FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Semi-Automatic Spine Labeling on T1- and T2-weighted MRI Volume Data

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Medizinische Informatik

eingereicht von

Maria Wimmer
Matrikelnummer 0725248

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Mitwirkung: Dipl.-Math. Dr.techn. Katja Bühler, VRVis

Wien, 20.01.2015

(Unterschrift Verfasserin) (Unterschrift Betreuung)

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Semi-Automatic Spine Labeling on T1- and T2-weighted MRI Volume Data

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Medical Informatics

by

Maria Wimmer
Registration Number 0725248

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller
Assistance: Dipl.-Math. Dr.techn. Katja Bühler, VRVis

Vienna, 20.01.2015

(Signature of Author) (Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Maria Wimmer
Niederfraunleiten 4, 4490 St. Florian

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

(Ort, Datum) (Unterschrift Verfasserin)

Acknowledgements

From the Institute of Computer Graphics, I want to thank Eduard Gröller for accepting me as a master student and for reviewing my work.

I want to thank Katja Bühler for giving me the opportunity to carry out this thesis at the VRVis Research Center. Thank you for supporting and reviewing my work and for providing me with the necessary facilities. I thank my colleagues from the Biomedical Visualization Group for their inspiring ideas and valuable discussions. Special thanks to David Major, for introducing me to the topic and reviewing my thesis.

I like to thank my colleagues in the diploma student office and workmates also from other areas at VRVis, who became friends along the way. Especially Harald and Michael, for the good mood you spread and our little talks in coffee breaks. Thank you Michael for your advice of all kinds!

Thanks to my friends from the university, my former student housing and my hometown. Thank you for accompanying me through my studies, not only at the university. Special thanks to the "Medizinischer Informatik Stammtisch" for the good times we spend together and for showing me, that writing a thesis always takes longer than expected.

Thank you Stephan, for your support, help and your open ears at any time – no matter where I was studying.

Thank you Markus, for reminding and encouraging me to step out, *to see the world* and that the last limit is oneself – not only when writing a thesis.

Thank you Dominik, for your belief in me, for listening and motivating me. Thank you for your support in all kinds, especially in the final, demanding phase of my thesis.

Thank you Astrid, Marie-Luise and Sabrina, for your encouraging words whenever I need(ed) them. Above all, thank you for your friendship! I am very happy to be a part of this "fantastic four" dream team.

Special *Thank you* to Edith, for accompanying me the last couple of years side-by-side. Not only at the university, but most important as very good friend. Thank you for your help and support in every belonging!

Finally, I want to express all my gratitude to my parents Maria and Franz, my brother Christian and all my grandparents and family. Thank you for supporting me during my education. Above all, thanks for your patience while I was writing this thesis, for encouraging me and believing in me.

Abstract

In medical diagnosis, the spine is often a frame of reference and so helps to localize diseases (e.g. tumors) in the human body. Automated spine labeling approaches are in demand, in order to replace time consuming, manual labeling by a radiologist. Different approaches have already been proposed in the literature, mainly for Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) data. While CT scans exhibit a generalized intensity scale, MR images come with a high variability within the data and hence the tissues. Several factors influence the appearance of vertebrae and intervertebral disks in MRI data: different scanners, changes of acquisition parameters, magnetic field inhomogeneities or age-related, structural changes of the spinal anatomy. These factors compound the development of semi- and fully automatic spine labeling systems.

The main goal of this thesis is to overcome these variations and find a generalized representation for different kinds of MR data. Furthermore, it aims for a semi-automatic labeling approach on these preprocessed scans where the user has to provide an initial click. Entropy-optimized Texture Models are applied to normalize the data to a standardized, reduced intensity scale. With Probabilistic Boosting Trees, intervertebral disk feature points are detected, whereby the disk center is selected with a Shape Particle Filter.

The results achieved with the proposed pipeline are promising in terms of data normalization, timing and labeling accuracy. With a mean overall processing time of 6.0 s for normalizing and labeling a dataset (0.8 s per disk), the algorithm achieves a precision of 92.4% (recall = 86.8%). Using a higher resolution of the data for disk detection (average timing of 1.6 s per disk resp. 12.4 s per dataset), reduces the number of missed disk candidates and hence increases the recall to 91.7% (with a precision of 91.9%).

Kurzfassung

Die Wirbelsäule wird in der medizinischen Diagnose oft als Referenz verwendet und unterstützt so die Lokalisierung von Krankheiten (z.B. Tumoren). Verfahren, die der Wirbelsäule automatisch ihre anatomischen Bezeichnungen zuordnen sind gefragt, um zeitraubendes manuelles Beschriften durch einen Radiologen einzusparen. Verschiedene Algorithmen wurden bisher in der Literatur vorgestellt, wobei die meisten mit Computertomographie- (CT) und Magnetresonanztomographie-Daten (MRI) arbeiten. Während CT-Daten eine einheitliche Intensitätsskala aufweisen, treten in MR-Daten hohe Variabilitäten innerhalb der Daten und folglich innerhalb der Gewebe auf. Verschiedene Faktoren beeinflussen das Aussehen von Wirbeln und Bandscheiben in MRI-Bildern: unterschiedliche Scanner, Änderungen in den Aufnahmeparametern, Inhomogenitäten des Magnetfeldes oder altersbedingte Änderungen in der Struktur der Wirbelsäulenanatomie. Diese Faktoren erschweren die Entwicklung von halb- und vollautomatischen Systemen, die der Wirbelsäule ihre anatomischen Beschriftungen zuordnen.

Hauptziel dieser Diplomarbeit ist es, diese Variabilität zu überwinden und eine generalisierte Repräsentation für verschiedene Arten von MR-Daten zu finden. Weiters soll auf diesen normalisierten Bildern die Anatomie der Wirbelsäule beschriftet werden, wobei ein Benutzer einen Startpunkt zur Verfügung stellt. Entropie-optimierte Texturmodelle werden verwendet, um die Daten zu normalisieren und in eine standardisierte, reduzierte Intensitätsskala umzuwandeln. Mit Probabilistic Boosting Trees werden mögliche Punkte innerhalb der Bandscheiben detektiert, wobei der Mittelpunkt der Bandscheibe mit Hilfe eines Shape Particle Filters ausgewählt wird.

Die Ergebnisse, die mit der vorgestellten Methode erzielt werden, sind vielversprechend hinsichtlich der Datennormalisierung, benötigten Zeit und erzielten Beschriftungs-Genauigkeit. Mit einer mittleren gesamten Verarbeitungszeit von 6.0 s für Normalisierung und Beschriftung eines Datensatzes (0.8 s pro Bandscheibe) liefert der Algorithmus eine Genauigkeit von 92.4% (Trefferquote = 86.8%). Verwendet man eine höhere Auflösung der Daten für die Detektierung der Bandscheiben (durchschnittliche Verarbeitungszeit von 1.6 s pro Bandscheibe beziehungsweise 12.4 s pro Datensatz), so reduziert sich die Anzahl nicht gefundener Bandscheiben und erhöht somit die Trefferquote auf 91.7% (mit einer Genauigkeit von 91.9%).

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

The spinal column forms an important part in the human body. It is responsible for the body's upright position and enables movement of the torso. Moreover, it protects the spinal cord – a bundle of nerves which runs from the brain to the rest of the body [17].

It is often a target of medical imaging, e.g. with X-ray, Computed Tomography (CT) or Magnetic Resonance Imaging (MRI), due to several reasons. On one hand, back pain is among the most common diseases. According to a survey from Statistik Austria [3], about 2.3 million Austrians suffer from chronic problems with the spine. Causes for back pain are e.g. herniated disks or fractured vertebrae (see Figure 1.1) [13]. On the other hand, the spinal column is usually visible if images of the abdomen or upper body are acquired.

For medical diagnosis, the spinal column is a frame of reference in the human body when it comes to describing the location of an organ or a pathology. Hence it is important, that a reliable anatomical labeling of the spine is available. Within this scope, it is sufficient to localize disks and vertebrae and provide their corresponding anatomical label (e.g. $L4$ for the forth lumbar vertebra). However, manual placing of landmarks is time consuming and tedious, especially if only parts of the spinal column are visible on the scans. Radiologists have to ensure the quality of scans and correctness of the labeling on the acquired datasets. Hence a reliable computer-aided system for automatic detection of landmarks and labeling right after acquisition is in demand [55].

1.1 Motivation for Spine Labeling

Several reasons and fields of application motivate the development of spine labeling algorithms.

The rough localization of the spinal column can be employed as reference for full segmentation of organs in the abdomen [66]. The spinal column is detected first in scans and then used to derive the relative location of organs based on a statistical model.

Another field of application is automated scan planning. Usually the planning of scans is carried out by an operator on a low resolution scan acquired in advance. For every examination,

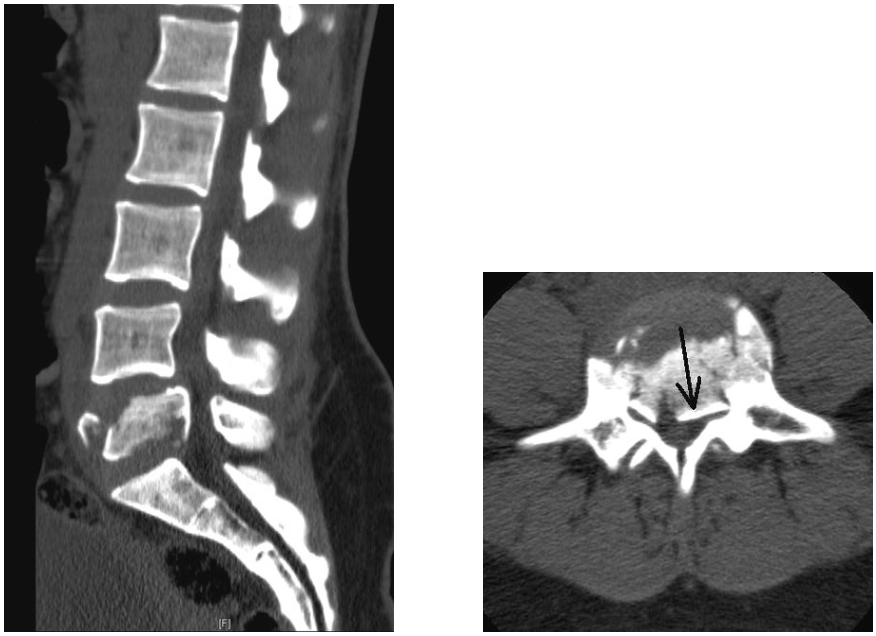


Figure 1.1: Lateral (from the side) view of the spinal column with a fractured fifth lumbar vertebrae (left). Axial (from the top) view of the fracture (right). The arrow indicates a part of the vertebra that presses into the spinal canal. Image from web page [13].

she manually determines scan parameters. The aim of automated planning tools is to ensure consistency across the acquisitions and hence improve the workflow. Therefore a robust labeling of the spinal anatomy is needed, so that the necessary parameters are derived correctly [47].

Besides localization of spinal interest points, there are use cases for full segmentation of the spinal anatomy as well. Segmentation algorithms that separate bones from soft tissue are in use for example within computer-assisted surgery of the spinal column [26], [9]. From segmented parts, a precise 3D model of the anatomy is reconstructed and shown to the surgeon in order to assist at the intervention. Full segmented intervertebral disks and vertebrae furthermore enable pathologic anatomical assessment of these [42].

Generally speaking, localization and full segmentation approaches of the spinal anatomy are of use in Computer Aided Diagnosis (CAD) systems, which support clinical decision making processes. With these, pathologies of the spinal column, disks or vertebrae can be detected by automated analysis of the recognized parts [64]. From segmented vertebrae, measures such as height and density can be derived, which give information about possible osteoporosis. Spine disorders, such as disk degeneration or pathological curvatures can be detected and monitored by the analysis of metrics (e.g. thickness or volumes) [41].

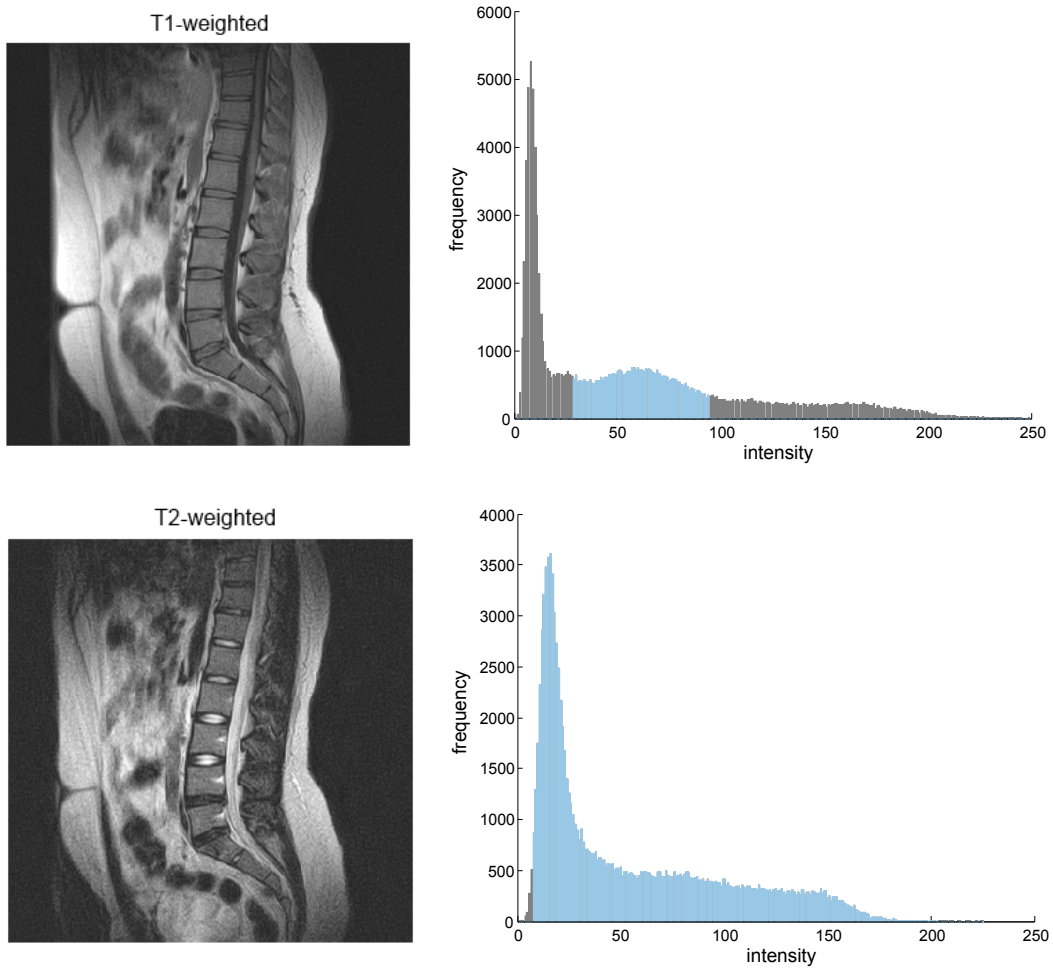


Figure 1.2: T1-weighted and T2-weighted slice image from volume d13_1_t1w respectively d13_1_t2w (left) and corresponding histograms (right). The intensity range of intervertebral disks in the dataset is depicted by light blue bars. Data by courtesy of AGFA HealthCare [25].

1.2 Problem Statement

Numerous spine labeling algorithms have been designed on various imaging modalities, including X-Ray, CT and MRI. Within CT data, the intensities follow a standardized scale [58]. Unfortunately this is not the case for MR datasets. Different scans exhibit different intensity ranges, due to changes in the acquisition protocol. This results in varying intensities for the same tissue across different scans. Furthermore, noise and low contrast in the images cause these variations within a single slice. These factors, as well as a poor signal-to-noise ratio make generalized image processing tasks more complex on MR data [26].

Figure 1.2 illustrates intensity differences for two groups of MR scans, namely *T1-weighted*

(T1w) and *T2-weighted (T2w)* scans. They are acquired on the same subject, but with a different acquisition parameter setup. This causes varying gray values *within* intervertebral disks between both scans. The global intensity ranges for the images are $t1 = [0, 249]$ for the T1w and $t2 = [0, 225]$ for the T2w scan (gray area). Looking at the disk intensities (blue area), a huge difference is seen immediately. They lie within the range $t1 = [29, 94]$ and $t2 = [8, 203]$, respectively.

Algorithms designed for one modality are (usually) not directly applicable to another one. Models and algorithms have to be retrained for their application on different modalities [32]. In the literature, one work mentions that it was trained on a set of MR scans and is also directly applicable to CT scans without retraining [36]. However, for MR data they focus on T2w scans and do not report results for other types of MR data, such as T1w images.

Currently state-of-the-art spine detection and labeling algorithms usually focus on a specific set of MR scans, e.g. T1w or T2w scans. This motivates for a general approach, that is applicable to several kinds of MR datasets without a necessary retraining of feature detection and labeling algorithms. The goal of this thesis is to design an algorithm, that performs semi-automatic spine labeling (with an initial click from a user) on a range of MR scans. To be more specific, the system should be able to process the two main groups of T1w and T2w scans without knowing, which kind of dataset is currently processed. To accomplish this goal, the following two processing stages are integrated:

1. Normalization of the input data with an *Entropy-optimized Texture Model (ETM)*, which was proposed by Zambal et al. [67]: From sparse annotations covering the lumbar spine, a model is learned for normalizing MR scans. The goal is to reduce the number of present gray values in the data to a fixed number of s target values without losing relevant anatomical information.

2. Feature detection and labeling of the spinal anatomy: Starting from a click position, disk candidate voxels are extracted on the normalized data with *Probabilistic Boosting Trees (PBTs)* in an iterative manner. The best voxel is selected as disk center point.

1.3 Thesis Overview

The thesis is organized as follows: At the end of this chapter, the mathematical notation used in this thesis is introduced. Chapter 2 gives a detailed overview about the anatomy of the human spine. Furthermore imaging modalities on the spinal column are introduced, whereby the focus lies on MRI. Chapter 3 reviews related work on spine labeling algorithms and summarizes preprocessing methods. The theoretical background of the methods integrated in this work is given in Chapter 4. This includes on one hand algorithms for preprocessing and data normalization, and on the other hand feature detection and classification algorithms. Implementation and integration of those is detailed in Chapter 5. Data normalization, disk detection and labeling results on lumbar datasets are discussed in Chapter 6. Finally, Chapter 7 concludes this work and gives an outlook to future work.

1.4 Mathematical Notation

The mathematical notation used in this work is summarized in the following, with respect to its field of application.

Data and Annotations

$\mathcal{S}$	annotated set of datasets (2D or 3D)
$\mathcal{S}_{tr} \subset \mathcal{S}$	annotated set of training datasets
I_k	k-th dataset of $\mathcal{S}$, also referred to as image data, i.e. a 2D image or 3D volume
Δ_x^k	voxel size of I_k in x-direction
λ_i	anatomical disk label, whereby $\lambda_i \in \{T8/T9, T9/T10, \ldots, L4/L5, L5/S1\}$
κ_j	anatomical vertebra label, whereby $\kappa_j \in \{T9, T10, \ldots, L5, S1\}$
d_i^k	intervertebral disk center i with label λ_i in dataset I_k
c_i^k	spinal canal landmark i with label λ_i in dataset I_k
z_i^k	cylinder i in dataset I_k, i.e. the cylinder placed around disk center d_i^k
v_j^k	vertebra body center j in dataset I_k

Entropy-Optimized Texture Models

$m = \lvert\mathcal{S}_{tr}\rvert$	number of training datasets
l	number of corresponding landmark points
T_k	texture extracted from image data I_k
N	number of extracted texel
t_j	model texel, whereby $j = 1 \ldots N$
r_k	number of source values (source bins), whereby $k = 1 \ldots m$
s	number of target values (target bins)
g_i'	mapped target value, whereby $g_i' \in \{1 \ldots s\}$
f_k	mapping from r_k source values to s target values, whereby $k = 1 \ldots m$
p_j	probability density function (PDF) of model texel t_j
$H(p_j)$	information entropy of corresponding PDF p_j
$H\left(f_k\left(T_k\right)\right)$	image entropy of training texture T_k
H^{tex}	entropy of entire training texture set
H^{model}	model entropy
S	shape instance
U	texture $U = (u_1, \ldots, u_N)$ of unseen dataset, that is currently overlapped from the model
U'	normalized texture U, i.e. mapped to s target bins
$P\left(S\right)$	prior probability of shape instance S
$P\left(U'\middle\lvert S\right)$	likelihood of the observed normalized texture U', given the shape instance S
sd	sampling distance
β	downsampling factor
$\mathcal{M}_i$	trained ETM
$mode_d(\mathcal{M}_i)$	mode of intervertebral disk histogram in model $\mathcal{M}_i$
$mode_v(\mathcal{M}_i)$	mode of vertebrae histogram in model $\mathcal{M}_i$

Bias Field Correction

B_k	background image extracted from image I_k
F_k	estimated bias field from image I_k
I_k^*	corrected image I_k
g_i	gray value of selected point i, whereby $i = 1 \ldots n$
$h\,(x, y; a_1, \ldots a_m)$	Function h, fitted to data points (x, y) with coefficients $a_1, \ldots, a_m$

Feature Detection and Labeling

$II\,(x, y)$	integral image
∇I_x	partial differentiation in x-direction
$\nabla I'_x$	approximated partial derivative in x-direction, obtained by filtering
G_{xy}	gradient magnitude of partial derivatives $\nabla I'_x$ and $\nabla I'_y$
M_x	filter mask in x-direction
W	number of resolution levels
C_w	classifier C on resolution level w, whereby $w \in \{0 \ldots W\}$
Φ_L^w	detector for lumbar disks at resolution level w
Φ_T^w	detector for thoracic disks at resolution level w

CHAPTER 2

Medical Background

This chapter aims to give an introduction into the anatomy and pathologies of the human spine and medical imaging techniques related to it. Section 2.1 describes the functionality and general anatomy of the spinal column, as well as anatomical and possible pathological variations. Commonly used imaging techniques for the spine are explained in Section 2.2. Finally, Section 2.3 introduces MRI and its application on the spinal column.

2.1 The Spine

The *spine* (lat.: *columna vertebralis*), also called *vertebral column* or *spinal column*, is characteristic for all vertebrate. It is located in the back of the torso and functions as support for the torso. The spine is a bony chain of vertebrae, which is stabilized with ligaments and muscles. The muscles enable movement of the torso. Soft cushions called *intervertebral disks* act as ligaments between two adjacent vertebrae [17].

Besides supporting the torso and enabling movement of it, the spine has another important task to fulfill. It protects the central nervous system in several ways: On one hand, the disks enable elasticity and suspension of the vertebral column, which is important to protect the brain from concussion. On the other hand, the chain of vertebrae forms the spinal canal, which holds and protects the spinal cord [35], [17].

The vertebral column usually consists of 24 free vertebrae, which are connected by 23 intervertebral disks. The spine is grouped in three main regions (from top to bottom): the *cervical*, *thoracic* and *lumbar* spine. These parts are followed by two additional ones: the *sacrum* and *coccyx*. The sacrum is a fusion of vertebrae, which is flexibly connected with the fifth lumbar vertebra. The bottommost part of the spine is the coccyx. Its small, bony elements are a rudiment of a tail. That is why this part is also called tailbone [17]. Table 2.1 and Figure 2.1 give an overview about the regions described.

Different sizes and heights of vertebrae and intervertebral disks give the spine its typical form (see lateral view in Figure 2.1). The cervical and lumbar region have an inward curvature,

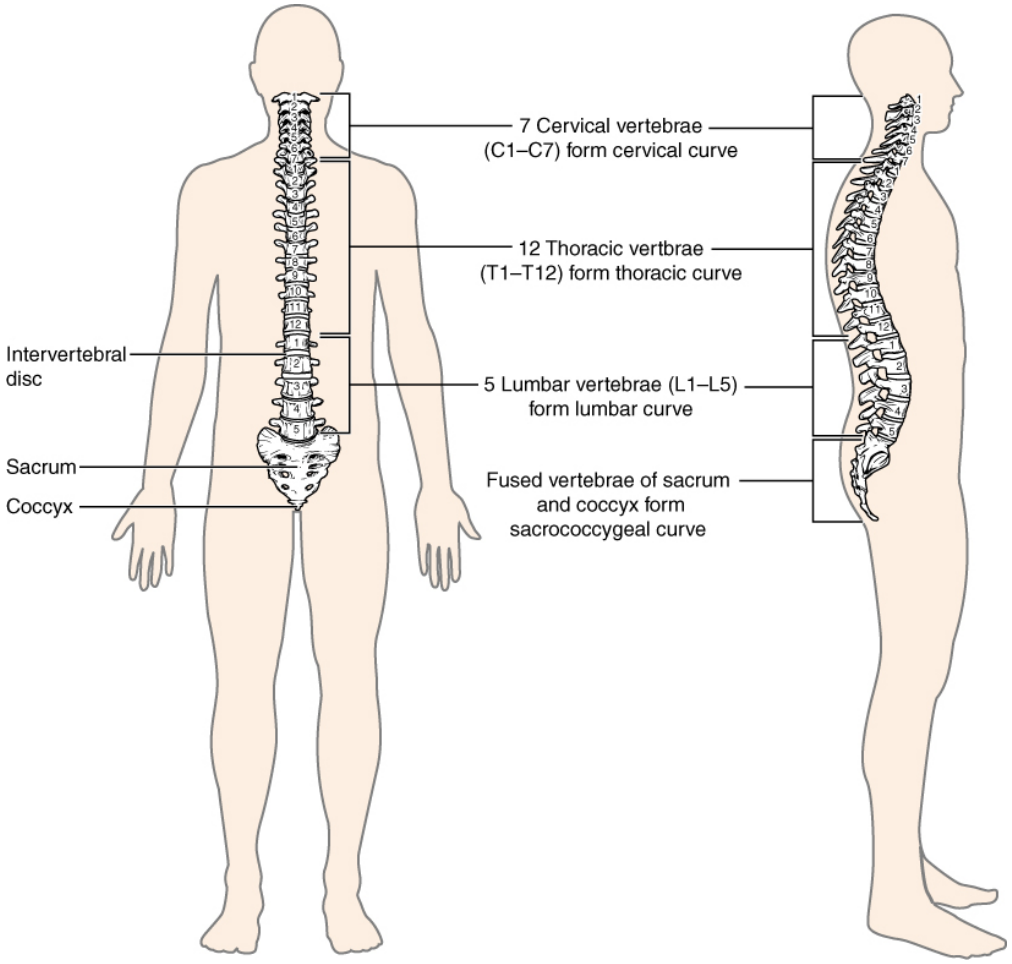


Figure 2.1: Anterior (from the front) and lateral (from the side) view of the spinal column. The lateral view shows the characteristic curvature. Image from OpenStax [10].

the thoracic and sacral region an outward curvature. The inward curvature (concavity) is called *lordosis*, while the outward curvature (convexity) is called *kyphosis*. The alternating order of lordosis and kyphosis results in the double-S curvature of the vertebral column. This characteristic provides elasticity and therefore helps to absorb pressure while walking, running and jumping [17].

2.1.1 Vertebrae

Cervical, thoracic and lumbar vertebrae have the same structure in general, but show anatomical variations, depending on their location in the spine. The typical structure of a vertebra is shown in Figure 2.2: A vertebra has a vertebral body and a vertebral arch. The arch is formed from the paired pedicles and the paired laminae. The pedicles extend from the body and the laminae from the pedicles. The laminae fuse in the midline, where the spinous process originates. The

Notation	Latin Notation	Body Area	Number of Vertebrae	Abbreviation
Cervical	Vertebrae cervicales	Neck	7	C1 - C7
Thoracic	Vertebrae thoracales	Chest	12	T1 - T12
Lumbar	Vertebrae lumbales	Low Back	5	L1 - L5
Sacrum	Vertebrae sacrales / Os sacrum	Pelvis	5 (fused)	S1 - S5
Coccyx	Os coccygis	Tailbone	3-6	—

Table 2.1: The five parts of the human vertebral column.

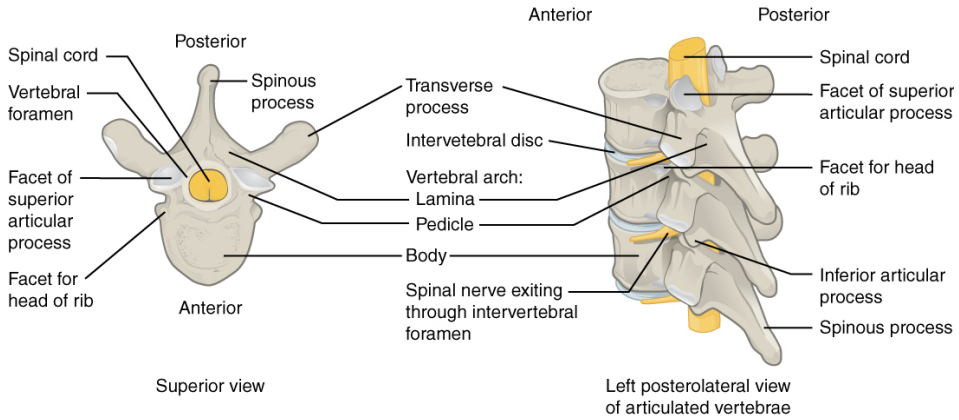


Figure 2.2: Parts of a typical vertebra. Image from OpenStax [10].

arch and the posterior of the vertebral body bound the vertebral foramen. The foramen from all vertebrae form the spinal canal, which holds and hence protects the spinal cord. Furthermore, vertebrae have three paired processes, which originate from the vertebral arch: Two transverse processes, two superior articular processes and two inferior articular processes. The superior articular processes (facing upwards) and inferior articular processes (facing downwards) join with the articular processes from adjacent vertebrae. The processes form joints and act as sites, where muscles and ligaments connect and hence enable movement [17], [10].

The anatomy of vertebrae varies between the cervical, thoracic and lumbar body region. The size of vertebrae increases from top to bottom (see Figure 2.1), since lumbar vertebrae (see Figure 2.3) have to bear more weight than cervical vertebrae. Thoracic vertebrae (see Figure 2.4) have additional facets for connection with the ribs. One facet is located on each side of the body for articulation with the head of the rib. Another pair of facets is located at the transverse processes. There the tubercle of the ribs connect via ligaments with the vertebrae. Cervical vertebrae (see Figure 2.5) have two transverse foramen which give passage for the vertebral artery, vein and nerves [17], [10], [35]. The topmost vertebrae C1 and C2 show other variations, which are not explained within this thesis. Additional information can be found in the literature [17].

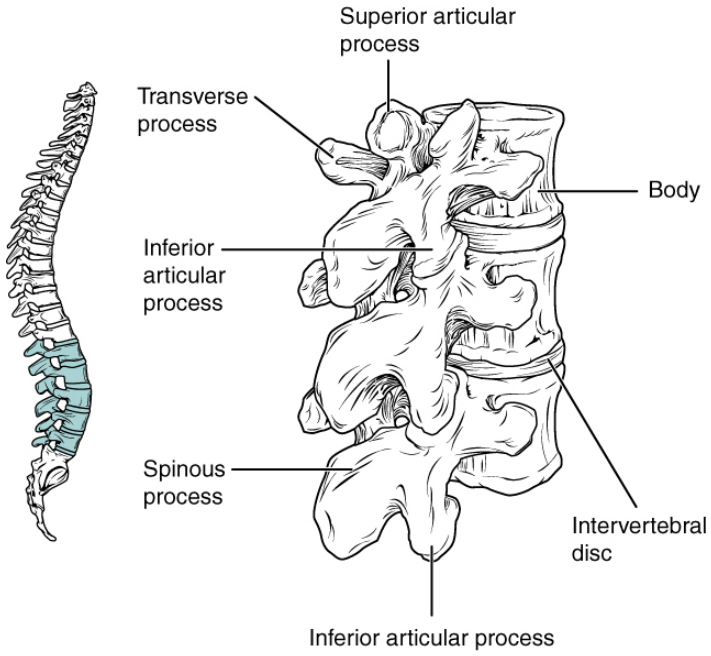


Figure 2.3: Lumbar vertebrae have a large, thick body and are characterized by a short spinous process. Image from OpenStax [10].

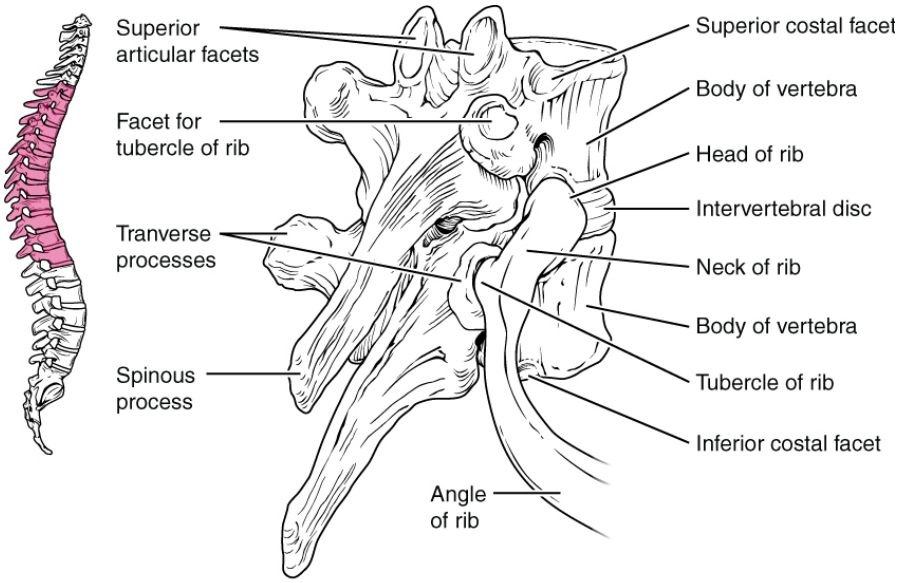


Figure 2.4: Thoracic vertebrae have additional facets for connection with the ribs and a long spinous process. Image from OpenStax [10].

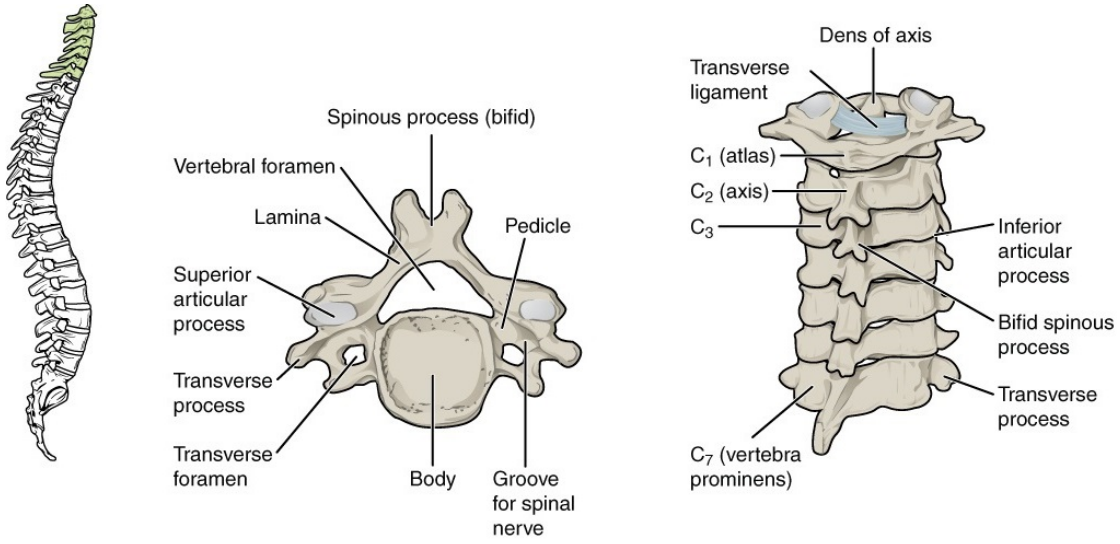


Figure 2.5: Cervical vertebrae are characterized by a small body and additional foramen, that give passage e.g. for nerves. Image from OpenStax [10].

2.1.2 Intervertebral Disks

Intervertebral disks are soft cushions that lie between two adjacent vertebrae in the spine. They make up 25% of the length of the vertebral column and are responsible for the natural double-S curvature, due to their uneven thickness in sagittal direction [17], [19]. An intervertebral disk consists of two parts (see Figure 2.6):

anulus fibrosus The fibrous ring on the outer side of the disk is built of 10-15 concentric layers of collagen fibres. It protects the inner, softer part of the disk and prevents it from getting pressed out.

nucleus pulposus The enclosed central portion of the disk is the actual cushion. It gets flatter under pressure and distributes the pressure evenly to the whole disk. The fibrous ring absorbs the pressure, rebounds, and brings the nucleus back in position. The nucleus releases water under pressure, which results in a decline of the height of the vertebral column up to 3 cm during the day. This process is reversible when lying down. The water content of the nucleus is about 85-90% and decreases with advancing age [19]. This structural change of the intervertebral disks can be observed with Magnetic Resonance Imaging (see Section 2.3.3) .

2.1.3 Variations and Pathologies of the Spine

Besides natural anatomical variations, physiological and pathological variations of vertebrae, disks and the whole spinal column can occur. Some are mentioned in the following.

