other planes based on the camera ID. Since we have three slicing directions, we use the shader three times per sequence and calculate the displayed coordinates.

Vertex shader The vertex shader is used to render a geometry that covers the current viewport to trigger the fragment shader processing. We transform the model coordinates, which means the original coordinates in our data volume of the segmentation data, to the so-called clip space in order to project them on the screen. The clip space can be imagined as a cube where each side is 2 units long and the origin $(0, 0, 0)$ is in the center of the cube. The coordinates range from $(-1, -1, -1)$ to $(1, 1, 1)$. Rasterization of this clip space results in the 2D image which is rendered [Par12, ML13]. Furthermore, we prepare the texture sampling coordinates for the segmentation rendering. We rescale the 2D coordinates of our viewport to a range between 0 and 1 to match the coordinate range with which we sample our texture. This rescaling enables an orthogonal projection of each virtual camera to its corresponding plane, since we define the viewport coordinates to be on the plane. These values are then interpolated for each fragment when passed to the fragment shader, so we sample uniform fragments for each pixel on the screen.

Fragment shader In order to define an output color for each fragment, we calculate the plane-volume intersection in the fragment shader. We imagine our segmentation volume to be a unit cube in the center of an orthonormal basis which is the standard basis E . As an outcome, we want to sample the original segmentation data volume with the 3D standard coordinates of the plane in the standard basis E . We consider the derived 2D viewport coordinates from the vertex shader, which range between 0 and 1, being on the plane. With these plane coordinates, we can compute the volume coordinates based on the current plane translation and rotation. We want to compute the intersection between the plane and the volume in the standard basis E , which equals to expressing the plane itself in E . Since we only have the plane coordinates in the plane basis B , we need to transform them to the standard basis E . The input plane normal \vec{n} and plane tangent1 \vec{t}_1 define two of the basis vectors of the plane basis B that are expressed in the standard basis E . As a third vector tangent2 \vec{t}_2 , we calculate the cross product $\vec{t}_2 = \vec{n} \times \vec{t}_1$. Initially, the 3D standard coordinates are composed of the 2D plane coordinates $(x, y)_B$ and $0.5(z)_B$. By default, the center of the volume $(0.5, 0.5, 0.5)_E$ is considered to be the center of rotation. In the 2D plane coordinates, the rotation center

is in $(0.5, 0.5)_B$ by default. Before computing the rotation, we subtract the normalized rotation center vector, which represents the center of mass of our segmentation volume, from the 3D standard coordinates. To compute the rotation of the plane, we first multiply the transformation matrix composed of the two tangents \vec{t}_1 and \vec{t}_2 and the plane normal \vec{n} with the shifted and normalized standard coordinates. This rotation represents a transformation from the plane basis B to the standard basis E . Afterwards, we add again the rotation center vector to the standard coordinates, as well as the normalized depth with right to the volume center. The depth represents the current slice number in the volume depending on the camera ID. To account for translation, we also add the normalized translation vector to the standard coordinates based on the camera ID.

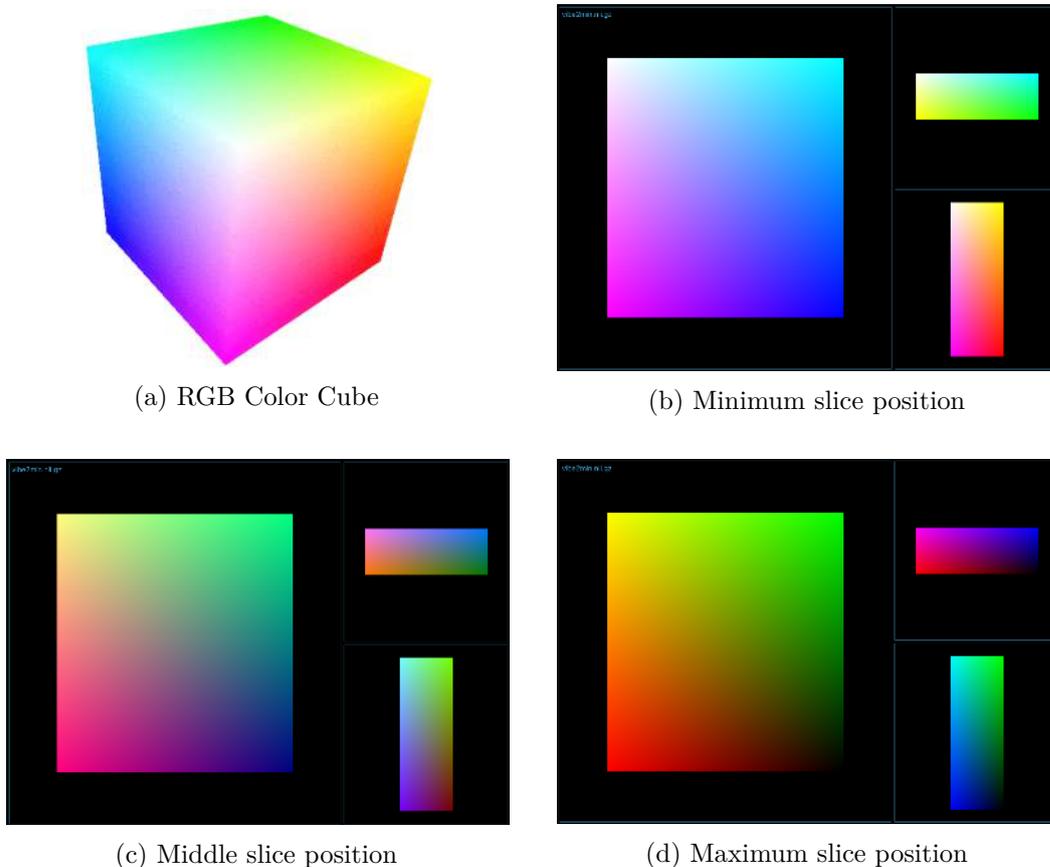


Figure 6.2: Coordinate color visualization in the WebGL shader. (a) shows an RGB color cube (from Congote et al. [CSK⁺11]). The white corner represents the point $(1.0, 1.0, 1.0)$, while the black one (not visible here since it is on the bottom back corner) represents $(0.0, 0.0, 0.0)$. (b), (c) and (d) show example plots visualizing the coordinates of the segmentation volume (shape: 232, 256, 88) and the behavior when scrolling through it in each slice direction without applying rotation.

User interaction If the user translates the segmentation mask, we update the translation vector, which we send to the shader, accordingly. In case of rotation, we need to adapt the plane normal \vec{n} and the plane tangent1 \vec{t}_1 . Depending on the user interaction, we apply a positive or negative rotation angle to both vectors. In the shader, the updated vectors are then used to calculate the second tangent \vec{t}_2 .

Color Cube Quality Check In order to determine whether the coordinates are transformed properly in the WebGL shader and to examine if the behavior on user interaction works as desired, it can be helpful to visualize the coordinates of the plane intersecting the data volume. WebGL uses RGB values. Therefore, we can visualize the 3D standard coordinates in a range from 0 to 1 as a resulting RGB color cube (see [Figure 6.2](#)). If we scroll through the volume in different slice directions, we can observe how the colors change based on the current position of the plane. For a rotation around the z axis, we see that the colors rotate around the center point of the projected image (or the specified center of rotation in 2D). Rotation around other axes show us different sides of the color cube depending on the angle.

Magic Lens In order to not only have a side-by-side comparison between two sequences, we introduce a magic lens visualization [\[BSP+93\]](#) that makes it possible to show the reference data in a square around the current mouse position with the segmentation. We implement this feature in the fragment shader. As an additional input, the shader receives the current mouse position in 2D in a range between -1 and 1 for both dimensions. The mouse position depends on the current viewport. Furthermore, we pass the data array of the reference sequence to the shader program. We define the size of the square and render the volume data at the positions within the square. Still, we show the segmentation with the chosen opacity as an overlay to allow to determine how well the current position of the segmentation matches the tumor borders. When translating and rotating the segmentation, the data representation in the lens will update as well as we sample the reference volume with the same coordinates as computed for the segmentation data.

6.1.3 Data Handling and Interaction

For reading NIfTI data in JavaScript, we use the open-source NIfTI reader [\[LM16\]](#) that loads compressed and uncompressed NIfTI data into an array buffer. We then normalize the data and re-arrange the values in the array for the different slicing directions. We store three data objects for each sequence, corresponding to the three slicing directions. Each object contains:

- the sequence and segmentation data, each as an array of normalized gray values,
- information about the original shape of the data,
- the current slicing depth depending on the slicing direction,
- the scene and camera, both with unique identifiers,

- the name of the container element to define the viewport and render the scene,
- the position and zoom factor of the object in the scene, which can be changed on user interaction,
- rotation and translation parameters of the segmentation, and
- the current mouse position and link to the current reference data to show the magic lens on demand.

6.1.4 Package Management

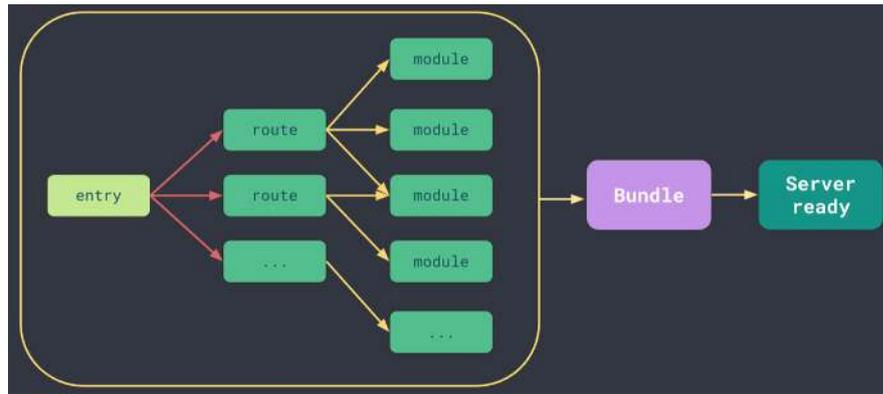
We use the cross-platform Node.js [Fou09] package manager **npm** [npm10] for JavaScript. It allows for the installation of open-source code packages from an online repository. npm comes along with a command-line client to download and install packages. The `package.json` file contains the packages with the currently used versions and dependencies. The installation of Node.js includes the npm package manager. To install or update the packages and dependencies, one can run the command `npm install` in the `frontend` directory. The command `npm run dev` provided by npm starts the local development server (localhost) on the specified port (here: port 3000).

6.1.5 Application Build

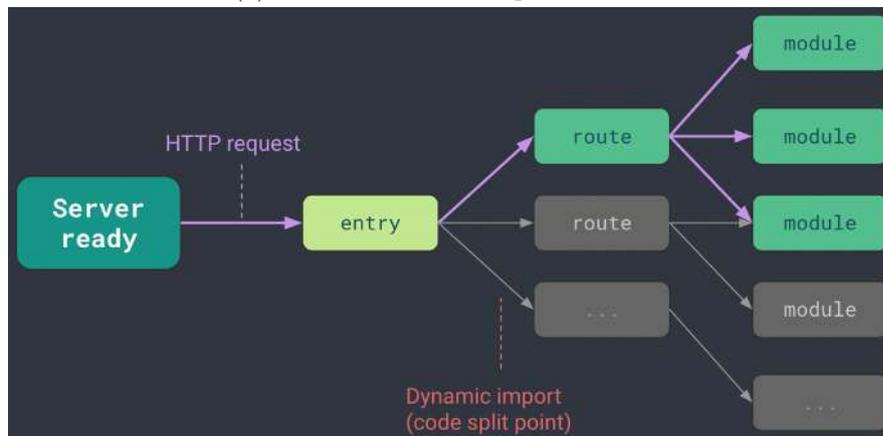
To build our application, we use the JavaScript build tool Vite [You19]. Until recently, bundle-based tools were common in web development. Such tools collect, process, and link source modules in files supported to run in a web browser. In contrast to bundle-based tools like webpack [Kop12], Vite supports unbundled development build. This kind of build is now possible since browsers natively support ES modules. In this way, Vite sends the code only on demand to the browser, so it is only processed if it reflects changes in the browser. Therefore, Vite offers an easy-to-use and, compared to bundling, extremely fast server start and code updates in the browser. We depict the differences in Figure 6.3. Vite is available with the installation of Node.js.

6.1.6 Server Requests

To make asynchronous get, post, put, or delete requests to the backend, we use the **Axios** [Zab14] promise-based HTTP client. Axios is an alternative to the native `fetch()`-API and is also supported by older browsers. Furthermore, it already implements several commonly used features, for example the automatic JSON transformation of request and response data. To register the data after mask adaptation, for example, we send one request per sequence. The request includes the reference sequence and segmentation data, as well as the data of the sequence that should be aligned. Furthermore, we include the transformation matrix in the request. We only send those sequences to the backend where the segmentation mask has been modified. If sequences have been grouped, they are sent in one request to process them together in the backend. After registration, we receive the transformed sequence and update the respective view.



(a) Bundle-based development server



(b) Native ESM-based development server

Figure 6.3: (a) Bundle-based vs. native (b) ECMAScript Modules (ESM)-based development server, from the Vite webpage [You19].

6.2 Backend

The REST framework for the communication between frontend and backend is provided by **Flask** [Ron10]. Flask is a Python micro web framework that offers only core functionality compared to other all-in-one frameworks like Django [Fou05], while being more flexible and easily extensible. It supports developers in structuring the web application and resource management in a standardized way. Furthermore, it offers a quick setup and is easy to use. In our `api` directory, the file `app.py` contains the Flask setup and resource management to distribute the requests to the corresponding service. Running this file starts the Flask server on localhost port 5000.

6.2.1 Image Processing

For most image processing steps, we use the Python library **NumPy** [Oli06] to operate with multidimensional arrays. For example, we need to crop the volumes in a certain area around the segmentation in the automated shape matching. We compute the minimum and maximum extent of the segmentation in each axis to find these borders by using NumPy functions. Also, the histogram trimming is performed by indexing the NumPy arrays containing the original data. We apply image filters like **GGM** or a Gaussian filter by using **SciPy**'s [VGO⁺20] `ndimage` package. For image processing steps that involve translation and rotation of imaging data, we use **SimpleITK**. Transformations with SimpleITK involve interpolation and, therefore, lead to smoother results. When working with data in both NumPy and SimpleITK, it is essential to keep in mind that NumPy operates in a space where the coordinates have an XYZ order, while SimpleITK makes computations in a ZXY space.

Also, for the computation of the cost function in the automated shape matching, we use NumPy operations to crop the volumes, compute the segmentation hull and calculate components like the standard deviation. The interpolation of the segmentation mask computed by SimpleITK might lead to results with float values between 0.0 and 1.0 (instead of only 0 and 1 in the original mask). We threshold the mask for the hull computation and set values below 0.9 to 0 and values that are bigger or equal 0.9 to 1, since setting all values above 0 to 1 would result in too large hulls. As mentioned in Chapter 4, we perform the optimization by using SciPy's [VGO⁺20] dual annealing approach in the `optimize` package. To create example plots, we work with the Python library **Matplotlib** [Hum07].

6.2.2 Registration

We use **Elastix** [KSM⁺09] as medical image registration framework as it offers landmark-based registration. It is a command-line tool and includes a library called **SimpleElastix** [MBSK16], which makes the Elastix commands available in Python, Java, and R. The installation requires ITK [MLI⁺14]. Elastix can be compiled with CMake [HMK⁺00] after downloading the code from the GitHub repository. SimpleElastix offers an object-oriented interface, where a default `ElastixImageFilter` can be modified. Different parameter maps containing specific variables can be used in combination in one image filter, for example a combination of rigid and bspline transformations. The parameters for rigid registration are defined by the inverse matrix we receive from the frontend since we want the images to fit the original mask position. We transform the non-zero mask indices by multiplication with this inverse matrix. The indices are then used by SimpleElastix as landmark indices for the registration. In case of a registration request from the frontend, we transform the data into a 3D NumPy array with the original data shape. Before registration, the arrays need to be transformed to ITK images to be used by SimpleElastix. Afterwards, we re-transform the resulting image to a NumPy array and send it back to the frontend.

Case Studies

In this chapter, we will examine two cases of our data set in greater detail to emphasize the potential of our approach and to identify difficulties. The two cases are representative for our data set, where the sequences are misaligned by small rotation and translation parameters. After showing the results of the automatic shape matching, we also compare final registration results to outcomes of SimpleElastix [MBSK16] without using our approach based on segmentation masks as landmarks. We determine the quality of the results by visual evaluation after identifying the tumor borders together with our collaborating radiologists. In this way, we can identify if a segmentation mask fits the underlying tumor borders.

