

Master's Thesis

# Image Processing Methods for Analysing Glioblastoma Multiforme

in

## MRI Data

*submitted by*

**Sabine Müller**

on February 20, 2014

*Supervisor*

Prof. Dr. rer. nat. Joachim Weickert

*Reviewers*

Prof. Dr. rer. nat. Joachim Weickert

Dr.-Ing. Christian Schmaltz

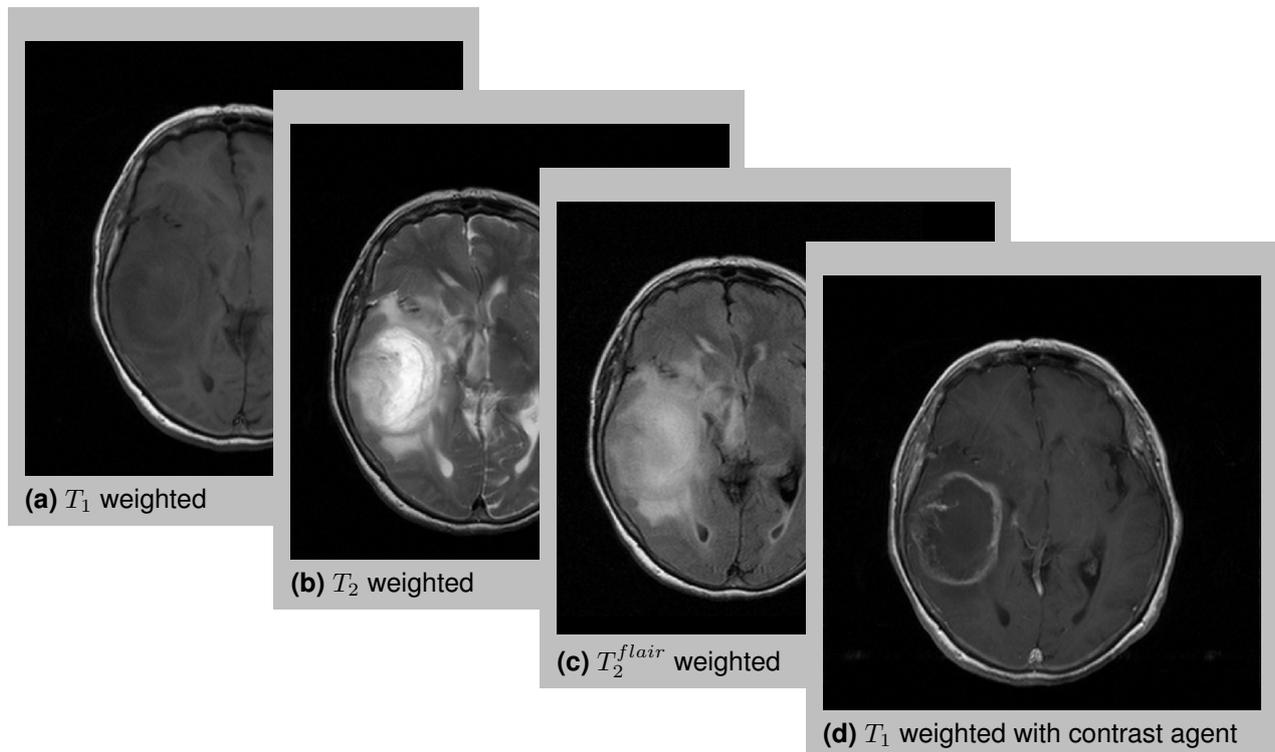
Germany, 66123 Saarbrücken



# Abstract

The aim of this thesis is to give a collection of image processing methods that allow for a reproducible and accurate extraction of tumour tissue in MRI data. We have in total four different types of MRI sequences. As shown in **Figure 1**, each type of MRI scan displays specific information. We combine them in the segmentation procedure to gain as most reliable data as possible.

Due to the acquisition process, the images are noisy and the different MRI scans are not registered. Thus, we denoise and align the MRI data before we combine their information. This is necessary in order to segment various parts of tumour tissue.



**Figure 1:** Different types of MRI scans.



---

## Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

## Statement in Lieu of an Oath

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis.

## Einverständniserklärung

Ich bin damit einverstanden, dass meine (bestandene) Arbeit in beiden Versionen in die Bibliothek der Informatik aufgenommen und damit veröffentlicht wird. Diese gedruckte Version und die am Ende der Arbeit angefügte, elektronische Version sind identisch.

## Declaration of Consent

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department. This printed version and the electronic version attached at the end are identical.

Saarbrücken, .....  
(Datum / Date)

.....  
(Unterschrift / Signature)

---



---

# Acknowledgments

First of all, I want to thank Prof. Dr. Joachim Weickert. He opened my mind for the field of visual computing, provided me with a really interesting topic for my thesis and was always helpful and friendly.

I am also very thankful to Prof. Dr. Norbert Graf for his nice cooperation and being kind and helpful all the time.

Of course, I also thank Dr. Christian Schmaltz for being so kind and accepting my request to be my second reviewer.

Furthermore, I have to thank Leif Bergerhoff, Hosna Sattar and Julian Steil for inspiring conversations and reviewing my thesis.

Moreover, I wish to thank Dr. Markus Mainberger for his advice and helpful conversations.

Last but not least, I am very grateful for the endless support my family and friends gave me.

---



---

# Introduction

## Motivation

---

Brain tumours account only for a very small proportion of all types of cancer and are among the rare diseases. However, they are also among the most fatal forms of cancer. There are many different types of brain tumours while most of them develop from the glial cells that support the nerve cells of the brain - such tumours are called gliomas. Based on the World Health Organization classification of central nervous system tumours, they are put into groups according to how quickly they probably grow [30]. There are 4 groups, called grades I to IV, whereby the more quickly the brain tumour is likely to develop, the higher the grade. Hence, grade IV gliomas are the fastest growing brain tumours at all. When in addition, a tumour is likely to come back after surgery and may spread to other parts of the brain, it is called *malignant*. Depending on whether the cancer starts in the brain or spreads from another cancer somewhere else in the body, it is called a *primary* or secondary brain tumour. Grade IV gliomas are also called *Glioblastoma multiforme* and are the most common malignant primary brain tumors.

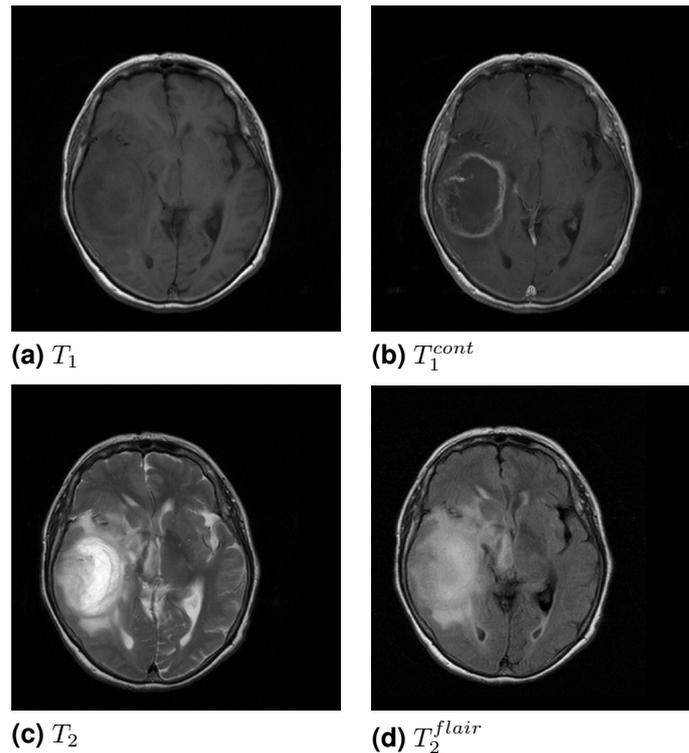
The initial treatment of high-grade glioma usually involves surgery to remove as much of the tumour cells as possible followed by a radiochemotherapy. Unfortunately, high-grade brain tumours always have microscopic tumour cells that grow beyond the border of the tumour. Hence, the tumour eventually regrows and the prognosis for patients diagnosed with the highly aggressive glioblastoma multiforme remains bleak: Current treatment provides an median overall survival up to 15 months [33].

The radiotherapy causes in addition a change in the permeability of the *blood brain barrier* [60]. Therefore, it is hard to determine whether there is a real or a pseudo-progression of the tumour. Certainty is only given by a biopsy, a neurosurgery by which tissue is taken from the affected area.

In order to spare high-grade glioma patients this additional surgery we will present image processing methods for analysing glioblastomae multiforme in Magnetic Resonance Imaging (**MRI**) data. It can be possible that the processed **MRI** data can be used to extract image features which contain enough information such that a classification algorithm can discriminate between pseudo and real progression.

---

Within this thesis, we will use the  $T_1$  weighted scan as the reference cf. **Figure 2a**. *Necrosis* is a cell injury that results in the premature death of cells in living tissue [56]. The central area of necrosis is best visible in a  $T_1$  weighted scan with contrast agent. This is demonstrated by **Figure 2b**. **Figure 2c** and **Figure 2d** depict  $T_2$  weighted scans, where fat is differentiated from water. In the case of  $T_2^{flair}$  the signal from free water, i.e. cerebrospinal fluid, is suppressed such that the oedematous tissue is observable.



**Figure 2:** Different sequences of subject *G13*, Slice 7.

## Contents

In the beginning, we give a short introduction to **MRI** as well as some mathematical background and basic knowledge of image processing as far as it is necessary to explain the used procedures. Afterwards, we show in the second chapter how the first step in the process chain, the denoising, can be done and explain in the third chapter in which way the different scans can be registered. In the fourth chapter we illustrate the process of segmenting the brain tumour. Due to the extensive process chain, several parameters have to be selected. How this is done is demonstrated in the fifth chapter, followed by an evaluation of our procedure in chapter six.

In the end, we conclude this thesis with a summary and some proposals for extensions of the given process chain.

# Contents

<b>1</b>	<b>Background Information</b>	<b>1</b>
1.1	Basics of Magnetic Resonance Imaging	1
1.1.1	$T_1$ weighted MRI Scan	2
1.1.2	$T_1$ weighted MRI Scan with Contrast Agent	4
1.1.3	$T_2$ weighted MRI Scan	4
1.1.4	$T_2^{flair}$ weighted MRI Scan	5
1.2	Fundamentals of Image Processing	6
1.2.1	Digital Images	6
1.2.2	Fourier Transform	7
1.2.2.1	Continuous Fourier Transform	7
1.2.2.2	Discrete Fourier Transform	9
1.2.2.3	Fast Fourier Transform	10
1.2.3	Lowpass Filtering	11
1.2.4	Highpass Filtering	13
1.2.5	Partial Derivatives	13
1.2.6	Structure Tensor	16
1.2.6.1	Eigenvalues and Eigenvectors	16
1.2.7	Diffusion Reaction Systems	17
1.2.7.1	Linear Diffusion	18
1.2.8	Variational Methods	19
1.2.9	Active Contour Models	20
1.2.9.1	Edge-based Active Contours	20
1.2.9.2	Region-based Active Contours	22
1.2.10	Thomas Algorithm	23
1.3	Visualisation	25
1.3.1	Pseudo Colouring	25
1.3.2	Colour Plot of the Gradient Domain	25
<b>2</b>	<b>Denoising</b>	<b>27</b>
2.1	Noise Model	28
2.2	Quality Measure	28
2.3	Neighbourhood Filters	29
2.3.1	Bilateral Filtering	30
2.3.2	Non-Local Means	32

<b>3</b>	<b>Registration</b>	<b>37</b>
3.1	Registration of Grid Dimensions	39
3.1.1	Classical Interpolation	39
3.1.2	B-Spline Interpolation	41
3.2	Coherence Enhancing Diffusion Filtering	44
3.3	Nonflat Morphological Filtering	46
3.3.1	Dilation	48
3.3.2	Erosion	50
3.3.3	Opening	51
3.3.4	White Top Hat	53
3.4	Chan-Vese Active Contour Model	55
3.5	Connected Component Labelling	61
3.6	Affine Body Transform	63
3.6.1	Principal Axes Transform	63
<b>4</b>	<b>Tumour Segmentation</b>	<b>67</b>
4.1	Chan-Vese Active Contour Model for Vector Valued Images	68
4.2	Chan-Vese Active Contour Model with a Multiphase Level Set Representation	70
4.3	Chan-Vese Active Contour Model for Vector Valued Images with a Multiphase Level Set Representation	72
<b>5</b>	<b>Parameter Selection</b>	<b>79</b>
5.1	Non-Local Means	79
5.2	Coherence Enhancing Diffusion	83
5.3	White Top Hat	88
5.4	Bilateral Filtering	90
5.5	Chan-Vese Active Contour Model	93
5.6	Chan-Vese Active Contour Model for Vector Valued Images with a Multiphase Level Set Representation	100
<b>6</b>	<b>Experimental Results</b>	<b>105</b>
<b>7</b>	<b>Conclusion and Outlook</b>	<b>119</b>
7.1	Conclusion	119
7.2	Outlook	120

---

---

<b>A</b>	<b>Figures</b>	<b>121</b>
<b>B</b>	<b>Acronyms</b>	<b>125</b>
<b>C</b>	<b>Index</b>	<b>127</b>
<b>D</b>	<b>References</b>	<b>129</b>

---

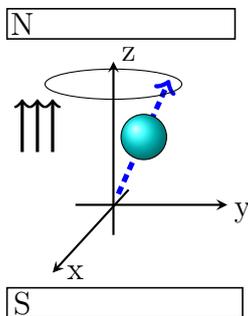


# 1 Background Information

In this chapter we give a basic introduction of *MRI* as well as some fundamental knowledge of mathematical and image processing essentials, which are necessary to develop the procedures we use to process the *MRI* scans.

## 1.1 Basics of Magnetic Resonance Imaging

*MRI* is also called magnetic resonance tomography (MRT) and was mainly developed by Paul C. Lauterbur [3, 28]. *MRI* is a medical image acquisition technique which is based on the different properties of human tissue in strong magnetic fields. It is used in radiology to visualise noninvasive internal structures of the body. In contrast to other imaging methods like X-rays, *MRI* does not use ionising radiation. Soft tissues like the brain, muscles, connective tissue and most tumours contain many hydrogen atoms. *MRI* is therefore well suited for imaging those, as it is explained in the following.



**Figure 1.1:** Proton aligned along a magnetic field

the hydrogen nuclei spins around the axis of the magnetic field. If a radio frequency impulse at their *Larmor frequency*<sup>1</sup> is applied, the protons are stimulated to perform a gyroscopic procession and are forced to spin synchronously, or *in phase*.

The physical principle of this technique is based on the *Nuclear Magnetic Resonance*, that was first described and measured by Isidor Rabi [44]. It describes a phenomenon in which nuclei with an odd number of nucleons absorb and re-emit electromagnetic radiation in magnetic fields. *MRI* scanners exploit the fact that hydrogen is present in water, fat and most other tissues in the human body. A hydrogen nuclei, which has exactly one nucleon which consists only of a proton, behaves in strong magnetic fields like a small magnetic dipole aligning itself either parallel or antiparallel along the field. While aligned, the proton of

<sup>1</sup> *Larmor frequency*:  $\omega = \gamma * B$ ,  $\omega$  denotes the Larmor frequency,  $\gamma$  the gyromagnetic ratio which is a nuclei specific constant. I.e. for hydrogen  $\gamma \approx 42.6 \frac{Mhz}{Tesla}$ ,  $B$  is the magnitude of the strength of the magnetic field.

After this excitation the protons slowly return to their thermodynamic equilibrium and realign again with the magnetic field. During this relaxation, the protons release the received energy, called *spin echo*, which is proportional to the number of excited protons. This signal is measured with receiver coils and transformed to an image. The spin echo itself can be divided into a *transversal* or *spin-spin*<sup>2</sup> and a *longitudinal* or *spin-lattice*<sup>3</sup> relaxation.

The time required for a 63% decay transversal relaxation is called  $T_2$  the analogous for the longitudinal relaxation  $T_1$ . Thereby the *spin-spin relaxation*, which decreases exponentially, accords to the dephasing of the x and y components, whereby the *spin-lattice relaxation* describes the realignment to the axis of the magnetic field [3, 50].

Protons of hydrogen return in different tissues to their equilibrium at different relaxation rates. Several tissue variables, including  $T_1$  and  $T_2$  relaxation times and also the injection of a contrast agent can be used to construct images. Consequently, depending on the information one is interested in, the type of MRI scan should be chosen accordingly.

The main parameters to describe such a sequence of various radio frequency pulses and the selected measurement of a signal are given by the echo time  $T_E$  and the repetition time  $T_R$ .  $T_E$  denotes the time between the pulse and the measurement and  $T_R$  expresses the time between two stimulation cycles. Thus, we will in the following illustrate the properties of different MRI sequences and how they can be constructed.

### 1.1.1 $T_1$ weighted MRI Scan

As mentioned before,  $T_1$  denotes the half-value time of the longitudinal relaxation of the stimulated photons. Thus, the  $T_1$  contrast of the resulting image is heavily influenced by the repetition time  $T_R$  which describes the time given the protons for recovering between two stimulation cycles. After the first stimulation cycle the spins are deflected and the relaxation begins. In contrast to materials with long  $T_1$ , tissues with short  $T_1$  are already completely returned to their equilibrium, when the second stimulation is induced with short  $T_R$ , i.e.  $T_R < 600 \text{ msec}$ . Hence, these tissues will emit a higher signal after the second excitement and will appear brighter in the resulting image [69].

---

<sup>2</sup> For water-based tissues  $T_2$  is in the 40-200ms range, for fat-based it is in the 10-100ms range.

<sup>3</sup> For water-based tissues  $T_1$  is in the 0.4-1.2s range, for fat-based it is in the 0.1-0.15s range.

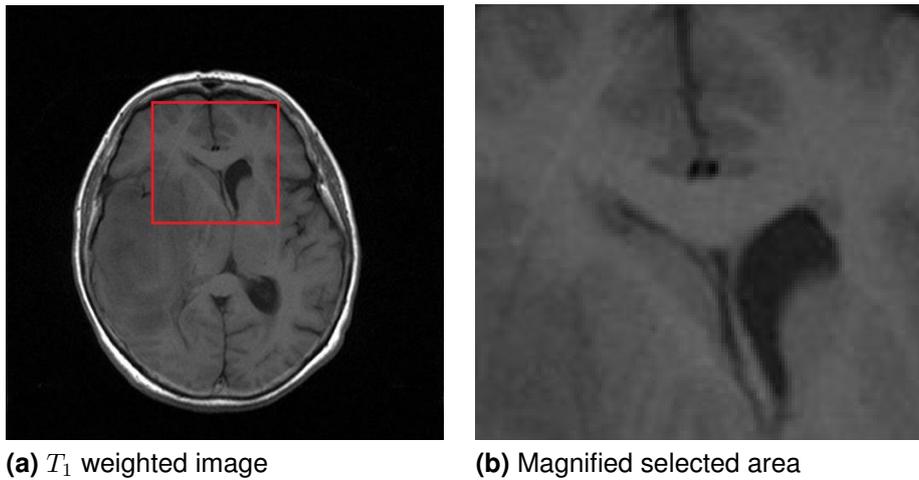
---

Tissue	$T_1$ (relaxation time)	Brightness ( $T_1$ weighted)
Fat	260	white
White Matter	780	light grey
Grey Matter	920	grey
Cerebrospinal Fluid	> 4000	dark grey

**Figure 1.2:**  $T_1$  relaxation rates in *msec* for different tissues ( $1, 5T$ ) [69]

Some important  $T_1$  relaxation rates are given in **Figure 1.2**. These relaxation times are reflected in **Figure 1.3b**. *White matter* has an insulator wrapped around it, called *myelin layer* or *myelin sheat*. The main purpose of the myelin layer is to increase the speed at which impulses propagate along the myelinated fiber [4]. However, as the myelin sheat is made of fat, the white matter appears light grey.

*Grey matter*, which does not have such an insulator and thus a higher  $T_1$  relaxation rate as white matter, turns up darker. Regarding the *cerebrospinal fluid*, we gain nearly no signal. Its  $T_1$  relaxation time is very long, thus the longitudinal relaxation of the protons of its hydrogen has not progressed far. Due to these contrasts between different tissues,  $T_1$  weighted images are well suited to define the anatomy.



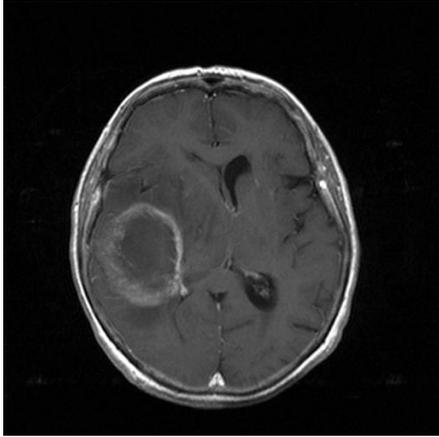
**Figure 1.3:** Different tissues in a  $T_1$  weighted Image.

Hence we use  $T_1$  weighted **MRI** scans in the registration step as reference images, see **Chapter 3**.

The brain can be divided into a left and right half, i.e. *hemispheres*. Normally, they are almost symmetric. Due to the visibility of the brain structure, we can see in **Figure 1.3a** that the left hemisphere is deformed. We will illustrate in the next section how the tumour causing this deformation, can be made visible.

### 1.1.2 $T_1$ weighted MRI Scan with Contrast Agent

*MRI contrast agents*, shortening the  $T_1$  relaxation time of protons located nearby, are used to improve the visibility of internal structures in *MRI*.



**Figure 1.4:**  $T_1$  weighted Image with Contrast Agent

The most frequently used ones are gadolinium-based, which are very hydrophilic compounds and thus not able to pass an intact *blood brain barrier*, separating the circulating blood from the central nervous system [56]. Hence, only due to defects in this barrier the gadolinium-compound can be transported into the *stroma*, i.e. the supportive framework of the tissue [56].

Glioblastoma multiforme is a highly aggressive malignant primary brain tumour which degrades the blood brain barrier. Thus, the contrast agent gets in the stroma and makes the damaged areas of the barrier visible.

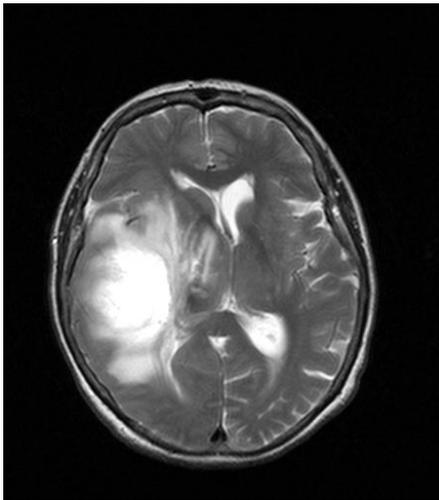
These blood vessels are responsible for the

### 1.1.3 $T_2$ weighted MRI Scan

The half-value time of the exponential decay of the spin-spin relaxation is denoted by  $T_2$  [50], which is much shorter than  $T_1$ . This can be seen in **Figure 1.2** and **Figure 1.5**. The echo time determines the influence of  $T_2$  to the image contrast. If the echo time is short, i.e.  $T_E < 30msec$  [69], the signal differences between the various tissues are small. The  $T_2$  relaxation has just begun and the signals are not significantly subsided. Hence the  $T_2$  contrast is low. However, for long echo times in the range of the occurring  $T_2$  relaxation times, i.e.  $T_E > 60msec$  [69], the contrast between different tissues is high. The faster the dephasing proceeds, the lower is the remaining signal intensity. Tissues with shorter  $T_2$  relaxation rate are

Tissue	$T_2$ (relaxation time)	Brightness ( $T_2$ weighted)
Fat	80	black
White Matter	90	grey
Grey Matter	100	light grey
Cerebrospinal Fluid	> 2000	white

**Figure 1.5:**  $T_2$  relaxation rates in *msec* for different tissues ( $1, 5T$ ) [69]



**Figure 1.6:**  $T_2$  weighted Image

therefore darker on the resulting image. On the other hand, tissues with a slow transversal relaxation still emit a high signal and occur consequently brighter in the image [50, 69].

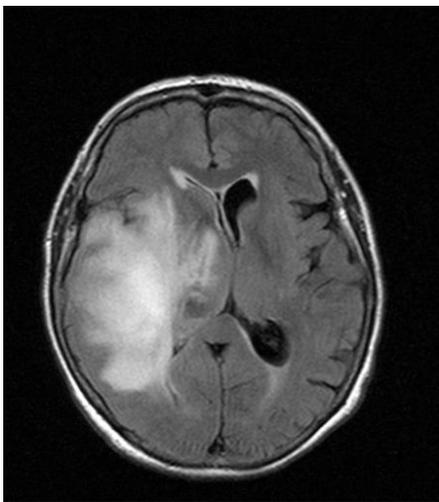
The different signal intensities are reflected in **Figure 1.6**. The cerebrospinal fluid has a very slow spin-spin relaxation and therefore retained its signal intensity. Hence it appears bright in the image, in contrast to white matter which has a fast transversal relaxation.

Although it is possible to identify the vasogenic oedema and the central area of necrosis in  $T_2$ -weighted images [18], the cerebrospinal fluid still appears bright.

How the measurement can be adjusted to a  $T_2^{flair}$  sequence in which the signal of free water is suppressed, is explained in the following.

#### 1.1.4 $T_2^{flair}$ weighted MRI Scan

In a  $T_2$  sequence, tissues with a slow spin-spin relaxation appear brighter. Thus it is difficult to distinguish between free water and oedematous areas. With an additional radio frequency pulse and a manipulation of the magnetic gradients of the MRI scanner, this type of scanning can be converted to a  $T_2^{flair}$  sequence where the signal from the cerebrospinal fluid is suppressed [69].



**Figure 1.7:**  $T_2^{flair}$  weighted Image

Hence the oedematous area is clearly visible and the  $T_2^{flair}$  is well suited to detect the vasogenic area surrounding the central area of necrosis and *lesions*, that are abnormal areas of tissue in the brain [18, 56].

## 1.2 Fundamentals of Image Processing

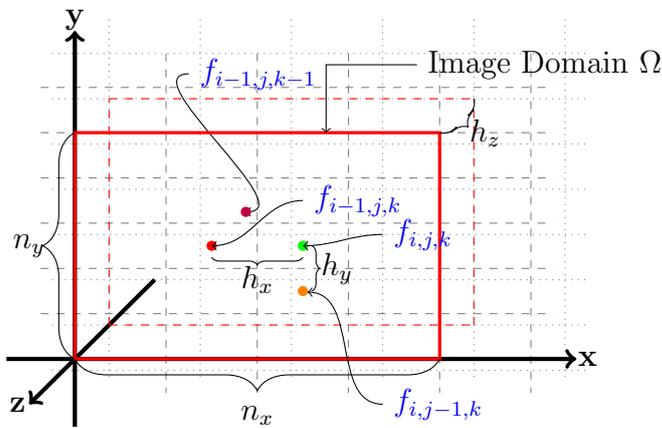
This section gives an overview of basic definitions and concepts that we will use in this thesis.

### 1.2.1 Digital Images

We model images and consequently also image stacks as functions. Thus, we denote an given image by

$$f(\mathbf{x}) : \Omega \rightarrow \mathbb{R} \quad (1.1)$$

where  $\mathbf{x} := (x, y)^\top$  and the rectangular *image domain*  $\Omega \in \mathbb{R}^2$  for 2D images and  $\mathbf{x} := (x, y, z)^\top$ ,  $\Omega \in \mathbb{R}^3$  is a cubic area for 3D images respectively.



**Figure 1.8:** Image dimensions

In the following,  $\mathbf{x} \in \Omega$  describes the position within the image domain. The co-domain of  $f$  denotes the image intensities, hence  $f(\mathbf{x})$  is a mapping from locations in the image domain to brightness values.

In order to get a digital image, the continuous image is sampled<sup>4</sup> to a discrete grid which means a reduction from the continuous to the *discrete image domain*:

$$f(\mathbf{x}) : \Gamma \rightarrow \mathbb{N} \quad (1.2)$$

where  $\Gamma := \{1, \dots, n_x\} \times \{1, \dots, n_y\}$  for 2D images and  $\Gamma := \{1, \dots, n_x\} \times \{1, \dots, n_y\} \times \{1, \dots, n_z\}$  for 3D images respectively. Thereby  $n_x$ ,  $n_y$  and  $n_z$  are the image dimensions in  $x, y$  and  $z$ -direction. The grid sizes  $h_x$  and  $h_y$  represent the grid spacing in  $x$ - and  $y$ -direction, respectively. Additionally for 3D images,  $h_z$  denotes the grid spacing in  $z$ -direction.

We obtain for 2D images the discretisation

$$f(x_i, y_j) \approx f_{i,j} \quad (1.3)$$

where  $x_i := (i - \frac{1}{2})h_x$ ,  $i \in \{1, \dots, n_x\}$  and  $y_j := (j - \frac{1}{2})h_y$ ,  $j \in \{1, \dots, n_y\}$ .

<sup>4</sup> The *Nyquist-Shannon sampling theorem* is fundamental in digital signal processing. It states, that aliasing starts at a sampling frequency lower than the *Nyquist-frequency* [21, 40, 51].

Analogously, we get for 3D images the discretisation

$$f(x_i, y_j, z_k) \approx f_{i,j,k} \quad (1.4)$$

where  $x_i := (i - \frac{1}{2})h_x, i \in \{1, \dots, n_x\}$ ,  $y_j := (j - \frac{1}{2})h_y, j \in \{1, \dots, n_y\}$  and  $z_k := (k - \frac{1}{2})h_z, k \in \{1, \dots, n_z\}$ .

In the remainder, we will name  $f_{i,j}$  as *pixel*, so a discrete location  $(i, j)^\top$  in a 2D image  $f$  and  $f_{i,j,k}$  as *voxel*, so a discrete position  $(i, j, k)^\top$  within a 3D image  $f$ . We will also denote a single image as 2D image and an image stack as 3D image.

## 1.2.2 Fourier Transform

The mathematician Fourier stated in the 19th century that any periodic function can be expressed as the sum of sines and/or cosines of different frequencies multiplied by coefficients [19].

An extension of these *Fourier series* is the *Fourier transform*, which allows to express non-periodic functions with finite integrals as the integral of sines and/or cosines multiplied by a weighting function [21]. Hence, also complicated functions (with a finite area under the curve) break down in the *Fourier domain* to simple waves.

In this section we will give a short overview of the Fourier transform and its properties.

### 1.2.2.1 Continuous Fourier Transform

The *Continuous Fourier Transform (CFT)* of a continuous function  $f(x)$  and a continuous variable  $x$  is defined by

**Definition 1 (Continuous 1D Fourier Transform)**

$$\hat{f}(u) := \mathcal{F}[f](u) := \int_{-\infty}^{\infty} f(x) e^{-i2\pi ux} dx$$

where the frequency  $u \in \mathbb{R}$  is also continuous. After integration, the only variable left is the frequency, thus we call the domain of the Fourier transform the *frequency domain*. With Euler's formula  $e^{-i\phi} = \cos(\phi) - i \sin(\phi)$  it can be rewritten, to

$$\hat{f}(u) = \int_{-\infty}^{\infty} f(x) (\cos(2\pi ux) - i \sin(2\pi ux)) dx \quad (1.5)$$

Hence, if  $f(x) \in \mathbb{R}$ , its transform  $\hat{f}(u)$  is in general complex. The *inverse Fourier transform*, denoted by

**Definition 2 (Inverse Continuous 1D Fourier Transform)**

$$f(x) := \mathcal{F}^{-1}[\hat{f}](u) := \int_{-\infty}^{\infty} \hat{f}(u) e^{i2\pi ux} du$$

allows to reconstruct the original signal. Hence, **Definition 1** and **Definition 2** comprise the so-called *Fourier transform pair* [21].

The continuous 2D Fourier Transform is given by

$$\hat{f}(u, v) := \mathcal{F}[f](u, v) := \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-i2\pi(ux+vy)} dx dy \quad (1.6)$$

which we can reformulate to **Equation 1.7**. The **CFT** is therefore separable and m-dimensional transforms break down to sequences of m 1D transforms [66].

$$\hat{f}(u, v) = \int_{-\infty}^{\infty} \left( \int_{-\infty}^{\infty} f(x, y) e^{-i2\pi ux} dx \right) e^{-i2\pi vy} dy \quad (1.7)$$

Analogously to the 1D case, the inverse **CFT** for the 2D case is denoted by

$$f(x, y) := \mathcal{F}^{-1}[\hat{f}](x, y) := \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \hat{f}(u, v) e^{i2\pi(ux+vy)} du dv \quad (1.8)$$

We rewrite it again in the same way as the 2D **CFT**:

$$f(x, y) = \int_{-\infty}^{\infty} \left( \int_{-\infty}^{\infty} \hat{f}(u, v) e^{i2\pi ux} du \right) e^{i2\pi vy} dv \quad (1.9)$$

Hence, also the backtransform is separable.

We will state in the following some properties of the **CFT** which we use in the remainder, while [66] gives a more complete overview.

It is rotationally invariant, which means that if the image is rotated, its **CFT** is

**Theorem 3 (Rotation Invariance)**

Let  $rot_{\alpha}f$  denote a rotation of  $f$  around the angle  $\alpha$ . Then

$$\mathcal{F}[rot_{\alpha}f] = rot_{\alpha}\mathcal{F}[f]$$

rotated by the same angle [66], as denoted by **Theorem 3**.

The Fourier transform is also shift invariant in the sense that the Fourier spectrum is not affected by a shift in the spatial domain, although it results in a rotation of the phase angle in the Fourier domain.

**Theorem 4 (Shift Theorem)**

$$\mathcal{F}[f(x - x_0, y - y_0)](u, v) = e^{-i2\pi(ux_0 + vy_0)} \mathcal{F}[f](u, v)$$

The last property of the Fourier transform, that we use is stated by the convolution theorem:

**Theorem 5 (Convolution Theorem)**

$$\mathcal{F}[f(x) * g(x)] = \mathcal{F}[f] \mathcal{F}[g]$$

Thus, a convolution, defined as

**Definition 6 (2D Convolution)**

$$(f * g)(x, y) := \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x - x', y - y') g(x', y') dx dy$$

in two dimensions, can easily be computed as a multiplication in the Fourier domain. The **CFT** is based on a continuous signal with infinite extend. Unfortunately, an image has a finite number of samples. The **CFT** is for this reason mostly used for designing filters, while they are implemented with the discrete Fourier transform [66]. We will give in the following a short introduction to the discrete Fourier transform and how problems, caused by the finite extend of the signal, can be handled.

### 1.2.2.2 Discrete Fourier Transform

The *Discrete Fourier Transform (DFT)* is the analogue transform for discrete signals to the previous explained **CFT** [66]. The discrete 1D Fourier transform is given by

**Definition 7 (Discrete 1D Fourier Transform)**

$$\hat{f}_p := \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} f_m e^{-\frac{i2\pi pm}{M}}$$

where  $M$  is the number of samples. Its inverse transform is stated by

**Definition 8 (Inverse Discrete 1D Fourier Transform)**

$$f_m := \frac{1}{\sqrt{M}} \sum_{p=0}^{M-1} \hat{f}_p e^{\frac{i2\pi pm}{M}}$$

The **DFT** has to some extent the same properties as the **CFT**. For instance, the convolution theorem holds as well as the shift theorem, if the signal is extended periodically. In contrast, rotation invariance can only be approximated due to the discrete grid [66].

As mentioned before, the fundamental difference to the **CFT** is the finite number of samples. The complex exponential function is periodic and causes unfortunately a periodic continuation of the signal in the Fourier domain. This carries over to the spatial domain and can create artefacts at the boundaries such as wraparound errors in connection with convolutions [66].

We solve this issue by mirroring the images at the boundaries. Hence, the length of our signal is doubled in each dimension which leads to a high computational load.

### 1.2.2.3 Fast Fourier Transform

The computational effort to solve the **DFT** is quite high. In addition, the images are mirrored at the boundaries to avoid artifacts.

The *Fast Fourier Transform (FFT)* reduces this computational load by a so-called "Divide and Conquer"-strategy. Thus, it divides the problem of size  $M$  recursively into two subproblems of size  $\frac{M}{2}$  until  $M = 1$  while producing exactly the same results as the **DFT**. Hence, the **FFT** is defined the same way as the **DFT**, whereas the *inverse FFT* is slightly different.

**Definition 9 (Inverse 1D Fast Fourier Transform)**

$$f_m = \bar{f}_m := \frac{1}{\sqrt{M}} \sum_{p=0}^{M-1} \hat{f}_p e^{-\frac{i2\pi pm}{M}} \quad f_m \in \mathbb{R}$$

Instead of weighting the Fourier coefficients  $\hat{f}_p$ , their complex conjugates  $\bar{\hat{f}}_p := \text{Re}(\hat{f}) - i\text{Im}(\hat{f})$  are used<sup>5</sup>.

Each term of the Fourier transform  $\mathcal{F}(f)$  of  $f$  contains all values weighted by sine and cosine functions, cf. **Section 1.2.2**. It is therefore mostly impossible

<sup>5</sup> There are many numerical packages available containing the **FFT**. We use the NVIDIA CUDA Fast Fourier Transform library [38] within this thesis.

to see direct correspondences between an image and its transform.

Nevertheless, frequencies are related to spatial rates of changes in the intensity. Thus, they can be associated with patterns of intensity variations in the transformed image [21]. The low frequencies corresponding to low variations of the intensities are centred around the origin. Hence, the further away a frequency is from the origin, the higher is the intensity change it describes. Low frequencies depict therefore nearly homogeneous areas and background artefacts while high ones can be associated with edges and noise [66]. Filtering in the Fourier domain is based on modifying the Fourier transform of an image. Thus, given an image  $f(x, y)$  in two dimensions it can be denoted by

$$f_{\mathcal{D}}(x, y) = \mathcal{F}^{-1}(\mathcal{D}(u, v)\mathcal{F}(u, v)), \quad (1.10)$$

where  $f_{\mathcal{D}}(x, y)$  is the filtered image and  $\mathcal{F}(u, v)$  and  $\mathcal{D}(u, v)$  are the Fourier transforms of the image itself and the filter, respectively.

A filtering technique which attenuates high frequencies and does not affect low ones is called *lowpass filter*, while a filter behaving the other way around is named a *highpass filter*.

### 1.2.3 Lowpass Filtering

Lowpass filters attenuate noise and small scale details, but also act stabilising. Hence, we use this filtering technique frequently to stabilise our algorithms. The most popular lowpass filter is the *Gaussian convolution*, where a signal is convolved with a Gaussian *kernel* defined by

**Definition 10 (2D Gaussian Kernel)**

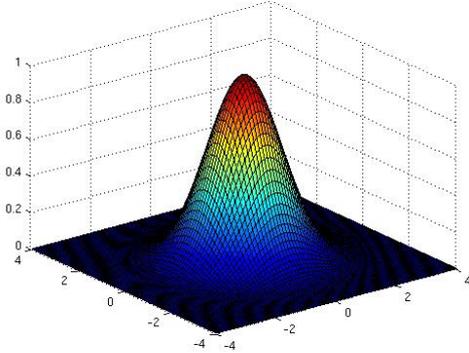
$$K_{\sigma}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} dx dy$$

in two dimensions, while it is denoted by

$$K_{\sigma}(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{(2\pi\sigma^2)^{\frac{3}{2}}} e^{-\frac{x^2+y^2+z^2}{2\sigma^2}} dx dy dz$$

in three dimensions, respectively. Thereby,  $\sigma^2$  is the *variance*, which describes the width of the kernel. *Rotational invariance* describes the property that no direction is favoured. Gaussian kernels are the only separable and *rotationally invariant* convolution kernels [66].

Due to its separability and symmetry, the computational effort reduces for an image with  $N$  pixels to  $O(N\sigma)$  if the kernel is truncated at a multiple of the



**Figure 1.9:** 2D Gaussian kernel,  $\sigma = 1$

standard deviation  $\sigma$ . It is therefore well suited for a computation in the spatial domain if  $\sigma$  is small [67]. We mentioned in **Section 1.2.2** that a convolution in the spatial domain corresponds to a multiplication in the Fourier domain.

Hence, the convolution of an image with a Gaussian kernel breaks down to multiplications in the Fourier domain. The Fourier transform itself can be computed efficiently while

the computational burden is independent of the standard deviation of the Gaussian kernel. Computing the Gaussian convolution in the Fourier domain is consequently preferable for large  $\sigma$  [67].

Over and above that, it is possible to regularise a differentiation by so-called *Gaussian derivatives*.

**Definition 11 (3D Gaussian Derivative)**

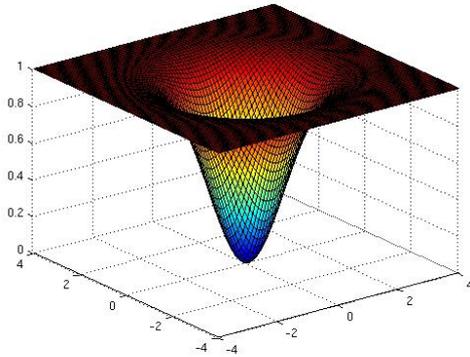
$$\partial_x^n \partial_y^m \partial_z^r (K_\sigma * f) = (\partial_x^n \partial_y^m \partial_z^r K_\sigma) * f$$

Thus, convolving with a Gaussian before differentiation is equivalent to a convolution with the derivatives of the Gaussian. Unfortunately, the Gaussian convolution has some drawbacks: It blurs also important structures such as edges and structures can become dislocated. We illustrate these disadvantages in detail in **Chapter 2** when we explain the method that we use for denoising.

---

### 1.2.4 Highpass Filtering

Highpass filters let pass high frequencies but attenuate signals with low frequencies. Hence, they are well suited to remove background distortions and to sharpen blurry structures but act also destabilising [66].



A highpass filter can for example be constructed as the difference between the identity and a low-pass filter, i.e. Gaussian convolution. This results in a filtering kernel as depicted in **Figure 1.10**.

**Figure 1.10:** 2D highpass filter,  $\sigma = 1$

As we explain in **Section 3.3.4**, we use morphological highpass filtering in the registration step to remove background perturbations.

### 1.2.5 Partial Derivatives

Derivatives are a measure how a quantity is changing in space. Thus, we often consider derivatives of the functions describing the image data to get information about structural properties like edges. We assume in the latter that the functions are sufficiently smooth to compute the derivatives.

We deal often with *Partial Differential Equations (PDE)* and the functions usually depend on at least two variables. Hence we mostly consider *partial derivatives* and denote the partial derivative of a function  $f(x, y)$  in 2D, or  $f(x, y, z)$  in the 3D case respectively, with respect to  $x$  equivalently by

$$\frac{\partial f}{\partial x} = \partial_x f = f_x \quad (1.11)$$

We handle discrete image data and have therefore to approximate these derivatives. For this we use finite difference approximations obtained by a Taylor expansion. There are several ways for a numerical approximation of the first order derivative  $\partial_x f$ . The *forward difference*, defined by

**Definition 12 (Forward Difference)**

$$f_{x,i}^+ = \frac{f_{i+1} - f_i}{h_x}$$

as well as the *backward difference*, denoted by

**Definition 13 (Backward Difference)**

$$f_{x,i}^- = \frac{f_i - f_{i-1}}{h_x}$$

have an approximation order of  $O(h)$ . As higher approximation orders give better accuracies, the *central difference*

**Definition 14 (Central Difference)**

$$f_{x,i} = \frac{f_{i+1} - f_{i-1}}{2h_x}$$

which has an approximation order of  $O(h^2)$  is more frequently used, cf. [45]. The standard finite difference approximation of the second derivative  $\partial_{xx}f$  is given by

**Definition 15 (Second Derivative)**

$$f_{xx,i} = \frac{f_{i+1} - 2f_i + f_{i-1}}{h_x^2}$$

With this, we can define the *2D gradient* of a function  $f(x, y)$  as

$$\nabla_2 f = \begin{pmatrix} f_x \\ f_y \end{pmatrix} \quad (1.12)$$

This vector points in the direction of the greatest change in intensity. Thus, it is a very simple structure descriptor and edge detector. The length of the 2D gradient, the *2D gradient magnitude* is denoted by

$$|\nabla_2 f| = \sqrt{f_x^2 + f_y^2} \quad (1.13)$$

Equivalently we define the *3D gradient* of  $f(x, y, z)$  as

$$\nabla_3 f = \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} \quad (1.14)$$

and the *3D gradient magnitude* as

$$|\nabla_3 f| = \sqrt{f_x^2 + f_y^2 + f_z^2} \quad (1.15)$$

As the gradients contain directional information they suffer from rotational variance. However, as the gradient magnitude discards the directional information, it is rotationally invariant.

---

In the latter, we also use the *divergence* expression which is defined as

$$\operatorname{div} \mathbf{j} = \nabla_2^\top \mathbf{j} = (\partial_x, \partial_y) \begin{pmatrix} j_1 \\ j_2 \end{pmatrix} = \partial_x j_1 + \partial_y j_2 \quad (1.16)$$

where  $\mathbf{j} = (j_1(x, y), j_2(x, y))^\top$  is a vector valued function in two dimensions. The 3D case is denoted by

$$\operatorname{div} \mathbf{j} = \nabla_3^\top \mathbf{j} = (\partial_x, \partial_y, \partial_z) \begin{pmatrix} j_1 \\ j_2 \\ j_3 \end{pmatrix} = \partial_x j_1 + \partial_y j_2 + \partial_z j_3 \quad (1.17)$$

where  $\mathbf{j} = (j_1(x, y, z), j_2(x, y, z), j_3(x, y, z))^\top$  is a vector valued function. Furthermore we also make use of the Laplace operator

$$\Delta u = \operatorname{div}(\nabla u) = \nabla^2 u = \sum_1^n \frac{\partial^2 u}{\partial x_n^2} \quad (1.18)$$

which describes the flux density of the gradient flow of a function.

### 1.2.6 Structure Tensor

We mentioned in the preceding that the gradient  $\nabla f$  is a very simple edge detector. Unfortunately, it is rather susceptible to noise. Convolution of  $\nabla f$  with a Gaussian kernel  $K_\sigma$ , stated by

$$\nabla f_\sigma = K_\sigma * \nabla f \quad (1.19)$$

leads to  $\nabla f_\sigma$ , which is more robust against small scale details and noise [63]. Regrettably, for larger *noise scales*  $\sigma$  adjacent gradients with the same orientation but opposite direction cancel each other.

Thus, the structure descriptor  $\nabla f_\sigma$  is not able to find parallel structures. Replacing  $\nabla f_\sigma$  by its tensor product denoted by

$$J(\nabla f_\sigma) := \nabla f_\sigma \nabla f_\sigma^\top \quad (1.20)$$

makes it invariant under sign changes. Additionally, the resulting matrix is positive semidefinite [63]. Its eigenvectors are parallel and orthogonal to  $\nabla f_\sigma$  and the corresponding eigenvalues are denoting the contrast in the eigendirections [63].

However, for large  $\sigma$  neighbouring gradients with the same direction but opposite orientations are cancelled out and it is therefore unsuited for finding interrupted lines in flow-like structures.

To overcome this problem, the orientations are averaged by applying a Gaussian convolution on some *integration scale*  $\rho$ , which results in the *structure tensor*.

#### Definition 16 (*Structure Tensor*)

$$J_\rho(\nabla f_\sigma) = K_\rho * (J(\nabla f_\sigma)).$$

This matrix is also positive semidefinite and the eigenvalues describe the average contrast in the eigendirections, so the texture size should be reflected by  $\rho$  [63, 67]. Consequently, the structure tensor is a reliable structure descriptor and can be used for different image analysis problems such as finding coherent flow-like structures [25].

#### 1.2.6.1 Eigenvalues and Eigenvectors

The structure tensor is stated by the positive semidefinite and symmetric matrix

$$J_\rho = \begin{pmatrix} j_{11} & j_{12} \\ j_{21} & j_{22} \end{pmatrix} \quad (1.21)$$

Its eigenvalues can easily be computed by

$$\mu_{1,2} = \frac{1}{2} \left( \text{trace}(J_\rho) \pm \sqrt{\text{trace}^2(J_\rho) - 4\det(J_\rho)} \right), \quad (1.22)$$

where  $\text{trace}(J_\rho) = j_{11} + j_{22}$ ,  $\det(J_\rho) = j_{11}j_{22} - j_{12}j_{21}$  with  $\mu_1 \geq \mu_2$  [64]. The corresponding eigenvectors  $w_1, w_2$  are given by  $w_1 = \begin{pmatrix} \cos \zeta \\ \sin \zeta \end{pmatrix}$ , where  $\zeta$  satisfies

$$\tan(2\zeta) = \frac{2j_{12}}{j_{11} - j_{22}} \quad j_{11} \neq j_{22}. \quad (1.23)$$

The eigenvector  $w_1$  that corresponds to the larger eigenvalue is the dominant eigenvector of  $J_\rho$  and points in the direction with the highest fluctuation.

Hence, the eigenvector  $w_2$ , corresponding to the smaller eigenvalue, describes the *coherence orientation* [63, 64, 65]. Thus, the expression

$$\kappa = (\mu_1 - \mu_2)^2 \quad (1.24)$$

measures the coherence. Isotropic structures are characterised by  $\mu_1 = \mu_2 = 0$ , straight edges leads to  $\mu_1 \gg \mu_2 = 0$  and corners give  $\mu_1 \geq \mu_2 \gg 0$ . Accordingly,  $\kappa$  tends to zero for constant areas and becomes large for very different eigenvalues [63, 64].

## 1.2.7 Diffusion Reaction Systems

Diffusion is a physical process over time which equilibrates concentration differences while preserving the mass [61].

PDE's can be classified as hyperbolic, parabolic, elliptic and ultra-hyperbolic, cf. [45]. The diffusion process is expressed by the parabolic *diffusion equation*,

### Definition 17 (*Diffusion Equation*)

$$\partial_t u = \text{div}(D\nabla u)$$

where  $D$  denotes the diffusion tensor,  $t$  the time and  $u$  the concentrations.

Diffusion in image processing acts in the same manner but  $u$  is denoting color values instead of concentrations. Depending on how the *diffusion tensor*  $D$  is adapted to local image structures, we can distinguish between several cases of diffusion filtering.

The principle of diffusion filtering is the calculation of a processed version  $u(\mathbf{x}, t)$  of  $f(\mathbf{x})$  as the solution of a diffusion equation with the bounded image  $f$  as initial condition and reflecting boundary conditions, that are also named *Neumann boundary conditions* [45].