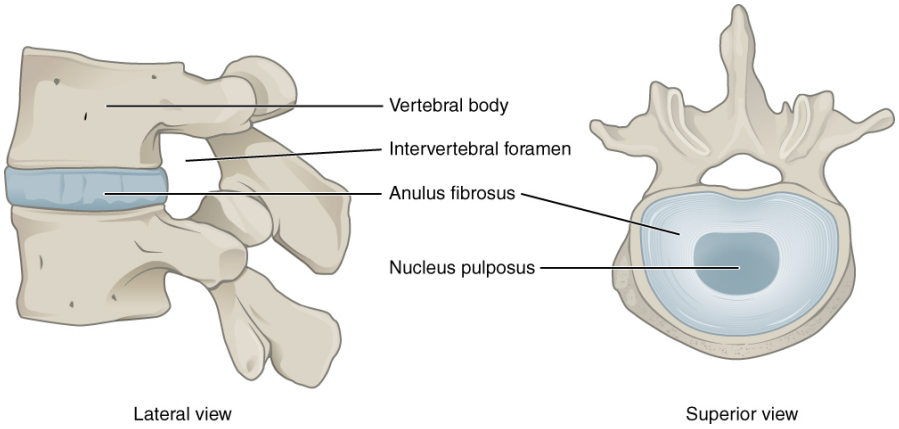


Figure 2.6: Lateral and superior view of an intervertebral disk, showing the embedding between two adjacent vertebra (left) and the two parts of a disk (right). Image from OpenStax [10].

Especially in the lumbar spine, the number of vertebrae varies. Instead of fusing with the sacrum, the first sacral vertebra may also develop a vertebral body and appear as sixth lumbar vertebra, whereas the sacrum only consists of four fused vertebra. This phenomenon is called *lumbarization* [17].

The fusion of two or more vertebra is called *block vertebra*. Several factors may cause block vertebrae, like inherited or age-related reasons or development due to trauma. Block vertebrae mainly occur in the cervical spine [17], [50].

Furthermore, vertebrae can suffer from *osteoporosis*. This common age-related disease reduces the density and strength of bones, which makes them prone to fractures. Osteoporosis reduces the height of vertebrae and hence changes the curvature of the spine. This may lead for example to an excessive kyphosis of the thoracic vertebral column [50], [35].

Another pathology regarding the curvature is *scoliosis*. A scoliotic spine shows an abnormal lateral curvature, which is accompanied by twisting of the spine [50].

Narrowing of the spinal canal (i.e. the vertebral foramen), is called *stenosis*. It leads to compression of the spinal cord and spinal nerves [50].

Intervertebral disks can suffer from *disk herniation*, which happens when the nucleus moves into the outer fibrous rings. There are several stages, whereby in the worst case, parts of the nucleus are completely pressed out of the intervertebral disk and the connection to the remaining nucleus is lost [50].

2.2 Overview of Imaging Modalities for the Spine

In radiological diagnostics of the vertebral column, usually two radiographs are acquired first (one in anterior-posterior direction, the second in lateral direction). Radiographic images are created by casting X-rays from a beam onto the desired part of the body. Every tissue absorbs and scatters the rays in a different way, depending on the thickness, density and the atomic number of the underlying tissue. For dense tissues, the attenuation of X-rays is high and the

appearance in the obtained image is bright (e.g. bones). Brightness decreases for soft tissues, water, fat and air, with air appearing darkest in radiographs [50].

For further investigations and to support diagnosis in case of special diseases (e.g. herniated disks or tumors), CT and MRI can be applied. In contrast to X-ray, which captures only a 2D image, these two methods capture multiple slices of the examined body region. Therefore a 3D reconstruction of the region is possible, which provides a better insight [50].

CT is also based on X-rays. The patient lies on the back on a desk and is put into a tube, where a 360° rotating beam casts the X-rays onto the desired body parts. A series of transverse slices is obtained, which results in a 3D dataset. The reconstructed images show the attenuation coefficients of the tissues. In order to compare different tissues in CT scans and overcome the indifferences caused by the radiation energy, Hounsfield Units (HU) are introduced [50]. The Hounsfield scale normalizes the attenuation coefficients of tissues, based on the attenuation of water and air. Air maps to -1000 HU and water to 0 HU. The range of the Hounsfield scale usually lies between -1000 and 3000 HU. While air, fat, water and bones are significantly different in CT scans, the contrast for soft tissues is very poor as they lie within a small Hounsfield range [58].

The main disadvantage of X-ray and CT imaging is the exposure of the body to radiation. The amount of radiation is indicated by the effective dose. It describes the biological effect of ionizing radiation and takes the radiation sensitivity of organs into account. The unit of a radiation dose, e.g. the effective dose, is *Sievert (Sv)*. The effective dose for an examination of the spine ranges from 0.09 - 1.8 mSv for X-rays (in two directions) and from 3 - 10 mSv for CT scans (depending on the examined spinal region). In comparison, humans are normally exposed to a radiation of 2.1 mSv per year in their natural environment [50].

2.3 Magnetic Resonance Imaging

In contrast to X-ray and CT, MRI is not based on radiation. It rather uses a property of the nucleus (i.e. of protons and neutrons) of an atom, which is called *spin*. This spin is an angular momentum, which can be observed and is the basis of MRI (see detailed in Section 2.3.1).

MRI has several advantages over CT. It provides the best contrast for soft tissues and is the method preferred for the locomotor system (except bones) and bone marrow. It is very important for examining the central nervous system (i.e. the brain and spinal cord) and intervertebral disks [50]. Furthermore, it is possible to acquire images in arbitrary directions, e.g. coronal or sagittal images (see Section 2.3.1). Probably the main advantage of MRI is the absence of ionizing radiation. However, problems can occur because of the strong magnetic field, e.g. with electronic implants like pacemakers or magnetizable metallic objects. Another disadvantage are artifacts, that occur during image acquisition due to the relatively long examination time and the complex imaging technique [58], [50]. Possible reasons for artifacts are:

- movement of the patient
- flow artifacts due to blood flow

- extinction or distortion of the signal caused by local magnetic field inhomogeneities
- signal interference
- chemical-shift-artifacts, which appear at boundaries between fat and water

Changes in the acquisition technique and averaging over multiple acquisitions can reduce movement artifacts. Also clarification with the patient is important in order to reduce movement [8].

2.3.1 Image Acquisition

In MRI the spin of nuclei is the basis for imaging, as mentioned in the previous section. Usually, the nucleus of hydrogen (which consists only of one proton) is used, because hydrogen is very abundant in the human body [58], [65]. When acquiring an MR scan, the patient is also put into a tube as in CT imaging. In case of MRI, the scanner produces a strong, longitudinal magnetic field. The strength of the field is measured in Tesla (T). Imaging devices nowadays operate with a strength of 1 T to 3 T for human full body scans [58]. The obtained magnetic field aligns the spin of the nuclei of all atoms of the body to the external field. This is the resting state, where the examination starts from. In order to produce measurable resonance, a perpendicular high frequency (HF) signal is switched on, which leads to the excitation of the protons. The spins of excited atoms are projected into the xy-plane (in a certain flip angle), which is orthogonal to the outer magnetic field B_0. When the HF signal is switched off, the protons (i.e. the spins) start returning to their resting state (see Figure 2.7). This process is called *relaxation*. The relaxation consists of a simultaneous longitudinal and transverse relaxation. They are described by the tissue-dependent time constants T1 and T2 [19], [58]:

Longitudinal relaxation and T1 time The longitudinal relaxation is finished, when all spins are aligned to the magnetic field of the MRI scanner again, i.e. when the magnetization M_z is restored (see Figure 2.7). The time constant for this exponential process is given by the T1 time and lies between 300 - 2000 ms, depending on the tissue.

Transverse relaxation and T2 time The spins of the protons induce individual magnetic fields. Right when the HF signal is turned off, all spins are in phase and give the maximum signal in transverse direction M_{xy}. With the proceeding of time, the protons dephase and the signal decreases (see Figure 2.7). The T2 time describes the exponential decay of the transverse magnetization M_{xy} towards its initial state. It lies between 30 - 150 ms and is as the T1 a tissue-dependent constant.

In order to generate images out of the produced magnetic resonance signal, additional gradient fields (x-, y- and z gradients) are needed. Before the HF signal is switched on, a gradient field in the direction of the outer magnetic field is applied. The spin frequency of the protons depends on the location in the gradient field. When applying the HF signal, only the protons are excited whose spin frequencies correspond to the frequency of the HF signal. This process is called *slice selection*. By overlying additional gradient fields in xy-direction, the location of the protons within the slice is determined. These gradient fields are not necessarily perpendicular

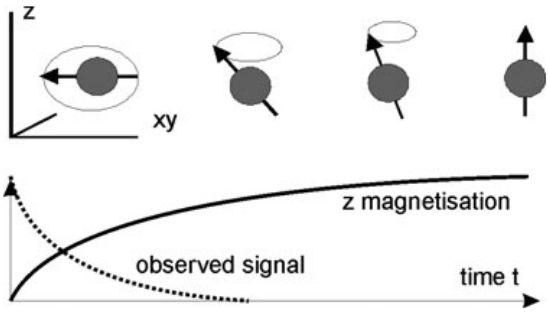


Figure 2.7: The HF signal causes the spin of the nuclei to flip in the xy-direction (top row). When the signal is switched off, the observed signal decreases faster than the longitudinal magnetization is restored (bottom row). The corresponding restoration of the spins z-magnetization is shown in the top row. Image from Toennis [58].

to the longitudinal magnetic field of the scanner. They can be generated in arbitrary directions, which enables imaging in any preferred direction.

In order to suppress noise in the obtained scan, multiple excitations are performed and the obtained signal is averaged. Multiple consecutive HF impulses are called *sequence* or *pulse sequence* (see below). The time between two excitations, i.e. the time between two consecutive HF pulses applied to the same slice, is called *repetition time (TR)*. The *echo time (TE)* describes the time between excitation and echo of the signal. In contrast to the T1 and T2 time, TR and TE are tissue independent measurement parameters. They can be reset by the operator for every new acquisition. These parameters influence the image contrast, which will be explained in Section 2.3.2.

As mentioned, predefined settings of the measurement parameters are denoted as sequences. They influence the image contrast, acquisition time and hence image quality. A commonly used sequence is the *spin-echo sequence (SE)*, which provides the best tissue contrast and is not prone to inhomogeneities of the magnetic field. Disadvantageous is the relatively long examination time. The *fast-spin echo (FSE)* or *turbo-spin echo (TSE)* sequence and the *gradient-echo (GE, GR)* sequence provide faster acqusition. However, the tissue contrast is lower within these sequences and more artifacts are present [50]. Details on these sequences and others can be found in the literature [65], [43].

2.3.2 Image Contrast

The image contrast in MR scans is subject to tissue-dependent parameters (T1, T2 and proton density), but also to sequence parameters (TE, TR) and the sequence type (SE, GE, TSE, etc.). Tissue-dependent parameters define the weighting of the image: In T1w and T2w images the image contrast originates mainly from the T1 and T2 relaxation times, respectively. Whereby the echo time controls the T1 contrast, the repetition time is responsible for the T2 contrast. In proton density (PD) weighted images, the PD of the tissues defines the appearance [50]. Unlike within CT, a normalized scale does not exist [58]:

T1-weighting (T1w) In T1w scans, the intensity of the tissues is determined by their T1 time. The data is acquired with a short TR and TE [50]. A short TR interval (400 - 800 ms in practice) is necessary so the T1 relaxation is not yet over within the TR interval. In order to suppress T2 influences, a short TE is required, usually below 25 ms. Tissues with a long T1 time appear dark, whereas tissues with short T1 time have a brighter appearance (see T1w scan in Figure 2.8). Water for instance is very dark, because it has the longest T1 time. Fat has the shortest T1 time and is therefore very bright [19].

T2-weighting (T2w) In T2w scans, the image contrast originates from the T2 time of the tissues. Contrary to T1w images, a long TR and TE are characteristic for T2w scans [50]. To minimize the T1 influences in T2w images, the T1 relaxation has to finish completely. Tissues with long T2 time appear brighter than tissues with short T2 relaxation time (see T2w scan in Figure 2.8). Water has the longest T2 time and hence is very bright. For good contrast in T2w images, a TE about 70 - 150 ms and a TR above 2500 ms is used in practice [19].

PD-weighting (PDw) In PDw scans, tissues with high PD appear bright (e.g. water). Tissues with low PD appear dark in the image (e.g. bones and air). PDw scans are acquired with a long TR and short TE. The T1 and T2 times of the tissues have almost no influence on the image contrast [50].

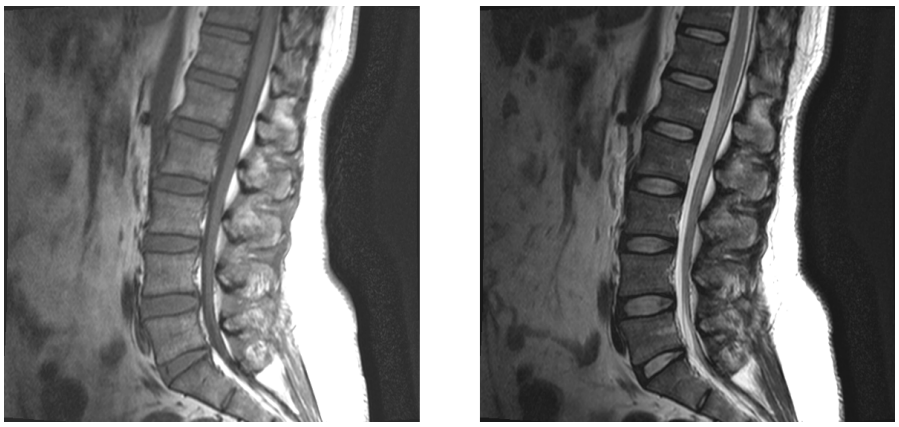


Figure 2.8: T1w (left) and T2w (right) image of the lumbar vertebral column. Images from Dataset VOLUMEMERGIX by OsiriX [46].

2.3.3 MRI on the Vertebral Column

In the general MRI acquisition pipeline, a low resolution *scout* scan (also referred to as *survey* dataset) is acquired first. This sparse scan usually consists only of a few slices in sagittal and coronal direction [16]. It is necessary for the operator in order to plan the follow-up diagnostic scan. He determines the scan geometries, such as field-of-view and angulation [47]. In this way, he decides on preferred image planes (regarding the orientation) for diagnosis. For regions

of interest, e.g. for vertebrae showing fractures or herniated disks, additional image data can be acquired. Since the planning of scans depends greatly on the experience and skill of the operator, there are methods towards automatic planning of follow-up scans on the spinal column [16], [47] (see Sections 3.2 and 3.4).

Besides the scan geometries, one has to decide also on the measurement parameters. As reviewed in the previous sections, they are used to control the image contrast. The different contrasts T1w, T2w and PDw provide insight into different medical problem statements. When it comes to imaging the vertebral column, T1w scans are suitable for examining bone marrow and T2w images for the spinal cord, spinal canal and intervertebral disks. Pathologies like tumors and inflammations exhibit a high signal in T2w scans, due to the increased amount of liquid within this tissues [50].

Not only pathological variations, but also physiological, age-related changes of the structure of the spine are visible in MR images. T1w scans show a steady increase of the signal intensity of bone marrow with advancing age, whereas the appearance of disks remains constant. The intensity of intervertebral disks decreases in T2w scans, due to the age-related dehydration of the disks [8] (see Figure 2.9). The evolution of the spinal column from childhood to advanced age is depicted in Figure 2.10 on T1w and T2w MR scans.

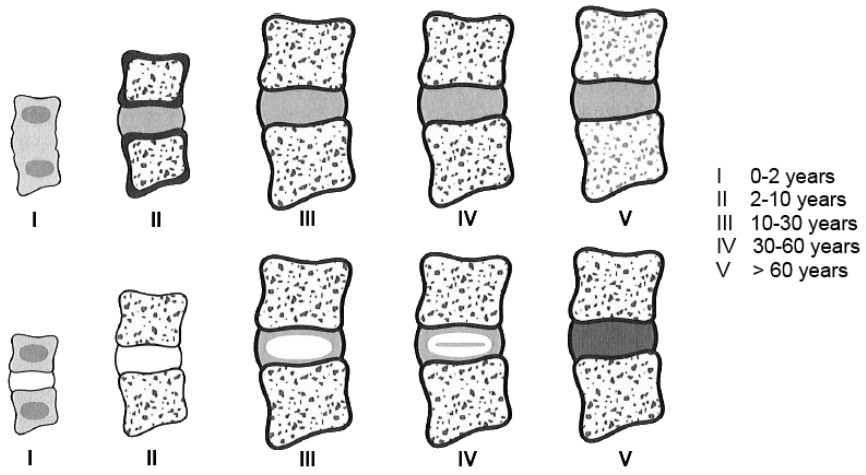
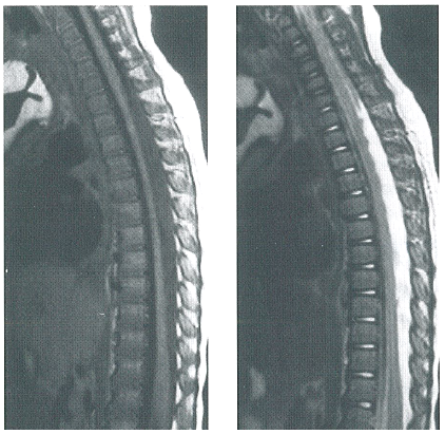
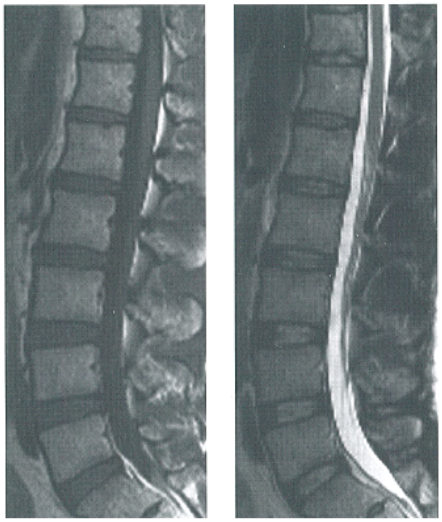


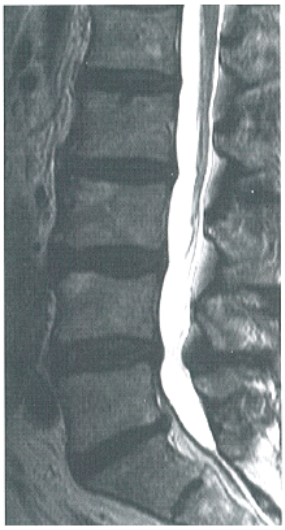
Figure 2.9: Schematic illustration of changing signal intensities of intervertebral disks and bone marrow with advancing age in T1w (top row) and T2w (bottom row) MR scans. Image from Breitenseher [8].



(a) T1w (left) and T2w scan (right) of a 1 year old child. The T1w image shows, that the intensity of vertebrae is slightly higher than the signal of disks. In the thoracic spine, the distinction between disks and vertebrae is difficult. The T2w scan provides a high signal intensity of the disks nucleus. Due to the dark signal of the fibrous ring, disks discriminate very good from vertebrae.



(b) Physiological T1w (left) and T2w scan (right) of a 40 year old man. Vertebrae show a higher signal than disks in T1w images. The nucleus and vertebrae show a medium signal, whereby the fibrous ring appears darker in T2w scans.



(c) T2w scan of a 62 year old man. The image shows a reduced signal intensity of the disks due to age-related dehydration.

Figure 2.10: Changing signal intensities of intervertebral disks and vertebrae with advancing age. Image from Forsting et al. [19].

CHAPTER 3

Related Work

In general, spine labeling algorithms can be reviewed and categorized in terms of different criteria. Depending on the built-up of the method, different aspects should be kept in mind when reviewing such algorithms. Possible categories are listed in the following:

Data As mentioned in Section 2.3, MR data shows high variability depending on the acquisition protocol used. Several approaches focus on low resolution scout images, especially the ones which aim for automated scan planning. Furthermore, one can distinguish between the image contrast (e.g. T1w or T2w), as well as between 2D and 3D acquisition.

Preprocessing For the present intensity inhomogeneity fields, different correction algorithms exist. The correction is performed either during acquisition or afterwards by means of, e.g. segmentation or filtering [62]. There are also methods which go one step further and cope with the problem of varying intensities across different MRI acquisition sequences.

2D vs. 3D methods Many approaches apply 3D segmentation and feature detection algorithms on the data. Other methods rather work on a selected sagittal slice and aim to detect the region or landmark of interest in a 2D manner.

Automation In the literature, semi- and fully automatic segmentation, localization and labeling techniques are present. Semi-automatic methods require user interaction to a certain extent. That means for example, the selection of the "best slice" (i.e. were disks or vertebrae are visible best) or one or more initial clicks. Fully automatic algorithms manage to perform without any input from a user.

Localization vs. segmentation Many approaches roughly localize the region of interest, whereby others provide a full segmentation of the intervertebral disks or vertebrae before labeling.

Model-driven vs. data-driven methods Another possible differentiation is the usage of anatomical knowledge within the algorithm. Model-driven approaches learn from a training set and incorporate anatomical knowledge in the learned model, which is then applied

on unseen data. On the other hand, data-driven methods integrate hardly any knowledge about the underlying anatomy. They infer the required information from the target data and are usually not based on machine learning [38].

From this list of criteria two are chosen as outline for the following chapter, namely *preprocessing* and *2D vs. 3D labeling methods*. Section 3.1 gives a short overview about preprocessing approaches, which are detailed in the following chapter. Section 3.2 and Section 3.3 review 2D and 3D state-of-the-art techniques regarding detection, localization, segmentation and labeling of the spinal anatomy, respectively. The focus hereby lies on intervertebral disk localization and labeling on MR data, since it is also the main goal of this thesis. Statements regarding other criteria, such as the dimensionality of the data and automation, are also made for the algorithms mentioned. Finally, alternative approaches to intervertebral disk localization and labeling are presented in Section 3.4. These methods rather focus on full segmentation, vertebrae detection or non-machine-learning techniques. Section 3.5 concludes the related work and motivates the proposed approach.

3.1 Preprocessing Methods

Various methods exist to enhance MR data, especially when it comes to reducing *intensity inhomogeneities*. Vovk et al. [62], Belaroussi et al. [5] and Hou [27] reviewed several methods for correcting those. Due to inhomogeneities of the magnetic field and non-uniformities in the radio frequency, the same tissues (e.g. disks) do not exhibit the same intensity in an image. The appearance of the tissue rather depends on the location in the image. The intensity inhomogeneities (also referred to as *bias field*, *gain field* or *intensity non-uniformity* in the literature [27]) appear as shadow in the border region of the acquired MR scan (see Figure 3.1).

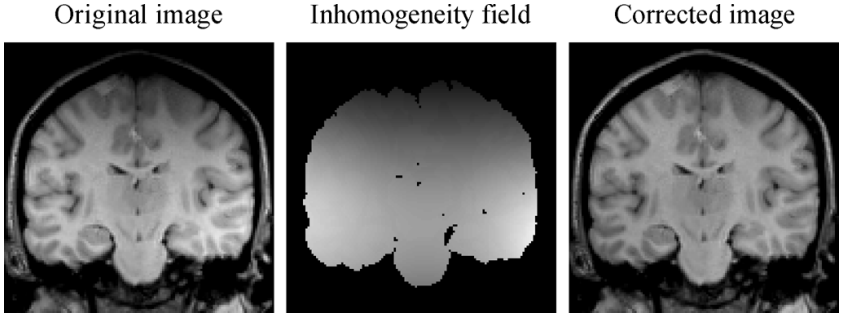


Figure 3.1: MR image of a brain with intensity inhomogeneities. Image from Vovk et al. [62].

A fast, retrospective, histogram-based correction method is presented by Nyúl et al. [45]. They try to overcome the intensity inhomogeneities by learning parameters for a standardized histogram in a way that the same tissues in the same body region acquired with the same MRI protocol have the same intensity. Parameters are learned from training data to obtain a standardized transformation. They adjust new images so that the histogram is mapped onto the standardized histogram. The authors state, that the histograms are either unimodal or bimodal and the

same body regions have the same type of histogram. They tested their method on T1w and T2w images (FSE and GE sequences). The acquired studies cover brain and foot data. This approach is similar to the proposed entropy-based method from Zambal et al. [67] (see Section 4.1.2). The difference is, that Nyúl et al. [45] learn a standardized histogram for each protocol-body-region group and the method from Zambal et al. [67] works across different MRI protocols and even modalities with only one trained model.

For correcting the bias field, a surface fitting approach proposed by Juntu et al. [31] is integrated in this work. The authors extract a background image from the scan and fit a surface to it. The obtained function describes the inhomogeneity field, which is then removed from the original scan. On top of the bias field removal, the method from Zambal et al. [67] is used for normalization of the different MR scans (T1w, T2w). The outcome of this step is a standardized intensity scale.

3.2 Two-dimensional Spine Labeling Approaches

Peng et al. [48] are among the first who addressed automated whole spine analysis on MR data. Within their survey, they propose a method for fully automatic disk localization and vertebrae segmentation on sagittal images. First they localize the best slice, i.e. the one with the most visible intervertebral disks. The best slice is obtained as follows: By convolving all slices separately with a predefined disk model, they find disk clue points, which are used to fit a polynomial. The extracted intensity profile along the fitted polynomial shows possible disk centers (i.e. local peaks in the profile) and vertebrae locations (local minima). The variation of the height of vertebrae is calculated and the slice with minimum variation is selected as best slice. Starting from one disk clue position, the disk center positions are refined by incorporating knowledge about approximate shape and size of disks and vertebrae. A large distance between two intervertebral disks is interpreted as a missing disk, which is then searched in adjacent slices. This is the case when dealing with scoliotic spine images, for instance. Finally the boundaries of the vertebral bodies are extracted. Edge points are detected with the Canny edge operator and refined. They tested their method only on five MR scans, where they detect 100% of disk locations and between 87.5% and 100% of vertebrae corners of the whole spine. The scans have seven sagittal slices each and were acquired on 1.5 T scanners. The authors did not state further details about the data.

Pekar et al. [47] developed a labeling approach for automated MRI scan planning. As mentioned in Section 1.1, an operator is responsible for a correct and consistent parameter setting of the acquisition. The goal of automated planning tools is to derive the acquisition parameters automatically from the low resolution scout scan. Therefore a robust labeling of the spinal anatomy is needed. Since the labeling is performed on scout scans, this approach is one step beyond the afore-mentioned work from Peng et al. [48]. They work on already existing diagnostic scans. Pekar et al. [47] extract disk candidates from 2D sagittal slices by detecting approximately horizontal lines. The centers of mass of the detected structures represent potential disk centers. Progressive search is performed from the determined starting point (topmost candidate position in lumbar scans and bottommost candidate in cervical scans) by looking in defined dis-

tances from the last found disk for the next disk center. In case of missing points, artificial disk centers are inserted and the search continues from there. For the automated scan planning, the detected landmarks are used to find correlations with manually defined plan sets. The goal is to find a suitable mid-sagittal plane and planned vertebrae and discs, where transverse slices will be obtained. They evaluated their method on lumbar and cervical scout scans. The 3D T1w FFE (Fast-Field Echo) scans were all acquired on a 1.5 T Philips scanner with the same field-of-view and voxel size. The labeling is correct in 29/30 cases for the cervical scans and 25/30 cases for the lumbar area. The overall processing time for anatomy detection and scan geometry planning is 6 s.

Corso et al. [15] propose a two-level probabilistic model for intervertebral disk localization and labeling of lumbar MR scans. They aim to overcome the problematic intensity variations in disks and hence increase robustness with their method. The first level models low level pixel information, such as the local disk appearance (i.e. the intensity). The spatial information is also incorporated by assuming an elliptic shape for the intervertebral disks. The second level adds high-level information to the model, i.e. geometric and contextual knowledge of all lumbar disks. The expected location of disk centers and distances between neighboring disks are taken into account. The local disk intensity models, disk distance distributions, and so forth are learned from labeled training data. The optimum solution for the labeling problem is determined by means of the generalized expectation maximization algorithm. They tested their method on 20 T2w scans from healthy patients, acquired on a 3 T Philips Scanner. From these scans, they extracted the four middle slices, which resulted in 80 2D images in total. They evaluated their method with leave-two-out cross validation. Every cross validation round two datasets (8 slices) are left for testing and the remaining 72 slices are used for training. The labeling accuracy reached is 96.2%.

Alomari et al. [2] continued the work from Corso et al. [15] and added also abnormal cases to their model. They extended their dataset to 105 cases, now including subjects with various disk abnormalities and patients of different age and height. They report, that it is not necessary to train a separate model for normal and abnormal cases. When using different training and testing sets, they achieve an accuracy of about 90 % for training and testing set (87% when using only abnormal cases). This proves the robustness of their method and shows that they did not overfit their model. They work an T2w SPIR (Spectral Presaturation Inversion Recovery) images, where intervertebral disks discriminate very good from other anatomical structures.

3.3 Three-dimensional Spine Labeling Approaches

Schmidt et al. [52] propose a parts-based probabilistic graphical model for locating and labeling the spine. It models the local appearance of the disks and also the relative location of disk pairs. Candidate points of the local parts (i.e. the disks) are detected by a set of randomized tree classifiers, which are learned from the training data. The most probable locations of these candidates are then determined to find the global-optimum configuration of parts. This is performed with the A^* algorithm. The algorithm is trained and tested on 3D T1w FFE MR full

spine scout scans, acquired on one scanner. The disks appear usually bright whereas vertebrae give no signal. The dataset covers inhomogeneities due to the magnetic field and also fractures. The method proposed can cope with missing parts, but fails if the geometry of detected parts is not covered by the training data (e.g. severe fractures). The authors report an average ground truth distance of 7.8 mm for novel data.

Kelm et al. [32] propose a learning-based approach, which they trained on MR and CT data alike. First they roughly localize the cervical, thoracic and lumbar part of the vertebral column with the Marginal Space Learning (MSL) approach, proposed by Zheng et al. [68]. This algorithm is designed to find a specific object within a volume, e.g. a landmark or organ. MSL does not search the whole nine dimensional parameter space (three position, three orientation and three scale parameters) at once. The algorithm stepwise determines position, orientation and scale parameters after another by following only the most promising candidate detections from each step (see Figure 3.2). For determining intervertebral disk candidates, the authors extended MSL and formulate an iterated approach, because they aim to find multiple objects of the same type (i.e. the disks) within the localized spinal parts. The basic MSL method and the iterative extension are shown in Figure 3.2. Haar-like features and boosting trees are applied as position detectors, which are also used in this thesis. Selecting and labeling of the disk candidates is performed by matching a global probabilistic spine model through solving a maximum a-posteriori optimization problem. As last (optional) step, detailed segmentation is performed with the estimated position, orientation and scale of the disk landmarks. A patient-adaptive segmentation approach based on graph cuts [7] is applied, which is able to adapt to the current underlying data. Fore- and background models are obtained on-the-fly during processing and not beforehand from training data, which makes the segmentation very robust. Segmentation is provided for intervertebral disks and vertebrae on MR and CT data, respectively.

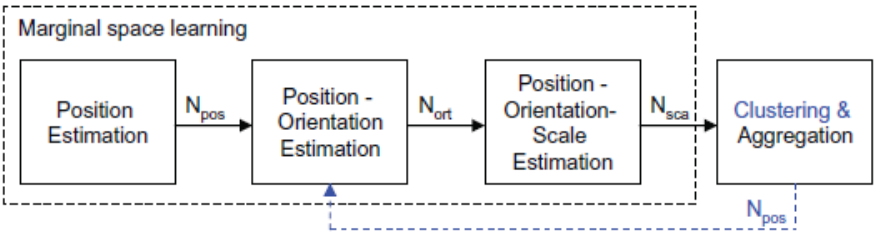


Figure 3.2: The original MSL algorithm (in black) and the iterative extension (in blue). Image from Kelm et al. [32].

The evaluation was carried out on full spine MR datasets, acquired on two different scanners from healthy patients. They used a T1w VIBE (Volume Interpolated Breath Hold) sequence, which provides real 3D GE data with Fourier interpolation. Advantages over 2D GE sequences are a higher signal-to-noise ratio, thinner slices and no gaps between slices, due to the 3D acquisition [43]. Artifacts and intensity inhomogeneities were present in the data, but no correction algorithm was applied. The overall processing time is 11.5 s on average and the detection of disks yields a sensitivity of 98.6% and only 0.073 false positive detections per volume. The accuracy is also very high, with an average position error of only 2.42 mm. However, their

approach cannot deal with scans with varying scan ranges. The algorithm needs external information, provided by a user or an automatic scan range detector [32].

3.4 Alternative Approaches

Apart from the reviewed spine labeling algorithms, this section reviews methods based on segmentation, non-machine-learning approaches, or algorithms that focus on vertebrae detection first.

Dong et al. [16] also build a graphical model, similar to Schmidt et al. [52], Corso et al. [15], and Alomari et al. [2], but aim for detecting vertebrae first. Furthermore, the detection of the vertebrae and disks is replaced by a non-machine-learning approach. Hence no training is required, but constraints and assumptions are added to the graphical model. Also more input from the user is required as initialization. One has to select a sagittal slice, where all vertebrae are visible and pick the center of the first and last visible vertebra, which makes the method rather hard to use. The detected vertebral bodies are the initialization for the intervertebral disk detection. By means of particle filtering, disks are detected between vertebral bodies on sagittal slices. With this detection result, the nearest coronal slice showing the disk can be determined and the 3D geometrical information (center, orientation, and so forth) is derived. The authors also aim for automated scan planning and therefore deal with scout scans as input data. They focus on data from the lumbar and thoracic body region. With an execution time of 1 s per disk and a mean error of 5 mm (distance-to-disk-center) and 5° (disk plane orientation), the results are acceptable for scan planning.

Neubert et al. [42] propose a fully automatic segmentation technique using 3D statistical shape models. They only segment the lower spine, but do not perform labeling. Meshes are generated from the manually segmented disks and vertebrae with the marching cubes algorithm. After finding corresponding points between the triangulated meshes, they get procrustes aligned and a mean shape is computed, i.e. the mean position of every corresponding point. By means of eigenanalysis, each shape - and also new shapes - can be described by the mean shape and a weighted sum of eigenvectors (i.e. the modes of variation). The statistical shape model describes then the mean shape and its possible variations in each corresponding point. Every shape in the database is described with this model and 1D intensity profiles for every surface point on a shape are extracted. When segmenting a new MR scan, first the 3D spine curve is located. They use an enhanced, automated approach of the curved planar reformation method proposed by Vrtovec et al. [63]. The approximate disk and vertebrae centers are determined by analysing the intensity profile along the extracted curve. These points are used as initialization for the statistical shape models. The extracted intensity profiles from the mesh surfaces guide the shape deformation during the segmentation process. Their method is evaluated on three different datasets, having six scans each: T1w lumbar scans, 3D T2w lumbar scans and 3D T2w thoracic scans. The 3D datasets are acquired with the SPACE (Sampling Perfection with Application optimized Contrasts using different flip angle Evolution) sequence, which exhibits a very high resolution - similar to CT. The sagittal in-plane resolution is 0.34 mm and the slice thickness is 1 mm - 1.2 mm. The similarity between the manually segmented structures and the automated

segmentation is described by the dice similarity measure. The authors state a mean similarity of 0.83 for the T1w scans and 0.85 - 0.87 for the 3D datasets. However, they have to derive six 3D statistical shape models, in order to be able to deal with the variability within the data: one for intervertebral disks and one for vertebrae for each of the three datasets.

Neubert et al. [41] evaluated their approach also on another dataset consisting only of T2w 3D SPACE scans of the lumbar and thoracic vertebral column. They derive twelve statistical shape models: two for lumbar and thoracic disks and ten for the vertebrae $T8$ to $L5$, due to the high shape variability of vertebrae. They report a mean dice similarity measure of 0.89 for intervertebral disks and 0.91 for vertebrae. However, with an average segmentation time of 26 s for one intervertebral disk and 21 s for one vertebrae, the proposed algorithm is rather slow. Since their algorithm is currently single-threaded, there is some room for improvement of the processing time, e.g. by parallelization of profile matching.

Lootus et al. [36] introduce a pipeline for automatic detection of labeled bounding boxes around vertebral bodies. Their algorithm works in 2D and processes every sagittal slice from a stack of slices individually. The authors train two feature detectors (vertebrae bodies and sacrum) for the detection of bounding boxes around the anatomy (based on the Deformable Parts Model framework [18]). For training, 2D patches showing single vertebrae are extracted and Histogram of Oriented Gradients features are calculated from the patches. On an unseen dataset, the feature detectors provide bounding boxes (around vertebrae and the sacrum) on every sagittal slice. With non-maxima suppression, false positive detections are eliminated and the remaining bounding boxes are fed into a graphical model which delivers the final labeled bounding boxes. The method is trained and tested on lumbar T2w scans. The authors yield a correct identification rate of 84.1% for all vertebrae, with a mean vertebra body center error of 3.3 mm $\pm$ 3.2 mm. Overall processing time is less than a minute. The method can cope with pathologies, such as scoliosis and deformed vertebrae or disks. It is robust against imaging artifacts and also intensity variations. The authors claim that it is applicable to CT data without retraining. This makes it interesting for multi modal feature detection. However, the authors do not state the performance on T1w scans, which exhibit usually a lower edge contrast as T2w scans.