7.1 Shape Matching

In Figure 7.1 and Figure 7.2, we show a comparison between the segmentation position before and after the automatic shape matching in the different sequences. The upper row visualizes the VIBE reference sequence with a red semi-transparent overlay of the segmentation fitting the tumor borders. We cropped the sequences to the uterus area, which is the round structure around the tumor (in VIBE, the uterus appears brighter than the tumor and background with bladder and soft tissue). To visualize the tumor position in the underlying sequences, we highlight the tumor border with a bright blue outline, as identified with our medical experts. Note that this might not correspond to the exact border, but it provides indications about the tumor location in the various sequences. We highlight mismatches between the segmentation position and the actual tumor border in yellow.

In the first case (Figure 7.1), we can see that the automatic shape matching led to satisfying results the first three sequences, namely PE, TTP and AUC. Those sequences have been grouped and processed together. After the automatic shape matching, the segmentation mask matches the tumor borders, as indicated by the blue tumor outlines.

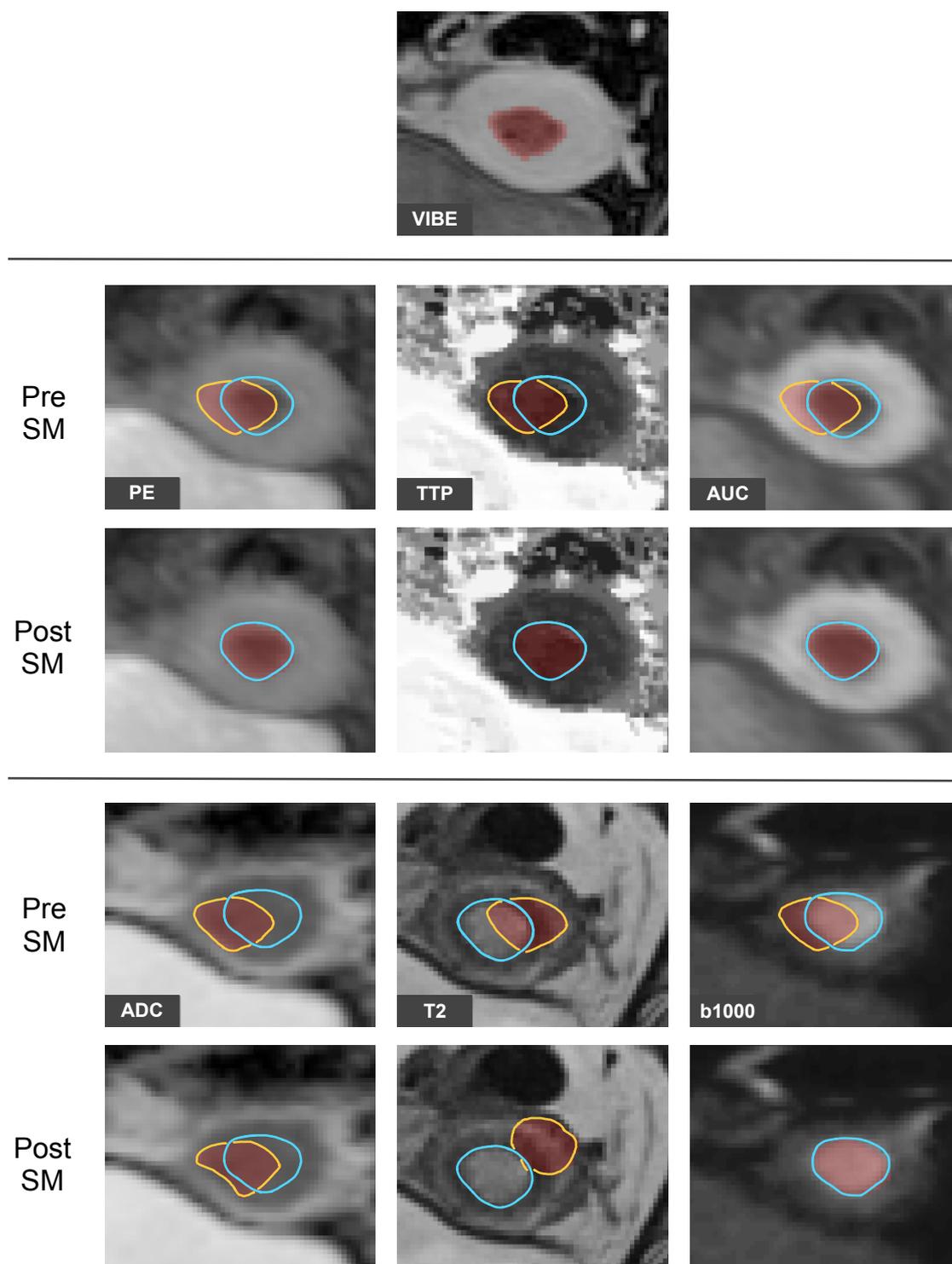


Figure 7.1: Comparison between pre and post Shape Matching (SM) of Case 1. The first row shows the VIBE reference sequence. The segmentation mask is shown as a semi-transparent overlay in red, while the tumor borders in the underlying sequences are highlighted in bright blue. Mismatches between the actual tumor border and the segmentation mask are delineated in yellow.

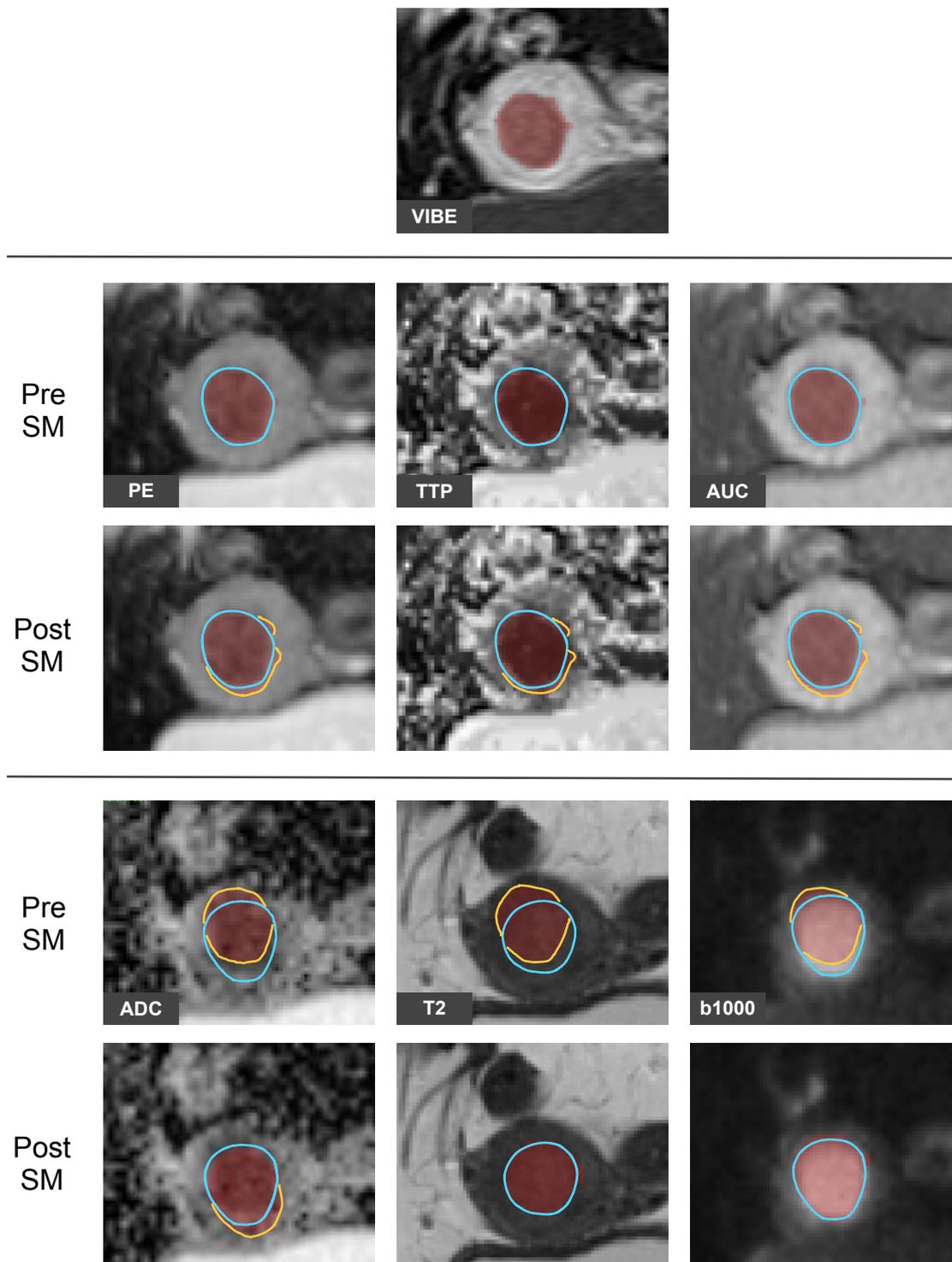


Figure 7.2: Comparison between pre and post Shape Matching (SM) of Case 2. The tumor borders in the underlying sequences are highlighted in bright blue. Mismatches between the actual tumor border and the segmentation mask are delineated in yellow.

Also, the search in the b1000 sequence yielded a good fit. For [ADC](#) and T2, we can observe two problems that may occur. In [ADC](#), it is visually and algorithmically challenging to distinguish the tumor from the surrounding tissue in this use case. Instead, the contrast is high around the outer parts of the uterus. Due to low contrast at the tumor border, the algorithm struggles to find a good match and terminates with a cost function minimum which usually represents the uterus border. In T2, the segmentation terminates at a local minimum between the tumor border and uterus. Furthermore, the T2 volume, in this case, varies slightly in gray values compared to other T2 sequences in our data set (see [Figure 7.2](#) for a typical example). Therefore, the tumor border might be less prominent than the uterus border after histogram trimming. We saw another example in [Figure 4.5](#) in [Chapter 4](#), where the value distribution is different than in other subjects. Last but not least, [SA](#) is a probabilistic optimization method and involves a random component. Therefore, there is always a slight chance of ending up in local minima, even though it is less likely than, for example, in gradient descent approaches.

For the second use case ([Figure 7.2](#)), [PE](#), [TTP](#), and [AUC](#) are already quite well aligned before shape matching and registration. When scrolling through the slices, a minimal translation in the axial-oblique slicing direction becomes visible. Also, the translation found by our algorithm is small but also involves a small rotation. The borders in the T2 sequence are difficult to identify. Still, the automatic shape matching finds a suitable position. Also in b1000 and [ADC](#), the mask fits better than before the shape matching. Only small adaptations in rotation might be necessary, as visible in this slice when looking at the lower tumor border in [ADC](#), and the upper tumor border in b1000.

7.2 Registration

As mentioned earlier, we assume that the tumor position in the volumes of one patient can only vary in translation and rotation parameters. Therefore, to analyze only tumorous tissue, rigid registration would be sufficient. However, in order to take also surrounding tissue into account, especially the uterus, deformable registration could play an essential role in the future. This is due to the fact that, in contrast to the tumor itself, the uterus is a soft structure and can be deformed based on bladder filling and intestinal movements. For this reason, we applied both rigid and deformable registration to our use cases and visually compare the outcomes.

7.2.1 Rigid Registration

[Figure 7.3](#) and [Figure 7.4](#) depict the volumes of our use cases as a result of rigid registration with Elastix (without landmarks) and MuSIC (using our interactive segmentation-landmark approach) in comparison. Our medical experts placed the segmentation masks for the registration with MuSIC during the usability evaluation. MuSIC leads to more accurate results in almost all cases, since the results reflect the mask positions as set by the medical expert. We can see that the locations of the segmentation masks fit the tumor borders. Surprisingly, Elastix seems to have problems especially with the b1000

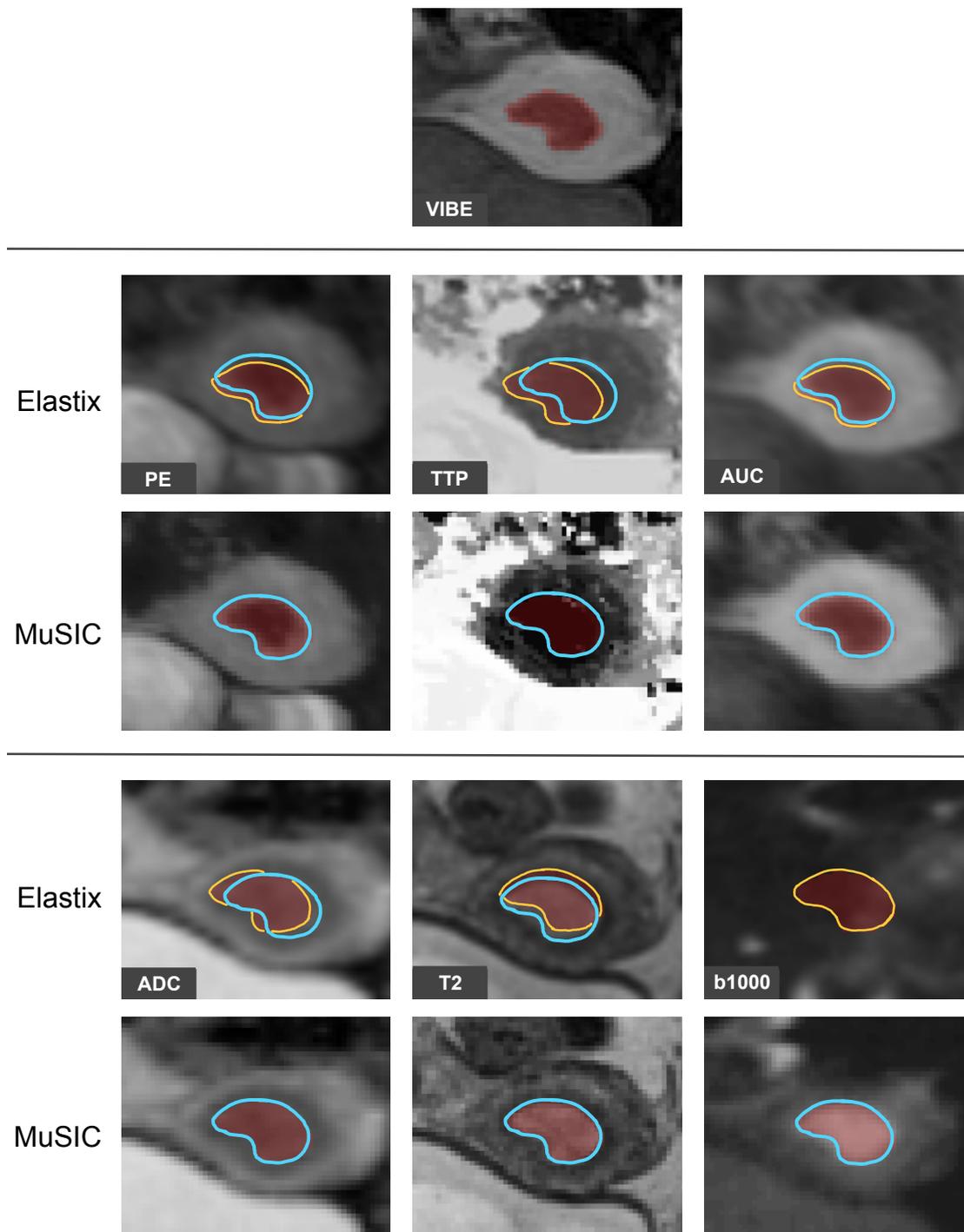


Figure 7.3: Comparison of rigid registration outcomes between Elastix and MuSIC of Case 1. The tumor borders in the underlying sequences are highlighted in bright blue. Mismatches between the actual tumor border and the segmentation mask are delineated in yellow.

