Data and Knowledge Engineering for Medical Image and Sensor Data

Franz Graf

München 2011

Data and Knowledge Engineering for Medical Image and Sensor Data

Franz Graf

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig–Maximilians–Universität München

vorgelegt von
Franz Graf

München, den 20.10.2011

Erstgutachter: Prof. Dr. Hans-Peter Kriegel,
Ludwig-Maximilians-Universität München

Zweitgutachter: Prof. Dr. Bernhard Wolf,
Technische Universität München

Tag der mündlichen Prüfung: 31.01.2012

"Grant me the serenity to accept the things I cannot change,
Courage to change the things I can,
And wisdom to know the difference."

Zusammenfassung

Geräte zur medizinischen Bildgebung sind mittlerweile zu einem festen Bestandteil medizinischer Einrichtungen in industrialisierten Ländern geworden, so dass die medizinische Bildgebung und deren Möglichkeiten aus dem Gesundheitswesen nicht mehr wegzudenken sind. Die gewonnenen Bilddaten müssen jedoch wie alle anderen medizinische Daten auch für eine vergleichbar lange Zeit gespeichert und archiviert werden. Eines der größten Probleme, mit denen sich derartige Archive beschäftigen stellt die Durchsuchbarkeit des Datenbestandes dar, da sich derzeitige Suchoptionen oft auf den Inhalt digitaler und digitalisierter Berichte und Annotationen beschränkt.

Der erste Teil dieser Arbeit beschäftigt sich mit einem Problem aus dem Arbeitsablauf von Radiologen, die mit Computer Tomographie (CT) Daten arbeiten. In diesem Teil wird ein neues Verfahren vorgestellt, das Query-By-Example Anfragen in CT Volumendaten mit einem Minimum an Anfrageinformation realisiert. Im weiteren Verlauf wird ein Verfahren zur automatischen Detektion der Wirbelsäule vorgestellt. Die Ergebnisse dieses Verfahrens können zum Beispiel zur Initialisierung semi-automatischer Verfahren verwendet werden, die derzeit manuell initialisiert werden müssen.

Der zweite Teil dieser Arbeit beschäftigt sich mit der Analyse medizinischer Sensordaten. Die Bedeutung körperlicher Aktivität im Bereich medizinischer Vorsorge als auch im Bereich der Therapie ist unumstritten. Es ist jedoch nicht ganz einfach, die körperliche Aktivität eines Patienten zu messen, sobald er sich nicht mehr in einem kontrollierten Umfeld (z.B. einer Reha-Klinik) aufhält, sondern zu Beispiel zuhause. Derzeitige Lösungen setzen zum Teil

voraus, dass der Patient detailliert über seine Aktivitäten Buch führt, was für den Patienten weder bequem noch sonderlich objektiv ist. In diesem Teil der vorliegenden Arbeit wird ein neuartiges Verfahren zur Analyse und Klassifikation von Daten vorgestellt, die von einem miniaturisiertem 3D Accelerometer aufgenommen wurden. Des Weiteren wird eine Software vorgestellt, mit Hilfe derer sowohl Algorithmen erstellt werden können, die jedoch auch als Basis für die Benutzeroberfläche der entsprechenden Anwendung für Ärzte und Patienten verwendet werden kann um die Daten der körperlichen Aktivität, die der Sensor aufgenommen hat, entsprechend aufzubereiten und darzustellen.

Insbesondere im Bereich der computergestützten (medizinischen) Bildgebung ist es üblich, dass sog. Feature Vektoren benutzt werden um Bilder oder Teile von Bildern zu beschreiben. Dasselbe gilt für die Analyse der Sensordaten im Anwendungsbereich der körperlicher Aktivität und deren Erkennung und Klassifikation. Um Bildinhalte oder Aktivitäten entsprechend genau zu beschreiben sind diese Feature Vektoren meist hochdimensional, was spätestens dann Probleme verursacht, wenn diese Vektoren zur effizienten Suche in Datenbanksystemen verwendet werden sollen. Insbesondere nächste-Nachbarn-Anfragen können in diesem Fall oft nicht effizient durchgeführt werden. Jedoch stellt genau diese Art Anfragetyp einen essentiellen Schritt in den vorgestellten Verfahren dar. Dieses Problem wird im dritten Teil dieser Arbeit beleuchtet. Dort wird zunächst ein generalisiertes Verfahren zur nächste-Nachbarn-Suche in hochdimensionalen Daten vorgestellt. Im Gegensatz zu konventionellen Datenbanksystemen werden in diesem Fall die Daten nicht zeilenweise sondern spaltenweise organisiert. Diese Methode arbeitet jedoch nur effektiv, wenn die Dimensionalität der Features groß genug ist. In Fällen wie der Aktivitätserkennung ist die Dimensionalität der entsprechenden Vektoren zu niedrig für diese Methode, jedoch immer noch zu hoch für klassische Indexstrukturen. Dieses Problem wird im letzten Teil der Arbeit beleuchtet, bei dem die Auswirkung neuer Hardware auf die Effizienz untersucht wird.

Abstract

Several modalities of medical imaging have become standard equipment in modern health care facilities of industrialized countries so that it is unimaginable to do without medical imaging in current and future health care. Like other medical data as well, the produced imagery data has to be stored in archives for a comparatively long time. One of the largest problems is the searchability of such data archives. Current search options are often restricted to a plain text search that can only search within digital (or digitalized) reports and possibly also annotations.

The first part of this thesis focuses on a problem stated by radiologists that are working on Computer Tomography (CT) data. In that part, a new technique is presented that allows a query-by-example search in CT volume scans that requires only a minimal set of input data to obtain a very accurate result. The second part of the medical imaging topic in this thesis covers the automatic detection of the vertebra within a single CT image. The results of this method can be used as an initialization for several other techniques that are yet only semi-automatic as they often need a manual initialization.

The second part of the thesis is concerned with the analysis of medical sensor data. This work was motivated by the importance of physical activity to modern health care. The importance of physical activity in medical prevention and in therapy is non-controversial. However it is comparatively hard to monitor the physical activity of a patient if he is not in a controlled environment like a hospital. Currently this can be accomplished if the patient is writing a detailed log, yet this is neither convenient nor objective. In this

part of the thesis, a new algorithm is presented that analyses the accelerometer data of a miniaturized sensor in order to classify the activity that has been performed. Furthermore a software is presented which is used for building the algorithms as well as the GUI which should be used by attending doctors as well as patients to monitor the physical activity.

Especially in the field of computer vision and imaging, the use of feature vectors is common to describe an image or just parts of an image. Same can be said in the case of activity classification where feature vectors are also used to describe segments in the raw data. In order to describe an image content or activity precisely enough for the according use case, such feature vectors usually have a dimensionality that causes severe problems if they should not just be stored but also effectively retrieved from a data base. Especially nearest neighbor queries can often not be executed in an efficient way. However such queries are essential in the shown algorithms. This motivates the third part of the thesis. First a generalized method for nearest neighbor search in very high dimensional data is proposed. In contrast to common data base systems, this system employs a vertical decomposition of the data. However this method only performs if the dimensionality is large enough, like for example in the medical imaging task. In cases like the activity recognition, the dimensionality of the feature vectors is a too low for this technique but yet too large for common index structures. This issue is addressed in the second half of this part where the impact of new hardware on classical index structures is evaluated.

Contents

Zusammenfassung **vii**

Abstract **viii**

I Preliminaries **1**

1 Introduction **3**

1.1 Medical Imaging 3

1.2 Sensor Data 4

1.3 Indexing 5

2 Outline of the Thesis **7**

II Medical Imaging **9**

3 Introduction **11**

4 Slice Localization **19**

4.1 Use Cases 19

4.2 Problem Description 22

4.3 Related Work 24

4.4 Mutli Represented Descriptor 25

4.4.1 Introduction 25
4.4.2 Feature Extraction 27
4.4.3 Localization 32
4.5 Radial Descriptor 34
4.5.1 Introduction 34
4.5.2 Feature Extraction 35
4.5.3 Localization 40
4.6 3D Detection 41
4.6.1 Introduction 41
4.6.2 3D Features 42
4.6.3 Prediction 43
4.7 Evaluation 44
4.7.1 Data Set 44
4.7.2 Annotation 46
4.7.3 Multi Represented Descriptor 47
4.7.4 Radial Descriptor 54
4.7.5 3D Detection 62

5 Vertebra Detection 65
5.1 Introduction 65
5.2 Related Work 67
5.3 Static Detection 67
5.3.1 Introduction 67
5.3.2 Algorithm 69
5.3.3 Refinement 75
5.3.4 Performance Tuning 76
5.4 Weighted Detection with Dynamic Resize 76
5.4.1 Introduction 76
5.4.2 Algorithm 77

5.5 Evaluation 85
5.5.1 Data Set 85
5.5.2 Static Detection 87
5.5.3 Weighted Detection with Dynamic Resize 96

III Medical Sensor Data 105

6 Introduction 107

7 Activity Recognition 111
7.1 Introduction 111
7.2 Related work 112
7.3 Feature Extraction 114
7.3.1 Signal Reconstruction 114
7.3.2 Segmentation 115
7.3.3 Feature Extraction 117
7.3.4 Linear Discriminant Analysis 122
7.4 Classification 123
7.4.1 Classifying Features 123
7.4.2 Reclassification 123
7.5 Experiments 124
7.5.1 Data 124
7.5.2 Evaluation 125

8 *Knowing*: A Generic Data Analysis Application 129
8.1 Introduction 129
8.2 Architecture 132
8.2.1 Data Storage 132
8.2.2 Data Mining 133
8.2.3 User Interface 135

8.2.4 Modularity 135
8.3 MedMon 135

IV Indexing 139

9 Introduction 141

10 BeyOND – unleashing BOND 145
10.1 Introduction 145
10.2 BOND revisited 148
10.3 BeyOND BOND 150
10.3.1 Sub Cubes 152
10.3.2 MBR Caching 154
10.3.3 Experiments 155

11 Impact of Solid State Drives on Spatial Indexing 161
11.1 Introduction 161
11.2 Related Work 164
11.3 Changes in the access path 165
11.3.1 Caching 165
11.3.2 New Storage Media 166
11.4 Evaluation 169
11.4.1 Datasets 169
11.4.2 Hardware 169
11.4.3 Software 170
11.5 Experimental Results 171
11.5.1 System Load and Storage Device Utilization 172
11.5.2 Impact of the Page Size 174
11.5.3 Query Size 175
11.5.4 Dimensionality 176

V Conclusions 179

12 Summary 181

12.1 Preliminaries 181
12.2 Medical Imaging (Part II) 182
12.3 Medical Sensor Data (Part III) 183
12.4 Indexing (Part IV) 184

13 Future Directions 187

13.1 Medical Imaging 187
13.2 Medical Sensor Data 189
13.3 Indexing 190

Image Licenses 191

List of Figures 193

List of Tables 195

List of Algorithms 197

References 197

Part I

Preliminaries

Chapter 1

Introduction

1.1 Medical Imaging

Medical Imaging is one of the most important features currently used in the field of medical diagnosis. Imaging thereby comprised a very large field of all kinds of techniques that can create an image representation of the body, parts of the body, the body surface and of course and probably most important the inside of the body. The methods that are used in order to obtain the data used to create such images includes several different recording techniques like Ultra sound, magnetic fields (MRI/MRT) and also electro magnetic radiation like infra red and X-ray. The great advantage of the mentioned techniques is that they are working non invasive with very few risk compared to invasive methods and very fast - especially if little or no preparation of the patient is required.

With the growing popularity of diagnosis relying on medical imaging, the need for better post processing of the image data increased as well. Post processing the data is not only used to enhance the quality of the image which is displayed to the clinician. Post processing also includes techniques to derive enhanced information from the raw data. This can include the automatic detection of spine deformation, the extraction of blood vessels or even 4d

models of a beating heart in order to plan surgeries. Another advantage of such methods is the possibility to extract information which are not noted explicitly in the according radiological report.

1.2 Sensor Data

Besides Medical Imaging, data from sensors plays a big role in current and future science and medicine. The term *sensors* in this context is very generic and can comprise all kind of devices that can be used to obtain (and possibly also store) certain measurements in its environment. Such sensors can for example be devices to measure the temperature of the air, the skin, the respiration or some kind of activity. Parts of this thesis are dealing with the recognition and classification of activity in medical use cases. In such cases, the sensors may be operated in a controlled environment for example when a patient is doing a special training during a rehabilitation. Another application is the long term observation of a patient where the patient is carrying the sensor in 24/7 mode in an uncontrolled environment. Such observation is desired if the time spent in a rehabilitation clinic should be shortened but it should still be measurable that the patient is continuing to perform a certain level of activity. Other cases could be to detect sudden decreases of activity which could indicate a threat to the patient's health status like in the case of patients suffering of COPD (Chronic Obstructive Pulmonary Disease).