The resulting *Initial Boundary Value Problem (IBVP)* is thus stated by

**Definition 18 (*IBVP for the Diffusion Equation*)**

$$\partial_t u = \operatorname{div}(D\nabla u)$$

with the boundary conditions

$$\begin{aligned} u(\mathbf{x}, 0) &= f(\mathbf{x}) & \mathbf{x} \in \Omega \\ \frac{\partial u}{\partial \mathbf{n}}(\mathbf{x}, t) &= 0 & \mathbf{x} \in \partial\Omega \end{aligned}$$

where  $\mathbf{n}$  denotes the outer normal and  $\partial\Omega$  the boundary.

In the following, we explain the easiest case of diffusion filtering the *linear diffusion* and its equivalence to Gaussian convolution.

### 1.2.7.1 Linear Diffusion

The diffusion tensor  $D$  in the diffusion equation can be chosen in various ways. In the most easiest case, *linear diffusion*, it is constant over the whole image domain and the image structures are not taken into account. Hence, the diffusion process is homogeneous and its **IBVP** is stated for any bounded image  $f$  by

**Definition 19 (*IBVP for the Linear Diffusion Equation*)**

$$\partial_t u = \Delta u$$

with the boundary conditions

$$\begin{aligned} u(\mathbf{x}, 0) &= f(\mathbf{x}) & \mathbf{x} \in \Omega \\ \frac{\partial u}{\partial \mathbf{n}}(\mathbf{x}, t) &= 0 & \mathbf{x} \in \partial\Omega \end{aligned}$$

This problem has the unique solution

$$u(\mathbf{x}, t) = \begin{cases} f(\mathbf{x}), & (t = 0) \\ (K_{\sqrt{2t}} * f)(\mathbf{x}), & (t > 0), \end{cases} \quad (1.25)$$

which depends continuously on the initial image  $f$  and satisfies the *maximum-minimum-principle*

$$\inf_{\mathbb{R}^m} f \leq u(\mathbf{x}, t) \leq \sup_{\mathbb{R}^m} f \quad \forall \mathbf{x}, \forall t > 0,$$

where  $m$  denotes the number of dimensions [61]. Consequently, linear diffusion and Gaussian convolution are equivalent. Furthermore, we can observe that the

time  $t$  is related to the spatial width  $\sigma$  of the Gaussian convolution kernel with  $\sigma = \sqrt{2t}$  [61]. Thus, smoothing structures of order  $\sigma$  requires to stop the diffusion process at time

$$T = \frac{1}{2}\sigma^2. \quad (1.26)$$

Due to this equivalence, linear diffusion can easily be computed for any amount of dimensions by exploiting the separability of Gaussian convolution, cf. **Section 1.2.3**.

Linear diffusion filtering is wide spread in image processing. It is useful for applications where images have to be simplified and edges are unimportant. Furthermore, Gaussian smoothing has strong regularisation properties and its scale space-evolution is well-understood [61]. Nevertheless, it has some major drawbacks. Gaussian smoothing simplifies an image and removes thereby also semantically important structures and can even dislocate them.

## 1.2.8 Variational Methods

Variational methods are an image transformation that satisfies an optimality criterion. They are based on energy functionals like

$$E(u) = \int_{\Omega} F(x_1, \dots, x_n, u, u_{x_1}, \dots, u_{x_n}) \, d\mathbf{x} \quad (1.27)$$

that rate the quality of a function w.r.t. certain assumptions. In general, a variational method is minimised and the according function is called the minimiser fitting the best to the model assumptions. The first variation of the energy is given by

**Definition 20 (First Variation of the Energy Functional)**

$$\nabla E = F_u - \partial_{x_1} F_{u_{x_1}} - \dots - \partial_{x_n} F_{u_{x_n}}$$

with the natural boundary condition

$$\mathbf{n}^\top \begin{pmatrix} F_{u_{x_1}} \\ \vdots \\ F_{u_{x_n}} \end{pmatrix} = 0$$

at the image boundary with normal vector  $\mathbf{n}$

and acts like a gradient:  $\nabla E$  points in the direction of the greatest positive change and extrema occur when the first variation vanishes i.e.  $\nabla E = 0$ . This is the well known *Euler-Lagrange equation*. Similar to the standard calculus, it holds that if an energy functional is strictly convex and satisfies (20), then it has an unique minimiser [67].

### 1.2.9 Active Contour Models

The active contour technique, also known as snakes, has grown significantly since Kass et al. proposed their work "Snakes: active contour models" [26]. The fundamental idea in active contours is to evolve a curve in direction to an object boundary subject to constraints in the underlying image.

We introduce in this section two level set methods: An edge-based segmentation method, the *geodesic active contour*, and a region-based method, the *Mumford-Shah functional* [10, 37]. We will use a simplification of the second one in our work to extract on the one hand the skull from the different MRI sequences and on the other hand to describe the boundary of the tumour tissues.

#### 1.2.9.1 Edge-based Active Contours

The classical formulation of active contour models are energy minimisation approaches. As they suffer under difficulties to handle topological changes and numerical instabilities never methods are based on level-set formulations [41] and motion by mean curvature. These approaches allow for topological changes, corners and cusps [13].

A level set formulation exploits that a curve can be seen as the zero level set of a function in higher dimension and a evolving curve  $C$  is described implicitly as

$$C = \{\mathbf{x} \mid \phi(\mathbf{x}) = 0\} \quad (1.28)$$

where  $\phi$  is a Lipschitz continuous function. The evolution of the curve is given by the zero level isophote at time  $t$  of the function  $\phi(\mathbf{x}, t)$  and evolving the curve in normal direction corresponds to the Initial Value Problem (IVP) for curve evolution, defined as

##### **Definition 21 (IVP for Curve Evolution)**

$$\begin{aligned} \phi_t &= |\nabla\phi|F \\ \phi(\mathbf{x}, 0) &= \phi_0(\mathbf{x}) \end{aligned}$$

where  $F$  is the speed.

In the case of mean curvature motion the speed is represented by

$$F = \operatorname{div} \left( \frac{\nabla\phi}{|\nabla\phi|} \right). \quad (1.29)$$

Hence, the IVP for mean curvature motion is given by

##### **Definition 22 (IVP for Mean Curvature Motion)**

$$\begin{aligned} \phi_t &= |\nabla\phi| \operatorname{div} \left( \frac{\nabla\phi}{|\nabla\phi|} \right) \\ \phi(\mathbf{x}, 0) &= \phi_0(\mathbf{x}) \end{aligned}$$

The Perona-Malik diffusivity [42] is a very popular edge detector for edge-based active contours. It is positive and vanishes for large gradients, i.e. at boundaries, defined by

**Definition 23 (Perona-Malik Diffusivity)**

$$g(|\nabla\phi_\sigma|^2) = \frac{1}{1 + \frac{|\nabla\phi_\sigma|^2}{\lambda^2}}$$

as suggested by [67]. According to **Section 1.2.6** we use the blurred gradient  $\nabla\phi_\sigma$  as it is more robust against noise. If we add this function to the equation for the mean curvature motion

**Definition 24 (IVP of Geometric Active Contour Model)**

$$\begin{aligned}\phi_t &= g(|\nabla u_\sigma|^2)|\nabla\phi|\operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) \\ \phi(\mathbf{x}, 0) &= \phi_0(\mathbf{x})\end{aligned}$$

we get a geometric active contour model whose zero level isophote moves in the normal direction [9].

*Geodesic active contours* connect classical energy minimisation based snakes with geometric active contours. Its model is represented by

$$\inf \int_0^1 |C'(s)g(|\nabla u_0 C(s)|)| ds \quad (1.30)$$

which is a problem in the Riemann space whose metric is defined by the underlying image  $u_0$  [10]. As we mentioned before, the edge-stopping function  $g$  vanishes at object boundaries. Hence, a minimising curve  $C$  is obtained when the object boundary is reached. Its level set formulation is stated by

**Definition 25 (IVP of Geodesic Active Contours)**

$$\begin{aligned}\phi_t &= |\nabla\phi|\operatorname{div}\left(g(|\nabla u_\sigma|^2)\frac{\nabla\phi}{|\nabla\phi|}\right) \\ \phi(\mathbf{x}, 0) &= \phi_0(\mathbf{x})\end{aligned}$$

The main drawback of active contour models relying on an edge-function is their dependence on the image gradient. If the object boundary is smeared or diffuse the curve may pass through it.

The region-based active contours we illustrate in the latter neither use a stopping edge-function nor do they depend on the image gradient and are therefore also able to segment objects with diffuse boundaries.

### 1.2.9.2 Region-based Active Contours

The Mumford-Shah functional is a model for energy based image segmentation and defined as

#### Definition 26 (*Mumford-Shah Functional*)

$$E_{MS}(u, C) = \underbrace{\lambda \int_{\Omega} (f(\mathbf{x}) - u(\mathbf{x}))^2 d\mathbf{x}}_1 + \underbrace{\int_{\Omega \setminus C} |\nabla u(\mathbf{x})|^2 d\mathbf{x}}_2 + \underbrace{\alpha \text{Length}(C)}_3$$

Minimising the energy functional  $E_{MS}(u, C)$  implies the minimisation of a weighted combination of the fidelity term (1) which penalises deviations from the original image, the smoothing term (2) which forces the solution to be as smooth as possible in each segment and the geometric term (3) penalising the edge length  $\text{Length}(C)$  [37].

This equation has a normally non-unique minimiser with a closed edge-set suggesting this edge-set  $C$  as the segmentation boundary.

Unfortunately, it is a mathematically very difficult free boundary problem as the edge-set is unknown. Also because of this Mumford and Shah proposed

#### Definition 27 (*Piecewise constant Mumford-Shah Functional*)

$$E_{MS_{PC}}(u, C) = \int_{\Omega} (f(\mathbf{x}) - u(\mathbf{x}))^2 d\mathbf{x} + \alpha \text{Length}(C)$$

where the smoothing term is neglected resulting in a piecewise constant formulation with a closed edge-set  $C$  [37]. This is also called the *minimal partition problem* [13].

We use in our segmentation steps a further simplification of the Mumford-Shah function, the well known *Chan-Vese active contour model (CV model)* [11, 12, 13, 14], cf. **Section 3.4** and **Section 4.3**.

### 1.2.10 Thomas Algorithm

During the registration process, i.e. for the interpolation (**Section 3.1**) we have to solve systems with tridiagonal matrices  $\mathbf{B}\mathbf{u} = \mathbf{d}$  which are characteristic for 1D problems.

For this we use a simple variant of the Gaussian elimination, the so-called *Thomas algorithm* [54], which is stable for every diagonally dominant system and highly efficient [62]. Solving a tridiagonal system by this algorithm, can be regarded as a recursive filtering in three steps [62]. First the system of equations is decomposed by an *LU decomposition* which factors a matrix as a product of an upper and a lower bidiagonal matrix, denoted by

$$\mathbf{B} = \mathbf{L}\mathbf{U} = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \gamma_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \gamma_{N-2} & \alpha_{N-1} & \beta_{N-1} \\ & & & \gamma_{N-1} & \alpha_N \end{pmatrix}$$

$$\mathbf{L} = \begin{pmatrix} 1 & & & & \\ l_1 & 1 & & & \\ & \ddots & \ddots & & \\ & & l_{N-2} & 1 & \\ & & & l_{N-1} & 1 \end{pmatrix}, \mathbf{U} = \begin{pmatrix} m_1 & r_1 & & & \\ & m_2 & r_2 & & \\ & & \ddots & \ddots & \\ & & & m_{N-1} & r_{N-1} \\ & & & & m_N \end{pmatrix}$$

$$\mathbf{LU} = \begin{pmatrix} m_1 & r_1 & & & \\ l_1 m_1 & l_1 r_1 + m_2 & r_2 & & \\ & \ddots & \ddots & \ddots & \\ & & l_{N-2} m_{N-2} & l_{N-2} r_{N-2} + m_{N-1} & r_{N-1} \\ & & & l_{N-1} m_{N-1} & l_{N-1} r_{N-1} + m_N \end{pmatrix}$$

Hence,  $r_i = \beta_i$ ,  $\forall i = 1, \dots, N-1$  and the coefficients  $m_i$  and  $l_i$  are given by

$$\begin{aligned} m_1 &:= \alpha_1 \\ \forall i = 1, \dots, N-1 : l_i &:= \frac{\gamma_i}{m_i}, m_{i+1} := \alpha_{i+1} - l_i \beta_i \end{aligned} \quad (1.31)$$

In the second step, the *forward elimination* the system  $\mathbf{L}\mathbf{y} = \mathbf{d}$  is solved for  $\mathbf{y}$  by using the results from **Equation 1.31**,

$$\begin{pmatrix} 1 & & & & \\ l_1 & 1 & & & \\ & \ddots & \ddots & & \\ & & l_{N-2} & 1 & \\ & & & l_{N-1} & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-1} \\ y_N \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_{N-1} \\ d_N \end{pmatrix}$$

with the equations

$$\begin{aligned} y_1 &:= d_1 \\ \forall i = 2, \dots, N : y_i &:= d_i - l_i y_{i-1} \end{aligned} \tag{1.32}$$

In the last step, the *backward substitution*, we solve  $\mathbf{U}\mathbf{u} = \mathbf{y}$ ,

$$\begin{pmatrix} m_1 & \beta_1 & & & \\ & m_2 & \beta_2 & & \\ & & \ddots & \ddots & \\ & & & m_{N-1} & \beta_{N-1} \\ & & & & m_N \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \\ u_N \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-1} \\ y_N \end{pmatrix}$$

by proceeding with

$$\begin{aligned} u_N &:= \frac{y_N}{m_N} \\ \forall i = N - 1, \dots, 1 : u_i &:= \frac{y_i - \beta_i u_{i+1}}{m_i} \end{aligned} \tag{1.33}$$

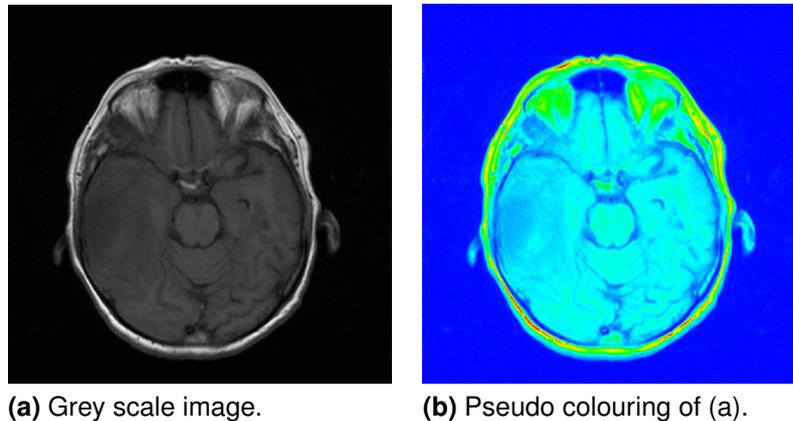
---

## 1.3 Visualisation

In the latter, we use several visualisations to highlight different aspects. Mostly we show the original grey scale images where the grey values are in the range  $[0, 4096]$ . Nevertheless, we also use a pseudo colouring of these grey value images and a colour plot of the gradient domain.

### 1.3.1 Pseudo Colouring

The human eye can distinguish only about 40 different grey scales but two million colours [66]. As the grey scale images we use consist in average of more than one thousand grey values, we use several times a pseudo colouring to make different aspects visible. The colouring blends thereby from blue to cyan to green to yellow to red, i.e. follows the visual spectrum between blue and red. An example is shown in **Figure 1.11**.

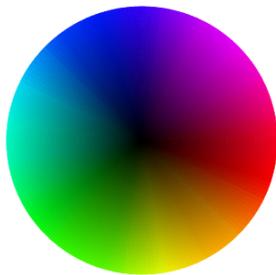


(a) Grey scale image.

(b) Pseudo colouring of (a).

**Figure 1.11:** Pseudo colouring of a grey scale image.

### 1.3.2 Colour Plot of the Gradient Domain



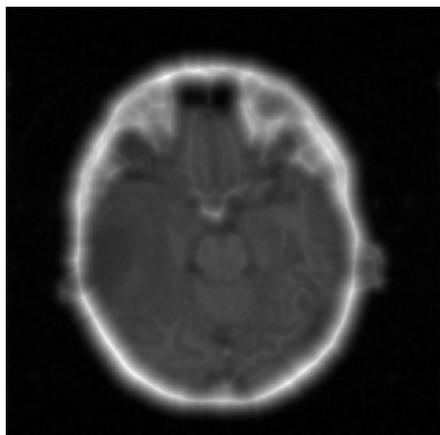
**Figure 1.12:** Colour plot for gradient magnitudes.

The visualisation of the gradient domain of an image is more complicated than the visualisation of grey scale images. On the one hand, gradients have a specific length which has to be visible, on the other hand, they also have a direction.

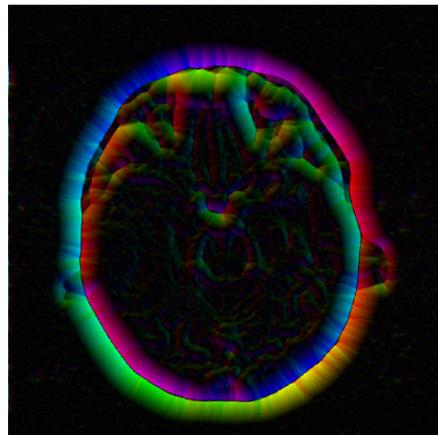
We follow [31] and visualise the gradient magnitude as brightness and the gradient direction as colour. The corresponding colour distribution is shown in **Figure 1.12**. The higher the gradient magnitude is, the brighter is the colour value. Hence, if the gradient magnitude tends to zero, its visualisation is nearly black.

An example for this visualisation is given in **Figure 1.13**. Inside the skull is a nearly homogeneous area with small gradient magnitudes, while there are long gradients at the border of the skull. Hence, the colour plot of the gradient domain is nearly black in the homogeneous area while the gradients at the border are coloured intensively.

---



(a) Grey scale image.



(b) Colour plot of the gradient domain of (a).

**Figure 1.13:** Colour plot of the gradient domain.

---

## 2 Denoising

Noise is one of the most frequent image perturbations. Within a digital image, noise is created by additional spurious information. The goal of image denoising methods is to remove the false information and to recover the original values. It is a crucial step to increase image quality and to improve the performance of all the tasks needed for quantitative imaging analysis.

**MRI** data are commonly corrupted by various types of noise including additive noise. They also contain many structured patterns and fine edges that are used to diagnose and study the anatomical structures of the human body. Unfortunately, denoising methods remove not only the noise itself but also small details, because they cannot see any difference between them. Thus, the choice of the denoising method is a tradeoff between loss of details and increase of image quality.

In the end, we want to segment the different areas of a brain tumour. Most of the segmentation methods are very sensitive to noise and low contrast regions. We have to use therefore a denoising algorithm that increases on the one hand the signal to noise ratio while small details are preserved. On the other hand, it should also enhance the contrast in the images.

There are several denoising algorithms like diffusion based filters, filters based on wavelet analysis, learning based algorithms and neighbourhood filters. Diffusion based filtering can be grouped in linear and non-linear as well as in isotropic and anisotropic filter techniques. A linear diffusion filtering, i.e. Gaussian smoothing, does not take care about the local image structure and blurs semantic important features, like edges [66]. Non-linear isotropic filters, like the well known Perona-Malik filtering [42] decrease smoothing at edges but can have problems when edges are noisy [67].

Filtering techniques based on wavelet analysis try to distinguish the image signal from the noise in the frequency domain, cf. **Section 1.2.2**. As they can create artifacts [16] and do not preserve small details, they are in our opinion not suited for denoising medical image data.

Learning based denoising algorithms, for example [17, 47], can have a good performance. However, as they need to be trained with a reasonable data set their application is restricted.

We decided to use a neighbourhood filter, the *Non-Local Means (NLMeans)* algorithm. In this chapter, we explain neighbourhood filter techniques in general and illustrate in detail the *NLMeans*.

In order to judge a denoising method, it is necessary to be able to simulate noise. Therefore, we introduce first the noise model and the used quality criterion, the *method noise* [8].

## 2.1 Noise Model

There are several types of noise, but the most important type is the *additive noise*. Thereby, the original values and the noise are assumed to be independent. Thus, it is denoted by

$$v(\mathbf{x}) = f(\mathbf{x}) + n(\mathbf{x})$$

where  $f(\mathbf{x})$  is the original value at position  $\mathbf{x}$ ,  $n(\mathbf{x})$  the noise perturbation and  $v(\mathbf{x})$  the resulting noisy value.

Noise can have different distributions but the most common one is the Gaussian distribution. In this case, the noise model is called *Gaussian noise* and the noise is independent and identically distributed.

We assume in the following, that the *MRI* sequences we consider in this thesis are mainly corrupted by Gaussian noise.

## 2.2 Quality Measure

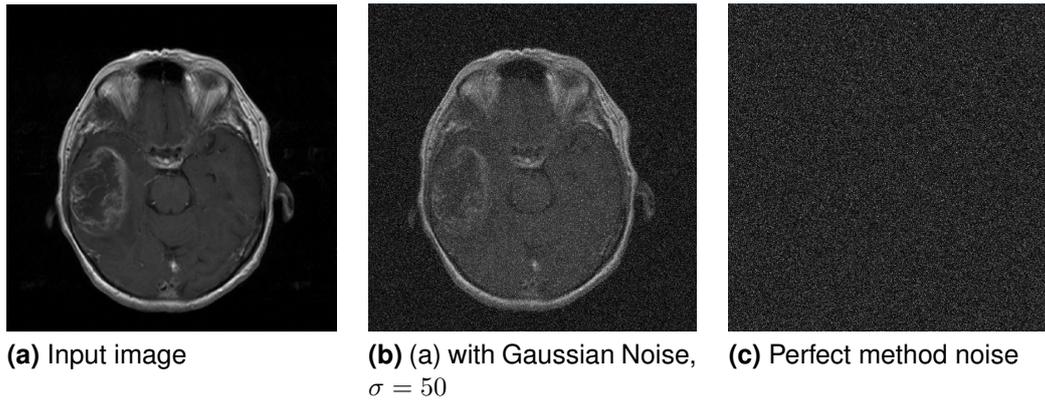
Denoising algorithms try to recover the original image  $f$  by splitting the noisy image  $v$  into a denoised image and the noise guessed by the method. Therefore, the result of a denoising method  $D$  can be described as a decomposition of any image  $v$  as

$$v(\mathbf{x}) = D_v(\mathbf{x}) + n_{D_v}(\mathbf{x}) \Leftrightarrow n_{D_v}(\mathbf{x}) = v(\mathbf{x}) - D_v(\mathbf{x}) \quad (2.1)$$

where  $v$  is the (noisy) input image,  $D_v$  the denoised image and  $n_{D_v}$  is the noise guessed by the method, the so-called method noise [8].

It is essential for denoising medical images to lose as few details as possible. Thus, this method noise should be similar to Gaussian noise and in order not to alter the original image it should contain as few structures as possible.

---

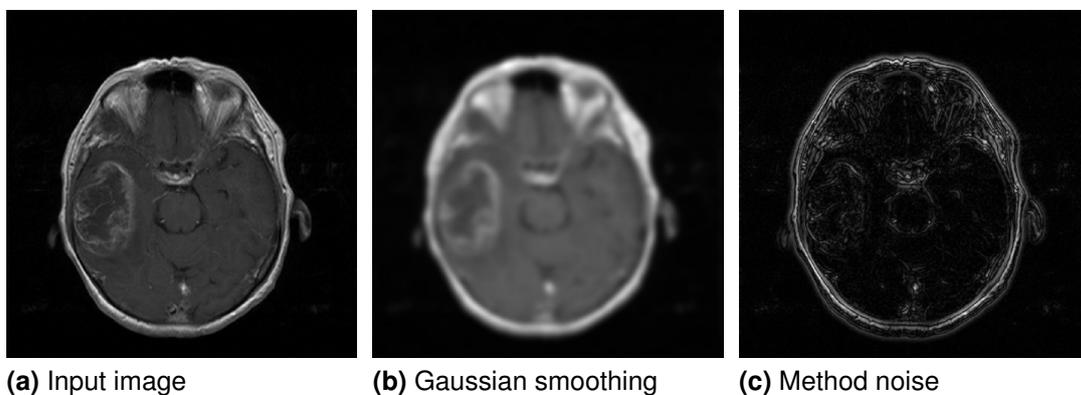


**Figure 2.1:** Perfect splitting in image and noise.

We will in the following illustrate neighbourhood filters in general and two approaches in detail. The latter one, **NLM** means filtering is used in this thesis to denoise the **MRI** sequences.

## 2.3 Neighbourhood Filters

Neighbourhood filters are a powerful method for denoising images. In order to restore the original grey value of a pixel, these filters perform an average of neighbouring pixels. Hereby, the various methods differ mainly in the choice of the considered neighbourhood.



**Figure 2.2:** Method noise of Gaussian smoothing,  $\sigma = 3$ . The method noise is gamma corrected,  $\gamma = 1.2$ .

The simplest case, Gaussian Smoothing makes no difference between edges and homogeneous regions as can be seen in **Figure 2.2**. Thus, semantically important structures are also smoothed. The method noise contains therefore a lot of structures, as we observe in **Figure 2.2c**. Based on the assumption that spatial neighbouring pixels are not necessarily similar, there are different approaches to find a neighbourhood of pixels that have a greater resemblance to the regarded pixel.

### 2.3.1 Bilateral Filtering

As mentioned before, Gaussian smoothing removes not only the noise but also the structures. The cause of this is the choice of the weights for the considered neighbourhood due to the spatial proximity. Convolution of a signal  $f \in \mathbb{R}^2$  with a Gaussian kernel, denoted as

$$(K_{\sigma_s} * f)(\mathbf{x}) := \int_{\Omega} K_{\sigma_s}(\mathbf{x} - \mathbf{y}) f(\mathbf{y}) d\mathbf{y} \quad (2.2)$$

averages each pixel  $\mathbf{x} \in \Omega$  with its spatial neighbourhood while the weights decrease with distance from  $\mathbf{x}$ . A *range filter*  $D_{\sigma_r}$  applied to a signal  $f \in \mathbb{R}^2$ , is defined as

$$D_{\sigma_r}[f](\mathbf{x}) := \int_{\Omega} K_{\sigma_r}(f(\mathbf{x}) - f(\mathbf{y})) f(\mathbf{y}) d\mathbf{y}. \quad (2.3)$$

Hence, the difference between pixel intensities is weighted by a Gaussian kernel. Bilateral filtering combines both concepts by weighting the pixels in the spatial neighbourhood with their corresponding relative pixel intensity [2, 55]. Weighting the spatial and intensity distances with Gaussian kernels is very popular, but there are several weighting functions that can be used. However, as a Gaussian kernel has some nice properties we decided to weight the distances with it. Hence, bilateral filtering can be defined as

**Definition 28 (Continuous Bilateral Filtering)**

$$BF[v](\mathbf{x}) = \frac{\int_{\Omega} K_{\sigma_s}(\mathbf{x} - \mathbf{y}) K_{\sigma_r}(f(\mathbf{x}) - f(\mathbf{y})) f(\mathbf{y}) d\mathbf{y}}{\int_{\Omega} K_{\sigma_s}(\mathbf{x} - \mathbf{y}) K_{\sigma_r}(f(\mathbf{x}) - f(\mathbf{y})) d\mathbf{y}}$$

where the Gaussians  $K_{\sigma_s}$  and  $K_{\sigma_r}$  weight the spatial and intensity distance, respectively.

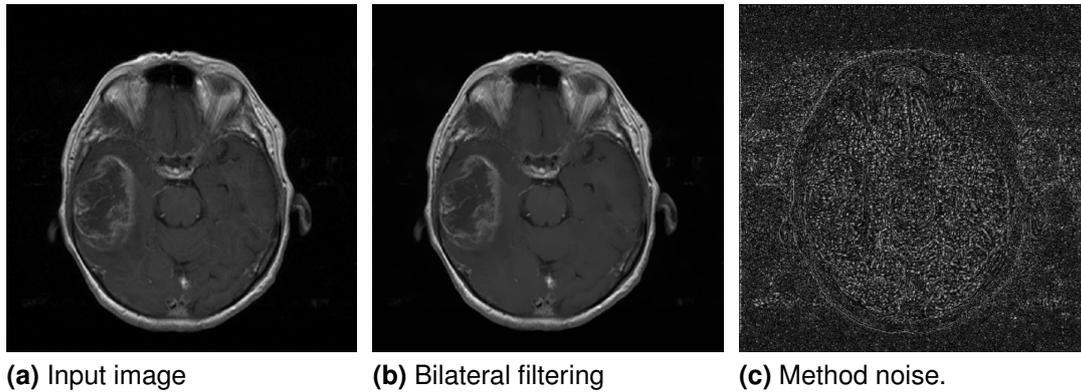
In contrast to the weighting of a Gaussian smoothing, pixels are not solely weighted because of their spatial distance. Thus, spatial neighbouring pixels that are very dissimilar in intensity from the central pixel are weighted down even though they may be located right next to it. As we are in a discrete setting, we use the discrete formulation, given by

**Definition 29 (Discrete Bilateral Filtering)**

$$BF[v]_i = \frac{\sum_{j \in \Gamma} K_{\sigma_s}(\mathbf{i} - \mathbf{j}) K_{\sigma_r}(f_i - f_j) f_j}{\sum_{j \in \Gamma} K_{\sigma_s}(\mathbf{i} - \mathbf{j}) K_{\sigma_r}(f_i - f_j)}$$

where  $m$  is the number of dimensions,  $i = (i_1, \dots, i_m)^\top \in \mathbb{N}^m$  and  $j = (j_1, \dots, j_m)^\top \in \mathbb{N}^m$  are pixel positions. The Gaussians  $K_{\sigma_s}$  and  $K_{\sigma_r}$  weight the spatial and intensity distance, respectively.

Because of this double weighting, only pixels close in spatial distance and



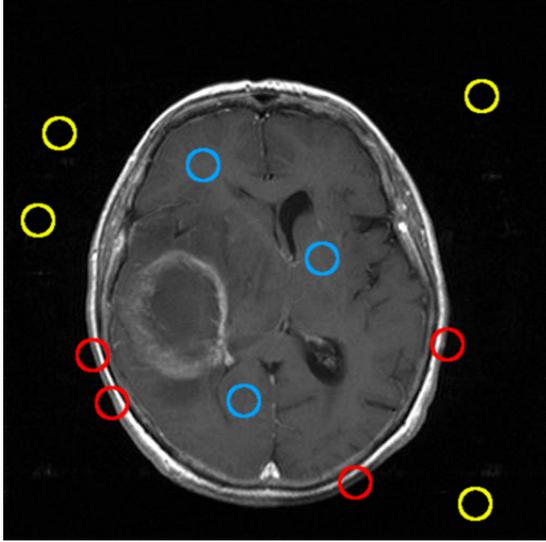
**Figure 2.3:** Method noise of Bilateral filtering,  $\sigma_s = 7, \sigma_r = 20$ . The method noise is gamma corrected,  $\gamma = 1.2$ .

intensity value are considered. In contrast to Gaussian smoothing, bilateral filtering preserves the edges. The corresponding method noise is shown in **Figure 2.3c**. It is obvious that less structures are present therein than in the Gaussian method noise (**Figure 2.2c**) and that it is relative similar to white noise. Nevertheless, the comparison of single pixel values is not robust when they are noisy. Additionally, bilateral filtering removes small structures in nearly homogeneous areas.

This is in our case a drawback and an advantage at the same time: On the one hand, we cannot use this filter to denoise the images that are used in the end to segment the tumour tissues. On the other hand, we do not need small structures when we register the images. Hence, we use this filter technique in our registration step.

### 2.3.2 Non-Local Means

The **NLMeans** algorithm was introduced by Buades et al. [7, 8]. Based on the assumption that the most similar pixels to a given pixel have no reason to be close at all, this algorithm picks up the basic ideas of bilateral filtering and extends them to a non-local filtering. As can be seen in 2.4, most image details occur repeatedly in natural images.



**Figure 2.4:** Redundant information in a natural image.

Each color denotes a group of patches in the image which can almost not be distinguished. Non-Local means try to take advantage of this high degree of redundancy. By this, this algorithm denoises a digital image while preserving edges, small structures and fine details. As mentioned before, **NLMeans** try to take advantage of the redundancy in natural images.

Thereby the main observation is that every small sample of a detail in a natural image has many similar samples in the same image. The square shaped *similarity window*  $N_{\mathbf{y}}$  or *patch* around a position  $\mathbf{y}$  is given by

$$N_{\mathbf{y}} := \{\mathbf{k} \in \Omega : |y_i - k_i| \leq d_i \quad \forall i : 1, \dots, m\} \quad (2.4)$$

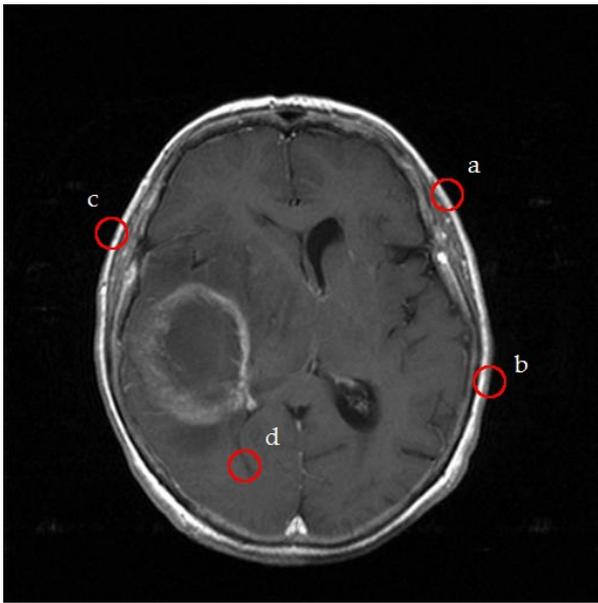
where  $d_i$  is the size of the patch in dimension  $m$ . A circular similarity window is defined as

$$N_{\mathbf{y}} := \{\mathbf{k} \in \Omega : \|\mathbf{y} - \mathbf{k}\| \leq r_N\} \quad (2.5)$$

where  $r_N$  is the radius of the circle. By considering the different patches, Buades et al. defined the neighbourhood of  $\mathbf{x} \in \Omega$  as a set of positions  $\mathbf{y} \in \Omega$  such that a window  $N_{\mathbf{y}}$  around  $\mathbf{y}$ , looks like a window  $N_{\mathbf{x}}$  around  $\mathbf{x}$  [8]. The similarity of two neighbourhoods  $N_{\mathbf{x}}$  and  $N_{\mathbf{y}}$  is determined by the Euclidean distance of their corresponding vectors

$$\|f(N_{\mathbf{x}}) - f(N_{\mathbf{y}})\| \quad (2.6)$$

Hence, the similarity between two positions  $\mathbf{x}$  and  $\mathbf{y}$  depends on the similarity of the windows around them. Thus, different patches can be regarded to be similar even though they are far away within the spatial image domain.



**Figure 2.5:** Similarity between different patches.

similarity patch and mirror at the border.

This principle is shown in **Figure 2.5**. The windows  $a$  and  $b$  are considered to be similar to the window  $c$  while the patch  $d$  is dissimilar. To better adapt this comparison to the image, the regarded windows can have different shapes and sizes, for example squares and circles like (2.4) and (2.5).

Additionally, the similarity windows have to be small enough to take care of details but also large enough to be robust against noise. As we prefer rotational invariant procedures, we use a circular shape for the

2

### Approach

For each position and adapted to the image, the **NLM** algorithm chooses a different average configuration [7]. As explained before, for a given position  $\mathbf{x} \in \Omega$ , this algorithm takes into account the similarity between the neighbourhood configuration of  $\mathbf{x}$  and all other positions in the image. In order to give similar patches a higher weight while dissimilar patches should have a lower weight, the Euclidean distance between  $N_{\mathbf{x}}$  and  $N_{\mathbf{y}}$  is used as input to a weight function, often a Gaussian. This is denoted by

#### **Definition 30 (Continuous Non-Local Means)**

$$NL[v](\mathbf{x}) := \frac{\int_{\Omega} K_{\sigma_r}(\|f(N_{\mathbf{x}}) - f(N_{\mathbf{y}})\|^2) f(\mathbf{y}) d\mathbf{y}}{\int_{\Omega} K_{\sigma_r}(\|f(N_{\mathbf{x}}) - f(N_{\mathbf{y}})\|^2) d\mathbf{y}}$$

where  $K_r$  denotes the Gaussian weight function with standard deviation  $\sigma_r$ .

Thus, the vector distance to the window around  $\mathbf{x}$  sets the weights for each position  $\mathbf{y}$ . Hence, positions with a similar neighbourhood to  $N_{\mathbf{x}}$  will have larger weights on the average. Due to the fast decay of the exponential kernel large Euclidean distances lead to nearly zero weights for dissimilar patches and acts as an automatic threshold [7]. We are in a discrete setting and use therefore the

discrete formulation given by

**Definition 31** (*Discrete Non-Local Means*)

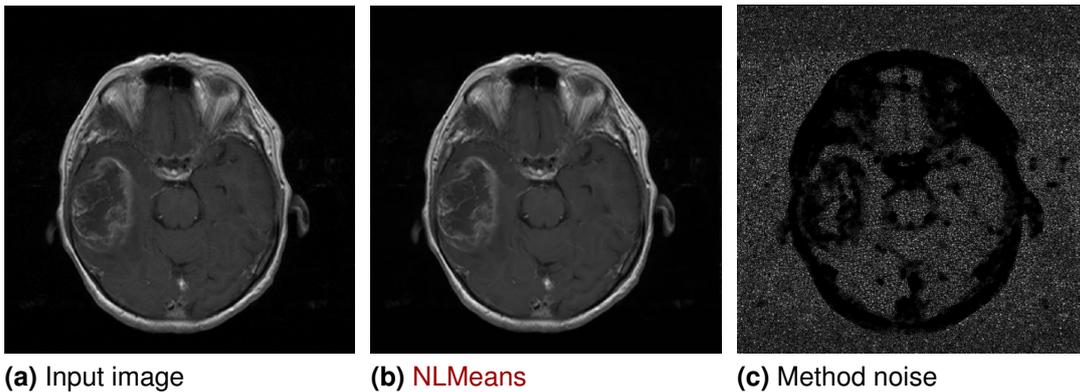
$$NL[v]_i := \frac{\sum_{j \in \Gamma} K_{\sigma_r}(\|f(N_i) - f(N_j)\|^2) f_j}{\sum_{j \in \Gamma} K_{\sigma_r}(\|f(N_i) - f(N_j)\|^2)}$$

where  $m$  is the number of dimensions,  $i = (i_1, \dots, i_m)^\top \in \mathbb{N}^m$  and  $j = (j_1, \dots, j_m)^\top \in \mathbb{N}^m$  are pixel positions.  $K_r$  denotes the Gaussian weight function with standard deviation  $\sigma_r$ .

As the unrestricted **NLM** compares for every pixel in the image its neighbourhood with all other neighbourhoods, it has an extreme computational burden. The search for similar pixels is for this reason restricted to a search window  $S_i$  around the regarded pixel  $\mathbf{i} \in \Gamma$ . This window can also have different shapes and sizes. We use again a circular shape to preserve the rotational invariance of the method and mirror at the border. Hence  $S_i$  is defined by its radius  $r_s$  as

$$S_i := \{\mathbf{k} \in \Gamma : \|\mathbf{i} - \mathbf{k}\| \leq r_s\}$$

**NLM** has consequently three parameters. As mentioned before, the similarity window  $N$  has to be small enough to preserve details and also large enough to be unsusceptible against noise. Additionally, the computation time is strongly connected to the radius  $r_N$ . The radius  $r_S$  of the search window  $S_i$  defines the area around  $\mathbf{i} \in \Gamma$  in which the algorithm searches for pixels similar to  $\mathbf{i}$ . For this reason, the size of  $S_i$  has also a high influence on the computational effort.

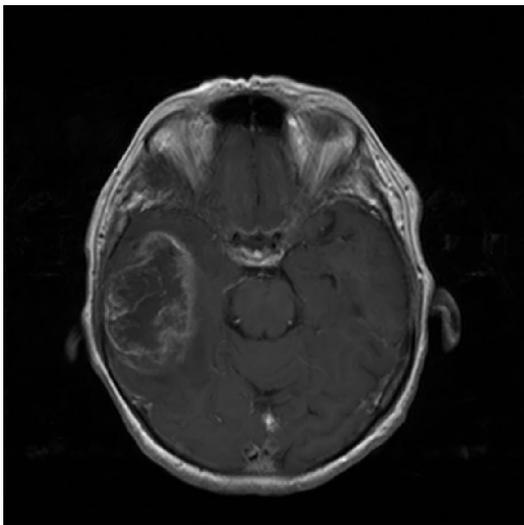


**Figure 2.6:** Method noise of **NLM**,  $r_N = 3, r_S = 5, \sigma_r \approx 51.2$ . The method noise is gamma corrected,  $\gamma = 1.2$ .

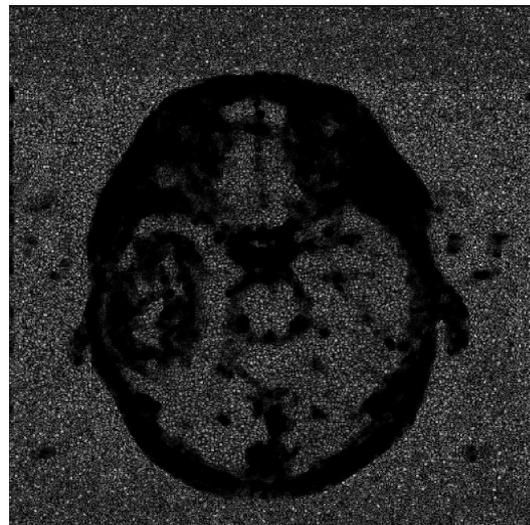
The third parameter, which is also the most important one is the standard deviation  $\sigma_r$  of the Gaussian kernel that weights the similarity between patches. If this parameter is set too small nearly no noise will be removed, but if  $\sigma_r$  is too large, the **NLM** will blur the image.

The result of **NLM** means is shown in **Figure 2.6**. Considering **Figure 2.6c**, we can observe that little structures are removed and that the method noise is very similar to Gaussian noise.

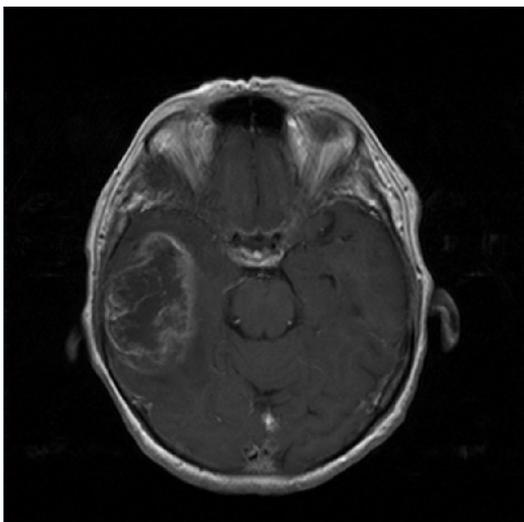
In **Figure 2.7** the influence of a wrong set parameter  $\sigma_r$  to the result is presented. If the standard deviation of the Gaussian weight function is too low, this filter does also not remove the noise, as depicted in **Figure 2.7d**. On the other hand, if this parameter is set too high, the method blurs the image and the method noise contains a lot of structures as shown in **Figure 2.7f**.



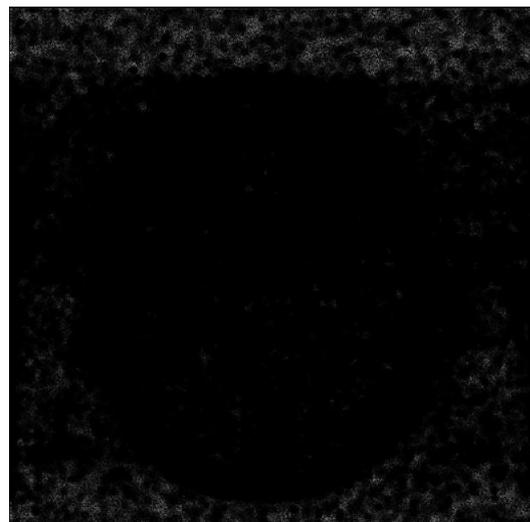
(a)  $r_N = 3, r_N = 5, \sigma_r = 42$ .



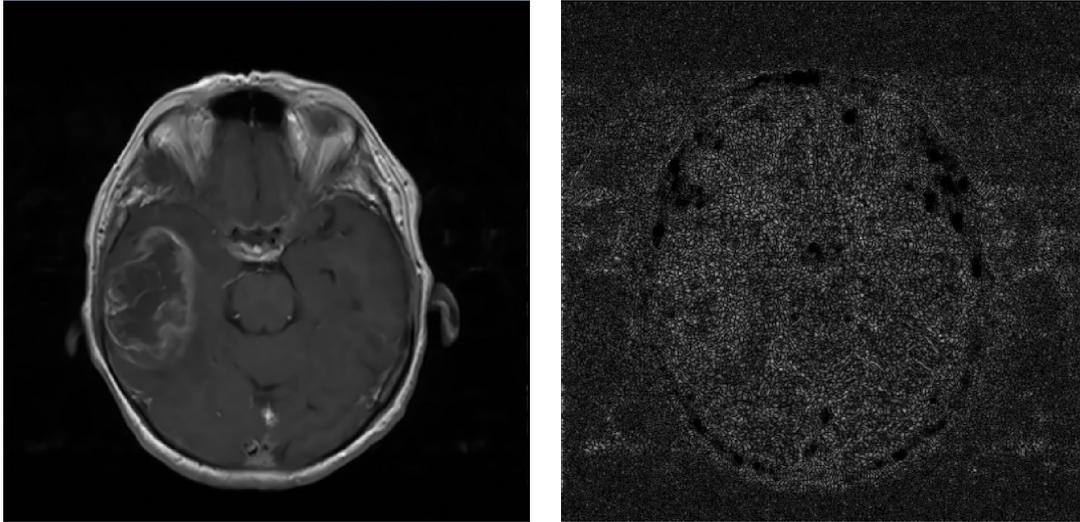
(b) Method noise of (a).



(c)  $r_N = 3, r_N = 5, \sigma_r = 10$ .



(d) Method noise of (c).



(e)  $r_N = 3, r_N = 5, \sigma_r = 170$ .

(f) Method noise of (e).

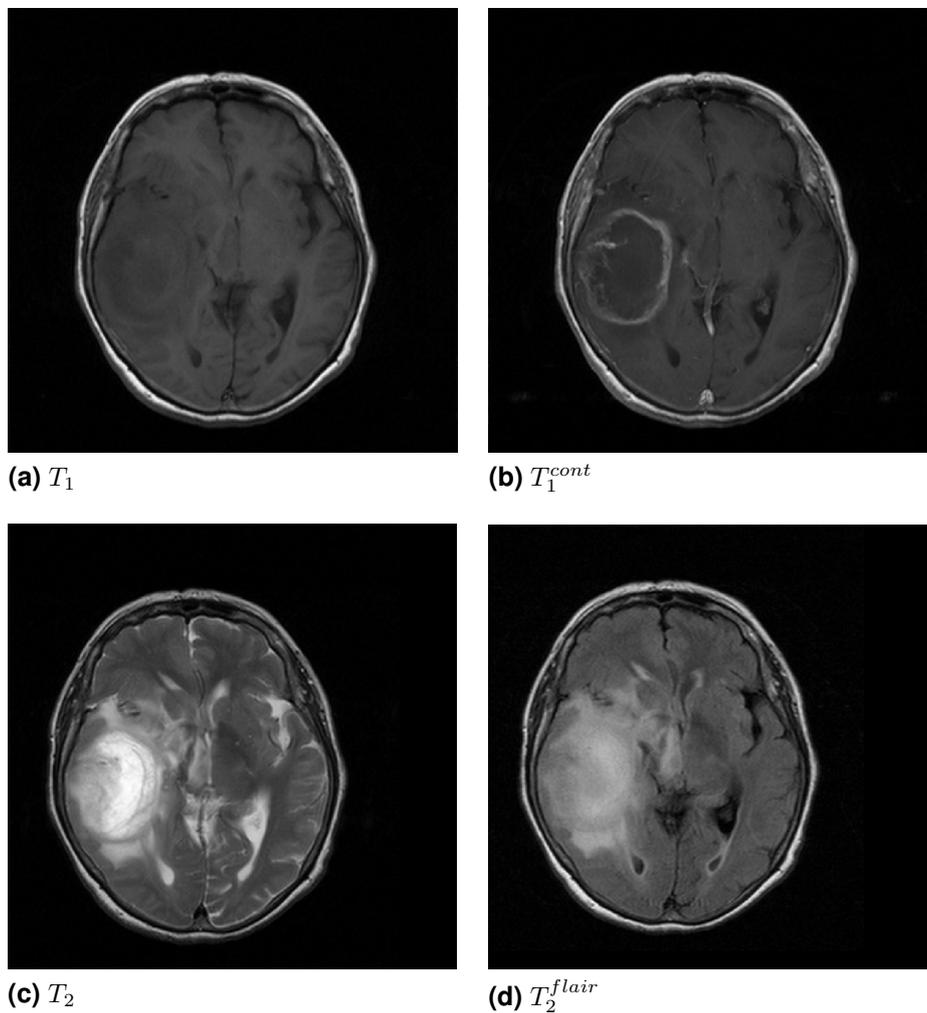
**Figure 2.7:** Influence of  $\sigma_r$  to **NLMeans**. The method noise is gamma corrected,  $\gamma = 1.2$ .

Comparing **Figure 2.7b** and **Figure 2.7f** one can observe that with a correct parameter setting, the **NLMeans** preserves all the structures while it blurs most of the details with a too large standard deviation of the Gaussian weighting function. We explain in detail in **Section 5.1** how the parameters can be adopted automatically to an image.

The **NLMeans** algorithm is well suited for a parallel implementation as the denoised value of each pixel can be computed separately. For this reason we use a CUDA[39] based GP-GPU implementation.