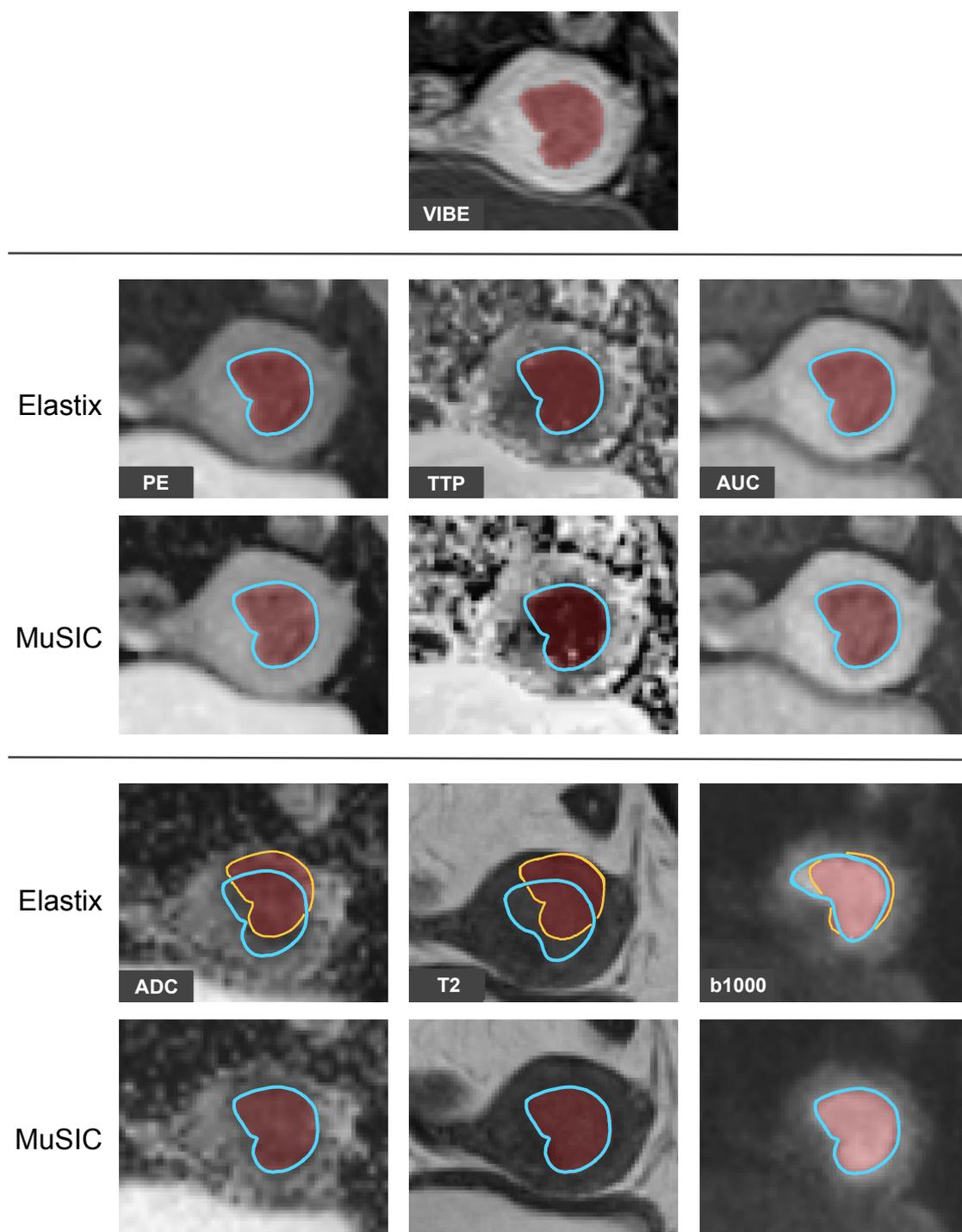
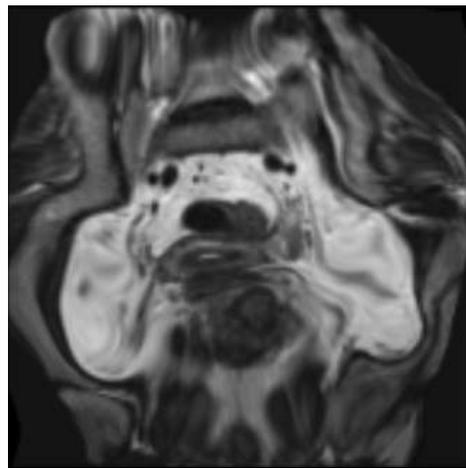


Figure 7.4: Comparison of rigid registration outcomes between Elastix and MuSIC of Case 2. The tumor borders in the underlying sequences are highlighted in bright blue. Mismatches between the actual tumor border and the segmentation mask are delineated in yellow.

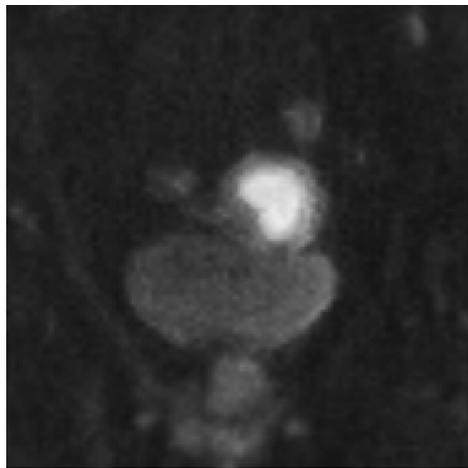
sequence, even though the tumor is well visible here. However, structures around the tumor appear with low contrast and a lot of noise. In Case 1 (Figure 7.3), the b1000 sequence is transformed to a completely different position, and also in Case 2 (Figure 7.4), the position is not accurate as shown by the mismatch between the actual tumor outline in blue and the segmentation mask in yellow. Using Elastix, most other sequences are closer to the actual tumor position compared to no registration. However, as we can see from the mismatching areas in yellow, they lack in precision (apart from the first three sequences of Case 2 in Figure 7.4, which almost did not require transformations).



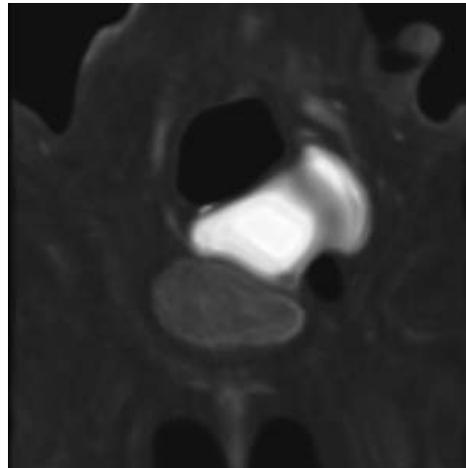
(a) T2 Original



(b) T2 Deformations beyond recognition



(c) b1000 Original



(d) b1000 Deformation to the uterus area

Figure 7.5: Deformations in non-rigid registration without landmarks. (a) shows the original T2 image, (b) visualizes the transformed T2 image with strong deformations in all areas, making the image unrecognizable. (d) visualizes an example where the b1000 sequence in (c) is transformed so the tumor expands to the uterus area.

7.2.2 Deformable Registration

To also align structures surrounding the tumor, such as the uterus, we also allow for deformable registration. A deformable registration could be essential for future research in radiomic tumor profiling. We observe unnatural deformations when adding a step of non-rigid registration after rigid transformation without landmarks, but maintaining the other settings. We show two examples in [Figure 7.5](#). What happens sometimes is that the tumor is interpreted as the uterus, and it is expanded to the uterus area in the [VIBE](#) sequence. This occurs especially in the case of big tumors and if the tumor appears bright in the moving sequence. A possible explanation is that the uterus has brighter gray values in [VIBE](#) and high contrast. In b1000, for example, the tumor has the highest gray values and shows the highest contrast to its surrounding, while all other structures appear rather dark and with low contrast.

Therefore, we adapted the settings of Elastix following the manual [\[KS18\]](#) to be able to compare the results better and describe differences. We apply a much higher number of iterations than the default which we use in MuSIC. Instead of the default 500, we use 2000 iterations after trying different values. Even higher values did not yield better results. Furthermore, as described in the manual [\[KS18\]](#), the parameter α can be tuned that determines the amount of deformation applied to the image. α is a gain factor that weighs the negative gradient of the cost function in gradient descent optimization to define the search direction. A high α can lead to strong deformations as we showed in [Figure 7.5](#). A too small α limits the search space and there might be too few degrees of freedom for deformations. To adapt this parameter, we use `StandardGradientDescent` instead of `AdaptiveStochasticGradientDescent`. We show the results of two settings ($\alpha = 500$ and $\alpha = 5000$) in [Figure 7.6](#) and [Figure 7.7](#) together with the outcomes of [MuSIC](#). Furthermore, we highlight several observations in the corresponding sequences that we discussed previously with our medical expert collaborators.

In the first use case ([Figure 7.6](#)), a setting with $a = 500$ transforms most of the sequences such that the segmentation mask is closer to the actual tumor borders than in the original sequences. [TTP](#) and [ADC](#) would still need more translation, as we highlight in [Figure 7.6](#). The tumor position in T2 seems acceptable, but too little deformation has been applied to the uterus. We emphasize the mismatch between the purple VIBE outline and the uterus in T2 in yellow. b1000 was transformed to a completely different position in both settings. With $a = 5000$, the uterus transformation of T2 resembles more the VIBE sequence. However, it becomes visible that the tumor was deformed when observing the tumor borders, as we depict with green arrows. In this sequence, the tumor is represented by the gray values within the darker outline. After registration with Elastix, an overlap becomes visible at the tumor border. These gray values, that now are inside the mask region, are most likely not part of the tumor. This phenomenon occurs also in [ADC](#). The region around the tumor in [TTP](#) was stretched to fit the uterus. We see that these deformations are probably not adequate, as they have not been applied to [PE](#) and [AUC](#). In [MuSIC](#), we process these sequences together since they are derived. Therefore, it is less likely to deform structures that represent the same biological state in a different way.

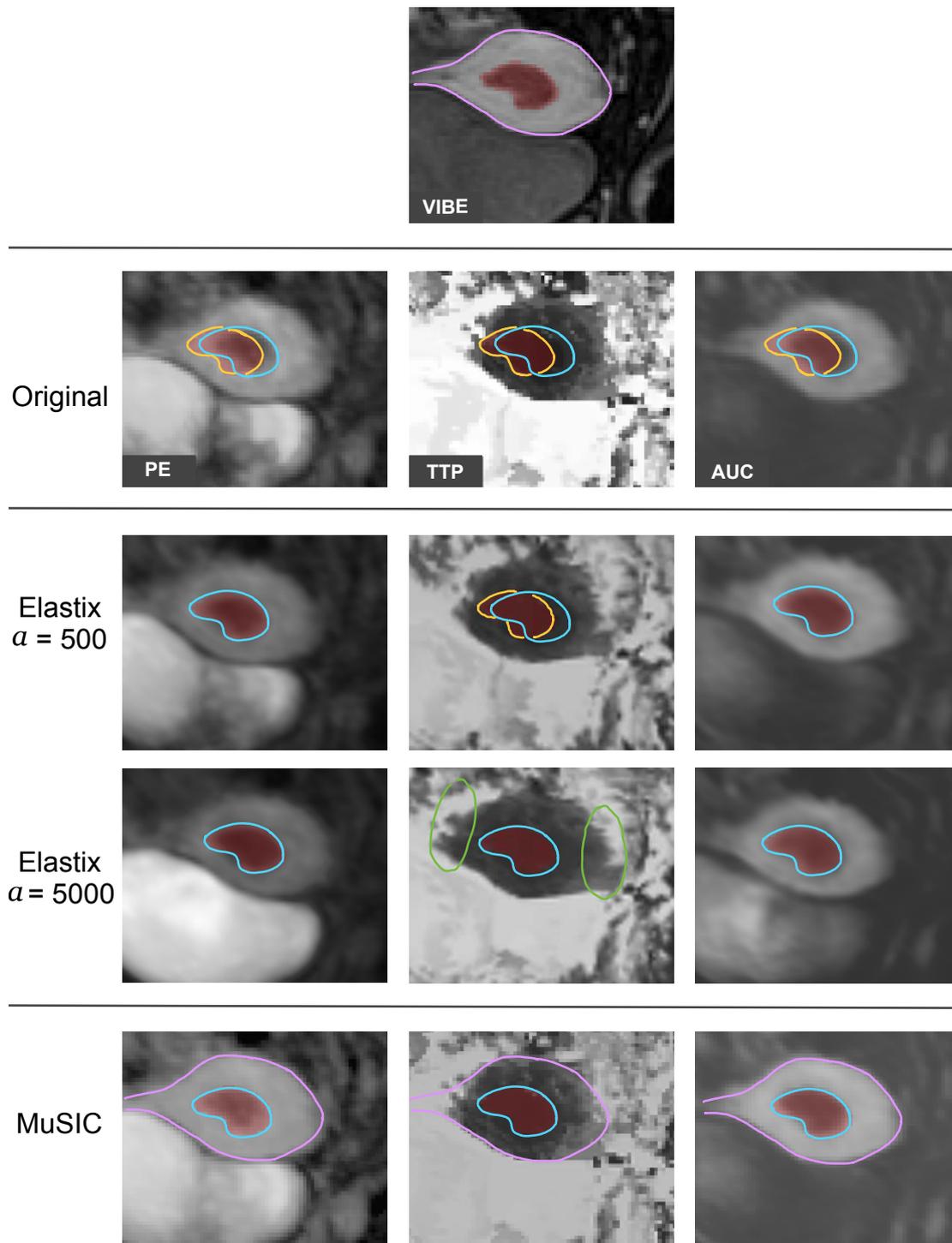


Figure 7.6: Comparison of deformable registration outcomes between only Elastix and MuSIC of Case 1 (PE, TTP, AUC). We use two settings for Elastix changing the parameter α which corresponds to the amount of expected deformation. The tumor borders in the underlying sequences are highlighted in bright blue. Mismatches between the actual tumor border and the segmentation mask are delineated in yellow. Strong deformations are shown in green. The uterus outline of VIBE is delineated in purple for the MuSIC results.

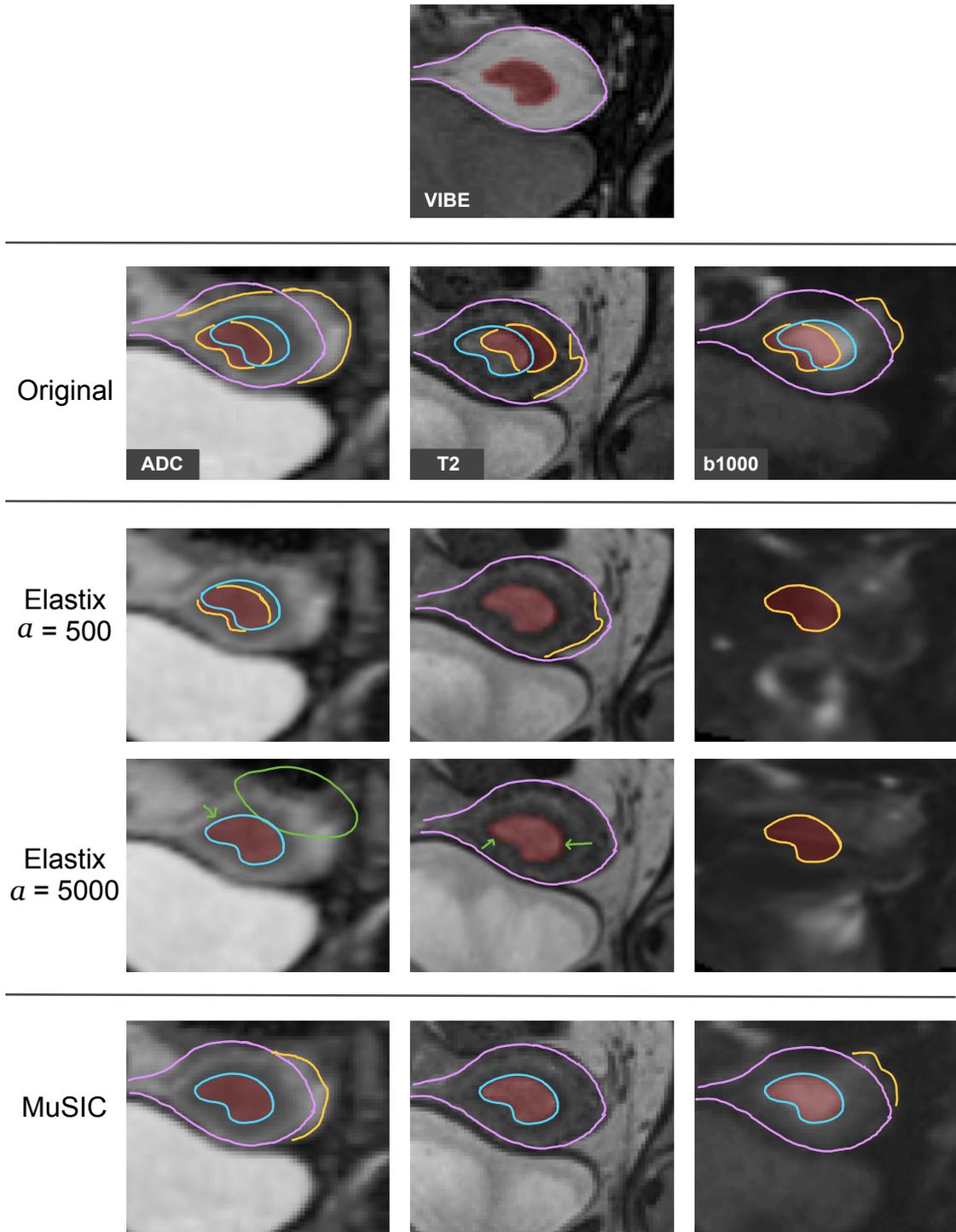


Figure 7.6: Comparison of deformable registration outcomes between only Elastix and MuSIC of Case 1 (ADC, T2, b1000). We use two settings for Elastix changing the parameter α which corresponds to the amount of expected deformation. The tumor borders in the underlying sequences are highlighted in bright blue. The uterus outline of VIBE is delineated in purple for several resulting sequences. Mismatches are delineated in yellow. Strong deformations are shown in green.