In such cases, the sensor hardware is facing several restrictions and requirements: The sensor needs to be as small as possible so that the patient's quality of life is not reduced or affected in a way that his behavior is affected. It also needs to be unremarkable so that the patient is not stigmatized which would also lower the acceptance of the device.

After recording the data, methods and algorithms are required that analyze the data that is recorded by the sensor. Depending on the recording rate of the sensors, a huge amounts of data can be created that need to be analyzed in order to extract important facts from the data. Also, this data needs to be

aggregated and prepared before being displayed to the user.

1.3 Indexing

Indexing and similarity search are very related topics. Indexing aims at creating approximations and aggregates of entities in order to improve retrieval in files and data bases. The aim of similarity search is to find entities that are similar to one or more query objects. To accomplish this, it is common to apply a certain kind of distance function on two entities in order to measure their distance or - vice versa - their similarity. In the trivial case, the distance between the query entity and all other entities in the data base is computed and the entities with the smallest distance are returned as the result set. The computational overhead in that case is $O(n)$ with n entities being stored in the data base.

The use of indexing in such a case aims at reducing the amount of data that needs to be taken into consideration for such a query. This is usually accomplished either by certain hasing functions or by approximating distance values, so that groups of entities can be pruned from the search space without calculating their exact distance to the query entity.

While there exist well known solutions for low-dimensional spaces, it is common sense that similarity search in high-dimensional spaces is inherently difficult. Yet, the features that are extracted from (medical) images and sensor data usually have high and very high dimensions. Especially image features often have a dimensionality of more than 50 or more than 100 dimensions which is far beyond the dimensionality where classical index structures can perform well.

Chapter 2

Outline of the Thesis

The following content of this thesis is organized as follows:

Part II deals with problems in the area of medical imaging. *Chapter 3* provides an introduction to medical imaging technologies and their history. *Chapter 4* presents some works for the localization of single CT slices in CT Volume scans. *Chapter 5* then presents an approach to detect the vertebra on a single CT scan.

Part III describes the work that has been done on sensor data and the according analysis. *Chapter 6* introduces the topic about sensor data and activity recognition. *Chapter 7* describes a method for the detection and classification of activity data obtained from medical sensors. *Chapter 8* presents a generic application for data mining that was created to simplify the development of algorithms and prototypes in this context.

Part IV deals with the problems of indexing and similarity search in high dimensional data. *Chapter 9* introduces the topic briefly. *Chapter 10* presents an improved approach to index very high dimensional data in order to improve similarity search. In *Chapter 11* the impact of Solid State Disks (SSDs) compared to classical Hard Disk Drives (HDDs) is evaluated as the access paradigms that have been driving and restricting the development of index structures in the past decades are very different in the case of SSDs.

Part V summarizes and discusses the major contributions of the thesis in *Chapter 12*, followed by *Chapter 13* where ideas for possible future research are listed.

Part II

Medical Imaging

Chapter 3

Introduction

Medical imaging comprises a large variety of different techniques that have been developed in the past decades and centuries with all different kinds of application scenarios. The technologies in the imaging area can roughly be categorized in techniques using sonic waves (ultra sound), magnetic fields (MRI/MRT) and electromagnetic radiation (X-ray, CT, IR). The following chapter will give a brief introduction to each of these techniques with a focus on techniques based on electromagnetic radiation as this will turn out to be the main subject of the thesis.

Ultra Sound The term ultrasound refers to sound waves with a frequency which is greater than what an average human hearing can recognize. Depending on the person, this limit varies. Yet the frequency of 20 kHz is commonly regarded as the upper limit of the human hearing[113, 91, 41]. In medical (ultra) sonography, ultra sound waves with a frequency of about 1–15 MHz are used to visualize internal body structures, like muscles, organs and fetuses in the womb[62]. Medical sonography was first published in 1942 by the Austrian neurologist Karl Theo Dussik [38] and has since then gained very much interest. Medical sonography devices usually consist of a sound emitter that emits directional sound waves and a receiver/microphone that records the echo. By taking into account the possible effects of reflection, scatter

and absorption, an image can be reconstructed from the information which is recorded from the receiver/microphone. A big advantage of sonography is the small size of the devices, the small cost impact and the fact that the diagnosed patient is not exposed to any radiation compared to X-ray, so that the patient is not posed to any risks. Especially the last factor is very important in the field of breast mammography and prenatal diagnostics. During the past decade, 4D reconstruction techniques were developed, so that for example limb movements can be visualized a week earlier than in simple 2D in case of prenatal diagnostics [87].

MRT/MRI Another well known medical imaging technique is magnetic resonance tomography (MRT) which is also known as magnetic resonance imaging (MRI). Compared to other medical imaging techniques, MRI is comparatively new as it was first presented in 1973 by the American radiologist Paul Christian Lauterbur. MRI is based on very strong magnetic fields and radio frequencies. During an MRI, the body is placed inside a strong magnetic field that aligns the hydrogen molecules inside the body. Then, a radio frequency is introduced on the body. Afterwards, an emitted resonance frequency can be measured by surrounding sensors. Thereby, different relaxation times of different nuclei (usually protons of hydrogen atoms) in the body are measured to reconstruct a 2D or 3D information about the magnetic gradient distribution in the body. The fact that different kinds of tissues in the body result in different relaxation times in the according body region is used for defining the contrast in the image. While the first MRT devices were only able to visualize 2D slices through the scanned body part, modern techniques now even provide 4D visualizations of the patient. In the past decades, MRI has become a very important imaging technique to visualize tendons and ligaments. One of the major advantages of MRT compared to CT is that MRT does not utilize any ionizing radiation. 30 years later in 2003, Paul C. Lauterbur and Sir Peter Mansfield received the Nobel Price in Physiology and Medicine "*for their discoveries concerning magnetic resonance imaging*" [102].

EM Spectra. From γ-rays to IR Maybe the most obvious application of medical imaging is the analysis of photographs that can be taken with regular cameras which are recording the visible part of the electromagnetic spectrum covering the range of about 380 to 780 nm. In dermatology for example this technique is applied in order to detect and classify skin cancer [145, 148]. Sadeghi *et al* [119] for example perform a graph based pigment network detection method in order to detect structures of the pigment network. The results are used to classify the presence or absence of malignant melanoma, an aggressive type of skin cancer which causes about 75 % of deaths related to skin cancer [70].

Infrared, Thermography Next to the visible spectrum, the infrared spectra covers the spectral range from 780 nm to 1 mm. The analysis of images taken in this spectrum is called infrared or thermal imaging as the temperature distribution of objects can also be measured in this spectral range. In medical imaging, this spectrum is used to monitor the temperature distribution of the skin [29, 73] as this can indicate abnormalities like malignancies, inflammation and infections. Besides the application of thermographic screening in clinics, mass screening passengers using infrared cameras at airports has gained interest after the outbreak of SARS in 2002 and H1N1/A (swine flu) in 2009 [108]. The advantage of this method lies in the quick and non invasive possibility to screen a large amount of people for the identification of febrile patients without causing too large transition delays [31]. Nevertheless current studies state that relying on thermographic scanning and indications of fever alone do not yet achieve feasible results to become a full replacement of other methods [101]. Another application of thermographic imaging is breast thermography [48, 73, 9]. Since 1982, breast thermography is approved by the FDA for breast cancer risk assessment [76]. Thermography uses the fact that vessel activity in pre-cancerous tissue and tissue surrounding breast cancer is significantly higher than in normal tissue which shows up as regions with higher temperature than normal tissue. As there is currently no test that perfectly detects all cancers and as mammography and thermography analyze different pathological processes they are not suitable to replace each other

but should be regarded to supplement each other. Finally, Jiang *et al* [71] review several other usages of thermography in the medical field.

On the other side of the visible electromagnetic spectrum, X-rays cover the range between 10 pm and 1 nm, followed by the spectrum of gamma rays with wave lengths with less than 10 pm.

Scintigraphy, SPECT, PET Gamma rays (γ-rays) are electromagnetic radiation of very high frequency which are produced (amongst others) during decay of radionuclides (radioisotopes). Scintigraphy is a medical imaging technique, where radiopharmaceuticals are used to visualize the radiation of radionuclides and radiopharmaceuticals in the body. Such radiopharmaceuticals (a.k.a. radioactive tracer) can be substances that are enriched to emit radiation. The advantage of such tracers is that the body cannot distinguish between the regular and enriched substances so that the tracer integrates into the regular metabolism without disturbance after being taken internally. After having accumulated in the according organ or skeletal part, special cameras are used to capture and visualize the emitted radiation in two dimensional images. Scintigraphy is used for example in case of the diagnosis of pulmonary embolism. Another application is bone scintigraphy, which is used to detect and visualize abnormalities or bone metastases [121]. Single photon emission computed tomography (SPECT) and positron emission tomography (PET) use the principle of scintigraphy in a way that not only static two dimensional images are produced. In SPECT and PET, special cameras rotate around the patient's body and record the produced radiation. By using an inverse radon transform, sectional views through the body can be reconstructed.

X-ray and CT Last but not least, there is the spectral range between 10 pm and 1 nm which is covered by X-rays. X-rays were discovered November 8th in 1895 by the german physicist Wilhelm Conrad Röntgen. In 1901 he received the Nobel Prize "*in recognition of the extraordinary services he has rendered by the discovery of the remarkable rays subsequently named after him*" [103]. On December 12th in 1895 Röntgen recorded an X-ray image

of his wife's hand which can be seen in Figure 3.1. This started the history of medical X-ray. Just one year later in 1896: F.H. Williams reported the

Figure 3.1: Image of the first "Röntgenogram" in history which started a new era. Röntgen's first medical X-ray at November 8th, 1895 shows an X-ray image of his wife's hand.

first chest X-ray [126, 35]. In 1902, G. E. Pfahler and C. K. Mills reported the first X-ray of a brain tumor. In 1913 William David Coolidge invented the first hot cathode x-ray tube for the easier generation of x-rays. Four years later in 1917, the Austrian mathematician Johann Radon published the mathematical fundamentals of the "Radontransformation" [115] which forms the basis for the calculation of spatial objects from filtered back projection. Allan McLeod Cormack reinvented the radon transformation in 1963–64 as he

only stumbled over Radon's work by chance in 1972 [95]. Finally, in 1972 the first commercial CT Scanner was demonstrated by Godfrey N. Hounsfield at the Mayo Clinic (Rochester, MN, USA). Only two years later in 1974, the first commercial scanner 'SIRETOM' from a medical manufacturer (Siemens) was announced and five years later in 1979 G.N. Hounsfield and A.M. Cormack received the Nobel Price in Physiology and Medicine "*for the development of computer assisted tomography*" [104].

In the past century, X-rays have gained huge attention in the medical imaging field [35]. During this time, the use of X-rays has proved for the detection and visualization of pathology not only in bone structure but also in soft tissue. Well known applications for example are chest X-rays for the detection of tuberculosis, pneumonia and lung cancer and abdominal X-rays for the detection of stones in the gall and kidney and also X-rays in orthodontics and dentistry to analyze the jawbones and the teeth.

In fact, this success has led to more than 62 million CT scans in the U.S. in 2007 [26] and about 9.85 million scans in Germany [45] (doubled from 1996) so that in 2006 the average radiation exposure was about 3 mSv per year in the U.S. and 1.9 mSv in Germany [45]. Regarding these numbers, it might seem that MRI is the better choice over CT as MRI images appear similar without applying ionizing radiation. Nevertheless, the decision between MRI and CT strongly depends on the type of the exam. Cancer, pneumonia and chest X-rays for example are typical uses for CT as well as bleedings in the brain, bone injuries or visualizations of organs and the lung. MRI in contrast is the choice in case of visualizing tendons and ligaments as well as the density, composition or injuries of the spinal cord or tumors in the brain. Concluding, CT/MRI is used in cases of visualizing the morphology, whereas SPECT/PET is used for examining the metabolism.

The downside of this huge and even growing amount of CT scans each year is not only the exposure to radiation of the patients. Another problem arises with the sheer amount of information that is produced by this massive amount of images that is produced with each scan.