## 3 Registration

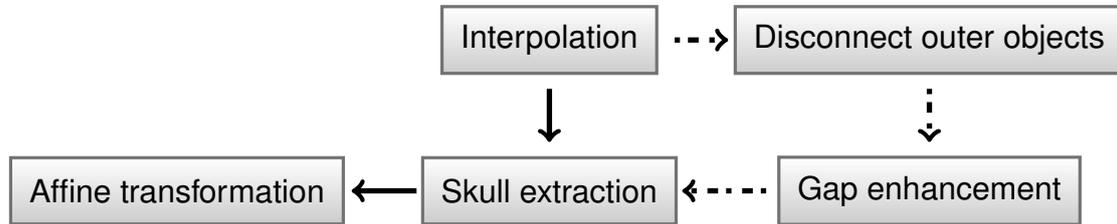
Information gained from different MRI sequences is usually of a complementary nature, cf. **Section 1.1**. Hence a proper integration of data obtained from the separate sequences is desired. A first step in this integration process is to bring the modalities involved into spatial alignment, a procedure referred to as *registration*.



**Figure 3.1:** Different sequences of subject *G13*, Slice 7.

To register two sequences means to align them, such that common features overlap and differences between them are emphasised and readily visible. There are

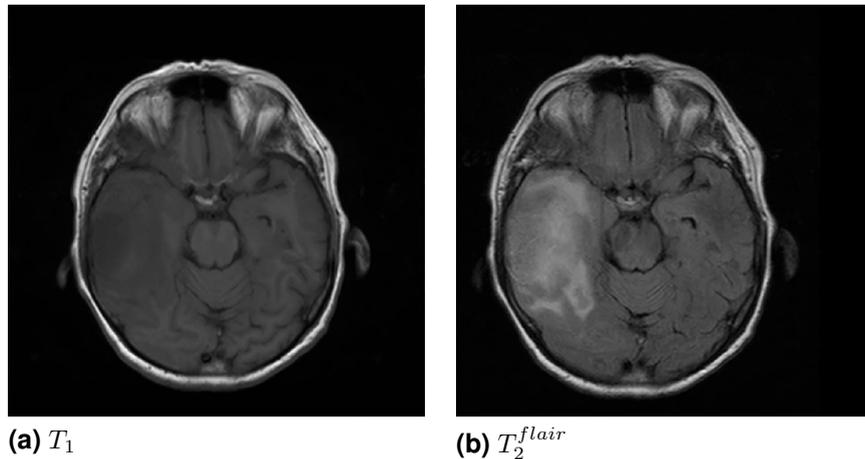
many clinical applications requiring registration, but in this thesis we only consider different sequences of **MRI** which are acquired directly after each other. When registering image sequences, we are determining a geometric transformation which brings the sequences in precise spatial correspondence to each other.



**Figure 3.2:** Scheme of the Registration Process.

We align therefore in the first step the sampling rates of the **MRI** scans, see **Section 3.1**. As shown in **Figure 3.1** the **MRI** sequences offer multi-modal information about the same object, i.e. the brain. The only information that is represented the same way in all the sequences is the shape of the skull. Because of this, we decided to use this information to register the **MRI** scans.

By means of a **CV model** we split the images into fore- and background, cf. **Section 3.4**. Then we select the largest segment, i.e. the skull and close holes inside it, see **Section 3.5**. We use this mask in the end to determine the geometric transformation between the **MRI** sequences, cf. **Section 3.6**. As we mentioned before, the scans are acquired in a row. We assume therefore that the transformations between the different sequences can be described by a subset of affine transformations, i.e. translation, rotation and scaling.



**Figure 3.3:** Skull shape of different sequences.

On the one hand, this approach is very efficient and robust against noise as the **CV model** does not rely on an edge detector. On the other hand, difficulties occur when the outer contour of the skull is not nearly identical for the images to be

registered. **Figure 3.3** shows the  $T_1$  and  $T_2^{flair}$  image of subject  $G13$  at  $z \approx 6\text{cm}$ , i.e. slice 10. In the  $T_1$  image, the ears are clearly observable while they are barely visible in the  $T_2^{flair}$  image. This results in two very different registration masks. Consequently, the registration of those images is of low quality.

We overcome this problem by preprocessing such images before we split them into fore- and background with the **CV model**. As the ears disturb our registration process, we have to separate them from the skull. We achieve this by a short evolution under *Coherence Enhancing Diffusion (CED)*, cf. **Section 3.2**. Afterwards, we enlarge the gap between the contour of the skull and the ear by extracting only bright details with a white top hat, see **Section 3.3**. After this step, both process chains for registration are identical.

In the latter, we explain first how we align the sampling rates of the **MRI** sequences.

## 3.1 Registration of Grid Dimensions

As mentioned before, the first step in the registration process is the alignment of the sampling rates of voxels, due to mismatching sampling rates of different **MRI** sequences. Furthermore, the acquisition has a non-homogeneous resolution resulting in a fine within-slice and a coarse across-slice resolution.

In order to maintain all information, we decided to rescale always to the finest sampling rate. Thus, we have to interpolate the coarser resolutions to a finer grid.

### 3.1.1 Classical Interpolation

*Interpolation* is the recovery of continuous data from discrete data within a known range of abscissa [53, 66]. The *classical interpolation formula* is given by

**Definition 32 (Classical Interpolation Formula)**

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^m} f_{\mathbf{k}} \varphi_{int}(\mathbf{x} - \mathbf{k}) \quad \forall \mathbf{x} = (x_1, x_2 \dots x_m)^\top \in \mathbb{R}^m, \varphi_{int} : \mathbb{R}^m \rightarrow \mathbb{R}$$

where  $f(\mathbf{x})$  is the interpolated value at coordinate  $\mathbf{x}$ , expressed as a weighted average of the discrete samples  $f_{\mathbf{k}}$  where the weights are determined by the *synthesis function*  $\varphi_{int}(\mathbf{x} - \mathbf{k})$ .

As denoted in **Definition 32**, the interpolated value is a linear combination of all samples  $f_{\mathbf{k}} \in \mathbb{Z}^m$ , irrespective of always finite number of known samples. Hence, to satisfy the formal conditions of **(32)** the finite amount of samples must be extended to infinity. This can be done by setting suitable boundary conditions, e.g. using mirror symmetries. Obviously, the only remaining freedom lies in the

synthesis function [52, 53, 66].

To restrict the interpolation formula in **Definition 32** to exact interpolation, it has to satisfy the *interpolation condition*

$$f(\mathbf{n}) = f_{\mathbf{n}} \quad \forall \mathbf{n} \in \mathbb{Z}^m. \quad (3.1)$$

Furthermore, the synthesis function must vanish for all integer values except at the origin, where it has to be 1. More formally, this is denoted by

$$\varphi(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} = \mathbf{0}^\top \\ 0 & \mathbf{x} \in \mathbb{Z}^m \setminus \mathbf{0}^\top \end{cases} \quad (3.2)$$

Optimising the classical interpolation approaches show to be diminishing. Thévenaz et al. suggest therefore to use the so-called *generalised interpolation formula*, defined by

**Definition 33 (Generalised Interpolation Formula)**

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{Z}^m} c_{\mathbf{k}} \varphi(\mathbf{x} - \mathbf{k}) \quad \forall \mathbf{x} = (x_1, x_2 \dots x_m)^\top \in \mathbb{R}^m, \varphi : \mathbb{R}^m \rightarrow \mathbb{R}$$

that allows to carry out the interpolation in two steps by introduction of coefficients  $c_{\mathbf{k}}$  instead of the sample values  $f_{\mathbf{k}}$  [52].

The classical interpolation formula is obviously a special case of the general one for  $c_{\mathbf{k}} = f_{\mathbf{k}}$ . However, by splitting the interpolation in two steps and neglecting the interpolation constraint, we allow for the use of a much broader class of synthesis functions, some with better properties like small support and excellent approximation order.

Unfortunately, this advantages come with a higher computational burden. Hence, we face a trade-off between quality and cost: The approximation error should be as small as possible to introduce the least amount of distortions while we deal with a large amount of volumetric medical image data.

The high computational effort can be reduced on the one hand by using basis functions which are separable, so

$$\varphi_{sep}(\mathbf{x}) = \prod_{i=1}^m \varphi(x_i) \quad \forall \mathbf{x} = (x_1, x_2 \dots x_m)^\top \in \mathbb{R}^m. \quad (3.3)$$

Then the data processing breaks down to one dimensional computations. On the other hand, we demand the basis functions to have a small support interval. As we also want to preserve the spatial relations, the symmetry

$$\varphi(\mathbf{x}) = \varphi(-\mathbf{x}) \quad \forall \mathbf{x} \in \mathbb{R}^m \quad (3.4)$$

of the basis functions is important. The registration of the grid dimensions is the first step in a large process chain. As image processing methods often need derivatives, we also restrict ourselves to basis functions with high regularity.

We explain in the following the popular B-spline interpolation. B-splines are maximally continuous by having the smallest possible support interval and an excellent approximation order.

### 3.1.2 B-Spline Interpolation

B-Splines belong to a family of functions that enjoy maximal order and minimal support called Moms [52]. All of these functions can be represented as the weighted sum of a B-Spline and its derivatives. As the resulting functions have a lower regularity, we decided to use pure B-Splines because they are those Moms-functions that are maximally differentiable. However there are functions in this family, i.e. the o-Moms, that have a slightly lower approximation error [52].

The most easiest synthesis function made of B-splines is the first order box function for symmetric nearest neighbour interpolation, denoted by

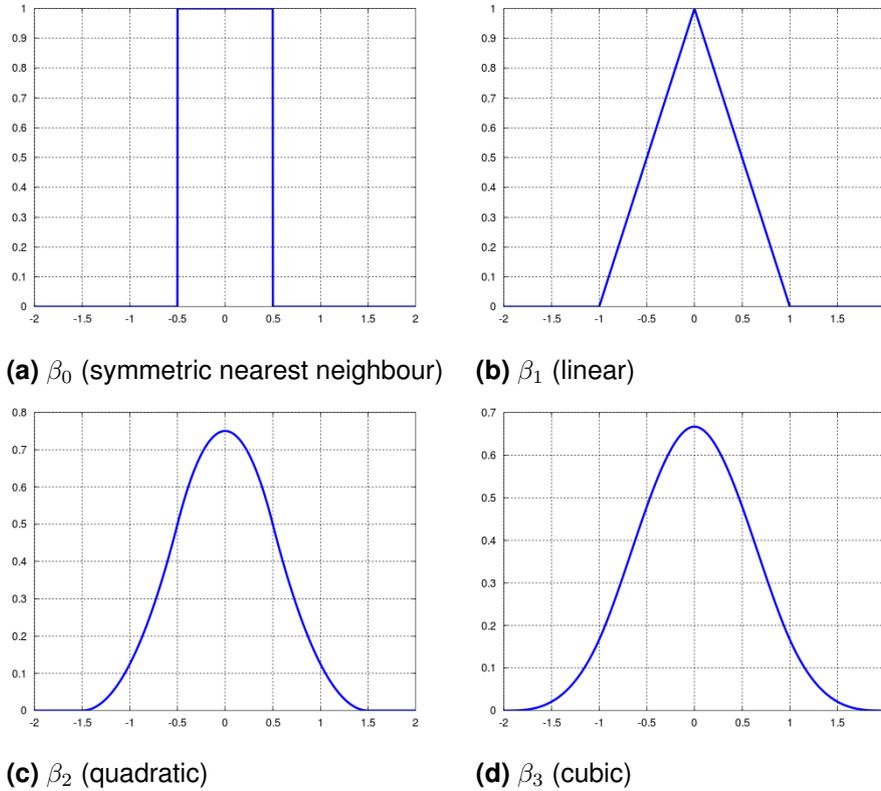
$$\beta_0(x) = \begin{cases} 1 & |x| < \frac{1}{2} \\ \frac{1}{2} & |x| = \frac{1}{2} \\ 0 & otherwise \end{cases} \quad (3.5)$$

As shown in **Figure 3.4a**, this spline is interpolating. It is very simple and efficient as always just one sample  $f_{\mathbf{k}}$  contributes to it [52].

However, as the  $\beta_0$ -spline is made of a square pulse, its approximation order is one and it can only represent piecewise-constant functions.

Before we illustrate how we derive higher-order B-splines, we introduce some notation. The *order*  $L$  of a basis function represents the approximation order. The *regularity* of a B-spline  $\beta$  is denoted as  $\beta \in C^m$  indicating that the resulting interpolant is  $m$ -times differentiable. The *support*  $W(\beta)$  characterises the interval contributing to the interpolant and the *degree*  $n$  specifies that any polynomial of degree  $n$  can be represented by this spline [52, 53].

Synthesis functions of higher order can be iteratively derived by convolving a B-spline with a square pulse, i.e. the synthesis function for  $\beta_0$ <sup>1</sup>.



**Figure 3.4:** Synthesis functions made of B-splines. (a)  $\beta_0$  and (b)  $\beta_1$  are interpolating as  $\beta_0(0) = \beta_1(0) = 1$ .

A B-spline of order  $L$  is then characterised as

$$\begin{aligned} \beta_n &= \beta_{n-1} * \beta_0, \\ n &= L - 1, \\ W(\beta_n) &= L, \\ \beta_n &\in C^{n-1}. \end{aligned} \tag{3.6}$$

The B-spline  $\beta_1$  of degree 1, defined as

$$\beta_1(x) = \begin{cases} 1 - |x| & |x| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

is also called linear interpolation. It can represent linear functions and is, as shown in **Figure 3.4b** interpolating. Due to their sharp transitions,  $\beta_0$  and  $\beta_1$  exacerbate the occurrence of blocking artifacts [52].

<sup>1</sup> As the theory of B-splines is very complex and we only illustrate some basic facts, we refer to [52, 53, 57, 58].

All B-splines of degree  $n \geq 1$  are on the one hand not interpolating and on the other hand they also do not have sharp transitions anymore. The first approximating synthesis function made of a B-spline is then

$$\beta_2(x) = \begin{cases} \frac{3}{4} - x^2 & |x| < \frac{1}{2} \\ \frac{1}{2}(\frac{3}{2} - |x|)^2 & \frac{1}{2} \leq |x| < \frac{3}{2} \\ 0 & \text{otherwise} \end{cases} \quad (3.8)$$

The synthesis made of  $\beta_2$  describes quadratic functions and is sketched in **Figure 3.4c**. As it is just one time differentiable according to (3.6) we derive the fourth order B-spline

$$\beta_3(x) = \begin{cases} \frac{2}{3} - x^2 + \frac{1}{2}|x|^3 & |x| < 1 \\ \frac{1}{6}(2 - |x|)^3 & 1 \leq |x| < 2 \\ 0 & \text{otherwise} \end{cases} \quad (3.9)$$

where

$$\begin{aligned} \beta_3 &= \beta_2 * \beta_0, \\ n &= 3, \\ W(\beta_3) &= 4, \\ \beta_3 &\in C^2. \end{aligned} \quad (3.10)$$

This synthesis function is depicted in **Figure 3.4d** and represent cubic functions. It looks similar to a Gaussian and it can be shown that  $\beta_n$  converges to a Gaussian for  $n \rightarrow \infty$  where the maximal relative error between  $\beta_3$  and a Gaussian with identical variance is about 3.5% [23, 52, 53]. The interpolation is computed in two steps. First we solve the generalised interpolation formula (33) for the coefficients  $c_{\mathbf{k}}, \mathbf{k} \in \mathbb{Z}^N$ . We do this by an one-dimensional algorithm as the B-spline interpolation is separable. We solve row by row and column by column the following linear system of equations:

$$\begin{pmatrix} \beta_3(0) & \beta_3(1) & \dots & \beta_3(N-1) \\ \beta_3(1) & \beta_3(0) & \dots & \beta_3(N-2) \\ \vdots & \vdots & \dots & \vdots \\ \beta_3(N-1) & \beta_3(N-2) & \dots & \beta_3(0) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix} \quad (3.11)$$

where  $N = n_x$  for a row and  $N = n_y$  for a column, respectively. The support interval of  $\beta_3$  is  $[-2, 2]$ . Therefore, (3.11) breaks down to a system with a tridiagonal matrix

$$\begin{pmatrix} \frac{2}{3} & \frac{1}{6} & 0 & \dots & \dots & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ 0 & \dots & \dots & 0 & \frac{1}{6} & \frac{2}{3} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ \vdots \\ c_{N-1} \\ c_N \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_{N-1} \\ f_N \end{pmatrix} \quad (3.12)$$

We solve this system with the *Thomas Algorithm* and mirror at the border, see **Section 1.2.10**. In the second step, we determine the desired values  $f(\mathbf{x})$ ,  $\mathbf{x} \in \Omega$  from the coefficients  $c_{\mathbf{k}}$ .

In our opinion, bi-cubic B-Splines are the best trade-off between computation time, quality and regularity. We resample the image data to the finest resolution occurring in the data. The sampling of the image data we have is always relatively dense in  $x$  and  $y$  direction, i.e.  $h_x \approx 0.4mm$ ,  $h_y \approx 0.4mm$ . Hence we have enough supporting points to guarantee a reasonable interpolation result. The across-slice resolution is unfortunately always very coarse and in average 15 times lower than the within-slice sampling, i.e.  $h_z \approx 6mm$ . As the approximation error is too high to get reliable data and the amount of data also increases dramatically by factor 15 for an equidistant sampling, we decided not to interpolate in the depth.

For this reason, we illustrate in the latter most of the procedures only for the two dimensional case. However, all of the methods can easily be extended to three dimensions.

## 3.2 Coherence Enhancing Diffusion Filtering

We mentioned before, that we use coherence-enhancing diffusion filtering to remove distortions from the contour of the skull. This filter technique is able to complete interrupted lines and to enhance flow-like structures [61, 63, 64].

In **CED** filtering, the symmetric and positive definite diffusion tensor  $D = (d_{ij}) \in \mathbb{R}^{2 \times 2}$  is chosen as a function of the local image configuration by adapting it to the structure tensor  $J_\rho(\nabla u_\sigma)$ , cf. **Section 1.2.6**.

The diffusion tensor  $D$  is constructed on the one hand by using the same eigenvectors as the structure tensor and on the other hand by the eigenvalues

$$\begin{aligned} \lambda_1 &:= \alpha \\ \lambda_2 &:= \begin{cases} \alpha & \text{if } \kappa = 0 \\ \alpha + (1 - \alpha) \exp\left(\frac{-C}{\kappa}\right) & \text{otherwise} \end{cases} \end{aligned} \quad (3.13)$$

where  $C$  is a threshold parameter and  $\kappa := (\mu_1 - \mu_2)^2$  is the strength of the local orientation. For  $\kappa \gg C$  the eigenvalue  $\lambda_2$  tends to 1 where  $\kappa \ll C$  leads to  $\lambda_2 \approx \alpha$ .

Hence, the **CED** filter acts mainly in the coherence-orientation and the smoothing increases with the strength of the local orientation of the structure [63].

The introduction of the exponential function guarantees that the smoothness of the structure tensor carries over to  $D$  and that  $\lambda_2$  does not exceed 1 while  $\alpha$  ensures that the diffusion process never stops [64]. Even if  $\kappa \rightarrow 0$ , i.e. a homogeneous area, a small linear diffusion with diffusivity  $\alpha > 0$  remains [63].

Finally, the **IBVP** of the diffusion equation

**Definition 34 (IBVP of CED)**

$$\begin{aligned}\partial_t u &= \operatorname{div}(D\nabla u), \\ u(\mathbf{x}, 0) &= f(\mathbf{x}) \quad \mathbf{x} \in \Omega, \\ \frac{\partial u}{\partial \mathbf{n}}(\mathbf{x}, t) &= 0 \quad \mathbf{x} \in \partial\Omega\end{aligned}$$

with

$$D = \begin{pmatrix} d_{1,1} & d_{1,2} \\ d_{2,1} & d_{2,2} \end{pmatrix} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$$

can be rewritten as

$$\partial_t u = \partial_x(a \partial_y u) + \partial_x(b \partial_y u) + \partial_y(b \partial_x u) + \partial_y(c \partial_y u) \quad (3.14)$$

The standard discretisation gives the stencil weights shown in **Figure 3.5**.

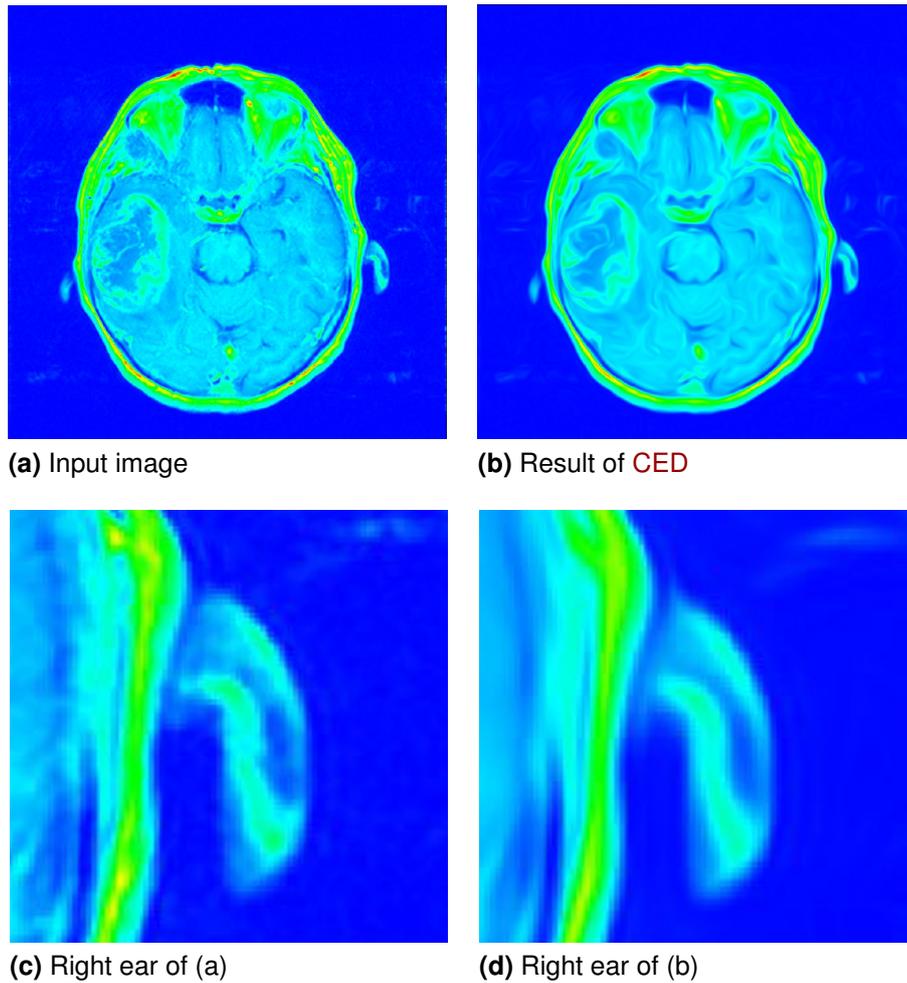
$\frac{-b_{i-1,j} - b_{i,j+1}}{4h_x h_y}$	$\frac{c_{i,j+1} + c_{i,j}}{2h_y^2}$	$\frac{b_{i+1,j} + b_{i,j+1}}{4h_x h_y}$
$\frac{a_{i-1,j} + a_{i,j}}{2h_x^2}$	$-\frac{a_{i-1,j} + 2a_{i,j} + a_{i+1,j}}{2h_x^2}$ $-\frac{c_{i,j-1} + 2c_{i,j} + c_{i,j+1}}{2h_y^2}$	$\frac{a_{i+1,j} + a_{i,j}}{2h_x^2}$
$\frac{b_{i-1,j} + b_{i,j-1}}{4h_x h_y}$	$\frac{c_{i,j-1} + c_{i,j}}{2h_y^2}$	$\frac{-b_{i+1,j} - b_{i,j-1}}{4h_x h_y}$

**Figure 3.5:** Stencil weights for standard discretisation.

This scheme requires a very small time step  $\tau$  to be stable,

$$\tau = \frac{1}{\frac{2}{h_x^2} + \frac{2}{h_y^2}}. \quad (3.15)$$

However, as we perform only a short evolution, this is not a problem. Nevertheless, there are several ways to address this drawback, for example the *Fast explicit diffusion* [23].



**Figure 3.6:** Result of CED filtering with a short evolution.  $T = 1.5$ ,  $C = 1$ ,  $\sigma = 0.5$ ,  $\rho = 1.5$ ,  $\alpha = 0.001$

We use CED to enhance the coherence along the skull, which disconnects the ears from it. This is shown in **Figure 3.6**. We explain in the following how the gap between the skull and the ear can be enlarged.

### 3.3 Nonflat Morphological Filtering

Mathematical Morphology, introduced by Serra and Matheron in the 1960's[34, 49], analyses the shape of objects in an image and is one of the most successful classes of image analysis methods.

It is invariant under monotone grey-level rescalings. Hence, brightness of illumination and image contrast do not matter [61].

The basic transforms of mathematical morphology are *dilation* and *erosion*, defined as an interaction between the image and a *structuring element*  $B$ .

**Definition 35 (Dilation with a Structuring Element)**

$$(f \oplus B)(\mathbf{x}) := \sup \{f(\mathbf{x} - \mathbf{y}) \mid \mathbf{y} \in B\}$$

Dilation replaces the grey value of a continuous image  $f(\mathbf{x})$  by its supremum in the structuring element  $B$  as denoted by **Definition 35**. The erosion is defined analogously, replacing the grey value by its infimum within the mask:

**Definition 36 (Erosion with a Structuring Element)**

$$(f \ominus B)(\mathbf{x}) := \inf \{f(\mathbf{x} + \mathbf{y}) \mid \mathbf{y} \in B\}$$

The structuring element can be of different shapes: circular elements are rotationally invariant while quadratic ones are separable. Unfortunately, there is no structuring element that offers both separability and rotational invariance.

Therefore we decided to use a more general framework, the *nonflat morphology* introduced by van den Boomgard [5]. Instead of a structuring element, nonflat morphology uses a structuring function  $b$ ,

$$b : \mathbb{R}^n \rightarrow \mathbb{R}. \quad (3.16)$$

where  $n$  is the number of dimensions. Its building blocks dilation

**Definition 37 (Dilation with a Structuring Function)**

$$(f \oplus b)(\mathbf{x}) := \sup \{f(\mathbf{x} - \mathbf{y}) + b(\mathbf{y}) \mid \mathbf{y} \in \mathbb{R}^n\}$$

and erosion

**Definition 38 (Erosion with a Structuring Function)**

$$(f \ominus b)(\mathbf{x}) := \inf \{f(\mathbf{x} + \mathbf{y}) - b(\mathbf{y}) \mid \mathbf{y} \in \mathbb{R}^n\}$$

are defined similar to their counterparts with a flat structuring element. This similarity is natural as the non-flat morphology resembles the flat one [61] when the structuring function  $b$  is defined as,

$$b(\mathbf{x}) := \begin{cases} 0 & \text{if } \mathbf{x} \in B \\ -\infty & \text{else.} \end{cases} \quad (3.17)$$

The main difference between both morphological filter techniques is that the non-flat morphology renounces the morphological invariance, i.e. it is not invariant under monotonically increasing grey scale transformations [61].

Nevertheless, as we are mainly interested in the always strong contours describing the shape of the skull this drawback has no influence in our application.

However, van den Boomaard showed that paraboloids are separable and rotationally invariant structuring functions [5]. Following his suggestion, we use

**Definition 39 (Quadratic Structuring Function<sup>2</sup>)**

$$b(\mathbf{x}, t) = -\frac{|\mathbf{x}|^2}{4t} \quad t > 0$$

as structuring function. The value of  $t$  controls the penalising for increasing distances to the regarded pixel. In the latter, we recycle our implementation with this structuring function for the reinitialisation of the level set function of the **CV model**, cf. **Section 3.4** to

$$b(\mathbf{x}) = -|\mathbf{x}|^2. \quad (3.18)$$

This is possible because a nonflat erosion of an image  $f(\mathbf{x})$  with foreground 0 and background  $-\infty$  is equivalent to the euclidean squared distance function [67]. We use this structuring function for our implementations of nonflat morphology.

### 3.3.1 Dilation

Analogously to the dilation with a structuring element, the dilation with a structuring function is one of the two basic transforms of the nonflat morphology. As structuring function we use (39) and the nonflat dilation is in our application therefore defined as

**Definition 40 (1D Dilation)**

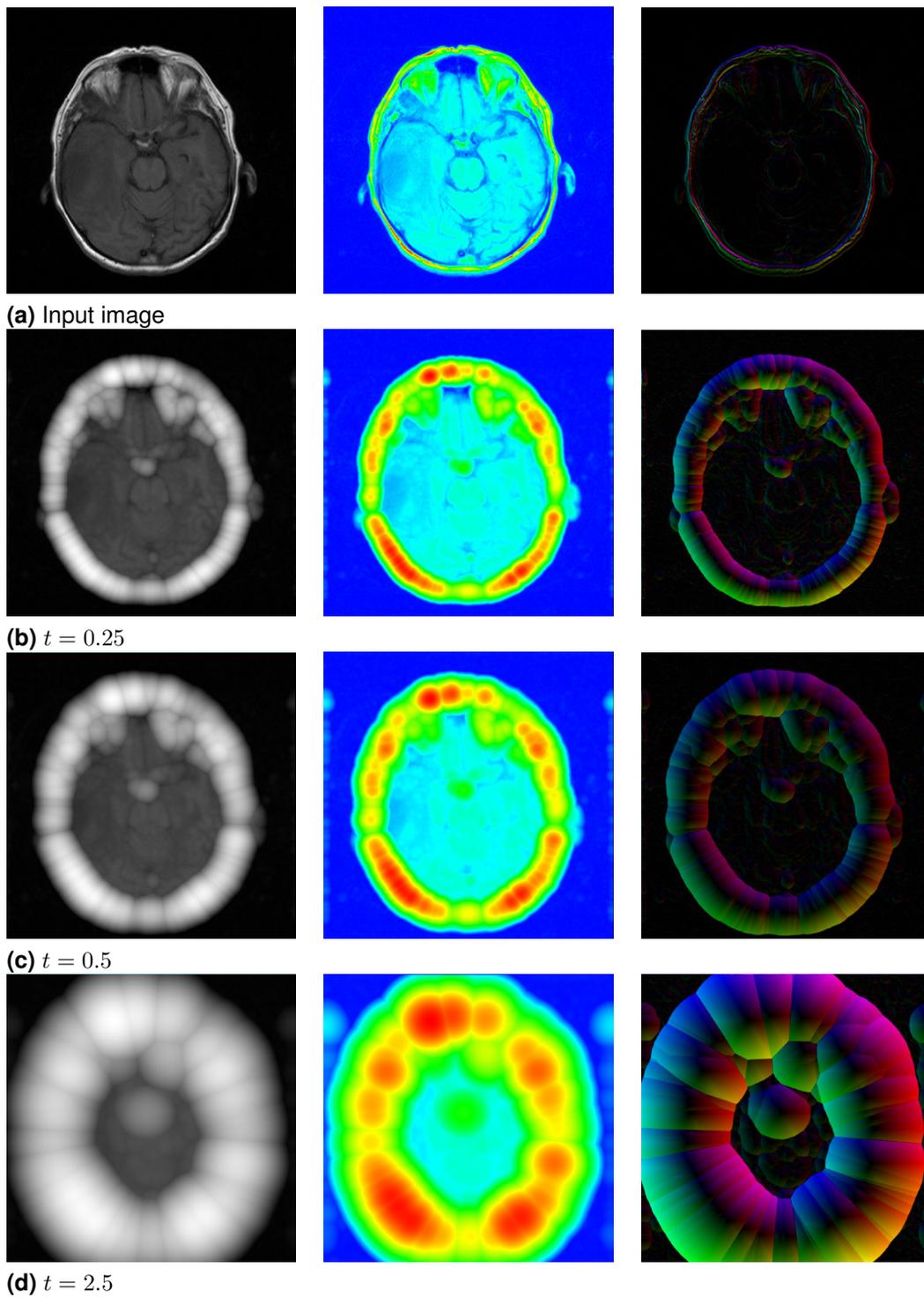
$$(f \oplus b)(x) := \sup_y \left\{ f(x - y) - \frac{y^2}{4t} \right\} = \sup_z \left\{ f(z) - \frac{(x - z)^2}{4t} \right\}$$

where  $z = x - y$ .

The grey value at position  $x$  is replaced by its supremum in the structuring function  $b$ . The main difference between nonflat (40) and flat dilation (35) is thereby the infinite support of the structuring function  $b$  in contrast to the limited area of the structuring element  $B$  [68].

**Figure 3.7** presents the images after dilation with different values of  $t$ . One can see that for increasing  $t$ , the distance from a pixel to the maximum value in its region is less penalised. Therefore bright areas grow for increasing  $t$ .

<sup>2</sup> Dilation of  $f(\mathbf{x})$  with this quadratic structuring function is known to be equivalent to the evolution of the **PDE**  $u_t = |\nabla u|^2$  at time  $t$  [6].



**Figure 3.7:** Dilation with structuring function (39). **Left column:** Grey scale images. **Middle column:** Pseudo colouring of the left row. **Right column:** Coloured gradients.

The right column shows the coloured gradients where we can notice that the gradients remain nearly zero at the boundary of the skull for increasing  $t$ .

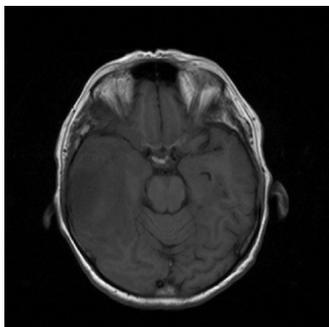
### 3.3.2 Erosion

*Erosion* with the structuring function given by (39) is defined as

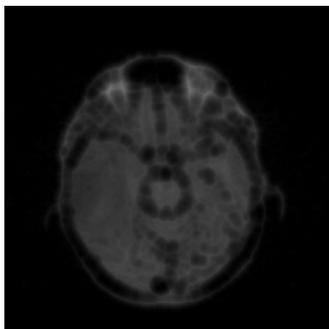
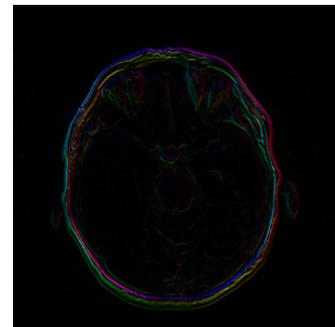
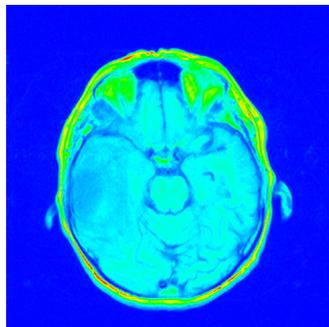
**Definition 41 (1D Erosion)**

$$(f \ominus b)(x) := \inf_y \left\{ f(x+y) + \frac{y^2}{4t} \right\} = \inf_z \left\{ f(z) + \frac{(z-x)^2}{4t} \right\}$$

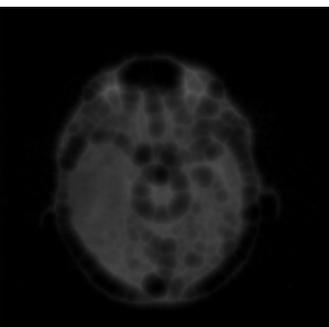
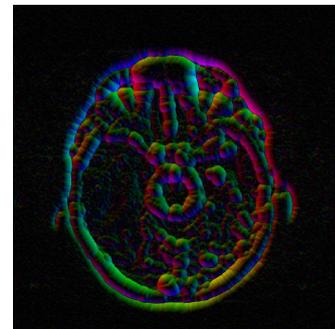
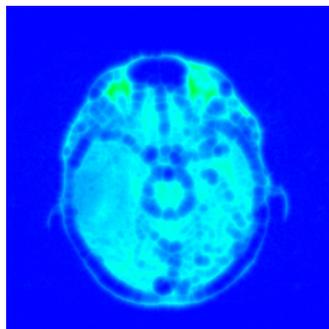
is the counterpart to **Definition 40** and replaces the grey value at position  $x$  by its infimum in the support of the structuring function  $b$ . We can consider in **Figure 3.8** that erosion acts exact in the opposite way to the dilation: The less the distance to a minimum is penalised the more bright details are removed. Thereby the remaining structures also shrink for increasing  $t$ .



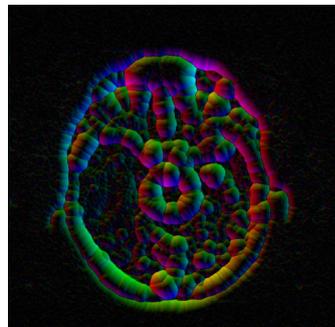
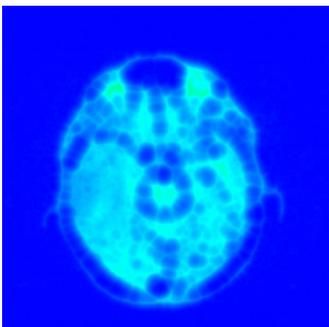
(a) Input image

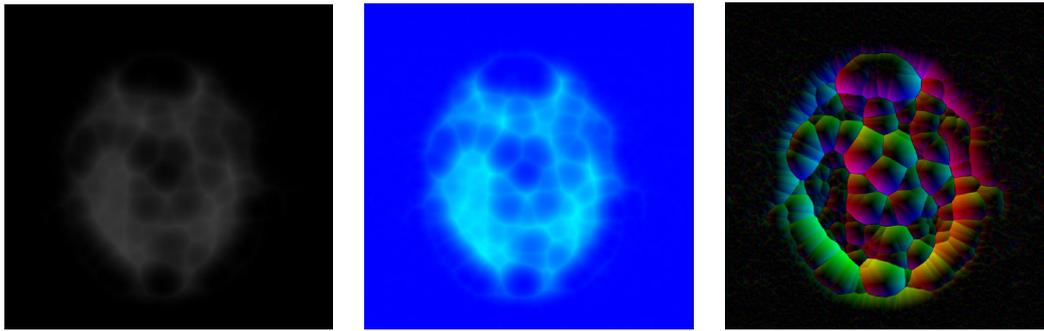


(b)  $t = 0.25$



(c)  $t = 0.5$



(d)  $t = 2.5$ 

**Figure 3.8:** Erosion with structuring function (39). **Left column:** Grey scale images. **Middle column:** Pseudo colouring of the left row. **Right column:** Coloured gradients.

### 3.3.3 Opening

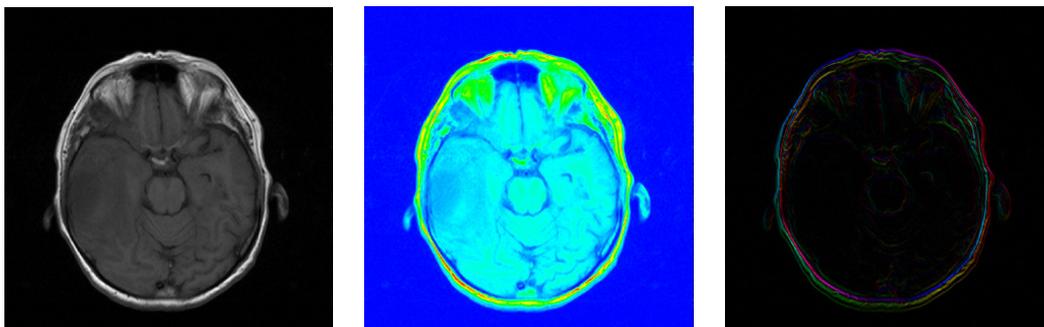
While dilation results in an expansion of bright areas, erosion shrinks those regions. Consequently, when objects are simplified and shrunk by an erosion, see Section 3.3.2, the shrinkage effects can be mostly removed by a subsequently dilation, cf. Section 3.3.1. The *opening*, defined as

**Definition 42 (1D Opening)**

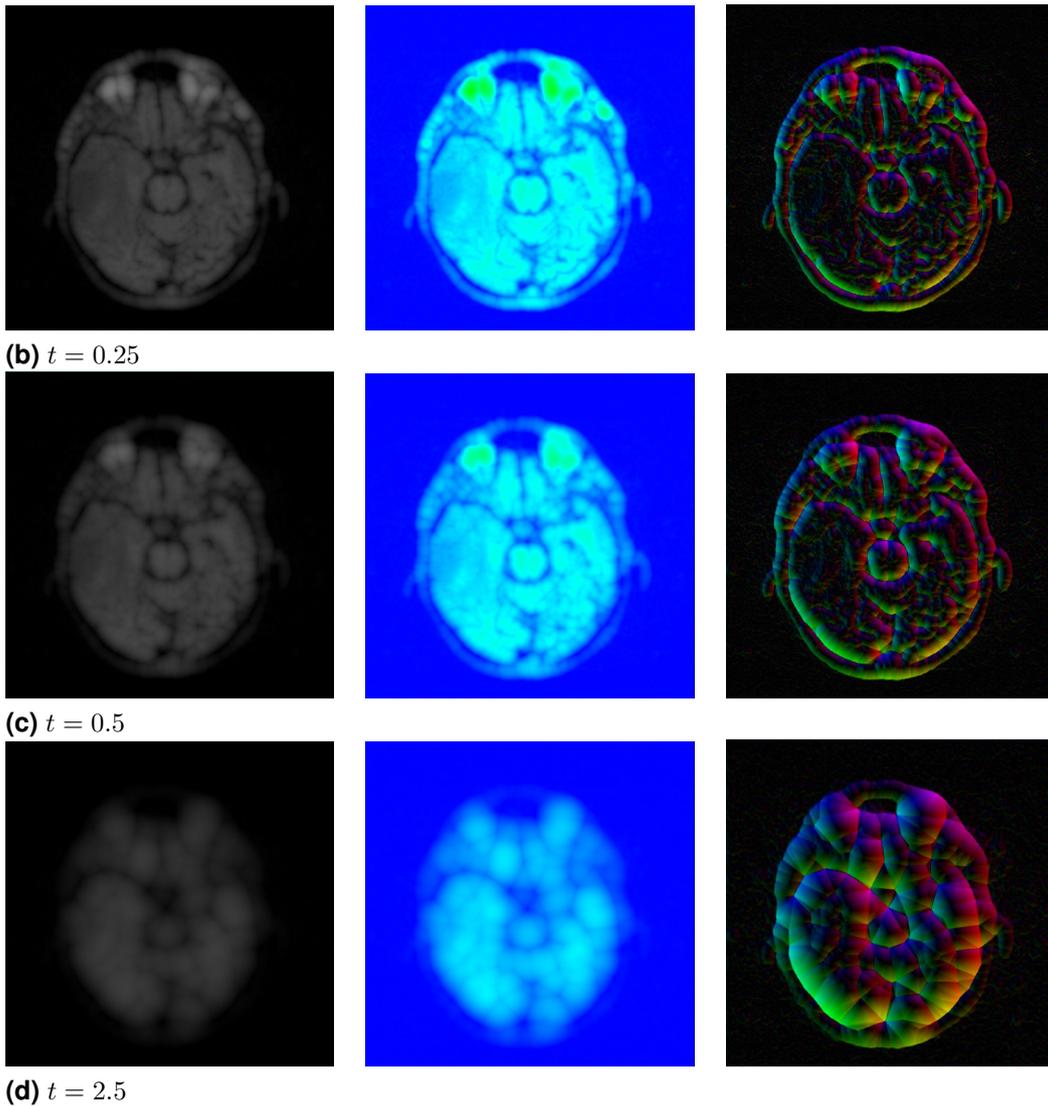
$$(f \circ b)(x) := ((f \ominus b) \oplus b)(x)$$

combines therefore both building blocks of morphology and removes bright details while reducing the shrinkage effects of erosion. Hence, opening is a morphological lowpass filter [66]. Additionally, it is an idempotent operation as it can be applied multiple times without changing the result beyond the initial application.

**Figure 3.9** displays openings of **Figure 3.9a** for different values of  $t$ .



(a) Input image



**Figure 3.9:** Opening with structuring function (39). **Left column:** Grey scale images. **Middle column:** Pseudo colouring of the left row. **Right column:** Coloured gradients.

Regarding **Figure 3.9** we can observe that openings with decreasing penalisation of the distance remove more and more bright details while the size of the remaining regions does not decrease that much. Although it looks in the grey scale and pseudo coloured images like a tremendous shrinkage of these regions, we can see in the gradient domain, that their area remain nearly the same but the morphological operations decreased the contrast.

However, this is the main drawback of the nonflat morphology compared to its flat counterpart. Due to the infinite support of the structuring function, it is not possible to guarantee that the shrinkage effects of the erosion are completely removed.

### 3.3.4 White Top Hat

Opening an image removes bright details, as we show in [Section 3.3.3](#). Consequently, we can extract those details by computing the difference of the original image and the image after an opening.

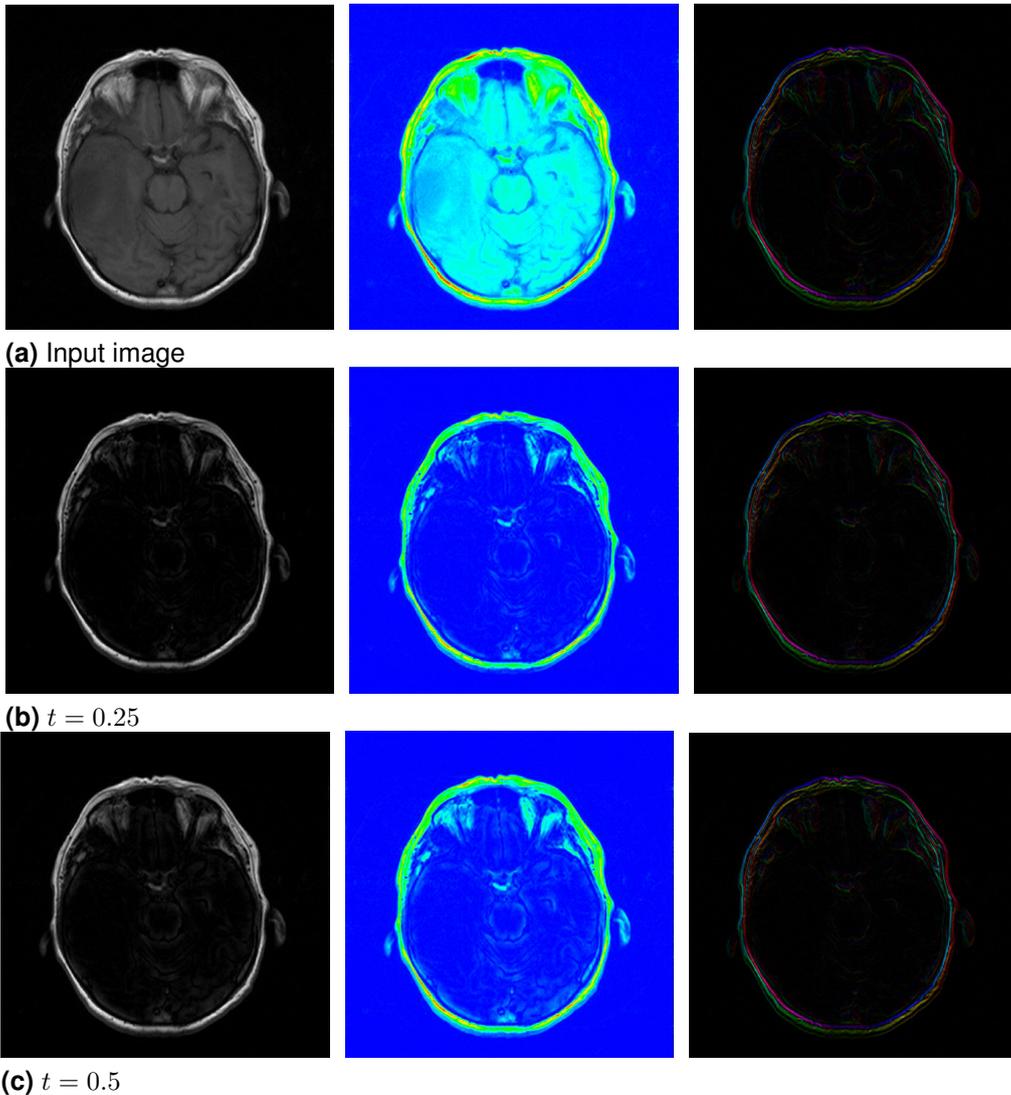
The name of this procedure is *white top hat* and it is defined as

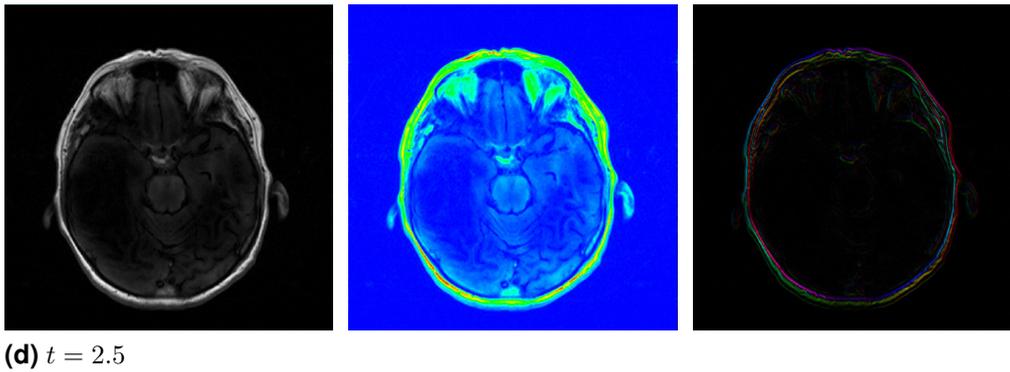
**Definition 43 (1D White Top Hat)**

$$WTH[f, b](x) := (f - (f \circ b))(x).$$

Hence, the white top hat is a morphological highpass filter and as the grey values of an opened image are for sure lower than in the original image, it guarantees a positive result by construction.