Using `MuSIC`, the masks seem to fit the actual tumor borders according to our medical experts. The deformations of the uterus are realistic, as we show with a purple overlay of the uterus outline of VIBE. Furthermore, the tumor itself is not deformed. In `ADC` and `b1000`, the uterus deformations could be stronger, as shown with the yellow mismatch lines.

The same setup with Elastix yields different results for the second use case (Figure 7.7). `b1000`, `PE` and `AUC` seem well-aligned after Elastix registration with $a = 500$. In contrast, the volumes of `T2` and `ADC` were not transformed enough to match the segmentation mask. `TTP` was transformed too much and the dark gray values in the tumor area were enlarged. This effect got stronger with $a = 5000$, as we show with the green mark. Instead, for `T2` the transformation is more adequate with an a of 5000 and is more similar to the outcome of `MuSIC`, as we can see by comparing the highlighted uterus and tumor outlines. Especially for `b1000`, the deformations of the tumor are too strong. In `AUC`, the uterus shows edgy areas in the lower part close to the bladder, that are not present in `VIBE`. Using `MuSIC`, the segmentation fits the tumor and most transformations in the uterus area look similar to `VIBE`, as we show with the purple uterus outline. `ADC` potentially needs stronger uterus adaptations as delineated by the yellow marks.

When applying deformable registration without segmentation mask, the structure that is aligned best is often the uterus, as it has the highest contrast to other structures. In many cases, non-rigid transformations are applied also to the tumor and the tumor is transformed to fit the tumor border in VIBE. In other cases, the transformation is based on other image features and deforms the tumor in the image. The transformations of `TTP` and `b1000` in the second use case (Figure 7.7) are good examples for the phenomenon of tumor deformation. By using segmentation masks as landmarks, we can avoid these tumor deformations.

7.2.3 Summary

In this chapter, we showed example outcomes of `MuSIC` in comparison with several settings of Elastix that do not involve landmarks. The quality of registration outcomes for our data was assessed visually together with our medical experts. We can assume a correct alignment with `MuSIC` especially for rigid registration, since the transformation applied to the images reflects the mask placement by the medical experts. For deformable registration, the quality is more difficult to assess since the transformations can happen on a local scale. A more detailed examination was necessary and our magic lens was helpful in that regard. We highlighted interesting areas that were visible in the depicted sequences for a visual outcome comparison of the approaches. In general, our medical experts were satisfied by the results. The provided examples underline the robustness of our landmark-based approach with `MuSIC`. The results we show using only Elastix were the best we could obtain. For deformable registration, we also tested values of a between 500 and 5000, but all experiments yielded results where some sequences were transformed too much, while others would have needed stronger transformations. Furthermore, also a

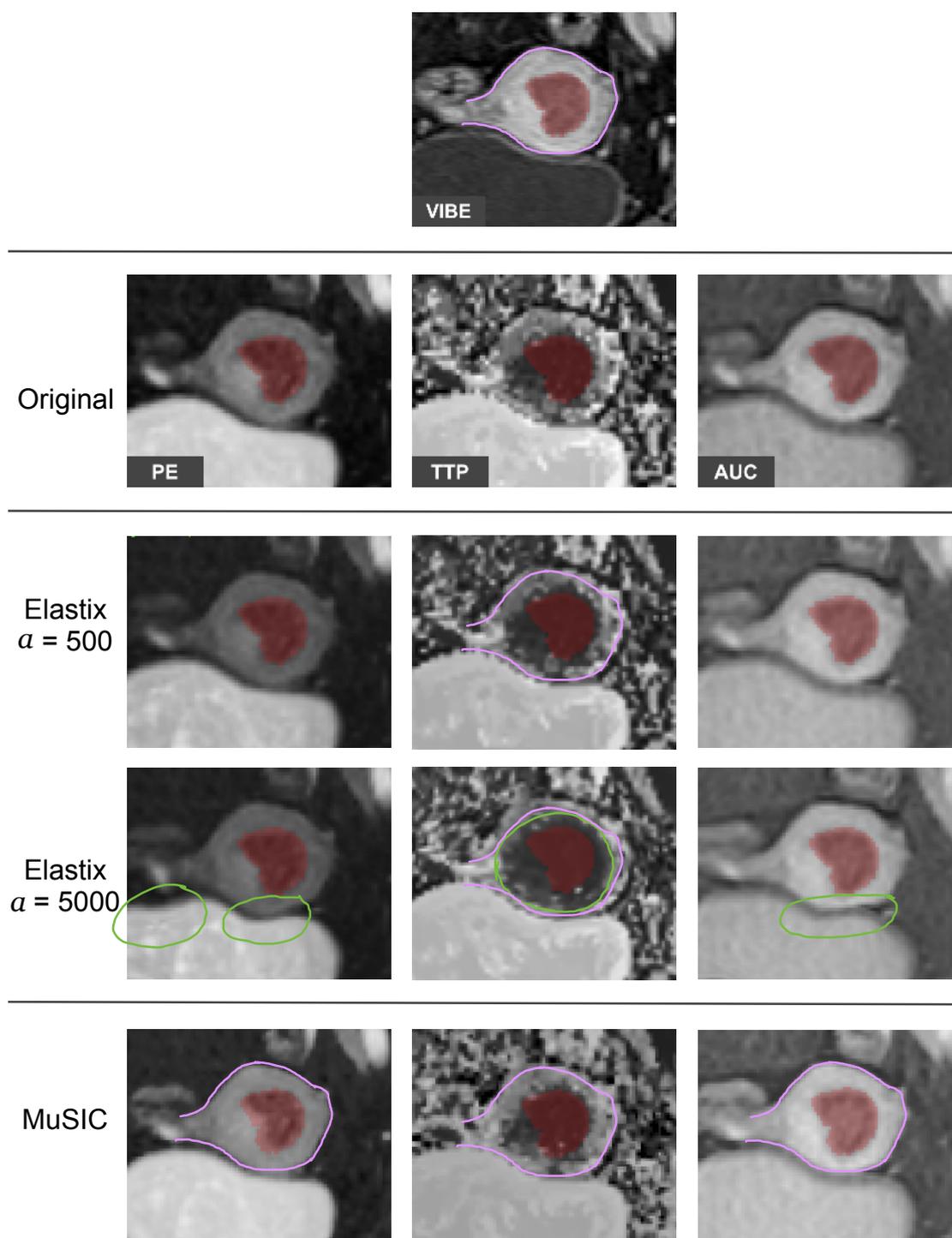


Figure 7.7: Comparison of deformable registration outcomes between only Elastix and MuSIC of Case 2 (PE, TTP, AUC). We use two settings for Elastix changing the parameter α which corresponds to the amount of expected deformation. Strong deformations are shown in green. The uterus outline of VIBE is delineated in purple for several sequences.

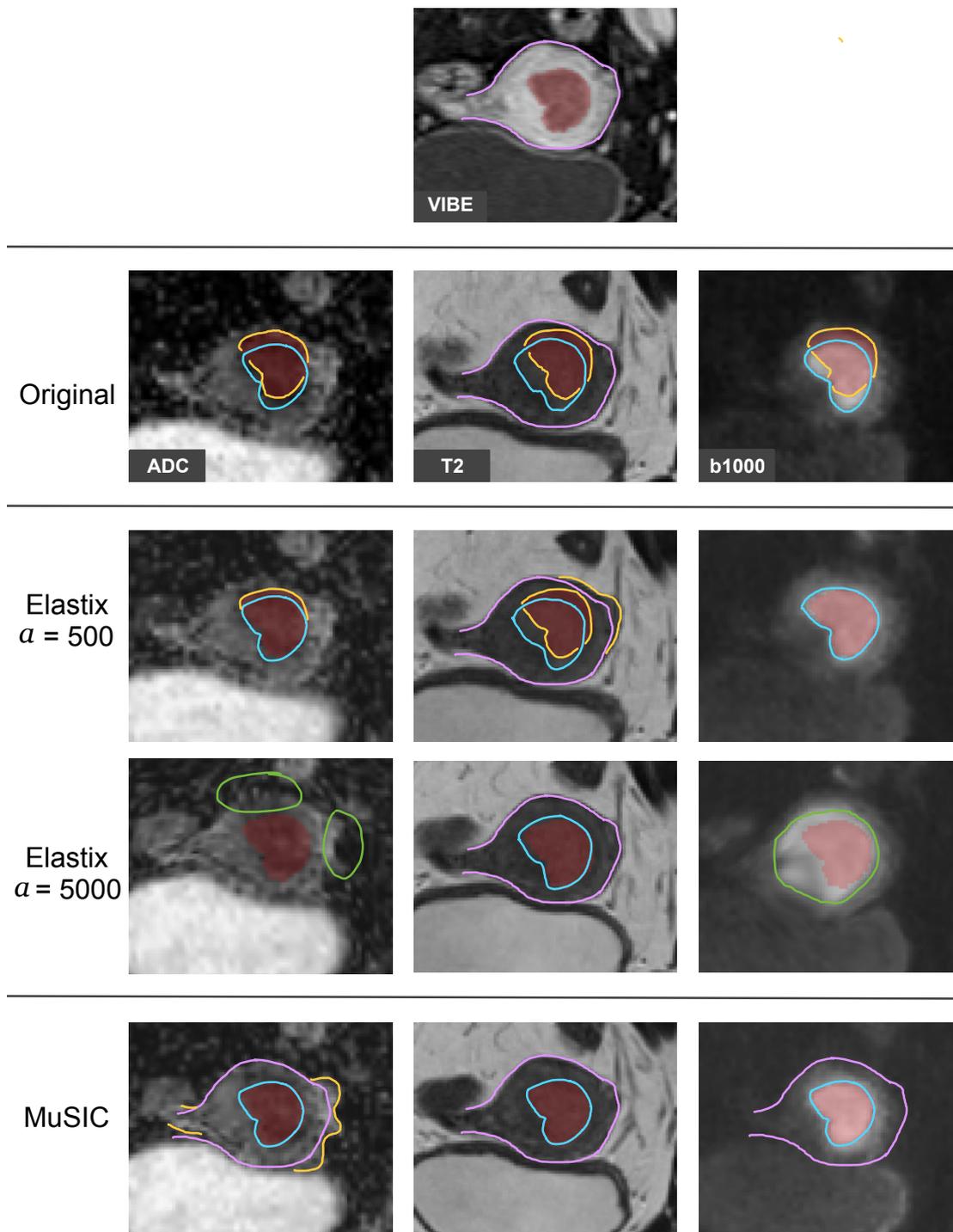


Figure 7.7: Comparison of deformable registration outcomes between only Elastix and MuSIC of Case 2 (ADC, T2, b1000). We use two settings for Elastix changing the parameter α which corresponds to the amount of expected deformation. The tumor borders in the underlying sequences are highlighted in bright blue. The uterus outline of VIBE is delineated in purple for several resulting sequences. Mismatches are delineated in yellow. Strong deformations are shown in green.

per-sequence-setup would not lead to satisfying results, as the required settings also change between two cases. Therefore, we would have to tune the parameters for each case and sequence individually to obtain optimal results.

In contrast, according to our medical experts, the results of [MuSIC](#) are less sensitive to patient- and sequence-specific characteristics, since the registration is based on matching landmarks. For future research in radiomic tumor profiling, it could be important to also evaluate the robustness of [MuSIC](#) for deformable registration in a larger cohort, especially for fuzzy borders as, for example, in b1000.

Usability Analysis

In order to prove the usability of MuSIC, we conducted a qualitative analysis with domain experts. After an introduction and demonstration, the participants were encouraged to try out the tool themselves. Furthermore, we let them also use the tool 3D Slicer [PHK04] for a comparison. 3D Slicer includes a functionality for interactive image registration based on landmarks set by the user. We chose a comparison with landmark-based registration as it is often used as a gold-standard and regarded as the most reliable method, especially when evaluating registration algorithms [PME16]. For evaluation, we used the System Usability Scale (SUS) and a MuSIC-specific evaluation form.

Participants Seven experts took part in the user evaluation. Among those were four female and three male participants. Five participants are medical experts (M1-M5). M1 and M2 are radiologists in research and are PhD candidates in the field of gynecological cancer. Both of them were involved during the development of MuSIC for medical questions regarding the data and preferred workflows. M3 is a postdoctoral researcher with more than ten years of experience in pre-clinical imaging. M4 is a medical physicist and also a PhD candidate in cancer imaging, and M5 is a radiographer.

M1-M5 represent the main target user group for this application. Apart from the medical expert participants, we also included two AI experts (A1 and A2) in the usability evaluation. Multi-sequential ML segmentation could potentially benefit from better co-registration. Therefore, we think AI experts could represent another user group for the tool. A1 is a PhD candidate in ML for cancer imaging and has a background in software engineering. Due to the experience in software development, the feedback in terms of usability is also very valuable. A2 is a postdoctoral researcher in the same field and has experience with co-registration.

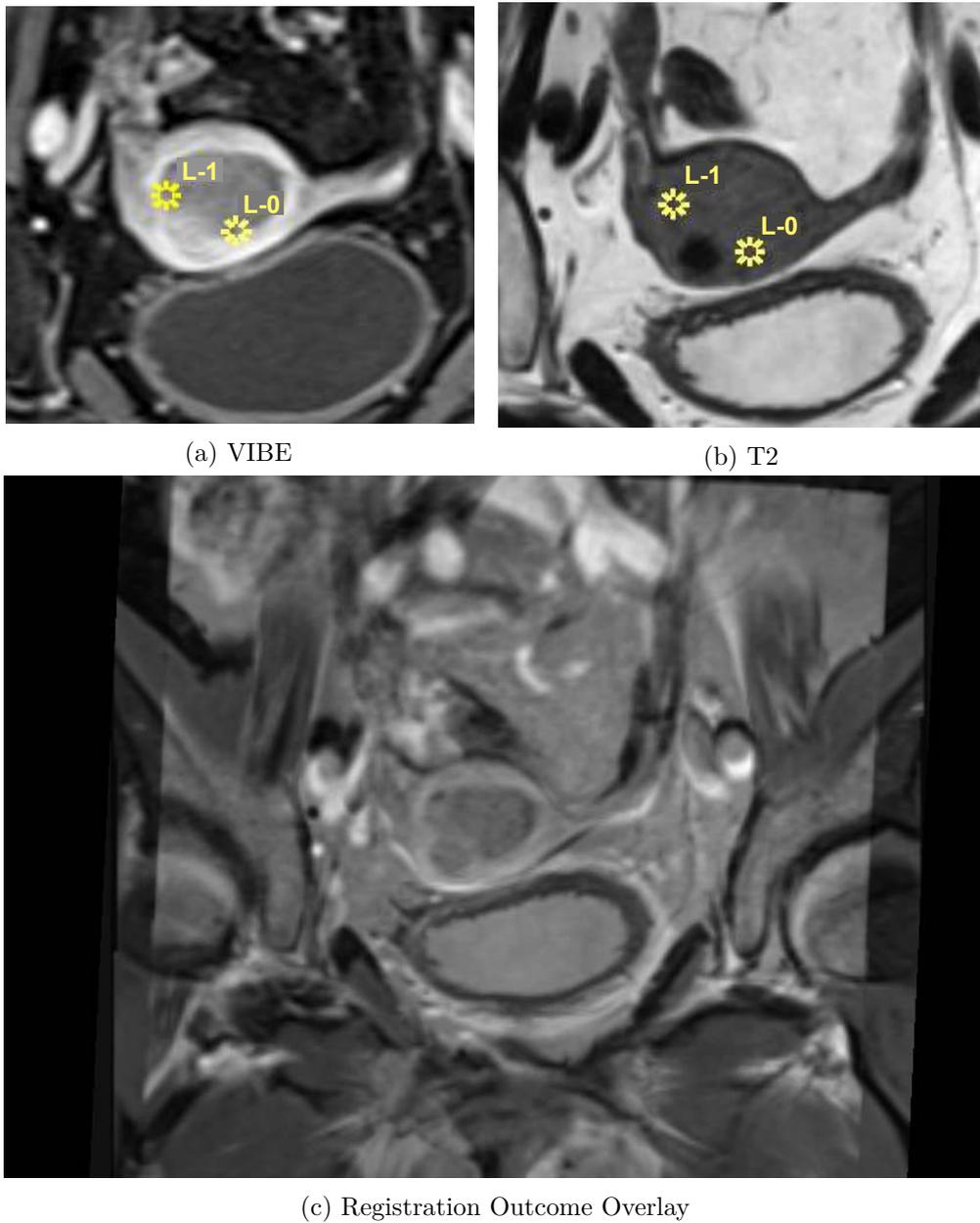


Figure 8.1: Landmark-based registration with 3D Slicer [PHK04]. (a) shows the fixed image with 2 landmarks, (b) the moving image with the corresponding landmarks and (c) the outcome visualization as transparent overlay.