During the past, it has become quite common to scan large parts of a patient's body. The amount of image data that is produced during such a scan of course depends on a variety of factors. The main factors for the resulting size are the resolution of the CT scanner in all 3 axis and the size of the scanned body region. Depending on these factors, a typical thorax scan that covers the area between the hips and the shoulders can result in image volumes from 40 Mb to more than 1 Gb. Each volume is thereby composed of several million 3 dimensional voxels, where all voxels in the same plane form a 2 dimensional slice. The complete scan is called a (3d) volume scan. Modern systems can also produce 4d volume scans with the 4th dimension being the time. In a 4d scan, the clinician can for example observe a beating heart in full 3d. The rest of the thesis will mostly deal with 2d and 3d scans while the extension to 4d is trivial for the cases described in this thesis. After recording the volume scan, the data is processed and delivered to the radiology information system (RIS) and archived in a picture archiving and communication system (PACS) where it has to be archived for several years depending on country and state regulations.

Without proper methods for large scale and fully automatic methods for knowledge discovery and data mining in medical imaging, these PACS contain a huge amount of implicit know how which is only accessible through either prior knowledge to a certain volume scan or through the according health reports.

Chapter 4

Slice Localization

This chapter of the thesis deals with the automatic localization of a single image within a CT volume scan. The remaining chapter is starting with a description of use case in Section 4.1 to motivate the topic, followed by a more detailed problem description Section 4.2 and a discussion of the related work in Section 4.3. Afterwards, the first approach using a multi represented (MR) descriptor is presented in Section 4.4 which was extended to the radial descriptor described in Section 4.5. For cases, where more than a single slice is available, an extension of the radial descriptor is shown in Section 4.6. The three methods are evaluated in a combined evaluation, shown in Section 4.7.

Material presented in this chapter is published in [40, 53] and [54] with smaller data sets. Thus there will be slightly different values in the evaluation chapters of this thesis compared to the publications.

4.1 Use Cases

Single volume scan: If a radiologist is performing his diagnostics based on the volume scan that he just recorded and which is loaded into the RIS, he might not need additional image data in order to create the radiology report. This standard work flow should be covered by most of the RIS tools on the

market.

Loading a scan by a single slice: This use case deals with the situation where a clinician starts with a single CT slice. This is a typical situation if the clinician receives a report for example via email or from a radiologist who just forwards the most relevant slice instead of a complete volume. However, if the clinician needs to inspect the body area close to this image, he currently only has the possibility to request the complete scan from the PACS, navigate to the according position and then continue his diagnosis.

What he actually would need would be a possibility to request only a small sub volume of the original scan. This could for example be solved, if the PACS provides an outline of the scan so that he just requests a sub volume. A further improvement would be a query by example (QBE), where he can tell the PACS that he needs the sub volume of a certain scan that surrounds the query image so that he defines the region implicitly by just referencing the image.

Comparison of multiple scans: A more advanced yet also usual scenario is the situation where a clinician needs to check the convalescence of a patient. In such a case, the radiologist needs to compare the scan that he just recorded with a scan that was recorded at a previous time in order to judge the possible advances of treatments. In this case he needs to query the PACS in order to load the according volume scan of the patient. Depending of the size of the scan that is used for comparison he needs to load several hundred megabytes up to a gigabyte from the PACS via the network. The time between the radiologist requesting a volume scan and having the complete volume loaded in the RIS depends on various factors like the speed of his local computer, the network speed and load and the speed of the PACS server and of course on the size of the requested volume which will have to be loaded from disc on the PACS server. Assuming ideal conditions (no overhead for encoding, no additional load on the network, instant response and zero load on the PACS side), the transfer time for a single GB via a 100 Mbit network is 83 s plus at

least 8 s for the time to read the volume from the PACS disc. This results in a total of more than 1.5 min of loading time. One could argue that transfer time could be reduced by compressing the data. But it can be assumed that the time saved by compression is compensated in a certain way by non-ideal conditions of the system. Interviews with people working in radiology have shown that waiting times up to some minutes are not uncommon in this scenario. During this time the clinician can only proceed to a very limited degree as he is waiting for the system. When the scan is finally loaded, the clinician needs to align the scan loaded from the PACS to the part of the body shown in the newly recorded scan. After loading and aligning, he can proceed with his actual work.

It should be mentioned that in such a case, the clinician is loading the complete scan from the PACS even though he usually only requires a very particular and small sub volume of the data. In an improved workflow, the clinician would query the PACS to obtain a preview or outline of the scan, then select only the sub volume of interest and transfer only this small part of the data and thus save a large part of resources needed for the transfer of data.

In an even more improved workflow, the clinician would start in his newly created scan. In this scan, he would navigate to the according body position and then just requests a sub volume from another scan from the PACS by just hitting a button without having to define any further details. The system would then analyze the data which is displayed to the clinician, determine the according sub volume area automatically and query the PACS for the particular data which is of course far less than a complete scan. Assuming, that the local volume scan is already loaded, the steps needed in the current and the improved workflow would be as in Table 4.1.

Comparison to similar cases and knowledge discovery: In the previous use case, it was assumed that the clinician already knew exactly which scan he needs to open for comparison. The problem becomes a lot more complicated if the clinician's query is more imprecise like "*search for similar*

Table 4.1: Comparison of a current and an improved workflow, assuming that the first scan is already loaded.

Current workflow	Improved workflow
- determine scan B for comparison	- determine scan B for comparison
- load complete remote volume B	- load sub volume of B
- wait for scan to load ($> 1\,\text{min}$)	- wait for scan to load ($< 0.25\,\text{min}$)
- align scans manually	- system aligns scans automatically
- continue with diagnosis	- continue with diagnosis

entities in the same body region of other volume scans". In this case, two factors complicate the issue: The first factor is the restriction to the same body position and the second one is the fuzzy formulation of similarity.

4.2 Problem Description

This part of the thesis will deal with the latter problem of the fully automatic determination of the relative position of a given slice within the body which is important in the last two use cases introduced on page 20 and 21. Manufacturers of CT devices tend to argue that this issue is not a problem as the table position is encoded in the meta data of the DICOM files so that the body position can be derived directly from this coordinate. However, relying on the DICOM header information raises a couple of problems:

First of all, even if the table position is encoded reliably and correct, the patient's position on the table needs to be calibrated manually before each scan. Otherwise, the position information contains a certain offset which would have to be detected and compensated. Gueld *et al* [50] have shown in their work, that DICOM meta data entries like 'patient position' or 'body part examined' are often imprecise or even wrong. In such cases, this error would have to be detected and compensated again. A brief analysis of the 'patient position' values from the data used in this thesis fully supports the finding of Gueld *et al*as the values differ up to 20 cm from the expected

position.

In the above case it was assumed that meta data information is available. Especially in the case, where only a single slice is available, the meta data information need not be present or accessible. If the query slice is for example embedded in a report, then it depends on the embedding program whether or not the meta data is not modified, removed or accessible at all.

Even if the position of the patient would be calibrated, accurate and accessible, a pretty natural problem remains. Patients differ in height so that the absolute position of a slice is not sufficient for queries concerning scans of persons with different body sizes.

Due to these reasons, parsing DICOM header meta data does not yield a viable solution for obtaining the relative position of a single CT slice. The above use cases and discussion poses the following requirements to a method that can compensate the problems identified above:

The method solving the problem stated above should be **fully automatic**, so that it can be applied to a large amount of data without human intervention. It should also consume resources in terms of CPU and memory in a manner that allows **large scale deployment** which is crucial if the possibility should be taken into account that the method should be deployed on a clinical PACS containing several years of unprocessed patient data. This also requires **stability** of the method, so that there are no highly sensitive internal parameters that, when changed in a very small manner, have extraordinary impact to the results. As CT scans are often performed with different settings or contrast media taken internal, the method is required to be **robust** against contrast media, image modalities and if possible also robust against artifacts caused by implants. And last but not least, it must be ale to map people with different body size and shape into a **uniform height model**.

To achieve this, the prediction of the relative position along the z-axis (cf. Figure 4.1) is proposed. From a technical point of view the methods are based on gradient and texture features and employ instance-based regression for making predictions of the relative position of the slices within the body.

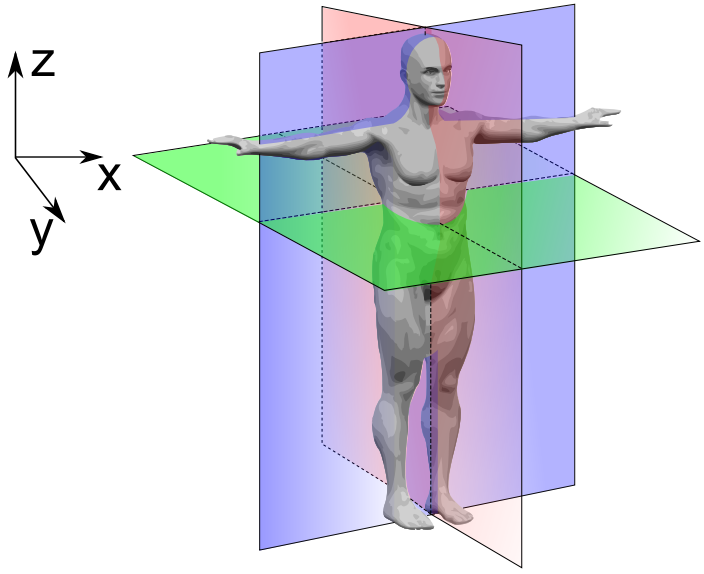


Figure 4.1: Schema showing the terminology for body planes. z: transverse plane (green), x: saggital plane (red), y: coronal plane (blue).

4.3 Related Work

As described in the previous sections, localizing a CT slice within a human body can enormously facilitate the workflow of a clinician. Nevertheless, this area of research has not yet received much attention. Even though there have been approaches to determine the body region from a topogram [27], the general approach is to localize invariant landmark positions as starting points and from there to interpolate for forming a relative coordinate system. Similarly, the authors of [60] propose an elastic mapping of the slice positions to a reference scale by detecting one of eight predefined anatomic points with known position and interpolating the position of the images between them. The authors of [32] propose a method to detect and localize a set of 10 different organs in CT images. They estimate both the location and the extent for each organ by predicting the bounding box containing each organ. They use a tree-based, non-linear regression approach based on multivariate

regression forests. These are similar to random forests but are able to predict continuous values instead of discrete classes. Seifert *et al* [123] proposed a method to detect invariant slices and single point landmarks in full body scans by using probabilistic boosting trees (PBT) [135] and HAAR-like features [107, 139]. Their algorithm detects up to 19 salient and robust landmarks within a volume scan. Subsequently, the detectors are incorporated into a Markov Random Field. Nevertheless, it cannot be used for localizing single slices or very small sub volumes as it operates on full body scans only. Also there need to be several landmarks detected in order for the algorithm to work.

So, previous techniques for localizing a CT slice within a human body model usually require more input than the actual single query slice. The approach which is most related to this work allows the localization of CT volume sets which was proposed by Feulner *et al* [43]. In this work the algorithm first detects the patient's skin and removes noise caused by the table and the surrounding air. From the remaining image, intensity histograms and SURF descriptors [11] are extracted and clustered into visual words. Afterwards, the method combines nearest neighbor classification with an objective function to classify and register the slices. The widths of the CT volume sets range between 44 mm and 427 mm. Using a scan with a high resolution such small sub volumes can comprise up to 50 slices. The average reported error lies between 44 mm for small query volumes and 16.6 mm for large query volumes. According to the authors, their method does not perform well when localizing single slices only.

4.4 Mutli Represented Descriptor

4.4.1 Introduction

The methods mentioned in section 4.3 usually generate complex models for large and pre-structured query input in form of CT sub volumes. This

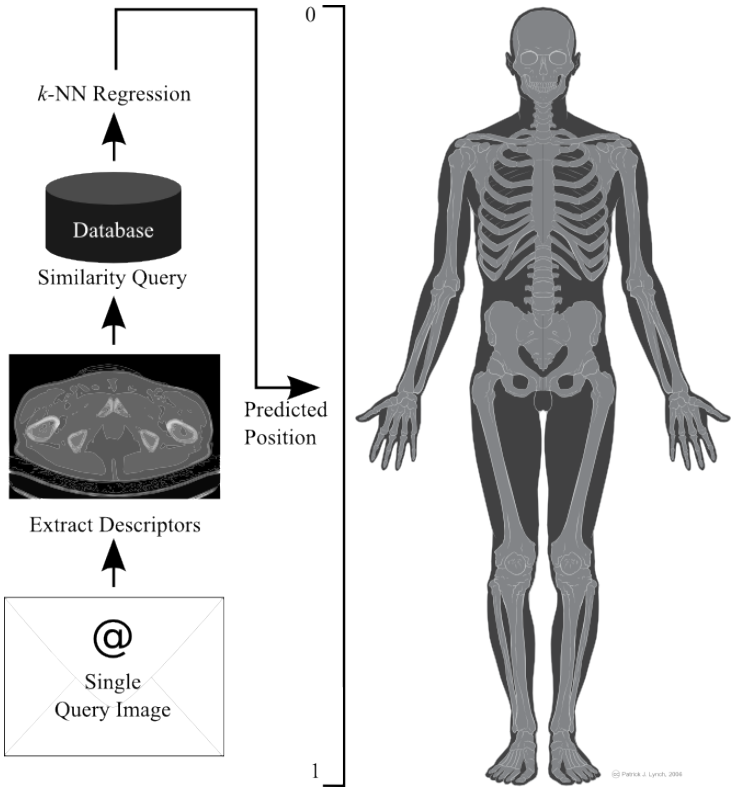


Figure 4.2: Slice localization by kNN regression.

method requires only a single query slice which is transformed into a feature vector $FV \in \mathbb{R}^{\mathcal{N}}$. This feature vector is used localize the image via k-nearest neighbor regression as illustrated in Figure 4.2. .