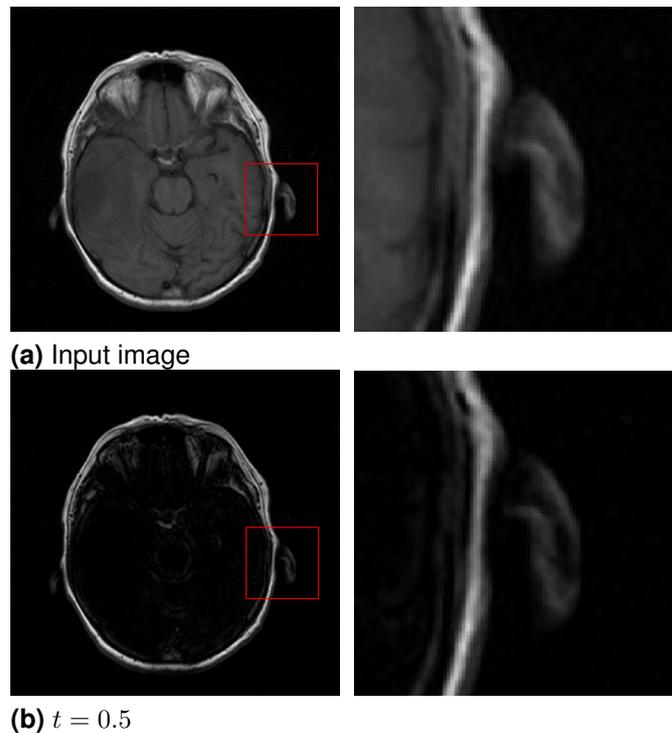
As shown in [Figure 3.10](#) for increasing  $t$ , white top hats of [Figure 3.10a](#) extract more bright structures.





**Figure 3.10:** White Top Hat with structuring function (39). **Left column:** Grey scale images. **Middle column:** Pseudo colouring of the left row. **Right column:** Coloured gradients.

In the registration process we are only interested in the shape of the skull. For this reason we use a low value for  $t$  to enforce that only very bright structures



**Figure 3.11:** Extraction of bright details with parabolic white top hat. **Left:** Input image and white top hat,  $t = 0.5$ . **Right:** Selected area before and after the white top hat.

are extracted. As we can see in **Figure 3.11** the shape of the skull is preserved while darker details, like the connection to the ears, are removed. As the nonflat morphology with structuring function (39) is separable, this algorithm is also well suited for parallel computing. We use therefore a CUDA[39] based GP-GPU implementation.

### 3.4 Chan-Vese Active Contour Model

The next step in our process chain to register the different **MRI** sequences is to extract the shape of the skull. For this we use the aforementioned **CV model** [13]. On the one hand this model is a reduction of the piecewise constant Mumford-Shah model, which is as denoted by **Definition 27** defined as

$$E_{MSFC}(u, C) = \int_{\Omega} (f(\mathbf{x}) - u(\mathbf{x}))^2 d\mathbf{x} + \alpha \text{Length}(C) \quad (3.19)$$

The image is partitioned in object and background as there are only two segments allowed, denoted by

$$u(\mathbf{x}) = \begin{cases} u_{in} & \mathbf{x} \text{ inside } C \\ u_{out} & \mathbf{x} \text{ outside } C \end{cases} \quad (3.20)$$

On the other hand, Chan and Vese added a term penalising the area enclosed by the curve. Hence, the Chan-Vese active contour model is defined as

**Definition 44 (Chan-Vese Active Contour Model)**

$$E_{CV}(u_{in}, u_{out}, C) = \lambda_1 \int_{\text{inside } C} (f(\mathbf{x}) - u_{in})^2 d\mathbf{x} + \lambda_2 \int_{\text{outside } C} (f(\mathbf{x}) - u_{out})^2 d\mathbf{x} + \alpha \text{Length}(C) + \mu \text{Area}(\text{inside } C)$$

where  $u_{in}$  and  $u_{out}$  are the arithmetic means of  $f(\mathbf{x})$  inside and outside the curve  $C$ .  $\lambda_1$  and  $\lambda_2$  are the weights for penalising the discrepancy between the input image  $f(\mathbf{x})$  and the arithmetic means inside and outside the curve [13].

In order to find a minimiser, the formulation requires minimising over all edge-sets  $C$ . This problem is addressed implicitly as Chan and Vese use a level set formulation and represent  $C$  as the zero-crossings of the Lipschitz continuous level set function  $\phi : \Omega \rightarrow \mathbb{R}$ , such that [13]:

$$\begin{cases} C = \partial w = \{\mathbf{x} \in \Omega : \phi(\mathbf{x}) = 0\} \\ \text{inside } C = w = \{\mathbf{x} \in \Omega : \phi(\mathbf{x}) > 0\} \\ \text{outside } C = \Omega \setminus \bar{w} = \{\mathbf{x} \in \Omega : \phi(\mathbf{x}) < 0\} \end{cases} \quad (3.21)$$

where  $\partial w$  is the border of  $w$ . With the Heaviside function  $H$  as indicator function for the set enclosed by  $C$ , defined as

**Definition 45 (Heaviside Function)**

$$H(z) = \begin{cases} 1 & (z \geq 0) \\ 0 & (z < 0) \end{cases}$$

and its distributional derivative<sup>3</sup> the one-dimensional Dirac measure  $\delta_0 = \frac{\partial}{\partial z} H(z)$  and following [71],

$$\begin{aligned}
 Length\{\phi = 0\} &= \int_{\Omega} |\nabla H(\phi(\mathbf{x}))| \, d\mathbf{x} = \int_{\Omega} \delta_0(\phi(\mathbf{x})) |\nabla \phi(\mathbf{x})| \, d\mathbf{x}, \\
 Area\{\phi \geq 0\} &= \int_{\Omega} H(\phi(\mathbf{x})) \, d\mathbf{x}, \\
 \int_{\phi > 0} (f(\mathbf{x}) - u_{in})^2 \, d\mathbf{x} &= \int_{\Omega} (f(\mathbf{x}) - u_{in})^2 H(\phi(\mathbf{x})) \, d\mathbf{x}, \\
 \int_{\phi < 0} (f(\mathbf{x}) - u_{out})^2 \, d\mathbf{x} &= \int_{\Omega} (f(\mathbf{x}) - u_{out})^2 (1 - H(\phi(\mathbf{x}))) \, d\mathbf{x}.
 \end{aligned} \tag{3.22}$$

The level set formulation of the **CV model** is then given by

$$\begin{aligned}
 E_{CV}(u_{in}, u_{out}, \phi) &= \lambda_1 \int_{\Omega} (f(\mathbf{x}) - u_{in})^2 H(\phi(\mathbf{x})) \, d\mathbf{x} \\
 &\quad + \lambda_2 \int_{\Omega} (f(\mathbf{x}) - u_{out})^2 (1 - H(\phi(\mathbf{x}))) \, d\mathbf{x} \\
 &\quad + \alpha \int_{\Omega} |\nabla H(\phi(\mathbf{x}))| \, d\mathbf{x} + \mu \int_{\Omega} H(\phi(\mathbf{x})) \, d\mathbf{x}
 \end{aligned}$$

The solution  $u$  (3.20) of this energy functional, can be written as

$$u(\mathbf{x}) = u_{in}H(\phi(\mathbf{x})) + u_{out}(1 - H(\phi(\mathbf{x}))), \quad \mathbf{x} \in \bar{\Omega}. \tag{3.23}$$

The minimisation of the energy functional is solved by alternately updating  $u_{in}$ ,  $u_{out}$  and  $\phi$  [13]. If  $\phi$  is fixed and the curve has a nonempty interior and exterior in  $\Omega$ , the current arithmetic means  $u_{in}$  and  $u_{out}$  are computed by

$$\begin{aligned}
 u_{in}(\phi) &= \frac{\int_{\Omega} f(\mathbf{x})H(\phi(\mathbf{x})) \, d\mathbf{x}}{\int_{\Omega} H(\phi(\mathbf{x})) \, d\mathbf{x}} \\
 u_{out}(\phi) &= \frac{\int_{\Omega} f(\mathbf{x})(1 - H(\phi(\mathbf{x}))) \, d\mathbf{x}}{\int_{\Omega} (1 - H(\phi(\mathbf{x}))) \, d\mathbf{x}}
 \end{aligned} \tag{3.24}$$

In the second update step,  $u_{in}$  and  $u_{out}$  are fixed and  $E_{CV}(u_{in}, u_{out}, \phi)$  is minimised w.r.t.  $\phi$ , i.e. we have to find a minimiser for

<sup>3</sup> The Heaviside function is not differentiable in the classical meaning. However, it can be differentiated in the sense of distributions, cf. [45].

$$\begin{aligned}
E_{CV}(\phi) = & \lambda_1 \int_{\Omega} (f(\mathbf{x}) - u_{in})^2 H(\phi(\mathbf{x})) \, d\mathbf{x} \\
& + \lambda_2 \int_{\Omega} (f(\mathbf{x}) - u_{out})^2 (1 - H(\phi(\mathbf{x}))) \, d\mathbf{x} \\
& + \alpha \int_{\Omega} |\nabla H(\phi(\mathbf{x}))| \, d\mathbf{x} + \mu \int_{\Omega} H(\phi(\mathbf{x})) \, d\mathbf{x}
\end{aligned} \tag{3.25}$$

The energy  $E_{CV}$  is non-convex and has to be solved numerically. As we use a gradient descent we regularise the Heaviside function  $H$  as suggested by [13] as

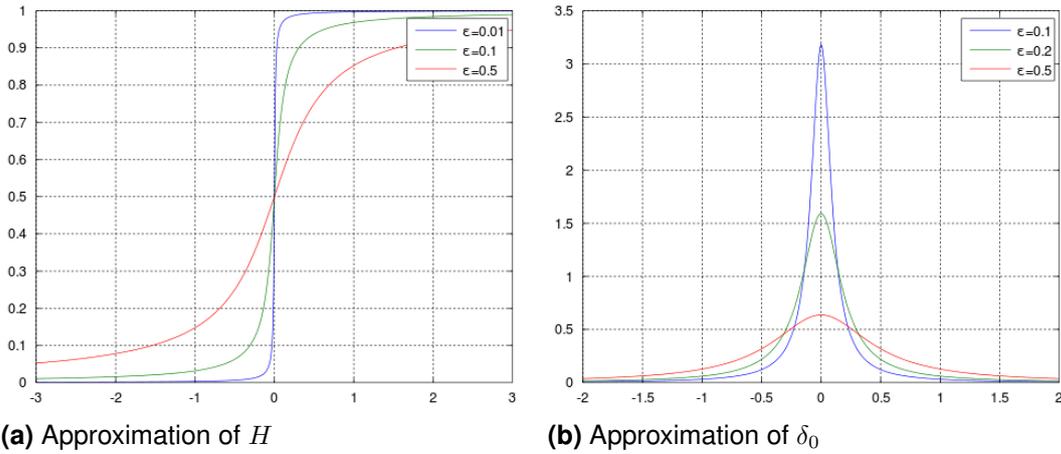
**Definition 46 (Regularised Heaviside Function)**

$$H_{\epsilon}(z) = \frac{1}{2} \left( 1 + \frac{2}{\pi} \arctan \left( \frac{z}{\epsilon} \right) \right)$$

and its derivative  $\delta_{\epsilon}$  as

$$\delta_{\epsilon}(z) = \frac{\epsilon}{\pi(\epsilon^2 + z^2)}. \tag{3.26}$$

The resulting approximations are shown in **Figure 3.12**. If  $H$  is regularised in



**Figure 3.12:** Approximations of  $H$  and  $\delta_0$  for different regularisation parameters  $\epsilon$ .

this way, the equation acts on all level curves and therefore more like a global minimiser [13]. We deduce the Euler-Lagrange equation of  $\nabla E_{CV}$

**Definition 47 (Euler-Lagrange Equation of  $\nabla E_{CV}$ )**

$$H'_{\epsilon}(\phi(\mathbf{x})) \left( \lambda_2 (f(\mathbf{x}) - u_{out})^2 - \lambda_1 (f(\mathbf{x}) - u_{in})^2 - \mu + \alpha \operatorname{div} \left( \frac{\nabla \phi(\mathbf{x})}{|\nabla \phi(\mathbf{x})|} \right) \right) = 0$$

with the boundary condition

$$\frac{\partial_{\epsilon}(\phi(\mathbf{x}))}{|\nabla \phi(\mathbf{x})|} \frac{\partial \phi(\mathbf{x})}{\partial \mathbf{n}(\mathbf{x})} = 0 \quad \mathbf{x} \in \partial \Omega$$

and the gradient descent w.r.t.  $\phi$  is then given by

**Definition 48 (Gradient Descent of  $\nabla E_{CV}$ )**

$$\partial_t \phi(\mathbf{x}) = \delta_\varepsilon(\phi(\mathbf{x})) \left( \lambda_2(f(\mathbf{x}) - u_{out})^2 - \lambda_1(f(\mathbf{x}) - u_{in})^2 - \mu + \alpha \operatorname{div} \left( \frac{\nabla \phi(\mathbf{x})}{|\nabla \phi(\mathbf{x})|} \right) \right)$$

with the boundary conditions

$$\begin{aligned} \phi(\mathbf{x}, 0) &= \phi_0(\mathbf{x}) & \mathbf{x} \in \Omega \\ \frac{\partial_\varepsilon(\phi(\mathbf{x}))}{|\nabla \phi(\mathbf{x})|} \frac{\partial \phi(\mathbf{x})}{\partial \mathbf{n}(\mathbf{x})} &= 0 & \mathbf{x} \in \partial\Omega \end{aligned}$$

where  $\frac{\partial \phi(\mathbf{x})}{\partial \mathbf{n}(\mathbf{x})}$  is the derivative of  $\phi(\mathbf{x})$  in the direction of the outer normal  $\mathbf{n}$  [13, 66].

We follow [13] and discretise (48) with finite differences as denoted by **Equation 3.27**, cf. **Section 1.2.5**. We also mix various finite differences to approximate

$$\operatorname{div} \left( \frac{\nabla \phi(\mathbf{x})}{|\nabla \phi(\mathbf{x})|} \right) = \partial_x \left( \frac{\partial_x \phi}{\sqrt{(\partial_x \phi)^2 + (\partial_y \phi)^2}} \right) + \partial_y \left( \frac{\partial_y \phi}{\sqrt{(\partial_y \phi)^2 + (\partial_x \phi)^2}} \right)$$

to center the result. The discretisation is given by

$$\begin{aligned} \frac{\phi_{i,j}^{n+1} - \phi_{i,j}^n}{\tau} &= \delta_\varepsilon(\phi_{i,j}^n) \left[ \lambda_2(f_{i,j} - u_{out})^2 - \lambda_1(f_{i,j} - u_{in})^2 - \mu \right. \\ &\quad + \alpha \left( \partial_x^- \left( \frac{\partial_x^+ \phi_{i,j}^{n+1}}{\sqrt{(\partial_x^+ \phi_{i,j}^n)^2 + (\partial_y \phi_{i,j}^n)^2}} \right) \right. \\ &\quad \left. \left. + \partial_y^- \left( \frac{\partial_y^+ \phi_{i,j}^{n+1}}{\sqrt{(\partial_y^+ \phi_{i,j}^n)^2 + (\partial_x \phi_{i,j}^n)^2}} \right) \right) \right] \end{aligned} \quad (3.27)$$

where  $\tau$  is the time step. Let

$$\begin{aligned} C_{1,i,j} &= \frac{\alpha}{\sqrt{(\partial_x^+ \phi_{i,j}^n)^2 + (\partial_y \phi_{i,j}^n)^2}} & C_{2,i,j} &= \frac{\alpha}{\sqrt{(\partial_x^+ \phi_{i-1,j}^n)^2 + (\partial_y \phi_{i-1,j}^n)^2}} \\ C_{3,i,j} &= \frac{\alpha}{\sqrt{(\partial_y^+ \phi_{i,j}^n)^2 + (\partial_x \phi_{i,j}^n)^2}} & C_{4,i,j} &= \frac{\alpha}{\sqrt{(\partial_y^+ \phi_{i,j-1}^n)^2 + (\partial_x \phi_{i,j-1}^n)^2}} \end{aligned}$$

and

$$R_{i,j} = \lambda_2(f_{i,j} - u_{out})^2 - \lambda_1(f_{i,j} - u_{in})^2 - \mu$$

then we can write (3.27) as

$$\frac{\phi_{i,j}^{n+1} - \phi_{i,j}^n}{\tau} = \delta_\varepsilon(\phi_{i,j}^n) [R_{i,j} + C_{1,i,j}(\phi_{i+1,j}^n - \phi_{i,j}^{n+1}) - C_{2,i,j}(\phi_{i,j}^{n+1} - \phi_{i-1,j}^n) + C_{3,i,j}(\phi_{i,j+1}^n - \phi_{i,j}^{n+1}) - C_{4,i,j}(\phi_{i,j}^{n+1} - \phi_{i,j-1}^n)]$$

Finally, we update  $\phi$  as

$$\phi_{i,j}^{n+1} = \frac{\phi_{i,j}^n + \tau \delta_\varepsilon(\phi_{i,j}^n) [R_{i,j} + C_{1,i,j}\phi_{i+1,j}^n + C_{2,i,j}\phi_{i-1,j}^{n+1} + C_{3,i,j}\phi_{i,j+1}^n + C_{4,i,j}\phi_{i,j-1}^{n+1}]}{1 + \tau \delta_\varepsilon(C_{1,i,j} + C_{2,i,j} + C_{3,i,j} + C_{4,i,j})}$$

In some experiments, we reinitialise  $\phi$  every step to the *euclidean signed distance function*

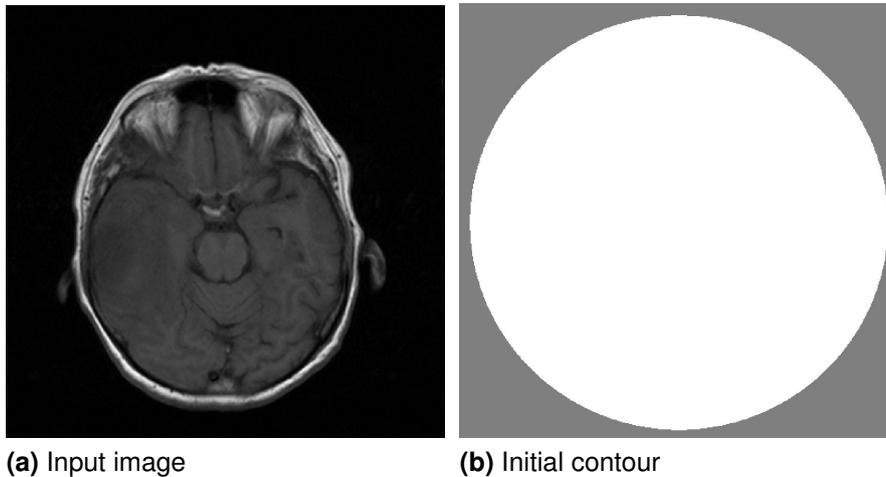
**Definition 49 (Euclidean signed Distance Function)**

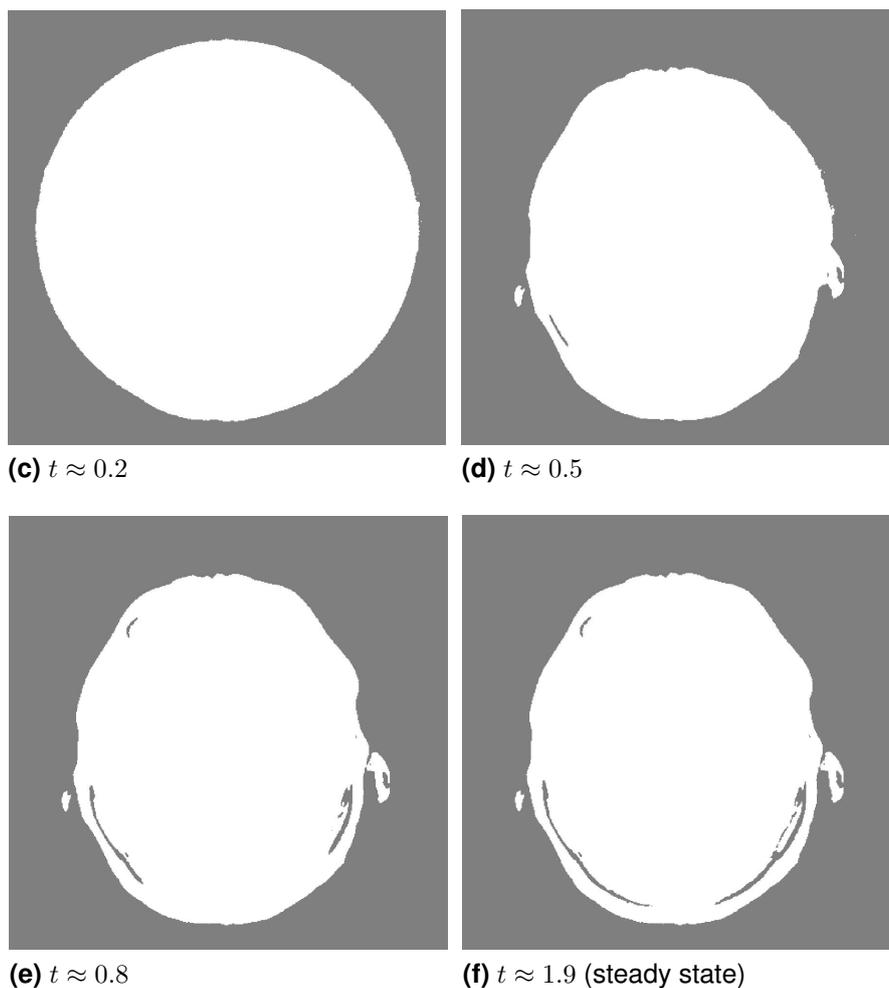
$$\psi(\mathbf{x}) = \begin{cases} \|\mathbf{x} - \phi_t(\mathbf{x})\| & \mathbf{x} \text{ inside } \phi \\ -\|\mathbf{x} - \phi_t(\mathbf{x})\| & \text{otherwise} \end{cases}$$

where  $\phi_t(\mathbf{x})$  is the solution at time  $t$

to its zero level curve. It is a rescaling and can prevent interior contours from growing [13]. For its implementation we use a morphological operation as described in Section 3.3. However, as the reinitialisation is extremely time consuming, we rarely use it.

The segmentation boundary of the steady state of the CV model is independent of the initialisation of the level set function [11]. We decided to use a circular initialisation, see Figure 3.13b. The advantage of this initialisation is, that we can guarantee that the object is enclosed by the contour, i.e. it is inside the contour. The evolution under the CV model for different evolution times is shown in Figure 3.13c to Figure 3.13f.





**Figure 3.13:** Evolution under the **CV model** for different evolution times,  $\lambda_1 = 1$ ,  $\lambda_2 = 1$ ,  $\mu = 0$ ,  $\alpha = 1$ ,  $\epsilon = h_x h_y$ ,  $\tau \approx 0.04$ , with reinitialisation of the level set function.

Inspired by CPU code we get from Levi Valgaerts<sup>4</sup>, we use a CUDA [39] based GP-GPU implementation of the **CV model**.

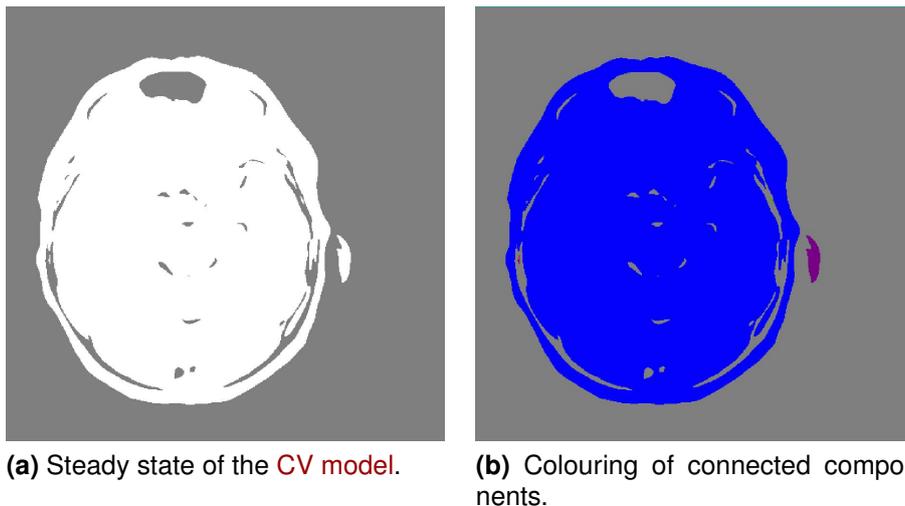
<sup>4</sup> [valgaerts@mpi-inf.mpg.de](mailto:valgaerts@mpi-inf.mpg.de)

### 3.5 Connected Component Labelling

The result of the segmentation can consist of several components while the largest component is always the skull. Due to this, our first step is to find the largest connected area. Connected component labelling scans an image and groups its pixels into components based on pixel connectivity. After all groups are determined, each pixel is labelled with a colour according to the component it was assigned to.

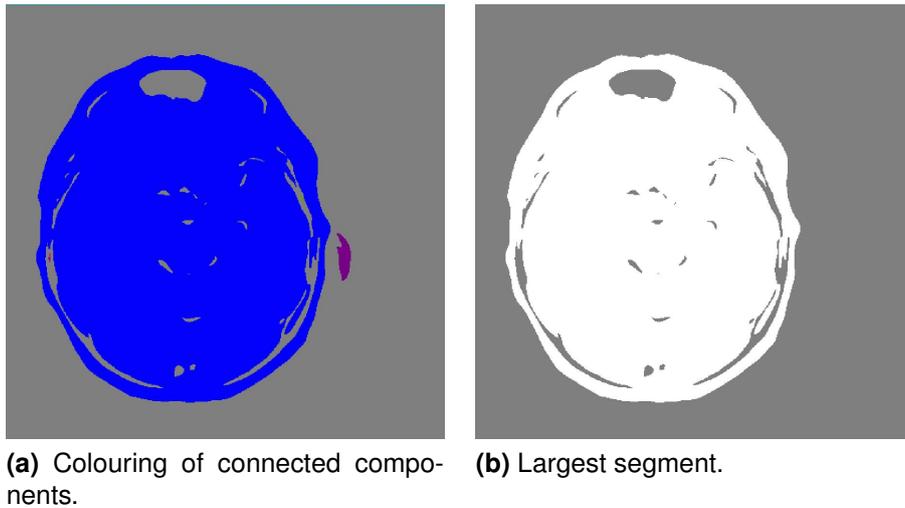
We use a classical connected component algorithm for graphs [46]. It is a row-by-row labelling with two passes. In the first pass we determine the equivalences and assign temporary labels.

We use a eight point neighbourhood to determine equivalences. Equivalence means that if the label of one of the eight direct neighbours of a pixel is different from the one assigned to the regarded pixel, both labels are regarded as equivalent and they are stored in the same equivalence class. We replace the temporary labels then in the second pass by the label of its equivalence class.



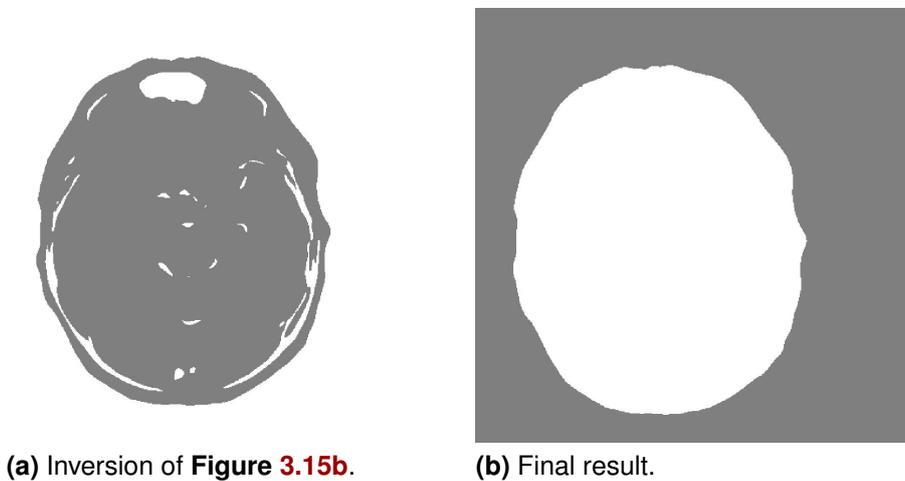
**Figure 3.14:** Labelling of connected components.

We use the connected component labelling in two ways. The first one addresses the segmentation of the shape of the skull. The steady state of the CV model can contain small segments. For this reason, we use the connected components algorithm to find the largest segment as shown in **Figure 3.14** and remove all others, cf. **Figure 3.15**.



**Figure 3.15:** Connected components.

Additionally, the largest segment can also contain holes, as depicted in **Figure 3.15b**. We invert therefore the mask image, shown in **Figure 3.16a**. Then we choose the largest component, i.e. the background of the image, and set all others to zero. Afterwards we invert the mask again and have a closed segment of the skull.



**Figure 3.16:** Closing of small holes in the skull segment.

The final result is presented by **Figure 3.16b**. It would also be possible to use a morphological operation to close small holes in the fore- and background. Unfortunately, this has in our opinion two major drawbacks. On the one hand

we do not know in advance the diameter of the holes. Hence, it would be difficult to determine the correct parameter for the morphology. On the other hand, a morphological process can also change the outer contour of our segment.

For this reason, we use the described connected component labelling to close holes. It does not need any parameter and preserves the outer contour.

## 3.6 Affine Body Transform

The final step in our registration process is the affine body transform to bring the MRI sequences in spatial correspondence. As we mentioned before, the MRI sequences are imaged in a row. We assume therefore that the transformations between them consist of a translation, rotation and a scaling. Hence, there is no need for a registration method which is capable to handle elastic transformations, like registration with mutual information [59] or normalised gradient fields [24].

### 3.6.1 Principal Axes Transform

We use for the registration the masks that we gain by the connected component labelling. The main observation is that the mask, i.e. the outer contour of the skull, is shaped like an ellipse in two dimensions and like an ellipsoid in three dimensions, respectively<sup>5</sup>.

An ellipse is uniquely located by the position of its center of mass and its orientation w.r.t. the center of mass [1, 27].

The result of the connected component labelling  $M$  is one for every pixel inside the segment and zero everywhere else. The mass centroids are then computed as [15]

$$c_x = \frac{\sum_{(i,j) \in \Gamma} iM(i,j)}{\sum_{(i,j) \in \Gamma} M(i,j)} \quad c_y = \frac{\sum_{(i,j) \in \Gamma} jM(i,j)}{\sum_{(i,j) \in \Gamma} M(i,j)}. \quad (3.28)$$

The *covariance matrix*  $\Sigma$  describes the variance of mass as a function of direction where

$$\Sigma = \begin{pmatrix} \sigma_X^2 & \sigma_{XY}^2 \\ \sigma_{XY}^2 & \sigma_Y^2 \end{pmatrix} \quad (3.29)$$

<sup>5</sup> We mentioned before, that the sampling rate w.r.t. the depth is too coarse to get a reasonable interpolation error. Consequently, we illustrate in the latter only the two dimensional registration. Nevertheless, the algorithm can be extended easily to three dimensions.

is symmetric and positive definite. As we know the centroid of each image, we compute (3.29) as

$$\begin{aligned}\sigma_X^2 &= \sum_{(i,j) \in \Gamma} (i - c_x)^2 f_{i,j} \\ \sigma_Y^2 &= \sum_{(i,j) \in \Gamma} (j - c_y)^2 f_{i,j} \\ \sigma_{XY}^2 &= \sum_{(i,j) \in \Gamma} (i - c_x)(j - c_y) f_{i,j}\end{aligned}\tag{3.30}$$

where  $c_x$  and  $c_y$  denote the  $x$  position and  $y$  position of the center of mass [27]. The covariance matrix permits an eigenvalue decomposition [36]:

$$\Sigma = D(\zeta)\Lambda D(\zeta)^{-1}\tag{3.31}$$

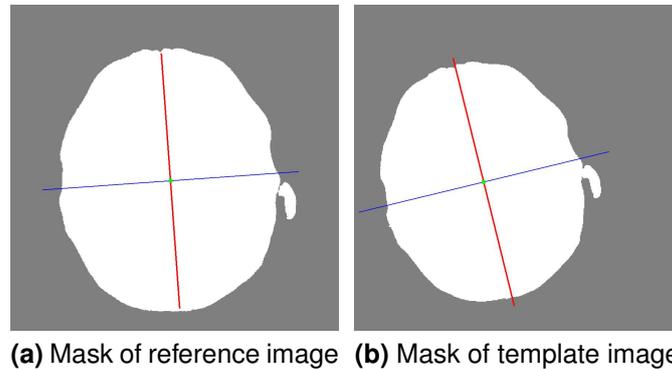
where  $D(\zeta)$  is a rotation and  $\Lambda$  a scaling matrix [48],

$$D(\zeta) = \begin{pmatrix} \cos(\zeta) & -\sin(\zeta) \\ \sin(\zeta) & \cos(\zeta) \end{pmatrix}, \quad \Lambda := \begin{pmatrix} \sigma_X^2 & 0 \\ 0 & \sigma_Y^2 \end{pmatrix}\tag{3.32}$$

The eigenvectors

$$\mathbf{w}_1 = \begin{pmatrix} \cos(\zeta) \\ \sin(\zeta) \end{pmatrix}, \quad \mathbf{w}_2 = \begin{pmatrix} -\sin(\zeta) \\ \cos(\zeta) \end{pmatrix}\tag{3.33}$$

span the principal axis [48] as shown in **Figure 3.17**.



**Figure 3.17:** Principal axes. **Red:** Dominant eigenvector. **Blue:** Non-dominant eigenvector.

With this information, we can register the images. We try to find a transformation  $\varphi(\mathbf{x}) : \Omega \rightarrow \Omega$  that produces the alignment between the reference  $R$  and the transformed version  $F$  of the template  $T$

$$F(\mathbf{x}) = T(\varphi(\mathbf{x})).\tag{3.34}$$

As we mentioned before, we assume an affine transformation. Hence,

$$\varphi(\mathbf{x}) = A\mathbf{x} + \mathbf{b} \quad (3.35)$$

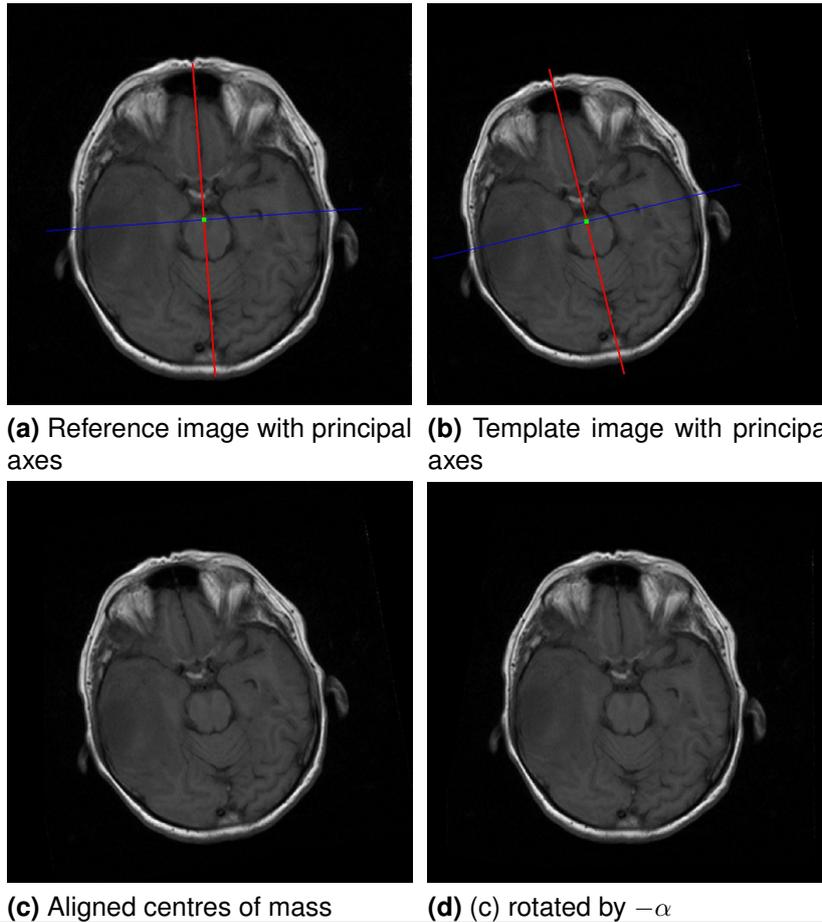
where

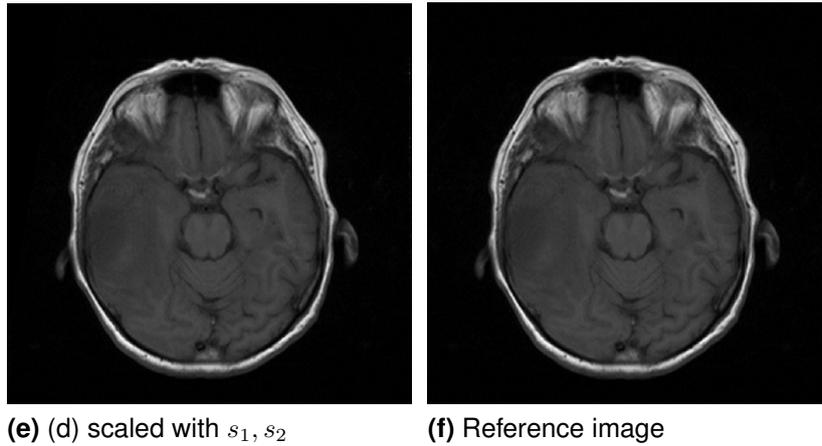
$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \underbrace{\begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}}_{\text{Rotation}} \underbrace{\begin{pmatrix} s_1 & 0 \\ 0 & s_2 \end{pmatrix}}_{\text{Scaling}} \quad (3.36)$$

$$\mathbf{b} = \underbrace{\begin{pmatrix} b_1 \\ b_2 \end{pmatrix}}_{\text{Translation}}.$$

To validate our procedure, we generated an artificial registration problem by rotating, scaling and translating an image, i.e. **Figure 3.18b** is the transformed version of **Figure 3.18a**.

We determine the transformation parameters in three steps. First we match the center of mass of the template image to the one of the reference image, i.e. the green markers in **Figure 3.18a** and **Figure 3.18b**. By this we compensate for the translation  $\mathbf{b}$ . An example for this is shown in **Figure 3.18c**.





**Figure 3.18:** Different transformations to align reference and template image.

The next step is to find the rotational transformation. For this, we compute the angle  $\alpha$  between the dominant eigenvectors of the template  $\mathbf{w}_{1,T}$  and the reference mask  $\mathbf{w}_{1,R}$  as

$$\cos(\alpha) = \frac{\mathbf{w}_{1,R}\mathbf{w}_{1,T}}{|\mathbf{w}_{1,R}||\mathbf{w}_{1,T}|} \quad (3.37)$$

We rotate then the template image by the negative angle, i.e.  $-\alpha$ . The result is depicted in **Figure 3.18d**.

The third step is to compensate for the scaling between different MRI sequences. We solve this by using the ratio of the eigenvalues of the covariance matrices

$$s_1 = \frac{\sigma_X^R}{\sigma_X^T} \quad s_2 = \frac{\sigma_Y^R}{\sigma_Y^T} \quad (3.38)$$

as they describe a scaling, shown in **Figure 3.18e**. Hence, the affine transformation can be written as

$$\varphi = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} s_1 & 0 \\ 0 & s_2 \end{pmatrix} + \begin{pmatrix} c_x^R - c_x^T \\ c_y^R - c_y^T \end{pmatrix}. \quad (3.39)$$

The last step is to apply this transformation to the template image. As the transformation possibly maps pixels to non-grid locations, we use a B-spline interpolation to determine the grey values of the transformed image  $F$ .

## 4 Tumour Segmentation

The segmentation of brain tumour images consists mainly of separating the tumour tissue from healthy brain tissue. As we mentioned in **Section 1.1**, the normal brain tissue can be split in grey matter, white matter and cerebrospinal fluid. A glioblastoma multiforme can be divided in central area of necrosis, oedema and lesions. Although it is relatively easy to detect abnormal tissues in **MRI** brain data, a reproducible and precise segmentation is difficult.

There are several ways to segment brain tumour images. A frequently used method is manual segmentation, where a human expert draws the boundaries of the tumour and the structures of interest. The main advantage of this method is, that human experts make also use of experience and additional knowledge like anatomy.

However, if the person defining the segmentation boundaries is not an expert, the segmentation result will be most likely of poor quality. Even if the manual segmentation is done by an expert, the task of marking the tumour regions slice by slice limits his view and generates jaggy segments. As a result, the segmented images are not optimal and show a striping effect [43].

Over and above that, manually drawn boundaries are also operator dependent. The study in [35] states that the variation between different operators ranges from 11% to 69% with an average variability rate of 28%. Additionally, they quantified a variability of 20% when the same expert repeats the task three times at one month intervals. Consequently, the selected segments are subject to large variability [70] and not reproducible.

Another popular method for brain tumour segmentation is thresholding. Thresholding creates binary images by setting all pixels below a specific threshold value to zero and all pixels above this threshold to one [66]. The major drawback of this method is that only the intensity is considered. All other relationships between the pixels, like spatial context, are neglected [66]. The segmentation is therefore often of poor quality. Additionally, finding a good threshold value can be a tedious and very time consuming task.

The most frequently used method for brain tumour segmentation is region growing. It is a simple region-based segmentation technique that is highly interactive

---

and is used to extract a connected region of similar pixels. Region growing starts with at least one seed, that belongs to the area of interest. This method uses the spatial context by checking if the neighbours of the seed satisfy the similarity criteria [66]. These criteria are determined for example by a range of pixel intensity values or other features.

However, region growing has some major drawbacks. It requires, that the region differs clearly from its background [66]. A glioblastoma multiforme has diffuse boundaries [20] and this segmentation method can therefore lead to undesirable growth [66]. Furthermore, the MRI data we have consist in average of more than one thousand different grey values. It is therefore nearly impossible for human observers to identify the object boundaries.

As we mentioned in **Section 3.4**, the **CV model** does not rely on edge detection. Accordingly, it is well suited to extract objects with diffusive boundaries. MRI sequences offer multi-modal information about the same object and we register the different sequences before we extract the tumour. Thus, we can regard them as different channels of a vector valued image.

## 4.1 Chan-Vese Active Contour Model for Vector Valued Images

The *Chan-Vese active contour model for vector valued images (CVec model)* allows us to include all information available to determine the tumour boundaries [11]. The definition of the **CV model** (**Definition 44**) is stated as

$$E_{CV}(u_{in}, u_{out}, C) = \lambda_1 \int_{\text{inside } C} (f(\mathbf{x}) - u_{in})^2 d\mathbf{x} + \lambda_2 \int_{\text{outside } C} (f(\mathbf{x}) - u_{out})^2 d\mathbf{x} \\ + \alpha \text{Length}(C) + \mu \text{Area}(\text{inside } C).$$

The parameter  $\mu$  sets the penalty for the area enclosed by the contour. Penalising this area is only meaningful when we have a prescribed inside versus outside of the segmented boundary.

As we do not have this information beforehand, we never penalise the enclosed area and the **CV model** becomes

$$E_{CV}(u_{in}, u_{out}, C) = \lambda_1 \int_{\text{inside } C} (f(\mathbf{x}) - u_{in})^2 d\mathbf{x} + \lambda_2 \int_{\text{outside } C} (f(\mathbf{x}) - u_{out})^2 d\mathbf{x} \\ + \alpha \text{Length}(C).$$

The **CVec model** is a straightforward extension of the **CV model** for scalar valued images to the vector valued case. Let  $\mathbf{f} := (f^1, \dots, f^V)^\top$  be a vector valued image

where  $V$  is the number of channels. Hence,  $f^i$  is the  $i$ th channel of a vector valued image  $\mathbf{f} \in \Omega$  with  $i = 1, \dots, V$ . The **CV model** for scalar valued images is extended to the **CVec model** [11] by

**Definition 50 (CV model for vector valued images)**

$$\begin{aligned} E_{CVec}(\mathbf{u}_{in}, \mathbf{u}_{out}, C) &= \frac{1}{V} \int_{\text{inside } C} \sum_{i=1}^V \lambda_1^i (f^i(\mathbf{x}) - u_{in}^i)^2 d\mathbf{x} \\ &+ \frac{1}{V} \int_{\text{outside } C} \sum_{i=1}^V \lambda_2^i (f^i(\mathbf{x}) - u_{out}^i)^2 d\mathbf{x} \\ &+ \alpha \text{Length}(C) \end{aligned}$$

where  $\mathbf{u}_{in}, \mathbf{u}_{out}, \lambda_1, \lambda_2 \in \mathbb{R}^V$ .

Equivalently to the **CV model** for scalar valued images, the edge-set  $C$  is represented by the zero-crossings of the level set function  $\phi$ . The level set formulation of the **CVec model** is then given by

$$\begin{aligned} E_{CVec}(\mathbf{u}_{in}, \mathbf{u}_{out}, \phi) &= \int_{\Omega} \sum_{i=1}^V \lambda_1^i (f^i(\mathbf{x}) - u_{in}^i)^2 H(\phi(\mathbf{x})) d\mathbf{x} \\ &+ \int_{\Omega} \sum_{i=1}^V \lambda_2^i (f^i(\mathbf{x}) - u_{out}^i)^2 (1 - H(\phi(\mathbf{x}))) d\mathbf{x} \quad (4.1) \\ &+ \alpha \int_{\Omega} |\nabla H(\phi(\mathbf{x}))| d\mathbf{x} \end{aligned}$$

where the Heaviside function  $H$  is used as indicator function for the set enclosed by the edge-set  $C$ . Similar to the scalar valued case, a minimiser of this energy functional is found by alternately updating the arithmetic means

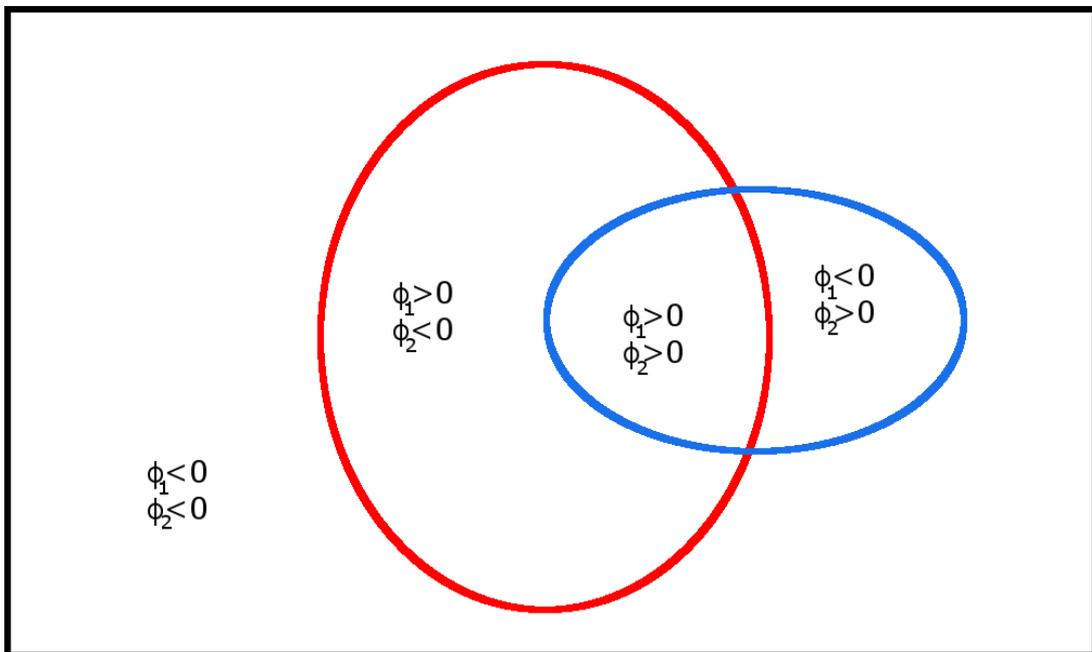
$$\begin{aligned} u_{in}^i(\phi) &= \frac{\int_{\Omega} f^i(\mathbf{x}) H(\phi(\mathbf{x})) d\mathbf{x}}{\int_{\Omega} H(\phi(\mathbf{x})) d\mathbf{x}} \\ u_{out}^i(\phi) &= \frac{\int_{\Omega} f^i(\mathbf{x}) (1 - H(\phi(\mathbf{x}))) d\mathbf{x}}{\int_{\Omega} (1 - H(\phi(\mathbf{x}))) d\mathbf{x}} \end{aligned} \quad (4.2)$$

and the level set function  $\phi$  [11]. Although this method is able to include information from different channels, it has a major shortcoming inherent in the use of one level set function. With just one level set function, the image can only be divided in fore-and background or two classes, separating the object from its surrounding area.

However, we are interested in segmenting a brain tumour. As we need therefore more than two classes, we make use of a further extension of the **CV model**, the *CV model for multiple level sets* [12].

## 4.2 Chan-Vese Active Contour Model with a Multiphase Level Set Representation

The basic idea of the **CV model** for multiple level sets is sketched in **Figure 4.1**. With  $m$  level sets, it is possible to divide the image in up to  $n = 2^m$  different classes.



**Figure 4.1:** Two curves given by the level set functions  $\phi_1$  and  $\phi_2$  partition the domain in four areas.

Hence, the energy functional of this method is given by

**Definition 51** (*CV model with a multiphase level set representation*)

$$E_{CVmult}^n(\mathbf{u}, \phi) = \sum_{K=1}^n \int_{\Omega} |f(\mathbf{x}) - u_K|^2 \chi_K \, d\mathbf{x} + \sum_{j=1}^m \alpha \int_{\Omega} |\nabla H(\phi_j(\mathbf{x}))| \, d\mathbf{x}$$

where  $n$  is the number of classes,  $m$  is the number of level sets,  $K = 1, \dots, n$  are the labels of the different classes,  $u_K$  is the average and  $\chi_K$  is the characteristic function of the class  $K$ , respectively [12, 14].