3.5 Conclusion

The reported state-of-the art semi- and fully automatic spine labeling approaches on MR datasets focus on a specific type of dataset instead of dealing with scans from different protocols.

Corso et al. [15] and Alomari et al. [2] trained a two-level probabilistic model on lumbar T2w scans. They were acquired on a single scanner.

Schmidt et al. [52] and Pekar et al. [47] proposed spine labeling on T1w scout scans, which were also acquired on one scanner. The latter report that all datasets exhibit the same voxel size.

Kelm et al. [32] propose an iterated MSL algorithm on real 3D T1w scans. These scans exhibit a high resolution and a high signal-to-noise ratio.

Neubert et al. [42], [41] present fully automatic segmentation of the spinal anatomy with statistical shape models. They report results on T1w and T2w scans. However, they need to train two separate models in order to process both kinds of data.

Lootus et al. [36] report that their algorithm was trained on T2w MR scans and is applicable without retraining to CT data. However, they do not state the performance on T1w MR scans which exhibits usually a lower edge contrast than T2w scans.

Nyúl et al. [45] presented an approach, which is somehow similar to the integrated method from Zambal et al. [67]. Their method tries to overcome intensity variations within one MR protocol by standardizing the intensity scale with a histogram-based approach. They tested their method on T1w and T2w scans, whereby they have to learn two standardized histogram, one for each protocol.

The goal of this thesis is to train only *one model* which captures the variability within T1w and T2w scans even across different MR scanners. The datasets are normalized to a reduced intensity scale with the learned model where disk detection is performed. The advantage is, that both types of scans can be processed without retraining or knowing, which type of dataset is currently processed. To the best of our knowledge, this approach is new within the field of MR spine labeling.

CHAPTER 4

Methods

This chapter aims at giving a detailed overview to existing methods, which will be integrated in our semi-automatic spine labeling pipeline proposed in Chapter 5. The presented literature is grouped into two main parts: *data preprocessing and normalization* and *feature extraction*. Section 4.1 introduces methods for preprocessing and normalizing MR scans. Within this scope, a noise correction method and *Entropy-optimized Texture Models (ETMs)* are reviewed. Section 4.2 gives an overview of low- and high-level feature extraction techniques. Haar-like features, image gradients, *Probabilistic Boosting Trees (PBTs)* and an extension of PBTs are detailed.

4.1 Data Preprocessing and Normalization

Many techniques for preprocessing and enhancing of CT data can be applied to MR data as well. However, the differences within MR scans have to be kept in mind when applying contrast enhancement, noise removal or similar techniques [58]. Especially intensity inhomogeneities (see Section 3.1) complicate preprocessing methods and image processing tasks in general [27]. The presence of such intensity inhomogeneities motivates the application of correction algorithms. In general, several methods exist for reducing intensity inhomogeneities. They are split into *prospective* and *retrospective* methods:

Prospective correction techniques are applied during the acquisition process. This group of methods tries to minimize the inhomogeneities by acquiring additional images with different coils or by merging imaging information from several coils. Special sequences may also be derived. These methods increase the acquisition time (due to additional measurements) and require extra hardware, which is unfavorable [62].

Retrospective algorithms on the other hand correct the MR data after the acquisition and hence overcome the afore-mentioned disadvantages. One group of these methods estimates the bias field by *fitting a surface* to image features containing information about

the inhomogeneity. The smooth surfaces are represented by splines or polynomial functions. The group of *filtering* algorithms assumes, that the inhomogeneities are present as a low-frequency noise in the image, which can be eliminated by low-pass or homomorphic filtering. *Statistical* methods assume that the bias field follows certain distributions, such as the Gaussian distribution [62], [27].

In the following, two approaches for preprocessing and data normalization are introduced:

- Juntu et al. [31] proposed an approach for the correction of a bias field in MR images, which is detailed in Section 4.1.1.
- Zambal et al. [67] presented ETMs, which reduce the intensity scale of the images to a standardized scale. Their texture model is based on entropy terms and can cope with images from different modalities. Section 4.1.2 provides a detailed overview about their work.

4.1.1 Bias Field Correction

In general, the intensity inhomogeneity field is present as a smooth, low frequency signal in the MR scan [27]. Usually it is not problematic for a reliable medical diagnosis, but it negatively influences the accuracy of image processing algorithms, e.g. segmentation tasks. The signal blurs the images and reduces high frequency content (for example edges) in the border region of the scans. Hence the intensity values change, which results in different intensities of the same tissue within an image (i.e. within a slice). This suggests to include a correction algorithm, with the goal to eliminate illumination variations before further processing. Juntu et al. [31] describe a parametric surface fitting approach for removal of intensity inhomogeneities, which does not need machine learning in advance. The main steps of the method are described in the following:

Extraction of Background Region A 2D image I_k showing illumination variations can be corrected by dividing it with an image estimating these variations (across the entire image) [31]. This image is referred to as *background image*. The 2D background image B_k is extracted by low-pass filtering, i.e. by convolving the original image I_k with a large Gaussian kernel. Formally

$$B_k = I_k * \kappa, \tag{4.1}$$

whereby κ denotes the Gaussian kernel, which has a size of about 2/3 of the size of the image I_k [31].

Surface Fitting In order to get a smoother estimate of the illumination variation, a surface is fitted to the background image. This surface is then used for correction. From the background image B_k, n data points are selected from a region where the MR signal intensity is not very low. Low signal intensity regions appear black in the images, as reviewed in Section 2.3.3. This is recommended, since this regions do not show a bias field, according to Juntu et al. [31]. For every selected point $(x_i, y_i), i = 1 \dots n$, its corresponding gray value g_i is extracted. The authors propose to apply the Levenberg Marquardt algorithm for fitting a function $h(x, y; a_1, \dots, a_m)$

to the selected data points, whereby $a_1 \dots a_m$ refer to the coefficients of h. As objective function, the method of nonlinear least squares is used. For details on the optimization with the Levenberg Marquardt algorithm, we refer to the work of Juntu et al. [31]. With the obtained function $h\,(x, y; a_1, \dots, a_m)$, a 2D bias field F_k having the same size as the underlying image I_k is generated.

Removal of Estimated Bias Field Finally, the original image I_k is divided by the estimated bias field F_k, formally:

$$I_k^* = \frac{I_k}{F_k}, \tag{4.2}$$

Advantageous to this method is, that it is only based on the current underlying data. Hence no learning phase is needed beforehand. Just the type of the surface has do be defined in advance, e.g. a quadratic or polynomial surface, a 2D spline, etc. The authors [31] propose to prefer simple surfaces (such as low order polynomials) over more complex surfaces, since they are smooth and enable simple parameter estimation. Also non-parametric surfaces using, e.g. neural networks, can be fitted to the data points.

4.1.2 Entropy-Optimized Texture Models

ETMs were introduced by Zambal et al. [67] as an extension of *Active Appearance Models (AAMs)* [11] and hence *Active Shape Models (ASMs)* [12]. All three approaches aim at matching shapes and objects. While they have the same representation of shape, the handling of texture differs. In the following, the basics of statistical models of shape and texture are outlined in Section 4.1.2.1. Furthermore the limitations of ASMs and AAMs are shown, which motivate the approach from Zambal et al. [67]. Building of their entropy-based texture model and matching of unseen data with it are described in Section 4.1.2.2 and Section 4.1.2.3, respectively.

4.1.2.1 ASMs and AAMs – Basics and Limitations

When building a shape model for a specific object, manually placed landmarks at the boundaries of the object of interest are needed. They delineate the object and are necessary to ensure correspondence across the training images. Landmarks are further used to align the shapes and derive a mean shape. The variation within the training dataset is described by their principal components (also called eigenvectors), which are determined through Principal Component Analysis (PCA). PCA reduces the dimensionality of the underlying data to dimensions which exhibit the highest variance within the data. The first eigenvector points into the direction of the highest variance. All other eigenvectors are orthogonal to this eigenvector [30]. Every training data shape and also new shapes can be described by the mean shape and a linear combination of the main modes of variation, i.e. the main principal components [12].

In ASMs, local statistical models describe the varying appearance around landmarks. For each landmark such a model is built separately. They guide the shape model matching in order to find the correct object boundary. As the shape model, these models are described by PCA [12], [54].

AAMs extend the texture representation from ASMs and incorporate not only local appearances at the landmark positions. The model covers the full texture, that lies within the convex hull spanned by the landmarks [11]. In order to extract the texture for the texture model in a consistent way, the training images are mapped onto the mean shape. This way, a "shape-free" representation is obtained, which is raster scanned into a texture vector (for every mapped training image). From these vectors, the texture model is built with PCA. Both models are combined into a single model by learning the correlation between shape and texture [11].

AAMs provide a fast, accurate and robust solution for matching shape and texture simultaneously [11]. However, Zambal et al. [67] review some aspects of the texture model and image matching in general, that inspired their approach, which also apply for this work:

- For the datasets observed by the authors, texture eigenvalues decrease much slower than shape eigenvalues. Hence, PCA may not be the optimal choice for modeling texture. One of the datasets mentioned in their work covers different MRI protocols (including T1w and T2w scans), which is also the case for the datasets evaluated in this thesis (see Section 6.1 for details).

- Usually the training textures are normalized before model building and matching. This keeps unwanted texture variations out of the model, like for example varying lightning conditions. Kittipanya-ngam and Cootes [34] reviewed several non-linear texture preprocessing algorithms on raw and gradient filtered image data, e.g. Gaussian smoothing or exponential distance functions. These methods increase the performance of AAMs, but in practice it is difficult to predict which method is good for which kind of data [67].

- Besides the mentioned texture variations, also tissues showing low contrast, e.g. lung in MR scan, or having a fuzzy appearance, e.g. spongy bone, compound texture model matching. When matching a new image, the goal is to find a model instance that is closest to the target image. The difference measures used to determine the model mismatch are usually based on texture differences. A commonly used criterion is the sum of squared differences between intensity values [34]. It is assumed, that large texture differences come along with a misalignment of the model. Fuzzy structures and low tissue contrast can lead to model misplacement.

To tackle the issues mentioned, Zambal et al. [67] propose to measure texture difference based on entropy, inspired by the mutual information criterion. It is commonly used in medical image registration, because it enables registration of images with different modalities [49], [61]. For the matching of unseen images, Bayesian reasoning is applied, which is wide-spread within pattern recognition [67].

4.1.2.2 ETM Model Building

In general, the construction of an ETM is similar to building an AAM. An annotated set of $m = |\mathcal{S}_{tr}|, k = 1 \dots m$ training images $I_k \in \mathcal{S}_{tr}$ with a fixed number of l corresponding landmarks is needed. Based on these landmarks, and texture T_k is extracted from every training

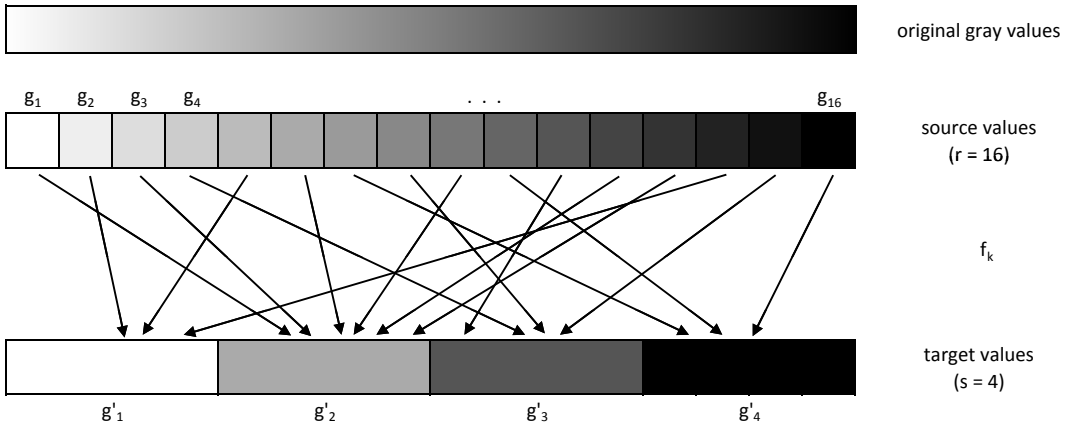


Figure 4.1: Schematic overview for the mapping within ETMs. The original gray values are quantized to $r = 16$ source values. For those, the authors try to find mappings f_k to the target value scale [67]. This sample shows the mapping to $s = 4$ target values.

image I_k. Every texture is represented by N *texels* and covers the same anatomical region of interest. A texel (or texture element) is an element of a texture, analogous to a pixel in an image.

The original gray values present in the image are quantized to r_k *source values* (also referred to as *source bins*), which leads to a binning of the original grayscale (see Figure 4.1). From the extracted textures T_k, the shape model is built by PCA in the same way as in ASMs and AAMs. For the texture model, the authors incorporate the following concepts.

Probability-based Modeling of Texture In AAMs, the variation of the texture is also modeled with PCA. For ETMs, Zambal et al. [67] suggest a different representation. They propose, that every model texel $t_j, j = 1 \dots N$ captures the statistical intensity variation observed at the corresponding texel t_j in the set of extracted training textures. One goal of the authors was, to keep unwanted intensity variations out of the model, e.g. due to different imaging modalities or noise. Therefore, they suggest not to model the source values r_k in the texture. Instead, Zambal et al. [67] map these values to a generalized scale (see Figure 4.1), which consists only of a few target bins. These values are then modeled in their texture model.

The goal is now to find such a mapping f_k for every training texture T_k, where source values r_k are mapped to s *target values* (also referred to as *target bins*) – formally:

$$f_k \colon \mathbb{Z}_{r_k} \to \mathbb{Z}_s, \qquad s \ll r_k, \qquad k = 1 \dots m \tag{4.3}$$

Hence we obtain an *individual mapping* f_k for every training texture T_k. A mapping assigns all N training texels (from an underlying texture T_k) a target value $g'_i \in \{1 \dots s\}$. As said before, every model texel t_j captures the variability of mapped intensities of training texels. That means for a texel t_j, m occurrences of the possible s target values are observed. These observations are interpreted as *probability density functions (PDF)* p_j (see Figure 4.2), where every PDF p_j

describes the variation within the corresponding texel t_j. The probability $p_j\left(g_i'\right)$ denotes then the probability, that the target value g_i' is observed at texel t_j.

Based on this, a quality criterion for mappings f_k is derived, which optimizes the mappings from source bins r_k to s target bins.

Entropy-based Quality Function Zambal et al. [67] suggest to assess the quality based on the predictability of the mapped bins. Figure 4.2 gives an example for a model with $N = 10$ model texel: The predictions for texel t_2 show only a single peak, while the PDF of t_{10} rather follows a uniform distribution. This suggests, that the information content of the predictions by the PDFs is higher for texel t_2 than for t_{10}.

In general, the information content of an image can be estimated by the entropy H [54]. It is a measure for the amount of uncertainty of an event (i.e. a random variable). In case of information entropy, this event is associated with a given PDF. Entropy increases, if the amount of uncertainty rises. This results in less predictable events [54]. That means, that for the presented example (see Figure 4.2), the entropy of t_{10} is higher than for t_2 because the reliability of the target value prediction is lower for texel t_{10} than for t_2.

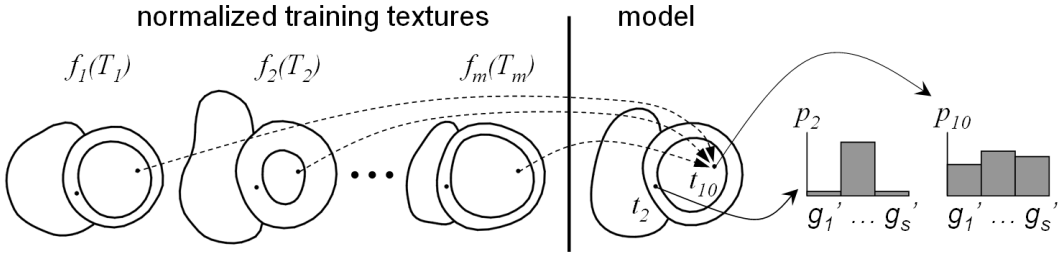


Figure 4.2: Normalized training textures with corresponding landmarks (left) and the learned model (right). Every texel t_j models the target value variation $g_i' \in \{1 \dots s\}$ with its PDF p_j. Image from Zambal et al. [67].

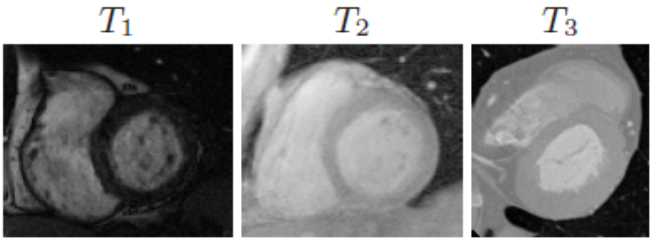
The authors [67] suggest to favor a reliable prediction over an uncertain prediction by minimizing the entropy of a corresponding PDF p_j:

$$H\left(p_j\right) = -\sum_{i=1}^{s} p_j\left(g_i'\right) \log_2\left(p_j\left(g_i'\right)\right) \tag{4.4}$$

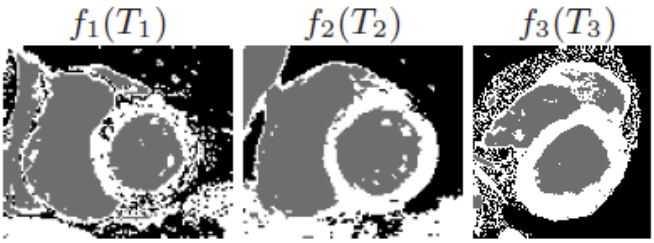
In order to increase the reliability of mappings, the entropy has to decrease for all N model texels. This results in the following cost function for the model, which aims at yielding a minimum:

$$H^{model} = \frac{1}{N} \sum_{j=1}^{N} H\left(p_j\right) \rightarrow \min \tag{4.5}$$

However, the transformed training images contain no information about their initial appearance anymore. Important structures could be lost, due to the reduction to the target values.



(a) original



(b) optimized

Figure 4.3: Three training images (see Figure 4.3a) and their mapping to $s = 3$ target values (see Figure 4.3b) are shown. From the training images, two are MR scans and one is a CT scan. Images from Zambal et al. [67].

The extreme case would be, that all source values r_k map to only one target value. Hence, the authors [67] propose to consider also the training textures T_k in the formulation of a cost function for the model building. The information content gained from the mapped training textures $f_k(T_k)$ should be maximized.

The *image entropy* $H(f_k(T_k))$ measures the information content in training texture T_k. In order to maximize the information content across the entire training set, one aims at maximizing

$$H^{tex} = \frac{1}{m} \sum_{k=1}^{m} H(f_k(I_k)) \rightarrow \max \tag{4.6}$$

The final cost function combines Eq. 4.5 and Eq. 4.6 and maximizes the information content in the training images while minimizing the uncertainty of the model:

$$\{f_1^\star, \ldots, f_m^\star\} = \underset{\{f_1,\ldots,f_m\}}{\operatorname{argmax}} \left(H^{tex} - H^{model} \right) \tag{4.7}$$

Figure 4.3 shows a training set of MR and CT scans and their mapping to $s = 3$ target bins.

4.1.2.3 ETM Model Matching

If matching an ETM, only shape parameters (scale, position and statistical shape parameters) are optimized by changing them randomly within their valid ranges while matching. This is in contrast to AAMs, where also the texture parameters are optimized iteratively [11].

If an unseen image I_k should be matched, a new model instance is initialized first. As with ASMs and AAMs, a good initialization regarding the position of the model instance is crucial. Zambal et al. [67] initialize their models close to the correct contours of the anatomical structure, which was learned by the model. The texture, which is currently overlapped by the model is denoted as texture $U = (u_1, \ldots, u_N)$ of the unseen image.

Two aspects are important for the matching of an unseen image with ETMs. First, a suitable initialization method of the texture U has to be formulated. Second, a criterion function has to be derived, which evaluates the fitness of the current shape instance S to the unseen image [67].

Texture Extraction and Mapping from an unseen Image According to the current shape instance S, the overlapped texture U is extracted. Since the model texels t_j capture the variation of target values $g'_i \in \{1 \ldots s\}$, the extracted texture has to be mapped to this scale as well. The task is to find a mapping

$$f_u \colon \mathbb{Z}_{r_u} \rightarrow \mathbb{Z}_s, \tag{4.8}$$

which normalizes the texture from r_u source values to s target values. Note that the original gray values of the unseen dataset have to be quantized to r_u source values before the optimization starts in order to be correspondent to the learned model. For the image quantized to r_u source values, Zambal et al. [67] propose to assign the target values, which lead to the *maximum likelihood* of the mapping f_u (i.e. of the texture). All model texels t_j, which observe a given gray value $\hat{u} \in \{1 \ldots r_u\}$, contribute to the likelihood according to their PDFs. The target value $g'_i \in \{1 \ldots s\}$ leading to the maximum likelihood is assigned – formally:

$$f_u(\hat{u}) = \underset{g'_i}{\operatorname{argmax}} \prod_{u_j = \hat{u}} p_j\left(g'_i\right) \qquad i = 1 \ldots s \tag{4.9}$$

Cost Function based on Bayesian Inference The quality of the current mapping is determined by comparing the normalized texture U' to the model PDFs. The authors suggest to apply Bayes' law [6] and maximize the *posterior probability* of $U' = f_u(U)$ subject to the shape parameters S (i.e. a shape instance).

In general, the posterior probability is proportional to the product of a likelihood and a prior probability (see Eq. 4.10, left) [6]. Within ETMs, the *posterior* $P(S|U')$ describes the probability of the current shape instance S, given the normalized texture U'. The *likelihood* $P(U'|S)$ describes the likelihood of the observed texture U', given the shape instance S. As reported by Zambal at el. [67], it is calculated as product of the probabilities of the normalized texels. The calculation of the *prior probability* $P(S)$ of the shape instance S is performed with a multivariate Gaussian distribution, which corresponds to the shape space (modeled with PCA). For details see Bishop et al. [6].

The combination of the introduced terms results in the final cost function [67]:

$$P\left(S|U'\right) \propto P\left(U'|S\right) P(S) = \left(\prod_{j=1}^{n} p_j\left(f_u(u_j)\right) \right) P(S) \rightarrow \max \tag{4.10}$$

4.2 Feature Extraction

Feature extraction methods aim at detecting specific structures in the data and reduce therefore the search space for subsequent image analysis or pattern recognition tasks [6]. A *feature* describes a property of the given image data, which provides information about a specific structure in the data [29]. Examples for features are lines, edges and corners or more complex structures like curvatures or texture information. As such, thresholding is also a simple image feature, which is calculated point-wise for the underlying image data [44]. A detailed review on different image features and feature representation is provided by Jähne [29].

In general, feature extraction algorithms can be differentiated into *low- and high-level methods* [44]. Low-level algorithms derive features automatically from an image. These features describe localities at a given point or region by considering only small sub-regions of interest in the image, for example edges, corners or curvatures. High-level feature extraction methods incorporate relations between features in order to find specific objects and shapes. One way to do so is by matching templates on the unseen data, e.g. with the Hough Transform. It is an efficient algorithm, which enables matching of simple structures like lines, but allows also for more complex, arbitrary shapes [4]. Another common approach is to define high-level feature detectors as a combination of extracted low-level interest points. This method enables building of more complex feature detectors and classifiers.

Feature detection methods can also be differentiated regarding their defined dimension. Generally, most of the common methods (e.g. sobel, canny or laplacian operator) are originally defined in 2D [44]. Some methods can be easily extended to 3D by combining the 2D extraction results, like for example the sobel edge detector (see Section 4.2.2). However, defining an extraction algorithm in 3D can become computationally too expensive regarding the temporal performance. This is the case for instance for the popular SIFT (Scale Invariant Feature Transform) feature detector, which was presented by Lowe [37], [53].

Figure 4.4 gives an overview of a selection of feature extraction methods, which are explained in the following. *Haar-like features* and *image gradients* are introduced as low-level feature extraction methods in Section 4.2.1 and Section 4.2.2, respectively. Building of a high-level classifier with extracted low level features is explained on PBTs in Section 4.2.3. A variant of those is further introduced in Section 4.2.4.

4.2.1 Haar-like Features

Viola and Jones [60] observed for face detection, that often the eye region is darker than the cheeks or the forehead. To detect this region in a facial image, they suggest to calculate the *intensity difference* between these two regions. If we look on images of the spinal column, we can observe, that also neighboring dark and bright regions are present, e.g. between intervertebral disks and vertebrae.

Viola and Jones [60] proposed a low-level feature, which considers this difference between sums of intensities within rectangular areas in the underlying image data – so called *Haar-like features*. The areas are adjacent and usually of the same size (see Figure 4.5, right).

In order to compute these sums of intensities fast and efficiently, Viola and Jones [60] introduced the concept of *integral images*. The integral image $II(x, y)$ at position (x, y) is defined

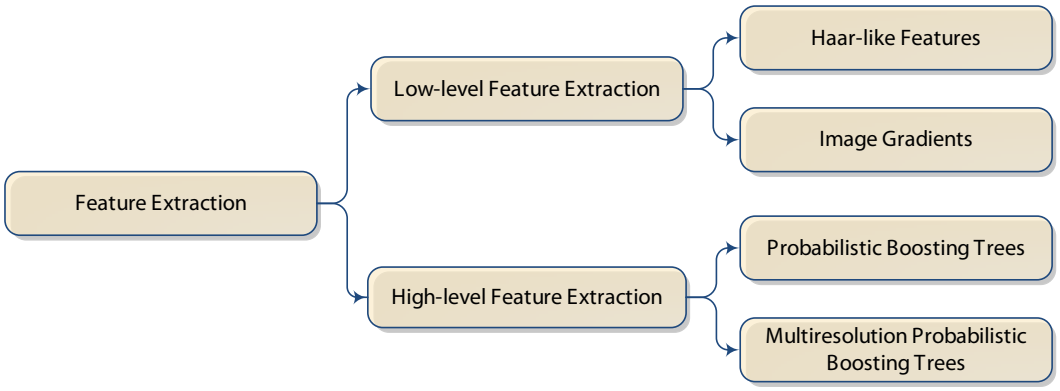


Figure 4.4: Overview of selected low- and high-level feature extraction methods.

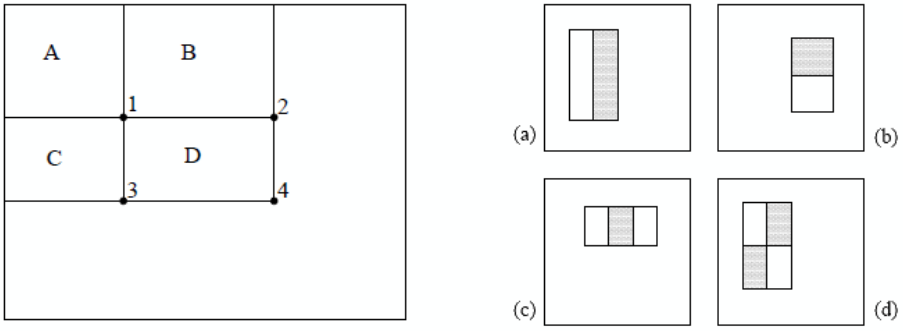


Figure 4.5: The value in the integral image at certain positions (left): at position 1, it is the sum of pixels in area A, at position 2 it is $A + B$, and so on. The sum within area D can be written as $4 + 1 - 2 - 3$. Examples for two-rectangle ((a) and (b)), three-rectangle (c) and four-rectangle (d) Haar-like features (right): only six, eight and nine lookups in the integral image are necessary to compute two-, three- and four-rectangular structures, respectively. Image from Viola and Jones [60].

as the sum of pixels above and to the left of position (x, y). Formally

$$II(x, y) = \sum_{x' \leq x, y' \leq y} I\left(x', y'\right), \tag{4.11}$$

whereby $I(x, y)$ denotes the intensity in the original image I at position (x, y). Any sum of a rectangular area in I can be computed by a minimum of four lookups in the integral image II (see Figure 4.5, left). In order to obtain the value of a feature, the areas are subtracted. For the three-rectangle feature (see Figure 4.5c) for example, the sum of the two white rectangles on the outside is subtracted from the intensity sum of the mid-rectangle.

Different patterns can be constructed with Haar-like features and used to detect also more complex characteristics in the data [60]. Furthermore, the concept of integral images can be extended to three dimensions. *Integral volumes* sum up the intensities also in z-direction – formally:

$$II(x, y, z) = \sum_{x' \leq x, y' \leq y, z' \leq z} I\left(x', y', z'\right). \tag{4.12}$$

4.2.2 Image Gradients

Rapid intensity changes between adjacent pixels in an image I are often of interest in image processing, since they delineate edges and object boundaries. This changes can be detected by differentiation between these points, i.e. pixels [54], [44]. So called *gradient operators* aim at approximating local derivatives, whereby a bigger derivative is obtained at image locations which undergo rapid intensity changes (e.g. sharp edges). Horizontal changes are determined by vertical differentiation of adjacent intensities at pixels and vertical changes by horizontal differentiation [44]. Mathematically the gradient of an image I in x-direction (i.e. the horizontal change) is defined as the partial derivative of the intensities in x-direction:

$$\nabla I_x = \frac{\partial I}{\partial x}. \tag{4.13}$$

In image processing, the partial derivative is usually approximated by convolving the original image with a filter mask. One of the most commonly applied filters in practice is the *sobel filter* [44]. Its masks for detecting horizontal (M_x) and vertical edges (M_y) are defined as follows:

$$M_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{4.14}$$

The approximated partial derivatives $\nabla I'_x$ and $\nabla I'_x$ in x- respectively y-direction are then obtained by convolving the image data I with these masks separately (see Figure 4.6):

$$\nabla I'_x = M_x * I \tag{4.15}$$

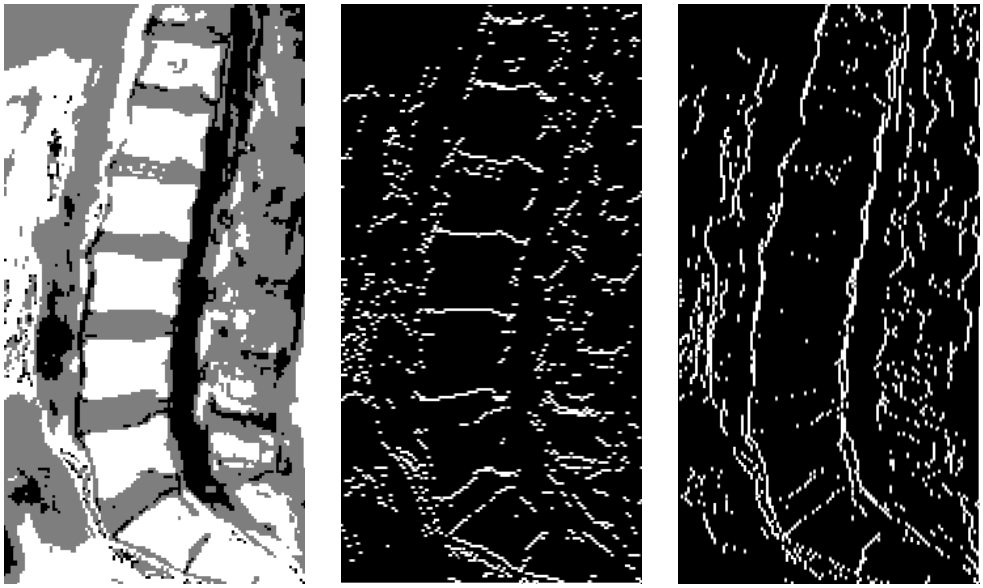
$$\nabla I'_y = M_y * I \tag{4.16}$$

The overall strength of an edge in an image is calculated by combining the partial derivatives in every direction. The resulting edge magnitude, or gradient magnitude, G_{xy} (see Figure 4.7) is defined by

$$G_{xy} = \sqrt{\nabla I'^2_x + \nabla I'^2_y}. \tag{4.17}$$

The calculation of the partial derivatives can be extended to 3D, by convolving the volume I not only with M_x and M_y, but also with a mask M_z. The obtained approximated partial derivative $\nabla I'_z = M_z * I$ then depicts the edge strength in z-direction. The overall gradient magnitude is calculated analogously to Eq. 4.17:

$$G_{xyz} = \sqrt{\nabla I'^2_x + \nabla I'^2_y + \nabla I'^2_z}. \tag{4.18}$$



(a) Optimized image (b) Horizontal edges (c) Vertical edges

Figure 4.6: Entropy-optimized T1w MR scan with $s = 3$ gray values and detected edges by filtering with M_x (horizontal edges) and M_y (vertical edges). Images based on data by courtesy of AGFA HealthCare [25].

4.2.3 Probabilistic Boosting Trees

While the previous sections introduced low-level feature extraction methods, a high-level approach combining those is reviewed in the following. The concept of *boosting* proposes that a *strong classifier* can be learned as a combination of *weak classifiers*, which was confirmed by Schapire [51]. A weak classifier gives only a slightly better classification result as random guessing, i.e. $> 50\%$. A strong classifier in contrast combines these weak classifiers and gives a prediction, that is correlated with the true classification result.

Tu [59] uses this concept and introduces PBTs. They are a general framework for learning a conditional posterior probability, which is referred to as *strong classifier*, from a set of weak classifiers (evidence, knowledge). This is inspired by AdaBoost [21], which approximates the posterior distribution. AdaBoost and decision tree algorithms in general learn one strong classifier from weak classifiers. The disadvantage of these methods is, that they may have to pick hundreds of weak classifiers. With PBTs, one tree classifier is learned recursively, where *every tree node* itself is a strong classifier.

Training Phase If we want to train a classifier for a two-class, i.e. binary, classification problem, we learn a *binary tree*. In the training phase, the tree is built recursively with a divide-and-conquer strategy. Starting at the root node of the tree, a strong classifier is learned with the boosting algorithm, i.e. weak classifiers are selected to build a strong classifier. As weak classifiers act likelihood classifiers on extracted image features, e.g. Haar-like features, image

Figure 4.7: Combined gradient G_{xy} from Figure 4.6b and Figure 4.6c. Image based on data by courtesy of AGFA HealthCare [25].

gradient, etc. All samples are then classified with the learned classifier (i.e. the posterior distribution), and propagated down into the left or right sub-tree, depending on their assigned class. In this way, samples which are hard to classify are passed further down in the tree, which leads to an expansion of it. To control overfitting of the learned tree node classifier, a margin is defined. Classified samples which fall into this region are passed down into both sub-trees. Training continues till all samples are classified correctly or a maximum tree depth is reached.

Classification with PBTs The classification of an unseen image works accordingly to the training (see Figure 4.8). For every sample, the posterior probability is calculated at every node of the tree and gathered in the root of the tree. It reports the overall approximate posterior probability for the sample. In order to classify a sample (i.e. pixel or voxel), the posteriors for the left and right sub-tree are calculated at every node. Starting from the root at level $l = 0$, the classification result for the sample is obtained by following the respective sub-trees with the higher posteriors till the lowest level l_n is reached.

The presented algorithm provides a very general framework, which can be used for classification, object recognition or segmentation. PBTs provide good computational and classification performance, but are far too slow for real time classification [59], [53]. Another issue is the overfitting problem, especially with a limited number of training samples. Recent publications on MR and CT spine labeling by Kelm et al. [32] and Major et al. [38] apply PBTs for feature detection as well. The latter use an improved, multiresolution approach, which is also integrated in this thesis and described in the following.

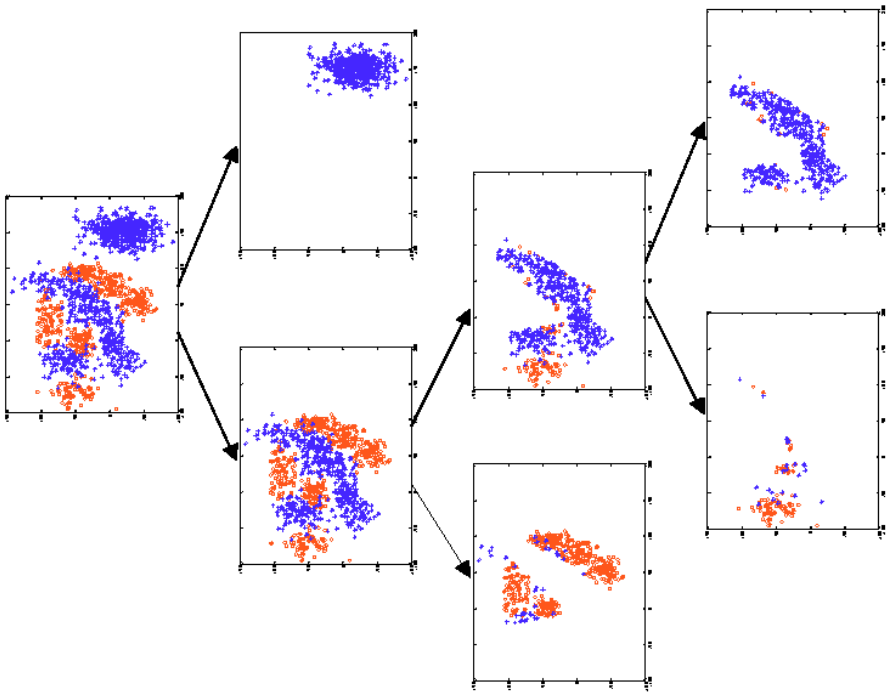


Figure 4.8: A synthetic dataset with 2000 points is classified by a PBT. The tree expands on nodes where positive and negative samples are present till a maximum depth is reached or all samples are classified correctly. At every node, the list of weak classifiers is evaluated, which was learned for this specific node in the training phase. A weak classifier is a likelihood classifier on a feature, e.g. a Haar-like feature or a distance feature. Image from Tu [59].