8.1 Evaluation Procedure

In total, the procedure took around 45 minutes per participant. For each participant, we first introduced the tool by explaining the purpose and possible user interactions in a live demonstration for one use case of our data set. We encouraged the participants to ask questions and express their opinions and comments at any time during the evaluation. Then, the participants had the opportunity to use the application and process one use case by trying out the different interaction possibilities. After working with MuSIC, we also shortly introduced them to landmark-based registration with 3D Slicer [PHK04]. They were encouraged to try out 3D Slicer as well by setting landmarks in the reference and one other sequence of their choice. The results were presented as a transparent overlay, and in a checkerboard visualization since we combine the idea of a checkerboard comparison with an interactive lens visualization [BSP⁺93]. We show the landmark-based registration approach with two volumes together with the transparent overlay of the registered volumes in Figure 8.1. Figure 8.2 depicts a 6x6 checkerboard visualization to compare two volumes. Finally, we asked the participants to fill out our evaluation form with the statements presented in Table 8.1.

Evaluation Form Besides the think aloud protocol, we prepared 23 statements for the evaluation form to identify the impact of our tool and how experts benefit from using the different features of the application. The participants assigned a score to each statement on a five-point Likert scale. 1 corresponds to strong disagreement and a maximum score of 5 to strong agreement. Besides the 23 MuSIC-specific statements, we also asked the participants to fill out the SUS evaluation form with ten statements that are formulated in the same manner and can be answered by indicating a score on a Likert scale. The SUS as introduced by Brooke et al. [Bro96] evaluates the usability of a system covering several components: *effectiveness*, examining the outcome quality and if the tool fulfills its purpose, *efficiency* to determine how easy the tool can be used, and *satisfaction*, including personal taste and opinions. It is represented by a score between 0 and 100. A score above 68 is considered to be good, and above 80.5 as excellent. The statements and the results are shown in Table 8.1. In both our own and the SUS form, several questions are formulated in a negative way. In Table 8.1, they are marked with a star and their score has been inverted.

We grouped the statements in the following manner:

- **G1 - G11:** General statements & interaction
 - *G1 - G3:* General usability
 - *G4 - G6:* Layout customization
 - *G7 - G8:* Grouping functionality
 - *G9 - G11:* Magic lens
- **S1 - S5:** Segmentation mask interaction

- **O1 - O3**: Outcome evaluation
- **C1 - C4**: Comparison to 3D Slicer (landmark-based approach)
- **SUS**: System Usability Scale



Figure 8.2: 6x6 Checkerboard visualization to inspect the alignment of two volumes (VIBE and T2). Generated with 3D Slicer [PHK04].

8.2 Evaluation Results

Overall, we received very positive feedback for our tool. Considering the results from Table 8.1, we infer that MuSIC has the potential to be a valuable tool for multi-sequential co-registration. Our participants especially appreciated the magic lens in addition to a side-by-side view for a more detailed comparison, the outcome preview, and a final evaluation of the results. The radiologists were also in favor of the possibility to group sequences and process them simultaneously if needed. Currently, for the task of segmentation, they have to open multiple windows to view several sequences next to each

Table 8.1: Results of our usability analysis showing the response of the participants to our questionnaire on a five-point Likert scale, where the different scores correspond to: 1: strongly disagree, 2: disagree, 3: neither agree nor disagree, 4: agree and 5: strongly agree. Statements marked with a star are rephrased to the positive form in this table with inverted scores for presentation purposes. In the rightmost column, average values are presented. The last row provides the results of the SUS questionnaire.

	Statements	M1	M2	M3	M4	M5	A1	A2	Avg.
G1	I see myself using MuSIC in the future.	5	5	4	5	5	3	4	4,43
G2	Defining a reference view is clear and intuitive.*	5	5	5	5	5	4	5	4,86
G3	The linked slice view interactions (zooming, scrolling, moving) are intuitive.	5	2	5	5	5	5	5	4,57
G4	Reordering the sequences in the grid view is helpful to visualize related modalities next to each other.	4	4	5	5	5	4	5	4,57
G5	It is helpful to resize the sequences in the grid view.*	5	5	5	4	4	5	5	4,71
G6	Switching one sequence of the grid to the main view provides me more details when needed.	5	4	5	5	5	5	5	4,86
G7	Grouping multiple sequences is quick and intuitive.	4	5	4	5	5	4	5	4,57
G8	After grouping multiple sequences it is clear which sequences are grouped.*	5	5	5	5	4	5	5	4,86
G9	Activating and interacting with the magic lens is easy and intuitive.*	5	5	5	5	5	5	5	5
G10	The magic lens helps me to determine differences in tumor position between sequences.*	5	5	5	5	5	5	5	5
G11	It is easier to assess differences in tumor position with the integrated visualization of two sequences via the magic lens than having two sequences next to each other.	4	3	5	5	4	5	5	4,43
S1	Moving the segmentation in X, Y and Z direction using the keyboard is intuitive.	5	5	5	5	5	4	5	4,86
S2	Rotating the segmentation using the keyboard is intuitive.	5	5	4	5	5	5	3	4,57
S3	The possibility to change the opacity of the segmentation helps me to determine the segmentation's fit.*	5	5	4	5	5	5	5	4,86
S4	The three slicing directions in the main view help me to understand the exact position in 3D.	3	5	4	5	5	5	5	4,57
S5	It is helpful being able to reset the segmentation movement of the currently processed sequence.*	5	5	5	5	5	5	4	4,86
O1	Based on the updated slice view for each sequence and the original segmentation position, I can see if the registration is performed correctly.	5	5	5	5	5	5	5	5
O2	The magic lens is useful to check the quality of the alignment after registration.*	5	5	5	5	5	4	5	4,86
O3	It is useful to be able to further adapt the registration if needed.	5	5	5	5	5	5	5	5
C1	I prefer to process multiple sequences at a time because it is easier to keep track of differences in the tumor position.	5	5	4	5	3	4	5	4,43
C2	I think it saves time to visualize and process multiple sequences at a time.*	5	5	4	5	5	4	5	4,71
C3	It is helpful to move a predefined segmentation mask instead of setting landmarks in pathological data.	5	5	5	5	5	5	5	5
C4	I prefer the magic lens visualization over a checkerboard.*	5	5	5	5	5	5	5	5
SUS	System Usability Scale	97.5	100	97.5	100	97.5	85	90	95.4

other, where linked user interaction is limited. All experts agreed on the clear structure and layout without visual overload even though many sequences are shown. M5 had been using multiple tools for both segmentation and registration and thinks MuSIC is very intuitive and does not require major instructions to use. Everyone strongly preferred our approach as an alternative to setting manual landmarks. When we asked about the reason, the participants answered that it can be very difficult to define landmarks in pathological data, as the tumor can appear slightly different in the different sequences due to imaging physics. So using a mask as sort of pre-defined landmarks makes this task easier, while being equally accurate.

During the testing phase of the evaluation, our participants expressed several wishes. The images in our data set are acquired in an axial oblique direction, perpendicular to the main axis of the uterus. Currently, we slice the volume perpendicular to this acquisition

direction for showing the volume in the other two slice views. One of our radiologists would prefer to see the other two slicing views as real axial and sagittal slices according to the body axes (slicing from the front to the back and from left to right instead of oblique slicing), as this would be more intuitive. Furthermore, three participants wish for orientation lines indicating the current position in the three slicing directions. M2 would have wished for a less sensitive zooming in the slice view interactions. The customizable layout was rated favourably. One participant would like to have a clearer visualization regarding where to click in order to drag or resize a component. Generally, our participants thought that the interaction with the segmentation mask via keyboard is intuitive. Especially for translation, two participants expressed the idea to additionally have mouse interaction to also reduce the time needed for bigger translations.

Among the medical experts, we achieved a very high **SUS**. For our **AI** experts, this score is slightly lower. One of them would prefer medical experts to do the task, since sometimes it can be difficult to find a good fit for the segmentation, for example in case of low contrast. The other **AI** expert instead would like to use the tool. He states that their automated segmentation algorithms could strongly benefit from better co-registration, even if their manual segmentation fitting includes more errors than if performed by a medical expert. One wish would be to include the possibility to change registration settings in the tool, for example to decide whether to perform rigid or deformable registration. Furthermore, one **AI** expert stated that for iterative registration adaptation, it would be useful to store previous transformations and always register the original images instead of already processed ones in order to avoid multiple interpolation and smoothing steps.

In general, our participants were enthusiastic about MuSIC and hope to use it soon to improve co-registration. Four of them expressed the wish to not only use it for co-registration, but also for *simultaneous* segmentation. Currently, our medical experts have no tool available that includes intuitive multi-modal segmentation visualization methods. Therefore, they mainly draw the segmentation on one of the sequences. When using our tool, however, they found that some parts of the segmentation might require adaptations because several tissue characteristics become visible throughout the various sequences. In these cases, they would strongly appreciate being able to adapt the segmentation mask.

Discussion & Conclusion

In this chapter, we conclude our work by providing a summary and discuss the approach and results of our tool. Furthermore, we present ideas for interesting directions regarding future work and the potential of [MuSIC](#) to be used in research practice.

9.1 Summary

In this thesis, we present [MuSIC](#), a web-based tool for interactive co-registration by adjusting the position of pre-defined tumor segmentation masks. The application supports radiologists in research in the task of co-registration of cancer imaging data, especially for cases where automatic co-registration fails. In our data set, we have one segmentation mask available that fits one reference modality (in our case [VIBE](#)). We propose an automatic search of the optimal segmentation positions in the other available imaging modalities by a simulated annealing optimization. Users can interactively adapt the resulting positions by translating and rotating the masks. We support the user in this task with multi-modal visualization and interaction techniques. A magic lens allows for an interactive and local comparison on demand. The lens updates on transformation of the segmentation mask and allows for assessing the alignment of different structures. Therefore, we enable a look in the future, so the user can inspect the structure alignment after rigid registration already during the interactive mask fitting. Finally, we register the sequences based on the segmentation transformations and use the corresponding voxels as landmarks. In contrast to other tools, we therefore have pre-defined landmarks, and we offer the possibility to process multiple sequences simultaneously.

From our usability analysis we conclude that [MuSIC](#) can be very valuable to improve co-registration results. Our participants appreciate the intuitive interface, interaction, and the multi-modal visualization. At the [MMIV](#) we collaborate with people from different fields, among them many medical experts, who would like to use [MuSIC](#) in the future. We further included [AI](#) experts in our usability analysis as they benefit from co-registered

medical imaging data for the purpose of developing multi-modal segmentation algorithms. Furthermore, radiomic tumor profiling also involves [AI](#) approaches where co-registered data is essential. It would be beneficial for [AI](#) experts to not necessarily have to involve medical experts to perform co-registration. Especially for cases where the tumor border is clearly visible throughout the different sequences, the tool can be very useful even for [AI](#) experts to improve the registration quality in the data set. With [MuSIC](#), we provide a solution to the co-registration problem that we can observe in our research group.

9.2 Discussion

Availability of segmentation masks Currently, our tool requires an existing segmentation mask. Generating a mask manually can require a substantial amount of time. However, nowadays, many [ML](#) algorithms for a large number of tumor types exist. The fact that those [ML](#) algorithms can already produce satisfying results that are close to manual segmentation quality for at least one sequence, makes the tool applicable for a broader range of applications.

Segmentation Fit Even if we have one mask available, we cannot guarantee for a perfect fit in all sequences. Varying imaging physics can cause a distinct appearance of the tumor in the different sequences, even though biologically the tumor has the same shape. The initial topic idea of this Master's thesis was to segment the tumor separately in all sequences and analyze the differences. However, analyzing the differences in shape would currently have less impact on radiomic tumor profiling since it has no biological basis. Still, it leads us to the question about the *real* tumor borders. We assume that the segmentation mask for each subject matches the actual tumor borders of the reference sequence. However, working with multiple sequences reveals more characteristics about the tumor and other tissues. It can be hard to tell which voxels belong to the tumor and which not if the appearances of tissues vary throughout the sequences. The advantage of [MuSIC](#) is that users can examine the quality of the segmentation mask in all sequences simultaneously while performing co-registration.

Deformable Registration We allow for both rigid and deformable registration. A potential problem of other deformable approaches is the change of the tumor shape in a sequence, even though we assume its structure is rigid and does not deform. Therefore, altering the tumor shape introduces more uncertainty since it does no longer represent the actual state in the human body. Generally, we can avoid such a deformation with our approach by applying non-rigid transformations mainly to the area outside the tumor mask. However, if we cannot find a perfectly satisfying fit of the segmentation and some parts that belong to the tumor are outside our mask, we might actually transform parts of the tumor. Again, this problem is caused by varying imaging physics. In case of manual segmentations, the lack of multi-modal visualization methods can result in segmentation masks with higher uncertainty. By using our tool, users can become aware of that fact.

Automatic Shape Matching We use [SA](#) to search for the tumor borders automatically. The probability of [SA](#) ending up in local minima of the cost function is less likely than in other optimization approaches. Currently, we search in a margin of 15 voxels around the largest extents of the segmentation in each direction. With this search margin, we assume to have only small variations in the tumor location, which is the case for most subjects in our data set. When increasing the search space to allow for larger changes in the tumor’s position, [SA](#) gets trapped in local minima more often. Furthermore, the search becomes slower. The algorithm already runs for quite a long time with the current search margin. For our largest images with a shape of (232, 256, 88) voxels, it can take up to almost 60 seconds per sequence including the pre-processing on a computer with an Intel Core i9-11900K 3.5 GHz processor and a NVIDIA GeForce RTX 3080 Ti graphics card. The user has to wait until all sequences are processed. Another option would be to run the automatic shape matching offline and load the resulting masks separately. We chose [SA](#) initially to avoid local minima. In our data, we might have many local minima, especially in data with low contrast in the tumor area. Still, it would be interesting to try and compare [SA](#) to other methods like gradient descent and variations of it. For example, stochastic gradient descent is also a probabilistic approach. In general, our tool can also be used without the initial shape matching, since the user interaction with the segmentation mask is intuitive and can be performed quickly.

Limitations The automatic shape matching involves manual work for the histogram trimming to set the cut-off values. We set them once for our data set. However, if the histogram distribution of one sequence differs between multiple subjects, the tumor border might not become visible in deviating sequences. We showed an example of this phenomenon in [Figure 4.5](#) in [Chapter 4](#). Such manual settings make the automatic shape matching less generalizable and more complex to use for other data sets. Improving the pre-processing to avoid manual setting of histogram values could be helpful for a better generalizability. Furthermore, our current approach takes up to almost 60 seconds for one case. If we reduce the number of iterations, the results potentially get less accurate. Therefore, we have to find a good trade-off between time and accuracy.

We use a NIfTI reader [LM16](#) to load the data in JavaScript. If only DICOM data is available, it would be necessary to convert it to NIfTI first. For future work, we would like to shift the data loading to the backend in Python and allow also for loading of DICOM images and other commonly used formats to avoid unnecessary data conversion.