The used number of level sets is a trade-off between a detailed partitioning of the

image and an over-segmentation. We decided to use two level sets to partition the image. Hence, **Definition 51** becomes

$$E_{CVmult}^4(\mathbf{u}, \boldsymbol{\phi}) = \sum_{K=1}^4 \int_{\Omega} |f(\mathbf{x}) - u_K|^2 \chi_K \, d\mathbf{x} + \sum_{j=1}^2 \alpha \int_{\Omega} |\nabla H(\phi_j(\mathbf{x}))| \, d\mathbf{x} \quad (4.3)$$

where  $\mathbf{u} = (u_1, u_2, u_3, u_4)^\top$  and  $\boldsymbol{\phi} = (\phi_1, \phi_2)^\top$ . We follow [12, 14] and reformulate (4.3) to

$$\begin{aligned} E_{CVmult}^4(\mathbf{u}, \boldsymbol{\phi}) &= \int_{\Omega} |f(\mathbf{x}) - u_1|^2 H(\phi_1)H(\phi_2) \, d\mathbf{x} \\ &\quad + \int_{\Omega} |f(\mathbf{x}) - u_2|^2 H(\phi_1)(1 - H(\phi_2)) \, d\mathbf{x} \\ &\quad + \int_{\Omega} |f(\mathbf{x}) - u_3|^2 (1 - H(\phi_1))H(\phi_2) \, d\mathbf{x} \\ &\quad + \int_{\Omega} |f(\mathbf{x}) - u_4|^2 (1 - H(\phi_1))(1 - H(\phi_2)) \, d\mathbf{x} \\ &\quad + \alpha \int_{\Omega} |\nabla H(\phi_1(\mathbf{x}))| \, d\mathbf{x} \\ &\quad + \alpha \int_{\Omega} |\nabla H(\phi_2(\mathbf{x}))| \, d\mathbf{x} \end{aligned} \quad (4.4)$$

where  $\mathbf{u} = (u_1, u_2, u_3, u_4)^\top$  and  $\boldsymbol{\phi} = (\phi_1, \phi_2)^\top$ . Again, the energy functional is minimised by alternately updating the arithmetic means

$$\begin{aligned} u_1 &= \frac{\int_{\Omega} f(\mathbf{x})H(\phi_1(\mathbf{x}))H(\phi_2(\mathbf{x})) \, d\mathbf{x}}{\int_{\Omega} H(\phi_1(\mathbf{x}))H(\phi_2(\mathbf{x})) \, d\mathbf{x}} \\ u_2 &= \frac{\int_{\Omega} f(\mathbf{x})H(\phi_1(\mathbf{x}))(1 - H(\phi_2(\mathbf{x}))) \, d\mathbf{x}}{\int_{\Omega} H(\phi_1(\mathbf{x}))(1 - H(\phi_2(\mathbf{x}))) \, d\mathbf{x}} \\ u_3 &= \frac{\int_{\Omega} f(\mathbf{x})(1 - H(\phi_1(\mathbf{x})))H(\phi_2(\mathbf{x})) \, d\mathbf{x}}{\int_{\Omega} (1 - H(\phi_1(\mathbf{x})))H(\phi_2(\mathbf{x})) \, d\mathbf{x}} \\ u_4 &= \frac{\int_{\Omega} f(\mathbf{x})(1 - H(\phi_1(\mathbf{x})))(1 - H(\phi_2(\mathbf{x}))) \, d\mathbf{x}}{\int_{\Omega} (1 - H(\phi_1(\mathbf{x})))(1 - H(\phi_2(\mathbf{x}))) \, d\mathbf{x}} \end{aligned} \quad (4.5)$$

and the level set functions  $\phi_1$  and  $\phi_2$  [12, 14]

$$\begin{aligned}
\frac{\partial \phi_1}{\partial t} &= \delta_\epsilon \left[ \alpha \operatorname{div} \left( \frac{\nabla \phi_1}{|\nabla \phi_1|} \right) \right. \\
&\quad - ((f - u_1)^2 - (f - u_3)^2) H(\phi_2) \\
&\quad \left. - ((f - u_2)^2 - (f - u_4)^2) (1 - H(\phi_2)) \right] \\
\frac{\partial \phi_2}{\partial t} &= \delta_\epsilon \left[ \alpha \operatorname{div} \left( \frac{\nabla \phi_2}{|\nabla \phi_2|} \right) \right. \\
&\quad - ((f - u_1)^2 - (f - u_3)^2) H(\phi_1) \\
&\quad \left. - ((f - u_2)^2 - (f - u_4)^2) (1 - H(\phi_1)) \right].
\end{aligned} \tag{4.6}$$

This formulation allows for four segments, triple junctions and complex topologies. Thereby the partitioning cannot have overlaps and no vacuum can arise [12, 14].

We use a combination of both, the **CV model** for vector valued images and the **CV model** with a multiphase level set representation, to segment the tumour tissues.

### 4.3 Chan-Vese Active Contour Model for Vector Valued Images with a Multiphase Level Set Representation

**MRI** brain tumour images contain more than two objects. On the one hand, there is the skull and its background. On the other hand, there is also the tumour, which is located inside the skull. We also have more than one **MRI** sequence. As we register the different **MRI** scans before we segment the tumour, we can regard the **MRI** data we have as vector valued images.

We explain in **Section 4.1** and **Section 4.2** two extensions of the **CV model**, cf. **Section 3.4**. The first one, the **CV model** for vector valued images, allows us to include all information available in the different channels. The second one, the **CV model** with a multiphase level set representation, is capable to partition the image in more than two segments.

Combining both extensions results in the formulation

**Definition 52** (*CVec model with a Multiphase Level Set Representation*)

$$E_{CVecMult}^{n,V}(\mathbf{u}, \phi) = \sum_{K=1}^n \sum_{i=1}^V \lambda^i \int_{\Omega} |f^i(\mathbf{x}) - u_K^i|^2 \chi_K \, d\mathbf{x} \\ + \sum_{j=1}^m \alpha \int_{\Omega} |\nabla H(\phi_j(\mathbf{x}))| \, d\mathbf{x}.$$

where we changed the formulation suggested by [12, 14] to take care of the importance of channels. We added for this reason the weight  $\lambda^i$  for channel  $f^i$ ,  $i = 1, \dots, V$ .

This method has  $m$  level set functions partitioning the image in  $n$  classes. Thereby, information from  $V$  channels is used to minimise the functional. We have in our application at most four different channels:  $T_1$ ,  $T_1^{cont}$ ,  $T_2$  and  $T_2^{flair}$ . We decided furthermore to use two level set functions.

Then **Definition 52** becomes

$$E_{CVecMult}^{4,4}(\mathbf{u}, \phi) = \sum_{K=1}^4 \sum_{i=1}^4 \lambda^i \int_{\Omega} |f^i(\mathbf{x}) - u_K^i|^2 \chi_K \, d\mathbf{x} \\ + \sum_{j=1}^2 \alpha \int_{\Omega} |\nabla H(\phi_j(\mathbf{x}))| \, d\mathbf{x}. \quad (4.7)$$

We reformulate this to

$$E_{CVecMult}^{4,4}(\mathbf{u}, \phi) = \sum_{i=1}^4 \lambda^i \int_{\Omega} |f^i(\mathbf{x}) - u_1^i|^2 H(\phi_1)H(\phi_2) \, d\mathbf{x} \\ + \sum_{i=1}^4 \lambda^i \int_{\Omega} |f^i(\mathbf{x}) - u_2^i|^2 H(\phi_1)(1 - H(\phi_2)) \, d\mathbf{x} \\ + \sum_{i=1}^4 \lambda^i \int_{\Omega} |f^i(\mathbf{x}) - u_3^i|^2 (1 - H(\phi_1))H(\phi_2) \, d\mathbf{x} \\ + \sum_{i=1}^4 \lambda^i \int_{\Omega} |f^i(\mathbf{x}) - u_4^i|^2 (1 - H(\phi_1))(1 - H(\phi_2)) \, d\mathbf{x} \\ + \alpha \int_{\Omega} |\nabla H(\phi_1(\mathbf{x}))| \, d\mathbf{x} + \alpha \int_{\Omega} |\nabla H(\phi_2(\mathbf{x}))| \, d\mathbf{x}. \quad (4.8)$$

This functional is minimised by alternately updating the arithmetic means of the

enclosed areas for all channels  $\forall i = 1, \dots, 4$  as

$$u_1^i = \frac{\int_{\Omega} f^i(\mathbf{x}) H(\phi_1(\mathbf{x})) H(\phi_2(\mathbf{x})) \, d\mathbf{x}}{\int_{\Omega} H(\phi_1(\mathbf{x})) H(\phi_2(\mathbf{x})) \, d\mathbf{x}}$$

$$u_2^i = \frac{\int_{\Omega} f^i(\mathbf{x}) H(\phi_1(\mathbf{x})) (1 - H(\phi_2(\mathbf{x}))) \, d\mathbf{x}}{\int_{\Omega} H(\phi_1(\mathbf{x})) (1 - H(\phi_2(\mathbf{x}))) \, d\mathbf{x}}$$

$$u_3^i = \frac{\int_{\Omega} f^i(\mathbf{x}) (1 - H(\phi_1(\mathbf{x}))) H(\phi_2(\mathbf{x})) \, d\mathbf{x}}{\int_{\Omega} (1 - H(\phi_1(\mathbf{x}))) H(\phi_2(\mathbf{x})) \, d\mathbf{x}}$$

$$u_4^i = \frac{\int_{\Omega} f^i(\mathbf{x}) (1 - H(\phi_1(\mathbf{x}))) (1 - H(\phi_2(\mathbf{x}))) \, d\mathbf{x}}{\int_{\Omega} (1 - H(\phi_1(\mathbf{x}))) (1 - H(\phi_2(\mathbf{x}))) \, d\mathbf{x}}$$

and  $\phi = (\phi_1, \phi_2)^\top$  [12]. This energy is non-convex and has to be solved numerically. We use a gradient descent and regularise the Heaviside function  $H$  the same way as for the ordinary CV model, cf. Section 3.4. We deduce the Euler-Lagrange equations of  $\nabla E_{CVecMult}^{4,4}$

**Definition 53 (Euler-Lagrange Equations of  $\nabla E_{CVecMult}^{4,4}$ )**

$$H'_\varepsilon(\phi_1(\mathbf{x})) \left[ \alpha \operatorname{div} \left( \frac{\nabla \phi_1}{|\nabla \phi_1|} \right) - \sum_{i=1}^4 \lambda^i ((f^i(\mathbf{x}) - u_1^i)^2 - (f^i(\mathbf{x}) - u_3^i)^2) H(\phi_2) \right. \\ \left. - \sum_{i=1}^4 \lambda^i ((f^i(\mathbf{x}) - u_2^i)^2 - (f^i(\mathbf{x}) - u_4^i)^2) (1 - H(\phi_2)) \right] = 0$$

$$H'_\varepsilon(\phi_2(\mathbf{x})) \left[ \alpha \operatorname{div} \left( \frac{\nabla \phi_2}{|\nabla \phi_2|} \right) - \sum_{i=1}^4 \lambda^i ((f^i(\mathbf{x}) - u_1^i)^2 - (f^i(\mathbf{x}) - u_3^i)^2) H(\phi_1) \right. \\ \left. - \sum_{i=1}^4 \lambda^i ((f^i(\mathbf{x}) - u_2^i)^2 - (f^i(\mathbf{x}) - u_4^i)^2) (1 - H(\phi_1)) \right] = 0$$

with the boundary conditions

$$\frac{\partial_\varepsilon(\phi_1(\mathbf{x}))}{|\nabla \phi_1(\mathbf{x})|} \frac{\partial \phi_1(\mathbf{x})}{\partial \mathbf{n}(\mathbf{x})} = 0 \quad \mathbf{x} \in \partial\Omega$$

$$\frac{\partial_\varepsilon(\phi_2(\mathbf{x}))}{|\nabla \phi_2(\mathbf{x})|} \frac{\partial \phi_2(\mathbf{x})}{\partial \mathbf{n}(\mathbf{x})} = 0 \quad \mathbf{x} \in \partial\Omega$$

The gradient descent w.r.t.  $\boldsymbol{\phi} = (\phi_1, \phi_2)^\top$  is then given by

**Definition 54** (*Gradient Descent of  $\nabla E_{CVecMult}^{4,4}$* )

$$\begin{aligned}\partial_t \phi_1(\mathbf{x}) &= \delta_\varepsilon(\phi_1(\mathbf{x})) \left[ \alpha \operatorname{div} \left( \frac{\nabla \phi_1(\mathbf{x})}{|\nabla \phi_1(\mathbf{x})|} \right) \right. \\ &\quad - \sum_{i=1}^4 \lambda^i ((f^i(\mathbf{x}) - u_1^i)^2 - (f^i(\mathbf{x}) - u_3^i)^2) H(\phi_2(\mathbf{x})) \\ &\quad \left. - \sum_{i=1}^4 \lambda^i ((f^i(\mathbf{x}) - u_2^i)^2 - (f^i(\mathbf{x}) - u_4^i)^2) (1 - H(\phi_2(\mathbf{x}))) \right] \\ \partial_t \phi_2(\mathbf{x}) &= \delta_\varepsilon(\phi_2(\mathbf{x})) \left[ \alpha \operatorname{div} \left( \frac{\nabla \phi_2(\mathbf{x})}{|\nabla \phi_2(\mathbf{x})|} \right) \right. \\ &\quad - \sum_{i=1}^4 \lambda^i ((f^i(\mathbf{x}) - u_1^i)^2 - (f^i(\mathbf{x}) - u_3^i)^2) H(\phi_1(\mathbf{x})) \\ &\quad \left. - \sum_{i=1}^4 \lambda^i ((f^i(\mathbf{x}) - u_2^i)^2 - (f^i(\mathbf{x}) - u_4^i)^2) (1 - H(\phi_1(\mathbf{x}))) \right]\end{aligned}$$

with the boundary conditions

$$\begin{aligned}\phi_1(\mathbf{x}, 0) &= \phi_{1,0}(\mathbf{x}), & \phi_2(\mathbf{x}, 0) &= \phi_{2,0}(\mathbf{x}) & \mathbf{x} &\in \Omega \\ \frac{\partial_\varepsilon(\phi_1(\mathbf{x}))}{|\nabla \phi_1(\mathbf{x})|} \frac{\partial \phi_1(\mathbf{x})}{\partial \mathbf{n}(\mathbf{x})} &= 0 & \frac{\partial_\varepsilon(\phi_2(\mathbf{x}))}{|\nabla \phi_2(\mathbf{x})|} \frac{\partial \phi_2(\mathbf{x})}{\partial \mathbf{n}(\mathbf{x})} &= 0 & \mathbf{x} &\in \partial\Omega\end{aligned}$$

We follow [13] and discretise (54) with finite differences as denoted by **Equation 4.9** and **Equation 4.10**, cf. **Section 1.2.5**.

$$\begin{aligned}\frac{\phi_{1,i,j}^{n+1} - \phi_{1,i,j}^n}{\tau} &= \delta_\varepsilon(\phi_{1,i,j}^n) \left[ - \sum_{i=1}^4 \lambda^i ((f_{i,j}^i - u_1^i)^2 - (f_{i,j}^i - u_3^i)^2) H(\phi_{2,i,j}) \right. \\ &\quad \left. - \sum_{i=1}^4 \lambda^i ((f_{i,j}^i - u_2^i)^2 - (f_{i,j}^i - u_4^i)^2) (1 - H(\phi_{2,i,j})) \right. \\ &\quad \left. + \alpha \left( \partial_x^- \left( \frac{\partial_x^+ \phi_{1,i,j}^{n+1}}{\sqrt{(\partial_x^+ \phi_{1,i,j}^n)^2 + (\partial_y \phi_{1,i,j}^n)^2}} \right) \right. \right. \\ &\quad \left. \left. + \partial_y^- \left( \frac{\partial_y^+ \phi_{1,i,j}^{n+1}}{\sqrt{(\partial_y^+ \phi_{1,i,j}^n)^2 + (\partial_x \phi_{1,i,j}^n)^2}} \right) \right) \right] \quad (4.9)\end{aligned}$$

$$\begin{aligned}
\frac{\phi_{2,i,j}^{n+1} - \phi_{2,i,j}^n}{\tau} = \delta_\varepsilon(\phi_{2,i,j}^n) & \left[ - \sum_{i=1}^4 \lambda^i ((f_{i,j}^i - u_1^i)^2 - (f_{i,j}^i - u_3^i)^2) H(\phi_{1,i,j}) \right. \\
& - \sum_{i=1}^4 \lambda^i ((f_{i,j}^i - u_2^i)^2 - (f_{i,j}^i - u_4^i)^2) (1 - H(\phi_{1,i,j})) \\
& \left. + \alpha \left( \partial_x^- \left( \frac{\partial_x^+ \phi_{2,i,j}^{n+1}}{\sqrt{(\partial_x^+ \phi_{2,i,j}^n)^2 + (\partial_y \phi_{2,i,j}^n)^2}} \right) \right. \right. \\
& \left. \left. + \partial_y^- \left( \frac{\partial_y^+ \phi_{2,i,j}^{n+1}}{\sqrt{(\partial_y^+ \phi_{2,i,j}^n)^2 + (\partial_x \phi_{2,i,j}^n)^2}} \right) \right) \right] \quad (4.10)
\end{aligned}$$

where  $\tau$  is the time step [12, 14]. Let

$$\begin{aligned}
C_{1,m,i,j} &= \frac{\alpha}{\sqrt{(\partial_x^+ \phi_{m,i,j}^n)^2 + (\partial_y \phi_{m,i,j}^n)^2}} & C_{2,m,i,j} &= \frac{\alpha}{\sqrt{(\partial_x^+ \phi_{m,i-1,j}^n)^2 + (\partial_y \phi_{m,i-1,j}^n)^2}} \\
C_{3,m,i,j} &= \frac{\alpha}{\sqrt{(\partial_y^+ \phi_{m,i,j}^n)^2 + (\partial_x \phi_{m,i,j}^n)^2}} & C_{4,m,i,j} &= \frac{\alpha}{\sqrt{(\partial_y^+ \phi_{m,i,j-1}^n)^2 + (\partial_x \phi_{m,i,j-1}^n)^2}}
\end{aligned}$$

and

$$\begin{aligned}
R_{m,i,j} = & - \left( \sum_{i=1}^4 \lambda^i ((f_{i,j}^i - u_1^i)^2 - (f_{i,j}^i - u_3^i)^2) H(\phi_{m,i,j}) \right. \\
& \left. + \sum_{i=1}^4 \lambda^i ((f_{i,j}^i - u_2^i)^2 - (f_{i,j}^i - u_4^i)^2) (1 - H(\phi_{m,i,j})) \right).
\end{aligned}$$

Then we can write (4.9) and (4.10) as

$$\begin{aligned}
\frac{\phi_{1,i,j}^{n+1} - \phi_{1,i,j}^n}{\tau} &= \delta_\varepsilon(\phi_{1,i,j}^n) \left[ R_{2,i,j} + C_{1,1,i,j}(\phi_{1,i+1,j}^n - \phi_{1,i,j}^{n+1}) - C_{2,1,i,j}(\phi_{1,i,j}^{n+1} - \phi_{1,i-1,j}^n) \right. \\
& \left. + C_{3,1,i,j}(\phi_{1,i,j+1}^n - \phi_{1,i,j}^{n+1}) - C_{4,1,i,j}(\phi_{1,i,j}^{n+1} - \phi_{1,i,j-1}^n) \right] \\
\frac{\phi_{2,i,j}^{n+1} - \phi_{2,i,j}^n}{\tau} &= \delta_\varepsilon(\phi_{2,i,j}^n) \left[ R_{1,i,j} + C_{1,2,i,j}(\phi_{2,i+1,j}^n - \phi_{2,i,j}^{n+1}) - C_{2,2,i,j}(\phi_{2,i,j}^{n+1} - \phi_{2,i-1,j}^n) \right. \\
& \left. + C_{3,2,i,j}(\phi_{2,i,j+1}^n - \phi_{2,i,j}^{n+1}) - C_{4,2,i,j}(\phi_{2,i,j}^{n+1} - \phi_{2,i,j-1}^n) \right]
\end{aligned}$$

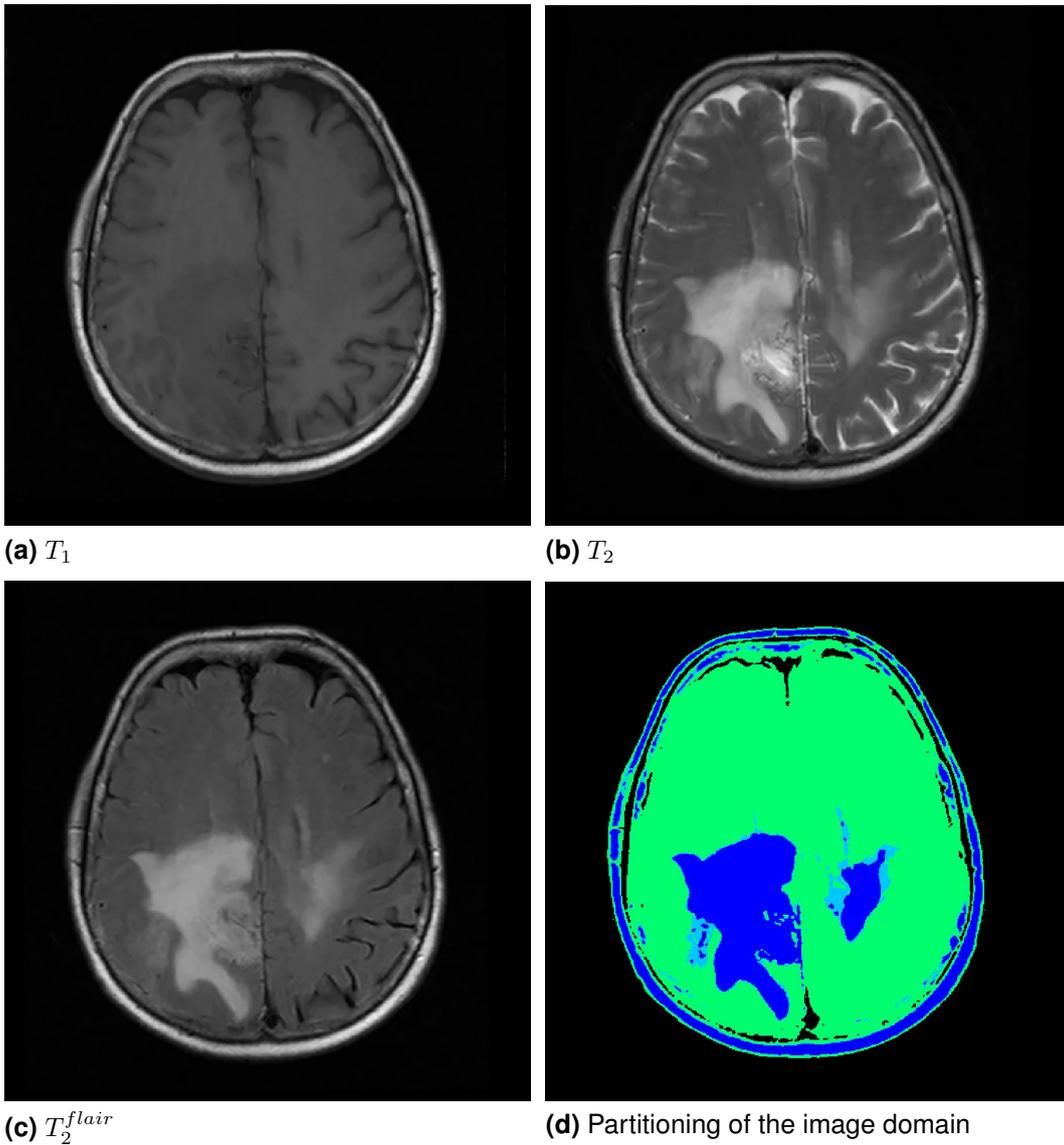
Finally, we update  $\phi_1$  as

$$\begin{aligned}
\phi_{1,i,j}^{n+1} &= \frac{\phi_{1,i,j}^n}{1 + \tau \delta_\varepsilon(\phi_{1,i,j}^n) (C_{1,1,i,j} + C_{2,1,i,j} + C_{3,1,i,j} + C_{4,1,i,j})} + \\
& \frac{\tau \delta_\varepsilon(\phi_{1,i,j}^n) \left[ R_{2,i,j} + C_{1,1,i,j} \phi_{1,i+1,j}^n + C_{2,1,i,j} \phi_{1,i-1,j}^{n+1} + C_{3,1,i,j} \phi_{1,i,j+1}^n + C_{4,1,i,j} \phi_{1,i,j-1}^{n+1} \right]}{1 + \tau \delta_\varepsilon(\phi_{1,i,j}^n) (C_{1,1,i,j} + C_{2,1,i,j} + C_{3,1,i,j} + C_{4,1,i,j})}
\end{aligned}$$

and  $\phi_2$  as

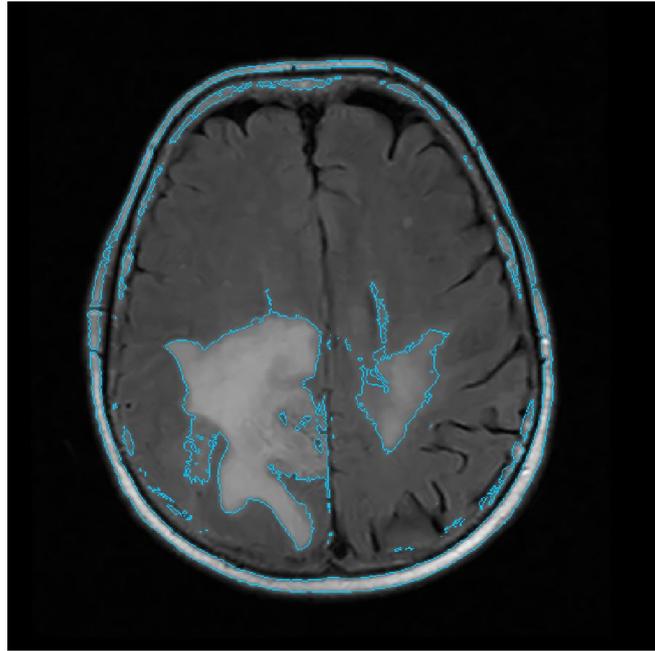
$$\phi_{2,i,j}^{n+1} = \frac{\phi_{2,i,j}^n}{1 + \tau\delta_\varepsilon(\phi_{2,i,j}^n)(C_{1,2,i,j} + C_{2,2,i,j} + C_{3,2,i,j} + C_{4,2,i,j})} + \frac{\tau\delta_\varepsilon(\phi_{2,i,j}^n) [R_{1,i,j} + C_{1,2,i,j}\phi_{2,i+1,j}^n + C_{2,2,i,j}\phi_{2,i-1,j}^{n+1} + C_{3,2,i,j}\phi_{2,i,j+1}^n + C_{4,2,i,j}\phi_{2,i,j-1}^{n+1}]}{1 + \tau\delta_\varepsilon(\phi_{2,i,j}^n)(C_{1,2,i,j} + C_{2,2,i,j} + C_{3,2,i,j} + C_{4,2,i,j})}$$

The variance of the intensity ranges of the different channels can be very high. We align therefore the grey values to the interval  $[0, 255]$ . The evolution under the **CVec model** with a multiphase level set representation is shown in **Figure 4.2**.



**Figure 4.2:** Evolution under the **CVec model** with a multiphase level set representation. (a)-(c) denoised and registered input slices. (d) Partitioning of the image domain.  $\alpha = 1$ ,  $\epsilon = h_x h_y$ ,  $\tau \approx 0.04$ ,  $\lambda^{T_1} = \lambda^{T_2} = 1$ ,  $\lambda^{T_2^{flair}} = 2$ .

**Figure 4.2** shows the 11th slice of all MRI sequences for subject G-25 and the detected boundary of the tumour tissue, i.e. the oedema. **Figure 4.2a**, **Figure 4.2b** and **Figure 4.2c** are the different channels of the vector valued input image. **Figure 4.2d** displays the partitioning of the image domain while each colour depicts one class. The tumour boundaries are clearly visible although there are some parts of the skull, which do not belong to the tumour but are in the same class. We will name this in the latter *false positive*.



**Figure 4.3:** Tumour boundaries in the  $T_2^{flair}$  data detected by the **CVec model** with a multi-phase level set representation (steady state).

**Figure 4.3** shows the detected tumour boundary in the  $T_2$  data. The oedema is clearly visible while most of the high signal of the cerebrospinal fluid is not present therein. Only areas having high intensities in most of the image channels are assigned to the this class. Consequently, the tumour but also the skull are assigned to the same class as the tumour tissue.

The oedematous tissue has brighter and darker areas. Therefore, two classes contain oedematous areas. We show in the latter mostly the determined boundary and neglect the visualisation of the classes.

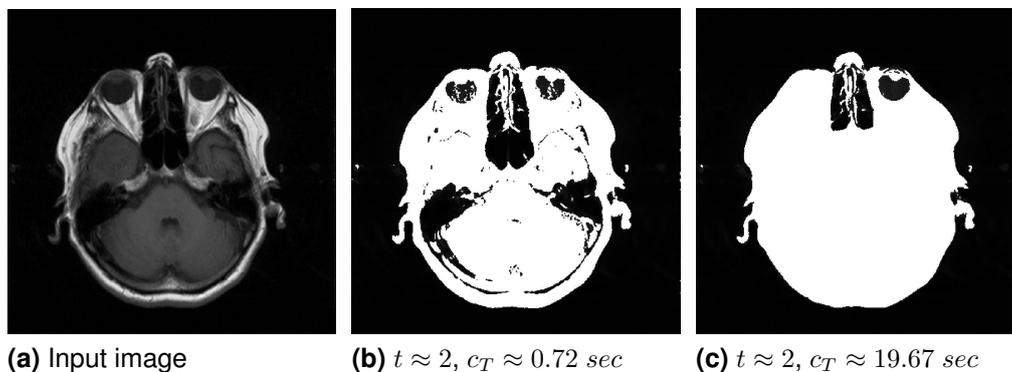
## 5 Parameter Selection

We use several methods to denoise, register and segment the **MRI** data. We explain in the following how the parameters for these methods can be chosen.

### 5.1 Non-Local Means

We use **NLM** means to denoise the **MRI** sequences. As it is important for denoising medical images to lose as few details as possible, the parameters of **NLM** means have to be chosen carefully.

The most important parameter is the standard deviation  $\sigma_r$  of the Gaussian weighting function as it influences which neighbourhoods are regarded as similar, cf. **Section 2.3.2**. Additionally the choice of this parameter is also the most tedious part.

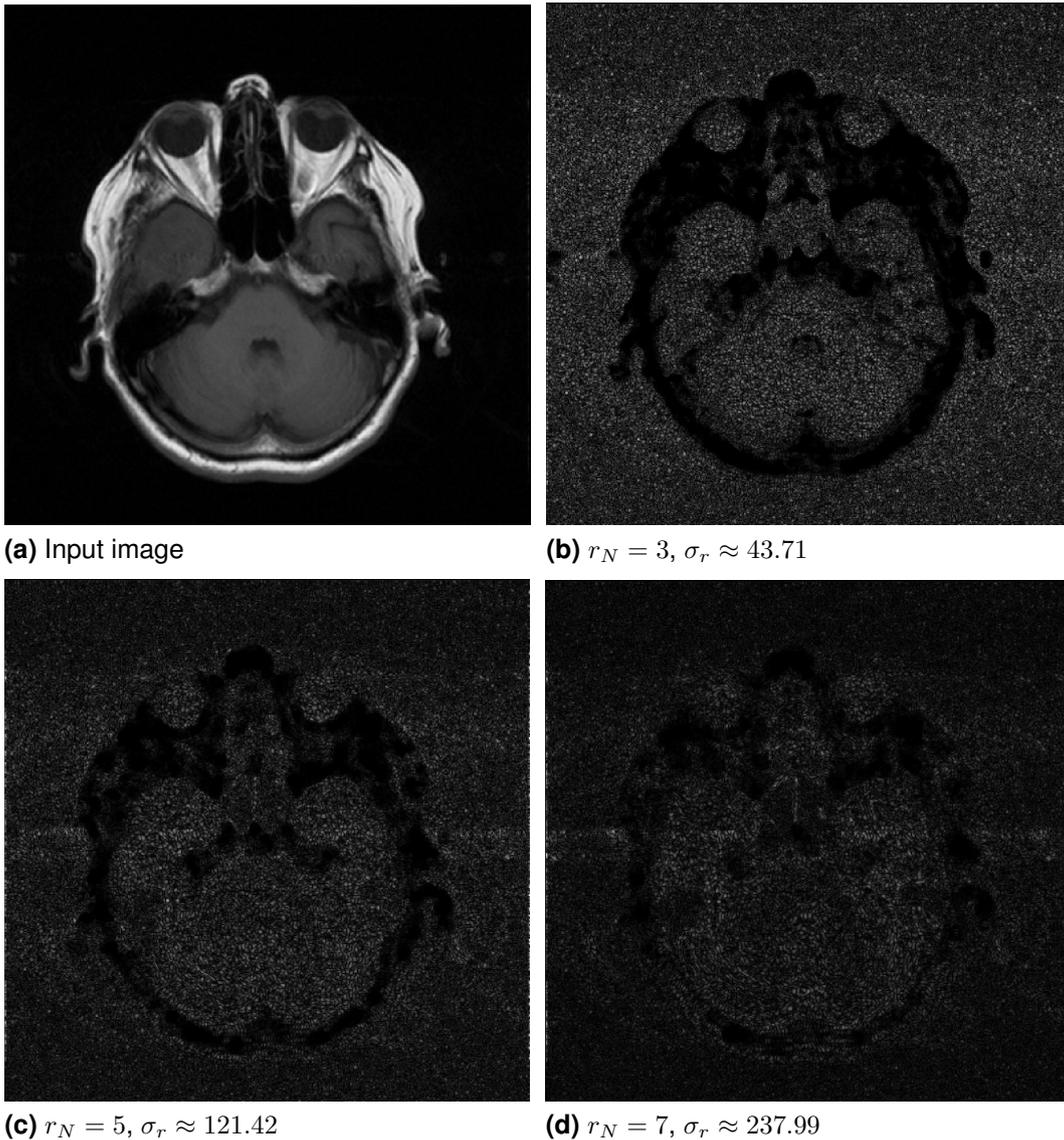


**Figure 5.1:** Steady states of an evolution under the **CV** model (b) without and (c) with reinitialisation of the level set function  $\phi$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 1$ ,  $\mu = 0$ ,  $\alpha = 1$ ,  $\epsilon = h_x h_y$ ,  $\tau \approx 0.04$ .  $c_T$ : computation time.

To overcome this difficulty we split the image in fore- and background with the **CV** model, see **Section 3.4**. Afterwards we compute the standard deviation  $\sigma_B$  of the background and use this as input for the **NLM** means [32].

The resulting segmentation is shown in **Figure 5.1**. The reinitialisation of the level set function  $\phi$  acts as a rescaling and prevents interior contours from growing [11].

We prefer a detailed splitting of the image in fore- and background. The reinitialisation is also the most time consuming step in the **CV model**. For this reason, we do not reinitialise the level set function when we compute the background of an image to determine the standard deviation. As experience has shown,  $\sigma_B$  fits best to a similarity window with  $r_N = 3$ . We adapt therefore  $\sigma_r$  to the area enclosed by the similarity window  $N$  by  $\sigma_r = (\frac{r_N}{3})^2 \sigma_B$ .

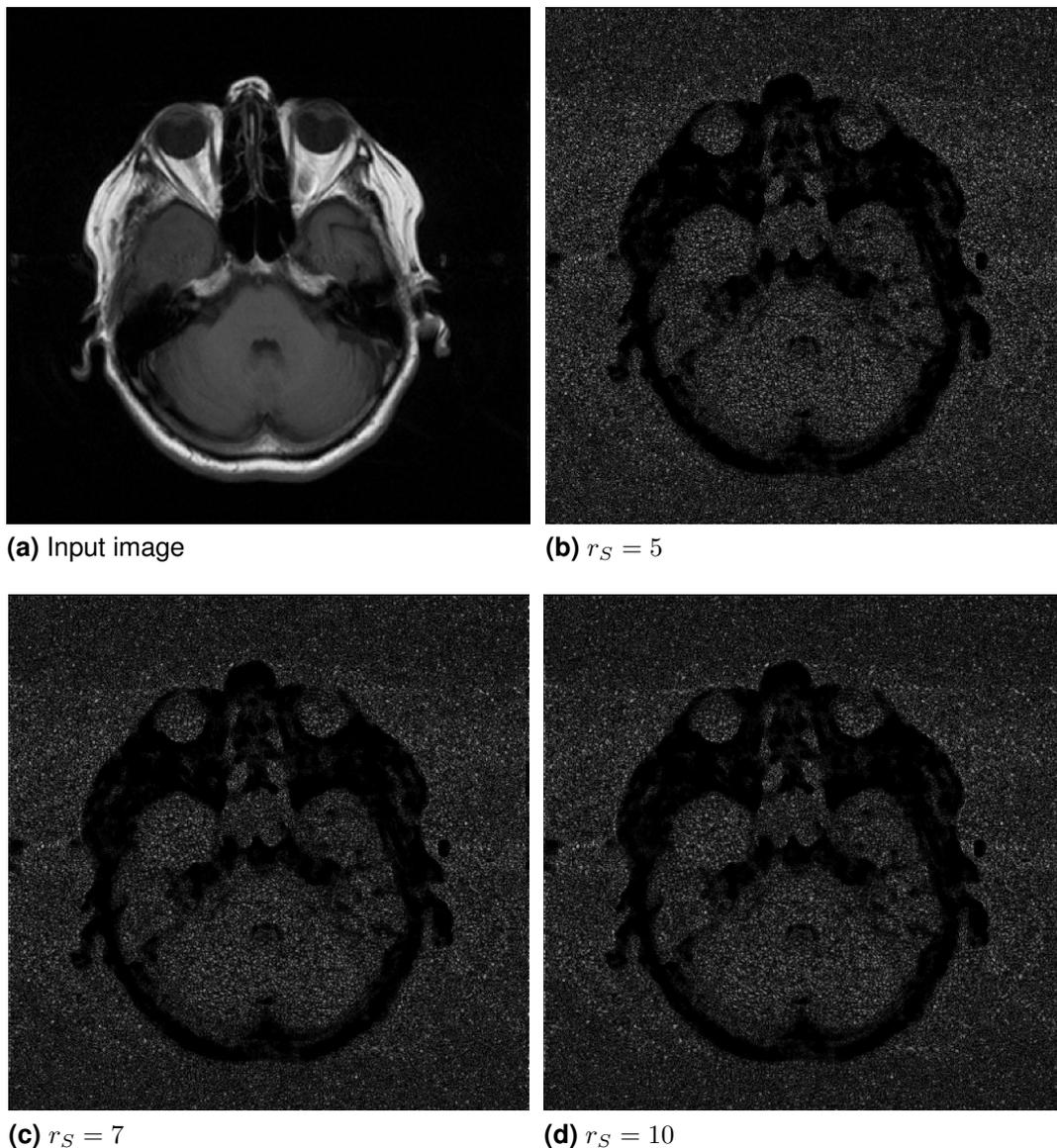


**Figure 5.2:** Method noise of **NLMMeans** for different radii  $r_N$  ( $r_S = 5$ ). The method noise is gamma corrected,  $\gamma = 1.2$ .

The second important parameter is the size of the similarity window  $r_N$ . It has to be large enough to be robust even in the presence of noise. On the other hand

it has to be sufficient small to take care about details. The noise guessed by **NLM** means for different radii  $r_N$  is shown in **Figure 5.2**. The method noise in **Figure 5.2b** is very similar to white noise while small details, for example structures near the eyes or inside the skull, are not present therein. We observe in **Figure 5.2c** and **Figure 5.2d** that for larger patches more and more details are removed. The radius of the regarded patches has furthermore a tremendous effect on the computation time, cf. **Table 5.1**. In our opinion, setting the radius  $r_N = 3$  is for this reason a good choice.

The last parameter to be set is the radius  $r_S$  of the search window  $S$ . As depicted in **Figure 5.3**, **NLM** means is very tolerant against changes of this parameter.



**Figure 5.3:** Method noise of **NLM** means for different radii  $r_S$  ( $r_N = 3$ ,  $\sigma_r \approx 43.71$ ). The method noise is gamma corrected,  $\gamma = 1.2$ .

Regarding **Figure 5.3b** and **Figure 5.3d**, there is nearly no difference observable. The main difference of  $r_S = 5$  and  $r_S = 10$  is the computation time, cf. **Table 5.1**.

---

	$r_S = 5$	$r_S = 7$	$r_S = 10$
$r_N = 3$	3.29	6.22	12.28
$r_N = 5$	—	15.01	29.68
$r_N = 7$	—	—	55.59

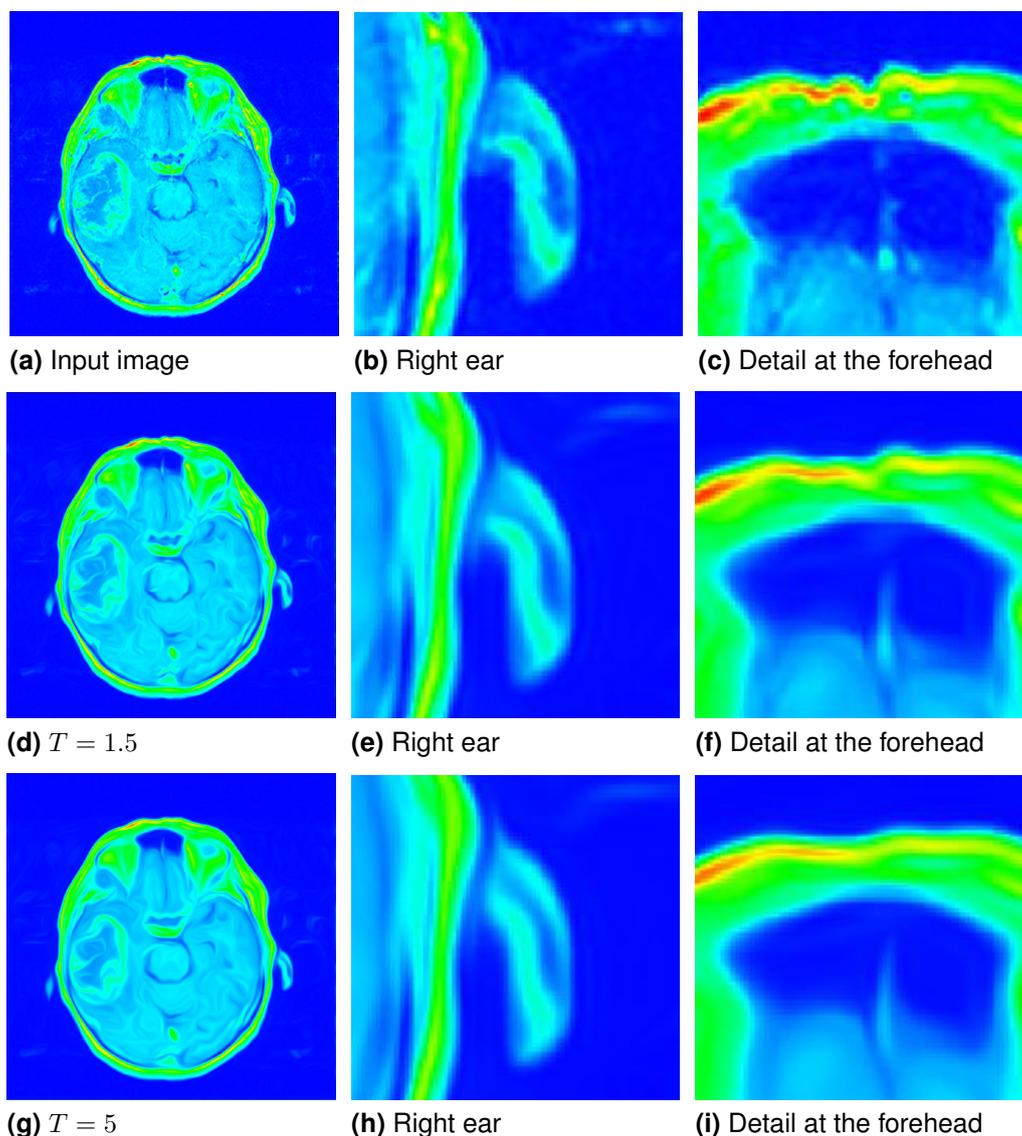
**Table 5.1:** Runtime of **NLMeans** for different parameter settings in seconds ( $r_N$ : Similarity window,  $r_S$ : Search window).

We suggest consequently a parameter setting with  $r_N = 3$ ,  $r_S = 5$  while  $\sigma_r$  is automatically adopted to the variance of the background.

---

## 5.2 Coherence Enhancing Diffusion

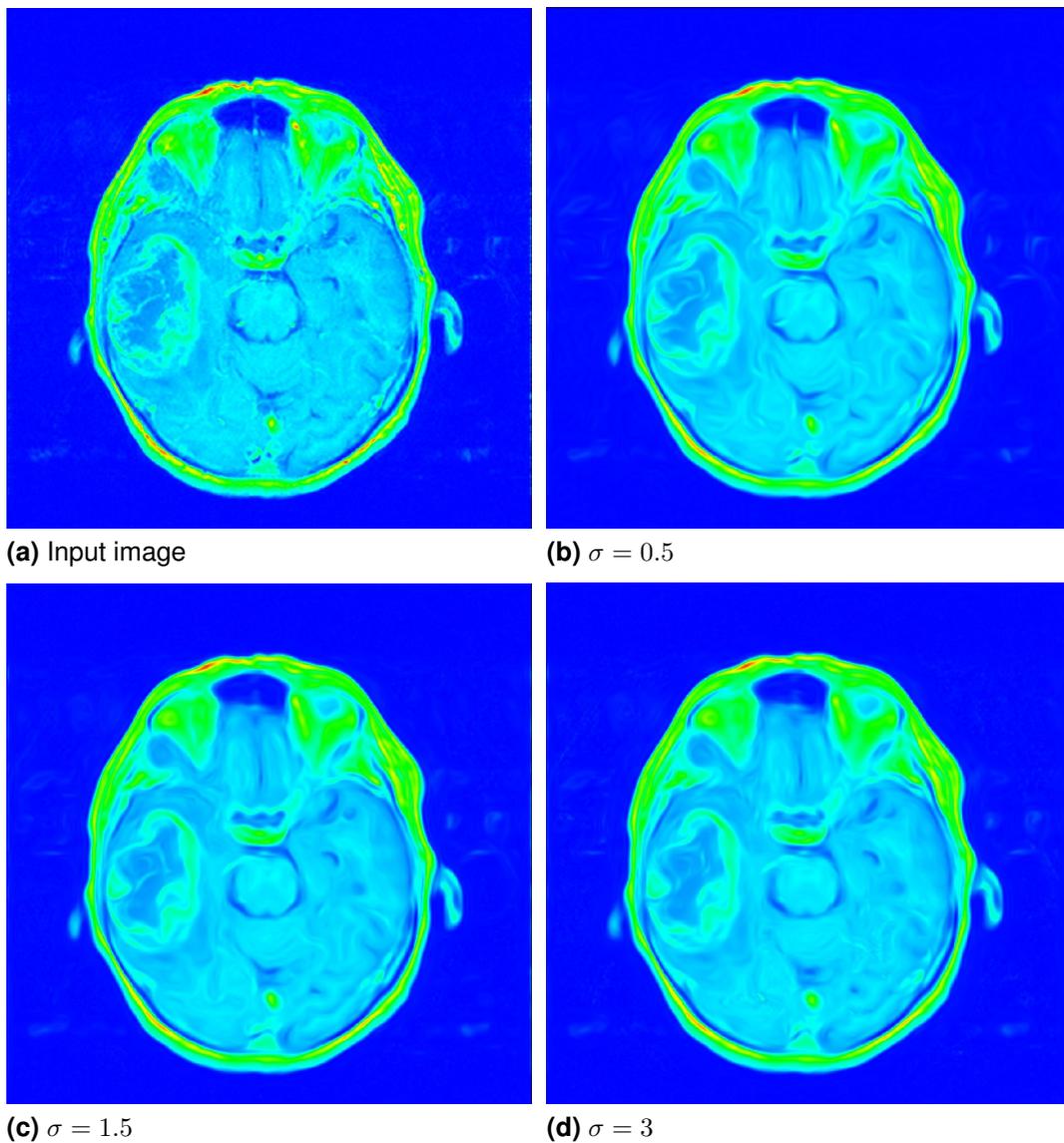
Our spatial registration is based on the outer contour of the skull. Consequently, the registration is of low quality when this shape varies for different MRI sequences. As the ears can for this reason disturb our registration process, we have to separate them from the skull. We overcome this problem by preprocessing affected images before we split them into fore- and background with the CV model. We do this with a short evolution under the CED [63]. CED has several parameters. We show in the following how those can be set to disconnect the ears. The most important parameter is the evolution time  $T$ .



**Figure 5.4:** Evolution under CED for different evolution times.  $\sigma = 0.5$ ,  $\rho = 1.5$ ,  $\alpha = 0.001$ ,  $C = 1$ .

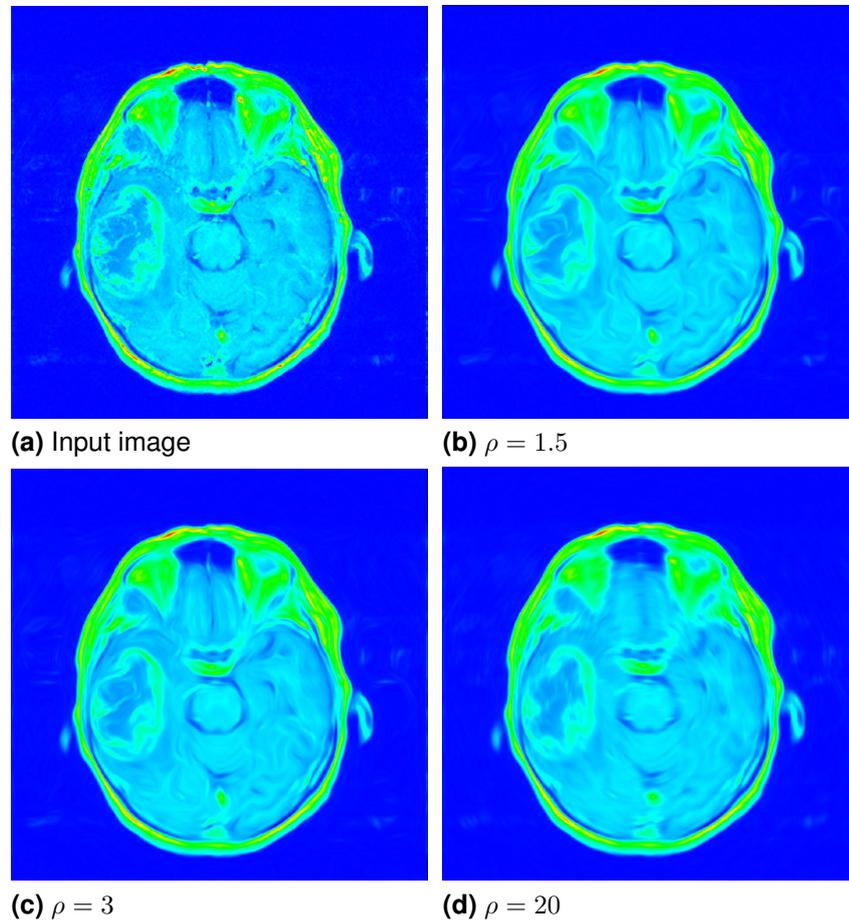
The evolution under the CED for different evolution times  $T$  is shown in **Figure 5.4**. After a short evolution, the connection between the ear and the skull is smeared out, depicted in **Figure 5.4e**. Also small details are smoothed but they remain in the image, cf. **Figure 5.4f**. If the evolution is too long, small details are completely removed, **Figure 5.4i**.