The idea of combining several feature representations is a well known technique in image retrieval and machine learning [94, 133, 8, 151, 33]. Therefore, the advantages of texture features and edge filters are leveraged by using the combination of histograms of oriented gradients (HoG) [24] and Haralick texture features [63] to describe the similarity between particular CT slices in order to optimally cover regions of enhanced uncertainty.

The extraction method itself is inspired by Lazebnik *et al* [89], who propose to use a spatial pyramid kernel to obtain locally sensitive features. In this approach, a modified spatial pyramid kernel is applied to obtain several

regular and rectangular, disjoint regions from the image. These regions act as information sources for the following extraction steps. Finally, each slice of a volume scan is represented by multiple feature descriptors of different kinds.

The localization process determines the position of the slice along the z-axis. An obvious but challenging problem of position prediction along the z-axis is the varying height of the patients. In order to solve this problem, each CT scan is scaled into a standardized height model with a domain of $[0, 1]$ with 1 representing the sole of the foot and 0 representing the top of the head. This mapping allows the localization of single slices independently of the persons' gender, height and age. In contrast to a method using absolute positioning, the proposed method is not prone to errors originating from patients of different heights.

4.4.2 Feature Extraction

Image descriptors using the first order derivate of the pixel data are well known from the field of object recognition [97] and scene recognition [89] and are usually applied to scenarios in the domain of digital photos or pictures. In the field of object recognition, feature extraction usually involves the extraction of multiple features per image with at least one feature vector describing an object of interest. The resulting bag of features is then stored in the database for later retrieval tasks. The advantage is that objects of interest can be described very locally and usually produce similar feature vectors even on different backgrounds. The drawback of this approach is usually a more complex distance measure. As two images are represented by bags of features, distance measures used to determine the (dis)similarity between images (like the sum of minimum distances or single link distance) often require $O(NM)$ runtime with N and M being the size of the feature bags.

In the field of scene classification, it is more common to use just a single feature vector in order to describe a complete image or scene. Typical descriptors are for example color histograms that are extracted from the complete image. As such a descriptor suffers from the loss of spatial information in the

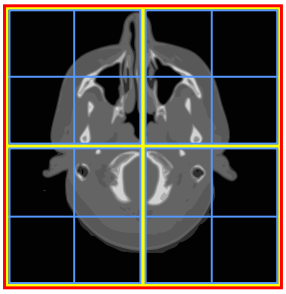
image, the idea of image gradients and texture features was combined with the idea presented in [24] where a spatial pyramid kernel was used for shape representation in order to classify regular images of the Caltech dataset[42]. Thus the descriptor in this approach describes image features from certain, fixed regions of the images. The resulting data is then concatenated and forms a single, compound feature vector that describes the complete image but retains local sensitivity according to the processed image regions.

Spatial Pyramid Kernel

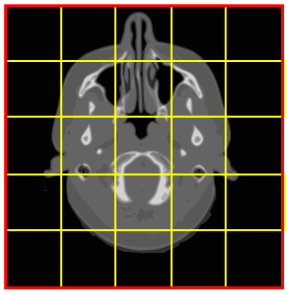
Since retrieving similar slices from volume sets is rather akin to scene classification than to object recognition and due to the more complex distance measure in case of a bag of features approach, it was decided to build a single feature vector for complete images. However as stated above, the price for this decision was the loss of spatial information if the descriptor completely ignored the spacial distribution of the pixel data. In order to keep track of the spatial distribution as well, a modified spatial pyramid kernel was applied.

This decision offers a compromise between a single global descriptor and many local descriptors. By employing a fixed spatial separation into sub regions, the features extracted from those sub regions also do not need to be handled as several independent vectors. If the separation into sub volumes is deterministic and the same for all images, there is the possibility to concatenate the features from the sub regions into one large feature vector. The advantage of this approach is that the improved distance computation distance computation during the $k$nn search compared to a multi instance feature representation.

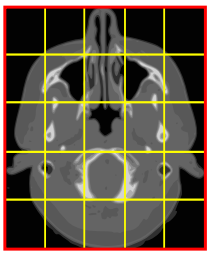
The original implementation of the spatial pyramid kernel extracts features from a region covering the complete image and then divides the image into four disjoint, equally-sized subregions as it is known from quad-trees [44, 127]. For each of these subregions, the extraction and divide steps are executed recursively until a certain level is reached. The resulting features are then weighted and serialized into a single feature vector. Obviously, the resulting



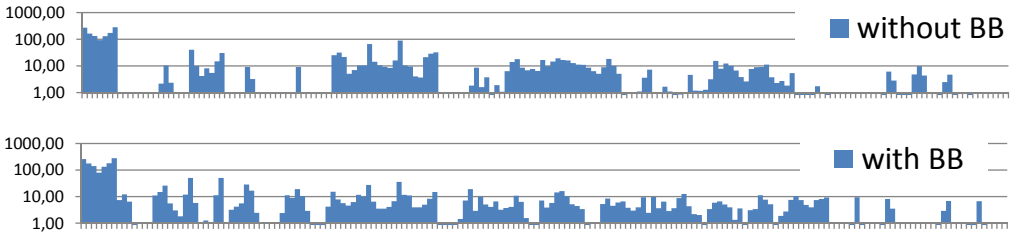
(a) Original pyramid kernel using 21 regions.



(b) Modified pyramid kernel using 26 regions.



(c) Modified pyramid kernel with bounding box applied.



(d) PHoG descriptor for Figures4.3(b) (complete image) and 4.3(c) (ROI only). The plots display the strongly varying feature values of the given images in log scale.

Figure 4.3: Modified pyramid kernels and impact of ROI detection on feature vectors.

dimensionality grows with more than $O(4^n)$ with n denoting the level of the subregions.

For the current scenario, this approach has two major drawbacks: First, to achieve a high resolution of the spatial distribution, a comparatively large number of levels would be required which is leading to a very high dimensionality of the resulting feature vector. Second, as mentioned above, splitting the image region into four equally-sized subregions requires a split in the middle of the x- and y-axis which is quite disadvantageous in the case of CT scans because patients are usually not absolutely centered upon the image. Thus, the first split is performed in the middle of the image but the split axis is hardly centered upon the center of the patient's body as the patient's position is varying between different scans. Therefore, significant

body structures like the spinal column are often either to the right or to the left side of the split which leads to strongly varying feature vectors for similar but not slightly shifted patients.

These issues lead to the modification of the spatial pyramid kernel in a way that the image region is split into 25 disjoint, equi-sized regions instead of only four regions as can be seen in Figure 4.3. This procedure has two advantages: The first advantage is that the spatial information gathered from the sub regions is much more robust against varying positions of the patient. The second advantage is that processing only one level of the recursion is significantly reducing the dimensionality of the resulting feature vector. A reason for the multiple levels in the original spatial pyramid kernel is robustness against scaling and object positioning. However, in this application there are no strong differences in the object position and scaling. Thus, the descriptors employs only two region levels. To compensate any remaining scaling and transversal effects, the following preprocessing step is applied.

Detecting Region of Interest

Partitioning a complete image into 5x5 disjoint regions can lead to image regions that are either almost empty (for example in the edges of the image, as can be seen in Figure 4.3(d)) or mostly occupied by the shape of the table on which the patient is lying. As these regions implicitly reduce the descriptiveness of the resulting feature vector, a region of interest (ROI) detection is employed to detect the bounding box around the patient's body.

Each border of a ROI is detected by scanning the image in a sweep line manner and and keeping track of the following variables: i, the index of the currently processed row/column, c_P, the amount of consecutive pixels larger than the defined threshold of $-600\,\mathrm{HU}$ and c_L: the amount of consecutive rows/columns that are regarded as border candidates.

In order to find the top border of the ROI within an image, the algorithm starts at the top of the image ($i = 0$) and scans the pixels of this line. If a

pixel has a value above $-600\,\mathrm{HU}$, c_P is raised by 1, otherwise, c_P is reset to 0.

As soon as $c_P > 100$ (indicating that 100 consecutive pixels had a value greater than the threshold), the algorithm decides that the current line is a border candidate. In that case, c_L is raised by 1 and the algorithm proceeds with the next line. If all pixels of a line are scanned without c_P exceeding the threshold, the line is not a border candidate, and c_L is reset to 0 and the next line is processed. As soon as $c_L > 20$ (which means that 20 consecutive border candidates were found), the algorithm stops and returns $\max(0, i - 20)$ as the top border of the ROI.

The above steps are repeated for each side of the image. The resulting borders enclose the ROI of the image which can then be used in the following feature extraction steps. Since the borders on each side do not have to display the same width, the method centers the patient. Furthermore, the expansion of the body on the image is unified and thus, the body regions of the 25 patches can be much better compared among scans displaying different patients.

Image Features

As mentioned before, Haralick texture features [63] are used as the first image patch representation in this method. For the proposed method, all 13 Haralick features for five different distance values $(1, 3, 5, 7, 11)$ are computed. This computation is done for each subregion of the spatial pyramid kernel defined above (including level 0, representing the complete image). After extracting the features for all subregions, the feature values of a level are serialized and normalized. This is done to achieve an equal weighting of the different levels of the spatial pyramid kernel. The resulting feature vector finally comprises $26 \cdot 13 \cdot 5 = 1690$ features. As stated in [63], some of the features are highly correlated. To remove the redundancies and correlations, a principal component analysis (PCA) is applied on the features.

The second image patch representation is a histogram of oriented gradients:

Before extracting gradient features from an ROI, some preprocessing steps have to be applied. This includes the application of a Gaussian blur with a radius of 1 px to remove noise, followed by the extraction of important edges P_{edge} from the image by applying the Canny operator[28] C. Important edges are defined by all locations where the Canny operator computes values greater than zero (4.1). In the next step, the gradient's angle $G(x, y)$ is computed at the location (x, y) of important edge pixels (4.2).

$$P_{\text{edge}} = \{(x, y) | C(x, y) > 0\} \tag{4.1}$$

$$G(x, y) = \arctan \frac{\partial y}{\partial x} \; ; \quad \text{where } (x, y) \in P_{\text{edge}} \tag{4.2}$$

Afterwards, a 7 bin histogram is built for all $G(x, y)$ within the ROI. The resulting histograms are serialized and normalized just as the Haralick features before. Finally, this process creates a feature vector with $(1 + 5 \cdot 5) \cdot 7 = 182$ dimensions. This representation is referred to as PHoG (pyramid histograms of oriented gradients) in the rest of the section. Even though the dimensionality of this representation is much lower compared to the Haralick representation, the dimensionality is still very large so that a PCA is also applied to this representation.

4.4.3 Localization

The objective of this task is to receive the slice descriptor presented in the previous section and predict its most likely position in the standard model. To solve this task, an instance-based regression model is employed which is based on a training set consisting of the CT slices from a number of patients. Each example slice x_i taken from the scan $s(x_i)$ is described by l feature representations $(x_{i,1}, .., x_{i,l}) \in R_1 \times .. \times R_l$ and is labeled with its relative position in the scan $y_i \in [0..1]$. From a machine learning point of view, localization can be regarded as a regression task. However, there are two important differences in the object representation that prevent ordinary regression techniques from offering accurate results in this scenario: The first difference is that it is needed to rely on all of the l object representations and

thus the learner should be suitable for multi-modal problems. The second difference is the heterogeneity of the example set. Since the example objects are combinations of various CT scans, the training set cannot be considered to be drawn from the same statistical distribution. Instead, the images within the same scan are usually more similar to each other than to the images of other scans having a comparable position. The proposed localization method is thus designed to consider both aspects to allow a good positioning accuracy.

The basic approach behind this method is to find in the training set the k-nearest neighbors to the target slice t and examine their positional labels. The final prediction is then derived by aggregating the labels of these neighbors. After having received the k-nearest neighbor positions, the mean value of the position labels is employed as target value. Hereby, the Euclidean distance is used to describe the difference between objects which is a standard metric in similarity search and instance-based learning tasks.