4.2.4 Multiresolution Probabilistic Boosting Trees

Schulze et al. [53] aim for a memory efficient feature extraction algorithm, which provides a detection result within seconds and is applicable especially on medical data. To accomplish this goal, the authors introduce several concepts in order to speed up the PBT approach from Tu [59] presented in the previous section.

Partial Cascading Viola and Jones [60] introduced a cascading framework together with AdaBoost. A cascading node discards negative classified samples and passes only positive samples down further in the tree (see Figure 4.9 (left)). Schulze et al. [53] propose so called *partial cascading*, where they combine cascading and boosted tree nodes in one tree. The benefit of cascading nodes is, that they reduce the number of classification tests and hence speed-up further processing. The advantage of boosted nodes on the other side is, that they capture a high variability of features, which increases flexibility. The authors suggest to add a cascading root node before the standard PBT, as seen in Figure 4.9 (right). A large number of voxels is therefore discarded in the beginning and positive samples are further classified by a PBT. Cascading nodes may also be inserted on other levels in the decision tree.

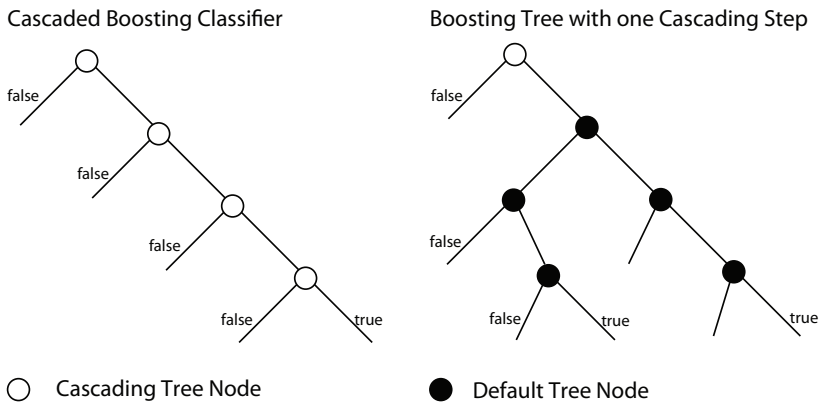


Figure 4.9: Cascaded sequence of classifiers (left) and PBT with cascading root node (right). Only positive paths are further processed. Image from Schulze et al. [53].

Classifier Sorting To further increase execution speed, fast performing classifiers should be preferred over expensive classifiers in tree nodes often visited. The cascading root node is visited most often and in an early classification stage. Hence this node is built with cheap classifiers with low computational costs, such as likelihood classifiers on Haar-like features and image intensities. They eliminate a large number of false candidates in the very beginning. More complex classifiers, e.g. gradients, principal curvatures or structure tensors, are moved further down in the PBT.

Multiresolution PBT In order to decrease the search space already in an early processing stage, boosting trees are also trained on downsampled data. For this purpose, a Gaussian image pyramid is built from the original volume. A separate PBT classifier C_w is learned for every resolution level $w \in \{0 \dots W\}$ (see Figure 4.10), whereby the original image has resolution level $w = 0$, i.e. no downsampling is performed. A new image I is classified on the lowest resolution level W with classifier C_w first. With the classifier C_w every voxel is classified and only those who were accepted by the classifier are candidates for the next higher level, i.e. for classifier C_{w-1}. Iteratively only accepted candidates are propagated till the original resolution is reached, i.e. $w = 0$.

Memory Management In order to decrease the occupied memory, only the original volume data is kept in the memory. Lower resolution volumes, gradients and curvatures are calculated block wise on demand. Calculated blocks are kept in a cache. If the memory gets too low for allocating new blocks, blocks which have been accessed the longest time ago are replaced by new ones. The classification of a dataset now follows two strategies: First, it follows resolution order, starting with the lowest resolution. Only positive samples are propagated to the next level. Within a resolution level, the classification is performed in a block-wise manner. Therefore only a small number of blocks must be cached.

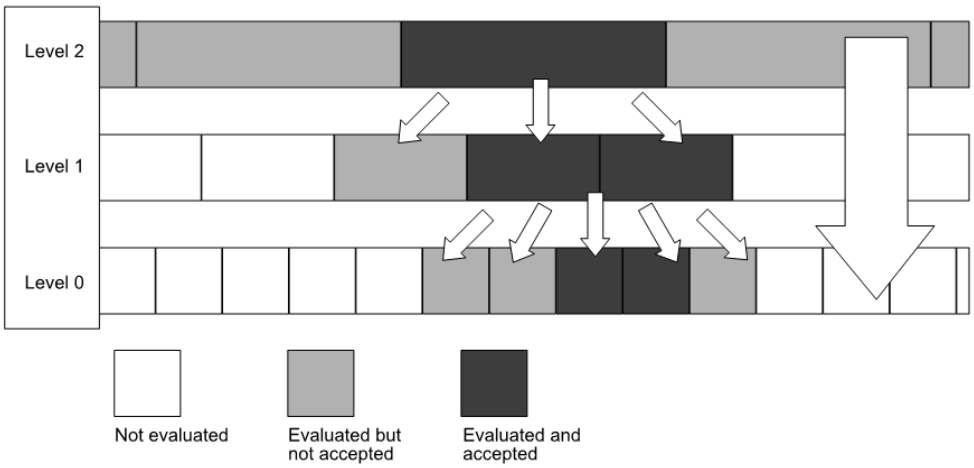


Figure 4.10: The multiresolution approach, where a strong tree classifier is evaluated in every resolution level. Only accepted voxels are propagated to the next higher resolution level. Rejected voxels at a low resolution are not taken into account in higher levels. In this example, we start at level $w = 2$ and propagate the accepted voxels to level $w = 1$. Voxels, which are accepted in this level are then passed down to the original resolution level $w = 0$. Image from Schulze et al. [53].

CHAPTER 5

Semi-automatic Spine Labeling on Normalized MR Data

The methods, which were reviewed in Chapter 4, are combined to a novel pipeline for semi-automatic labeling of the lumbar spine. Section 5.1 presents our proposed approach and explains its main steps. Section 5.2 details the implementation of the bias field preprocessing. The building of the model used for spine labeling is introduced in Section 5.3. Section 5.4 presents the application of the learned model on an unseen MR scan. Finally, Section 5.5 concludes the chapter with technical details on the implementation.

5.1 System Overview

Figure 5.1 gives an overview of the proposed, novel semi-automatic spine labeling pipeline and shows, how the previously presented algorithms will be applied. The input to the system is a T1w or T2w MR volume dataset, which is reconstructed from a stack of sagittal 2D slices. In the following, we will refer to a 3D volume dataset with I_k, i.e. 3D image data. Within this work, we focus on labeling of the *lumbar spine*. However, this approach should be applicable to other parts of the spinal column as well.

As *first step* in the pipeline, the dataset is preprocessed in two stages. The intensity inhomogeneities are removed with the bias field correction method, which was presented by Juntu et al. [31]. Afterwards, the dataset is normalized to a reduced scale with ETMs, presented by Zambal et al. [67]. The output of the first stage is a *normalized MR volume*. In the *second step* of the pipeline, intervertebral disks are detected by a feature detection framework. Fast and memory efficient PBTs, presented by Schulze et al. [53], were trained with image gradients and Haar-like features [60]. The detection framework delivers candidate points for every disk, whereby one point is selected as disk center and labeled with its anatomical label. The output of the system is the labeled input MR scan.

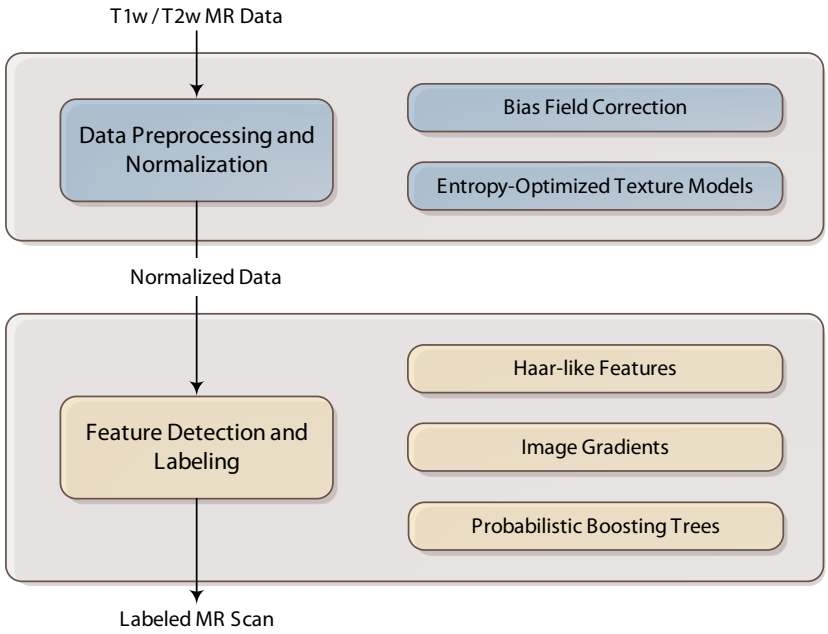


Figure 5.1: Overview of the suggested semi-automatic spine labeling pipeline.

Except for the bias field correction method, which is not a machine learning approach, the spine labeling system is *learning-based*. That means, we have to train a model first, in order to be able to normalize MR scans and label the spinal column. If a new, unseen dataset comes in, we apply the model and obtain the desired labeling. Figure 5.2 presents the main steps for the training of a model for spine labeling and its application to a new MR scan.

The input to the *model building* pipeline are annotated T1w and T2w volumes and the parameter configuration for ETMs, e.g. number of source bins r, number of target bins s, and PBTs, e.g. depth of boosting trees, initial resampling voxel size $\Delta_{x,y,z}$. The output of the model building phase is a learned ETM and trained feature detectors for lumbar Φ_L and thoracic Φ_T intervertebral disks. In the following, we will refer to a learned entropy model with $\mathcal{M}$. It has a specific parameter configuration, which will be introduced in Section 5.3.2.

If a new MR volume dataset has to be labeled, the learned spine model is applied. The input to the labeling system are the trained ETM $\mathcal{M}$, the feature detectors Φ_L and Φ_T, the unseen T1w or T2w volume dataset and an input from the user. This is on the one hand a clicked start point (x, y, z), that lies within an intervertebral disk or vertebrae. On the other hand, the user has to provide the anatomical label of the point. The output of the system is the labeled MR dataset.

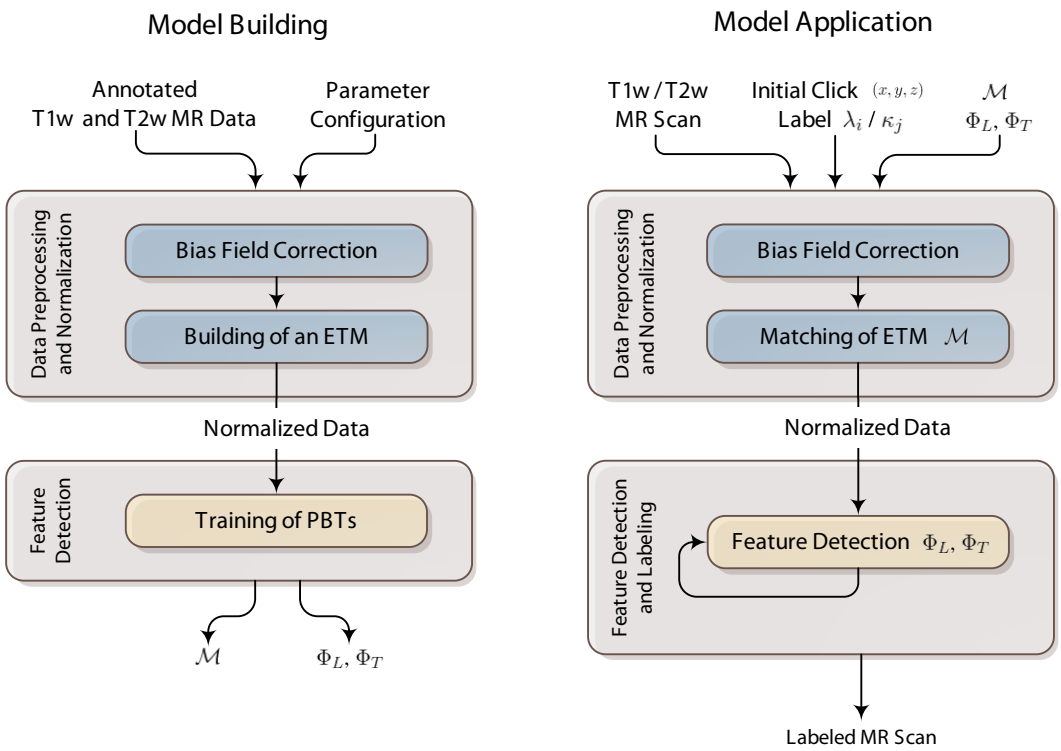


Figure 5.2: Overview of spine model building (left) and its application in the testing phase (right), i.e. the semi-automatic labeling of an unseen dataset.

5.2 Bias Field Preprocessing

The first step in the preprocessing pipeline is the bias field correction algorithm by Juntu et al. [31] (see Section 4.1.1). The preprocessing is performed before the model building respectively the model application. Therefore the following description is not included in both sections, i.e. Section 5.3 and Section 5.4. The correction algorithm was implemented in MATLAB [40] and all datasets were preprocessed in advance.

The presented algorithm [31] works with 2D images, therefore every slice of a volume I_k is processed by itself within our work. In the following, we refer to the e-th sagittal slice of a volume I_k with I_k^e.

For every sagittal slice image I_k^e, we extract its background image B_k^e. We convolve I_k^e with a Gaussian filter, which has 2/3 of the size of the slice. Based on the obtained filter size k, σ is calculated by $\sigma = \frac{k+1}{6}$. From the extracted background image B_k^e, n points are selected pseudo-randomly for surface fitting. A raster was constructed with a step size of 50 pixels and a second one with a step size of 25 pixels. This provides a good coverage of the whole image. Not all points at this raster were selected, but only those which belong to the *foreground*. The foreground region is obtained by thresholding over the background image. In order to avoid

image-dependent thresholds, the obtained background image B_k^e is normalized to the range $[0, 1]$ first. The background is denoted by pixel having intensities $i \leq 0.25$, i.e. the foreground image is $B_k^e > 0.25$. Several thresholds were tested, whereby with 0.25 the best results were achieved for all volumes I_k.

We fit a second order polynomial surface of the form

$$h(x, y) = a_1x^2 + a_2xy + a_3y^2 + a_4x + a_5y + a_6 \tag{5.1}$$

to the n points extracted from the background image B_k^e with the `fit` method provided by MATLAB [40]. Finally, the original slice image I_k^e is divided by the bias field calculated from the fitted polynomial. Figure 5.3 gives an overview about the images obtained in the different steps. The mid-slice image refers to the middle sagittal slice of the 3D MR volume, i.e. of the stack of sagittal slices. Two different variants for the bias field preprocessing were evaluated, which is reported in Section 6.3. The corrected sagittal slices were reconstructed and the preprocessed volumes were stored on the hard disk for further processing.

5.3 Model Building for Spine Labeling

Building a model for our proposed semi-automatic labeling pipeline involves several steps:

1. Training datasets $\mathcal{S}_{tr}$ have to be annotated by an expert with sparse landmarks covering the lumbar spine. The annotations used in this thesis are introduced in Section 5.3.1.

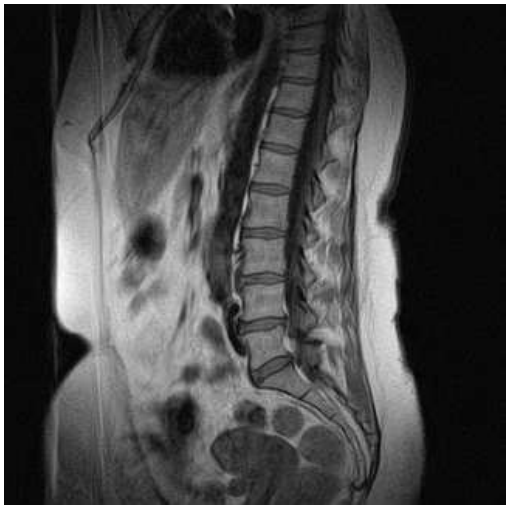
2. Based on this landmarks, an ETM is learned on the training MR scans. All necessary steps for building such a texture model are reviewed in Section 5.3.2.

3. Applying the learned model to the training data results in the desired normalized datasets on which PBTs are learned for interest point detection. Section 5.3.3 gives details about training of the feature detectors.

The input for the model building pipeline are the annotated training volumes $I_k \in \mathcal{S}_{tr}$ and a parameter configuration for the training of an entropy model and the boosting trees.

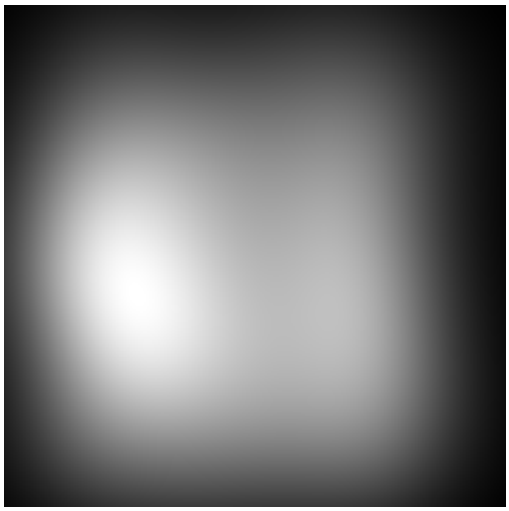
5.3.1 Dataset Annotations

The following introduced annotations (and notation of it) are based on the work by Major et al. [38]. From 14 datasets of different patients, a set $\mathcal{S}$ of 28 volumes $I_k \in \mathcal{S}, k = 1 \dots 28$ was extracted and annotated manually. Figure 5.4 visually describes the annotated landmarks, which are introduced in the following. For each disk i in volume $I_k \in \mathcal{S}$, the disk center d_i^k was annotated with an anatomical label $\lambda_i \in \{T8/T9, T9/T10, \dots, L4/L5, L5/S1\}$. Around every disk center d_i^k, a cylinder z_i^k defined by a radius, a height and with a certain orientation was placed. The cylinder is oriented in a way that it approximates the disk shape and covers only disk tissue. For every disk center d_i^k, a corresponding point c_i^k was set in the spinal canal. c_i^k lies in the center of the spinal canal, at approximately the same height as the corresponding disk center.

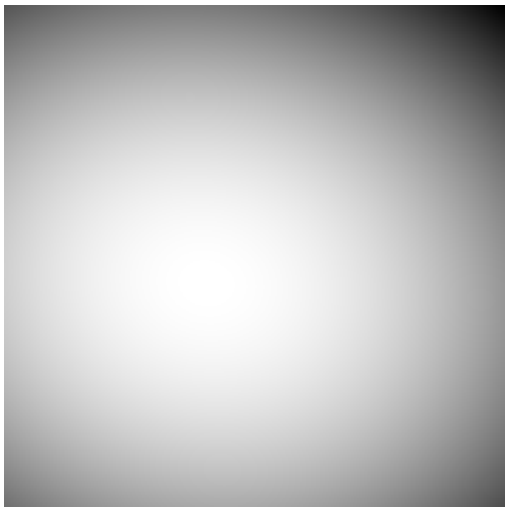


(a) Original mid-slice image

(b) Corrected image



(c) Background image

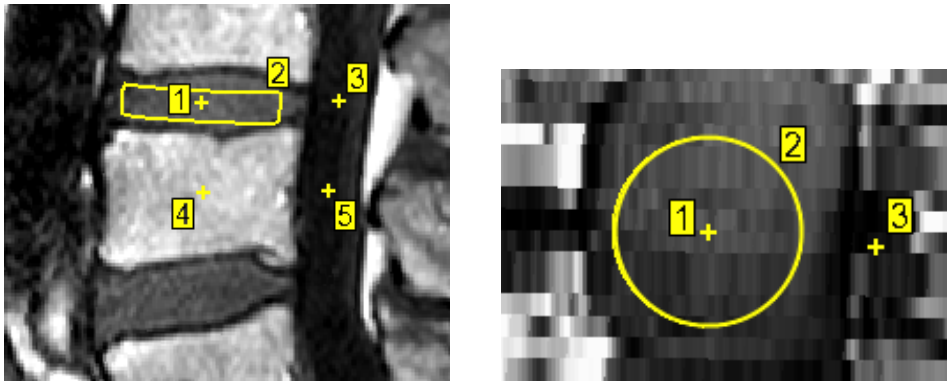


(d) Estimated bias field

Figure 5.3: Processing steps of bias field correction shown on T1w volume d17_t1w. Data by courtesy of AGFA HealthCare [25].

Vertebrae were annotated by placing a landmark in the vertebral body. The point v_j^k denotes the body center j in volume I_k with anatomical label $\kappa_j \in \{T9, T10, \ldots, L5, S1\}$. Analogously to the disks, for every vertebra body center v_j^k, a landmark was placed in the spinal canal c_j^k.

The annotations are saved in XML-format (eXtensible Markup Language). Statistics about the size of disks and vertebrae are derived automatically, e.g. the mean lumbar disk height, the mean lumbar vertebra height, etc. The mean dimensions, e.g. radius and height, for individual disks, e.g. $L3/L4$, are also calculated. These statistics are used later on in the spine labeling pipeline.



(a) Sagittal view (b) Axial view

Figure 5.4: Illustration of annotated landmarks in the training volumes: (1) disk center d_i^k, (2) cylinder z_i^k, (3) corresponding spinal canal landmark c_i^k to disk d_i^k, (4) vertebra body center v_j^k, (5) corresponding spinal canal landmark c_j^k to vertebra v_j^k. Image based on data by courtesy of AGFA HealthCare [25].

5.3.2 ETM Spine Model Building

Several steps are necessary for building an ETM, which provides the desired intensity mapping. In contrast to Zambal et al. [67], the model used within our work is built from sparse annotations. It is aimed to cover the lumbar spine in order to learn the appearance and provide a good intensity mapping. The authors matched single objects (e.g. the heart) with dense landmarks located at the border of the object.

The main approach, as presented by Zambal et al. [67], is illustrated in Figure 5.5. Out of the total number of 28 annotated volumes, up to $m = 11$ were used in the training phase. From these datasets, corresponding landmarks are extracted and the shape model is built. Then the texture is extracted on the preprocessed scans. The mappings $f_k, k = 1 \ldots m$ from r source values to s target values are optimized iteratively with simulated annealing [33].

For the building of an ETM, several input parameters are necessary:

- number of source values r
- number of target values s
- sampling distance sd of the model
- landmark extraction method

Various ETMs were trained, with different parameter configurations. An extensive evaluation is provided in the Section 6.4. The individual steps (cf. Figure 5.5) for learning of an ETM are detailed in the following.

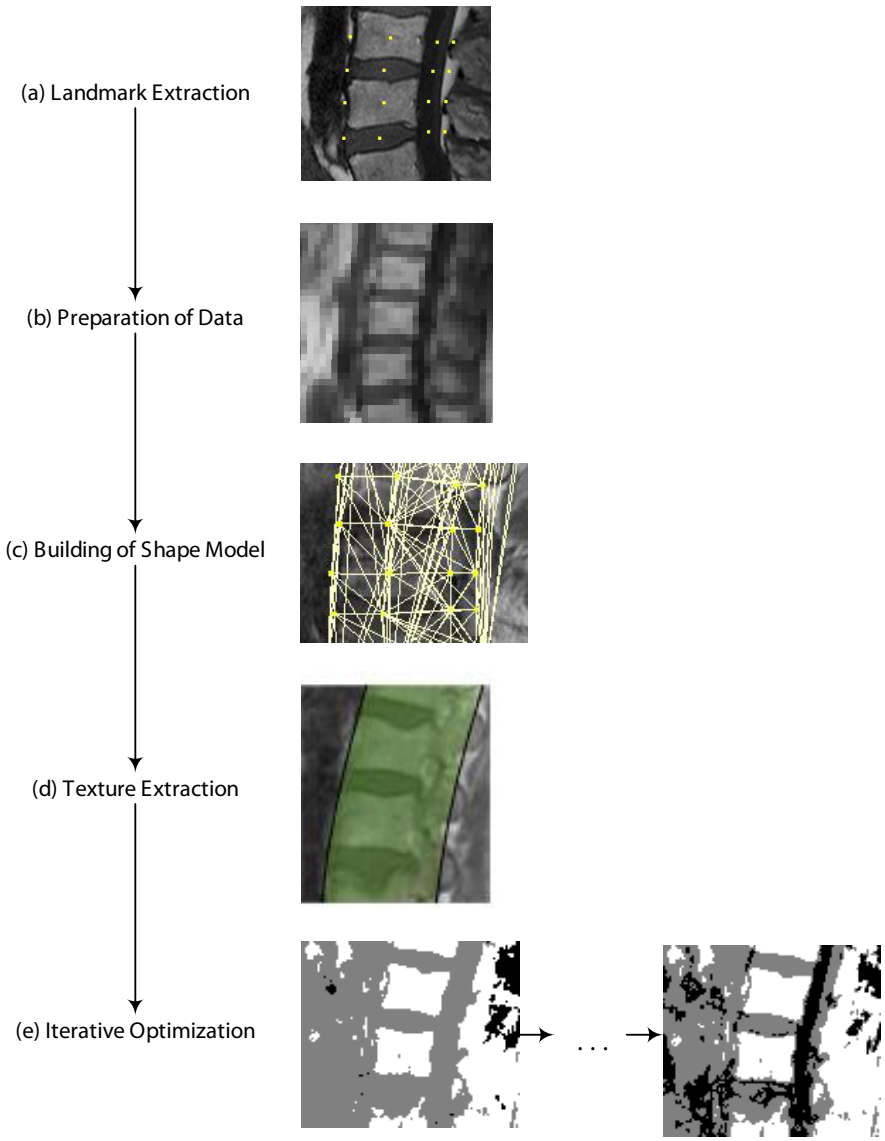


Figure 5.5: The main steps for training of an ETM. From the input data, landmarks are extracted (a) and the scans are aligned and downsampled if necessary (b). Then the shape model is built (c) and the texture extracted (d). Finally the training textures are optimized iteratively (e). Image based on data by courtesy of AGFA HealthCare [25].

(a) Landmark Extraction We integrated three different landmark extraction methods within this work. They are based on the sparse annotations of the MR volume data, as introduced in Section 5.3.1). We refer to the extraction methods as *Disk-Spinal-Centers-Plane*, *Disk-Cylinder-Points* and *Disk-Cylinder-Spinal-Points*. With these methods, different landmarks for model

building are extracted, which are illustrated in Figure 5.6.

- *Disk-Spinal-Centers-Plane* From the training datasets I_k, the disks centers d_i^k having label λ_i (point #1) and vertebra body centers v_j^k with label κ_j (point #4) are extracted (see Figure 5.6a). Additionally, their corresponding spinal center points c_i^k (point #3) and c_j^k (point #5) are obtained as well. Around these points, an artificial hull is constructed by adding extra landmarks. For a disk center c_i^k, a point is added in distance of the radius of cylinder z_i^k in anterior direction (point #2). For a vertebra body center v_j^k, a point is added according to the mean size of the vertebral body, also in anterior direction (point #7). The mean size is based on the derived statistics, as mentioned in Section 5.3.1. For every spinal point c_i^k and c_j^k, a point within a distance of 10 mm is added in posterior direction (point #6).

- *Disk-Cylinder-Points* The second extraction method is shown in Figure 5.6b. For every disk, the center d_i^k (point #1) and points at the surface of the corresponding cylinder z_i^k are extracted (points #2). They are calculated from the annotations, i.e. from the annotated disk cylinder center d_i^k, the height of the cylinder and its orientation. The centers from the top and bottom base are added to the landmark set. Points at the border of every base are sampled in an angle of $\alpha = 60°$, which results in 12 additional landmarks – 6 from each base. Furthermore, the vertebra body centers v_j^k (point #4) are taken into account.

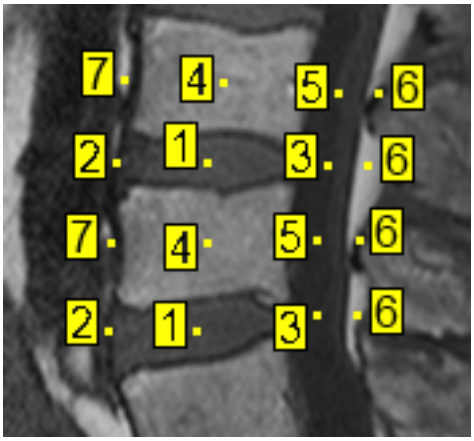
- *Disk-Cylinder-Spinal-Points)* This method extracts the same points as the previous one described. In addition, corresponding spinal points c_i^k (point #3) and c_j^k (point #5) are added in order to cover also the spinal canal (see Figure 5.6c).

(b) Preparation of Training Data The shapes defined by the previously extracted landmarks need to be aligned first in order to enable comparison between them. They exhibit variation in scale, translation and rotation, e.g. due to varying body sizes of the patients, pathologies of the spine or differences in image acquisition. These variations are removed by a Procrustes analysis [22]. In Procrustes analysis, the shapes are iteratively centered and aligned to a uniformly scaled mean shape, which is re-calculated after every iteration. In the first round, the mean shape is chosen arbitrarily. The alignment is finished when the mean shape converges. As optimization criterion, the Procrustes distance is minimized, i.e. the square root of the sum of squared distances between corresponding points. This means for two shapes:

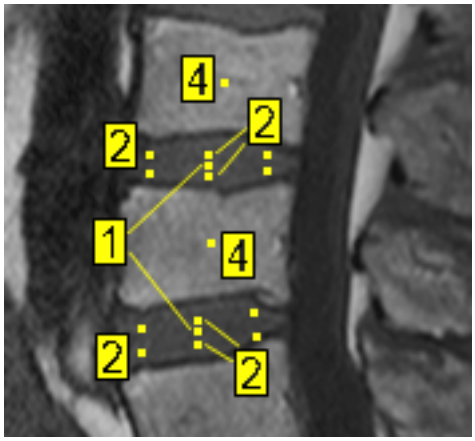
$$d = \sqrt{\sum_{i=1}^{l} (u_i - x_i)^2 + (v_i - y_i)^2 + (w_i - z_i)^2}\,, \tag{5.2}$$

whereby (u_i, v_i, w_i) and (x_i, y_i, z_i) denote the coordinates of the current corresponding landmark i, $i \leq l$ (i.e. the number of landmarks).

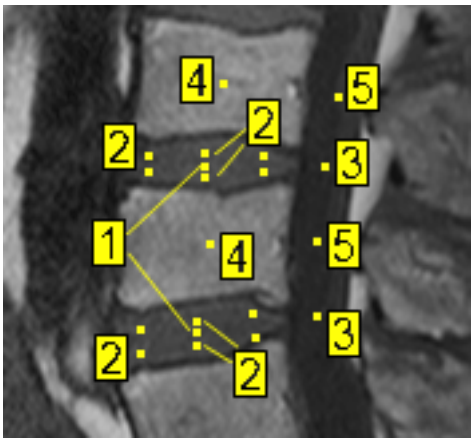
At the end of this stage, the training volumes may be downsampled. This depends on their individual voxel sizes and the *sampling distance* sd parameter of the model. It controls the voxel sizes of the volumes in the training set, i.e. that all volumes exhibit similar voxel sizes.



(a) Disk-Spinal-Centers-Plane



(b) Disk-Cylinder-Points



(c) Disk-Cylinder-Spinal-Points

Figure 5.6: The extracted landmarks for the three different methods shown on dataset d6_t1w: (1) disk center d_i^k, (2) points at the surface of cylinder z_i^k, (3) corresponding spinal canal landmark c_i^k to disk d_i^k, (4) vertebra body center v_j^k, (5) corresponding spinal canal landmark c_j^k to vertebra v_j^k, (6) additional landmark to a spinal canal landmark in distance of 10 mm, (7) additional vertebra landmark according to the mean size of the vertebra v_j^k.
Note that the points are projected on one slice, hence not all landmarks (points #2) are visible for the *Disk-Cylinder-Points* and *Disk-Cylinder-Spinal-Points* method. Images based on data by courtesy of AGFA HealthCare [25].

Based on the sampling distance sd, an individual downsampling factor β^k for every volume I_k is calculated – formally:

$$\beta^k = \left\lfloor \frac{sd}{\min\left(\Delta_x^k, \Delta_y^k\right)} \right\rfloor \tag{5.3}$$

It defines, if the underlying volume I_k is downsampled, i.e. $\beta^k \geq 2$, or not, i.e. $\beta^k = 1$. Δ_x^k

and Δ_y^k refer to the voxel sizes of I_k in x- and y-direction, respectively.

(c) Building of Shape Model For building of the shape model, PCA is performed on the aligned shapes. The obtained eigenvectors and eigenvalues describe the shape variation, as already reviewed in Section 4.1.2.

(d) Texture Extraction The *convex hull*, which contains all extracted landmarks l, is tetrahedralized first (see Figure 5.7). This is enabled by Delaunay Tetrahedralization. With the CGAL library [1], the subdivision of the convex hull is calculated. The result is a list of landmark pairs, whereby a pair a connection between the two landmarks depicts. On the tetrahedra, the texture is then extracted. The size of the texture, i.e. the number of texels t_j, depends on the extraction method and the number of voxels in the dataset. Hence the *Disk-Cylinder-Points* and *Disk-Cylinder-Spinal-Points* extraction method result in a larger texture than the *Disk-Spinal-Centers-Plane* method, because a larger region is covered. A small sampling distance sd also favors a large texture, because of a smaller voxel size of the training volumes. Table 5.1 gives an overview of the approximate size of textures with different sampling distances and extraction methods. All trained models cover the lumbar and lower thoracic region ($L5$ to $T10$).

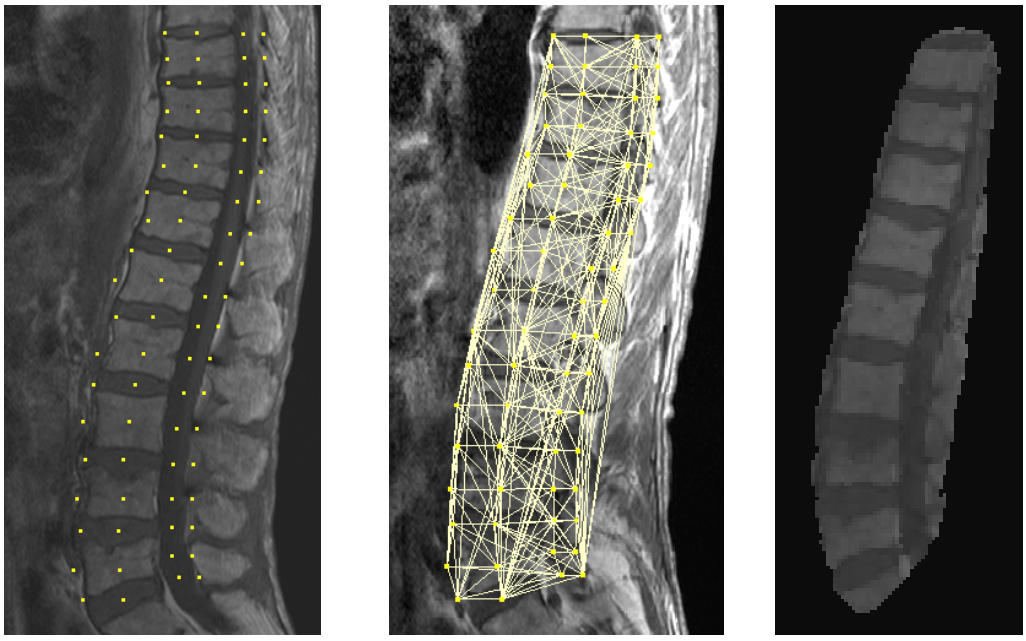


Figure 5.7: Mid-slice image extracted from volume I_1 with $l = 76$ landmarks (left), the obtained tetrahedralization based on the landmarks (middle) and the extracted texture T_1 with $N = 12163$ texel (right). Images based on data by courtesy of AGFA HealthCare [25].