The parameter  $\sigma$  denotes the noise scale [63]. For larger values of  $\sigma$ , more noise and small-scale details are removed. In general, it should be chosen very small as it cancels adjacent gradients having the same direction, but opposite orientation [63]. However, we do not have parallel structures at the outer shape of the skull. Thus, a not accurate parameter setting of  $\sigma$  does not have a big influence, as shown in **Figure 5.5**.



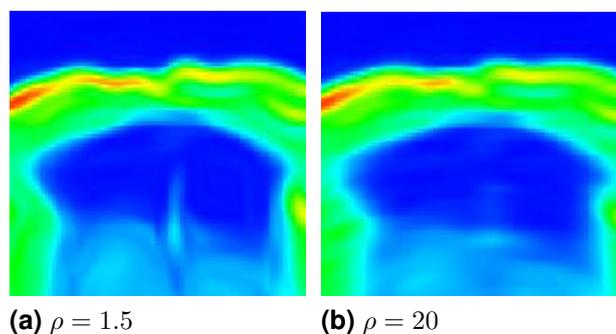
**Figure 5.5:** Evolution under CED for different noise scales.  $T = 1.5$ ,  $\rho = 1.5$ ,  $\alpha = 0.001$ ,  $C = 1$ .

The integration scale  $\rho$  averages directional information [63]. The images that we preprocess to disconnect outer parts have a closed contour. Hence, this parameter setting is also not critical, see **Figure 5.6**.

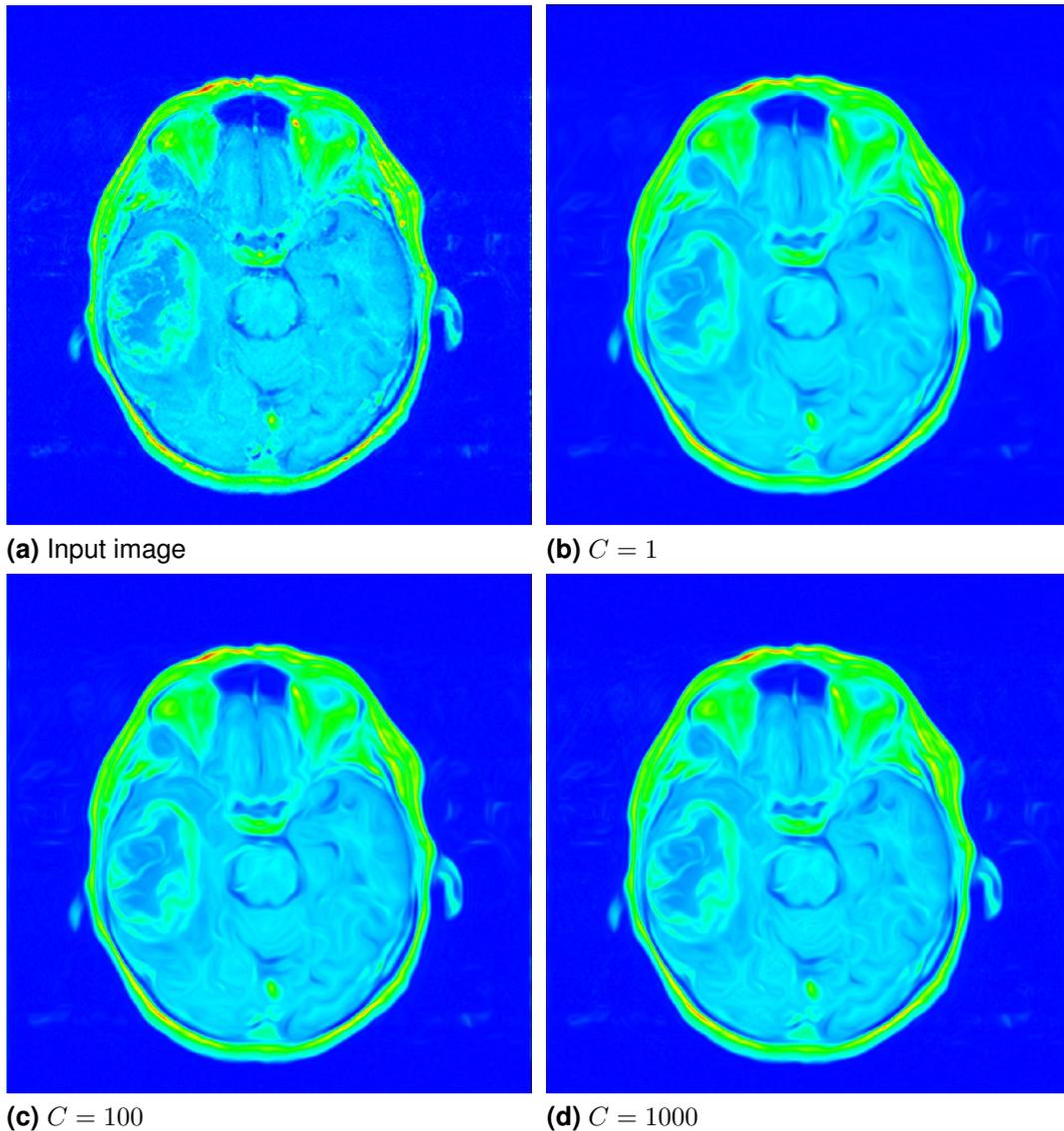


**Figure 5.6:** Evolution under CED for different integration scales.  $T = 1.5$ ,  $\sigma = 0.5$ ,  $\alpha = 0.001$ ,  $C = 1$ .

However, if the chosen value for  $\rho$  is much too large, CED removes details in the contour of the skull. This is observable in **Figure 5.7**.



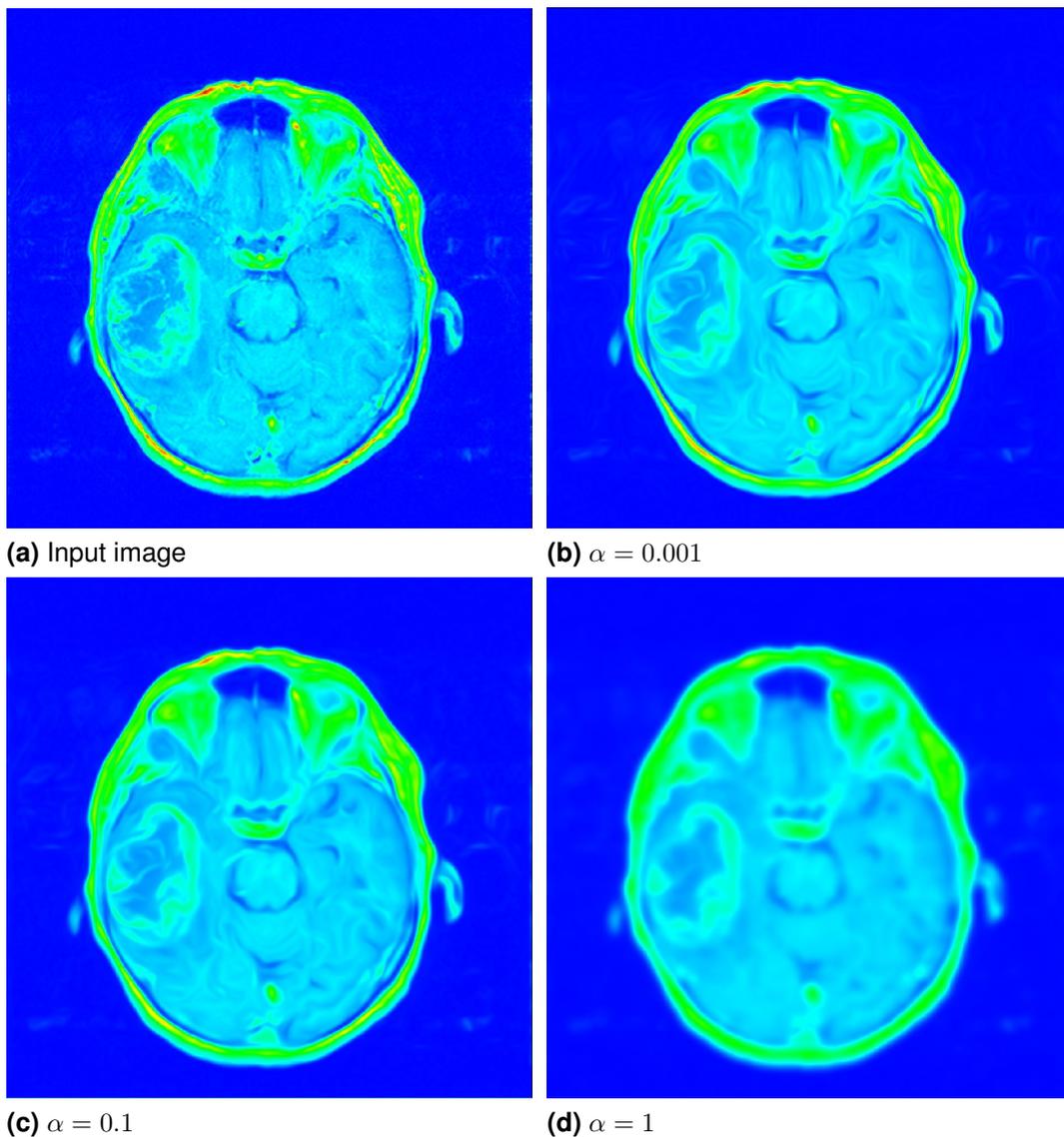
**Figure 5.7:** Effect of averaging too much directional information.



**Figure 5.8:** Evolution under CED for different threshold parameters.  $T = 1.5$ ,  $\sigma = 0.5$ ,  $\rho = 1.5$ ,  $\alpha = 0.001$ .

The parameter  $C$  acts as a threshold for the strength of the local orientation [63]. If the local coherence is smaller than  $C$ , the diffusion is more homogeneous. Comparing **Figure 5.8b** and **Figure 5.8d**, one can observe that in **Figure 5.8d** the brain is more smoothed and less structures are present therein.

However, the orientation at the outer contour of the skull is always high, i.e.  $\kappa \gg C$ . The CED filter acts mainly in the coherence-orientation, therefore the outer contour is not disturbed for high threshold values.



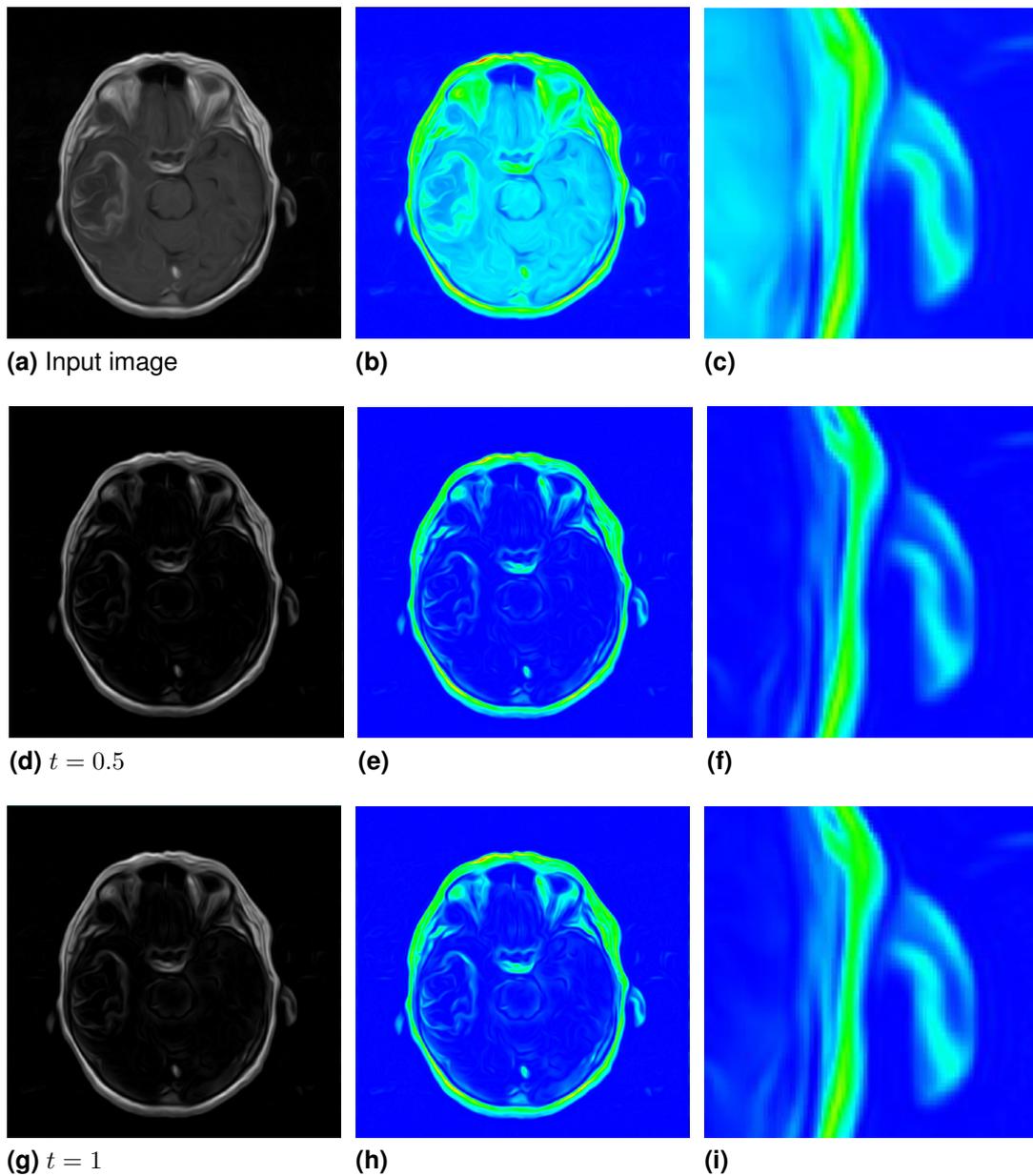
**Figure 5.9:** Evolution under CED for different regularisation parameters.  $T = 1.5$ ,  $\sigma = 0.5$ ,  $\rho = 1.5$ ,  $C = 1$ .

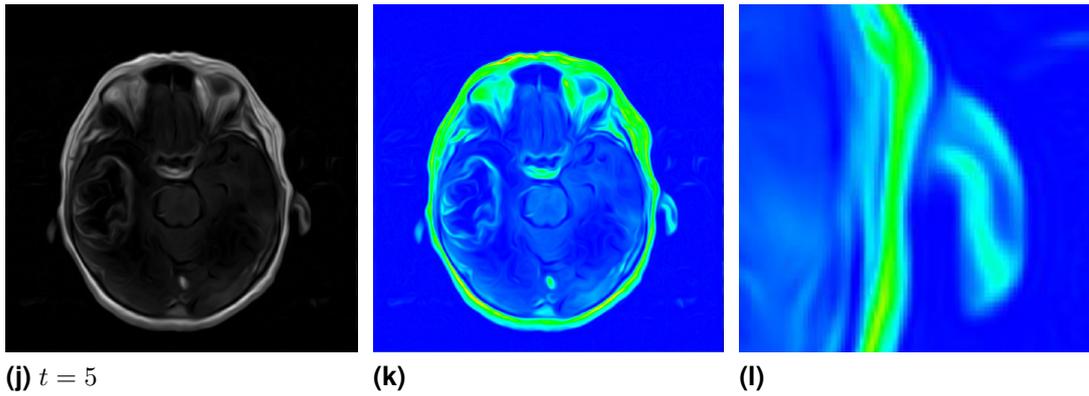
The regularisation parameter  $\alpha$  ensures a small amount of linear diffusion [63]. Thus,  $\alpha$  guarantees that the diffusion process never stops [64]. Consequently, if this parameter is set too high, the image is blurred, cf. **Figure 5.9d**.

In our experiments, the default parameter setting with  $T = 1.5$ ,  $\rho = 1.5$ ,  $\sigma = 0.5$ ,  $C = 1$  and  $\alpha = 0.001$  worked well in all cases.

### 5.3 White Top Hat

The nonflat white top hat has one parameter. This parameter  $t$  determines how many bright structures are extracted. The choice of the parameter setting for the white top hat is critical. We use this filtering technique to enlarge the gap between the outer contour of the skull and the ears. If  $t$  is too small, the contour of the skull is not completely extracted, shown in **Figure 5.10f**. If  $t$  is too large, the gap between skull and ear is not enlarged, cf. **Figure 5.10l**.



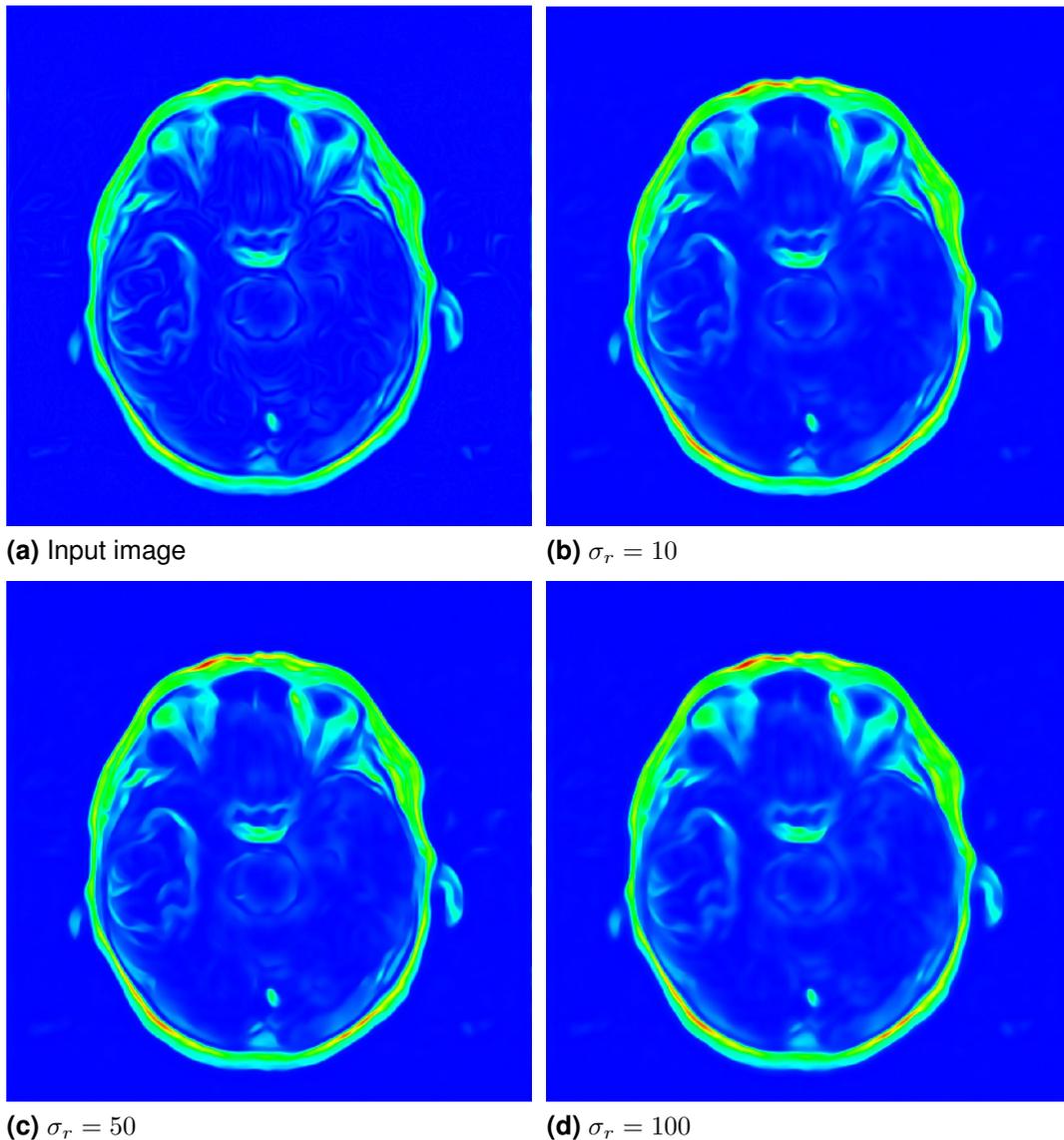


**Figure 5.10:** Result of white top hat filtering for different distance penaliser  $t$ .

Our experiments showed, that  $t = 1$  is a good trade-off. The outer contour of the skull is nearly completely extracted, while the gap between the skull and the ear is enlarged, **Figure 5.10i**.

## 5.4 Bilateral Filtering

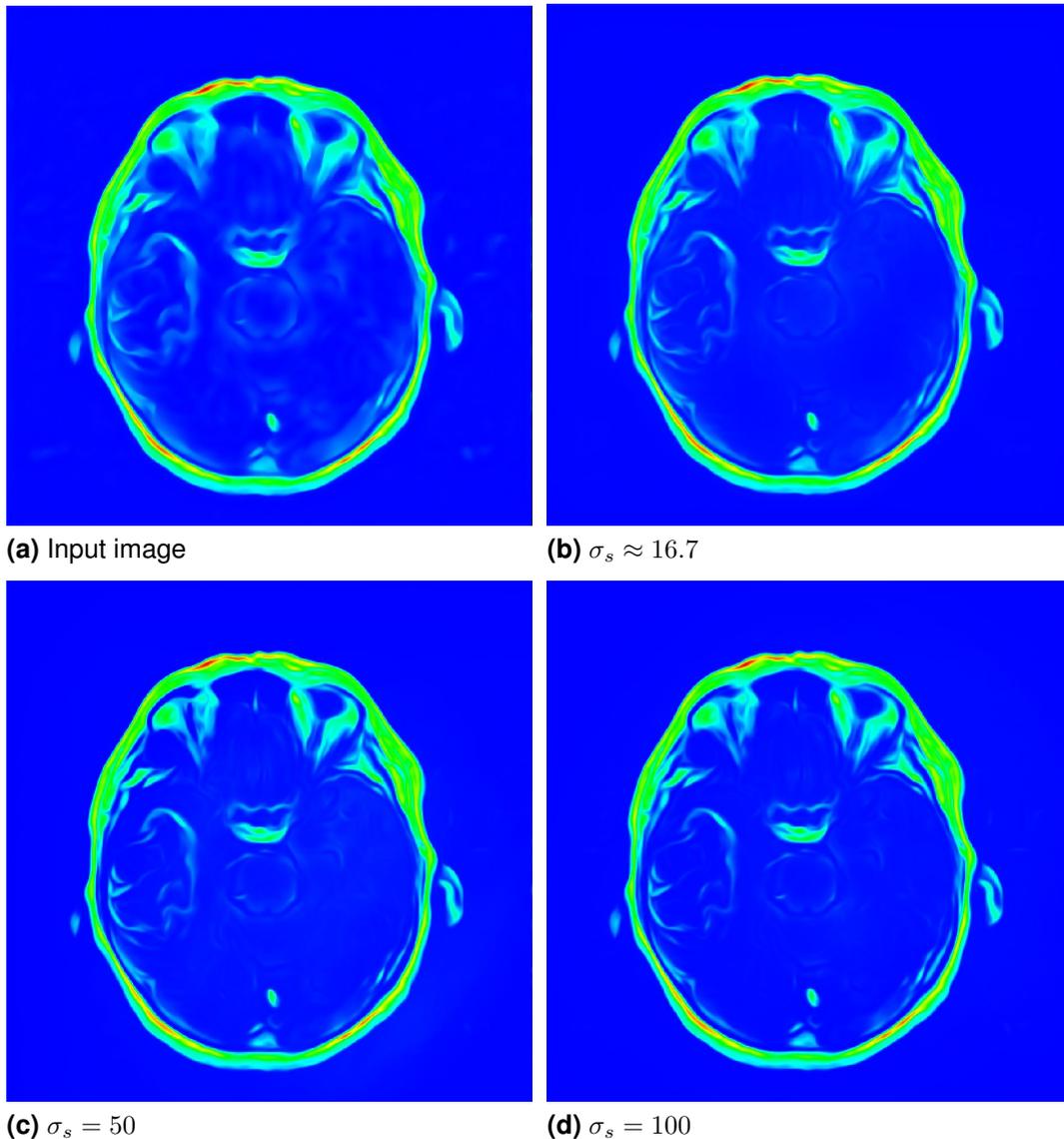
Bilateral filtering is an optional step when preprocessing MRI data to remove outer objects, i.e. the ears. We mentioned in **Section 2.3.1**, that bilateral filtering removes small structures in nearly homogeneous areas. It has two parameters. The standard deviation  $\sigma_r$  of the Gaussian kernel  $K_{\sigma_r}$  weights the distance between the intensities of the neighbouring pixel and the regarded pixel.



**Figure 5.11:** Bilateral filtering with different standard deviations of the range weighting function.  $\sigma_s \approx 16.7$ .

**Figure 5.11** shows the result of bilateral filtering for different values of  $\sigma_r$ . The gradient at the outer contour of the skull is in general very high. Hence, the pa-

parameter setting of  $\sigma_r$  is not critical. As shown in **Figure 5.11b** - **Figure 5.11d**, bilateral filtering removes more small details for increasing  $\sigma_r$ , while the outer contour of the skull is not affected. However, the areas of bone and background become more homogeneous. Thus, the **CV model** determines the shape of the skull easily.



**Figure 5.12:** Bilateral filtering with different standard deviations of the spatial distance weighting function.  $\sigma_r \approx 50$ .

The second parameter of the bilateral filtering is the standard deviation  $\sigma_r$  of the distance weighting function. **Figure 5.12** shows the result of bilateral filtering for different  $\sigma_r$ . Beforehand, we extract only bright structures with a white top hat. Neighbouring pixels are only regarded as similar, when they are near in

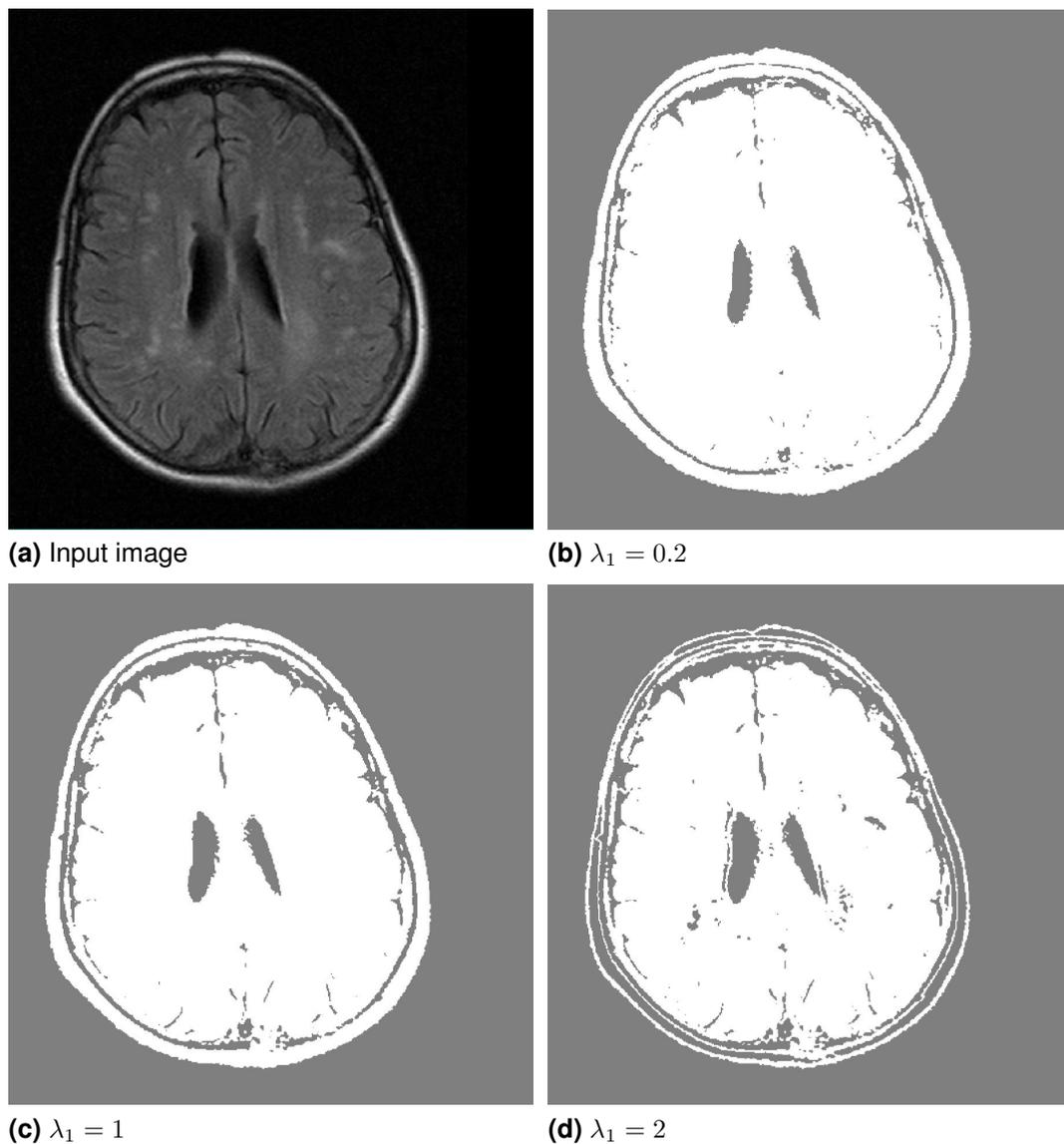
spatial and intensity distance. Consequently, it has nearly no effect on the outer contour of the skull when the size of the spatial neighbourhood is increased, shown in **Figure 5.12d**.

A human skull has an average thickness of  $\approx 7mm$  [29]. We decided therefore to adapt the spatial parameter to this value. The grid size of the pixels in the shown image of subject *G13* is  $h_x \approx 0.42mm, h_y \approx 0.42mm, h_z \approx 6mm$ . Hence, we choose for the spatial distance  $\sigma_r = \frac{7mm}{0.42mm} \approx 16.7$ . We also chose  $\sigma_s$  adopted to the image and set it to 5% of the maximal grey value  $max_{GV}(\Gamma)$  in the image. In our experiments, the standard parameter setting  $\sigma_r = \frac{7mm}{h_x}$  and  $\sigma_s = 0.05 * max_{GV}(\Gamma)$  works well in all cases.

---

## 5.5 Chan-Vese Active Contour Model

The **CV model** has several parameters. The most important ones are the weights  $\lambda_1$ ,  $\lambda_2$  for penalising the discrepancy between the input image and the arithmetic mean inside and outside the curve, respectively. The evolution under the **CV model** for different  $\lambda_1$  is shown in **Figure 5.13**.

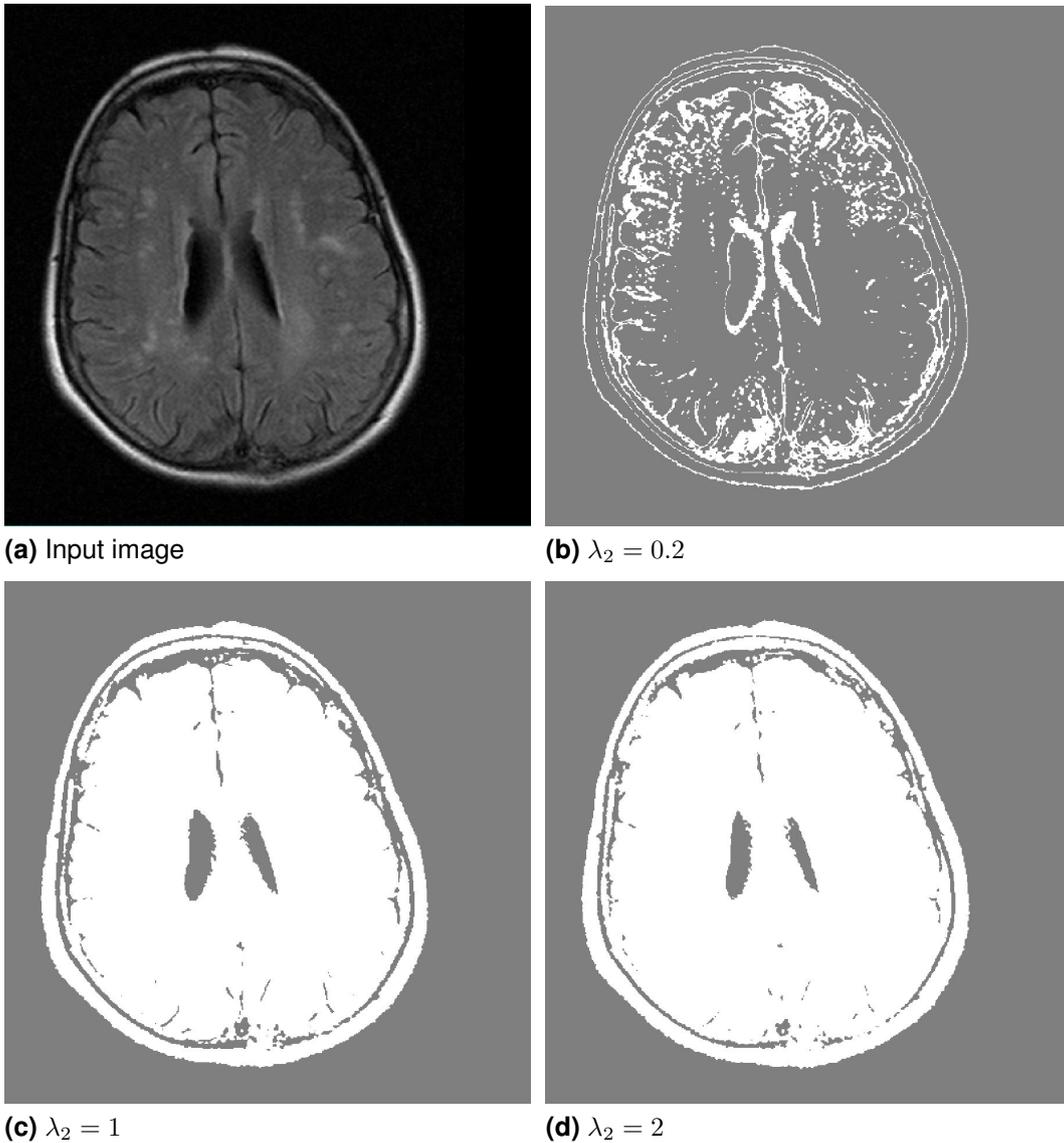


**Figure 5.13:** Evolution under the **CV model** for different weights for penalising the discrepancy between the input image and the arithmetic mean inside the curve.  $\alpha = 1$ ,  $\lambda_2 = 1$ ,  $\tau \approx 0.04$ ,  $\epsilon = h_x h_y$ , no reinitialisation of the level set function.

**Figure 5.13b** presents the steady state of the **CV model** when the weight for the difference between the average grey value inside the curve and the image is

low, i.e.  $\lambda_1 = 0.2$ . If this error term is less penalised than the one for the area outside the curve, the energy fits the background more accurately [11]. Similarly, finer details are detected when  $\lambda_1$  is large, see **Figure 5.13d**.

The according weight for the difference between the image and the arithmetic mean outside the curve is given by  $\lambda_2$ .

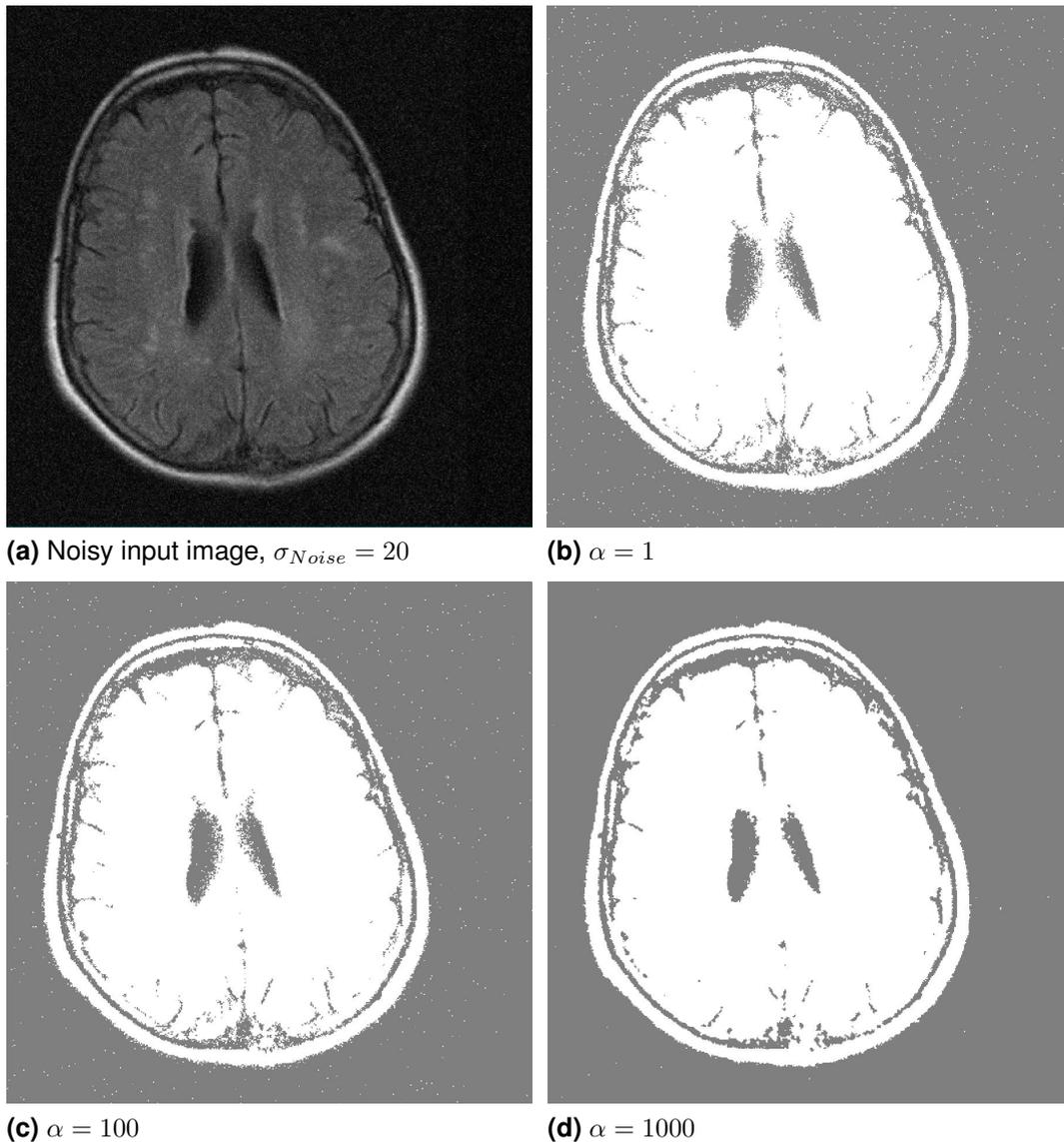


**Figure 5.14:** Evolution under the **CV model** for different weights for penalising the discrepancy between the input image and the arithmetic mean outside the curve.  $\alpha = 1$ ,  $\lambda_1 = 1$ ,  $\tau \approx 0.04$ ,  $\epsilon = h_x h_y$ , no reinitialisation of the level set function.

If the weight for this error term is low, the steady state of the **CV model** has more fluctuations in the background while the discrepancy in the foreground is minimised. This is depicted in **Figure 5.14b**. Likewise, the detected object

boundary contains less finer details when  $\lambda_2$  is large, see **Figure 5.14d**. We use the **CV model** to find the contour of the skull. Hence, we neither need a lot of details inside nor outside the skull. We suggest therefore to give the error terms of both classes the same weight, i.e.  $\lambda_1 = \lambda_2 = 1$ .

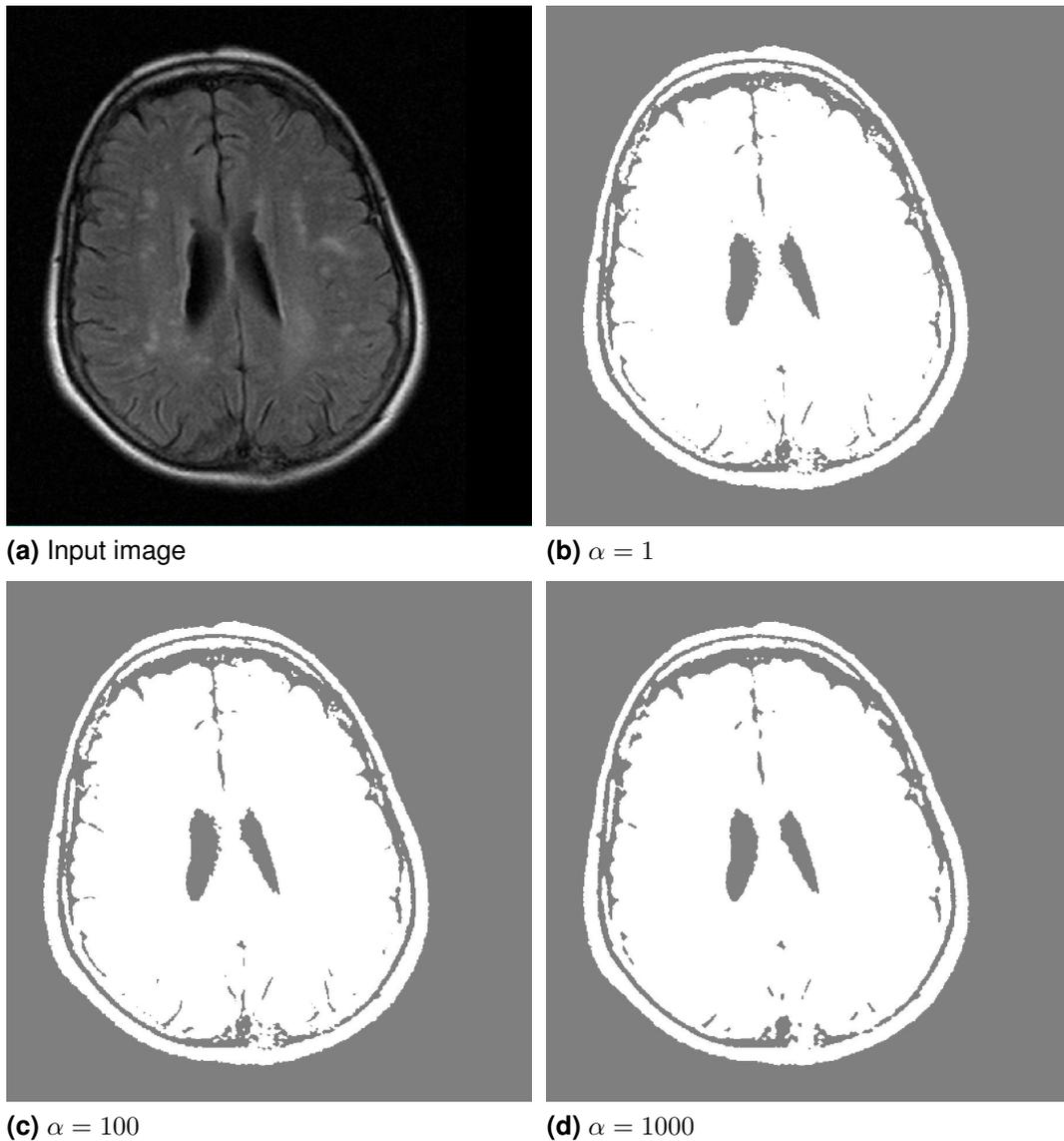
The parameter  $\alpha$  sets the penalty for the length of the segmentation boundary. The lower the length penalty is, the more detailed is the segmentation, as shown in **Figure 5.15** and **Figure 5.16**.



**Figure 5.15:** Evolution of a noisy input image under the **CV model** for different weights for penalising the length of the edge-set  $C$ .  $\lambda_1 = 0.1$ ,  $\lambda_2 = 0.1$ ,  $\tau \approx 0.04$ ,  $\epsilon = h_x h_y$ , no reinitialisation of the level set function.

Large settings for  $\alpha$  and/or small values for  $\lambda_1$ ,  $\lambda_2$  are necessary for noisy images.

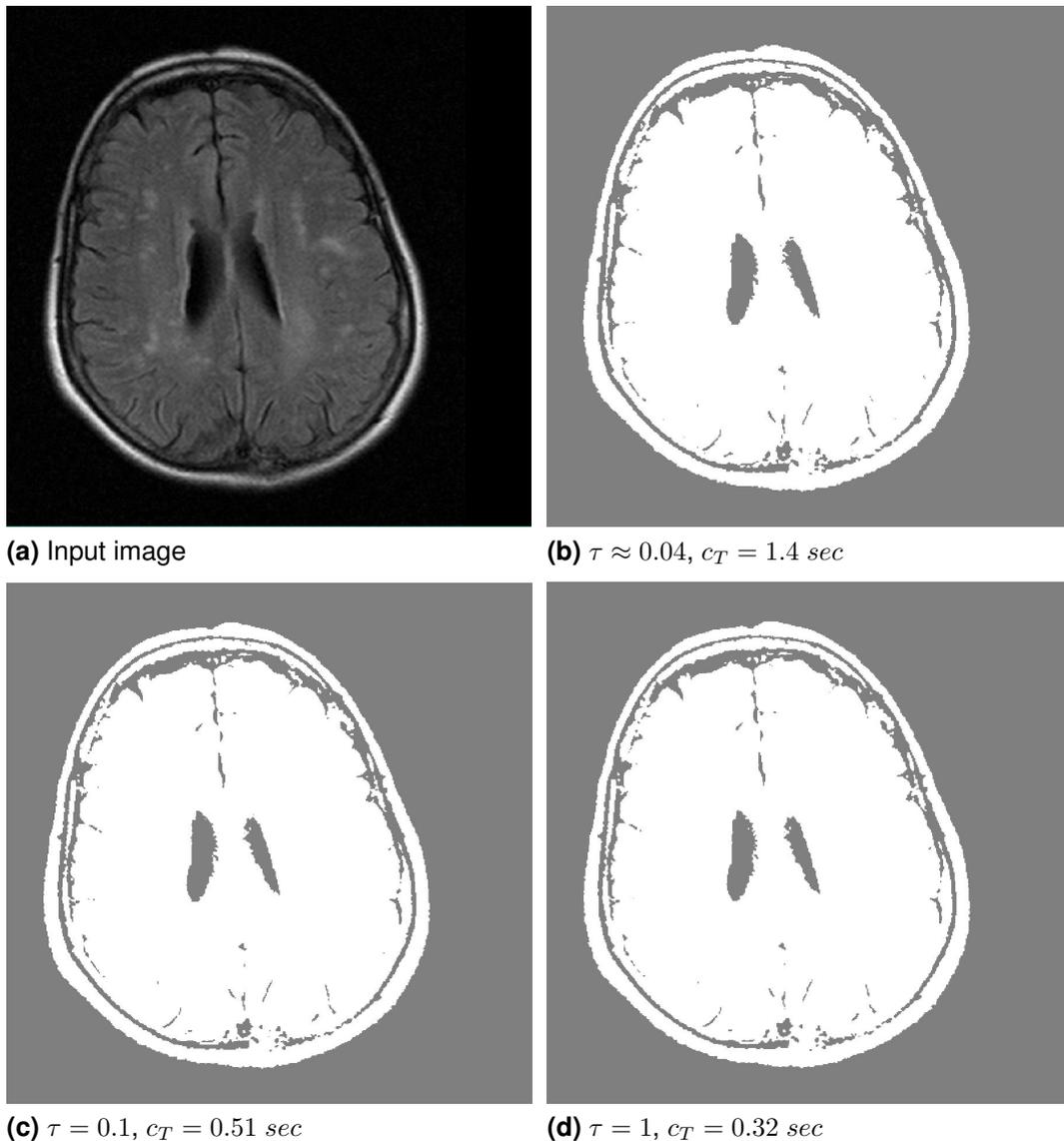
**Figure 5.15** shows the segmentation boundary for different length penalties in a noisy input image where we added Gaussian noise with  $\sigma = 20$ . However, we use the **CV model** in two ways. On the one hand we set the standard deviation of the Gaussian weighting function for the **NLMMeans** according to the variance in the background of the image. We do not need a perfect parameter setting in this case.



**Figure 5.16:** Evolution under the **CV model** for different weights for penalising the length of the edge-set  $C$ .  $\lambda_1 = 1$ ,  $\lambda_2 = 1$ ,  $\tau \approx 0.04$ ,  $\epsilon = h_x h_y$ , no reinitialisation of the level set function.

On the other hand, we use the detected object boundary as backbone of our affine registration process. Fortunately, we denoise the images with **NLMMeans**

beforehand. Consequently, we do not need to take too much care about noise. Accordingly, we do not have to adjust the parameters  $\lambda_1$  and  $\lambda_2$  to filter high frequent noise. The parameter setting for  $\alpha$  has therefore less effect on the resulting splitting, shown in **Figure 5.16**.