Having training examples taken from several similar but not identical distributions, i.e. various CT scans, sometimes causes problems for prediction. Basically two reasons for the similarity between the target slice and an example slice in the training set can be distinguished: The first is, that the positions of the slices in the scan are quite similar. The second is, that slices which are contained in complete scans, are generally quite similar in consecutive regions. While the first reason is the phenomenon the method is based on (high resolution scans), the second reason can seriously distort the prediction result by the following effect. Due to the general similarity between nearby images of a single scan the $k$nn search preferably takes examples from the most similar scan instead of taking the examples from various scans with comparable positions. To circumvent this effect the classical $k$nn search is modified in the following way: First the most similar CT slice is searched within each scan. From this set, the k slices having the smallest distance in the underlying feature space to the target slice are computed. By taking at most one slice from each scan, it is avoided that the localization process is overly dependent on a single scan but derives its results from k different scans.

As mentioned before, the method is based on different feature representations and thus, the learner has to be extended to base its prediction on a mixture of all input spaces. This has to be done as there is the problem that some feature representations are less suited for certain regions of the body, while they provide excellent results in certain other regions. For example, PHoG descriptors are well-suited for areas with a rich bone structure. However, they are less descriptive in the abdomen area. To integrate this diversity, this method bases its decision on the feature representation that most probably offers the best prediction quality for the current input image. In other words, the position of the current input slice is predicted in each of the available feature representations and afterwards the reliability / coherency of the prediction is predicted in each representation. To measure the degree of coherence, the variance of the positions within the k-nearest neighbors in each representation is calculated. If the variance is large, the k-nearest neighbors are placed in different parts of the body and thus the given representation does not yield a consistent statement about the position of the slice. On the other hand, if the labels of the k-nearest neighbors are placed in similar positions, the variance is small and the given representation offers a coherent prediction. In this case, the prediction corresponding to the representation providing the smallest positional variance for a given target slice t is chosen as a final result.

4.5 Radial Descriptor

4.5.1 Introduction

In Section 4.4 the Multi Represented-Descriptor (MR-Descriptor) was introduced. Even though the use of the MR-Descriptor resulted in an average error of a bit more than 3 cm, there are still body regions where the localization is larger than 10 cm as can be seen in Figure 4.8(a) (p. 50).

Motivated by these findings, there was the decision between either using

additional feature descriptions and machine learning techniques or to pay more attention to the anatomical structure of the human body,modify the descriptor accordingly and avoid additional machine learning steps in the processing chain. Adding additional feature descriptors combined with sophisticated machine learning techniques would have added some more additional degrees of freedom to the problem. The latter choice tries to make use of human perception and the question, how a radiologist / physician perceives and distinguishes different body regions. Discussions with a radiologist lead to the modification of the descriptor to adjust more to the human skeleton structure and the body shape itself.

The aim of the modification was to reach a possibly even smaller error average but even more important a smaller error variance and in general a reduced error rate in the shoulder and abdomen as these regions posed the largest errors in the former solution. Same as in section 4.4, the query should be represented by a single slice only and the use of land mark detectors should be avoided.

4.5.2 Feature Extraction

As stated before, the aim was to modify the feature descriptor to take the human skeleton and body shape into account than in the previous approach. To achieve this goal, the descriptor was refactored and thus improved.

Improvements to the MR-Descriptor

To improve the descriptor, the shape of the descriptor was changed to a radial representation inspired by the works of Belongie *et al* [13]. The main reason was that the rectangular shape of the bounding box often produced very sparse or even empty boxes in the corner of the bounding box. Another reason was the fact that by using a radial descriptor model instead of a box model it was able to produce a more fine grained model of the head and the chest and thus also to better distinguish between chest and abdomen. This is mainly

motivated by the fact that the rib cage is modeled much more accurate so that the presence of bones realized a significant element compared to other body regions.

The second change to the descriptor was the strategy of finding the region of interest (ROI) itself. The former strategy of finding the bounding box sometimes resulted boxes that were larger than required. Especially a better adaption with respect to the table on which the patient is lying was desired. Instead of the sweep line approach described in section 4.4.2 (p. 30), a particle cluster based approach was applied in this case.

The next change was to modify the strategy to create sub regions within the region of interest. In this step, the rectangular sub regions of the ROI were replaced by a sector / shell model. This has both the advantage that ribcage, head and shape of the body can be modeled much better and also that the subdivision strategy fits perfectly with the radial nature of the ROI. In contrast to the previous approach, the features are now only extracted from the sub regions - in the former approach, the features were extracted on the complete ROI combined with the features from multiple sub regions.

After changing the shape and division strategy, the features themselves were evaluated and adapted and adjusted. The major concern against gradients and Haralick features was the very noisy nature of CT images. Compared to plain old photography, CT images show a very noisy picture. Even with the application of blur filters and noise reduction, there remains quite some noise. This lead to the decision to either apply complex reconstruction techniques or to employ different features. While reconstruction often requires spatial or semantic information that was not existent with just one query slice, the decision was made to evaluate different features before trying to apply context-free reconstruction techniques. Thus, the gradient histograms and Haralick features were replaced by histograms of gray values which corresponds to measuring the distribution of certain tissue types (soft tissue, air, water, bones, etc) depending on the chosen HU range that should be taken into account. The combination of different HU ranges will be called compound radial image descriptor in the following.

Image Preprocessing

The process of generating the compound radial image descriptor consists of the following three steps: unifying the image resolutions, extracting the patient's body and combining the two image descriptors to a single radial descriptor.

Unifying Image Resolution: The resolution of a CT image is determined by the setting of the according recording device and may vary depending on several external factors. Thus it is needed to scale the image I to a common resolution (1.5 px/mm) to obtain scale invariance between different scans. The resulting image is defined as I_S.

Extracting the Body Region: In order to separate the body from the rest of the image, a compound region detection is performed on the scaled CT slice I_S: A compound region is defined as an area of pixels which is enclosed by a contour of pixels with $p(x, y) > \tau$. $p(x, y)$ defines the HU value of a pixel at the position (x, y) and τ defines the according threshold (-500 HU in this case). The resulting compound regions are extracted by starting a contour tracing algorithm from each pixel in I_S with $p(x, y) > \tau$. The applied algorithm is implemented by using the *analyze particles* function of ImageJ [1] which adapts the well known contour tracing algorithm of Pavlidis [109]. Afterwards the bounding box of the largest compound region defines the ROI represented by the area of the patient's body on the image I_S (cf. Figure 4.4(b)). I_S is then cropped to this bounding box, building the image I_{ROI}.

Feature Extraction

Model Creation: Next, a radial sector/shell model is created from which the two image descriptors will be extracted subsequently. The model is illustrated in Figure 4.4(c). The first descriptor focuses on dense structures (bones) while the second descriptor concentrates on soft tissues (like organs etc.). Both descriptors proposed in this section are represented by the circumcircle of I_{ROI} with radius r. In order to form the descriptors, the circular area is

divided into n_y shells and n_x sectors resulting in $n_x \cdot n_y = i$ bins. The size of such a sector is defined by $\phi = \frac{2\pi}{n_x}$, the size of a shell is defined by $\frac{r}{n_y}$. For each bin i, both the number of pixels of interest (POI) p_i and the number of other pixels (NPOI) n_i is calculated. A POI is defined as a pixel with $p(x, y) \geq \psi_1$ or $p(x, y) \leq \psi_2$ depending on the type of descriptor, which are described subsequently. The values of bins $\notin I_{ROI}$ are set to a penalty value of -0.25 to achieve a larger difference between descriptors from regions with different aspect ratios. Thus the value v_i of a bin i is defined as

$$v_i = \begin{cases} -0.25 & \text{if bin } i \notin I_{ROI} \\ \frac{p_i}{p_i + n_i} & \text{else.} \end{cases} \tag{4.3}$$

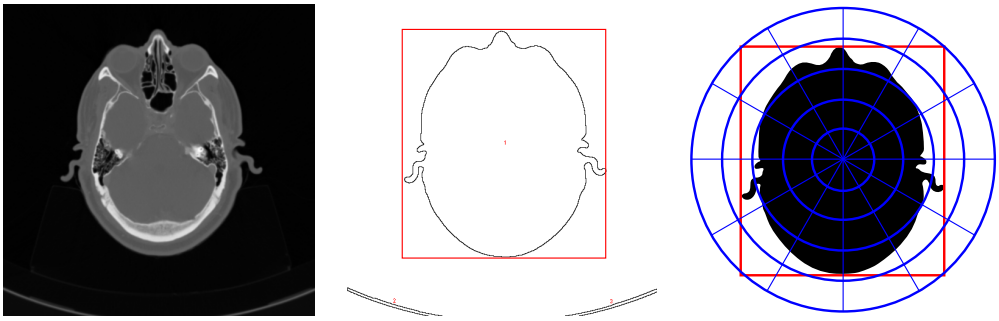
An alternative approach would have been to model the radial descriptor not by a circumcircle but to fit the ROI into an ellipse that fits the ROI better than a circle. Yet, this leads to the fact that the information about the aspect ration is lost or at least weakened. To compensate this lack of information, an additional dimension could have been added to the vector. This would have raised the issue of determining a proper weighting for this dimension compared to the other dimensions. Comparing the two possibilities, the principle of Occam's razor was applied and the former possibility of setting empty cells to a default value was applied.

A visualization of the model is illustrated in Figure 4.4.

Descriptor 1: Bone structure: The first part of the descriptor takes the form and location of bones within the body into account as the skeletal structure of the human body plays a big role in human classification of the body position. Thus, the threshold is set to $\psi_1 = 300\,\text{HU}$ so that the amopunt of all POIs is defined by

$$p_i = |\{p(x, y) \in I_{ROI} | p(x, y) \geq \psi_1\}|. \tag{4.4}$$

Regarding the spatial distribution of the bones (e.g. cranial bone, chest, shoulder joints, hip joints), it was observed that the outer shells of the descriptor are more relevant than the shells in the middle of the image as



(a) (b) (c)

Figure 4.4: Visualization of the feature extraction process for a neck scan image (a): the image is rescaled and the body (in this case the head) is detected (b) and approximated by a bounding box. Afterwards the sector/shell model is created (c) from which the features are extracted.

there where hardly any bones detected. So, each bin of the descriptor is weighted w.r.t. the shell index. In particular, for each bin i the value of p_i is weighted with the squared index of its shell:

$$p_i = p_i \cdot shell(i)^2 \quad ; \quad i \in [1, n_x] \tag{4.5}$$

where $shell(i)$ denotes the index of the shell containing the area of bin i. An extensive evaluation of the parameters proofed the best results with $n_x = 24$ and $n_y = 11$.

Descriptor 2: Distribution of soft tissue: Some areas in the human body like the abdomen display a comparatively small amount of dense structures. Therefore, a descriptor denoting the location and arrangement of soft tissues is created. The threshold for this descriptor is set to $\psi_2 = -500\,\mathrm{HU}$. Thus, the amount of POIs in this case is defined by

$$p_i = |\,\{p(x,y) \in I_{ROI} | p(x,y) \leq \psi_2\}\,|. \tag{4.6}$$

For this descriptor, another parameter evaluation proved the best results with $n_x = 18$ and $n_y = 8$. In contrast to the previous descriptor, no significant relation between bins and their shell indices could be detected so that all bins were weighted equally.

Same as in the case of the descriptor for the bone structures, variations of up to ± 4 of the parameter values of n_x and n_y do not have a large impact on the results. The complete settings for both descriptors can be seen in Table 4.2.

Table 4.2: Parameter setting for both descriptors.

Type	ψ	Sectors n_x	Shells n_y	Angle ϕ	Weighting	Bins
Bones	$\geq 300\,\mathrm{HU}$	24	11	15°	quadratic	240
Soft	$\leq -500\,\mathrm{HU}$	18	8	20°	equal	144

Combination and Dimensionality Reduction: Finally, both descriptors are concatenated to a single feature vector q. An additional step is the application of a principal component analysis to reduce the dimensionality of the feature vectors. In the experiments, the dimensionality could be reduced down to 50 dimension without loosing too much accuracy.

4.5.3 Localization

Same es in Section 4.4, the task of the prediction method is to localize (a.k.a. register) the query vector q representing the query slice with unknown position q_z to a value $z \in [0, 1]$ in the standardized height model. As the localization/prediction method proposed so far was still convenient, there was no change in the localization method. Thus the two level $k$nn search described in Section 4.4.3 (p. 32) was retained.

The only difference to the previous approach is the distance metric: In contrast to Section 4.4, the cosine distance measure (cf. (4.14)) is now used for distance computations instead of the Euclidean distance as it performed slightly better.