To process multiple sequences, we allow to group them. Currently, we only allow for one group of sequences. This is only a limitation for the automatic shape matching, because all sequences are processed simultaneously. For interactive mask placing, the groups can be changed in between registered separately. In subsequent work, it could be useful to allow for more than one group before the automatic shape matching. In our data set, we have one group of sequences (PE, TTP, AUC). b1000 is related to ADC, but there can be a short amount of time between the acquisition of the sequences. However, it can be sufficiently accurate to process them together in several cases.

9.3 Future Work

One aim for the future is to include [MuSIC](#) into the workflow of our collaborators at [MMIV](#). Apart from the points mentioned in the discussion, the usability analysis yielded most ideas for future work. Our collaborators appreciated the functionality of [MuSIC](#). In addition, they strongly wish to be able to adapt the segmentation mask. In a multi-sequential setup, they found several spots in the segmentation mask they would like to adapt. We show two examples in [Figure 9.1](#), where in T2 they found that the mask includes some voxels that are actually part of a myoma, which was not visible in [VIBE](#). The other example reveals slightly different tissue characteristics in b1000, which need further examinations to decide whether they are actually part of the tumor or not.

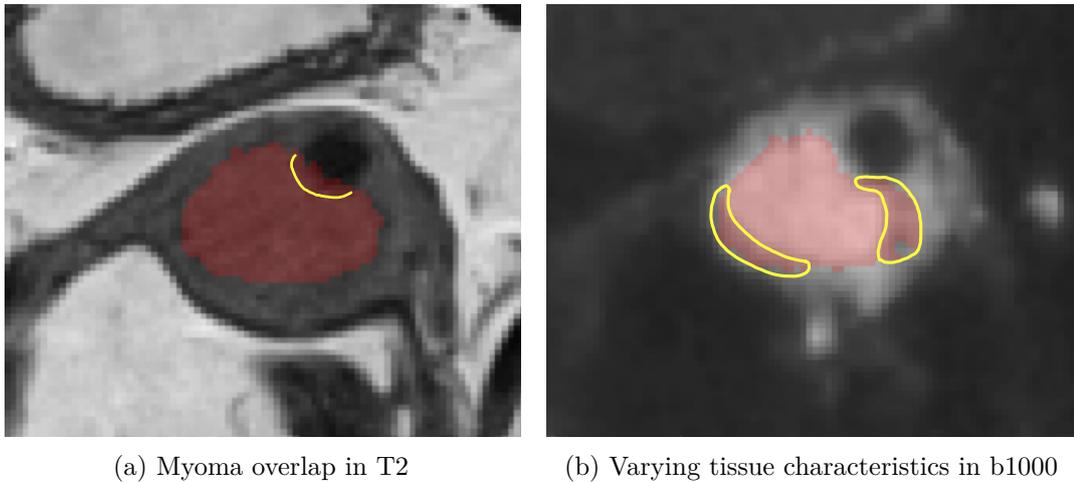


Figure 9.1: Varying imaging characteristics can cause the need to adapt existing segmentation masks. In (a), the segmentation also covers some parts of a myoma (small black circle), while fitting the tumor borders in the rest of the volume. The myoma was not visible in [VIBE](#). The b1000 sequence shows other tissue characteristics. In contrast to other sequences, some parts of the mask seem to be not part of the tumor tissue here and need further examination. We highlight these areas in yellow.

One possibility to handle the lack of a segmentation mask could be to allow for registration without segmentation and shift the volumes instead of the masks. The magic lens could help in this case to check the alignment. While this procedure would still solve the problem of co-registration for difficult cases, we are limited to perform rigid registration. Otherwise, we would run into the risk to deform the tumor. Allowing to start a segmentation from scratch would be the preferred idea to tackle the problem of non-existing masks.

Another way to improve the current workflow for our radiologists would be to automatically detect badly registered cases. They can then be processed with [MuSIC](#). For example, clustering methods could possibly find misaligned volumes in a data set. Our collaborators at [MMIV](#) work on active learning approaches for image segmentation. Active learning

methods select potentially complex cases with high uncertainty to the user and allow for manual corrections. Since one possible cause of high segmentation uncertainty is the misalignment of the underlying volumes, the algorithm would potentially also select those misaligned volumes for user adaptations. The interface we presented with [MuSIC](#) could be used in combination with the functionality to adapt existing masks to present selected cases to the user. The results could be fed back to the algorithm after both adapting the masks and registering the images. This approach has the benefit that we face the problem of segmentation and registration at the same time.

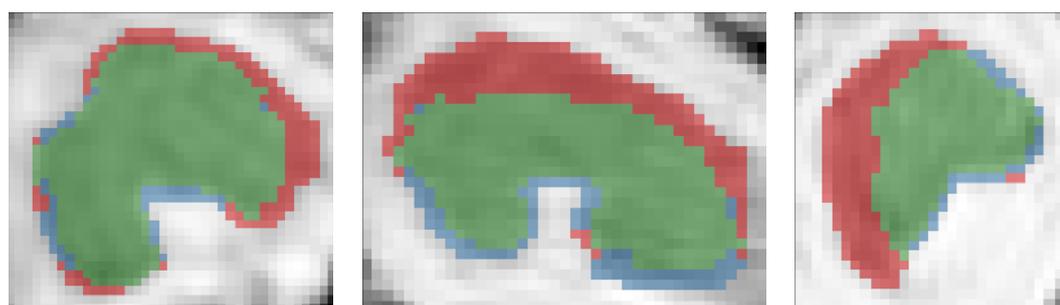
Currently, we leave the resulting registration evaluation to the visual inspection by our medical experts. In our use cases in [Chapter 7](#), they found the results very satisfying and promising. An additional evaluation is not strictly necessary for rigid registration, since an expert places the mask and the volume is transformed only based on the segmentation position. However, we plan to evaluate the accuracy of our deformable approach also numerically. Landmark-based evaluation is usually the preferred way to evaluate registration accuracy [\[PMEM16\]](#). As we could see in our usability analysis in [Chapter 8](#), it can be challenging to define accurate landmarks in pathological structures and soft tissues. The uterus is a mainly round structure, so placing landmarks there is possibly not accurate. Assessing the robustness of our approach by assessing the registration quality of the uterus in a bigger cohort would also be interesting for future research in radiomic tumor profiling.

Another option to assess the registration accuracy numerically would be to create an evaluation data set based on artificial transformations of a publicly available, registered cancer data set. However, synthetically transformed images for evaluation often deviate from realistic scenarios [\[PMEM16\]](#). Furthermore, an evaluation of the automatic shape matching can be valuable. Next to the accuracy, we could assess the time difference for the whole processing time. Additionally, we could possibly use an adapted version of the automatic shape matching for an optimization. Instead of making the user place the segmentation accurately, we could ask for a coarse alignment and find an optimal fit automatically.

A collaborating group in Amsterdam showed interest in our tool to use it for their prostate cancer data set. This collaboration gives us the opportunity to identify necessary adaptations to make the tool usable for other cancer types. Furthermore, an inter-institutional usability analysis with the group’s domain experts could yield valuable feedback to further improve our application.

9.4 Conclusion

In the introduction of this thesis ([Chapter 1](#)), we described the tool ICEVis (Interactive Clustering Exploration for tumor sub-region analysis in multiparametric cancer imaging) [\[MEH⁺20\]](#) that we developed to subdivide tumors with hierarchical clustering. In [Figure 9.2](#) and [Figure 9.3](#), we give an example of how poorly aligned sequences can influence the outcome of this subdivision.

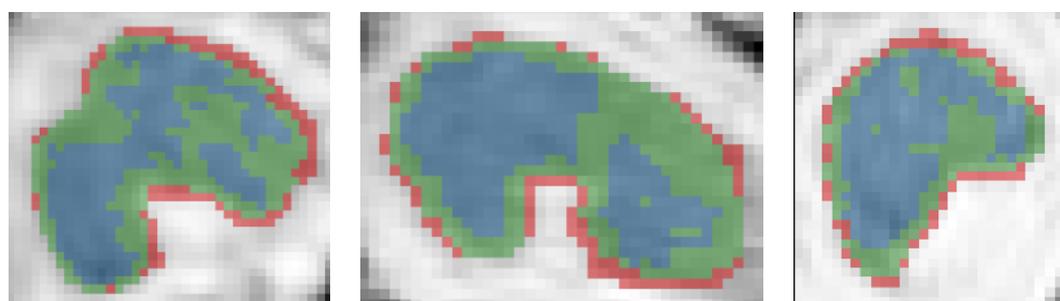


(a) Resulting sub-regions after hierarchical clustering

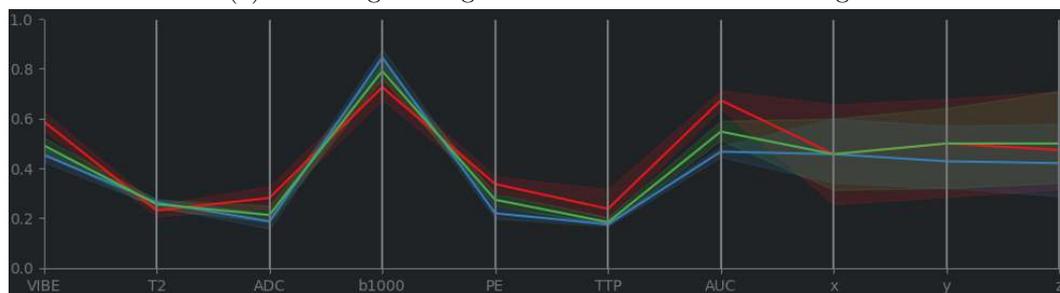


(b) Parallel coordinates plot

Figure 9.2: ICEVis [MEH⁺20] results with three clusters before co-registration.



(a) Resulting sub-regions after hierarchical clustering



(b) Parallel coordinates plot

Figure 9.3: ICEVis [MEH⁺20] results with three clusters after MuSIC co-registration.

Before registering the sequences of this case, the algorithm divides the mask in a red area that is actually not part of the tumor, and a green and blue area inside the tumor tissue (see [Figure 9.2](#)). In the parallel coordinates plot, this becomes visible in the b1000 sequence for the red region and is similar to [Figure 1.2](#). After registering the same case with [MuSIC](#), the subregions are based on features of the actual tumor tissue. In [Figure 9.3](#), we visualize the results. We can observe two areas within the tumor (blue and green), and one red region at the border. In this case, it would be interesting to examine the red region more in detail. There could be either the need to adapt the mask for all sequences, or the red region could, for example, represent enhancing tumor cells that change the image characteristics.

We conclude that our approach can not only improve rigid and deformable registration accuracy of cancer imaging data, but due to its multi-modal visualization support it is also a valuable tool for quality assessment of the segmentation mask. Furthermore, we believe that a deformable registration that restricts local deformations mainly to the area outside the tumor can have a substantial impact on future research in radiomic tumor profiling. A semi-deformable registration that avoids tumor deformation resembles the biological basis of a rigid tumor and soft surrounding tissues better, and allows medical experts to analyze also tissue characteristics of close structures. Last but not least, in the future, we could combine the task of interactive segmentation and registration for difficult cases. In turn, the results can be used to train and further improve [ML](#) algorithms for registration and segmentation.

List of Figures

1.1	A pipeline for radiomic tumor profiling in multi-modal medical imaging .	2
1.2	The tool ICEVis we developed in collaboration with the MMIV for exploring tumor subregions in multi-modal medical imaging. This example shows that we can identify badly registered sequences.	4
2.1	Illustration of the basic pelvic anatomy	8
2.2	T2-weighted sagittal and coronal view on the female pelvic area.	9
2.3	FIGO Staging of endometrial cancer	10
2.4	Depiction of an example slice in the different modalities from the endometrial cancer data set used for this work.	12
2.5	Example slices of a case in the data set with poorly aligned sequences. . .	13
2.6	Classification of medical image segmentation methods.	15
3.1	Intensity-based registration process.	20
3.2	Classification scheme for image registration methods.	21
3.3	Examples of 2D transforms in image registration.	23
3.4	The 3D tool RegistrationShop for interactive registration.	28
3.5	Checkerboard visualization after rigid and deformable registration	32
3.6	Visualizations to assess the accuracy of non-rigid medical image registration methods. Local dissimilarity is shown together with underlying data. . . .	33
3.7	Magic lens visualizations for medical volumetric data exploration	34
4.1	Workflow of the proposed tool including the main steps of, automated and interactive shape matching and co-registration	35
4.2	Preprocessing steps for the automatic shape matching	36
4.3	Comparison of adaptive histogram equalization with different bin sizes . .	37
4.4	Comparison of preprocessing steps for the automatic shape matching . . .	39
4.5	Difficulties of the pre-processing for automatic shape matching	40
4.6	Simulated Annealing algorithm with the number of iterations as stopping criterion	43
4.7	Example steps of the cost function computation in two scenarios	46
4.8	Example outcome of the shape matching in comparison to the original segmentation mask position	47
4.9	3D rotation problem of a plane-based shader approach	48

4.10 Sketch-based visualization of the intersection between the data volume and different planes based on the slice direction	49
4.11 Sketch-based visualization of plane rotation as intersection with the data volume	50
4.12 Magic lens before and after segmentation transformations	52
4.13 Process for the registration with Elastix	53
4.14 Registration components of Elastix	55
5.1 Detailed workflow diagram of the proposed tool	60
5.2 Screenshot of the MuSIC application	61
5.3 Elements of the MuSIC tool bar	62
5.4 The customizable layout of MuSIC	63
5.5 Slice view with button to switch sequence in the main view	64
5.6 Interactive grouping of sequences	65
5.7 Button to reset the segmentation mask to its unmodified state	66
5.8 Rotation interaction with the segmentation mask	66
5.9 MuSIC after grouping and adapting the segmentation positions	67
6.1 Tool structure for rendering of the different scenes	71
6.2 Coordinate color visualization in the WebGL shader	73
6.3 Bundle-based vs. native ESM-based development server	76
7.1 Comparison pre and post shape matching - Case 1	80
7.2 Comparison pre and post shape matching - Case 2	81
7.3 Comparison of rigid registration outcomes between Elastix and MuSIC - Case 1	83
7.4 Comparison of rigid registration outcomes between Elastix and MuSIC - Case 2	84
7.5 Deformations in non-rigid registration without landmarks	85
7.6 Comparison of deformable registration outcomes between Elastix and MuSIC - Case 1 (PE, TTP, AUC)	87
7.6 Comparison of deformable registration outcomes between Elastix and MuSIC - Case 1 (ADC, T2, b1000)	88
7.7 Comparison of deformable registration outcomes between Elastix and MuSIC - Case 2 (PE, TTP, AUC)	90
7.7 Comparison of deformable registration outcomes between Elastix and MuSIC - Case 2 (ADC, T2, b1000)	91
8.1 Landmark-based registration with 3D Slicer	94
8.2 Checkerboard visualization to check the alignment of two volumes	96
9.1 Varying imaging characteristics can cause the need to adapt existing segmentation masks	102
9.2 ICEVis results with three clusters before registration	104
9.3 ICEVis results with three clusters after registration	104

List of Tables

8.1	Results of our usability analysis showing the response of the participants to our questionnaire on a five-point Likert scale, where the different scores correspond to: 1: strongly disagree, 2: disagree, 3: neither agree nor disagree, 4: agree and 5: strongly agree. Statements marked with a star are rephrased to the positive form in this table with inverted scores for presentation purposes. In the rightmost column, average values are presented. The last row provides the results of the SUS questionnaire.	97
-----	---	----

Acronyms

- AC** Active Contours [15](#), [16](#), [22](#)
- ADC** Apparent Diffusion Coefficient [11](#), [82](#), [86](#), [89](#)
- AI** Artificial Intelligence [14](#), [93](#), [98](#)–[100](#)
- ANTs** Advanced Normalization Tools [27](#), [28](#)
- AUC** Area Under the Peak Enhancement Curve [11](#), [36](#), [79](#), [82](#), [86](#), [89](#)
- CNN** Convolutional Neural Network [24](#)–[26](#), [30](#)
- CT** Computed Tomography [1](#), [8](#), [10](#), [16](#), [20](#), [24](#)–[27](#), [31](#)
- DCE** Dynamic Contrast-Enhanced [11](#)
- DL** Deep Learning [2](#), [15](#), [16](#), [24](#), [26](#)
- DWI** Diffusion Weighted Imaging [10](#), [11](#)
- FIGO** International Federation of Gynecology and Obstetrics [8](#)
- GGM** Gaussian Gradient Magnitude [38](#), [45](#), [46](#), [77](#)
- GUI** Graphical User Interface [3](#), [6](#), [29](#)
- ITK** Insight Segmentation and Registration Toolkit [27](#), [28](#)
- ML** Machine Learning [2](#), [5](#), [11](#), [14](#), [24](#), [25](#), [59](#), [93](#), [100](#), [105](#)
- MMIV** Mohn Medical Imaging and Visualization Centre [1](#), [4](#), [5](#), [7](#), [11](#), [14](#), [99](#), [102](#)
- MRI** Magnetic Resonance Imaging [1](#)–[3](#), [7](#), [8](#), [10](#), [11](#), [15](#), [16](#), [20](#), [24](#), [26](#), [30](#), [31](#), [33](#), [36](#), [48](#), [50](#), [59](#)
- MSE** Mean Squared Error [26](#)