When the texture is extracted, all original gray values present in the underlying volume data are quantized to a fixed number of r source bins first, as proposed by Zambal et al. [67]. Recall, that this number is a parameter of the ETM. This binning is performed in order to build the

Sampling \ Extraction	Disk-Spinal-Centers-Plane	Disk-Cylinder-Points	Disk-Cylinder-Spinal-Points
$sd = 1.5$	12200	42500	48000
$sd = 2.0$	5200	18000	20200
$sd = 2.5$	2500	9200	10400

Table 5.1: Approximate texture size in texels (rounded to hundreds) for the three landmark extraction methods *Disk-Spinal-Centers-Plane*, *Disk-Cylinder-Points*, *Disk-Cylinder-Spinal-Points* with increasing sampling distance sd.

statistical model, which maps from source bins to target bins. For datasets, which exhibit more than r gray values in the data, the intensity scale is reduced. However, no anatomical information is lost, which was tested for all datasets.

If the texture gray value band is extracted from the underlying volume dataset, the position and anatomical label λ_i of the intervertebral disk centers c_i^k in the texture is extracted as well. The position is in this case the index j of the texel $t_j, j \leq N$, where the intervertebral disk center d_i^k is located. This position is evaluated during the texture extraction with lookups in the annotation file. For vertebra body centers the same information is extracted. This results in a lookup table $\mathcal{T}$ of texel indices $j, j \leq N$. For every anatomical label, that occurs in the learned model, a list of size m including the texel indices is stored, e.g. $\mathcal{T}(L3)$ delivers a list of texel indices for vertebra $L3$. This information is used for the initialization of the entropy model in the matching phase, which will be introduced in Section 5.4.1.

(e) Iterative Optimization The extracted training textures are optimized iteratively with simulated annealing [33], as it was proposed by Zambal et al. [67]. The simulated annealing algorithm is a heuristic for the global optimization problem. However, it does not aim at finding the best global solution of a given problem, but rather to find a good approximation of the global optimum.

Zambal et al. [67], [20] propose to start with a subset of all training textures and add the remaining textures iteratively till all m training textures are included in the optimization. This should increase robustness and avoid to get stuck in a local minimum. They suggest to start the optimization with $m' = 4$ textures. The mappings f_k are represented by lookup tables and are in the beginning initialized randomly. In every iteration, the model entropy H^{model} and the entropy of the training images H^{tex} are calculated. As long as the combined entropy $H = H^{tex} - H^{model}$ improves, the statistical shape parameters, position and scale are changed randomly within their valid ranges and the lookup tables are updated. After 5000 optimization rounds without improvement of the entropy H, a new training texture is added to the model. With this updated set of textures the iterative optimization continues till all m training textures are included and there is no further improvement. Tests showed, that with a smaller number of optimization rounds, some improvements were missed but with 5000 iterations, the optimization performs well with the current set of textures.

Figure 5.8 shows a texture which depicts the entropy of a trained model for $s = 3$ target bins. Bright texels mean areas of high entropy, i.e. high variation. The dark regions denote high

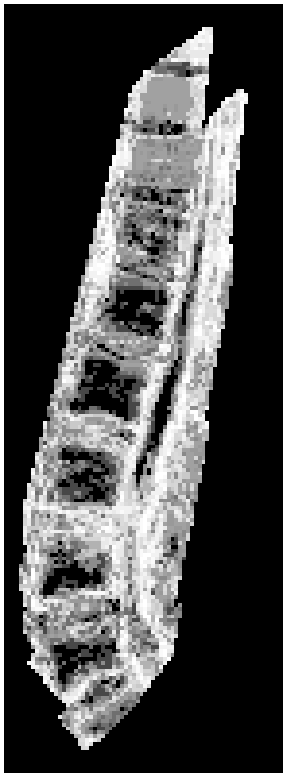


Figure 5.8: Image showing the entropy of the training texture: Bright texels mean a high entropy, dark texels denote low entropy for the corresponding training texel $t_j, j \leq N$. This reflects the phenomena of varying intensities across T1w and T2w for vertebrae and intervertebral disks, as reviewed in Section 2.3.3. Therefore the image represents a desired, good model.

predictability and hence low entropy. It describes directly the phenomena of varying intensities across T1w and T2w scans, which was reviewed earlier in Section 2.3.3. The high entropy in the disks in the lower lumbar disks, is due to the different appearance in T1w and T2w scans or age-related dehydration. Entropy decreases for disks in the border region, which can be explained by lower contrast in this area of the scans. The appearance of vertebrae does not vary as much as the disk's intensities, hence the entropy is lower for vertebrae. The model captures this variation and therefore the entropy image shows a desired, good model.

Model Storage The parameters used for the training of ETMs are saved to a XML-file. This includes the number of source bins r, number of target bins s, sampling distance sd and the landmark extraction method. Within the parameter evaluation of the ETMs, the models were trained on the original MR volume data and on preprocessed datasets, i.e. were the bias field was corrected. Therefore, a property indicating if preprocessing was performed or not is also saved in the file.

The lookup table with texel indices $\mathcal{T}$ is saved to a text file. For every anatomical label present in the model, the list of texel indices is saved.

For every texel $t_j, j \leq N$ of the learned model, the number of mappings to the target values $1 \ldots s$ are saved. This results in a table, where every row shows how many training textures map to $1 \ldots s$ target values.

The training volumes are normalized with the obtained texture transformations f_k and saved to the hard disk. They are needed for the subsequent training of the PBTs.

5.3.3 Training of PBTs

After building of an ETM, the datasets are normalized with the obtained texture transformations and PBTs are trained on the normalized data (see Section 4.2.3). The goal of this stage in the model building pipeline is to train classifiers, which are able to detect intervertebral disks in an unseen, normalized dataset. Hence they should be able to detect voxels in the volume, which belong to an intervertebral disk and these voxels should form a *disk-shaped cloud* for every detected disk.

Two detectors are learned with maximum tree depth of three, one for lumbar disks (Φ_L) and another one for thoracic disks (Φ_T). For both body regions, the respective detector consists of two boosted trees for two levels of resolution. The classifiers Φ_L^0 and Φ_T^0 classify the data at the original resolution level, whereas Φ_L^1 and Φ_T^1 operate on one time downsampled data. The trees are trained with Haar-like features and gradient features, whereby only Haar-like features were selected during the training phase for the detectors. Two variants for both regions were trained, with differences in the initial resampling of the training volumes. One detector was trained with an initial voxel size of $\Delta_{x,y,z} = 1.0$ mm and the other with $\Delta_{x,y,z} = 1.5$ mm. These variants are compared regarding detection accuracy and timing.

The obtained disk detectors Φ_L and Φ_T are saved in XML-format to their corresponding ETM $\mathcal{M}$.

5.4 Labeling of an Unseen Dataset

If an unseen MR scan has to be labeled, a user has to provide two inputs to the system:

- An initial click position (x, y, z) inside an intervertebral disk or vertebra
- The anatomical label for the clicked position, e.g. $L3/L4$, $L4$, etc.

With the user input, the previously learned ETM $\mathcal{M}$ and the trained feature detectors Φ_L and Φ_T, the labeling of an unseen MR scan follows three main processing stages:

1. As first step, bias field correction is done on the unseen dataset.
2. Based on the initial click position (x, y, z) and its corresponding label provided by the user, the learned entropy model $\mathcal{M}$ is matched to the dataset. Afterwards the dataset is normalized according to the obtained texture transformation. Section 5.4.1 gives a detailed overview about matching of an ETM.

3. On the normalized dataset, the intervertebral disks are localized. Starting from the initial position (x, y, z), iteratively one disk after the other is detected with the feature detectors Φ_L and Φ_T. Details on the semi-automatic labeling approach are provided in Section 5.4.2.

5.4.1 ETM Spine Model Matching

When matching a new volume, we match a new instance to the dataset. We aim to find the best mapping from source values r to target values s *for the unseen volume*, based on the learned model $\mathcal{M}$. These parameters are provided by the model $\mathcal{M}$. Figure 5.9 gives an overview of the normalization pipeline for a new, unseen MR scan. First the input dataset is prepared, i.e. downsampled, subject to the trained model (a). Then, new model instances are initialized (b) and optimized (d).

(a) Data Preparation First, the input volume I_k is prepared subject to the trained model $\mathcal{M}$. Rescaling may be performed, depending on the voxel size of the data, i.e. Δ_x and Δ_y, and the sampling distance sd of the model $\mathcal{M}$. The downsampling factor β_k is calculated (cf. Eq. 5.3.2) for the volume and the dataset is downsampled if necessary.

(b) Model Instance Initialization For a good spatial initialization of the model, Zambal et al. [67] initialize their model close to the correct object boundary. We propose to use the *extracted texel indices* $\mathcal{T}$ from the training phase. By looking up the input label from the user, i.e. $\mathcal{T}(\lambda_i)$ for a disk label or $\mathcal{T}(\kappa_j)$ for a vertebra, we retrieve a list with m texel indices $j, j \leq N$. These are possible locations in the texture for the given anatomical structure.

We propose to use these indices to initialize *multiple model instances* in order to increase the robustness of the model matching. We initialize the instances at slightly different positions and optimize them *in parallel*, i.e. every instance in an own thread. For every texel index in the list, we initialize a new instance by performing the following steps (see also Figure 5.10):

- We create a new instance and set it to the origin of the volume $(0, 0, 0)$. The instance is created around its *centroid*, which lies then in the origin of the volume.
- The world coordinates (x', y', z') for the texel $j, j \leq N$ in the current volume are calculated. Note that they can also lie outside of the volume.
- Between the input position form the user (x, y, z) and the position of the label in the model instance, i.e. the texel position, the distance is calculated. This distance indicates the translation which is necessary to move the model instance to the correct position.
- Based on the calculated translation, the model instance is moved to its start position. After this process, the instance is initialized in the region of interest, i.e. around the lumbar spine.

We do this for all m model instances and propose to optimize them in parallel, which is described in the next step.

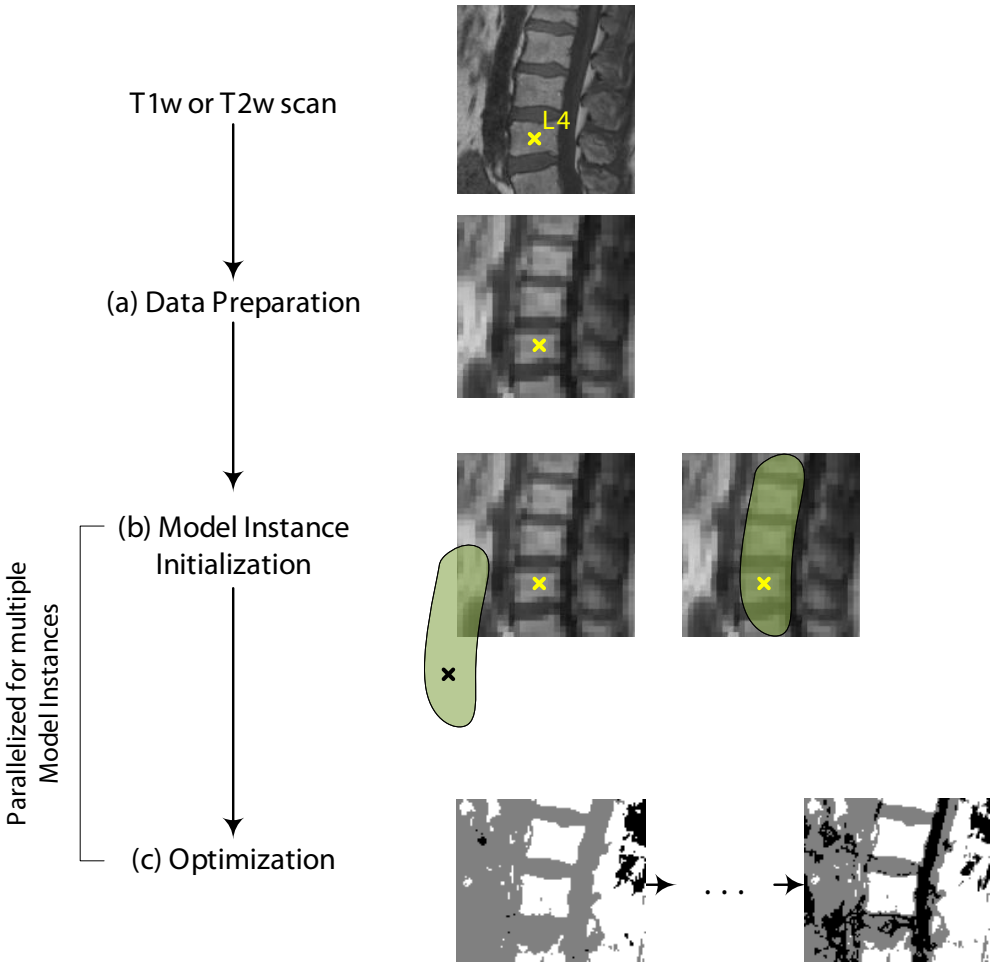


Figure 5.9: ETM model matching pipeline. The unseen scan is prepared, i.e. downsampled, according to the learned model first (a). A new model instance is initialized at the region of interest, based on the user input (b) and optimized (c). In order to enhance robustness, multiple instances are created at slightly different positions and the matching of those is performed in parallel. Image based on data by courtesy of AGFA HealthCare [25].

(c) Optimization If an ETM model instance is matched, only shape parameters (scale, position and statistical shape parameters) are optimized [67]. This makes exhaustive search for the best model instance feasible and is applied in this phase, as proposed by the Zambal et al. [67]. Since the optimization is performed in parallel with multiple instances, the following holds for every initialized model instance.

As introduced in Section 4.1.2.3, the texture transform of the new model instance is created as maximum likelihood of the given underlying texture and the texture model. In every iteration

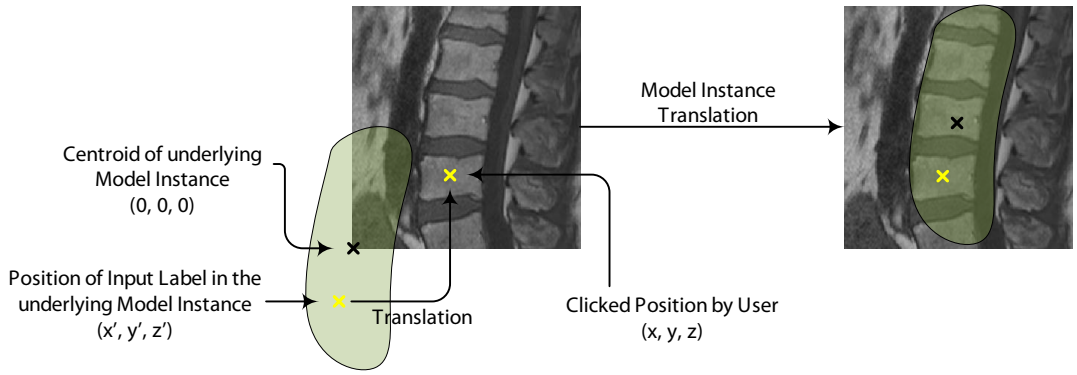


Figure 5.10: Spatial initialization of a new model instance. A new model instance is created at the centroid of the instance and moved to the origin of the volume $(0, 0, 0)$. For a texel j with an anatomical label, we calculate its position (x', y', z') in the initialized model instance, i.e. in the underlying volume. Based on the input position from the user (x, y, z), the distance to (x', y', z') is calculated. This distance indicates the translation, which is necessary, to move the model instance to its start position.

round, the parameters (scale, position and statistical shape parameters) of the current model instance are changed randomly within their valid range. Then the model instance is evaluated by calculating the likelihood of its overlapped texture U': For every texel t_j the underlying source value is extracted and transformed into a target value g'_i. The likelihood given g'_i at texel t_j is retrieved and summed up. If the texture likelihood improved, the current configuration is kept as basis for the next iteration. Otherwise the current parameter changes are discarded and the next iteration starts with the same model instance. Figure 5.11 shows normalized data after several iterations, where the current mapping f_u from r source values to s was applied on the dataset.

A maximum number of 100 optimization rounds is performed, in order to increase robustness and reduce the risk to get stuck in a local minimum. For all testing datasets a solution was found within the first 60 iterations at the most. In every optimization round, the current model instance is optimized with simulated annealing, with a maximum number of 500 iterations. Iteratively model shape parameters are changed randomly within their valid ranges and the current model instance is evaluated, i.e. the likelihood of texture U, which is currently overlapped by the model instance. If no better mapping is found within the 500 iterations, model matching stops. If a better solution is found, this mapping is selected as new best mapping and used as start configuration in the next optimization round. The heuristics are based on work from Zambal et al. [67], [20].

From all parallel optimized texture transformations f_k, the best one is applied to the MR scan. The parallel matching further increases the robustness of the matching. The corresponding model instance should now be matched to the spinal column. If we derive the landmark positions of the matched shaped model, i.e. the disk and vertebra body centers, we already obtain possible intervertebral disk center positions. However, with these positions we obtain a disk localization

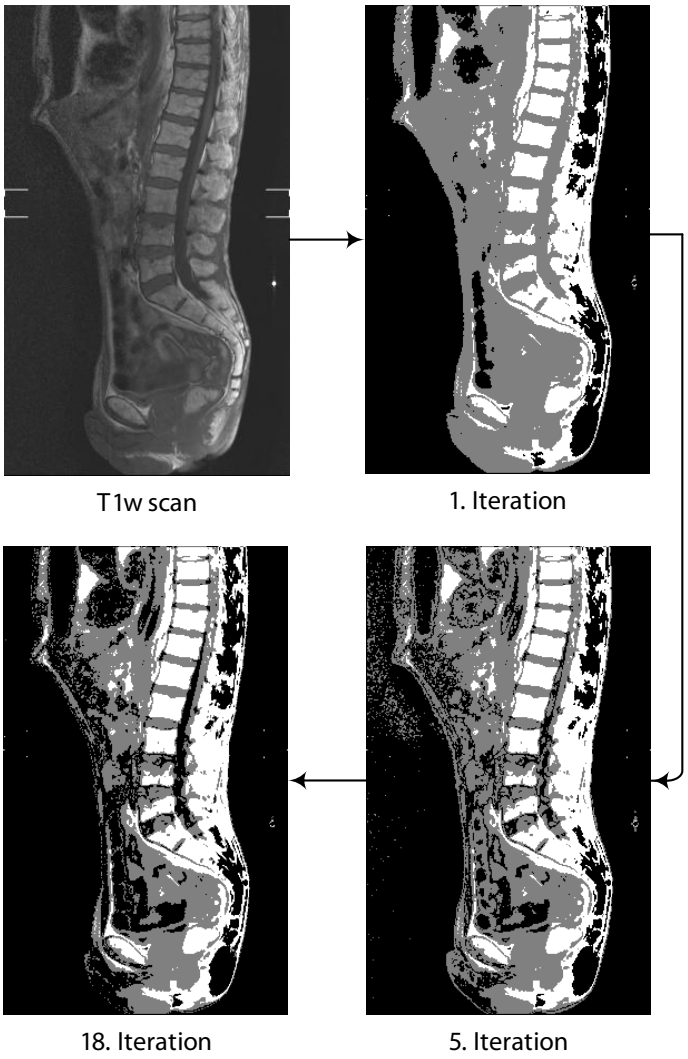


Figure 5.11: Normalized T1w data after the first, fifth and eighteenth iteration. Data by courtesy of AGFA HealthCare [25].

precision of 89.8% at a recall of 85.8% (see Section 6.5). To increase precision and recall, we suggest to normalize the volume I_k with the obtained texture transformation f_k and apply PBTs in order to find intervertebral disk centers.

5.4.2 Iterated Feature Extraction and Labeling

As last step in the labeling pipeline, the disks are localized in the entropy-optimized volumes with the feature detectors Φ_L and Φ_T obtained in the model training phase. Instead of searching

for disks in the whole dataset, we propose to use the user input in order to define a region of interest and search for the disks in an *iterative* manner. This reduces the time for disk detection, because we restrict the search space in the very beginning.

At the provided input position (x, y, z), we define a search region for the first intervertebral disk by a bounding box (see Figure 5.12). The dimension of the bounding box is obtained from the derived statistics from the annotated data, as described in Section 5.3.1. It is spanned by the mean anterior-posterior (x-direction), top-bottom (y-direction) and left-right (z-direction) distances (in mm) of the disk.

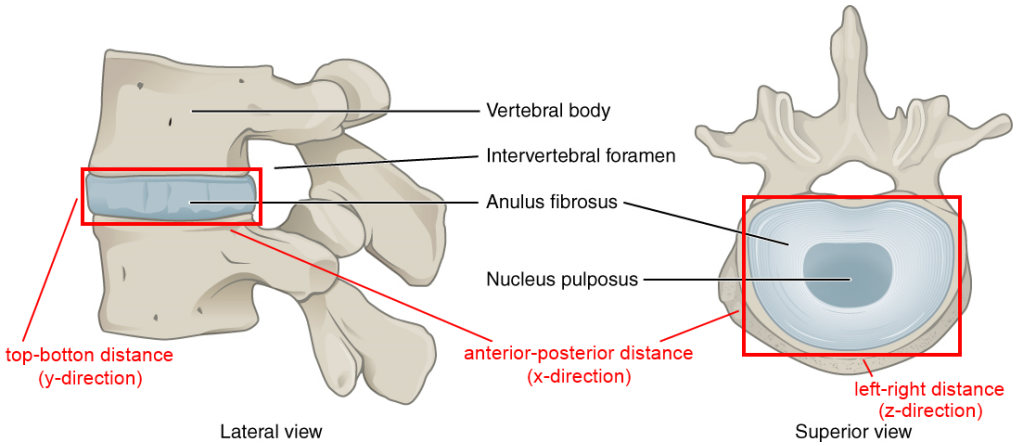


Figure 5.12: Visual illustration of the dimension of the bounding box (left-right, top-bottom and anterior-posterior distances) for intervertebral disk detection. Image adapted from OpenStax [10].

Within this bounding box, disk candidate voxels are extracted with the detector for the current body region, i.e. lumbar or thoracic. The trained feature detectors are executed from coarse to fine. For a lumbar disk that means for example, that we start at level $w = 1$ with Φ_L^1 and continue with Φ_L^0 at the original resolution, i.e. $w = 0$.

The detected voxels form a disk-shaped cloud (see 2D projection in Figure 5.13), from which the best center-candidate is then determined. Schulze et al. [53] propose to assign probabilities to every detected voxel and select the voxel with the highest probability as disk center. It is determined in the following way: Around every detected voxel we span another box, which roughly approximates the shape of an intervertebral disk. The dimension of this box is again based on the derived mean disk sizes. For every detected voxel, we count how many other detected disk voxel lie currently within the box and assign the ratio of

$$\frac{\#\text{detected disk voxel within box}}{\#\text{voxel of box}}$$

as its probability. Hence, voxels which lie close to the center of the detected point cloud have a high probability. The voxel with the highest probability is selected as disk center candidate.

Then the bounding box is propagated along the spinal canal to the next potential location of a disk. The distance is again based on statistics from the annotations, i.e. we translate the

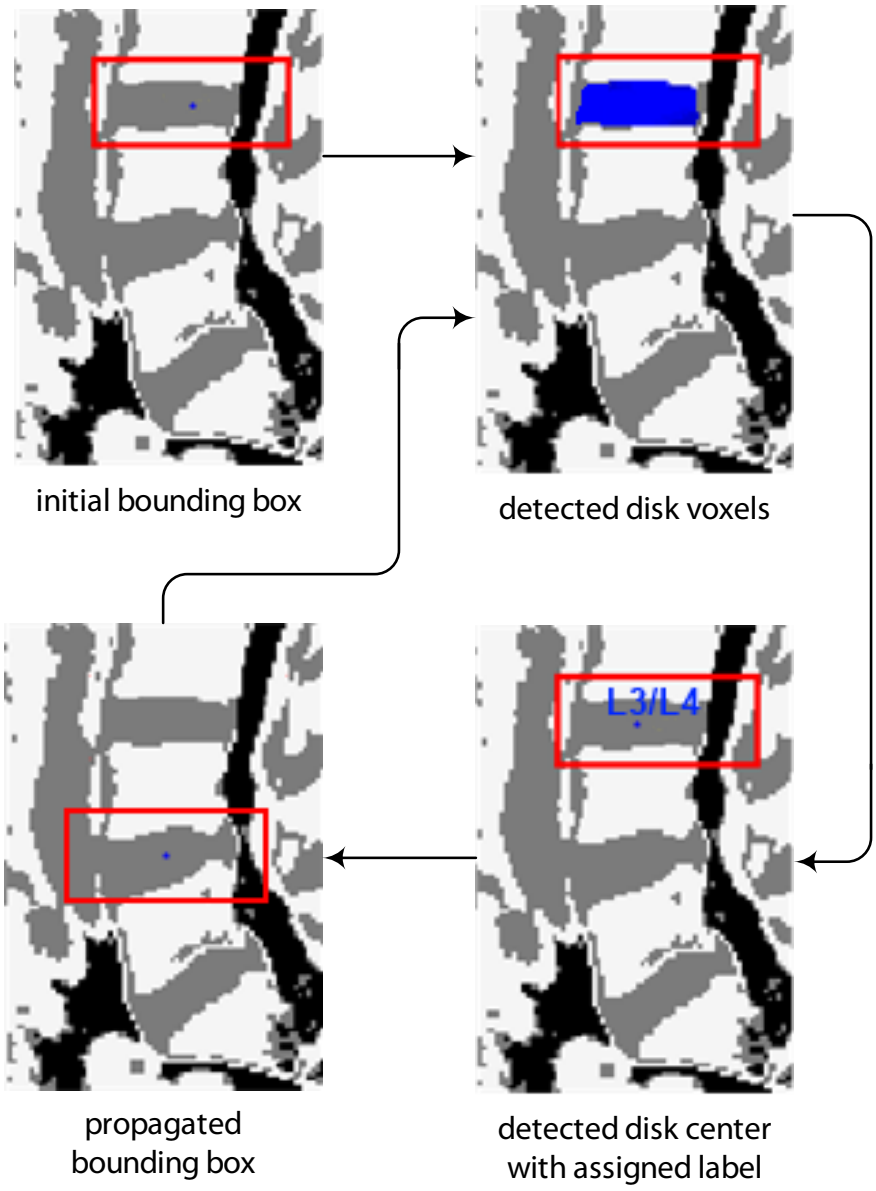


Figure 5.13: Iterated feature detection pipeline illustrated on normalized MR data. The disk candidate voxels are detected with the corresponding detector, i.e. Φ_L or Φ_T within a region of interest around the position (x, y, z), which is provided by the user. The best disk center candidate is selected and labeled. The bounding box is propagated to the next potential disk location, whereby the distance to this position is based on the derived statistics. At the new position, the search for the next disk center continues.

bounding box according to the mean height of lumbar or thoracic vertebrae. The iterations continue till no further disks are detected or the border of the dataset is reached. The assignment of anatomical disk and vertebrae labels is based on a standard spine dictionary (see Section 2.1).

5.5 Implementation Details

The training and semi-automatic spine labeling pipelines were integrated into an existing framework, which is written in Java. For the building of the spine model, a simple GUI (Graphical User Interface) was designed (see Figure, 5.14). The annotation file can be specified, as well as the necessary parameters for the ETM learning. For the training of PBTs, the configuration XML-files for the trees to learn have to be specified. They include the parameters, e.g. the initial resampling voxel size $\Delta_{x,y,z}$ or the number of resolution levels w. Furthermore, the features, which should be extracted, have to be specified. In our work, those are image gradients and Haar-like features.

For the semi-automatic spine labeling of an unseen MR scan, a plugin to the existing software was implemented in Java (see Figure 5.15). The anatomical label for the start disk or vertebra has to be selected from the drop-down list boxes (left). The labeling starts, if the user hits the start-button and clicks on the desired start position (x, y, z) in the volume in the imaging area (right).

For the Delaunay Tetrahedralization, the CGAL library [1] was compiled for 64 bit systems. The feature detection framework with PBTs is provided by VRVis [20], as well as part of the ETM implementation.

Bias field preprocessing was implemented in MATLAB [40]. The data was preprocessed in advance for the evaluation within this thesis. MATLAB was further used to generate plots and graphs. For the integrated flowcharts and other diagrams, Microsoft Visio [14] served well.

The labeling results were evaluated with a python-based tool, provided also by VRVis.

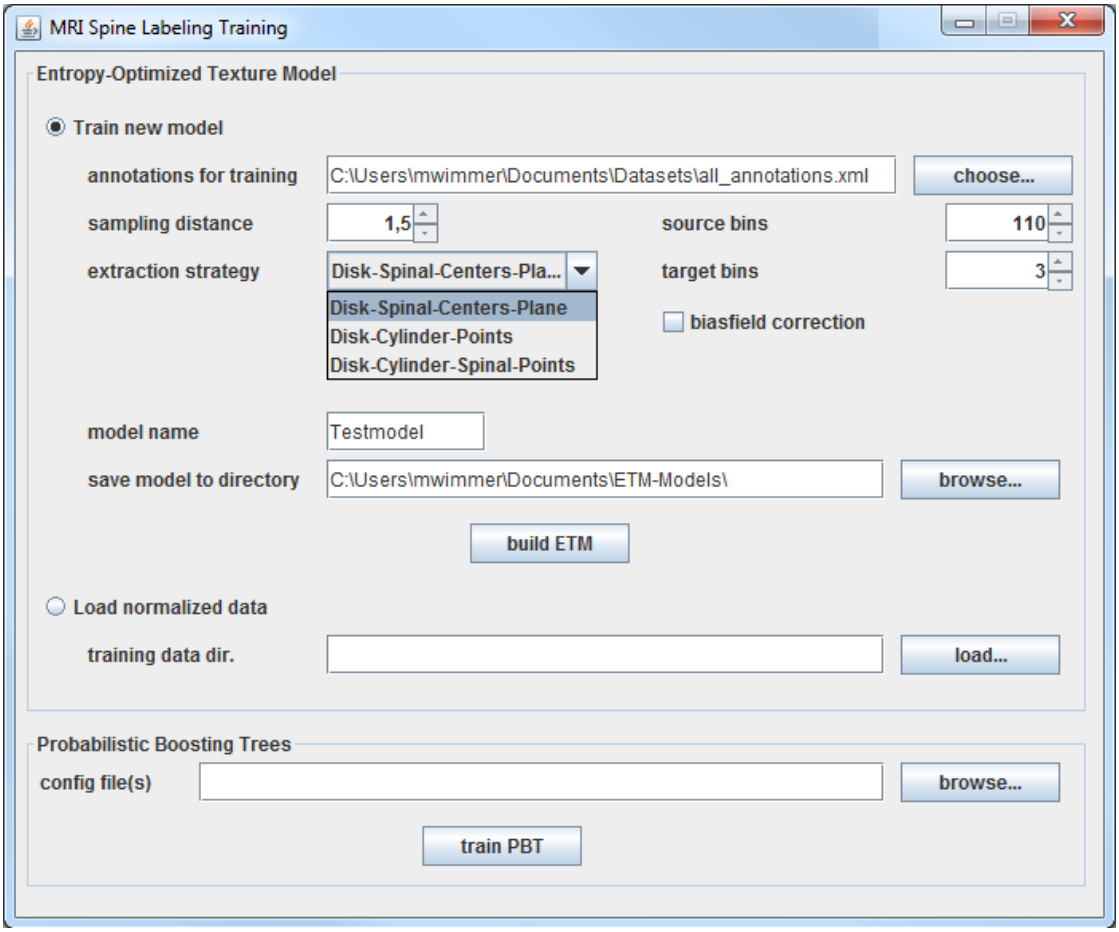


Figure 5.14: Screenshot of the GUI for the building of the spine labeling model. The parameters for the training of an ETM can be specified, e.g. source bins r, target bins s, etc. For the training of PBTs, a configuration file in XML-format can be loaded, which specifies the properties of the tree. Based on this, the classifier is learned.

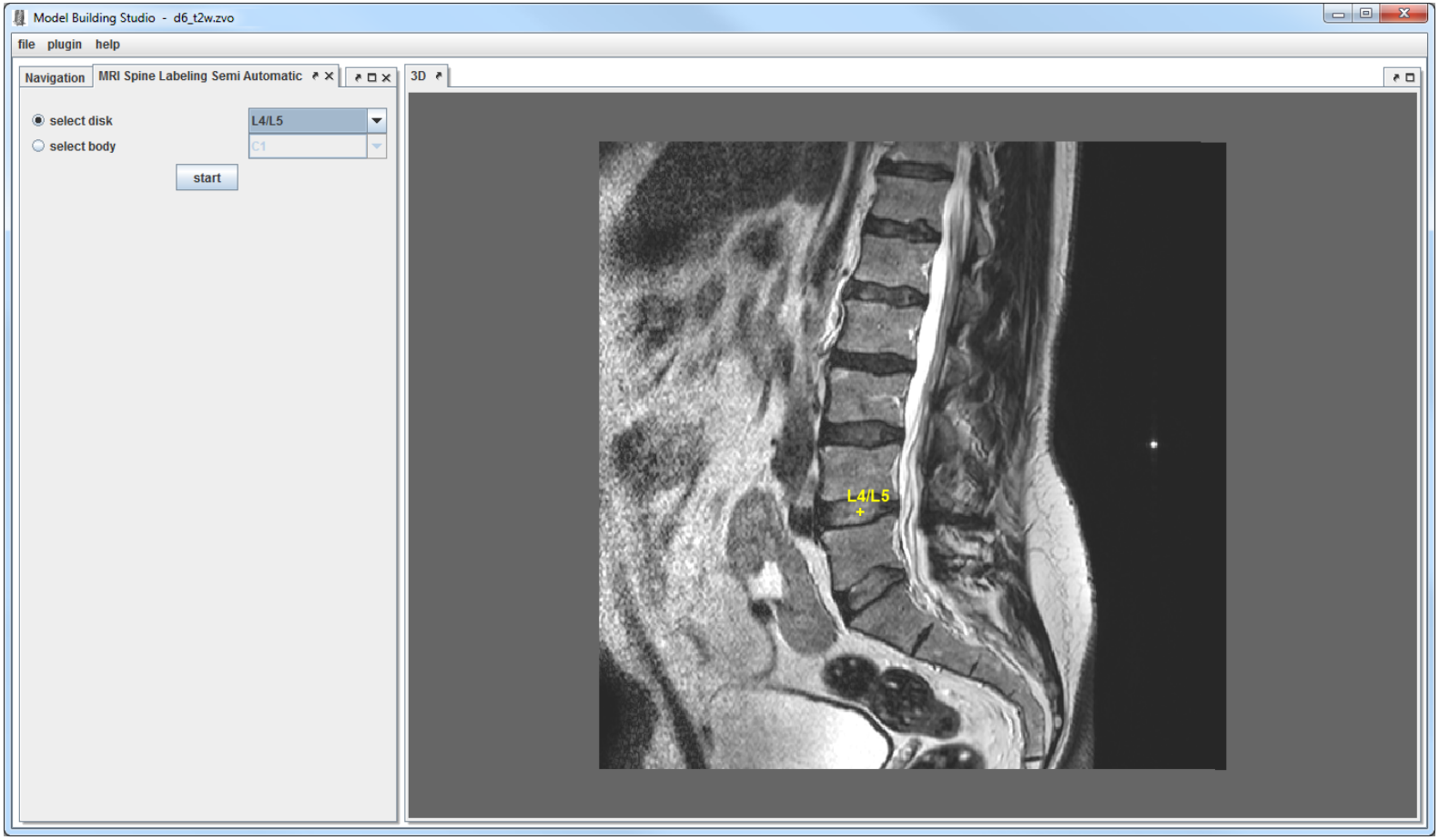


Figure 5.15: Screenshot showing the GUI for the semi-automatic spine labeling.

CHAPTER 6

Evaluation and Results

The following chapter reports the quality of data normalization with ETMs and the accuracy of disk center point detection. First, Section 6.1 gives an overview on the training and testing data used. Then, measures for evaluation of entropy optimization and disk detection are introduced in Section 6.2. Results based on these measures as well as timings are reviewed afterwards. Results from bias field preprocessing are shown in Section 6.3. Section 6.4 reports and discusses the results achieved for data normalization with ETMs. Section 6.5 shows the accuracy for the disk center detection. Notes on the training and testing system as well as the performance are given in Section 6.6. Section 6.7 concludes this chapter with a short summary of the achieved results.

6.1 Datasets

Datasets from 14 different patients are provided by courtesy of AGFA Healthcare [25], containing T1w and T2w SE sequences and/or T1w FFE 3D (FFE3D) sequences. From this set, a total number of 28 volumes covering the lumbar and lower thoracic region of the spinal column were extracted. Scans from three out of the 14 patients are originally full spine scans, but only cut outs from the desired body part are used for evaluation, i.e. the volumes d1_t1w, d1_t2w, d2_t1w, d2_t2w, d24_t1w and d24_t2w. According to the DICOM tags [28], the datasets were acquired on different Philips scanners, namely Achieva 1.5 T, Panorama 0.23 and Intera 1.5 T. They show women in the age of 37 - 71 years and men from 23 - 71 years.