4.6 3D Detection

4.6.1 Introduction

After the successful improvements from the MR-Descriptor in Section 4.4 to the combined radial descriptor in Section 4.5 it should be evaluated if the technique could also be extended to small volume scans. Even though [43] has shown an error rate of less than 5 cm in case of query volumes with a size of 44 mm, this approach still has the drawback that it does not perform well with smaller volumes. Thus, it should be evaluated, if the gap between one-slice-queries and small sub volumes could also be bridged by the use of the combined radial descriptor. Thereby, combinations of image descriptors and weighted combinations of spatially neighboring images as well as instance based regression should be used to combine the information of adjacent images to reduce the localization error even further.

Even though this seems very related to [43], there remains a significant difference: Feulner *et al* clearly aim at registering sub volumes whereas this approach just uses the information of several adjacent slices in order to form a single image feature vector which should be more robust compared to each other feature vector that is based on a single slice only.

Given the assumption that several adjacent query slices are present, it could be argued that the feature extraction process should be changed so that each voxel[1] can make use of the information of its 3D neighborhood. Such a feature vector could of course contain richer and possibly more reliable information compared to a descriptor that is composed of several 2D descriptors because each voxel can directly make use of the 3D neighborhood. If feature vectors are extracted from 2D images and then aggregated again, the relative information for each pixel is less than in the former case. Yet, the aim of aggregating several nearby feature vectors is to make the single feature vector more robust in case that one or more slices show some distortions or artifacts that would

[1] voxel: A 3D pixel which is described by an x, y and z coordinate plus an HU value describing the attenuation of radiation in that location.

be normalized and attenuated by adjacent slices.

But before trying to recreate a completely new feature descriptor for this case, it should first be attempted if the simpler approach of combining the already proven features would result in any improvement of the error and stability at all. The findings of this attempt and the results will be discussed in the following.

4.6.2 3D Features

As stated before, the approach proposed in this section addresses the localization issue by considering m preceding and m succeeding CT slices of the current query slice. As this process is executed as a post processing step after the creation of feature vectors from single 2D slices, it can be seen as an extension to the 2D method as the flexibility to use just a single CT slice still remains.

For this task, a new modified feature vector FV_i^{3D} is formed by calculating the weighted sum of the succeeding m and preceding m feature vectors. If the current vector is not preceded or succeeded by m vectors, only the existing vectors are used. Let FV_i denote the i-th feature vector in the sorted list of vectors for a single CT volume comprising n slices. Then FV_i^{3D} is calculated as follows:

$$FV_i^{3D} = \sum_{k=\max(0,i-m)}^{\min(i+m,n)} f(|k-i|) \cdot FV_k, \tag{4.7}$$

where $f(x)$ is one of the following weighting functions:

$$f_{\text{inverse}}(x) = \frac{1}{x+1} \tag{4.8}$$

$$f_{\text{sigmoid}}(x) = \frac{2}{1+e^{0.3x}} \tag{4.9}$$

$$f_{\text{polynome}}(x) = -\frac{x^3}{(m+0.1)^3} + 1 \tag{4.10}$$

$$f_{\text{linear}}(x) = -\frac{x}{m+0.5} + 1 \tag{4.11}$$

$$f_{\text{inverse-squared}}(x) = \frac{1}{x^2} \tag{4.12}$$

$$\begin{aligned} g(x) &= \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \\ f_{\text{Gaussian}}(x) &= \frac{g(x)}{g(0)} \qquad ;\mu = 0; \sigma = \frac{m}{2} \end{aligned} \tag{4.13}$$

All these functions have in common that $f(0) = 1$ and that the value of $f(x)$ decreases with increasing x, so that the weight decreases with increasing distance to the source feature vector FV_i.

4.6.3 Prediction

Same as in Section 4.4 and Section 4.5, the task is to map the feature vector FV_i of a CT image with unknown position to a value in the standardized height model in the domain $[0; 1]$. The prediction itself remains the two-stage k-nn search which is described in Section 4.4.3 (p. 32).

The distance computation is executed by employing the Cosine distance (4.14) instead of the Euclidean distance (4.15) as it performs usually better on high dimensional feature vectors than the Euclidean distance [114, 131].

$$\text{dist}_{\text{cosinus}}(q,p) = 1 - \frac{\sum_{j=1}^{d} q_j \cdot p_j}{\|q\| \cdot \|p\|} \tag{4.14}$$

$$\text{dist}_{\text{euclidean}}(q,p) = \sqrt{\sum_{j=1}^{d} (q_i - p_i)^2} \tag{4.15}$$

4.7 Evaluation

4.7.1 Data Set

The data set used for this thesis comprises 97 CT volume scans (38 neck scans, 59 thorax scans) recorded from 74 patients (43 male, 31 female) of different age, resulting in a total number of 53 437 DICOM images using more than 26 GB disk space. All scans are composed of multiple images which are represented in 16 bit Hounsfield Units (HU) and have a resolution of 512×512 pixels.

During the initial setup of the data set, it was ensured, that each patient contributed at most one head and/or one thorax scan to the data set in order to avoid adding near duplicate scans. Otherwise it could happen that one patient contributes several thorax scans to the data set. If these scans are taken within a comparatively short time, it can be assumed that a nearest neighbor search will favor scans of the same patient again for localization. While this might be welcome in the real application, it is not advisable in the test environment as these patients will most likely always produce better results than patients who just contributed one scan per body region.

All these scans cover the complete area between the top of the head up to the end of the coccyx. It should be mentioned that the data set shows multiple kinds of heterogeneity as the data set represents a real world data set that was recorded under real conditions. Also, the scans were recorded with 5 different Siemens CT scanners and different settings so that a variety within the data is provided as ground truth. Moreover, the transversal resolution (z-axis) varies between 66 and 1 700 slices per scan. The resolutions along the x- and y-axis are varying in the range of about 1.09 – 1.76 px/mm for thorax scans and about 1.34 – 2.3 px/mm for scans of the head. Besides those technical diversities, there are of course also challenges posed by the use of contrast media, medical devices/artifacts like cables, cardiac pacemakers or simply metallic dental implants (which obviously cause major disturbances in the images) and of course the differing shapes of the patients' bodies.

In the following, the evaluation of the methods described in Section 4.4, Section 4.5 and Section 4.6 are explained. For the evaluation of the proposed methods, it was decided to use a leave-one-scan-out validation. In case of a classical leave-one-out validation, only a single slice would be removed from the test set and used as a query. As the used slice is very similar to the adjacent slices of the same scan, this evaluation would over fit to the scan from which the slice originates. Therefore, the evaluation is done using a leave-one-scan-out. In this case, a complete volume scan is first removed from the data set. Afterwards each slice of this scan is used as a query against the remaining data set. The average of all single errors is then returned as the total average mean error. As no patient contributed two thorax or neck scans, it is guaranteed that the query does not over fit to the patient to which the query slice belongs.

The quality of the methods is measured by the distance between the true (annotated) position of the query slice and the result of the localization. The difference is denoted as the error of a query. In addition to the average error and the error histogram, the cumulative distribution function (CDF) for the errors is computed as well. This indicates the probability that the error stays below x cm: $P(\text{error} < x\,\text{cm})$. This is mainly done for two reasons: First, it is then possible to compare to the results given in the work of Feulner *et al* [43] who demonstrate the accuracy of their work by the values of the according CDFs. On the other hand, the CDF has the advantage that is visualizes the ratio of large errors in the overall experiments. This is important, because it is preferable to lower the ratio of larger errors compared to optimizing the ratio of small errors or just the mean error.

In order to provide a comprehensible error measure, the values of the standard $[0, 1]$ model are multiplied by 180 so that the complete model represents an average western European male. The localization results are usually measured in a 180 bin error histogram so that each bin corresponds to a body region with a width of 1 cm. As the data set only covers regions from the top of the head up to the coccyx, only the first half of the error histogram is shown. The rest of the histogram is zero as there is no data for

this body region.

4.7.2 Annotation

As mentioned before, one cannot rely on the information of the DICOM header for obtaining the position of a slice. Additionally, different scans are highly varying with respect to resolution and patient body size. Thus, two computer scientists annotated the data above independently by hand using the annotation tool shown in Figure 4.5.

Thereby, even a small mistake in the annotation tool of about 5 pixels leads to an annotation error of up to 1.5 cm. Thus, a small positioning error of the data remains in the ground truth which limits the accuracy of the method on this data set. Another issue is that the patients in the data set have their arms raised above their heads in case of thorax scans, whereas the annotation tool only provides a skeleton with the arms beside the thorax (cf. Figure 4.5).

This annotation alone might not fully respect differing proportions of the human body between patients as the template skeleton in Figure 4.5 is of fixed size. To address this problem, the annotation process was extended to an additional annotation of stable landmarks[2] that were defined by a medical expert. The advantage of these landmarks is that an annotator need not be a medical expert to be able to identify the according positions in a CT scan.

In the next step, the annotation using the annotation tool and the landmark information were combined. Each slice, for which a landmark was identified has 2 labels: the z-value $\in [0; 1]$ from the manual annotation and the landmark label. In order to map this annotation back into the domain $[0; 1]$ the z-value from the standard annotation tool. Vice versa, each landmark

[2]In neck scans: cranial crista galli, cranial sella turcica, cranial dens axis, caudal plate of cervical vertebrae #4, caudal plate of cervical vertebrae #7.
In thorax scans: cranial sternum, caudal xiphoid process, caudal plate of thoracic vertebrae #12, sacral promontory, caudal os coccygis.

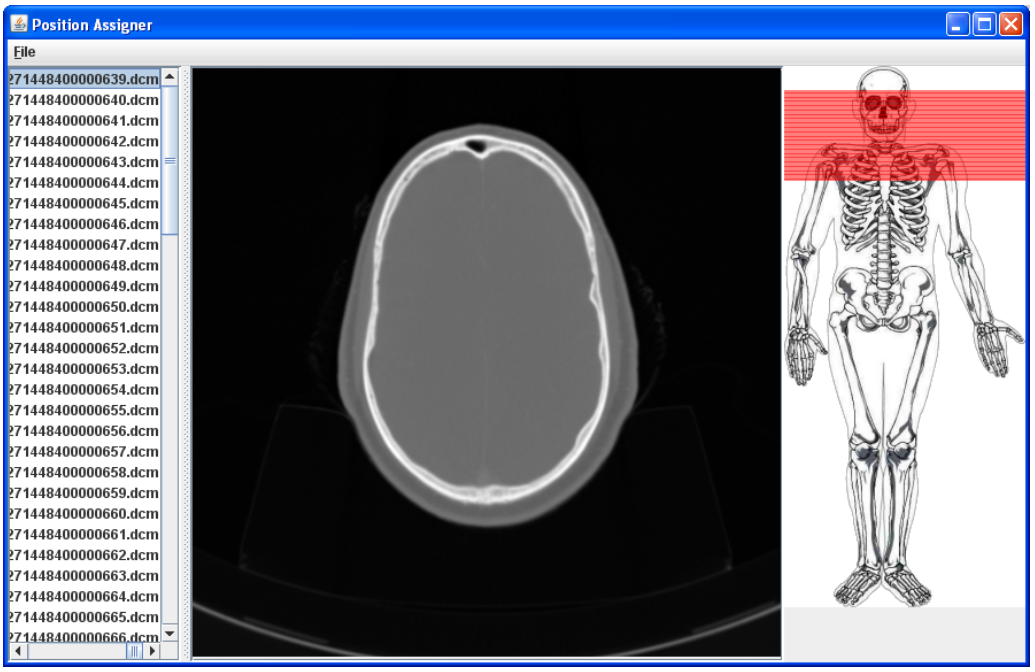


Figure 4.5: Annotation tool

has $1 \ldots n$ different z-values (n being the amount of CT scans).

After the annotation, each land mark is projected to a single z-value by averaging all z-values of this landmark. The position of the slices between succeeding land marks is then interpolated linearly. An illustration of the process is shown in Figure 4.6.