MuSIC Multi-Sequential Interactive Co-registration [61](#), [86](#), [89](#), [92](#), [99](#), [100](#), [102](#), [103](#), [105](#)

PE Peak Enhancement [11](#), [36](#), [37](#), [79](#), [82](#), [86](#), [89](#)

PET Positron Emission Tomography [10](#), [16](#), [20](#)

SA Simulated Annealing [30](#), [41](#), [44](#), [82](#), [101](#)

SUS System Usability Scale [93](#), [95](#), [96](#), [98](#)

TTP Time To Peak [11](#), [36](#), [79](#), [82](#), [86](#), [89](#)

TVU Transvaginal Ultrasound [10](#)

US Ultrasound [25](#), [26](#), [30](#)

VIBE Volumetric Interpolated Breath-Hold Examination [11](#), [13](#), [44](#), [59](#), [65](#), [79](#), [86](#), [89](#), [96](#), [99](#), [102](#)

Bibliography

- [AGM17] Charlotte Alibert, Bruno Goud, and Jean-Baptiste Manneville. Are cancer cells really softer than normal cells? *Biology of the Cell*, 109(5):167–189, 2017.
- [APT06] Sigurd Angenent, Eric Pichon, and Allen Tannenbaum. Mathematical methods in medical image processing. *Bulletin of the American Mathematical Society*, 43(3):365–396, 2006.
- [Ash12] John Ashburner. SPM: a history. *Neuroimage*, 62(2):791–800, 2012.
- [ATS09] Brian B Avants, Nick Tustison, and Gang Song. Advanced normalization tools (ANTS). *Insight j*, 2(365):1–35, 2009.
- [AURU⁺16] Fakhre Alam, Sami Ur Rahman, Sehat Ullah, Adnan Khalil, and Aziz Uddin. A review on extrinsic registration methods for medical images. *Technical Journal University of Engineering and Technology Taxila*, 21:110–119, 10 2016.
- [BA04] Walid Ben-Ameur. Computing the initial temperature of simulated annealing. *Computational Optimization and Applications*, 29(3):369–385, 2004.
- [BFS⁺18] Freddie Bray, Jacques Ferlay, Isabelle Soerjomataram, Rebecca L Siegel, Lindsey A Torre, and Ahmedin Jemal. Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, 68(6):394–424, 2018.
- [BK15] Michael S Baggish and Mickey M Karram. *Atlas of Pelvic Anatomy and Gynecologic Surgery, 4th Edition*. Elsevier Health Sciences, 2015.
- [BME99] Paul Butler, Adam WM Mitchell, and Harold Ellis. *Applied Radiological Anatomy, 2nd Edition*. Cambridge University Press, 1999.
- [Bro96] John Brooke. SUS: a "quick and dirty" usability scale. *Usability Evaluation in Industry*, 189(3), 1996.

- [BSP⁺93] Eric A Bier, Maureen C Stone, Ken Pier, William Buxton, and Tony D DeRose. Toolglass and magic lenses: the see-through interface. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, pages 73–80, 1993.
- [ÇAL⁺16] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: learning dense volumetric segmentation from sparse annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 424–432. Springer, 2016.
- [CCCD93] Vicent Caselles, Francine Catté, Tomeu Coll, and Françoise Dibos. A geometric model for active contours in image processing. *Numerische Mathematik*, 66(1):1–31, 1993.
- [CDPRF21] Xiang Chen, Andres Diaz-Pinto, Nishant Ravikumar, and Alejandro F Frangi. Deep learning in medical image registration. *Progress in Biomedical Engineering*, 3(1):012003, 2021.
- [CK09] M Rex Cheung and Karthik Krishnan. Interactive deformation registration of endorectal prostate MRI using ITK thin plate splines. *Academic Radiology*, 16(3):351–357, 2009.
- [CKMK19] Luca Canalini, Jan Klein, Dorothea Miller, and Ron Kikinis. Segmentation-based registration of ultrasound volumes for glioma resection in image-guided neurosurgery. *International Journal of Computer Assisted Radiology and Surgery*, 14(10):1697–1713, 2019.
- [CKS97] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.
- [CSK⁺11] John Congote, Alvaro Segura, Luis Kabongo, Aitor Moreno, Jorge Posada, and Oscar Ruiz. Interactive visualization of volumetric data with WebGL in real-time. In *Proceedings of the 16th International Conference on 3D Web Technology*, pages 137–146, 2011.
- [CV01] Tony F Chan and Luminita A Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001.
- [CW18] Evelyn Chee and Zhenzhou Wu. Airnet: Self-supervised affine registration for 3D medical images using neural networks. *arXiv preprint arXiv:1810.02583*, 2018.
- [CZZ18] Xi Cheng, Li Zhang, and Yefeng Zheng. Deep similarity learning for multimodal medical images. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 6(3):248–252, 2018.

- [Dan12] Brian Danchilla. Three.js framework. In *Beginning WebGL for HTML5*, pages 173–203. Springer, 2012.
- [DCS⁺00] Mark C DeLano, Thomas G Cooper, James E Siebert, Michael J Potchen, and Karthik Kuppusamy. High-b-value diffusion-weighted MR imaging of adult brain: image contrast and apparent diffusion coefficient map features. *American Journal of Neuroradiology*, 21(10):1830–1836, 2000.
- [Dic45] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.
- [EGSEB11] Ahmed Elnakib, Georgy Gimel’farb, Jasjit S Suri, and Ayman El-Baz. Medical image segmentation: a brief survey. In *Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies*, pages 1–39. Springer, 2011.
- [EP18] Koen AJ Eppenhof and Josien PW Pluim. Pulmonary CT registration through supervised learning with convolutional neural networks. *IEEE Transactions on Medical Imaging*, 38(5):1097–1105, 2018.
- [FBYH⁺18] Kristine E Fasmer, Atle Bjørnerud, Sigmund Ytre-Hauge, Renate Grüner, Ingvild L Tangen, Henrica MJ Werner, Line Bjørge, Øyvind O Salvesen, Jone Trovik, Camilla Krakstad, and Ingrid S Haldorsen. Pre-operative quantitative dynamic contrast-enhanced MRI and diffusion-weighted imaging predict aggressive disease in endometrial cancer. *Acta Radiologica*, 59(8):1010–1017, 2018.
- [FLW⁺20] Yabo Fu, Yang Lei, Tonghe Wang, Walter J Curran, Tian Liu, and Xiaofeng Yang. Deep learning in medical image registration: a review. *Physics in Medicine & Biology*, 65(20):20TR01, 2020.
- [FOGM18] Enzo Ferrante, Ozan Oktay, Ben Glocker, and Diego H Milone. On the adaptability of unsupervised CNN-based deformable image registration to unseen image domains. In *International Workshop on Machine Learning in Medical Imaging*, pages 294–302. Springer, 2018.
- [Fou05] Django Software Foundation. Django: The web framework for perfectionists with deadlines. <https://www.djangoproject.com>, GitHub: <https://github.com/django> (accessed: 19-04-2022)., 2005.
- [Fou09] OpenJS Foundation. Node.js. <https://www.npmjs.com>, GitHub: <https://github.com/nodejs/nodejs.org> (accessed: 19-04-2022). Version 16.11.1., 2009.

- [FW01] J Michael Fitzpatrick and Jay B West. The distribution of target registration error in rigid-body point-based registration. *IEEE Transactions on Medical Imaging*, 20(9):917–927, 2001.
- [FWW20] Lingling Fang, Xin Wang, and Lujie Wang. Multi-modal medical image segmentation based on vector-valued active contour models. *Information Sciences*, 513:504–518, 2020.
- [GER⁺12] Ioannis S Gousias, A David Edwards, Mary A Rutherford, Serena J Counsell, Jo V Hajnal, Daniel Rueckert, and Alexander Hammers. Magnetic resonance imaging of the newborn brain: manual segmentation of labelled atlases in term-born and preterm infants. *Neuroimage*, 62(3):1499–1509, 2012.
- [GLH⁺19] Zhe Guo, Xiang Li, Heng Huang, Ning Guo, and Quanzheng Li. Deep learning-based image segmentation on multimodal medical imaging. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 3(2):162–169, 2019.
- [GNBP11] Rocco Gasteiger, Mathias Neugebauer, Oliver Beuing, and Bernhard Preim. The FLOWLENS: A focus-and-context visualization approach for exploration of blood flow in cerebral aneurysms. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2183–2192, 2011.
- [GPME⁺14] Yaron Gordon, Sasan Partovi, Matthias Müller-Eschner, Erick Amarteifio, Tobias Bäuerle, Marc-André Weber, Hans-Ulrich Kauczor, and Fabian Rengier. Dynamic contrast-enhanced magnetic resonance imaging: fundamentals and application to the evaluation of the peripheral perfusion. *Cardiovascular Diagnosis and Therapy*, 4(2):147, 2014.
- [HBR⁺17] Iyas Hamdan, Julien Bert, Catherine Cheze Le Rest, Jean Pierre Tasu, Nicolas Bousson, Antoine Valeri, Guillaume Dardenne, and Dimitris Visvikis. Fully automatic deformable registration of pretreatment MRI/CT for image-guided prostate radiotherapy planning. *Medical Physics*, 44(12):6447–6455, 2017.
- [HDWL⁺21] Erlend Hodneland, Julie A Dybvik, Kari S Wagner-Larsen, Veronika Šoltészová, Antonella Z Munthe-Kaas, Kristine E Fasmer, Camilla Krakstad, Arvid Lundervold, Alexander S Lundervold, Øyvind Salvesen, Bradley J Erickson, and Ingfrid S Haldrosen. Automated segmentation of endometrial cancer on MR images using deep learning. *Scientific Reports*, 11(1):1–8, 2021.
- [HHD⁺21] Erling A Hoivik, Erlend Hodneland, Julie A Dybvik, Kari S Wagner-Larsen, Kristine E Fasmer, Hege F Berg, Mari K Halle, Ingfrid S

- Haldorsen, and Camilla Krakstad. A radiogenomics application for prognostic profiling of endometrial cancer. *Communications Biology*, 4(1):1–12, 2021.
- [HKHH19] Alessa Hering, Sven Kuckertz, Stefan Heldmann, and Mattias P Heinrich. Enhancing label-driven deep deformable image registration with local distance metrics for state-of-the-art cardiac motion tracking. In *Bildverarbeitung für die Medizin 2019*, pages 309–314. Springer, 2019.
- [HKR93] Daniel P Huttenlocher, Gregory A Klanderma, and William J Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.
- [HKY20] Grant Haskins, Uwe Kruger, and Pingkun Yan. Deep learning in medical image registration: a survey. *Machine Vision and Applications*, 31(1):1–18, 2020.
- [HMK⁺00] Bill Hoffman, Ken Martin, Brad King, Dave Cole, Alexander Neundorff, and Clinton Stimpson. CMake (cross-platform make). <https://cmake.org> (accessed: 19-04-2022). Version 3.22.4., 2000.
- [HQY18] Rui Hao, Yan Qiang, and Xiaofei Yan. Juxta-vascular pulmonary nodule segmentation in PET-CT imaging based on an LBF active contour model with information entropy and joint vector. *Computational and mathematical methods in medicine*, 2018, 2018.
- [HS16] Ingrid S Haldorsen and Helga B Salvesen. What is the best preoperative imaging for endometrial cancer? *Current Oncology Reports*, 18(4):25, 2016.
- [HSG⁺14] Ingrid S Haldorsen, Ingunn Stefansson, Renate Grüner, JA Husby, Inger Johanne Magnussen, Henrica Maria Johanna Werner, Ø O Salvesen, Line Bjørge, Jone Trovik, Torfinn Taxt, Lars A Akslen, and Helga Brigitte Salvesen. Increased microvascular proliferation is negatively correlated to tumour blood flow and is associated with unfavourable outcome in endometrial carcinomas. *British Journal of Cancer*, 110(1):107–114, 2014.
- [Hun07] John D Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(03):90–95, 2007.
- [HWS⁺07] Heinz Handels, René Werner, Rainer Schmidt, Thorsten Frenzel, Wei Lu, Daniel Low, and Jan Ehrhardt. 4d medical image computing and visualization of lung tumor mobility in spatio-temporal CT image data. *International Journal of Medical Informatics*, 76:S433–S439, 2007.

- [Jac12] Paul Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, 1912.
- [JAFGGFGG19] Javier Juan-Albarracín, Elies Fuster-Garcia, Germán A García-Ferrando, and Juan M García-Gómez. ONCOhabitats: A system for glioblastoma heterogeneity assessment through MRI. *International Journal of Medical Informatics*, 128:53–61, 2019.
- [JAFGM⁺15] Javier Juan-Albarracín, Elies Fuster-Garcia, Jose V Manjon, Montserrat Robles, F Aparici, L Martí-Bonmatí, and Juan M Garcia-Gomez. Automated glioblastoma segmentation based on a multiparametric structured unsupervised classification. *PLoS One*, 10(5):e0125143, 2015.
- [JBB⁺12] Mark Jenkinson, Christian F Beckmann, Timothy EJ Behrens, Mark W Woolrich, and Stephen M Smith. FSL. *Neuroimage*, 62(2):782–790, 2012.
- [KBD17] András P Keszei, Benjamin Berkels, and Thomas M Deserno. Survey of non-rigid registration tools in medicine. *Journal of Digital Imaging*, 30(1):102–116, 2017.
- [KDSU⁺16] Michael D Ketcha, Tharindu De Silva, Ali Uneri, Gerhard Kleinszig, Sebastian Vogt, Jean-Paul Wolinsky, and Jeffrey H Siewerdsen. Automatic masking for robust 3D-2D image registration in image-guided spine surgery. In *Medical Imaging 2016: Image-Guided Procedures, Robotic Interventions, and Modeling*, volume 9786, page 97860A. International Society for Optics and Photonics, 2016.
- [KGV83] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [KK19] Avinash Kori and Ganapathi Krishnamurthi. Zero shot learning for multi-modal real time image registration. *arXiv preprint arXiv:1908.06213*, 2019.
- [KM04] Peter B Kingsley and W Gordon Monahan. Selection of the optimum b-factor for diffusion-weighted magnetic resonance imaging assessment of ischemic stroke. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 51(5):996–1001, 2004.
- [Kop12] Tobias Koppers. Webpack. <https://webpack.js.org>, GitHub: <https://github.com/webpack/webpack> (accessed: 19-04-2022)., 2012.

- [KP06] Dow-Muh Koh and Anwar R Padhani. Diffusion-weighted MRI: a new functional clinical technique for tumour imaging. *The British Journal of Radiology*, 79(944):633–635, 2006.
- [KPSV09] Stefan Klein, Josien PW Pluim, Marius Staring, and Max A Viergever. Adaptive stochastic gradient descent optimisation for image registration. *International Journal of Computer Vision*, 81(3):227, 2009.
- [KS18] Stefan Klein and Marius Staring. Elastix - the manual. <https://usermanual.wiki/Document/elastix490manual.1389615963/view>, 2018. Last accessed: 2022-02-22.
- [KSM⁺09] Stefan Klein, Marius Staring, Keelin Murphy, Max A Viergever, and Josien PW Pluim. Elastix: a toolbox for intensity-based medical image registration. *IEEE Transactions on Medical Imaging*, 29(1):196–205, 2009.
- [KWT88] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
- [KWW18] Eamon Koh, Edward RJ Walton, and Phil Watson. VIBE MRI: An alternative to CT in the imaging of sports-related osseous pathology? *The British Journal of Radiology*, 91(1088):20170815, 2018.
- [KYWH12] Can Kirmizibayrak, Yeny Yim, Mike Wakid, and James Hahn. Interactive visualization and analysis of multimodal datasets for surgical applications. *Journal of Digital Imaging*, 25(6):792–801, 2012.
- [LCIB13] Bradley Christopher Lowekamp, David T Chen, Luis Ibáñez, and Daniel Blezek. The design of SimpleITK. *Frontiers in Neuroinformatics*, 7:45, 2013.
- [LGS99] Thomas Martin Lehmann, Claudia Gonner, and Klaus Spitzer. Survey: Interpolation methods in medical image processing. *IEEE Transactions on Medical Imaging*, 18(11):1049–1075, 1999.
- [LL19] Alexander Selvikvåg Lundervold and Arvid Lundervold. An overview of deep learning in medical imaging focusing on MRI. *Zeitschrift für Medizinische Physik*, 29(2):102–127, 2019.
- [Llo82] Stuart Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
- [LM16] Jack L Lancaster and Michael J Martinez. Mango nifti-reader-js. GitHub: <https://github.com/rii-mango/NIFTI-Reader-JS> (accessed: 25-03-2022): Version 0.5.4., 2016.