The voxel sizes in the datasets are highly anisotropic: The sagittal in-plane voxel size ranges from 0.59 mm to 1.19 mm (i.e. $\Delta_{x,y}$), whereby the slice distance lies in the range of 4 mm to 6 mm (i.e. Δ_z). The in-plane resolution for the FFE3D scans is 288 × 384 voxels and for the SE scans 320 × 320, 324 × 324 or 512 × 512 voxels. The cut-outs from the full body scans are between 512 × 844 and 512 × 908 voxels in size. The number of slices depends on the acquisition method. For the T1w and T2w SE sequences it varies between 11 and 16 slices, whereby most of the scans have 12 or 13 slices. The FFE3D scans are acquired in 3D and have

16 or 18 slices. The data also shows a high variability regarding intensity ranges. As mentioned in Section 2.2, MR data does not have a standardized scale as CT. Most of the datasets are in the range $[0, 255]$, but there are also volumes which have a maximum intensity between 1270 and 4000. A detailed overview on the field of view in the volumes and properties of the data is provided in the Appendix (see Table A.1 and A.2).

6.2 Measures

In order to evaluate the quality of entropy mappings and furthermore the detected disk center points, proper measures need to be defined. For the entropy models, comparing the trained texture models by their entropies H^{tex} and H^{model} is not sufficient in order to evaluate the quality of the intensity mapping. A trained ETM should yield a rather homogeneous intensity mapping for each tissue of interest in order to provide a good basis for the following feature detection. A method for comparing ETM mappings is introduced in Section 6.2.1. For evaluation of the disk centers, the position of the detected disk center candidates is compared to the expert-annotated disk cylinders. The quality is described by the well-known measures *precision* and *recall* [39], [57], which are reviewed in Section 6.2.2.

6.2.1 Intensity Homogeneity in ETMs

For evaluation of the entropy optimization, we look at the normalized training data $I'_k \in \mathcal{S}_{tr}$ obtained with the trained models $\mathcal{M}$. The quality of the mapping provided by a model $\mathcal{M}_i$ is measured based on the achieved *intensity homogeneity* within disk and vertebrae tissue in the training data.

From the mapped volumes $I'_k \in \mathcal{S}_{tr}$, we derive histograms for disks d_i^k (see Figure 6.1) and vertebrae v_i^k (see Figure 6.2) for every model $\mathcal{M}_i$ based on the annotated ground truth. From the annotated disk cylinders z_i^k, the percentage of every target value $g'_j \in \{1 \dots s\}$ from model $\mathcal{M}_i$ is calculated on a voxel level. The volumes are re-sampled to a voxel size of $\Delta_{x,y,z} = 1.0$ mm. The voxel inside a cylinder z_i^k in volume I_k (with anatomical label λ_i) having the same intensity g'_j are counted and the ratio to all voxel in z_i^k is calculated – formally:

$$h_i^k\left(g'_j\right) = \frac{\#\left\{\,(x,y,z)\,|\,(x,y,z) \in z_i^k \wedge I_k\,(x,y,z) = g'_j\right\}}{\#\left\{\,(x,y,z)\,|\,(x,y,z) \in z_i^k\right\}} \tag{6.1}$$

Calculating this percentage value for all target values $g'_j \in \{1 \dots s\}$ in a disk d_i^k results in its *relative intensity histogram* h_i^k (analogous for vertebrae).

Looking at the peak of such a tissue histogram, we can derive its *mode* immediately. It is defined as the most frequently occurring bin/value in a histogram. The mode bin should exhibit a very high peak (i.e. close to 1), because then the intensity mapping is very homogeneous within the tissue. However, a high homogeneity of a disk or vertebra in a model is not a sufficient criteria for a good normalization. Each tissue should map to a specific target value/range and to another value/range as its adjacent disks or vertebrae. The histograms of neighboring tissues have to be put in relation to each other. Figure 6.1 and Figure 6.2 show histograms for disk $L4/L5$ and vertebra $L4$ for every training volume $I_k \in \mathcal{S}_{tr}, k = 1 \dots 11$.

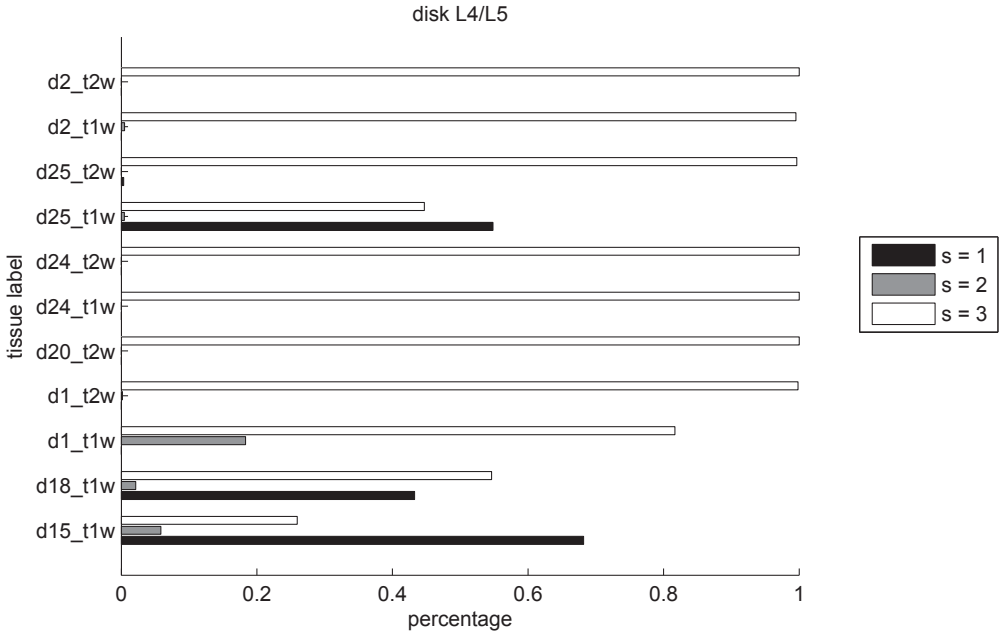


Figure 6.1: Relative intensity histograms for disk $L4/L5$ across the normalized training volumes. Three out of eleven volumes do not exhibit a single peak for $L4/L5$, whereby two of them map to the same target value $s = 3$ as the adjacent vertebra $L4$ (see Figure 6.2).

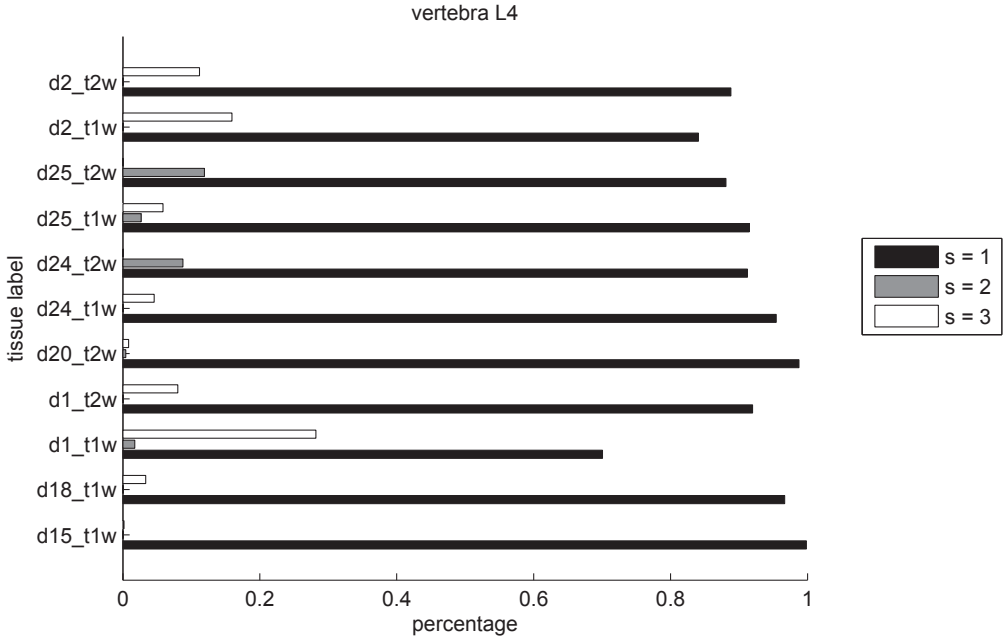


Figure 6.2: Relative intensity histograms for vertebra $L4$ across the normalized training volumes. All histograms are rather homogeneous and exhibit a peak at $s = 1$.

We calculate relative histograms for all disks and vertebrae across the mapped training volumes from a trained model $\mathcal{M}_i$ (see Figure 6.3 as sample for one training volume). Having a set of relative intensity histograms for disks and vertebrae, the next step is to calculate the *mode* for disks respectively vertebrae of $\mathcal{M}_i$. Across all disks in model $\mathcal{M}_i$ we determine the maximum occurring target value g_i' and call it the mode for disks $mode_d(\mathcal{M}_i)$ in model $\mathcal{M}_i$ (analogues for vertebrae $mode_v(\mathcal{M}_i)$). Based on the modes, we calculate the *Hamming distance* [24] for all disks and vertebrae in the mapped training data of model $\mathcal{M}_i$. The Hamming distance is a measure in information theory, that describes in how many positions two strings differ.

For this work, we consider the Hamming distance for disks and vertebrae of a model $\mathcal{M}_i$. For every disk in the training set, we calculate its mode and compare it with the disk mode $mode_d(\mathcal{M}_i)$ of the model, respectively $mode_v(\mathcal{M}_i)$ for vertebrae. Mismatches increase the respective Hamming distance.

By considering both relative Hamming distances (normalized to the range $[0, 1]$), we judge the mapping quality of a trained entropy model $\mathcal{M}_i$. Hence we prefer models with low Hamming distances.

6.2.2 Spatial Precision and Recall

In pattern recognition and computer vision, the measures *precision* and *recall* are often used to evaluate the quality of classification and recognition algorithms [57]. They were originally introduced within the scope of information retrieval [39], in order to evaluate retrieval results. Generally, the task in information retrieval is to retrieve relevant information from a set of objects (e.g. words, images, documents, etc.), based on a query. The precision describes then, how many relevant documents are retrieved, i.e. the ratio of relevant retrieved documents to all retrieved documents. Recall denotes the fracture of retrieved relevant documents to all relevant documents, i.e. how many of all relevant documents were retrieved by the algorithm.

Within the scope of intervertebral disk localization, Major et al. [38] describe precision and recall in a spatial context. They aim for measuring the accuracy of detected disk landmarks, i.e. if the detected disk center point lies within the expert-annotated disk cylinder or not. For formulating precision and recall, criteria for *true positive (TP)*, *false positive (FP)* and *false negative (FN)* detections have to be derived.

A detected disk landmark is a TP candidate, if it lies within the expert-annotated cylinder of the corresponding intervertebral disk. FP landmarks are detected by the algorithm, but are not present in the ground truth. FN labels lie either outside the cylinders or are not detected at all.

With these terms, precision and recall are defined as:

$$precision = \frac{|TP|}{|TP| + |FP|} \qquad recall = \frac{|TP|}{|TP| + |FN|} \tag{6.2}$$

The precision describes then the ratio of correct retrieved disks to all retrieved disks, whereby recall denotes the ratio of correct retrieved disks to all ground truth disks.

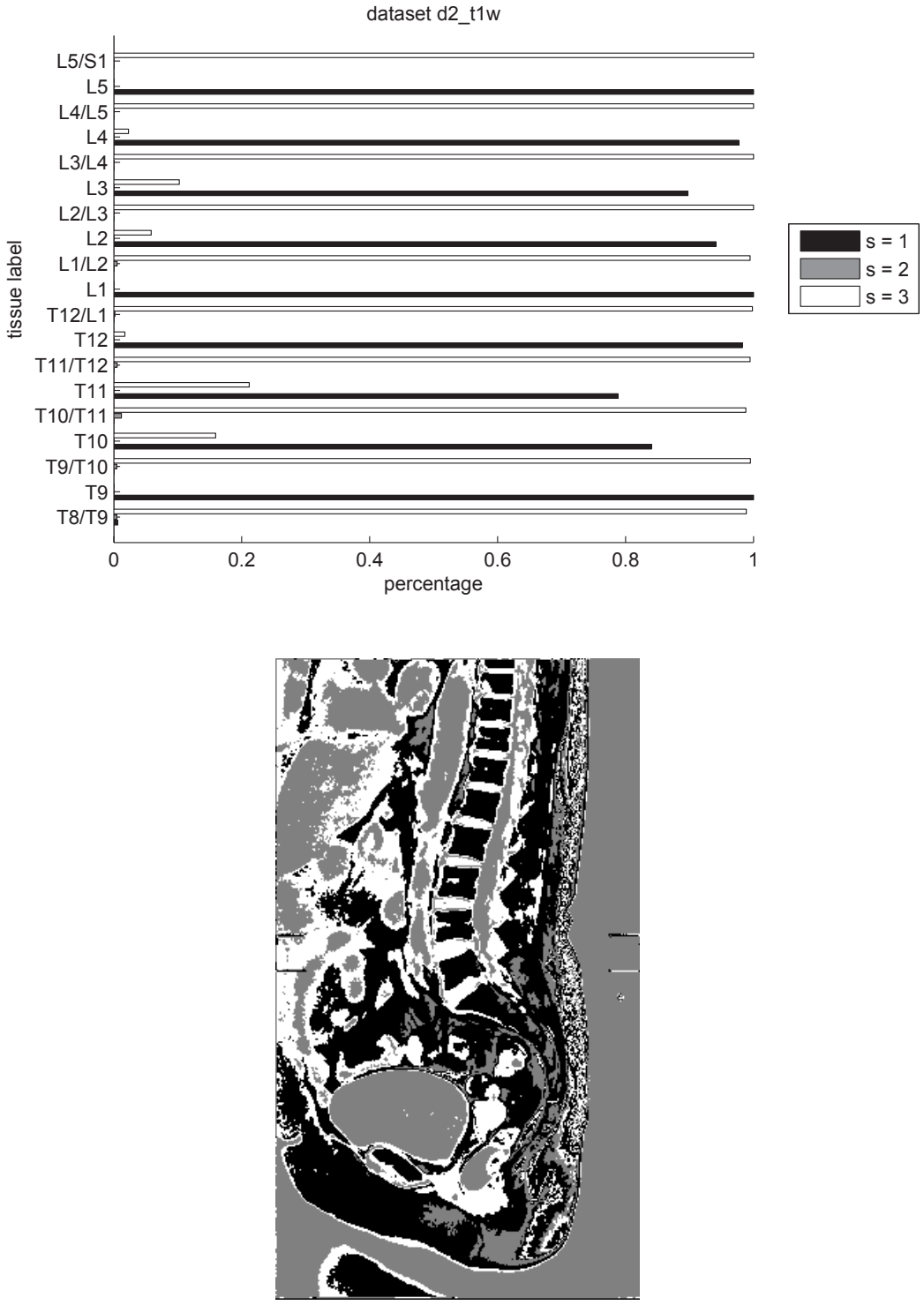


Figure 6.3: Relative tissue intensity histograms (top) for volume d2_t1w and the obtained normalized volume (bottom). The dataset was normalized to $s = 3$ target gray values with model $\mathcal{M}_3$ (see Table 6.2 for parameter details). The histograms show, that the ETM delivers the desired homogeneous mapping for disks respectively vertebrae in the training volume.

6.3 Bias Field Correction

Two different bias field correction methods, based on the work from Juntu et al. [31] were tested: For the first method, the bias field was estimated based solely on the mid-slice of the volume. Every slice of the volume was divided by this bias field. The second method approximates the bias field for every slice and hence every slice is divided by the bias field of itself. Regarding timing, it is slower than the first method, because the surface fitting is done for every slice (see performance evaluation in Section 6.6.3). Both methods reduce the bias field (see Figure 6.4). Looking at the intensity ranges for the disks $T9/T10$, $T10/T11$ and $T11/T12$, we see that they are now closer to the other disks in the volume. For the mid-slice method they are closer together than for the method with every slice. The latter shifts the disk intensity ranges and also enlarges the range, as seen in the plot. The mid-slice method is also faster then the other approach, since only a single surface fitting step is needed. Hence this approach is integrated as preprocessing method.

6.4 ETM Parameter Optimization

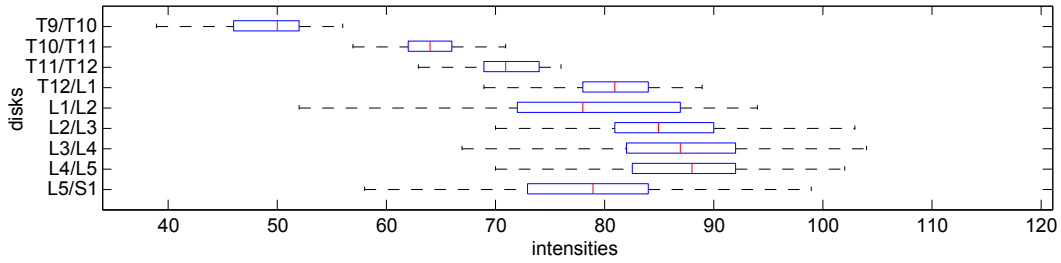
Eleven volumes $I_k \in \mathcal{S}_{tr}, k = 1 \dots 11$ were selected for training of the texture models. The volumes were chosen in a way that they show not only the lumbar spine but also the lower part of the thoracic spine. This ensures, that the learned texture model covers the lower part of the spinal column well. The training set contains a mixture of T1w (six volumes) and T2w (five volumes) scans. In this way, the intensity variability from both weightings should be covered by the model.

On the training data, parameter space exploration was performed for several variables of the texture model (see Table 6.1).

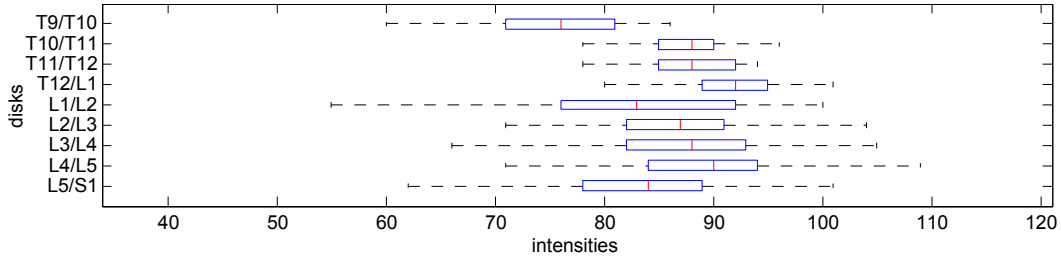
Parameter	**Values or Range**	**# Values**
source values	$r_k \in [70, 75, \dots 130]$	13
target values	$s \in [2, 3, 4, 5]$	4
sampling distance	$sd \in [1.5, 2.0, 2.5]$	3
extraction strategy	Disk-Spinal-Centers-Plane Disk-Cylinder-Points Disk-Cylinder-Spinal-Points	3
bias field correction	on/off	2
		936 configurations

Table 6.1: Parameter ranges used for the evaluation of ETMs.

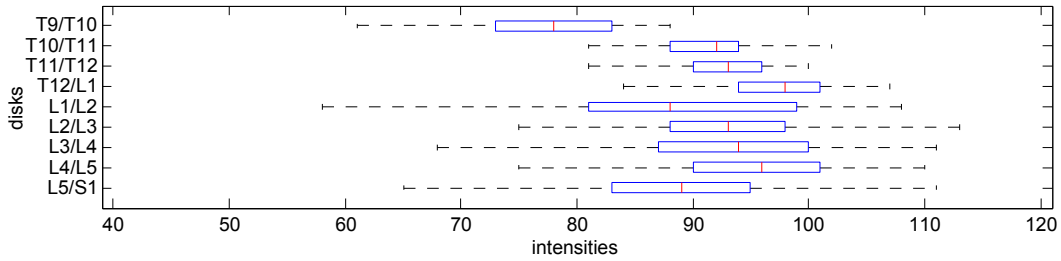
Even though the valid ranges for the model parameters have been restricted to a reasonable subset, a model set $\mathcal{M}$ with a total number of $|\mathcal{M}| = 936$ *model configurations* is obtained. Only one training set $\mathcal{S}_{tr}$ was extracted out of all 28 volumes and no cross-validation was performed. On the one hand, it is very time consuming and computationally expensive to train multiple sets, where for every subset $\mathcal{S}_{tr} \in \mathcal{S}$ 936 parameter configurations have to be trained. On the other hand, not all datasets cover the same anatomical region, e.g. only the disks $T11/T12$ to $L5/S1$.



(a) Original volume.



(b) Every slice of the volume corrected with the bias field of the mid-slice.



(c) Every slice of the volume corrected with the bias field of itself

Figure 6.4: Boxplots showing the disk intensity ranges for the original volume d17_t1w (see Figure 6.4a) and after applying the two proposed variants of the bias field correction method. Figure 6.4b denotes the corrected intensities, if the volume is divided only by the estimated bias field of the mid-slice of the volume. Figure 6.4c shows the corrected volume, where the bias field is estimated for every slice.

Hence models would be obtained, that cover a smaller region of interest. The performance of the normalization might suffer for volumes showing vertebrae and disks not covered with the trained model. However, this hypothesis needs further investigation and can be considered as future work.

For visual exploration of the ETM training results, the HTML5 PivotViewer [23] (see Figure 6.5) was used. This tool allows to browse through a collection of images, whereby every image is associated with meta data. Within this work, the meta data describes the trained ETMs, i.e. one row corresponds to one trained model. The data table includes on the one hand the training parameters of the models, such as number of target bins s, sampling distance sd, texture extraction strategy, etc. On the other hand, model properties and data statistics, such as disk and vertebrae mode $mode_d(\mathcal{M}_i)$ and $mode_v(\mathcal{M}_i)$ of the model $\mathcal{M}_i$ are included. For every model, one image is generated, which represents this model. It is a collection of normalized mid-slice images, which are obtained by normalizing the eleven training volumes with the trained model $\mathcal{M}_i$.

For a qualitative evaluation of the trained entropy models, we look at the modes and Hamming distances (see Section 6.2.1). Relative Hamming distances for disks and vertebrae for all models $\mathcal{M}_i, i = 1 \ldots 936$ are plotted against each other (see Figure 6.6). It is seen immediately, that the distances are lowest for $s = 2$ and $s = 3$ target values. Models that map to $s = 4$ or $s = 5$ target bins often result in noisy images (see Figure 6.7, right), although for high-resolution training images they can keep more details because the level of abstraction is lower as for example with $s = 2$ or $s = 3$ bin models (see Figure 6.8). However, we cannot rely on unseen testing images which exhibit a high resolution.

Investigating models with Hamming distances < 0.2 in more depth (see Figure 6.9), we see that the models with the lowest Hamming distances map to $s = 2$ target values at sampling distance $sd = 2.5$. Also – what is not shown in this plot – more than half of the models in this area use bias field correction (see Figure 6.10). The models mapping to $s = 3$ target bins use mainly the *Disk-Spinal-Centers-Plane* landmark extraction method. It seems to generalize better for the three-bin model than the other extraction methods. No such trend can be detected within two-bin models.

Several models are selected with distances < 0.2 in order to compare labeling results on models with different parameter settings. The evaluation is provided in the next section.

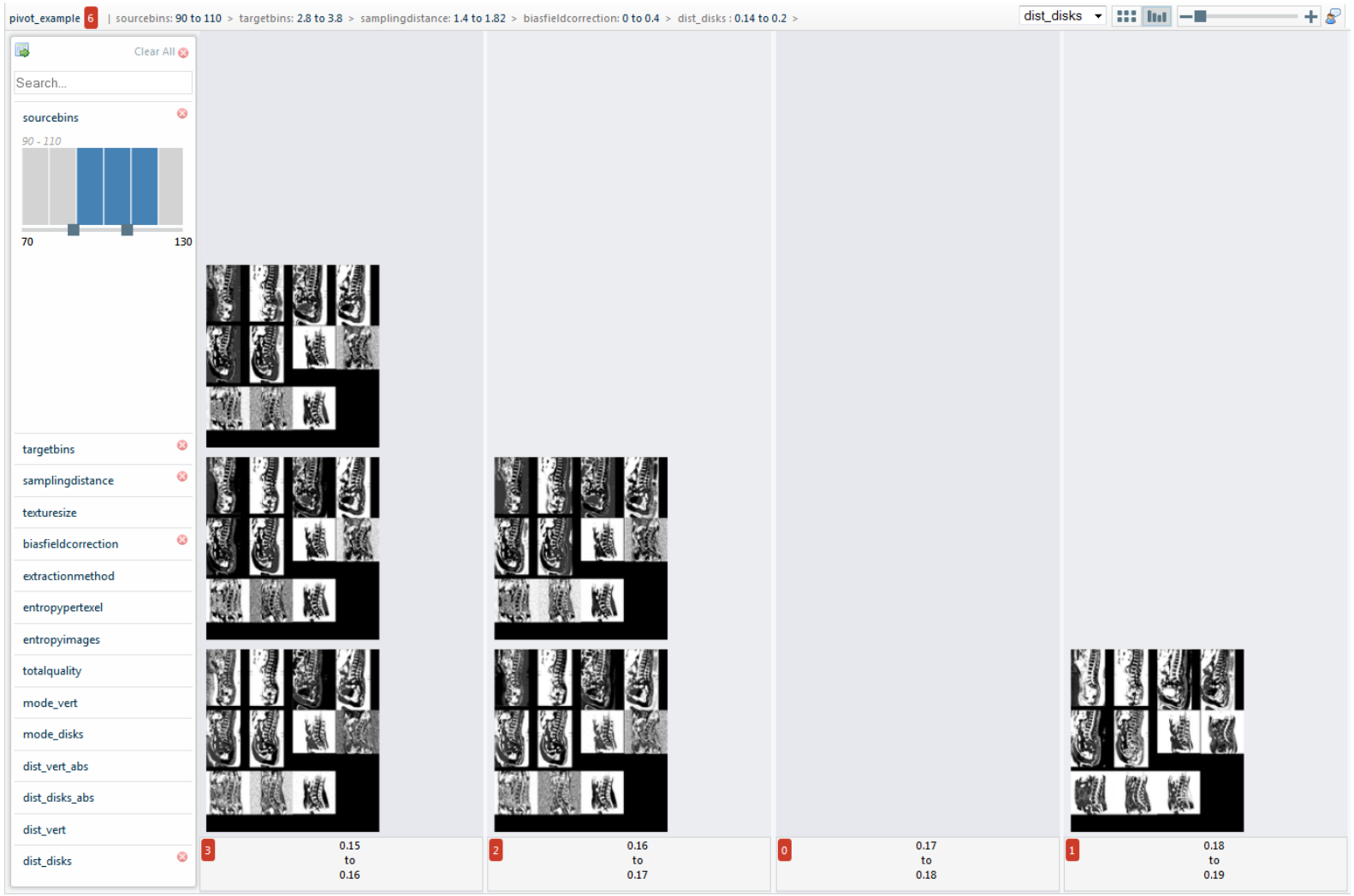


Figure 6.5: Screenshot showing the HTML5 PivotViewer Tool [23]. On the left, the properties of the trained models $\mathcal{M}_i$ are listed (e.g. source bins, target bins, relative hamming distances, etc.). By selecting value ranges for properties of interest, one can browse through the images related to the models. Note that one image belongs to one specific model $\mathcal{M}_i$. It is a collection of normalized mid-slice images, which are obtained by normalizing the eleven training volumes with the trained model $\mathcal{M}_i$. The images corresponding to the selected properties are shown in the image area to the right. The images can be explored in more detail with respect to one selected property. In this example, the images are grouped by their disk relative Hamming distances (see on the bottom).

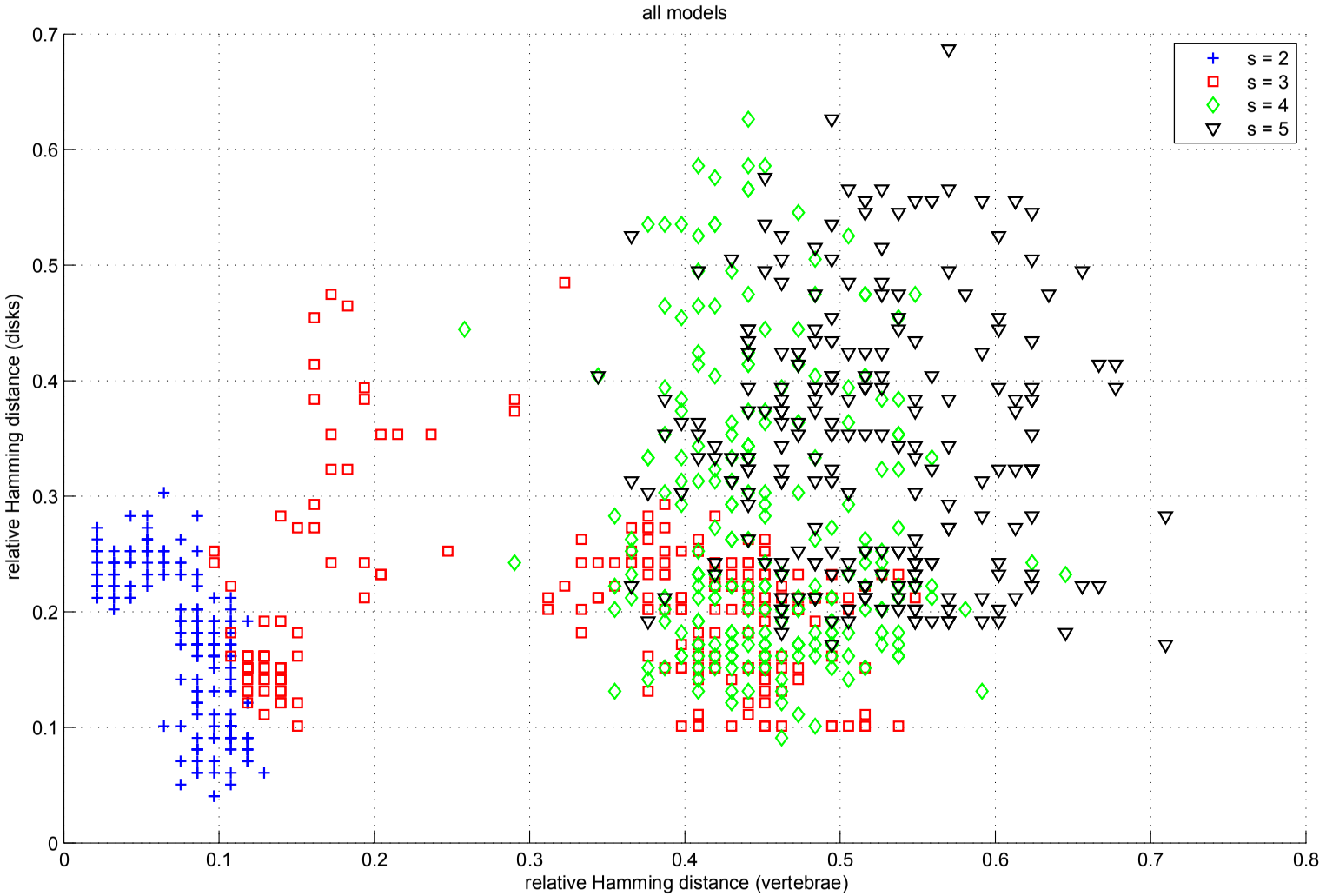
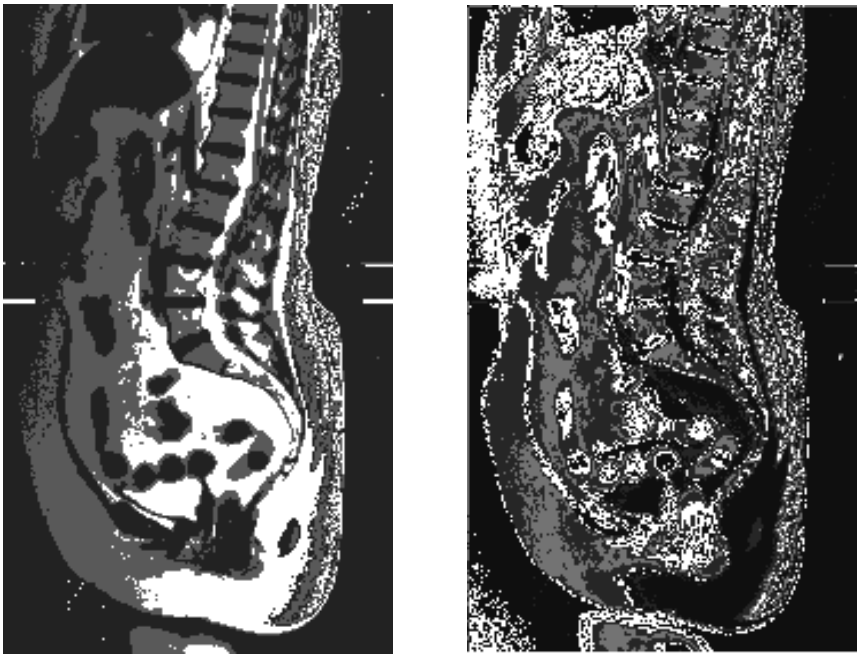
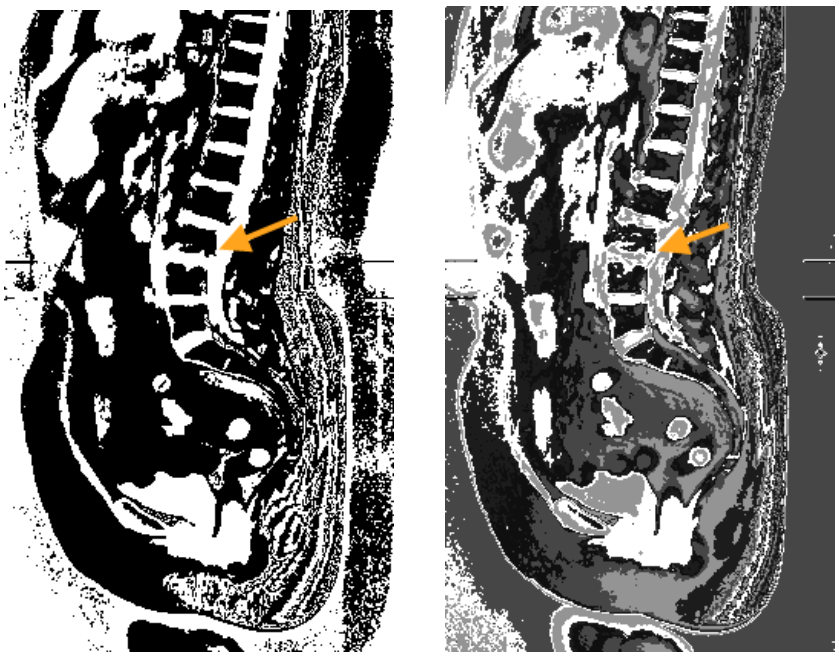


Figure 6.6: The plot shows the relative Hamming distances for the set of training volumes, that were normalized with the 936 trained models. One point indicates the distances for one model $\mathcal{M}_i$. For every model $\mathcal{M}_i$, the relative distance for disks (y-axis) is plotted against the relative distance for vertebrae (x-axis). The color-coding gives information about the model's target values $g_i' \in \{2 \dots 5\}$.



(a) (b)

Figure 6.7: Sample normalization results of trained ETM models at sampling distance $sd = 2.0$ with $s = 3$ (see Figure 6.7a) and $s = 4$ (see Figure 6.7b) target bins (95 source bins). A higher number of target values s often results in noisy images (see also Figure 6.6), especially in images with lower resolution. Data by courtesy of AGFA HealthCare [25].



(a) (b)

Figure 6.8: Mid-slice images of sample normalization results of trained ETM models at sampling distance $sd = 1.5$ with $s = 2$ (see Figure 6.8a) and $s = 5$ (see Figure 6.8b) target bins (110 source bins). More details are lost with the two-bin model than with the five-bin model. This is seen around vertebra $L3$, indicated by the orange arrow. The two-bin model provides a higher level of abstraction. Data by courtesy of AGFA HealthCare [25].