4.7.3 Multi Represented Descriptor

In the following, the evaluation of the Multi Represented (MR) Descriptor (Section 4.4) is explained. First the method is evaluated by using a single representation only. Then the effect of combining different representations is shown and the influence of the parameter k of k-NN regression on the accuracy is discussed. Afterwards a discussion about the impact of reducing the dimensionality of the feature vectors using principal component analysis (PCA) on accuracy and runtime is done. Finally, it is shown that this method

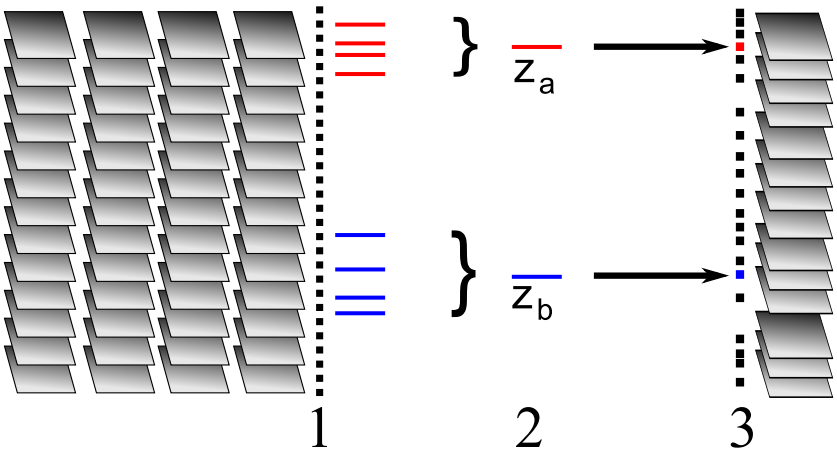


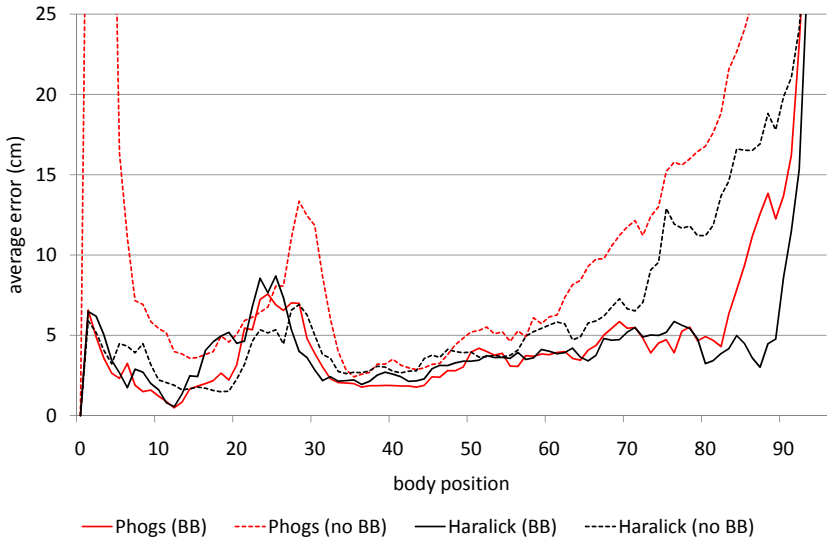
Figure 4.6: 1: 4 volume scans (left) are annotated with 2 different landmark labels (a = red, b = blue). 2: The height value for each label is averaged so that each label has a single height value (z_a, z_b). 3: The height values of the slices annotated with land marks are updated to z_a, z_b. Height values of slices without land mark annotations are interpolated linearly.

performs better than the approach shown in [43] in the case of 44 mm query volumes.

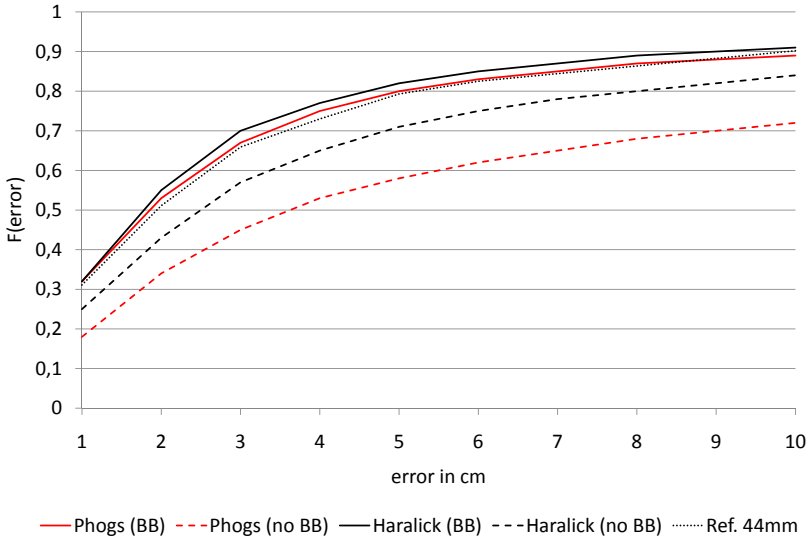
Single Representations

Before explaining the use of multiple representations, single representations should be evaluated first. Figure 4.7(a), shows the error histograms of both PHoG and Haralick features, both with and without an applied bounding box. In spite of achieving quite acceptable error rates in the area of the head (< 3 cm), there are strong errors in the region of the shoulders ($> 5 - 10$ cm) and in the lower thorax (up to 25 cm). Also it seems that both PHoG and Haralick features perform comparable if the bounding box detector was applied. Nevertheless, the prediction error in the area of the shoulder (at about $x = 25$) is larger than 5 cm in both representations.

An interesting observation at this point is, that it cannot be said that the application of a bounding box clearly impacts the prediction result in

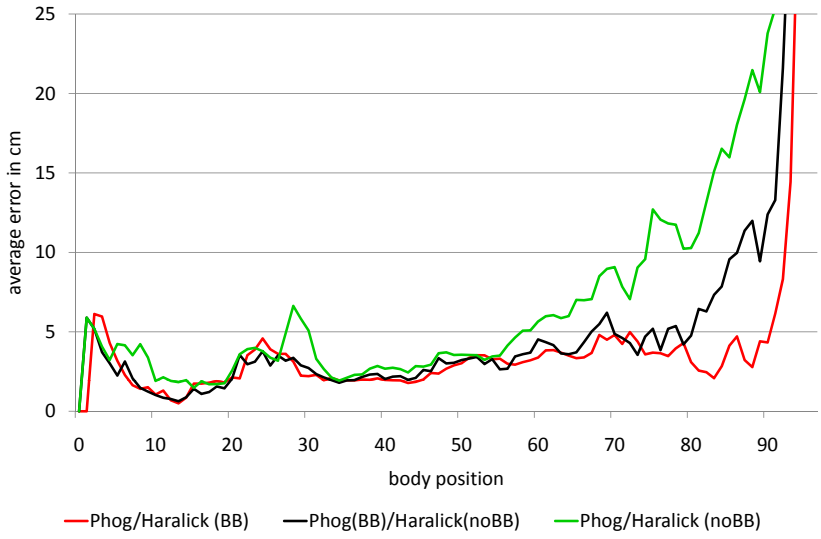


(a) PHoG vs. Haralick separately

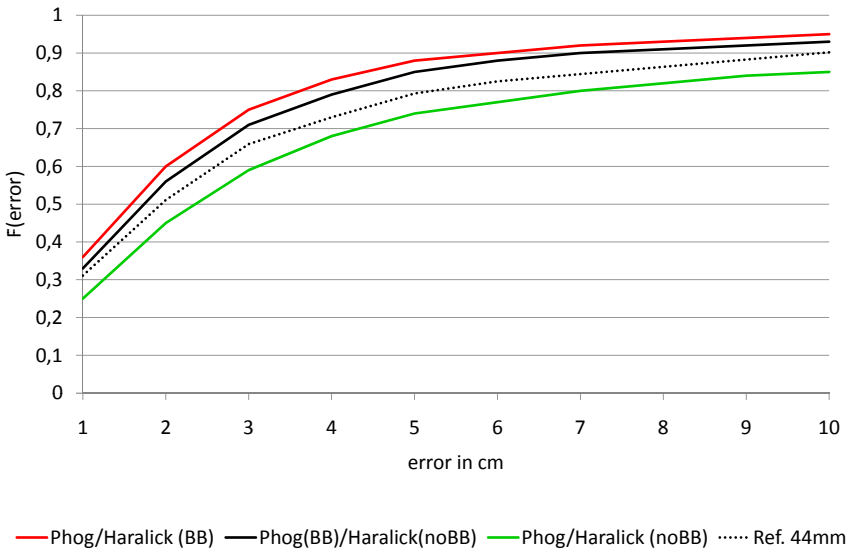


(b) CDF: $F(error \leq x\,\text{cm})$ of the representations in (a).

Figure 4.7: Comparison of single feature representations. (BB) indicates that features were extracted only from the region defined by a bounding box, (no BB)indicates that the whole image is used for the feature extraction. Ref. 44 mm in (b) shows the CDF given in Feulner *et al* [43] for 44 mm volumes. The x-axis in (a) show the body position in cm relative to a body height of 180 cm with 0 indicating the top of the head.

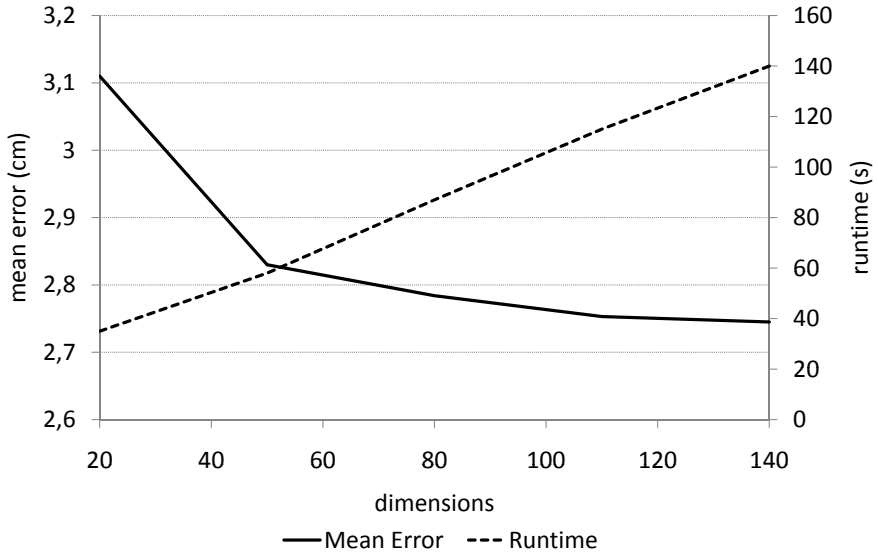


(a) Combined representations

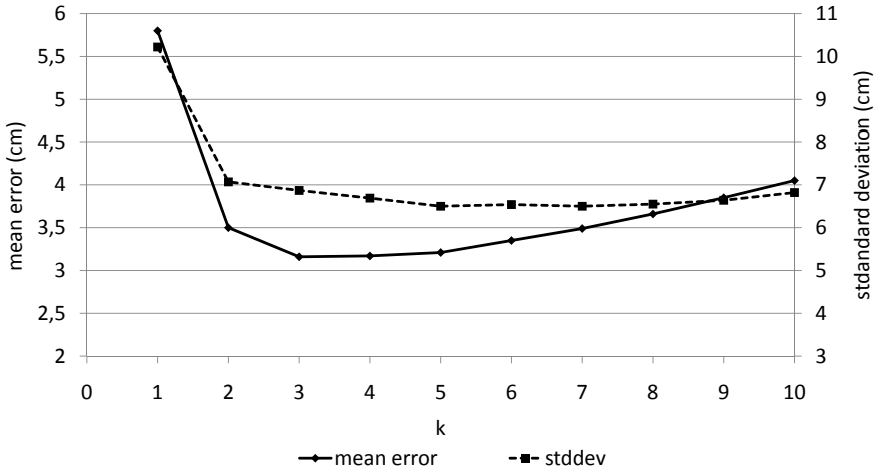


(b) CDF: $F(error \leq x\,\text{cm})$ of the representations in (a).

Figure 4.8: Comparing feature representations: (BB) indicates that features are extracted only from the bounding box. (no BB) indicates that feature extraction is done on the complete image. Ref. 44 mm in (b) shows the CDF used in [43] for 44 mm volumes. The x-axis in (a) shows the body position in cm relative to a body height of 180 cm with 0 indicating the top of the head.



(a) Dimensionality reduction vs. accuracy and runtime



(b) Impact of k to e_{mean} and stadard deviation.

Figure 4.9: Impact of the parameter k and the amount of dimensionality reduction to runtime and e_{mean}.

Figure 4.7(b). In case of the Haralick features for example, the bounding box raises the error in the neck area by about 2 cm but improves the detection in the lower thorax. In case of the PHoG features, the application of the bounding box seems to have positive impact in most of the body areas.

Concluding this setting, it can be said that using one representation alone might not perform very well as this always implies that a higher error rate must be accepted in some other body regions compared to another feature representation. remarkable is the fact that the features where the bounding box was applied are already outperforming the volume set approach shown in [43], which can be seen in Figure 4.7(b).

Combinations of Features

Motivated by this findings, several combinations of the above representations were evaluated. In order to avoid overloading Figure 4.8(a) and 4.8(b), only three combinations are shown which realize the worst, medium and best combinations according to their mean errors. The mean error values of the remaining combinations can be seen in Table 4.3.