- [LSBP18] Kai Lawonn, Noeska N Smit, Katja Bühler, and Bernhard Preim. A survey on multimodal medical data visualization. In *Computer Graphics Forum*, volume 37, pages 413–438. Wiley Online Library, 2018.
- [LXN⁺05] Guang Li, Huchen Xie, Holly Ning, Jacek Capala, Barbara C Arora, C Norman Coleman, Kevin Camphausen, and Robert W Miller. A novel 3D volumetric voxel registration technique for volume-view-guided image registration of multiple imaging modalities. *International Journal of Radiation Oncology - Biology - Physics*, 63(1):261–273, 2005.
- [LYZW18] Jun Lv, Ming Yang, Jue Zhang, and Xiaoying Wang. Respiratory motion correction for free-breathing 3D abdominal MRI using CNN-based image registration: A feasibility study. *The British Journal of Radiology*, 91(1083):20170788, 2018.
- [Mah18] Dwarikanath Mahapatra. GAN based medical image registration. *arXiv preprint arXiv:1805.02369*, 2018.
- [MBSK16] Kasper Marstal, Floris Berendsen, Marius Staring, and Stefan Klein. SimpleElastix: A user-friendly, multi-lingual library for medical image registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 134–142, 2016. <https://simpleelastix.github.io>, GitHub: <https://github.com/SuperElastix/elastix> (accessed: 19-04-2022).
- [MEH⁺20] Eric Mörth, Tanja Eichner, Ingfrid Salvesen Haldorsen, Stefan Bruckner, and Noeska Smit. ICEVis: Interactive clustering exploration for tumor sub-region analysis in multiparametric cancer imaging. Submitted for Publication., 2020.
- [ML13] Kouichi Matsuda and Rodger Lea. *WebGL Programming Guide: Interactive 3D Graphics Programming with WebGL*. Addison-Wesley, 2013.
- [MLC⁺16] Philippe Morice, Alexandra Leary, Carien Creutzberg, Nadeem Abu-Rustum, and Emile Darai. Endometrial cancer. *The Lancet*, 387(10023):1094–1108, 2016.
- [MLI⁺14] Matthew Michael McCormick, Xiaoxiao Liu, Luis Ibanez, Julien Jomier, and Charles Marion. ITK: enabling reproducible research and open science. *Frontiers in Neuroinformatics*, 8:13, 2014.
- [MP10] Fabrice Michel and Nikos Paragios. Image transport regression using mixture of experts and discrete Markov Random Fields. In *2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1229–1232. IEEE, 2010.

- [MR] Artem Mikheev and Henry Rusinek. Firevoxel. <https://firevoxel.org>. Last accessed: 2022-03-07. Radiology Department, NYU School of Medicine.
- [MRR⁺53] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.
- [MS89] David Bryant Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.
- [MT96] Tim McInerney and Demetri Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91–108, 1996.
- [MV98] JB Antoine Maintz and Max A Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.
- [MWLH⁺20] Eric Mörth, Kari Wagner-Larsen, Erlend Hodneland, Camilla Krakstad, Ingrid S Haldorsen, Stefan Bruckner, and Noeska N Smit. RadEx: Integrated visual exploration of multiparametric studies for radiomic tumor profiling. In *Computer Graphics Forum*, volume 39, pages 611–622. Wiley Online Library, 2020.
- [MZ04] Ilias Maglogiannis and Elias Zafiroopoulos. Automated medical image registration using the simulated annealing algorithm. In *Hellenic Conference on Artificial Intelligence*, pages 456–465. Springer, 2004.
- [npm10] npm, Inc. npm: A JavaScript package manager. <https://www.npmjs.com>, GitHub: <https://github.com/npm/cli> (accessed: 19-04-2022). Version 8.0.0., 2010.
- [Oli06] Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.
- [OT14] Francisco PM Oliveira and Joao Manuel RS Tavares. Medical image registration: a review. *Computer Methods in Biomechanics and Biomedical Engineering*, 17(2):73–93, 2014.
- [PAA⁺87] Stephen M Pizer, E Philip Amburn, John D Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B Zimmerman, and Karel Zuiderveld. Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*, 39(3):355–368, 1987.

- [Par12] Tony Parisi. *WebGL: Up and running*. O’Reilly Media, Inc., 2012.
- [PC16] Jyotsna Pundir and Arri Coomarasamy. *Gynaecology: Evidence-Based Algorithms*. Cambridge University Press, 2016.
- [Pec09] Sergio Pecorelli. Revised FIGO staging for carcinoma of the vulva, cervix, and endometrium. *International Journal of Gynecology & Obstetrics*, 105(2):103–104, 2009.
- [PHK04] Steve Pieper, Michael Halle, and Ron Kikinis. 3D Slicer. In *2004 2nd IEEE International Symposium on Biomedical Imaging: Nano to Macro*, pages 632–635. IEEE, 2004.
- [PMEM16] Josien PW Pluim, Sascha EA Muenzing, Koen AJ Eppenhof, and Keelin Murphy. The truth is hard to make: Validation of medical image registration. In *23rd International Conference on Pattern Recognition (ICPR)*, pages 2294–2300, 2016.
- [RBR⁺18] Stefania Rizzo, Francesca Botta, Sara Raimondi, Daniela Origgi, Cristiana Fanciullo, Alessio Giuseppe Morganti, and Massimo Bellomi. Radiomics: the facts and the challenges of image analysis. *European Radiology Experimental*, 2(1):1–8, 2018.
- [Rea] React. React-grid-layout. GitHub: <https://github.com/react-grid-layout/react-grid-layout> (accessed: 24-03-2022). Version 1.3.4.
- [RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.
- [RLL⁺99] Neil M Rofsky, Vivian S Lee, Gerhard Laub, Michael A Pollack, Glenn A Krinsky, David Thomasson, Michael M Ambrosino, and Jeffrey C Weinreb. Abdominal MR imaging with a volumetric interpolated breath-hold examination. *Radiology*, 212(3):876–884, 1999.
- [Rog00] Jadwiga Rogowska. Overview and fundamentals of medical image segmentation. *Handbook of Medical Imaging, Processing and Analysis*, pages 69–85, 2000.
- [Ron10] Armin Ronacher. Flask. *Web Development, One Drop at a Time*, 2010. <https://flask.palletsprojects.com/en/2.0.x/> (accessed: 24-03-2022).
- [RS10] Daniel Rueckert and Julia A Schnabel. Medical image registration. In *Biomedical Image Processing*, pages 131–154. Springer, 2010.

- [RS14] Daniel Rueckert and Julia A. Schnabel. *Registration and Segmentation in Medical Imaging*, pages 137–156. Springer, 2014.
- [RvdHD⁺15] Renata Raidou, Uulke A van der Heide, Cuong Viet Dinh, Ghazaleh Ghobadi, Jesper Follsted Kallehauge, Marcel Breeuwer, and Anna Vilanova. Visual analytics for the exploration of tumor tissue characterization. In *Computer Graphics Forum*, volume 34, pages 11–20. Wiley Online Library, 2015.
- [SDP13] Aristeidis Sotiras, Christos Davatzikos, and Nikos Paragios. Deformable medical image registration: A survey. *IEEE Transactions on Medical Imaging*, 32(7):1153–1190, 2013.
- [Sen01] Rifat Nuri Sener. Diffusion MRI: apparent diffusion coefficient (ADC) values in the normal brain and a classification of brain disorders based on ADC values. *Computerized Medical Imaging and Graphics*, 25(4):299–326, 2001.
- [SFF06] Hanno Schumacher, Astrid Franz, and Bernd Fischer. Weighted medical image registration with automatic mask generation. In *Medical Imaging 2006: Image Processing*, volume 6144, pages 740–747. SPIE, 2006.
- [SFJ⁺16] Matthias Schlachter, Tobias Fechter, Miro Jurisic, Tanja Schimek-Jasch, Oliver Oehlke, Sonja Adebahr, Wolfgang Birkfellner, Ursula Nestle, and Katja Bühler. Visualization of deformable image registration quality using local image dissimilarity. *IEEE Transactions on Medical Imaging*, 35(10):2319–2328, 2016.
- [SGBM⁺16] Martin Simonovsky, Benjamín Gutiérrez-Becker, Diana Mateus, Nassir Navab, and Nikos Komodakis. A deep metric for multimodal registration. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 10–18. Springer, 2016.
- [SHS⁺14] Noeska N Smit, Berend Klein Haneveld, Marius Staring, Elmar Eismann, Charl P Botha, and Anna Vilanova. RegistrationShop: An interactive 3D medical volume registration system. In Ivan Viola, Katja Buehler, and Timo Ropinski, editors, *Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association, 2014.
- [SLW⁺10] Gregory C Sharp, Rui Li, John Wolfgang, G Chen, Marta Peroni, Maria Francesca Spadea, Shinichro Mori, Junan Zhang, James Shackelford, and Nagarajan Kandasamy. Plastimatch: an open source software suite for radiotherapy image processing. In *Proceedings of the XVIth International Conference on the use of Computers in Radiotherapy (ICCR), Amsterdam, Netherlands*, 2010.

- [SMM⁺18] Christodoulidis Stergios, Sahasrabudhe Mihir, Vakalopoulou Maria, Chassagnon Guillaume, Revel Marie-Pierre, Mougiakakou Stavroula, and Paragios Nikos. Linear and deformable image registration with 3d convolutional neural networks. In *Image Analysis for Moving Organ, Breast, and Thoracic Images*, pages 13–22. Springer, 2018.
- [SRM⁺19] Matthias Schlachter, Renata Raidou, Ludvig P Muren, Bernhard Preim, Paul Martin Putora, and Katja Bühler. State-of-the-art report: Visual computing in radiation therapy planning. In *Computer Graphics Forum*, volume 38, pages 753–779. Wiley Online Library, 2019.
- [SZ18] Li Sun and Songtao Zhang. Deformable MRI-ultrasound registration using 3D convolutional neural network. In *Simulation, Image Processing, and Ultrasound Systems for Assisted Diagnosis and Navigation*, pages 152–158. Springer, 2018.
- [TA13] Vahid Tavakoli and Amir A Amini. A survey of shaped-based registration and segmentation techniques for cardiac images. *Computer Vision and Image Understanding*, 117(9):966–989, 2013.
- [TGK⁺17] Christian Tominski, Stefan Gladisch, Ulrike Kister, Raimund Dachsel, and Heidrun Schumann. Interactive lenses for visualization: An extended survey. In *Computer Graphics Forum*, volume 36, pages 173–200. Wiley Online Library, 2017.
- [Thr18] Michael Christoph Thrun. Approaches to cluster analysis. In *Projection-Based Clustering through Self-Organization and Swarm Intelligence*, pages 21–31. Springer, 2018.
- [Thr19] Three.js. WebGL2 materials texture2darray. GitHub: https://github.com/mrdoob/three.js/blob/dev/examples/webgl2_materials_texture2darray.html (accessed: 13-04-2022)., 2019.
- [TS96] Constantino Tsallis and Daniel A Stariolo. Generalized simulated annealing. *Physica A: Statistical Mechanics and its Applications*, 233(1-2):395–406, 1996.
- [Tsa88] Constantino Tsallis. Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, 52(1):479–487, 1988.
- [TSK05] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. Cluster analysis: Basic concepts and algorithms. *Introduction to Data Mining*, pages 487–568, 2005.

- [vdWSN⁺14] Stéfan van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. Scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014.
- [VGO⁺20] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17:261–272, 2020.
- [VLM⁺17] Rémy Vandaele, François Lallemand, Philippe Martinive, Akos Gulyban, Sébastien Jodogne, Philippe Coucke, Pierre Geurts, and Raphaël Marée. Automated multimodal volume registration based on supervised 3D anatomical landmark detection. *SciTePress Digital Library*, 2017.
- [VWI97] Paul Viola and William M Wells III. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.
- [Wal11] Jordan Walke. React.js. <https://reactjs.org> (accessed: 24-03-2022), 2011.
- [WDO⁺05] He Wang, Lei Dong, Jennifer O’Daniel, Radhe Mohan, Adam S Garden, K Kian Ang, Deborah A Kuban, Mark Bonnen, Joe Y Chang, and Rex Cheung. Validation of an accelerated ‘Demons’ algorithm for deformable image registration in radiation therapy. *Physics in Medicine & Biology*, 50(12):2887, 2005.
- [WGH⁺98] Roger P Woods, Scott T Grafton, Colin J Holmes, Simon R Cherry, and John C Mazziotta. Automated image registration: I. general methods and intrasubject, intramodality validation. *Journal of Computer Assisted Tomography*, 22(1):139–152, 1998.
- [XG00] Yang Xiang and Xingao Gong. Efficiency of generalized simulated annealing. *Physical Review E*, 62(3):4473, 2000.
- [XN19] Zhenlin Xu and Marc Niethammer. DeepAtlas: Joint semi-supervised learning of image registration and segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 420–429. Springer, 2019.
- [XSFG97] Yang Xiang, DY Sun, W Fan, and Xingao Gong. Generalized simulated annealing algorithm and its application to the Thomson model. *Physics Letters A*, 233(3):216–220, 1997.

- [Yan17] Xiao Yang. *Uncertainty quantification, image synthesis and deformation prediction for image registration*. PhD thesis, The University of North Carolina at Chapel Hill, 2017.
- [YHDL⁺18] Sigmund Ytre-Hauge, Julie A Dybvik, Arvid Lundervold, Øyvind O Salvesen, Camilla Krakstad, Kristine E Fasmer, Henrica M Werner, Balaji Ganeshan, Erling Høivik, Line Bjørge, Jone Trovik, and Ingrid S Haldorsen. Preoperative tumor texture analysis on MRI predicts high-risk disease and reduced survival in endometrial cancer. *Journal of Magnetic Resonance Imaging*, 48(6):1637–1647, 2018.
- [YJY⁺17] Lina Yu, Hengle Jiang, Hongfeng Yu, Chi Zhang, Josiah Mcallister, and Dandan Zheng. iVAR: Interactive visual analytics of radiomics features from large-scale medical images. In *2017 IEEE International Conference on Big Data*, pages 3916–3923. IEEE, 2017.
- [You19] Evan You. Vite: Next generation frontend tooling. <https://vitejs.dev>, GitHub: <https://github.com/vitejs/vite> (accessed: 19-04-2022). Version 1.0.0 (React Plugin)., 2019.
- [YWW⁺20] Wenguang Yuan, Jia Wei, Jiabing Wang, Qianli Ma, and Tolga Tasdizen. Unified generative adversarial networks for multimodal segmentation from unpaired 3D medical images. *Medical Image Analysis*, 64:101731, 2020.
- [YZN⁺09] Deshan Yang, Jie Zheng, Ahmad Nofal, Joseph Deasy, and Issam M El Naqa. Techniques and software tool for 3D multimodality medical image segmentation. *Journal of Radiation Oncology Informatics*, 1(1):1–22, 2009.
- [Zab14] Matt Zabriskie. Axios: A promise-based HTTP client. <https://axios-http.com>, GitHub: <https://github.com/axios/axios> (accessed: 19-04-2022). Version 0.21.4., 2014.
- [ZFC⁺17] Azar Zandifar, Vladimir Fonov, Pierrick Coupé, Jens Pruessner, D Louis Collins, and Alzheimer’s Disease Neuroimaging Initiative. A comparison of accurate automatic hippocampal segmentation methods. *NeuroImage*, 155:383–393, 2017.
- [ZRC19] Tongxue Zhou, Su Ruan, and Stéphane Canu. A review: Deep learning for medical image segmentation using multi-modality fusion. *Array*, 3:100004, 2019.
- [ZRC⁺20] Muhammad Asyaari Zakaria, Nor Fadilah Rajab, Eng Wee Chua, Gayathri Thevi Selvarajah, and Siti Fathiah Masre. The roles of tissue rigidity and its underlying mechanisms in promoting tumor growth. *Cancer Investigation*, 38(8-9):445–462, 2020.