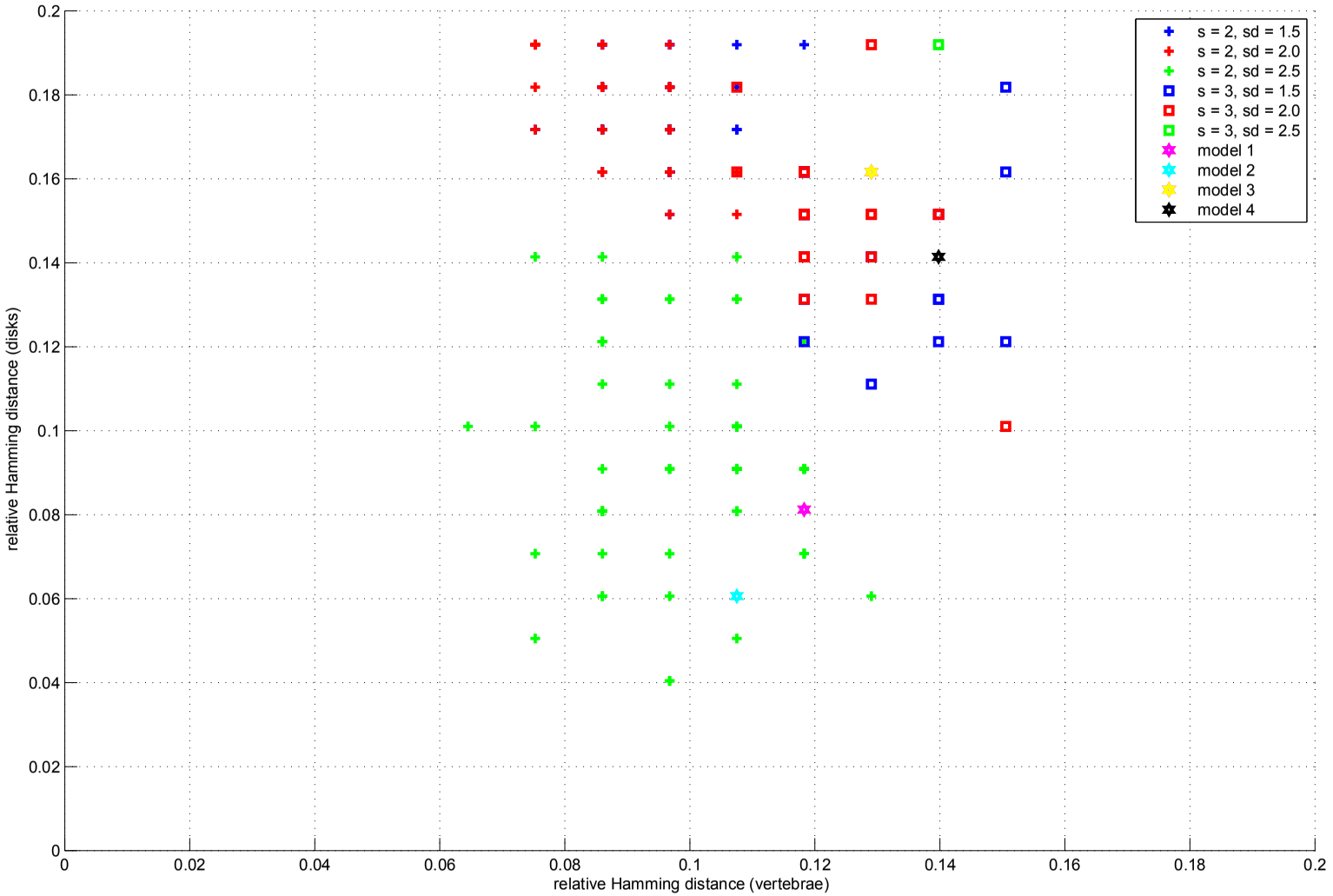
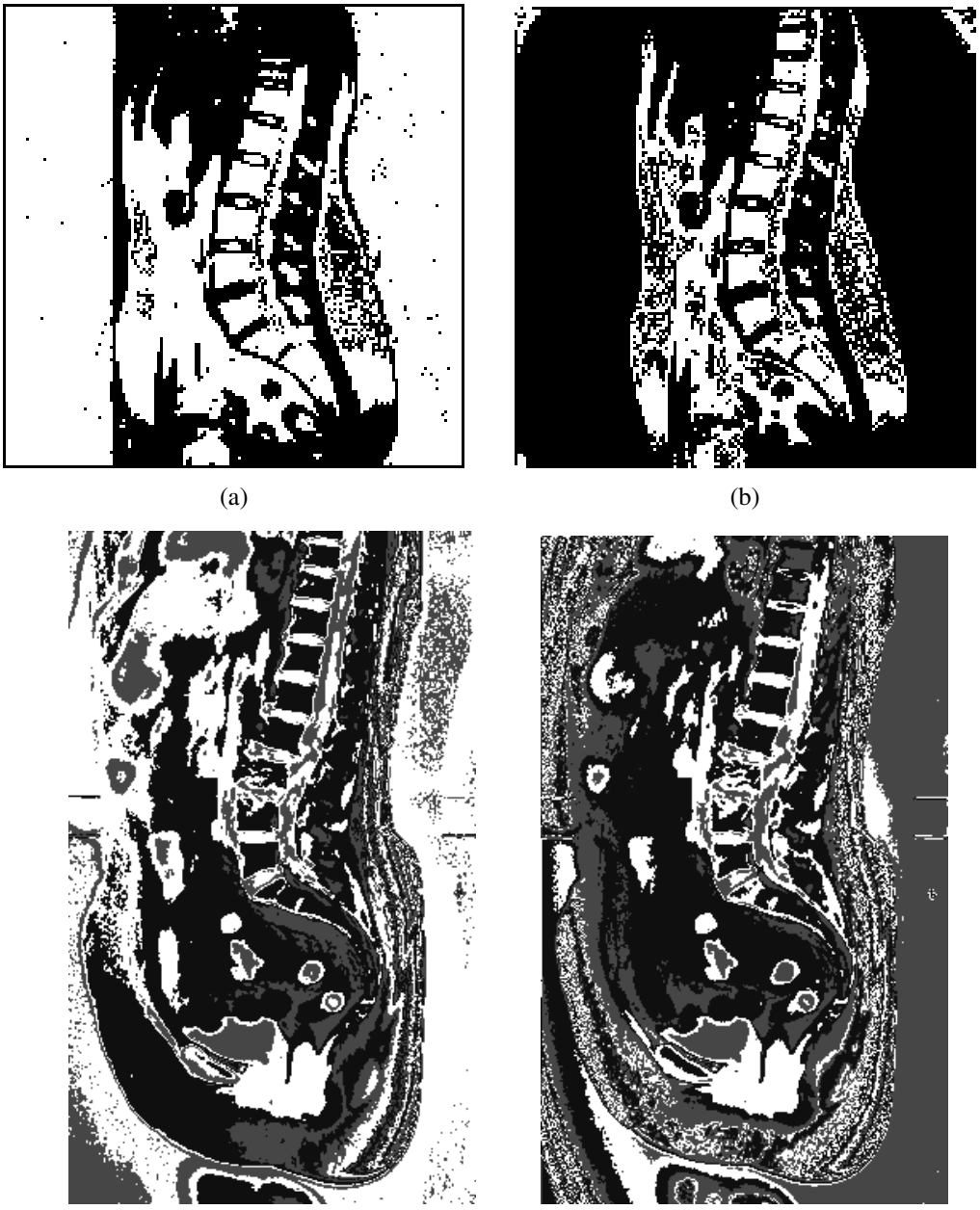


Figure 6.9: Models with relative Hamming distance < 0.2. The symbols indicate the number of target values and the color-coding the sampling distance. Note that many models exhibit equal distances, hence overlaps are present in this plot. Models selected for subsequent feature detection are highlighted. Their parameter configuration (e.g. number of source bins s, sampling distance sd) is listed in Table 6.2.



(a) (b)

(c) (d)

Figure 6.10: Sample mid-slice images of normalized training volumes with (see Figure 6.10b and Figure 6.10d) and without (see Figure 6.10a and Figure 6.10c) bias field correction. One model uses $s = 2$ target bins (see Figure 6.10a and Figure 6.10b)), the other model uses $s = 3$ target bins (see Figure 6.10c and Figure 6.10d). It seems that the two-bin model performs better with bias field correction than the three-bin model. However, the dataset in the bottom row is a cut-out of a fullspine scan, so not a real bias field is present there as in the dataset in the top row. Data by courtesy of AGFA HealthCare [25].

6.5 Disk Center Detection on Lumbar and Lower Thoracic Spine

The evaluation was carried out on 17 annotated testing volumes covering the lumbar and lower thoracic spine. In every volume, the disks in the range between $T11/T12$ and $L5/S1$ are in the ground truth, i.e. seven disks. Some volumes cover also the disks $T9/T10$ and $T10/T11$. In total, 133 disks were annotated in the 17 volumes (see Appendix Table A.1). Several models were tested, mapping the input data to $s = 2$ and $s = 3$ target bins. For the final comparison, four models $\mathcal{M}_i, i = 1 \ldots 4$ were selected (see Table 6.2 and highlighted in Figure 6.9). The models use initially $r = 110$ source bins and map to $s = 2$ or $s = 3$ target bins. Different sampling distances, landmark extraction methods and bias field processing are used. For every model, a lumbar and thoracic PBT detector were trained with two different initial re-sampling voxel sizes, i.e. $\Delta_{x,y,z} = 1.0$ mm and $\Delta_{x,y,z} = 1.5$ mm.

For every selected configuration, a batch evaluation was performed with a testing framework provided by VRVis [20]. The complete spine labeling pipeline was done for every testing volume, starting from every annotated intervertebral disk in the volume. This results in 133 iterations for one tested model configuration $\mathcal{M}_i$.

Figure 6.11 presents the labeling accuracy of the disk center positions obtained after the ETM model matching. No subsequent feature detection was performed. The plots report the number of missed disks (top), i.e. a disk center which was not detected by the algorithm but is present in the ground truth annotation, and the mean distance-to-disk-cylinder error per dataset (bottom). One value in the matrix indicates the mean distance-to-disk-cylinder error for a complete ETM preprocessing and labeling run, with initial click position in disk d_i^k with label λ_i (y-axis) in dataset I_k (x-axis). This holds analogously for the missed disk plot, where one value indicates the number of missed disk center points for the respective testing run. Only detected disks are considered for error calculation. Missed and extra disks do not contribute with a penalty term.

The results show, that no disk center points were missed, which is desired. However, the mean distance-to-disk-cylinder error suggests, that many disks lie outside the annotated cylinder, which is not a good result. This is also reflected in the recall of 85.5% and precision of 89.8%.

In the following, the labeling performance with a subsequent feature detection with PBTs on entropy-optimized training data is reported. Section 6.5.1 reviews the results achieved for mapping to two target bins and Section 6.5.2 summarizes results for the selected three-bin models. The influence of bias field correction is also reported for both target values.

Model	r_k	s	sd	extraction strategy	bias field correction
$\mathcal{M}_1$	110	2	2.5	Disk-Cylinder-Points	off
$\mathcal{M}_2$	110	2	2.5	Disk-Cylinder-Points	on
$\mathcal{M}_3$	110	3	1.5	Disk-Spinal-Centers-Plane	off
$\mathcal{M}_4$	110	3	1.5	Disk-Spinal-Centers-Plane	on

Table 6.2: List of trained models $\mathcal{M}_i$, which are selected for evaluation of spine labeling. Source bins r_k, target bins s, sampling distance sd, extraction strategy and bias field correction settings are listed for ETM matching.

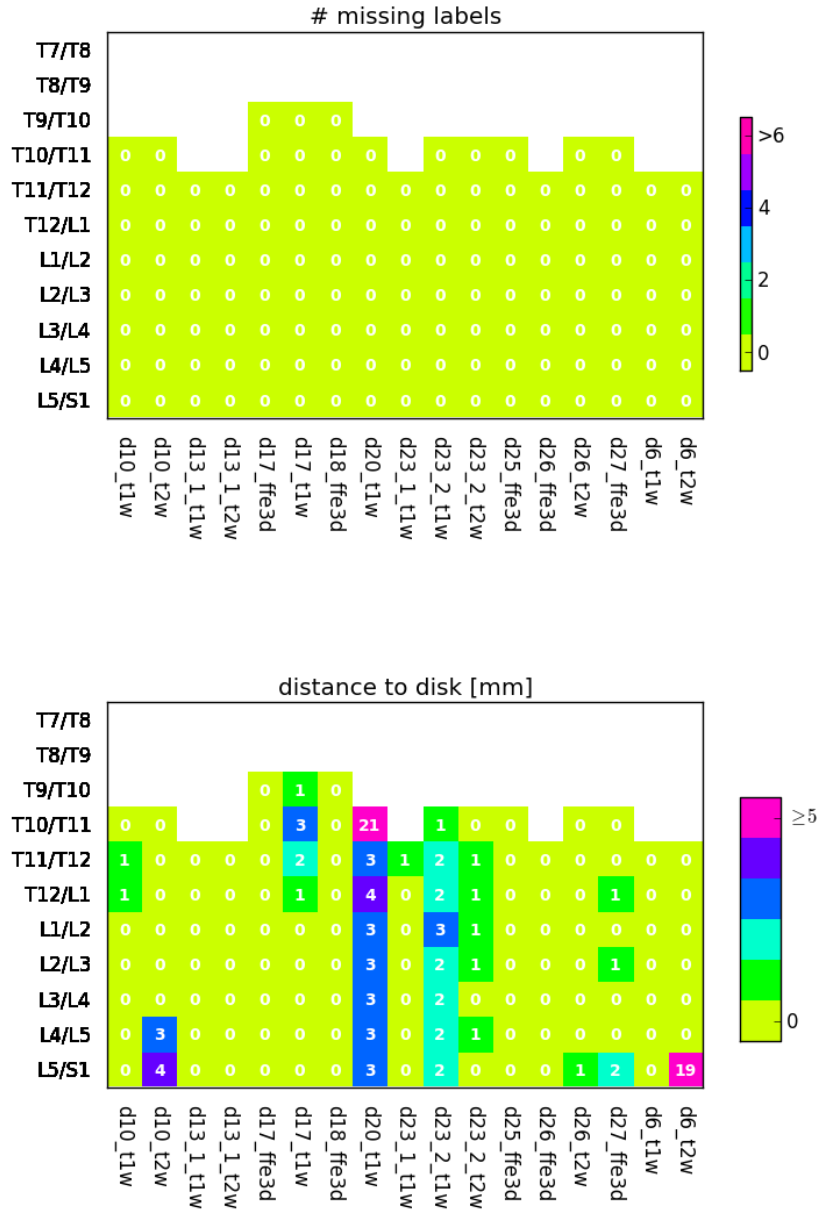


Figure 6.11: Labeling results achieved with matched three-bin model $\mathcal{M}_4$ on preprocessed volume data. No feature detection was performed, but the positions of the matched landmarks were evaluated. This method achieves a precision of 89.8% at a recall of 85.8%. Processing time for one dataset is only 2.66 s on average, since only model matching is performed. This corresponds to a mean processing time of 0.34 s per disk.

6.5.1 Labeling Results on Two-bin Models

The disk detection results achieved with model $\mathcal{M}_1$ and $\mathcal{M}_2$ are shown in Figure 6.12 and Figure 6.13.

The results obtained with model $\mathcal{M}_1$ trained on unprocessed data show, that a lot of intervertebral disks are not detected by the algorithm. For one reason, this happens due to missed disks in the border region of the scan (see Figure 6.14, left). The image shows, that with the two-bin model the low contrast region at the upper border maps only to one target intensity level (i.e. black). Hence no disks can be detected in scans that exhibit this low contrast region, i.e. a bias field. In three cases, the entropy optimization fails (see seven missed disks in Figure 6.12, dataset d17_ffe3d and d23_2_t1w) if the initial click is at the border region of the MR scans.

Applying model $\mathcal{M}_2$, which was trained and tested on bias field corrected scans, decreases the number of missed disks and the mean distance-to-disk-cylinder error (see Figure 6.13). Looking again at the sample result from volume d23_2_t1w (see Figure 6.14), it is apparent that the model provides a better mapping for the low contrast border region of the image. However, an extra disk is found which is not present in the ground truth data.

In terms of spatial precision and recall, model $\mathcal{M}_1$ achieves a precision of 94.7% at a recall of 82.1%. Using bias field correction (model $\mathcal{M}_2$) increases the recall to 88.3%, but also decreases precision to 92% due to more detected false positive disks in the thoracic region of the scans. Processing time is lower for model $\mathcal{M}_1$ due to a lower resolution of the data in the feature detection phase (re-sampling to $\Delta_{x,y,z} = 1.5$ mm). Mean processing time for one dataset is 4.8 s, which results in 0.6 s per disk. With model $\mathcal{M}_2$ processing time rises to 10.7 s per dataset resp. 1.4 s per disk on average. A detailed overview about timing performance is given in Section 6.6.

One problem with two-bin models is that they exhibit a very high level of abstraction. This can result in a loss of basic anatomical information in the mapped data, e.g. in low contrast regions at the border (see Figure 6.14). Disks and vertebrae map to the same target value. Even though bias field correction increases mapping quality, the mappings are not always optimal. Disk center points were missed for example because disks (nucleus and fibrous ring) mapped to black in the training data, but where normalized to white on the testing data (as seen on the missing candidates in $L5/S1$ in Figure 6.14).

6.5.2 Labeling Results on Three-bin Models

Comparing the results of both three-bin models $\mathcal{M}_3$ (see Figure 6.15, unprocessed data) and $\mathcal{M}_4$ (see Figure 6.16, preprocessing), we see again that bias field correction reduces the number of missed disk center points and the mean distance-to-disk-cylinder error. Again, one value in the matrix indicates the mean distance-to-disk-cylinder error for a complete ETM preprocessing and labeling run, with initial click position in disk d_i^k with label λ_i (y-axis) in dataset I_k (x-axis).

In terms of spatial accuracy, model $\mathcal{M}_3$ achieves a precision of 91.6% with a recall of 90.9%. Normalizing the data with model $\mathcal{M}_4$ gives only a slight increase of precision and recall to 91.9% resp. 91.7%. Extra detected disks (false positives) and disk points that lie outside the expert-annotated cylinder (false negatives) contribute negatively to precision and recall. Post-

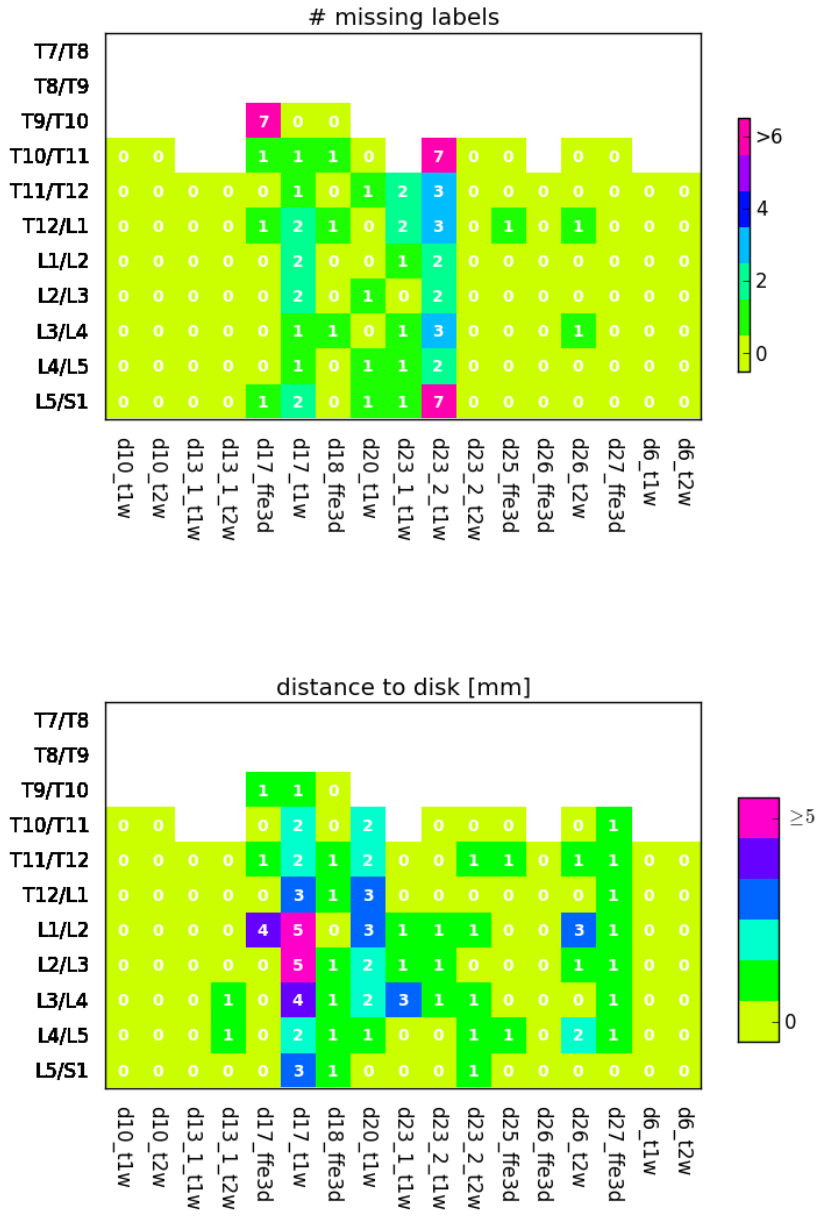


Figure 6.12: Labeling results achieved with the two-bin model $\mathcal{M}_1$ on unprocessed volume data. Lumbar and thoracic detectors Φ_L and Φ_T use initial re-sampling to $\Delta_{x,y,z} = 1.5$ mm. The algorithm achieves a precision of 94.7% at a recall of 82.1%.

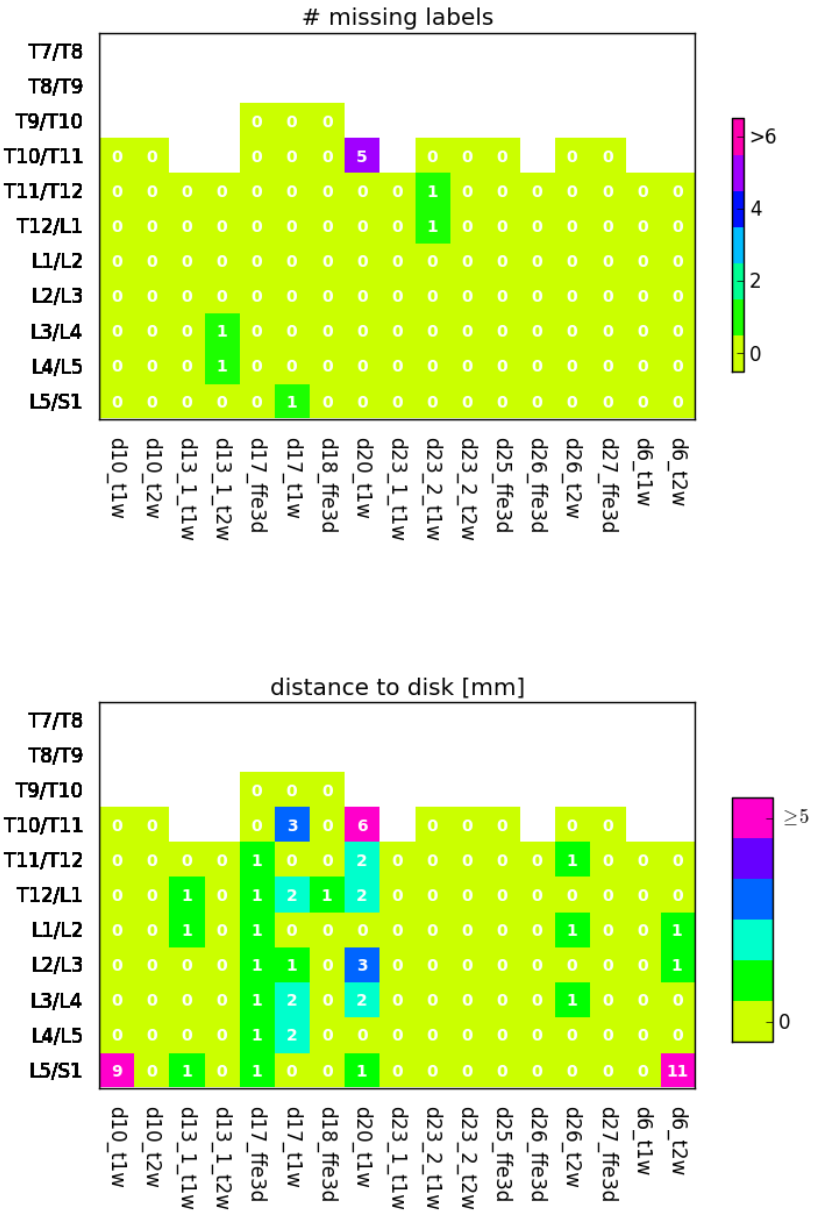
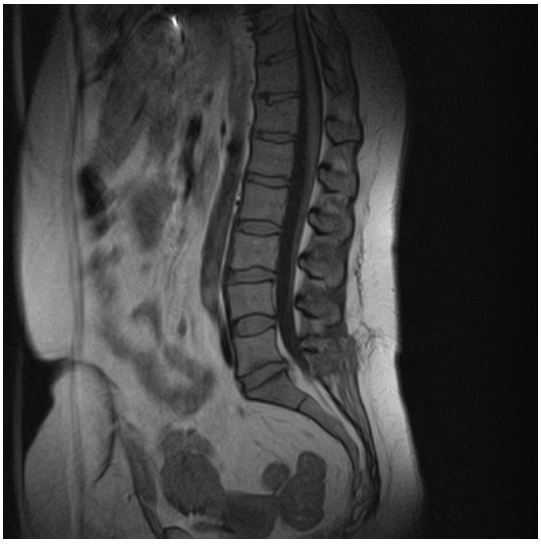
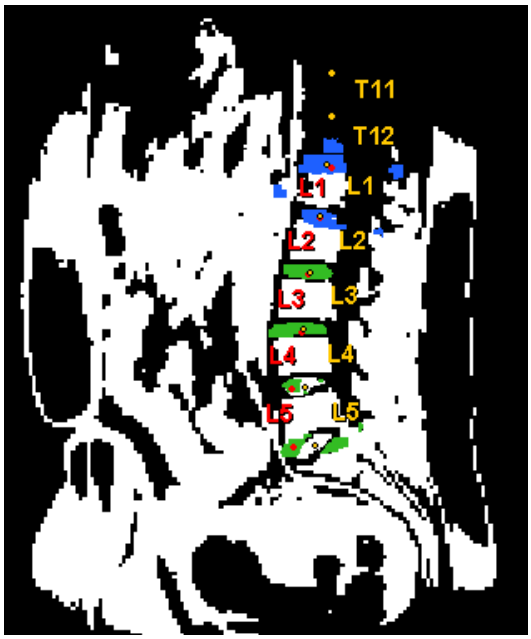


Figure 6.13: Labeling results achieved with the two-bin model $\mathcal{M}_2$ on bias field corrected volume data. Lumbar and thoracic detectors Φ_L and Φ_T use initial re-sampling to $\Delta_{x,y,z} = 1.0$ mm. The algorithm achieves a precision of 92.0% at a recall of 88.3%.



(a)



(b)

(c)

Figure 6.14: Original mid-slice image of dataset d23_2_t1w (see Figure 6.14a) and labeling results obtained with two-bin models $\mathcal{M}_1$ (see Figure 6.14b) and $\mathcal{M}_2$ (see Figure 6.14c). The extracted points are projected onto the normalized mid-slice image from the volume dataset. Detected disks and corresponding labels are shown in red, ground truth annotations in yellow. Feature points detected with the Φ_L and Φ_T detectors are shown in green and blue, respectively. On the unprocessed data, disks are missed in the thoracic region, whereas with preprocessing all disks are detected, as well as a disk which was not annotated ($T9/T10$). Data by courtesy of AGFA HealthCare [25].

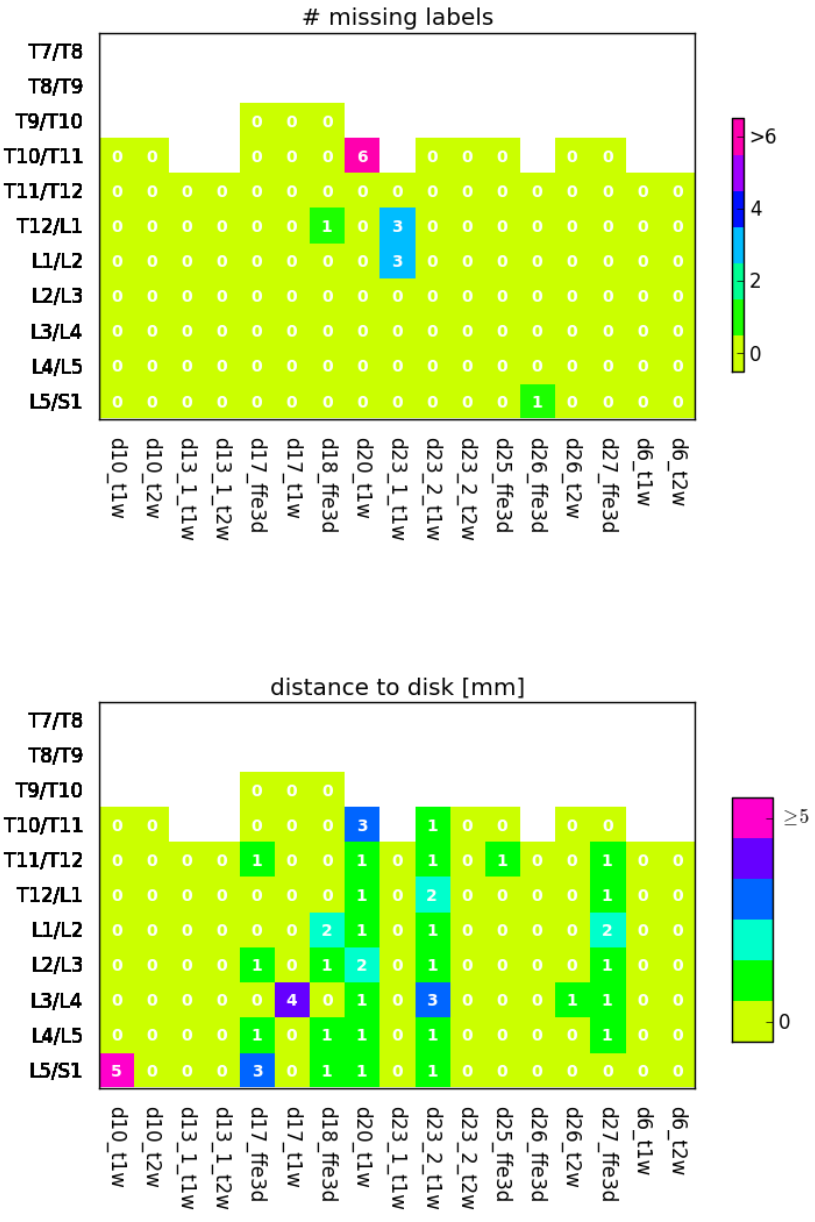


Figure 6.15: Labeling results achieved with the three-bin model $\mathcal{M}_3$ on unprocessed volume data. Lumbar and thoracic detectors Φ_L and Φ_T use initial re-sampling to $\Delta_{x,y,z} = 1.0$ mm. The algorithm achieves a precision of 91.6% at a recall of 90.9%.

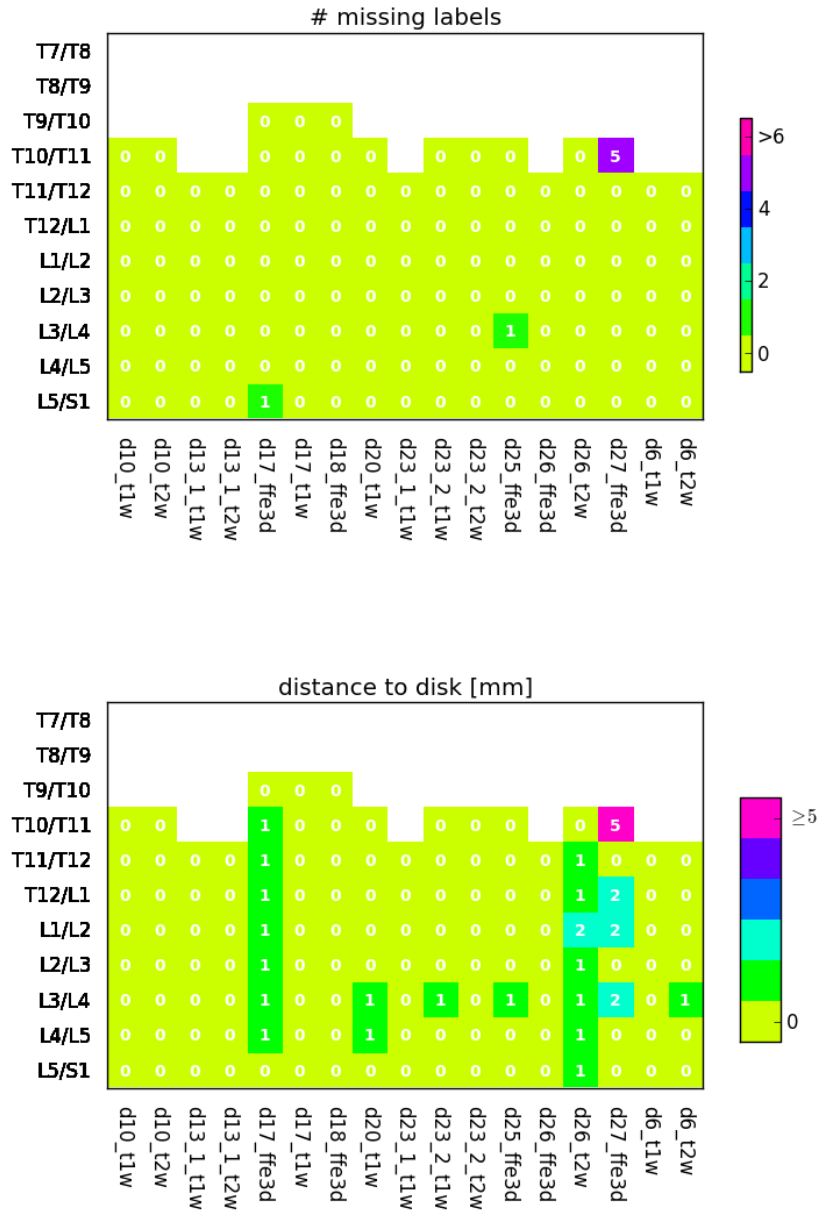


Figure 6.16: Labeling results achieved with the three-bin model $\mathcal{M}_4$ on bias field corrected volume data. Lumbar and thoracic detectors Φ_L and Φ_T use initial re-sampling to $\Delta_{x,y,z} = 1.0$ mm. The algorithm achieves a precision of 91.9% at a recall of 91.7%.

processing, e.g. fitting a spline to the detected points, could further enhance labeling accuracy at the cost of a higher processing time.

Processing time with model $\mathcal{M}_3$ for one dataset is 10.7 s on average, which corresponds to a mean processing time of 1.4 s per disk. Applying model $\mathcal{M}_4$ achieves a similar processing time of 12.4 s on average for one dataset, which corresponds to a mean processing time of 1.6 s per disk. A detailed overview of timing performance is given in Section 6.6.

Both results were obtained at a high resolution for feature detection ($\Delta_{x,y,z} = 1.0$ mm). Applying detectors Φ_L and Φ_T with initial re-sampling to $\Delta_{x,y,z} = 1.5$ mm to model $\mathcal{M}_4$ reduces timing to 6.0 s per dataset (0.8 s per disk). With this, an even higher precision of 92.4% at the cost of a lower recall of 86.8% is achieved.

Looking at the mapping results of the three-bin model $\mathcal{M}_4$ (see Figure 6.17), we see that for both weightings a reliable mapping is obtained with the trained ETM. In the T2w scan (bottom) we can observe a higher signal for the lower lumbar disks $L4/L5$ and $L5/S1$ than in the other disks. This results in a mapping to another target value (in this case black) than the disk mode $mode_d(\mathcal{M}_4) = 2$ for some voxels. Nevertheless, the model is capable to normalize the majority of nucleus voxels correctly and detect the disk centers within the cylinder. For other datasets where the nucleus signal is even higher (see Figure 6.18, left), the model maps the majority of voxels not to the disk mode, but also not to the vertebra mode as it was the case for the two-bin model. The nucleus is mapped to the third gray value, which was learned by the PBTs and therefore enables a detection of the disk.