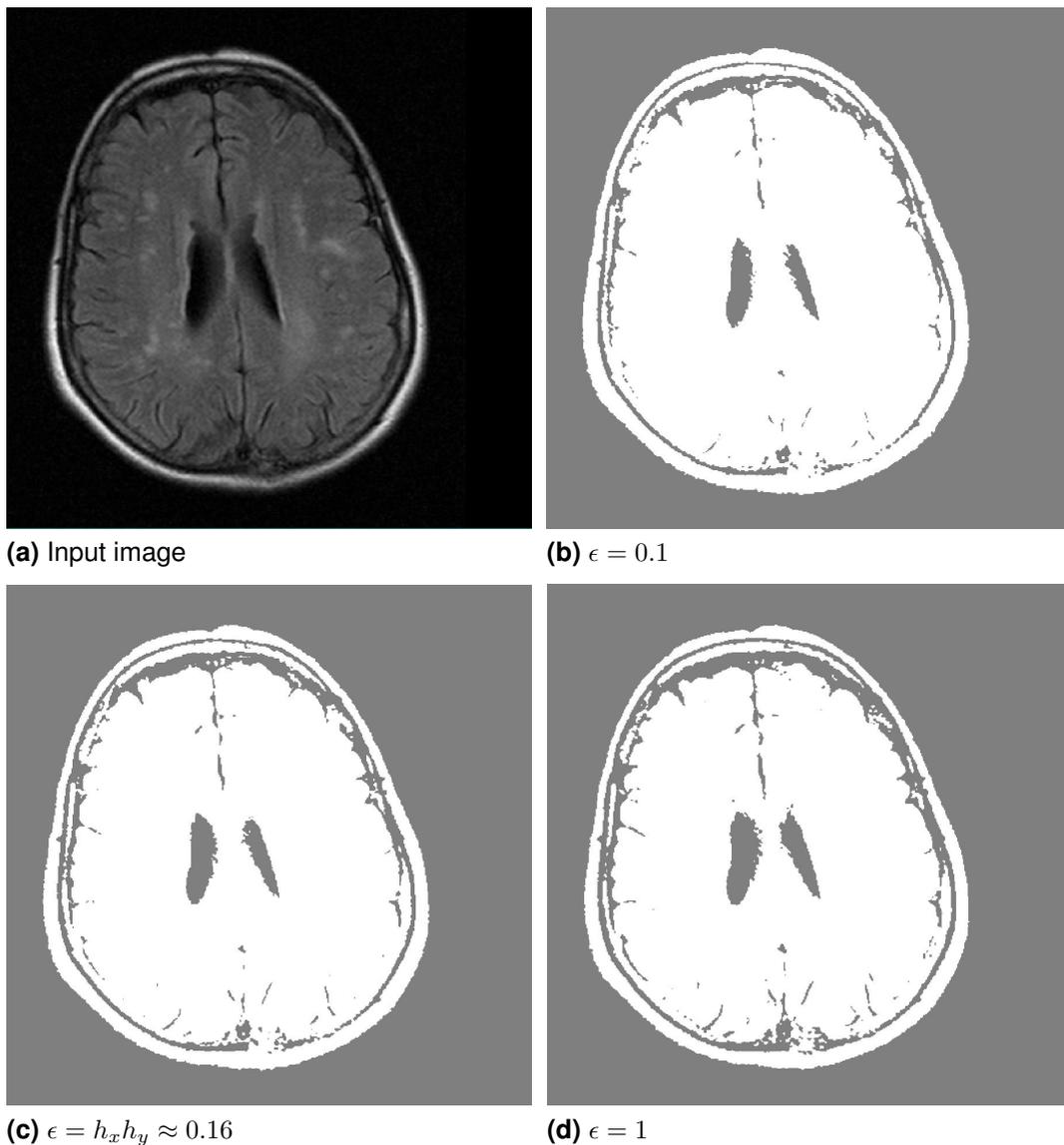
**Figure 5.17:** Evolution under the **CV model** for different time step sizes.  $\alpha = 1, \lambda_1 = 1, \lambda_2 = 1, \epsilon = h_x h_y$ , no reinitialisation of the level set function.  $c_T$ : computation time.

The parameter  $\tau$  determines the step size of the gradient descent. If  $\tau$  is too large, the algorithm might not reach the global minimum. Comparing **Figure 5.17b** and **Figure 5.17d** shows, that the steady state of the **CV model** for a smaller time step size gives are more detailed object boundary. This is especially visible

at the ventricles, i.e. the black area inside the brain. Unfortunately, if the step size is chosen to small, the computation time increases. We adopt therefore the time stepping to the grid dimensions of the image and set  $\tau$  the same way as for CED, see Section 3.2,

$$\tau = \frac{1}{\frac{2}{h_x^2} + \frac{2}{h_y^2}}. \quad (5.1)$$

We also have to choose the regularisation parameter  $\epsilon$  for the Heaviside function.

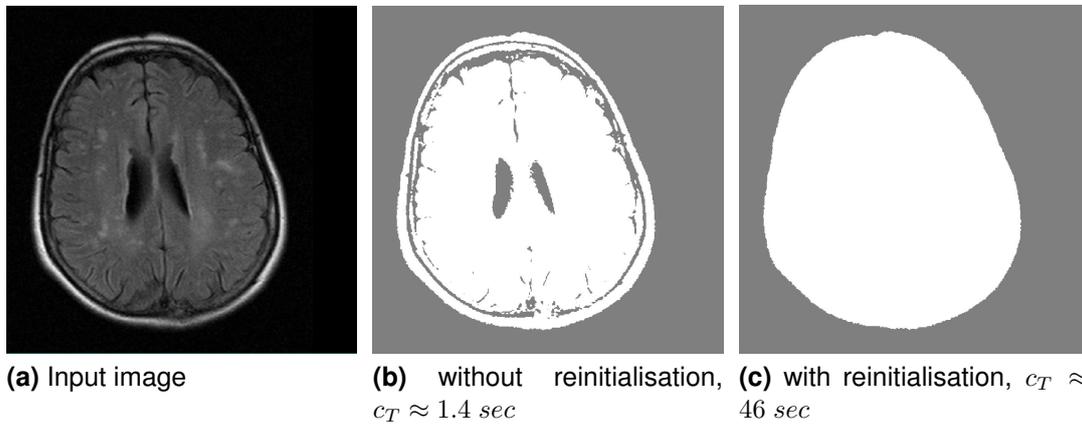


**Figure 5.18:** Evolution under the CV model for different regularisation parameters.  $\alpha = 1$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 1$ ,  $\tau \approx 0.04$ , no reinitialisation of the level set function.

**Figure 5.18** presents the steady states of the CV model for different regularisa-

tions. We decided to follow [11, 12, 13, 14] and set  $\epsilon$  according the grid dimension, i.e.  $\epsilon = h_x h_y$ .

We mentioned in **Section 3.4**, that it is also possible to reinitialise the level set to the euclidean signed distance function to its zero level curve.

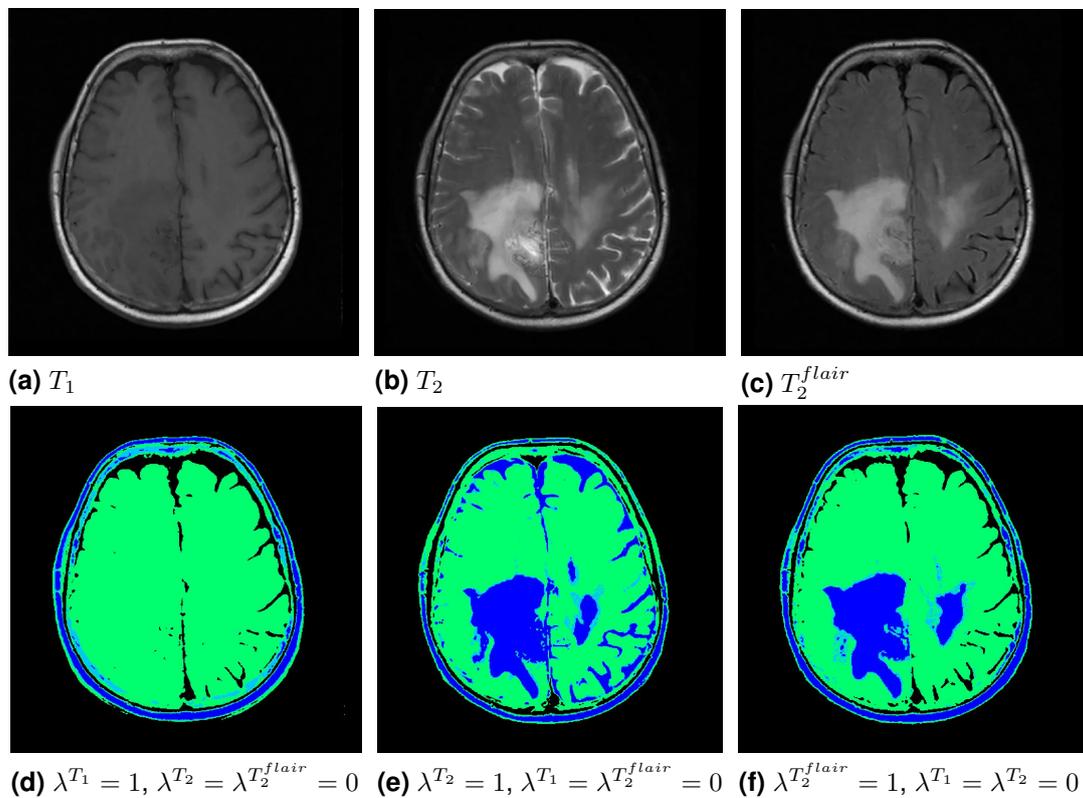


**Figure 5.19:** Evolution under the CV model (b) with and (c) without reinitialisation of the level set function.  $\alpha = 1$ ,  $\lambda_1 = 1$ ,  $\lambda_2 = 1$ ,  $\tau \approx 0.04$ ,  $\epsilon = h_x h_y$ .  $c_T$ : computation time.

This reinitialisation is a rescaling of the level set function  $\phi$ . If  $\phi$  is reinitialised every step, it prevents interior contours from growing [11], see **Figure 5.19c**. On the one hand, we are not interested in the interior contours when we determine the boundaries of the skull. On the other hand, the reinitialisation is a very time consuming step. We decided consequently to not rescale the level set function.

## 5.6 Chan-Vese Active Contour Model for Vector Valued Images with a Multiphase Level Set Representation

Similar to the *CV model*, the *CVec model* with a multiphase level set representation has several parameters that have to be chosen. We suggest to set  $\alpha$ ,  $\tau$  and  $\epsilon$  the same way as for the original formulation, see [Section 5.5](#). In the *CV model*, the parameters  $\lambda_1$  and  $\lambda_2$  sets the penalty for the discrepancy between the input image and the arithmetic means inside and outside the edge-set  $C$ , respectively. In the more sophisticated *CVec model* with a multiphase level set representation, we have to set the parameter  $\lambda^i$  for each channel  $f^i, i = 1, \dots, V$ .

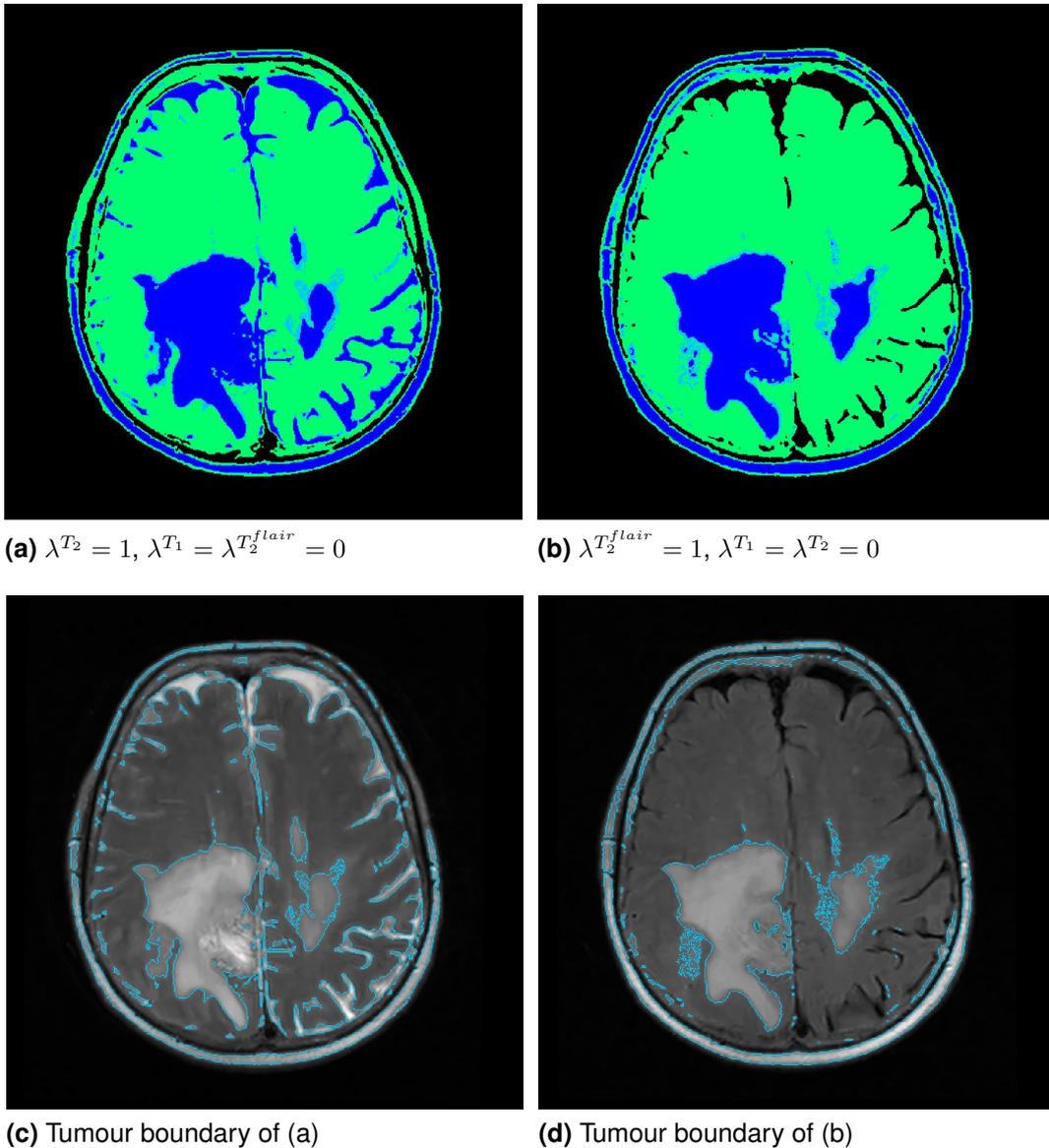


**Figure 5.20:** Evolution under the *CVec model* with a multiphase level set representation with different weights for the error terms of each channel.  $\alpha = 1$ ,  $\tau \approx 0.04$ ,  $\epsilon = h_x h_y$ .

These parameters weight the error terms of each channel. **Figure 5.20** shows the partitioning when just one channel is weighted, i.e. when the information from the other channels are neglected.

The most information about the oedema are present in the  $T_2$  and the  $T_2^{flair}$  sequences. If the information in the  $T_2$  image is considered exclusively, the tumour boundary is visible. However, there are also a lot of false positives inside

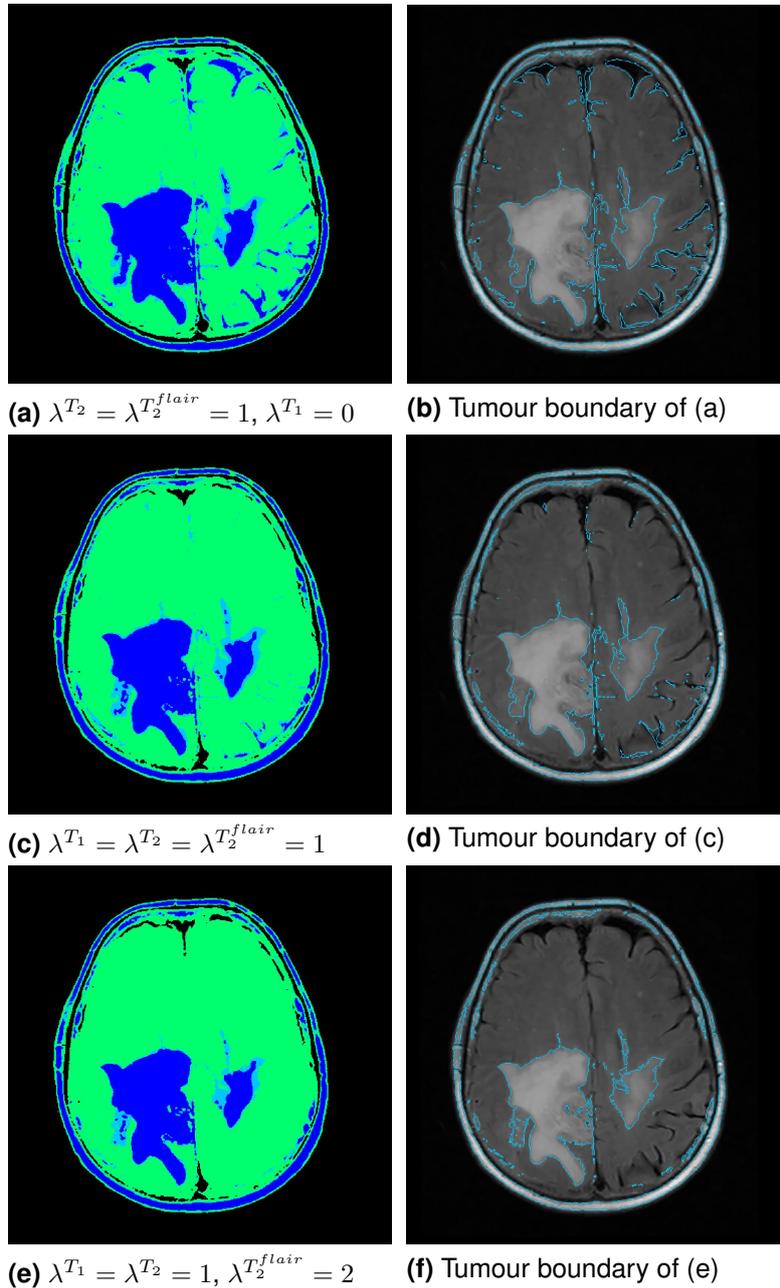
the brain, see **Figure 5.21c**. Similarly, if just the  $T_2^{flair}$  data is taken into account, the boundary of the oedema is visible. Nevertheless, also in this case, the segmentation boundary is suboptimal. Although there are nearly no false positives, the tumour is not completely detected, see **Figure 5.21d**.



**Figure 5.21:** Evolution under the **CVec model** with a multiphase level set representation for different combinations of channel weights.  $\alpha = 1, \tau \approx 0.04, \epsilon = h_x h_y$ . no reinitialisation of the level set functions.

To overcome this problem, we combine all information we have, i.e. more than one channel has to influence the partitioning. **Figure 5.22** presents the partitioning of our model for different weight combinations, while the result is based

on data from different channels. The segmentation boundary shown in **Figure 5.22b** has less false positives than **Figure 5.21d** but a lot of wrongly detected boundaries are still present therein. Nearly no information about the oedematous area is observable in the  $T_1$  sequence. Nevertheless, including this information improves the result. This is demonstrated by **Figure 5.22d**.



**Figure 5.22:** Evolution under the **CVec model** with a multiphase level set representation for different combinations of channel weights.  $\alpha = 1$ ,  $\tau \approx 0.04$ ,  $\epsilon = h_x h_y$ . No reinitialisation of the level set functions.

---

The oedema is still completely detected while less false positives are present in the segmentation boundary. **Figure 5.22f** shows the resulting partitioning of our model, when a priori knowledge is included. The oedematous tissue is most observable in  $T_2$  and  $T_2^{flair}$  sequences. However, also the cerebrospinal fluid appears bright in the  $T_2$  scan. Accordingly, a higher weight for the  $T_2^{flair}$  channel results in a better partitioning.

The selection of the weights  $\lambda^i$  for the different channels is the most tedious part in our complete process chain. We suggest for this reason to use a priori knowledge about the images. The oedema is best visible in a **MRI** sequence with  $T_2^{flair}$  weighting. Hence, the  $T_2^{flair}$  image should have a higher weight than other types of **MRI** scans, when the oedema has to be extracted. Similarly, the central area of necrosis is best observable in just one of the **MRI** sequences, i.e.  $T_1^{cont}$ . Accordingly, the  $T_1^{cont}$  scan should have the highest weight when this part of the tumour is extracted.



## 6 Experimental Results

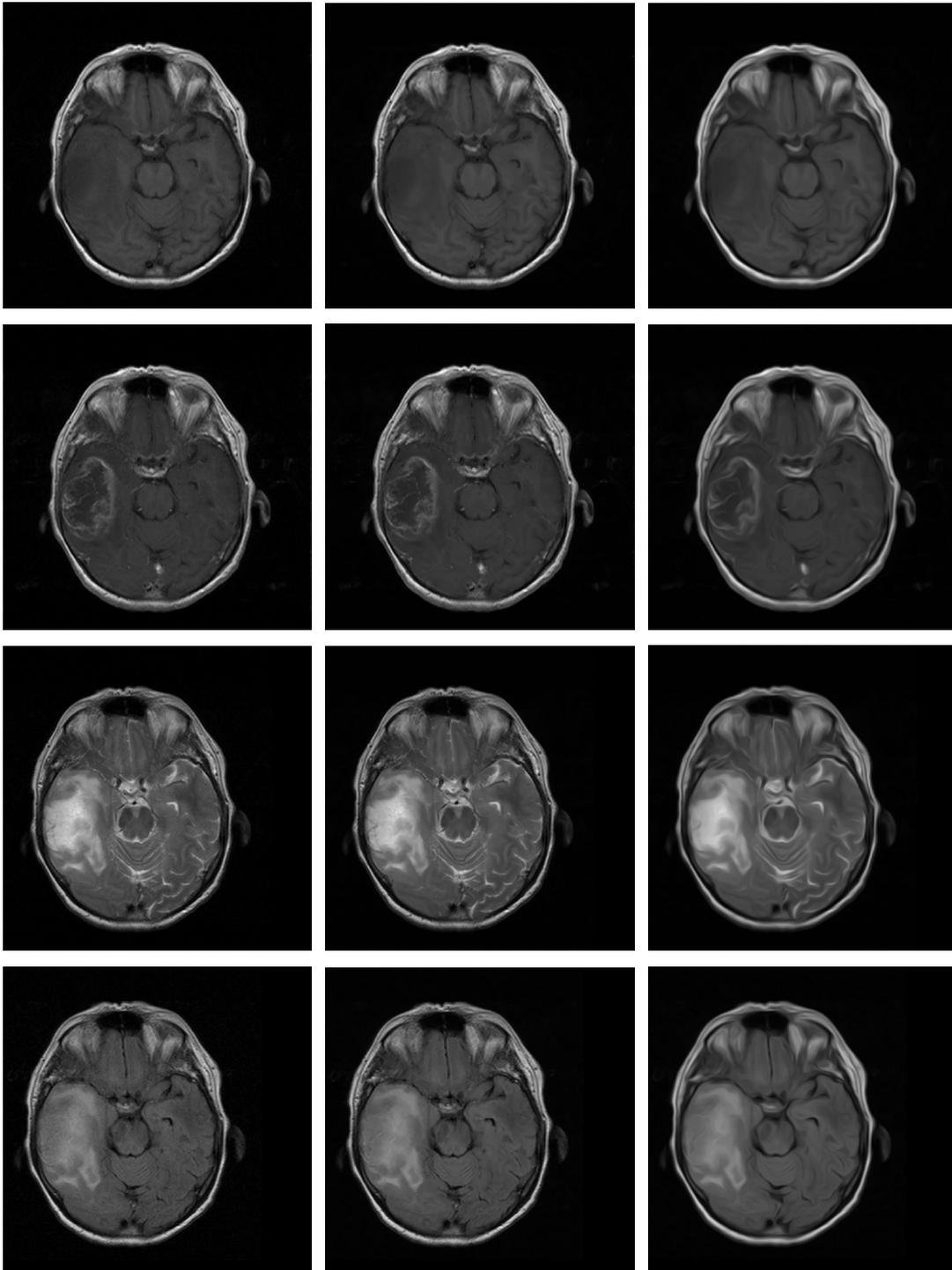
We use in the following several image stacks to evaluate our process chain. We make use of these data to show the capabilities and limitations of our methods.

All calculations for **CV model** based methods are performed using a regularisation parameter for the Heaviside function of  $\epsilon = h_x h_y$  and a length penalty of  $\alpha = 1$ . If not stated otherwise, we always apply a parameter setting of  $T = 1.5$ ,  $\rho = 1.5$ ,  $\sigma = 0.5$ ,  $C = 1$  and  $\alpha = 0.001$  for **CED**. Additionally, we always use the parameter setting  $t = 1$  for the white top hat filtering and  $\sigma_r = \frac{7mm}{h_x}$ ,  $\sigma_s = 0.05 * max_{GV}(\Gamma)$  for bilateral filtering, respectively.

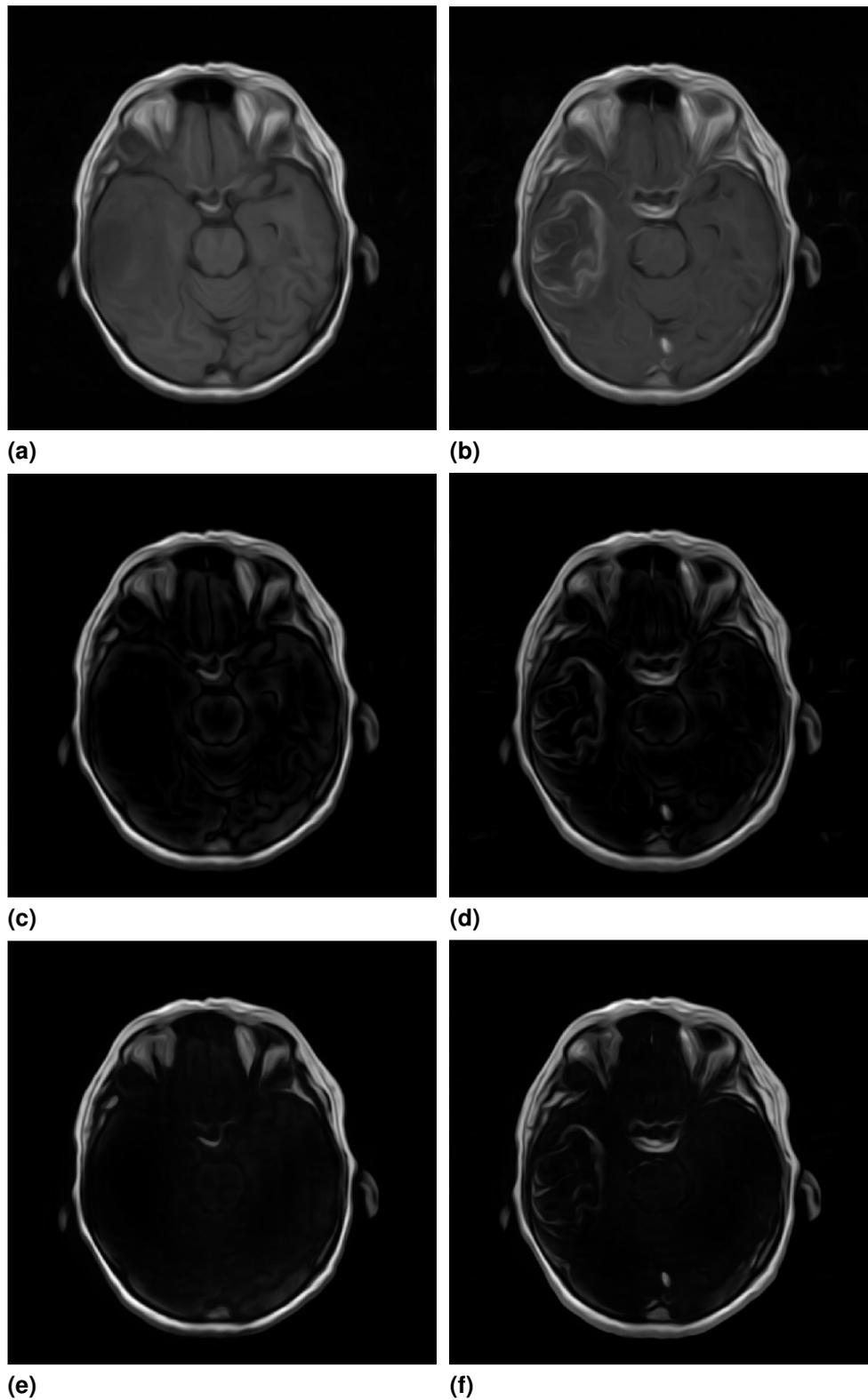
All computations that need a time stepping, i.e. **CED** and the **CV model** based methods, use the time step size  $\tau = \frac{1}{\frac{2}{h_x^2} + \frac{2}{h_y^2}}$ .

The figures are shown as follows. First, we show a series of images for each channel, that is available. Each series contains also the intermediate steps of denoising and registration as well as the segmentation boundary of the tumour tissue. As shown in **Section 5.6**, choosing the parameters for the **CVec model** with a multiphase level set representation requires some a priori knowledge about the image data. Consequently, we always give the  $T_2^{flair}$  sequence a higher weight, when we want to extract the oedema. Similarly, we apply also a higher weight for the  $T_1^{cont}$  sequence, when we extract the central area of necrosis.

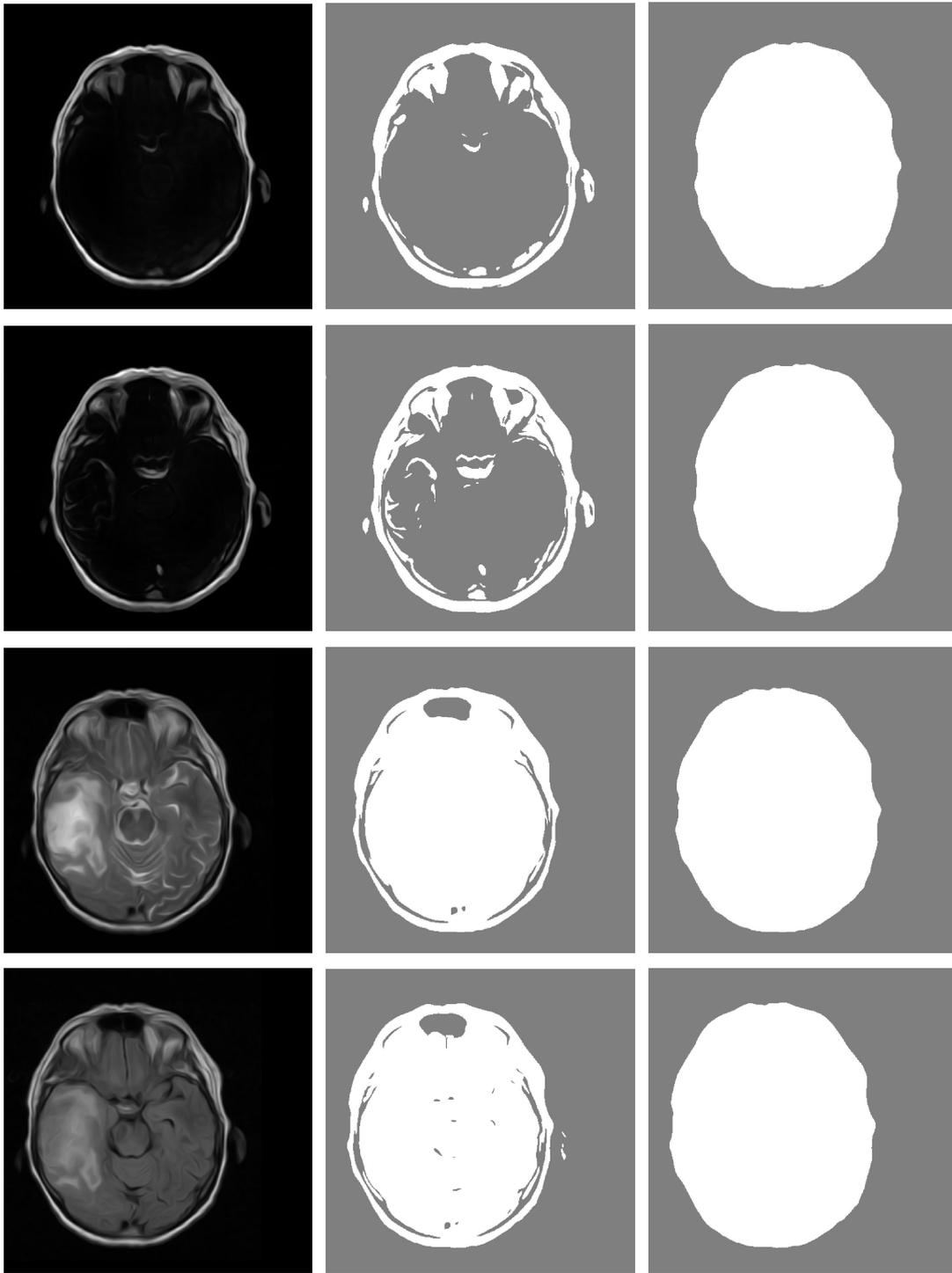
**Figure 6.1** to **Figure 6.4** show the intermediate steps to determine the boundaries of tumour tissue for  $G13$  slice 10. As shown in **Figure 6.1**, the contours of the skull are different for the  $T_1$  and  $T_1^{cont}$  images. The ears are imaged in both of them while they are barely visible in the  $T_2$  and  $T_2^{flair}$  sequences. **Figure 6.2** presents therefore the process chain to separate the skull and the ears. **Figure 6.3** displays the steps to gain the registration masks. As we can observe, the registration masks are nearly identical. Thus, the subsequent registration is accurate and the segmentation of the oedematous tissue is precise, see **Figure 6.4**. However, the parameter choice for detecting the boundaries of the central area of necrosis is tedious. As shown in **Figure 6.4**, even for an adjusted parameter setting there are a lot of false positives present therein. Although the necrotic area is segmented, the segmentation boundary is fuzzy and unsatisfactory.



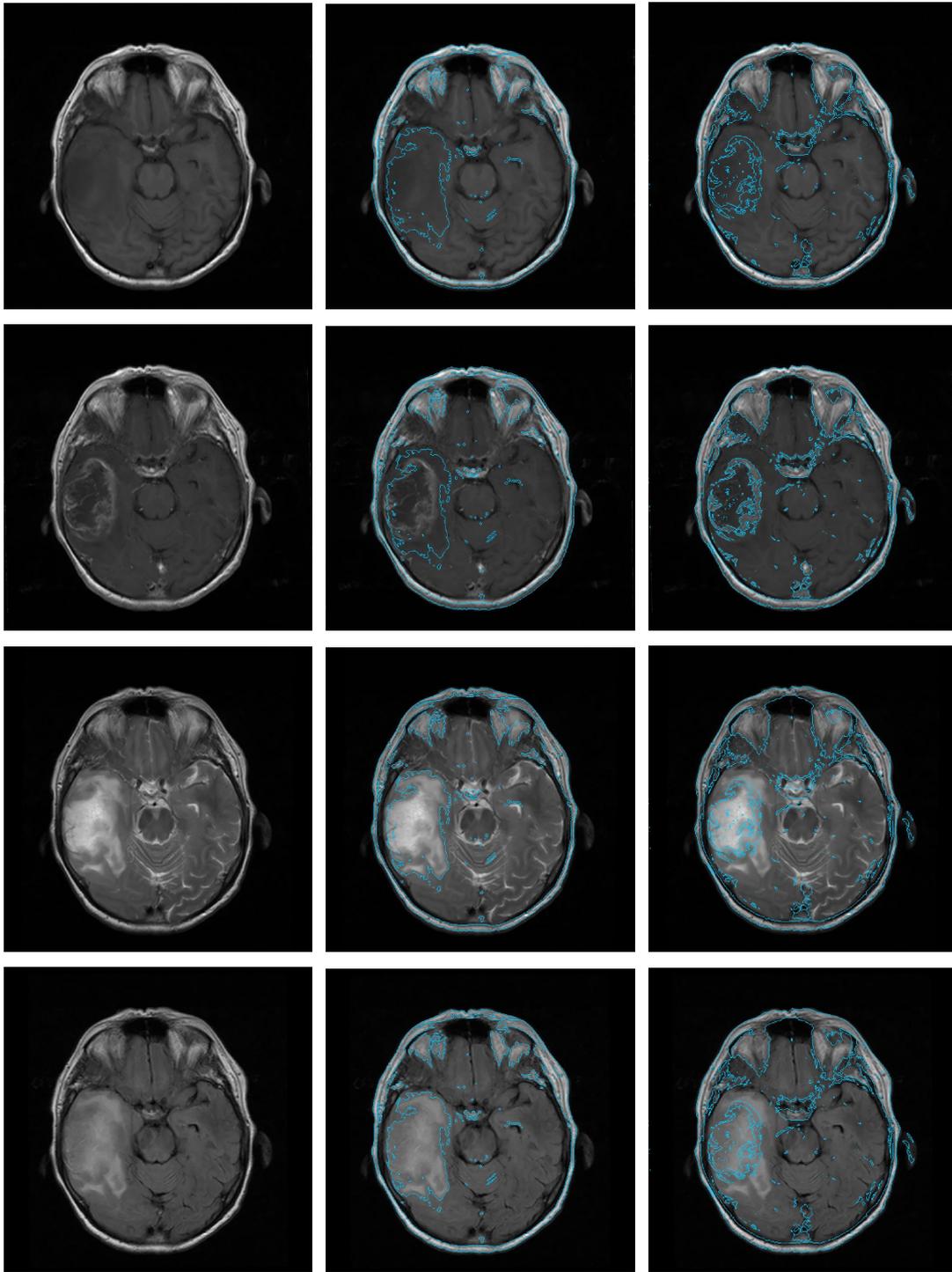
**Figure 6.1:** Experimental results of  $G13$  slice 10. **NLMeans** and **CED**. **Left:** From top to bottom:  $T_1$ ,  $T_1^{cont}$ ,  $T_2$ ,  $T_2^{flair}$ . **Middle:** Result of **NLMeans**. **Right:** Result of **CED**.



**Figure 6.2:** Experimental results of  $G_{13}$  slice 10. **CED**, white top hat and bilateral filtering. **Left column:**  $T_1$  sequence. **Right column:**  $T_1^{cont}$  sequence. From top to bottom: Result of **CED**, white top hat filtering and bilateral filtering.

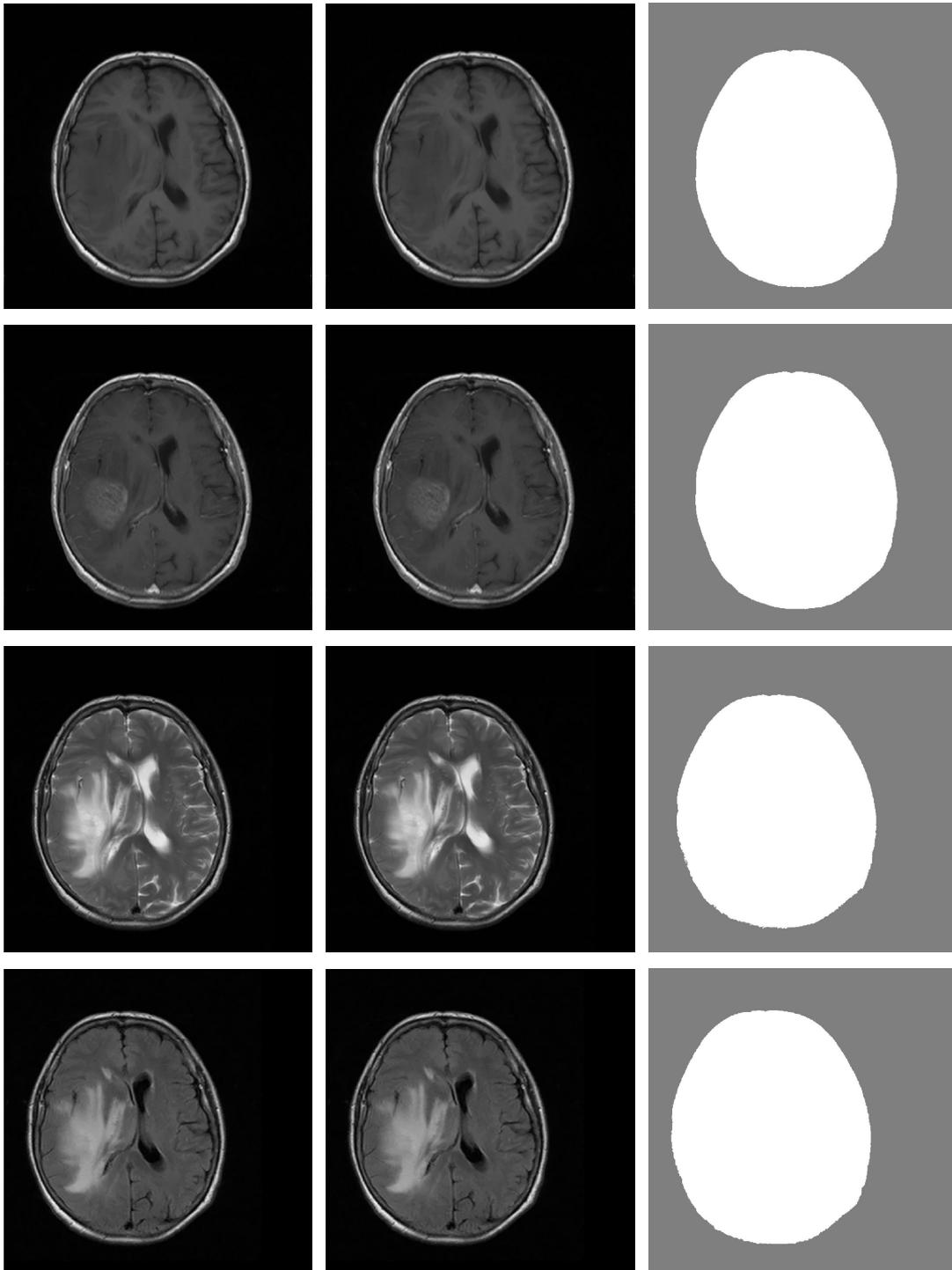


**Figure 6.3:** Experimental results of  $G_{13}$  slice 10. **CV model** and connected component labelling. **Left:** From top to bottom:  $T_1, T_1^{cont}, T_2, T_2^{flair}$ . **Middle:** Steady state of the **CV model**. **Right:** Registration mask.



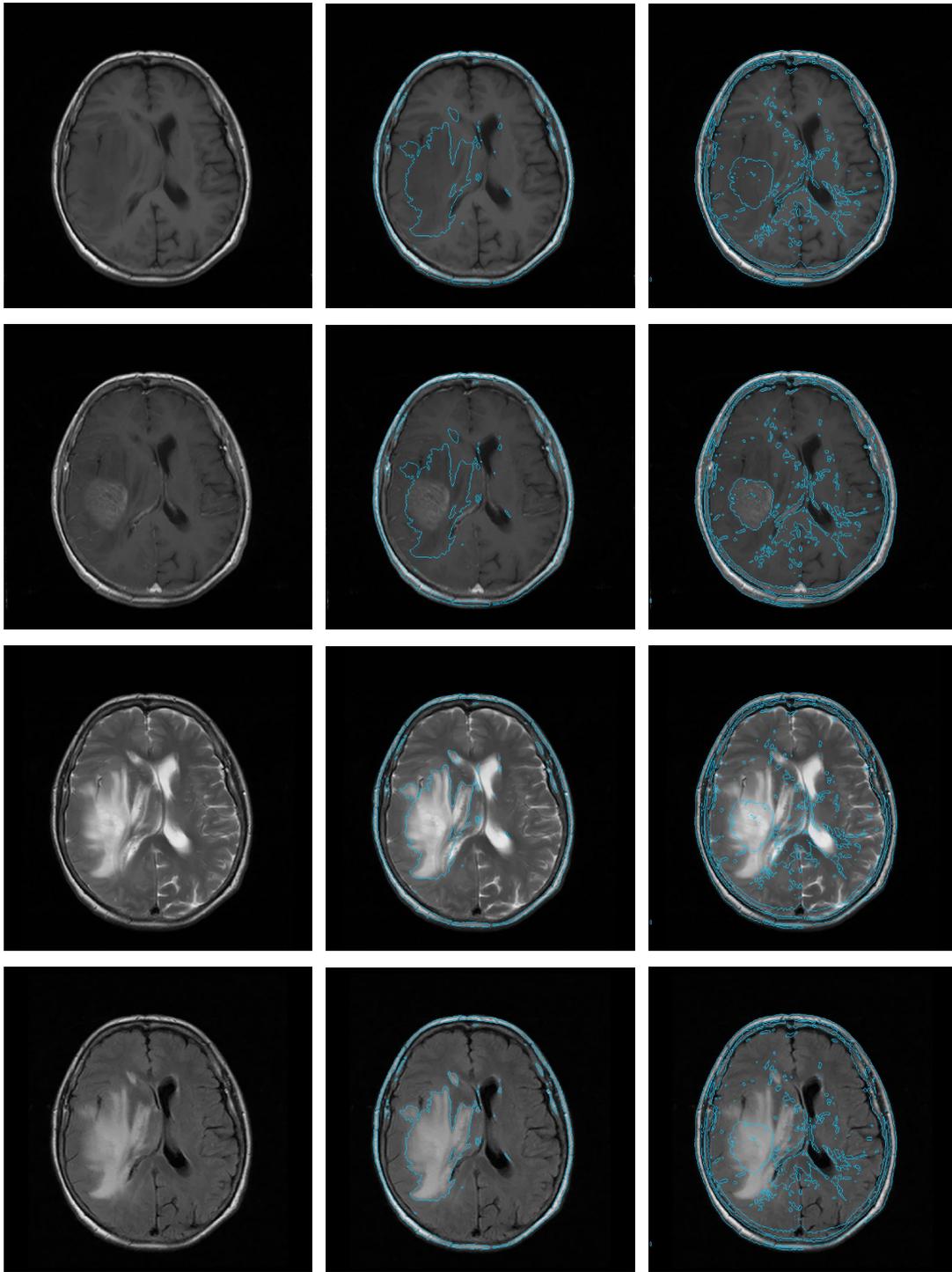
**Figure 6.4:** Experimental results of  $G_{13}$  slice 10. Registration, oedema boundary and necrosis boundary. **Left:** From top to bottom:  $T_1$ ,  $T_1^{cont}$ ,  $T_2$ ,  $T_2^{flair}$ , registered. **Middle:** Oedema boundary,  $\lambda^{T_1} = \lambda^{T_1^{cont}} = 1$ ,  $\lambda^{T_2} = 2$ ,  $\lambda^{T_2^{flair}} = 10$ . **Right:** Necrosis boundary.  $\lambda^{T_1} = 1$ ,  $\lambda^{T_1^{cont}} = 10$ ,  $\lambda^{T_2} = 0$ ,  $\lambda^{T_2^{flair}} = 0$ .

**Figure 6.5** and **Figure 6.6** present the intermediate steps of our approach for  $G13$  slice 8. As shown in **Figure 6.5**, the contour of the skull is not disturbed in the  $T_1$  and  $T_1^{cont}$  scans. Thus, we extract directly the shape of the skull and



**Figure 6.5:** Experimental results of  $G13$  slice 8. **NLMeans** and registration mask. **Left:** From top to bottom:  $T_1$ ,  $T_1^{cont}$ ,  $T_2$ ,  $T_2^{flair}$ . **Middle:** Result of **NLMeans**. **Right:** Registration mask.

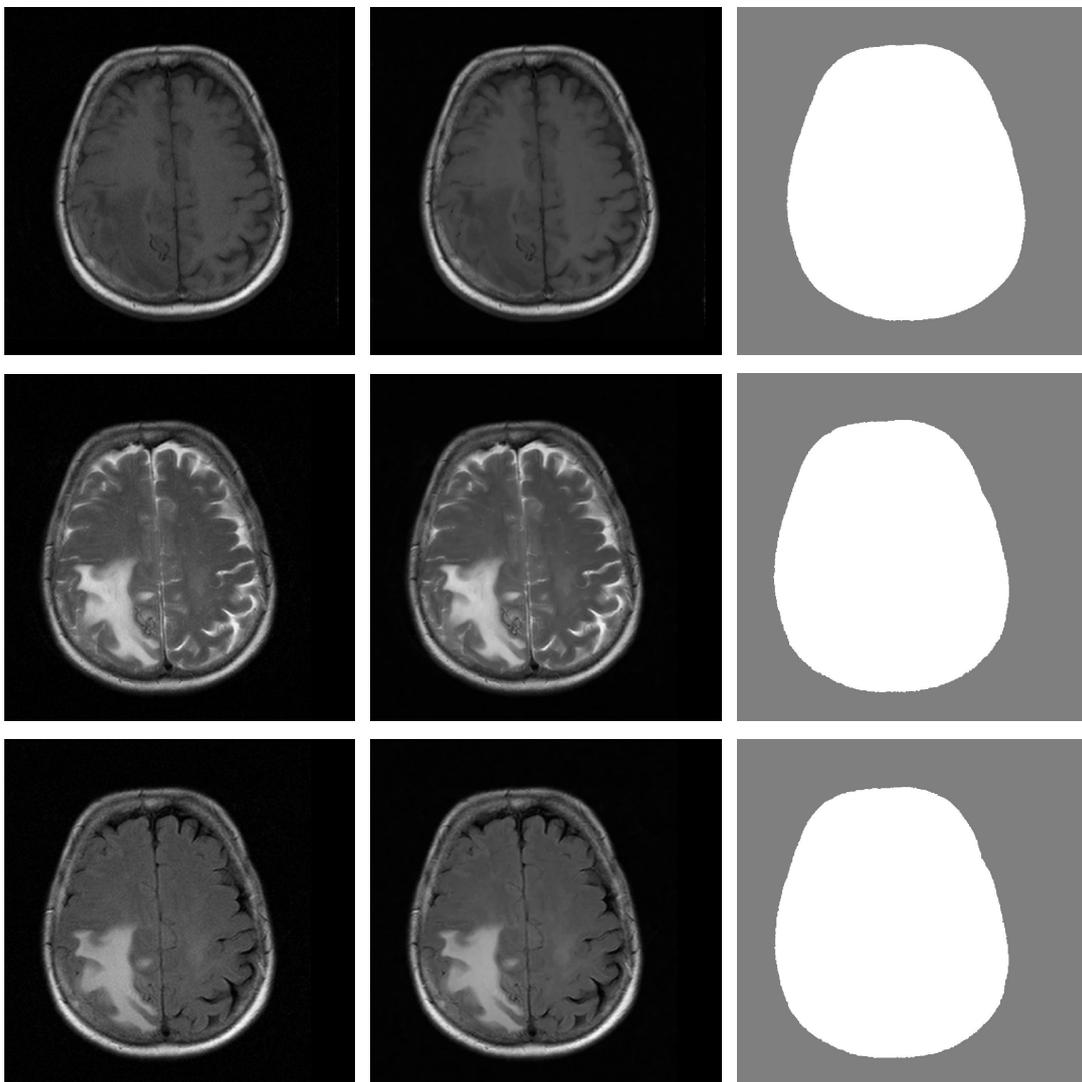
register the images. As observable in **Figure 6.6** we get an accurate segmentation of the oedema. Unfortunately, the necrotic area suffers under an extreme



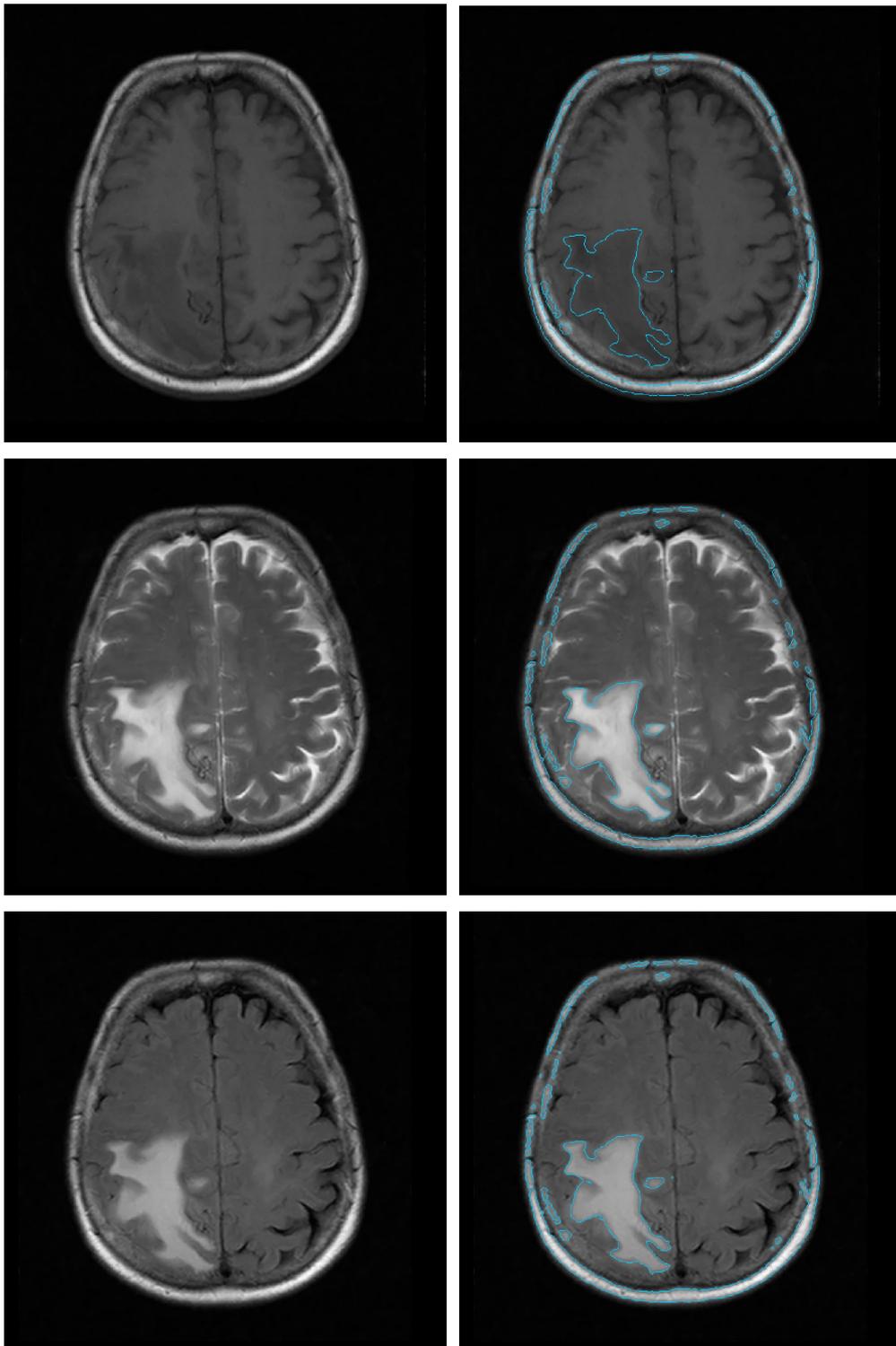
**Figure 6.6:** Experimental results of  $G_{13}$  slice 8. Registration, oedema boundary and necrosis boundary. **Left:** From top to bottom:  $T_1$ ,  $T_1^{cont}$ ,  $T_2$ ,  $T_2^{flair}$ , registered. **Middle:** Oedema boundary,  $\lambda^{T_1} = \lambda^{T_1^{cont}} = 1$ ,  $\lambda^{T_2} = 2$ ,  $\lambda^{T_2^{flair}} = 10$ . **Right:** Necrosis boundary.  $\lambda^{T_1} = 0.5$ ,  $\lambda^{T_1^{cont}} = 10$ ,  $\lambda^{T_2} = 0$ ,  $\lambda^{T_2^{flair}} = 0.5$ .

amount of false positives. The central area of necrosis is surrounded by tissue with lower intensity values. Hence, for a human observer, the necrotic area seems to have higher intensity values. Unfortunately, this is not the case - the intensity values of the necrotic area are very similar to those of white matter. Thus, the amount of false positives is huge. Fortunately, the central area of necrosis is always surrounded by oedematous tissue. Consequently, we can neglect all segments that do not overlap with the oedema. Hence, the amount of false positives reduces dramatically.

**Figure 6.7** and **Figure 6.8** display the process chain for  $G25$  slice 6. Also in this case, the segmentation of oedematous tissue is successful and accurate.



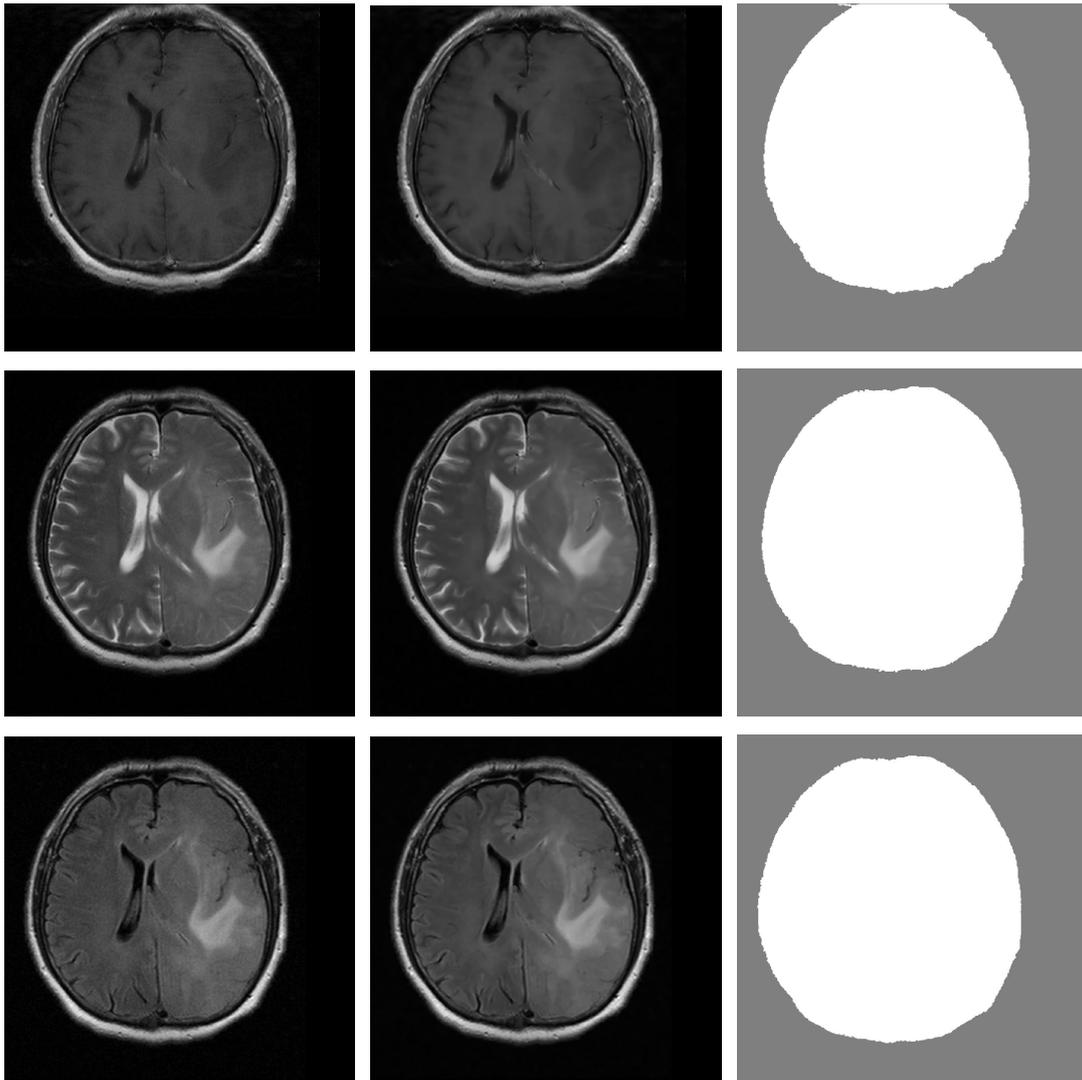
**Figure 6.7:** Experimental results of  $G25$  slice 6. **NLMMeans** and registration mask. **Left:** From top to bottom:  $T_1$ ,  $T_2$ ,  $T_2^{flair}$ . **Middle:** Result of **NLMMeans**. **Right:** Registration mask.



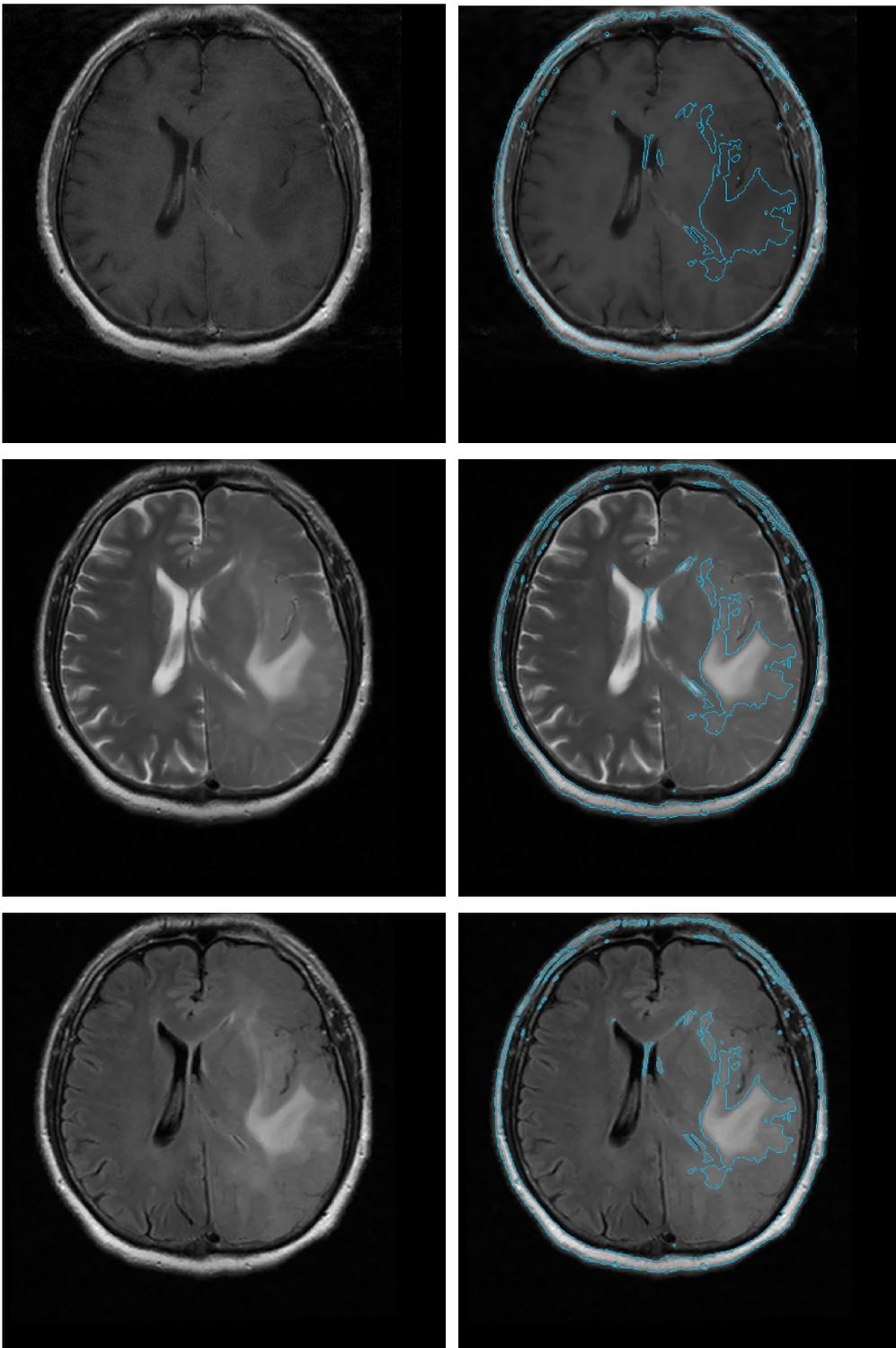
**Figure 6.8:** Experimental results of  $G_{25}$  slice 6. Registration and oedema boundary. **Left:** From top to bottom:  $T_1$ ,  $T_2$ ,  $T_2^{flair}$ , registered. **Right:** Oedema boundary,  $\lambda^{T_1} = 1$ ,  $\lambda^{T_2} = 1$ ,  $\lambda^{T_2^{flair}} = 2$ .

**Figure 6.9** and **Figure 6.10** show the intermediate steps for  $G2$  slice 8. This application is more challenging because the skull is not completely imaged in the reference sequence, i.e. the  $T_1$  scan.

However, as shown in **Figure 6.10**, the result of the registration is nevertheless acceptable and the oedematous tissue is segmented very well.



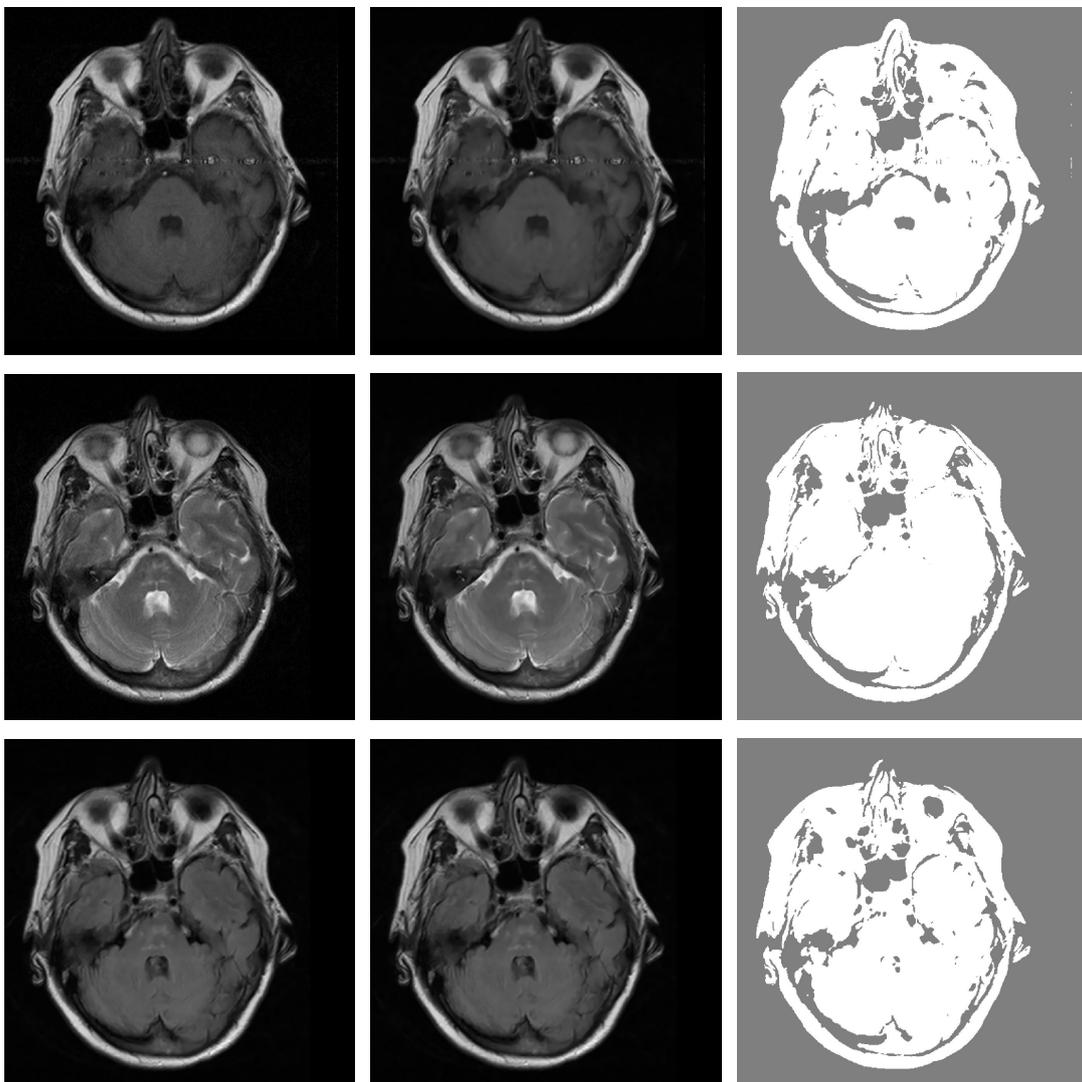
**Figure 6.9:** Experimental results of  $G2$  slice 8. **NLMMeans** and registration mask. **Left:** From top to bottom:  $T_1$ ,  $T_2$ ,  $T_2^{flair}$ . **Middle:** Result of **NLMMeans**. **Right:** Registration mask.



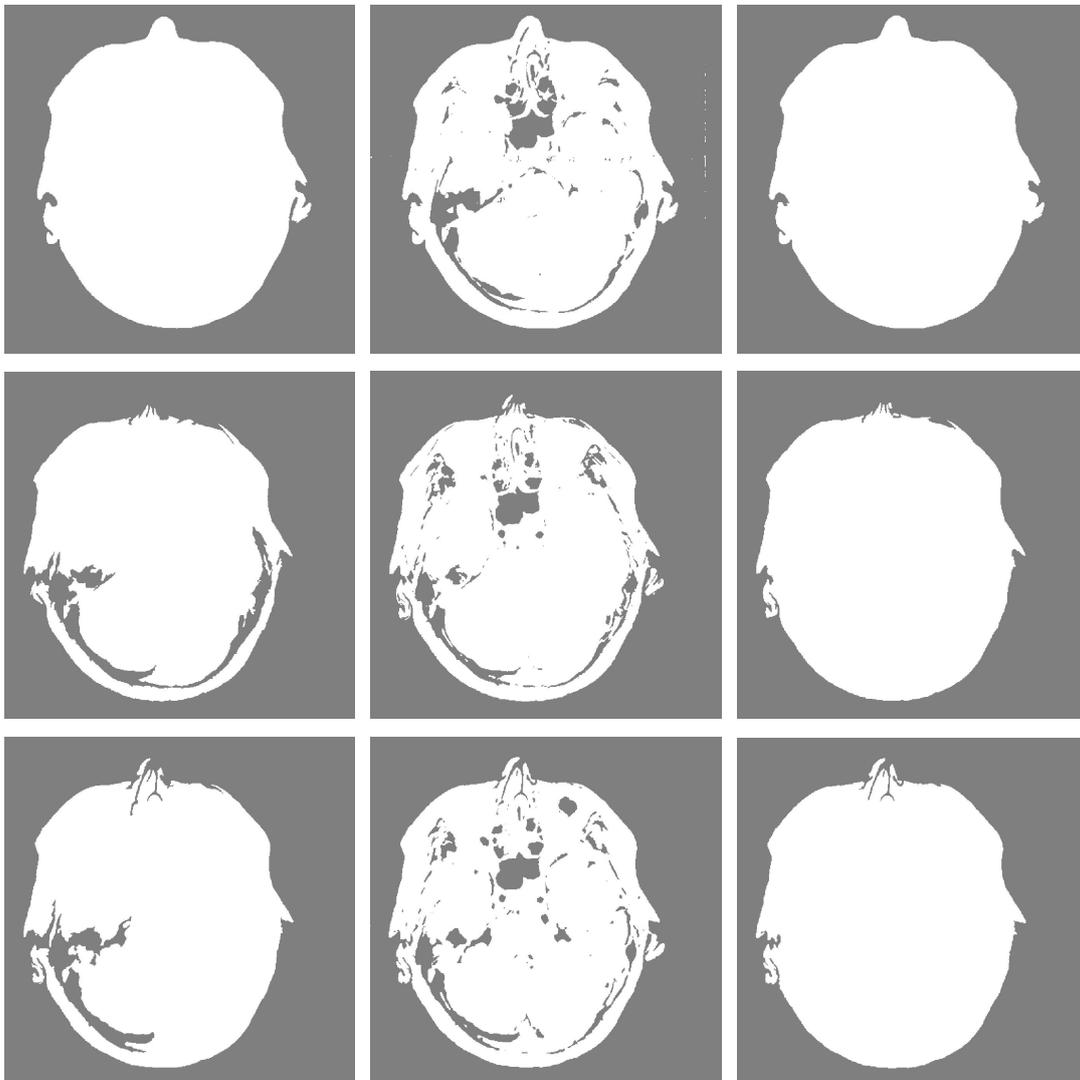
**Figure 6.10:** Experimental results of  $G_2$  slice 8. Registration and oedema boundary. **Left:** From top to bottom:  $T_1$ ,  $T_2$ ,  $T_2^{flair}$ , registered. **Right:** Oedema boundary,  $\lambda^{T_1} = 1$ ,  $\lambda^{T_2} = 1$ ,  $\lambda^{T_2^{flair}} = 2$ .

**Figure 6.11** and **Figure 6.12** present the processing of  $G25$  slice 19. As visible in **Figure 6.11**, the contour of the skull is not a homogeneous area. Hence, the splitting of the image in fore- and background assigns small areas of the bone to the background. Thus, our registration mask has holes and the registration fails.

**Figure 6.12** shows also the steady state of the **CV model** for a different parameter setting. Here, we highly penalise the error terms for the discrepancy in the background. Unfortunately, it is also not possible with this parameter setting to gain the same contour for all sequences.



**Figure 6.11:** Experimental results of  $G25$  slice 19. **NLMeans** and **CV model**. **Left:** From top to bottom:  $T_1$ ,  $T_2$ ,  $T_2^{flair}$ . **Middle:** Result of **NLMeans**. **Right:** Result of **CV model**.



**Figure 6.12:** Experimental results of  $G_{25}$  slice 19. Connected component labelling, **NLMeans** and **CV model**. **Left:** From top to bottom:  $T_1$ ,  $T_2$ ,  $T_2^{flair}$ , registration mask for standard parameters. **Middle:** Result of **CV model**,  $\lambda_1 = 0.1$ ,  $\lambda_2 = 5$ . **Right:** Registration mask for the **CV model** with  $\lambda_1 = 0.1$ ,  $\lambda_2 = 5$ .



---

# 7 Conclusion and Outlook

## 7.1 Conclusion

We explored in this thesis how medical image data, i.e. **MRI** data of brain tumours, can be processed. In **Chapter 2**, we showed a possibility to denoise **MRI** sequences. We made use of the **NLM** algorithm and illustrated why it is well suited to denoise medical data. Furthermore, we explained how we align the sampling rates of the **MRI** sequences.

Afterwards, we illustrated in **Chapter 3** how different **MRI** scans can be registered when they are imaged in a row. The backbone of our registration method is the outer contour of the skull.

We showed, that a similar contour of the skull is sufficient to perform an affine registration. Thus, we derived two procedures to find those contours. When the shape of the skull is similar in the different **MRI** sequences, we extract this boundary with the **CV** model.

Regrettably, the procedure is slightly complicated when additional parts of the human body are imaged, i.e. ears. We explained, how we separate the skull to align the skull's shapes in the different scans.

The boundaries of the tumour tissues are very diffusive. Consequently, there is no easy way to segment the boundaries of all parts of a glioblastoma multiforme. Additionally, there are also several drawbacks to be reported. On the one hand, our registration process fails, when it is not possible to determine in all sequences the same contour of the skull. On the other hand, finding an appropriate parameter setting for the **CVec** model with a multiphase level set representation can be tedious when it comes to necrotic tissue. Hence, our procedure to segment the tumour tissues seems to be less suitable for necrotic areas.

Nevertheless, we showed that **CV** model based segmentation methods are well suited to determine the boundaries of oedematous tissue. Finding a good parameter setting to segment the oedema is uncomplicated and the determined boundary is very accurate.

---

## 7.2 Outlook

The work pursued in this thesis can be continued in different directions. On the one hand, the principal axes transform is not able to compensate a shearing. On the other hand, the assumption that the MRI data is acquired in a row is usually wrong. We suggest therefore to use a different registration method, e.g. normalised gradient fields [24].

Furthermore, it can be very tedious to find a suitable parameter setting to segment the glioblastoma multiforme. We used two level sets for the CVec model with a multiphase level set representation. In our experience, this amount of partitioned classes is mostly not sufficient to determine the central area of necrosis. Hence, the usage of more level sets to determine the necrotic tissue could overcome this problem.

Finally, the whole process chain can be accelerated. Although we implemented most of the methods in CUDA [39], some GPU-implementations are still missing, i.e. CED, b-spline interpolation and connected components labelling.

---

# Figures

1	Different types of MRI scans .....	III
	<i>source: [22]</i>	
2	Different sequences of subject <i>G13</i> , Slice 7. ....	X
	<i>source: [22]</i>	
1.1	Aligned proton .....	1
1.2	$T_1$ relaxation rates .....	3
1.3	Different tissues in a $T_1$ weighted Image. ....	3
1.4	$T_1$ weighted Image with Contrast Agent .....	4
	<i>source: [22]</i>	
1.5	$T_2$ relaxation rates .....	4
1.6	$T_2$ weighted Image .....	5
	<i>source: [22]</i>	
1.7	$T_2^{flair}$ weighted Image .....	5
	<i>source: [22]</i>	
1.8	Image dimensions .....	6
1.9	2D Gaussian kernel .....	12
1.10	2D highpass filter .....	13
1.11	Pseudo colouring. ....	25
1.12	Colour plot for gradient magnitudes. ....	25
1.13	Colour plot of the gradient domain. ....	26
2.1	Perfect splitting in image and noise. ....	29
2.2	Method noise of Gaussian smoothing. ....	29
2.3	Method noise of Bilateral filtering. ....	31
2.4	Redundant information in a natural image. ....	32
2.5	Similarity between different patches .....	33
2.6	Method noise of <b>NLM</b> means. ....	34
2.7	Influence of $\sigma_r$ to <b>NLM</b> means. ....	36
3.1	Different sequences of subject <i>G13</i> , Slice 7. ....	37
3.2	Scheme of the Registration Process. ....	38
3.3	Skull shape of different sequences. ....	38

3.4	Synthesis functions made of B-splines. . . . .	42
3.5	Stencil weights for standard discretisation. . . . .	45
3.6	Result of <b>CED</b> filtering with a short evolution. . . . .	46
3.7	Dilation with structuring function. . . . .	49
3.8	Erosion with structuring function. . . . .	51
3.9	Opening with structuring function. . . . .	52
3.10	White Top Hat with structuring function. . . . .	54
3.11	Extraction of bright details with parabolic white top hat. . . . .	54
3.12	Approximations of $H$ and $\delta_0$ for different regularisation parameters. . . . .	57
3.13	Evolution under the <b>CV model</b> . . . . .	60
3.14	Labelling of connected components. . . . .	61
3.15	Intermediate labelling of connected component labelling. . . . .	62
3.16	Closing of small holes in the skull segment. . . . .	62
3.17	Principal axes. . . . .	64
3.18	Different transformations to align reference and template image. . . . .	66
4.1	Two curves given by the level set functions $\phi_1$ and $\phi_2$ . . . . .	70
4.2	Evolution under the <b>CVec model</b> with a multiphase level set representation. . . . .	77
4.3	Tumour boundaries detected by the <b>CVec model</b> with a multiphase level set representation. . . . .	78
5.1	Steady states of an evolution under <b>CV model</b> . . . . .	79
5.2	Method noise of <b>NLMMeans</b> for different radii $r_N$ . . . . .	80
5.3	Method noise of <b>NLMMeans</b> for different radii $r_S$ . . . . .	81
5.4	Evolution under <b>CED</b> for different evolution times. . . . .	83
5.5	Evolution under <b>CED</b> for different noise scales. . . . .	84
5.6	Evolution under <b>CED</b> for different integration scales. . . . .	85
5.7	Effect of averaging too much directional information. . . . .	85
5.8	Evolution under <b>CED</b> for different threshold parameters. . . . .	86
5.9	Evolution under <b>CED</b> for different regularisation parameters. . . . .	87
5.10	Result of white top hat filtering for different distance penaliser. . . . .	89
5.11	Bilateral filtering with different $\sigma_r$ . . . . .	90
5.12	Bilateral filtering with different $\sigma_s$ . . . . .	91
5.13	Evolution under the <b>CV model</b> for different $\lambda_1$ . . . . .	93
5.14	Evolution under the <b>CV model</b> for different $\lambda_2$ . . . . .	94
5.15	Evolution of a noisy input image under the <b>CV model</b> for different $\alpha$ . . . . .	95
5.16	Evolution under the <b>CV model</b> for different $\alpha$ . . . . .	96
5.17	Evolution under the <b>CV model</b> for different time step sizes. . . . .	97
5.18	Evolution under the <b>CV model</b> for different regularisation parameters. . . . .	98
5.19	Evolution under the <b>CV model</b> with and without reinitialisation of the level set function. . . . .	99
5.20	Evolution under the <b>CVec model</b> with a multiphase level set representation with different weights for the error terms of each channel. . . . .	100

---

5.21 Evolution under the **CVec model** with a multiphase level set representation for different combinations of channel weights. . . . . 101

5.22 Evolution under the **CVec model** with a multiphase level set representation for different combinations of channel weights. . . . . 102

6.1 Experimental results of *G13* slice 10. **NLMeans** and **CED**. . . . . 106

6.2 Experimental results of *G13* slice 10. **CED**, white top hat and bilateral filtering. . . . . 107

6.3 Experimental results of *G13* slice 10. **CV model** and connected component labelling. . . . . 108

6.4 Experimental results of *G13* slice 10. Registration, oedema boundary and necrosis boundary. . . . . 109

6.5 Experimental results of *G13* slice 8. **NLMeans** and registration mask. 110

6.6 Experimental results of *G13* slice 8. Registration, oedema boundary and necrosis boundary. . . . . 111

6.7 Experimental results of *G25* slice 6. **NLMeans** and registration mask. 112

6.8 Experimental results of *G25* slice 6. Registration and oedema boundary. . . . . 113

6.9 Experimental results of *G2* slice 8. **NLMeans** and registration mask. 114

6.10 Experimental results of *G2* slice 8. Registration and oedema boundary. . . . . 115

6.11 Experimental results of *G25* slice 19. **NLMeans** and **CV model**. . . . 116

6.12 Experimental results of *G25* slice 19. Connected component labelling, **NLMeans** and **CV model**. . . . . 117





---

# Acronyms

**BVP** Boundary Value Problem

**CED** Coherence Enhancing Diffusion

**CFT** Continuous Fourier Transform

**CV model** Chan-Vese active contour model

**CVec model** Chan-Vese active contour model for vector valued images

**DFT** Discrete Fourier Transform

**FFT** Fast Fourier Transform

**IBVP** Initial Boundary Value Problem

**IVP** Initial Value Problem

**MRI** Magnetic Resonance Imaging

**NLM** Non-Local Means

**PDE** Partial Differential Equations



# Index

## Symbols

$T_1$ .....	2, 3
$T_1$ Contrast .....	4
$T_2$ .....	4
$T_2^{flair}$ .....	5

## B

Backward Difference .....	13
Bilateral Filtering .....	30, 90

## C

Central Difference .....	14
Colour Plot .....	25
Convolution .....	9

## D

Diffusion Filtering	
Coherence Enhancing ..	44, 83
Linear .....	18
Divergence .....	15

## E

Euler-Lagrange Equation .....	19
-------------------------------	----

## F

False Positive .....	78
Forward Difference .....	13
Fourier Series .....	7
Fourier Transform .....	7

Continuous .....	7
Fourier Transform Pair .....	8
Inverse .....	7
Separability .....	8

## G

Gaussian Convolution .....	11, 16
Gaussian Smoothing . . . . .	see Gaussian Convolution
Gradient .....	14
Gradient Magnitude .....	14
Grey Matter .....	3

## H

Hemisphere .....	3
Highpass Filtering .....	11, 13

## I

Interpolation .....	39
Classical Interpolation .....	39
Generalised Interpolation . . .	40
Interpolation Condition .....	40
Synthesis Function .....	39

## L

Laplace Operator .....	15
Lowpass Filtering .....	11

## M

Magnetic Resonance Imaging . .	1, 2,
--------------------------------	-------

	<b>37</b>
Maximum-Minimum Principle . . .	<b>18</b>
Minimal Partition Problem . . . . .	<b>22</b>
Mumford-Shah Functional . . . . .	<b>22</b>
Piecewise constant . . . . .	<b>22</b>
Myelin Layer . . . . .	<b>3</b>

## N

---

Non-Local Means . . . . .	<b>32, 79</b>
Nonflat Morphology . . . . .	<b>46</b>
Dilation . . . . .	<b>48</b>
Erosion . . . . .	<b>50</b>
Opening . . . . .	<b>51</b>
White Top Hat . . . . .	<b>53, 88</b>

## P

---

Partial Derivative . . . . .	<b>13</b>
Partial Differential Equation . . .	<b>13</b>
Pseudo Colouring . . . . .	<b>25</b>

## R

---

Range Filter . . . . .	<b>30</b>
------------------------	-----------

## S

---

Segmentation	
Manual . . . . .	<b>67</b>
Thresholding . . . . .	<b>67</b>
Structure Tensor . . . . .	<b>16</b>

## T

---

Thomas Algorithm . . . . .	<b>23</b>
----------------------------	-----------

## V

---

Visualisation . . . . .	<b>25</b>
-------------------------	-----------

---

## W

---

White Matter . . . . .	<b>3</b>
------------------------	----------

# References

- [1] N. M. ALPERT, J.F. BRADSHAW, D. KENNEDY AND J.A. CORREIA.  
*The Principal Axes Transformation - A Method for Image Registration.*  
 Journal of Nuclear Medicine  
 no. 31: 1717–1723 (1990) ..... 63
- [2] V. AURICH AND J. WEULE.  
*Non-Linear Gaussian Filters Performing Edge Preserving Diffusion.*  
 Mustererkennung  
 Springer, Berlin Heidelberg  
 538–545 (1995) ..... 30
- [3] D. BARTZ AND B. PREIM.  
*Visualization in Medicine: Theory, Algorithms and Applications.*  
 Morgan Kaufmann, San Francisco  
 ed. 1: 48–54 (2007) ..... 1, 2
- [4] K. BERGER.  
*The Developing Person Through the Lifespan.*  
 Worth Publishers, New York  
 ed. 7 (2007) ..... 3
- [5] R. VAN DEN BOOMGAARD.  
*The Morphological Equivalent of the Gauss Convolution.*  
 Nieuw Archief voor Wiskunde  
 vol. 10, no. 3: 219–236 (1992) ..... 47, 48
- [6] R. W. BROCKETT AND P. MARAGOS.  
*Evolution Equations for Continuous-Scale Morphology.*  
 IEEE International Conference on Acoustics, Speech, and Signal Processing  
 vol. 3: 125–128 (1992) ..... 48
- [7] A. BUADES, B. COLL AND J.-M. MOREL.  
*A non-local Algorithm for Image Denoising.*

- In Proc. of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition  
vol. 2: 60–65 (2005) ..... 32, 33
- [8] **A. BUADES, B. COLL AND J.-M. MOREL.**  
*A Review of Image Denoising Algorithms, with a new one.*  
SIAM Multiscale Modeling and Simulation  
vol. 4, no. 2: 490–530 (2005) ..... 28, 32
- [9] **V. CASELLES, F. CATTÉ, T. COLL AND F. DIBOS.**  
*A geometric model for active contours in image processing.*  
Numerische Mathematik  
vol. 66, no. 1: 1–30 (1993) ..... 21
- [10] **V. CASELLES, R. KIMMEL AND GUILLERMO SAPIRO.**  
*Geodesic Active Contours.*  
International Journal of Computer Vision  
vol. 22, no. 1: 61–79 (1997) ..... 20, 21
- [11] **T. F. CHAN, Y. SANDBERG AND L. A. VESE.**  
*Active contours without edges for vector-valued images.*  
Journal of Visual Communication and Image Representation  
vol. 11, no. 2: 130–141 (2000) ..... 22, 59, 68, 69, 79, 94, 99
- [12] **T. F. CHAN AND L. A. VESE.**  
*Image segmentation using level sets and the piecewise-constant Mumford-Shah model.*  
Computational Applied Math Group  
Tech. Rep. 0014 (2000) ..... 22, 70, 71, 72, 73, 74, 76, 99
- [13] **T. F. CHAN AND L. A. VESE.**  
*Active Contours without Edges.*  
IEEE Transactions on Image Processing  
vol. 10, no. 2: 266–277 (2001) ..... 20, 22, 55, 56, 57, 58, 59, 75, 99
- [14] **T. F. CHAN AND L. A. VESE.**  
*Active contour and segmentation models using geometric PDEs for medical imaging.*  
Geometric methods in bio-medical image processing  
Springer Berlin Heidelberg  
63–75 (2002) ..... 22, 70, 71, 72, 73, 76, 99
-

- [15] **N. CHUMCHOB AND K. CHEN.**  
*A robust affine image registration method.*  
 International Journal Of Numerical Analysis And Modeling  
 vol. 6, no. 2: 311–334 (2009) ..... 63
- [16] **I. DIAZ, P. BOULANGER, R. GREINER AND A. MURTHA.**  
*A critical review of the effects of de-noising algorithms on MRI brain tumor segmentation.*  
 Annual International Conference of the IEEE Engineering in  
 Medicine and Biology Society  
 3934–3937 (2011) ..... 27
- [17] **M. ELAD AND M. AHARON.**  
*Image denoising via sparse and redundant representations over learned dictionaries.*  
 IEEE Transactions on Image Processing  
 vol. 15, no. 12: 3736–3745 (2006) ..... 27
- [18] **W. FELTUS AND G. LIEBERMAN.**  
*The Evolution of Magnetic Resonance Imaging for Glioblastoma Multiforme (GBM): From Diagnosis to Image-Guided Treatment.*  
 Beth Israel Deaconess Medical Center  
 (2012) ..... 5
- [19] **J. B. J. FOURIER.**  
*The Analytical Theory of Heat.*  
 The University Press  
 (1878) ..... 7
- [20] **P. GERLEE AND S. NELANDER.**  
*The impact of phenotypic switching on glioblastoma growth and invasion.*  
 PLOS Computational Biology  
 vol. 8, no. 6 (2012) ..... 68
- [21] **R. C. GONZÁLEZ AND R. E. WOODS.**  
*Digital Image Processing.*  
 Pearson/Prentice Hall  
 ed. 3 (2008) ..... 6, 7, 8, 11
- [22] **N. GRAF**  
 Department of Nuclear Medicine, Saarland University ..... 121

- 
- [23] S. GREWENIG, J. WEICKERT, C. SCHROERS AND A. BRUHN.  
*Cyclic Schemes for PDE-Based Image Analysis.*  
 Technical Report No. 327, Mathematical Image Analysis Group,  
 Saarland University  
 March (2013) ..... 43, 45
- [24] E. HABER AND J. MODERSITZKI.  
*Intensity gradient based registration and fusion of multi-modal im-  
 ages.*  
 Medical Image Computing and Computer-Assisted Intervention  
 Springer Berlin Heidelberg  
 726–733 (2006) ..... 63, 120
- [25] B. JÄHNE.  
*Spatio-Temporal Image Processing - Theory and Scientific Applica-  
 tions.*  
 Lecture Notes in Computer Science  
 Springer vol. 751 (1993) ..... 16
- [26] M. KASS, A. WITKIN AND D. TERZOPOULOS.  
*Snakes: Active contour models.*  
 International Journal of Computer Vision  
 vol. 1, ed. 4: 321–331 (1988) ..... 20
- [27] P.J. KOSTELEK AND S. PERIASWAMY.  
*Image Registration for MRI.*  
 Modern Signal Processing.  
 vol. 46: 161–184 (2003) ..... 63, 64
- [28] P. C. LAUTERBUR.  
*Image Formation by Induced Local Interactions: Examples employing  
 Nuclear Magnetic Resonance.*  
 Nature  
 vol. 242, no. 5394: 190–191 (1973) ..... 1
- [29] H. LI, J. RUAN, Z. XIE AND H. WANG.  
*Investigation of the critical geometric characteristics of living human  
 skulls utilising medical image analysis techniques.*  
 International Journal of Vehicle Safety  
 vol. 2, no. 4: 345–367 (2007) ..... 92
- [30] D. N. LOUIS, H. OHGAKI, O. D. WIESTLER, W. K. CAVENEE, P.C.  
 BURGER AND A. JOUVET.  
*The 2007 WHO Classification of Tumours of the Central Nervous*
-

- System.*  
Acta Neuropathologica  
vol. 114 no. 2: 97–109 (2007) ..... IX
- [31] **M. MAINBERGER.**  
*Correspondence Problems in Computer Vision. Lecture Slides.*  
Mathematical Image Analysis Group, Saarland University  
(2012) ..... 25
- [32] **J. V. MANJÓN, J. CARBONELL-CABALLERO, J. J. LULL, G. GARCA-MART, L. MARTI-BONMATI AND M. ROBLES.**  
*MRI denoising using non-local means.*  
Medical image analysis  
vol. 12, no. 4: 514–523 (2008) ..... 79
- [33] **J. C. MARSH, J. GOLDFARB, T. D. SHAFMAN AND A. Z. DIAZ.**  
*Current Status of Immunotherapy and Gene Therapy for high-grade Gliomas.*  
Cancer Control  
vol. 20 no. 1: 43–48 (2013) ..... IX
- [34] **G. MATHERON AND J. SERRA**  
*The Birth of Mathematical Morphology.*  
ISMM 2000, Xeros Center Palo-Alto  
(2000) ..... 46
- [35] **G. P. MAZZARA, R. P. VELTHUIZEN, J. L. PEARLMAN, H. M. GREENBERG AND H. WAGNER.**  
*Brain tumor target volume determination for radiation treatment planning through automated MRI segmentation.*  
International Journal of Radiation, Oncology, Biology and Physics  
vol. 59, no. 1: 300–312 (2004) ..... 67
- [36] **J. MODERSITZKI**  
*Numerical Methods for Image Registration.*  
Oxford University Press  
(2004) ..... 64
- [37] **J. MUMFORD AND D. SHAH**  
*Boundary detection by minimizing functionals.*  
IEEE Conference on Computer Vision and Pattern Recognition  
San Francisco  
(1985) ..... 20, 22

- 
- [38] NVIDIA CUDA FAST FOURIER TRANSFORM LIBRARY  
<https://developer.nvidia.com/cuFFT> ..... 10
- [39] NVIDIA CUDA TOOLKIT  
<https://developer.nvidia.com/cuda-toolkit> ..... 36, 54, 60, 120
- [40] H. NYQUIST.  
*Certain topics in telegraph transmission theory.*  
 Transactions of the American Institute of Electrical Engineers  
 vol. 47 no. 2: 617–644 (1928) ..... 6
- [41] S. OSHER AND J. SETHIAN.  
*Fronts propagating with curvature-dependent speed: algorithms based  
 on Hamilton-Jacobi formulations.*  
 Journal of Computational Physics  
 vol. 79 no. 1: 12–49 (1988) ..... 20
- [42] P. PERONA AND J. MALIK.  
*Scale-space and edge detection using anisotropic diffusion.*  
 IEEE Transactions on Pattern Analysis and Machine Intelligence  
 vol. 12 no. 7: 629–639 (1990) ..... 21, 27
- [43] M. PRASTAWA, E. BULLITT, N. MOON, K. VAN LEEMPUT AND G.  
 GERIG.  
*Automatic brain tumor segmentation by subject specific modification  
 of atlas priors.*  
 Academic Radiology  
 vol. 10 no. 12: 1341–1348 (2003) ..... 67
- [44] I. I. RABI, J. R. ZACHARIAS, S. MILLMAN AND P. KUSCH.  
*A New Method of Measuring Nuclear Magnetic Moment.*  
 Physical Review  
 vol. 53, no. 4: 318–327 (1938) ..... 1
- [45] S. RJASANOW.  
*Numerics of Partial Differential Equations. Lecture Slides.*  
 Institute for Applied Mathematics, Saarland University  
 (2013) ..... 14, 17, 56
- [46] A. ROSENFELD AND J. L. PFALTZ.  
*Sequential operations in digital picture processing.*  
 Journal of the ACM  
 vol. 13, no. 4: 471–494 (1966) ..... 61
-

- [47] **S. ROTH AND M. J. BLACK.**  
*Fields of experts: A framework for learning image priors.*  
 IEEE Computer Society Conference on Computer Vision and Pattern Recognition  
 vol. 2: 860–867 (2005) ..... 27
- [48] **O. SCHMITT, J. MODERSITZKI, S. HELDMANN, S. WIRTZ AND B. FISCHER.**  
*Image registration of sectioned brains.*  
 International Journal of Computer Vision  
 vol. 73, no. 1: 5–39 (2007) ..... 64
- [49] **J. SERRA.**  
*Image Analysis and Mathematical Morphology.*  
 Mathematics, Technique Staining Microscopy  
 Academic Press  
 vol. 131 (1982) ..... 46
- [50] **S. SETZER.**  
*Image Acquisition Methods. Lecture Slides.*  
 Mathematical Image Analysis Group, Saarland University  
 (2012) ..... 2, 4, 5
- [51] **C. E. SHANNON.**  
*Communication in the Presence of Noise.*  
 Proceedings of the IRE  
 vol. 37, no. 1: 10–21 (1949) ..... 6
- [52] **P. THÉVENAZ, T. BLU AND M. UNSER.**  
*Interpolation revisited (medical images application).*  
 IEEE Transactions on Medical Imaging  
 vol. 19, no. 7: 739–758 (2000) ..... 40, 41, 42, 43
- [53] **P. THÉVENAZ, T. BLU AND M. UNSER.**  
*Image Interpolation and Resampling.*  
 Handbook of Medical Imaging, Processing and Analysis  
 Academic Press  
 393–420 (2000) ..... 39, 40, 41, 42, 43
- [54] **L. H. THOMAS.**  
*Elliptic Problems in Linear Difference Equations over a Network.*  
 Technical Report  
 Watson Scientific Computing Laboratory, Columbia University  
 vol. 19, no. 7: 739–758 (1949) ..... 23

- [55] C. TOMASI AND R. MANDUCHI.  
*Bilateral Filtering for Gray and Color Images.*  
IEEE International Conference on Computer Vision  
839–846 (1998) ..... 30
- [56] M. TREPEL.  
*Neuroanatomie: Struktur und Funktion.*  
Urban & Fischer Verlag / Elsevier GmbH  
ed. 5 (2011) ..... X, 4, 5
- [57] M. UNSER, A. AKRAM AND E. MURRAY.  
*B-spline signal processing. I. Theory.*  
IEEE Transactions on Signal Processing  
vol. 41, no. 2: 821–833 (1993) ..... 42
- [58] M. UNSER, A. AKRAM AND E. MURRAY.  
*B-spline signal processing. II. Efficiency design and applications.*  
IEEE Transactions on Signal Processing  
vol. 41, no. 2: 834–848 (1993) ..... 42
- [59] P. A. VIOLA.  
*Alignment by Maximization of Mutual Information.*  
Ph.D. thesis  
Massachusetts Institute of Technology (1995) ..... 63
- [60] M. VAN VULPEN, H. B. KAL, M. J. TAPHOORN AND S. Y. EL SHAROUNI.  
*Changes in Blood-Brain Barrier Permeability induced by Radiotherapy: Implications for Timing of Chemotherapy?(Review).*  
Oncology reports  
vol. 9, no. 4: 683–688 (2002) ..... IX
- [61] J. WEICKERT.  
*Anisotropic Diffusion in Image Processing.*  
Teubner, Stuttgart  
(1998) ..... 17, 18, 19, 44, 46, 47
- [62] J. WEICKERT, B.M. TER HAAR ROMENY AND M.A. VIERGEVER.  
*Efficient and reliable schemes for nonlinear diffusion filtering.*  
IEEE Transactions on Image Processing  
vol. 7, no. 3: 398–410 (1998) ..... 23
-

- [63] J. WEICKERT.  
*Coherence-enhancing diffusion filtering.*  
International Journal of Computer Vision  
vol. 31: 111–127 (1999) ..... 16, 17, 44, 83, 84, 85, 86, 87
- [64] J. WEICKERT.  
*Coherence-enhancing diffusion of colour images.*  
Image and Vision Computing  
vol. 17 n. 3: 201–212 (1999) ..... 17, 44, 87
- [65] J. WEICKERT.  
*Coherence-Enhancing Shock Filters.*  
Lecture Notes in Computer Science  
vol. 2781: 1–8 (2003) ..... 17
- [66] J. WEICKERT.  
*Image Processing and Computer Vision. Lecture Slides.*  
Mathematical Image Analysis Group, Saarland University  
(2012) ..... 8, 9, 10, 11, 13, 25, 27, 39, 40, 51, 58, 67, 68
- [67] J. WEICKERT.  
*Differential Equations in Image Processing and Computer Vision.*  
*Lecture Slides.*  
Mathematical Image Analysis Group, Saarland University  
(2013) ..... 12, 16, 19, 21, 27, 48
- [68] J. WEICKERT.  
*Private Talk*  
Mathematical Image Analysis Group, Saarland University  
(2013) ..... 48
- [69] D. WEISHAUP, V. D. KÖCHLI AND B. MARINCEK.  
*Wie funktioniert MRI?*  
Springer, Stuttgart  
ed. 6 (2009) ..... 2, 3, 4, 5
- [70] D. R. WHITE, A. S. HOUSTON, W. F. SAMPSON AND G. P. WILKINS.  
*Intra-and interoperator variations in region-of-interest drawing and their effect on the measurement of glomerular filtration rates.*  
Clinical nuclear medicine  
vol. 24, no. 3: 177–181 (1999) ..... 67

- [71] H. ZHAO, T. CHAN, B. MERRIMAN AND S. OSHER.  
*A variational level set approach to multiphase motion.*  
Journal of Computational Physics  
vol. 127, ed.1: 179–195 (1996) .....56
-