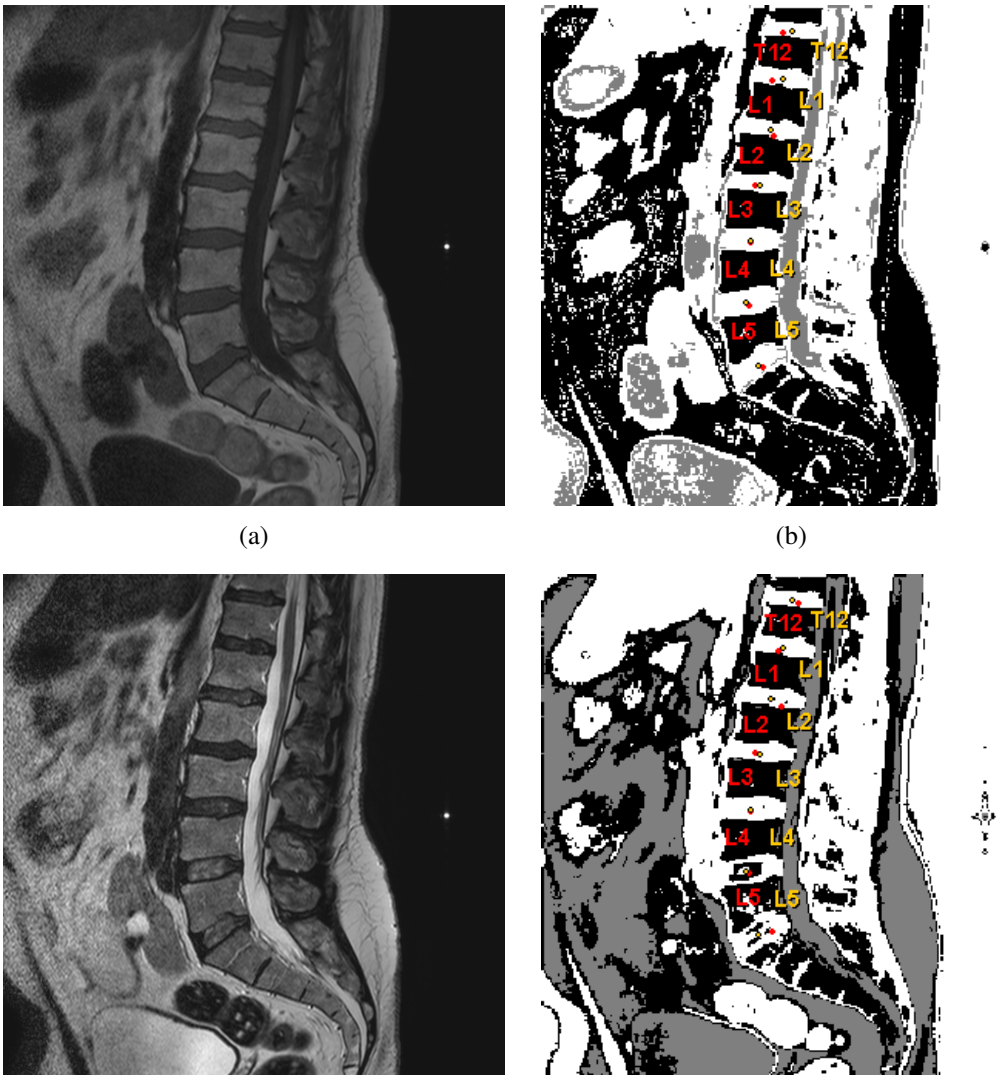
The presented three-bin models provide more reliable mappings than the two-bin models, which is reflected in the reported precision and recall of the models. Their level of abstraction is lower, which results in a better normalization and hence detection rate of the disk candidate voxels.

6.6 Hardware Setup and Performance

The computer used for implementation and testing of the presented framework has a Quad Core Intel i7 processor at 3.4 GHz and 8 GB of RAM. Training of ETMs, the bias field preprocessing and testing of the whole pipeline was performed on this computer. For training of PBTs, a Linux machine equipped with 32 GB of RAM was used. Time performance for bias field correction is reported in Section 6.6.1. Timings for the training and testing of ETMs and PBTs are reviewed in Section 6.6.2 and Section 6.6.3.

6.6.1 Bias Field Correction

Recall, that two methods for bias field correction were implemented in MATLAB. For the first method, the bias field is calculated only for the mid-slice of the volume. The mean processing time is 0.4 s $\pm$ 0.3 s. For the second method, the bias field is calculated for every slice independently. This results in a total timing of 1.8 s $\pm$ 1.0 s.



(a) (b)

(c) (d)

Figure 6.17: Original mid-slice images for T1w volume d6_t1w (see Figure 6.17a and T2w volume d6_t2w (see Figure 6.17c). Corresponding normalization results (see Figure 6.17b resp. Figure 6.17d) are obtained with model $\mathcal{M}_4$. Feature detection was performed with initial voxel size $\Delta_{x,y,z} = 1.0$ mm. Detected disk centers are shown in red, ground truth annotation is marked in yellow. The result is projected onto the normalized mid-slice image of the volume dataset. Data by courtesy of AGFA HealthCare [25].

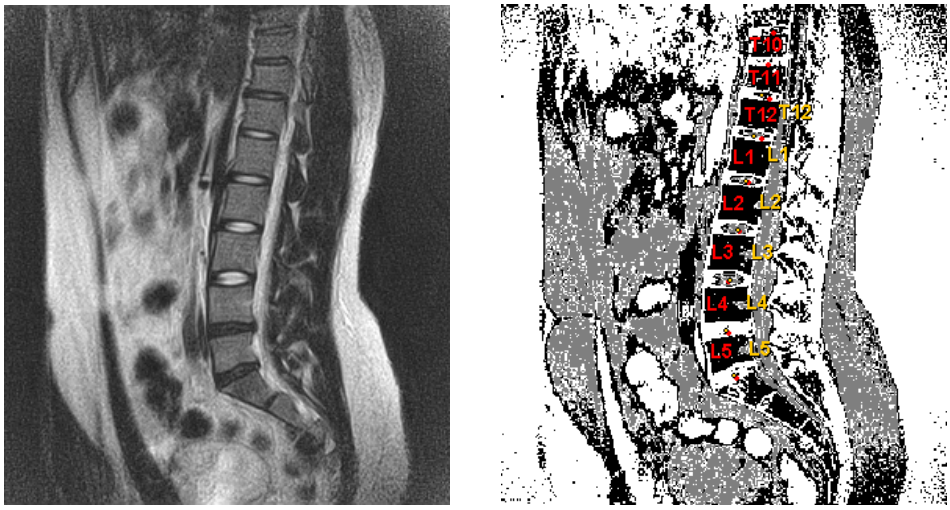


Figure 6.18: Original T2w mid-slice image from volume d13_1_t2w (left) and corresponding normalization result obtained with model $\mathcal{M}_4$ (right). Feature detection was performed with initial voxel size $\Delta_{x,y,z} = 1.0$ mm. Detected disk centers are shown in red, ground truth annotation is marked in yellow. The result is projected onto the normalized mid-slice image of the volume dataset. Data by courtesy of AGFA HealthCare [25]

6.6.2 Training of ETMs and PBTs

Training times for entropy models strongly depend on the parameter setup of the training. The average times vary between one and 27.5 min, whereby fast training is conducted with a high sampling distance of $sd = 2.5$ (see Figure 6.19) and a small number of target bins, i.e. $s = 2$. For this setting a minimum average timing of 1 min is reported (maximum average time 2.3 min). With an increasing number of target values s and for lower sampling distances $sd = 2.0$ (see Figure 6.20) and $sd = 1.5$ (see Figure 6.21), timing increases up to 27.5 min on average. Furthermore the size of the texture, i.e. the landmark extraction method, affects the timing.

Training of PBT classifiers with maximum tree depth of three and one level of downsampling $w = 1$ takes 32 min $\pm$ 6 min. Lumbar and thoracic detectors Φ_L and Φ_T were trained with Haar-like and gradient features (see Section 5.4.2).

6.6.3 Testing of Labeling Framework

The overall timing when labeling an unseen volume is determined by several factors:

1. *Bias field correction*
   For bias field correction, a processing time of 0.4 s $\pm$ 0.3 s was recorded (see Section 6.6.1).

2. *Data normalization*
   Matching of an entropy model requires between 1.0 s and 8.1 s on average (see Figure 6.22), whereby the parameter influence is similar as already reported for the training.

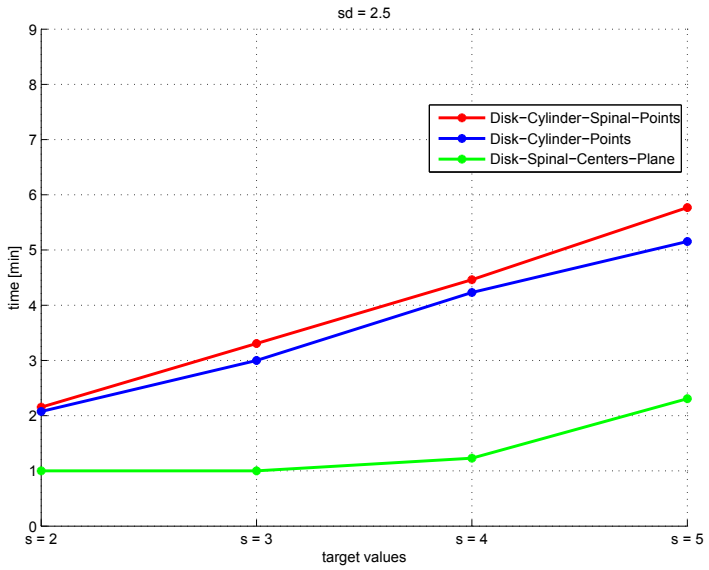


Figure 6.19: Average timing for training of an ETM at sampling distance $sd = 2.5$ for the three landmark extraction methods. The graph shows mean training times (in min) at four target values s (y-axis).

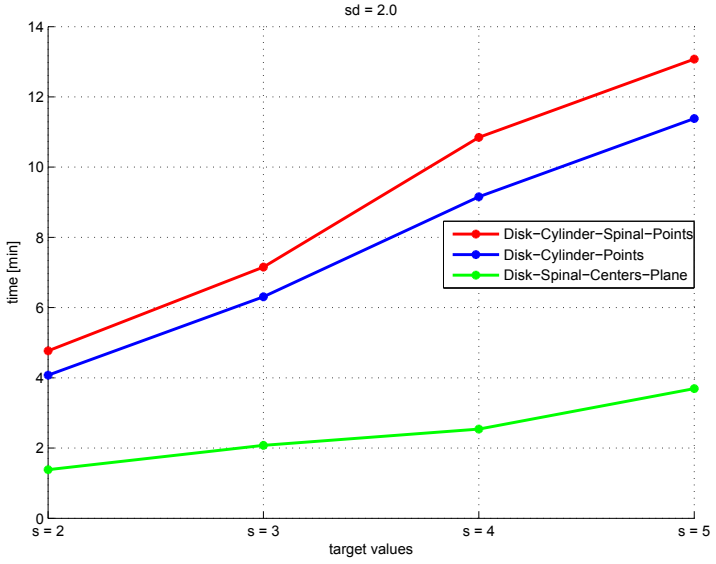


Figure 6.20: Average timing for training of an ETM at sampling distance $sd = 2.0$ for the three landmark extraction methods. The graph shows mean training times (in min) at four target values s (y-axis).

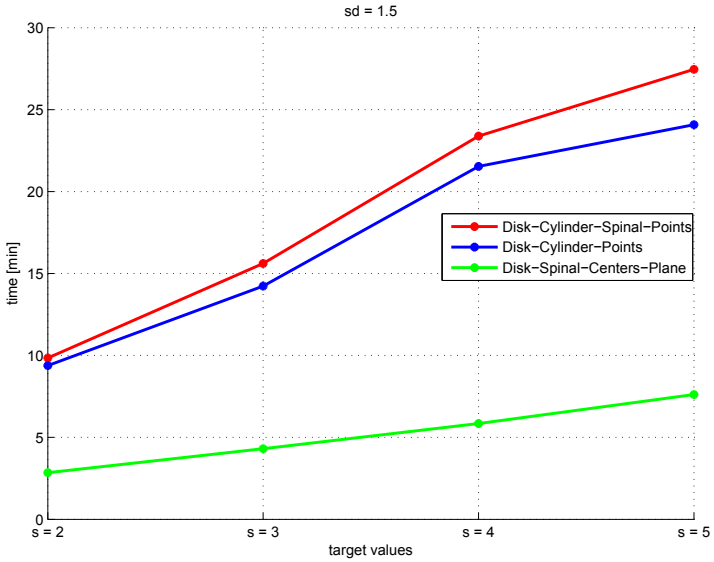


Figure 6.21: Average timing for training of an ETM at sampling distance $sd = 1.5$ for the three landmark extraction methods. The graph shows mean training times (in min) at four target values s (y-axis).

Timing increases with large textures and a small sampling distance, e.g. $sd = 1.5$. For both 3D extraction methods (*Disk-Cylinder-Points and Disk-Cylinder-Spinal-Points*), timing halves when increasing the sampling distance from $sd = 1.5$ to $sd = 2.0$. The plane-like texture extraction method (*Disk-Spinal-Centers-Plane*) provides fast model matching for all sampling distances. Maximum timing is 2.5 s on average at the smallest sampling distance $sd = 1.5$.

For the evaluated models in the previous section, an average matching time of 1.8 s for $\mathcal{M}_1$ and 1.9 s for $\mathcal{M}_2$ is reported for the two-bin models. Three-bin models $\mathcal{M}_3$ and $\mathcal{M}_4$ require 2.5 s resp. 2.8 s on average.

3. *Feature detection*
   The obtained PBT detectors from the training phase use Haar-like features and one level of downsampling. Two variants with initial re-sampling to $\Delta_{x,y,z} = 1.0$ mm and $\Delta_{x,y,z} = 1.5$ mm were tested within the evaluation of the labeling. Average timing for feature detection with initial resolution $\Delta_{x,y,z} = 1.0$ mm is between 8.2 s on $\mathcal{M}_3$ and 9.6 s on $\mathcal{M}_4$. Using re-sampling to $\Delta_{x,y,z} = 1.5$ mm decreases the mean timing to 3.0 s on $\mathcal{M}_1$.

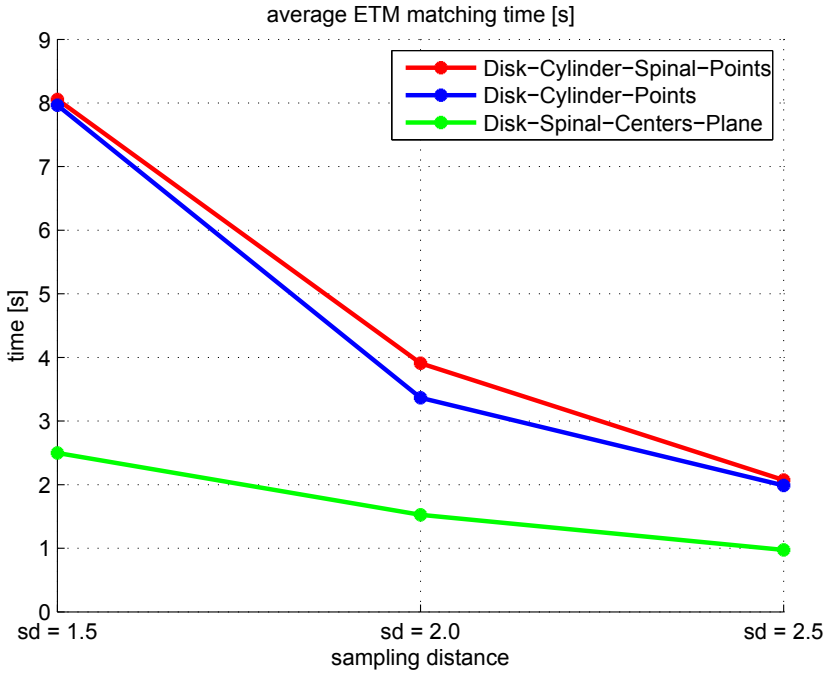


Figure 6.22: Average timing for matching an ETM at different sampling distances and extraction methods. The models evaluated use between 85 and 110 source bins.

6.7 Summary

The evaluation of the proposed semi-automatic labeling pipeline shows, that promising results can be achieved with this approach. To the best of our knowledge, no approach was presented so far in the literature, which *normalizes* and *labels* lumbar MR scans.

The entropy models learned from sparse annotations on unprocessed and preprocessed data capture the intensity variations in T1w and T2w MR scans. In order to locate intervertebral disks in unseen datasets, we evaluated the obtained positions from the ETM model matching, where we achieve a precision of 89.8%. at a recall of 85.5%. In order to increase the localization performance, we apply PBTs as disk detection algorithm on the normalized datasets, which are retrieved from the ETM model matching. The evaluation of various learned texture models shows, that mapping to $s = 3$ target values provides the best results in terms of the intensity mapping quality and generalization of the model. Using more than three target values results in noisier data. With $s = 2$ target bins, problems arise especially in low contrast regions, because too much anatomical information is lost due to homogeneous intensity mapping of adjacent disks and vertebrae. The problem of low contrast in the border regions of the dataset was addressed with an additional preprocessing algorithm. The integrated bias field correction method improved the quality of mappings for both evaluated two- and three-bin models. The improvement is also reflected in the evaluation of the disk center detection. For the three-bin model, precision increases from 91.6% to 91.9% and recall from 90.9% to 91.7%.

CHAPTER 7

Conclusion and Future Work

This final chapter concludes the thesis. Section 7.1 summarizes the presented work and the insights gained. Outlook on possible future work is then given in Section 7.2.

7.1 Conclusion

This thesis presented a new approach for semi-automatic spine labeling on T1w and T2w MR data acquired on various scanners. One main goal of this work was to learn one model, which is capable of normalizing T1w and T2w MR volumes to a reduced, standardized intensity scale. The task of labeling the spine should then be carried out on the preprocessed data. To achieve this goal, an *entropy-optimized texture model* [67] was learned from sparse annotations of the spinal anatomy. The model learns the intensity variation across the training set and normalizes the intensities to only three gray values without losing the main anatomical information from the original image data. *Probabilistic boosting trees* were trained on this normalized data for the localization of intervertebral disks. From the detected candidate voxels, the potential disk center was selected by weighting every detected voxel.

The presented processing pipeline achieves promising results in terms of data normalization and disk localization. The algorithm performs with a precision of 91.9% and recall of 91.7% on unseen data with a mean processing time of 1.6 s per detected disk.

To the best of our knowledge, no such spine labeling approach on normalized data was presented in the literature. The strength of this method lies in the capability of processing T1w and T2w MR scans without the necessity to retrain the entropy model or the feature detection algorithm when processing one scan of this type.

To be comparable to current state-of-the art algorithms in MR spine labeling, a development towards a fully automatic approach is necessary. Furthermore, time performance and labeling accuracy have to improve. However, the work from Kelm et al. [32] for example reports only results on high-resolution T1w data and needs re-training for CT scans. Our proposed approach

can process T1w and T2w scans and has the potential to develop one combined model for CT and MR data, according to the original work from Zambal et al. [67]. This idea is considered as future work.

7.2 Future Work

At the end of this thesis, a short overview about ideas and improvements originating from this work are given.

Automation For use in clinical practice, a fully automatic spine labeling approach is preferred over semi-automatic methods. This comes with several challenges with the proposed method. So far it is ensured that the ETM is initialized with the desired region of interest, because of the user's initial click. Now with a fully automatic approach, the task is to find good seed points for placing the model first. One possibility is to use the Histogram of Oriented Gradient Feature for the detection of candidate voxels. Lootus et al. [36] reported successful candidate detection on MR T1w scans and also on CT data without re-training of the algorithm. Performance on T2w scans is not stated and has to be investigated.

Body Region So far the ETM was trained on and applied to scans covering the lumbar and lower thoracic body region exclusively. An extension to other body regions and fullbody scans is necessary in order to compete with current state-of-the-art algorithms. Furthermore, for the use in a clinical environment, it is desired that the algorithm works on the whole spinal column.

Disk Detection Since the appearance of disks is simplified after the normalization, the complex PBT classifier could be exchanged with a simpler method. However, this needs further research in order to find suitable algorithms. Regarding the accuracy of the presented method, post-processing could further improve the result. Some detected disk centers were just outside the disk cylinder and therefore count as false negative detection. A correction with e.g. a spline could tackle this problem.

Time Performance The time performance can be further increased, e.g. by choosing a different disk detector. Since the appearance of disks is simplified after the normalization, the complex PBT classifier could be exchanged.

Multi-Modality Currently the algorithm works on T1w and T2w MR scans acquired with GR and TSE sequences. Zambal et al. [67] claim in their work, that the proposed texture model works not only with data from different scanners and various MRI sequences, but also on a mixture of modalities. Including CT scans to the training set could therefore provide a spine labeling framework that works on MR and CT data with only one training step. Furthermore the applicability of the already trained model to CT can be investigated. This area opens up a whole room for improvement and further research and development.

Gäbe es die letzte Minute nicht, so würde niemals etwas fertig.

Mark Twain

APPENDIX A

Datasets

All 28 training and testing volumes were acquired on scanners from Philips Medical Systems [56]. They were extracted from datasets from 14 different patients and cover the lumbar and lower thoracic spine (see Table A.1). Table A.2 gives a detailed overview of various properties like voxel sizes, resolution, etc. and patient-related information such as age and gender. The appendix of the volume names indicate the weighting of the data, hence `t1w` for T1w and `t2w` for T2w data (both TSE sequence), respectively. FFE scans (GR sequence) are identified by a trailed `ffe3d`.

Volume	Annotated Region
*d1_t1w	T8/T9 - L5/S1
*d1_t2w	T8/T9 - L5/S1
*d2_t1w	T8/T9 - L5/S1
*d2_t2w	T8/T9 - L5/S1
d6_t1w	T11/T12 - L5/S1
d6_t2w	T11/T12 - L5/S1
d10_t1w	T10/T11 - L5/S1
d10_t2w	T10/T11 - L5/S1
d13_1_t1w	T11/T12 - L5/S1
d13_1_t2w	T11/T12 - L5/S1
*d15_t1w	T9/T10 - L5/S1
d17_t1w	T9/T10 - L5/S1
d17_ffe3d	T9/T10 - L5/S1
*d18_t1w	T9/T10 - L5/S1
d18_ffe3d	T9/T10 - L5/S1
d20_t1w	T10/T11 - L5/S1
*d20_t2w	T9/T10 - L5/S1
d23_1_t1w	T11/T12 - L5/S1
d23_2_t1w	T10/T11 - L5/S1
d23_2_t2w	T10/T11 - L5/S1
*d24_t1w	T8/T9 - L5/S1
*d24_t2w	T8/T9 - L5/S1
*d25_t1w	T9/T10 - L5/S1
*d25_t2w	T9/T10 - L5/S1
d25_ffe3d	T10/T11 - L5/S1
d26_t2w	T10/T11 - L5/S1
d26_ffe3d	T11/T12 - L5/S1
d27_ffe3d	T10/T11 - L5/S1

Table A.1: Overview of training (marked with *) and testing volumes and the covered anatomical region.

Volume	**Scanner**	**Tesla**	**Sequence**	**Age**	**Sex**	**In-plane Resolution** in $voxel$	**# Slices**	**# Voxel**	**Voxelsize** $\nu_x \times \nu_y$ in mm	**Slice Distance** ν_z in mm	**Max. Intensity**
*d1_t1w	Achieva	1.5 T	TSE	71 y.	M	512 × 844	13	5,617,664	0.58594 × 0.58594	6	4000
*d1_t2w	Achieva	1.5 T	TSE	71 y.	M	512 × 844	13	5,617,664	0.58594 × 0.58594	6	4000
*d2_t1w	Achieva	1.5 T	TSE	68 y.	M	512 × 903	13	6,010,368	0.58594 × 0.58594	6	4000
*d2_t2w	Achieva	1.5 T	TSE	68 y.	M	512 × 908	13	6,043,648	0.58594 × 0.58594	6	4000
d6_t1w	Intera	1.5 T	TSE	56 y.	M	512 × 512	13	3,407,872	0.63477 × 0.63477	4.4	2914
d6_t2w	Intera	1.5 T	TSE	56 y.	M	512 × 512	13	3,407,872	0.63477 × 0.63477	4.4	1323
d10_t1w	Achieva	1.5 T	TSE	67 y.	M	512 × 512	16	4,194,304	0.63477 × 0.63477	4.4	2254
d10_t2w	Achieva	1.5 T	TSE	67 y.	M	512 × 512	14	3,670,016	0.63477 × 0.63477	4.4	1270
d13_1_t1w	Panorama	0.23 T	TSE	23 y.	M	324 × 324	12	1,259,712	1.1725 × 1.1725	5.5	255
d13_1_t2w	Panorama	0.23 T	TSE	23 y.	M	320 × 320	13	1,331,200	1.1872 × 1.1872	5	251
*d15_t1w	Panorama	0.23 T	TSE	65 y.	F	324 × 324	12	1,259,712	1.1725 × 1.1725	5.5	255
d17_t1w	Panorama	0.23 T	TSE	70 y.	F	324 × 324	11	1,154,736	1.1725 × 1.1725	5.5	255
d17_ffe3d	Panorama	0.23 T	GR	70 y.	F	288 × 384	16	1,769,472	0.93732 × 0.93732	4	250
*d18_t1w	Panorama	0.23 T	TSE	43 y.	F	324 × 324	12	1,259,712	1.1725 × 1.1725	5.5	253
d18_ffe3d	Panorama	0.23 T	GR	43 y.	F	288 × 384	18	1,990,656	0.93732 × 0.93732	4	255
d20_t1w	Panorama	0.23 T	TSE	54 y.	F	324 × 324	12	1,259,712	1.1725 × 1.1725	5.5	255
*d20_t2w	Panorama	0.23 T	TSE	54 y.	F	320 × 320	13	1,331,200	1.1872 × 1.1872	5	245
d23_1_t1w	Panorama	0.23 T	TSE	41 y.	F	324 × 324	12	1,259,712	1.1725 × 1.1725	5.5	255
d23_2_t1w	Panorama	0.23 T	TSE	42 y.	F	324 × 324	12	1,259,712	1.1725 × 1.1725	5.5	255
d23_2_t2w	Panorama	0.23 T	TSE	42 y.	F	320 × 320	13	1,331,200	1.1872 × 1.1872	5	255
*d24_t1w	Achieva	1.5 T	TSE	71 y.	F	512 × 846	13	5,630,976	0.58594 × 0.58594	6	4000
*d24_t2w	Achieva	1.5 T	TSE	71 y.	F	512 × 853	13	5,677,568	0.58594 × 0.58594	6	4000
*d25_t1w	Panorama	0.23 T	TSE	37 y.	F	324 × 324	12	1,259,712	1.1725 × 1.1725	5.5	245
*d25_t2w	Panorama	0.23 T	TSE	37 y.	F	320 × 320	13	1,331,200	1.1872 × 1.1872	5	245
d25_ffe3d	Panorama	0.23 T	GR	37 y.	F	288 × 384	18	1,990,656	0.93732 × 0.93732	4	248
d26_t2w	Panorama	0.23 T	TSE	41 y.	F	320 × 320	13	1,331,200	1.1872 × 1.1872	5	253
d26_ffe3d	Panorama	0.23 T	GR	41 y.	F	288 × 384	18	1,990,656	0.93732 × 0.93732	4	255
d27_ffe3d	Panorama	0.23 T	GR	68 y.	M	288 × 384	18	1,990,656	0.93732 × 0.93732	4	251

Table A.2: Overview about the properties of training (marked with *) and testing volumes.

Bibliography

[1] CGAL, Computational Geometry Algorithms Library. `https://www.cgal.org`. Online; Accessed: April, 5th 2014.

[2] R.S. Alomari, J.J. Corso, and V. Chaudhary. Labeling of lumbar discs using both pixel- and object-level features with a two-level probabilistic model. *IEEE Transactions on Medical Imaging*, 30(1):1–10, 2011.

[3] Statistik Austria. *Österreichische Gesundheitsbefragung 2006/2007*. 2007.

[4] D.H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111 – 122, 1981.

[5] B. Belaroussi, J. Milles, S. Carme, Y.M. Zhu, and H. Benoit-Cattin. Intensity non-uniformity correction in MRI: existing methods and their validation. *Medical Image Analysis*, 10(2):234–246, 2006.

[6] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[7] Y. Boykov and G. Funka-Lea. Graph cuts and efficient N-D image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.

[8] M. Breitenseher. *Der MR-Trainer, Wirbelsäule*. Thieme, 2011.

[9] C. Chevrefils, F. Chériet, G. Grimard, and C.-E. Aubin. Watershed segmentation of intervertebral disk and spinal canal from MRI images. In *Image Analysis and Recognition*, volume 4633 of *Lecture Notes in Computer Science*, pages 1017–1027. Springer Berlin Heidelberg, 2007.

[10] OpenStax College. The Vertebral Column. OpenStax CNX. `http://cnx.org/contents/e0231e7f-70fd-426d-87e5-574ce51411cb@4/The_Vertebral_Column`. Online; June 27, 2013. Accessed: February 4, 2014.

[11] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. In *Computer Vision – ECCV'98*, volume 1407 of *Lecture Notes in Computer Science*, pages 484–498. Springer, 1998.

[12] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models–their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[13] D.S. Corenman, M.D. Thoraco-Lumbar Spine Fractures. http://neckandback.com/conditions/thoraco-lumbar-spine-fractures. Online; Accessed: August, 29th 2014.

[14] Microsoft Corporation. Microsoft Visio. http://office.microsoft.com/en-001/visio/. Online; Accessed: April, 5th 2014.

[15] J.J. Corso, R.S. Alomari, and V. Chaudhary. Lumbar disc localization and labeling with a probabilistic model on both pixel and object features. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2008*, volume 5241 of *Lecture Notes in Computer Science*, pages 202–210. Springer Berlin Heidelberg, 2008.

[16] X. Dong, H. Lu, Y. Sakurai, G. Yamagata, H. Zheng, and M. Reyes. Automated intervertebral disc detection from low resolution, sparse MRI images for the planning of scan geometries. In *Machine Learning in Medical Imaging*, volume 6357 of *Lecture Notes in Computer Science*, pages 10–17. 2010.

[17] J. Fanghänel, F. Pera, F. Anderhuber, R. Nitsch, and A.J. Waldeyer. *Waldeyer – Anatomie des Menschen*. De Gruyter, 18th edition, 2009.

[18] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[19] M. Forsting, D. Uhlenbrock, and I. Wanke. *MRT der Wirbelsäule und des Spinalkanals*. Thieme, 2nd edition, 2009.

[20] VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH. http://www.vrvis.at/. Online; Accessed: April, 5th 2014.

[21] Y. Freund and R.E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory*, volume 904 of *Lecture Notes in Computer Science*, pages 23–37. Springer Berlin Heidelberg, 1995.

[22] J.C. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1), 1975.

[23] Computational Biology Research Group. HTML5 PivotViewer. http://www.cbrg.ox.ac.uk/data/pivotviewer/. Online; Accessed: June, 12th 2014.

[24] R.W. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 29(2):147–160, 1950.

[25] AGFA HealthCare. http://www.agfahealthcare.com/. Online; Accessed: June, 18th 2014.

[26] C.L. Hoad and A.L. Martel. Segmentation of MR images for computer-assisted surgery of the lumbar spine. *Physics in Medicine and Biology*, 47(19):3503, 2002.

[27] Z. Hou. A review on MR image intensity inhomogeneity correction. *International Journal of Biomedical Imaging*, 2006, 2006.

[28] Medical Imaging and Technology Alliance. DICOM. `http://dicom.nema.org/`. Online; Accessed: June, 6th 2014.

[29] B. Jähne. *Digitale Bildverarbeitung*. Springer, 7th edition, 2012.

[30] I.T. Jolliffe. *Principal Component Analysis*. Wiley Online Library, 2nd edition, 2005.

[31] J. Juntu, J. Sijbers, D. Van Dyck, and J. Gielen. Bias field correction for MRI images. In *Computer Recognition Systems*, volume 30 of *Advances in Soft Computing*, pages 543–551. Springer Berlin Heidelberg, 2005.

[32] B.M. Kelm, M. Wels, S.K. Zhou, S. Seifert, M. Suehling, Y. Zheng, and D. Comaniciu. Spine detection in CT and MR using iterated marginal space learning. *Medical Image Analysis*, 17(8):1283–1292, 2013.

[33] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simmulated annealing. *Science*, 220(4598):671–680, 1983.

[34] P. Kittipanya-ngam and T.F. Cootes. The effect of texture representations on AAM performance. In *IEEE International Conference on Pattern Recognition*, volume 2, pages 328–331. IEEE, 2006.

[35] H. Lippert, D. Herbold, and W. Lippert-Burmester. *Anatomie. Text und Atlas*. Urban & Fischer, 8th edition, 2006.

[36] M. Lootus, T. Kadir, and A. Zisserman. Vertebrae detection and labelling in lumbar MR images. In *Computational Methods and Clinical Applications for Spine Imaging*, volume 17 of *Lecture Notes in Computational Vision and Biomechanics*, pages 219–230. Springer International Publishing, 2014.

[37] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.

[38] D. Major, J. Hladůvka, F. Schulze, and K. Bühler. Automated landmarking and labeling of fully and partially scanned spinal columns in CT images. *Medical Image Analysis*, 17(8):1151 – 1163, 2013.

[39] C.D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 1st edition, 2008.

[40] MathWorks. MATLAB. `http://www.mathworks.de/products/matlab/`. Online; Accessed: April, 5th 2014.

[41] A. Neubert, J. Fripp, C. Engstrom, R. Schwarz, L. Lauer, O. Salvado, and S. Crozier. Automated detection, 3D segmentation and analysis of high resolution spine MR images using statistical shape models. *Physics in Medicine and Biology*, 57(24):8357–8376, 2012.

[42] A. Neubert, J. Fripp, K. Shen, O. Salvado, R. Schwarz, L. Lauer, C. Engstrom, and S. Crozier. Automated 3D segmentation of vertebral bodies and intervertebral discs from MRI. In *IEEE International Conference on Digital Image Computing Techniques and Applications*, pages 19–24. IEEE, 2011.

[43] W.R. Nitz, V.M. Runge, and S.H. Schmeets. *Praxiskurs MRT: Anleitung zur MRT-Physik über klinische Beispiele*. Thieme, 2nd edition, 2011.

[44] M.S. Nixon and A.S. Aguado. *Feature Extraction & Image Processing for Computer Vision*. Academic Press, 3rd edition, 2012.

[45] L.G. Nyúl and J.K. Udupa. On standardizing the MR image intensity scale. *Magnetic Resonance in Medicine*, 62:1072–1081, 1999.

[46] OsiriX. DICOM sample image sets. `http://www.osirix-viewer.com/datasets/`. Online; Accessed: October 17, 2013.

[47] V. Pekar, D. Bystrov, H.S. Heese, S.P.M. Dries, S. Schmidt, R. Grewer, C.J. Harder, R.C. Bergmans, A.W. Simonetti, and A.M. Muiswinkel. Automated planning of scan geometries in spine mri scans. In *Medical Image Computing and Computer-Assisted Intervention — MICCAI 2007*, volume 4791 of *Lecture Notes in Computer Science*, pages 601–608. Springer Berlin Heidelberg, 2007.

[48] Z. Peng, J. Zhong, W. Wee, and J.-H. Lee. Automated vertebra detection and segmentation from the whole spine MR images. In *IEEE International Conference of the Engineering in Medicine and Biology Society*, pages 2527–2530. IEEE, 2005.

[49] J.P.W. Pluim, J.B.A. Maintz, and M.A. Viergever. Mutual-information-based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, 22(8):986–1004, 2003.

[50] M. Reiser, F.-P. Kuhn, and J. Debus. *Duale Reihe Radiologie*. Thieme, 3rd edition, 2011.

[51] R.E. Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

[52] S. Schmidt, J. Kappes, M. Bergtholdt, V. Pekar, S. Dries, D. Bystrov, and C. Schnörr. Spine detection and labeling using a parts-based graphical model. In *Information Processing in Medical Imaging*, volume 4584 of *Lecture Notes in Computer Science*, pages 122–133. Springer Berlin Heidelberg, 2007.

[53] F. Schulze, D. Major, and K. Bühler. Fast and memory efficient feature detection using multiresolution probabilistic boosting trees. *Journal of WSCG*, 19(1):33–40, 2011.

[54] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Thomson, 3rd edition, 2008.

[55] R.M. Summers. Road maps for advancement of radiologic computer-aided detection in the 21st century. *Radiology*, 229(1):11–13, 2003.

[56] Philips Medical Systems. `http://www.medical.philips.com/`. Online; Accessed: September, 29th 2014.

[57] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 1st edition, 2011.

[58] K.D. Toennies. *Guide to Medical Image Analysis – Methods and Algorithms*. Advances in Computer Vision and Pattern Recognition. Springer London Limited, 2012.

[59] Z. Tu. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *Proceedings of IEEE International Conference on Computer Vision*, volume 2, pages 1589–1596. IEEE, 2005.

[60] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Proceedings on Computer Vision and Pattern Recognition*, volume 1, pages 511–518. IEEE, 2001.

[61] P. Viola and W.M. Wells III. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.

[62] U. Vovk, F. Pernus, and B. Likar. A review of methods for correction of intensity inhomogeneity in MRI. *IEEE Transactions on Medical Imaging*, 26(3):405–421, 2007.

[63] T. Vrtovec, S. Ourselin, L. Gomes, B. Likar, and F. Pernuš. Automated generation of curved planar reformations from MR images of the spine. *Physics in Medicine and Biology*, 52(10):2865–2878, 2007.

[64] M. Wels, B.M. Kelm, A. Tsymbal, M. Hammon, G. Soza, M. Sühling, A. Cavallaro, and D. Comaniciu. Multi-stage osteolytic spinal bone lesion detection from CT data with internal sensitivity control. In *Proceedings of SPIE Medical Imaging*, volume 8315, pages 1–8. International Society for Optics and Photonics, 2012.

[65] C. Westbrook, C.K. Roth, and J. Talbot. *MRI in Practice*. Wiley-Blackwell, 4th edition, 2011.

[66] J. Yao and R. M. Summers. Statistical location model for abdominal organ localization. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009*, volume 5762 of *Lecture Notes in Computer Science*, pages 9–17. Springer Berlin Heidelberg, 2009.

[67] S. Zambal, K. Bühler, and J. Hladůvka. Entropy-optimized texture models. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2008*, volume 5242 of *Lecture Notes in Computer Science*, pages 213–221. Springer Berlin Heidelberg, 2008.

[68] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, and D. Comaniciu. Four-chamber heart modeling and automatic segmentation for 3-D cardiac CT volumes using marginal space learning and steerable features. *IEEE Transactions on Medical Imaging*, 27(11):1668–1681, 2008.