Comparing the combination of PHoGs and Haralicks with bounding box from Figure 4.8(a) with the single representations in Figure 4.7(a), it is obvious that the multi-represented approach can enhance the accuracy of the method. Nevertheless, the error in the shoulder region is still comparatively high compared to the neighboring regions.

In [40], this approach was evaluated on a much smaller data set. In this case it was observed that the prediction result could be improved even further if the Haralick texture features were obtained once with and once without the bounding box detector being applied, so that the final descriptor consisted of three feature vectors. The same configuration in this case yielded $e_{\text{mean}} = 3.16\,\text{cm}$ which is also very competitive and only slightly better than the combination of just two representations.

The cumulative distribution function (CDF) describing the error prob-

Table 4.3: Error measures for all tested combinations of representations. (BB) and (noBB) indicate the area from which the features were extracted.

Representation(s)	**Mean Error**	**Std. Dev.**
PHoGs (BB)	4.31 cm	7.39 cm
PHoGs (noBB)	8.84 cm	11.49 cm
Haralick (BB)	3.93 cm	7.07 cm
Haralick (noBB)	5.89 cm	9.17 cm
PHoG/Haralick (BB)	**3.03** cm	**6.02** cm
PHoG/Haralick (noBB)	5.87 cm	10.05 cm
PHoG/Haralick (noBB, BB)	3.75 cm	7.32 cm
PHoG/Haralick (BB, noBB)	3.74 cm	7.55 cm
PHoG/2xHaralick (noBB, BB, noBB)	3.74 cm	**6.87** cm
PHoG/2xHaralick (BB, BB, noBB)	**3.16** cm	7.81 cm
2xPHoG/Haralick (noBB, BB, noBB)	4.12 cm	8.65 cm
2xPHoG/Haralick (noBB, BB, BB)	**3.25** cm	**6.85** cm

ability in Figure 4.8(a) leads to the observation, that the combination of the representations combines the positive characteristics of the single representations. The findings from [40] on a smaller data set also support this theory.

Impact of Parameter k to Accuracy

In the following, the impact of the Parameter k on k-NN regression is discussed. In Figure 4.9(b), the influence of k is measured by using on the best combination evaluated in the section above. It can be seen that the effect of k is small in the range between 3 and 5 whilst e_{mean} begins to increase slightly for $k > 5$. The decreasing performance can be explained by the number of scans in the database that are used in the $k$nn search. Regarding neck scans for example, only 37 scans (38 neck scans excluding the query scan) can even provide true hits. Thus drawing a large number k from 37 scans obviously

increases the error rate at a certain level of k.

Impact of dimensionality reduction to accuracy and runtime

Regarding the dimensionality of the extracted feature vectors (1 690 in case of Haralick, 182 in case of PHoG) it is obvious that the system cannot easily be supported by the use of index structures due to the well-known curse of dimensionality. Even though all experiments were run in main memory, the support of index structures was kept in mind. The computational cost for distance calculations of course also decreases with the reduction of dimensionality - by the price of accuracy, which should be evaluated in the following experiment which is illustrated in Figure 4.9(a). All experiments were run multi-threaded on a 3GHz Intel Xeon 5365 dual quad core with the given run times denoting the overall run time per experiment.

For the reduction of dimensionality, the well-known principal component analysis (PCA) was employed. As expected, the runtime scales almost linearly with increasing number of dimensions. At the same time, the mean error decreases significantly with a rising amount of dimensions until about 50 dimensions are reached. Though later on, there is still a decrease of the mean error, the improvement is almost negligible. Thus 50 dimensions were chosen for all experiments in Figure 4.7, 4.8, 4.9(b) and Table 4.3.

4.7.4 Radial Descriptor

In the following, the evaluation of the Radial Descriptor (Section 4.5) is explained. For comparison reasons, the method proposed in Section 4.4 is referred to as the MR-Descriptor.

Average Error per Body Region

To measure the accuracy and precision of the radial descriptor, the mean error e_{mean} and standard deviation σ of the prediction were measured. Applying

the prediction method introduced in Section 4.5.3, the mean error and the standard deviation could be reduced to $e_{\text{mean}} = 1.76\,\text{cm}$ and $\sigma = 2.74\,\text{cm}$. Compared to the MR-Descriptor, this means a reduction of the mean error and standard deviation by a factor of almost 1.7× and 2.2× respectively (cf. $e_{\text{mean}} = 3.03\,\text{cm}$, $\sigma = 6.02\,\text{cm}$). Table 4.4 illustrates the improvement.

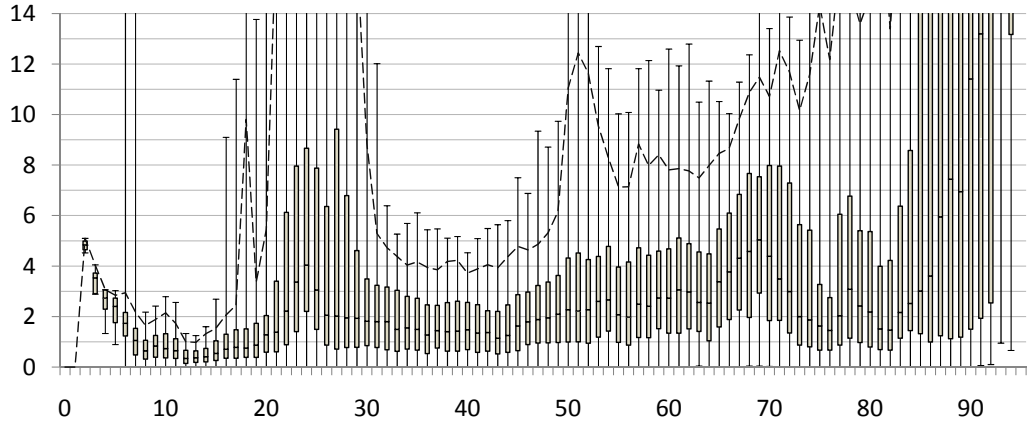
Table 4.4: Error values of the MR-Descriptor compared to the improved Radial Descriptor.

	e_{mean}	σ
MR-Descriptor	3.03 cm	6.02 cm
Radial Descriptor	1.76 cm	2.74 cm

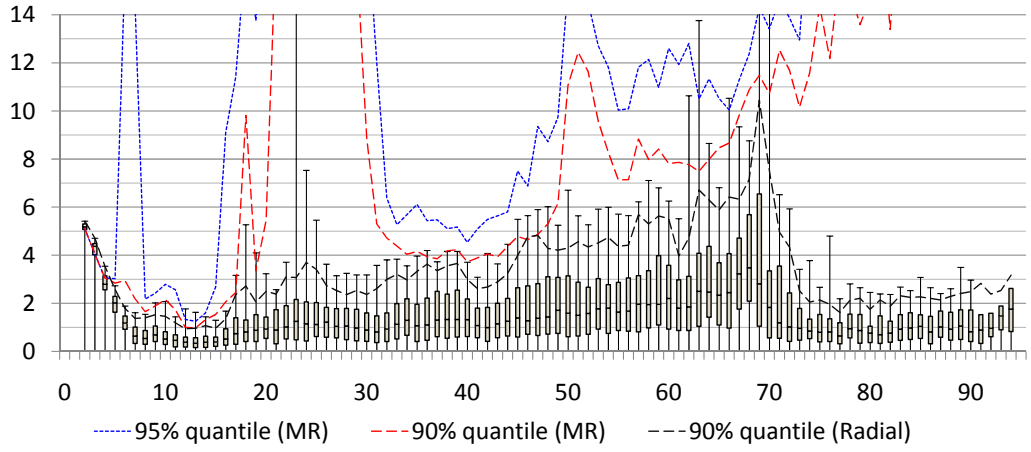
The improvement of σ is visualized by the box plots of Figure 4.10: It can be seen that the upper whiskers in the lower diagram of Figure 4.10 are clearly lower throughout the whole dataset than in the case of the previous MR descriptor. The improvement of e_{mean} is most significant in the areas between 0-10 cm (representing the head) and $> 70\,\text{cm}$ (region of the hips). Also, the decrease of error variance and thus the probability for larger errors can clearly be seen in this figure, especially in the regions between 20 – 30 cm and $> 70\,\text{cm}$.

Cumulative Distribution Function (CDF) of ε:

In order to evaluate the distribution of the error value, the cumulative distribution function (CDF) of ε, $F_E(\varepsilon) = F(E \leq \varepsilon)$ is calculated. Comparing to the MR-Descriptor, an improvement on the complete CDF could be observed as the probability for errors less than 1 cm ($F_E(\leq 1\,\text{cm})$) was raised from 0.3 to 0.5. $F_E(\varepsilon) \geq 0.9$ was hit at $\varepsilon = 3.5\,\text{cm}$ for the radial descriptor and $\varepsilon = 8\,\text{cm}$ for the MR-Descriptor. This means that 90 % of the observed errors were less than $\pm 3.5\,\text{cm}$ while the MR-Descriptor produced errors up to $\pm 8\,\text{cm}$ with the same probability. The complete CDF can be seen in Figure 4.11(a).

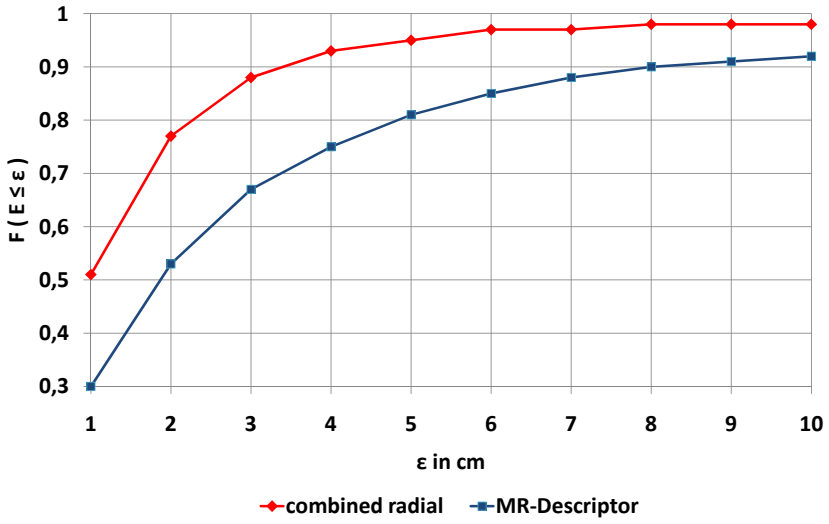


(a) MR-Descriptor

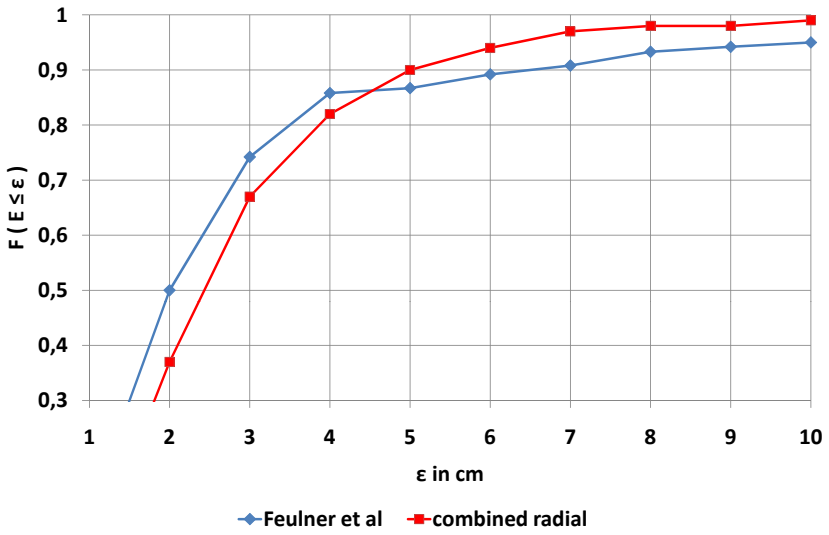


(b) Radial descriptor

Figure 4.10: e_{mean} for both descriptors. The x-axis displays body regions of 1 cm width. The y-axis displays the amount of errors in cm. The dashed black lines indicates the 90 % error quantiles. The red and blue dashed lines in (b) indicate the 90 % and 95 % quantiles from the MR-Descriptor for easier comparison. The box plots show the 0, 25, 50, 75 and 95 % quantiles of the according descriptors.



(a)



(b)

Figure 4.11: CDF of errors from the radial descriptor (red line) compared to the MR-Descriptor (a) on the full data set. (b) compares the radial descriptor (1 slice as query) to [43] (4.4 cm query volumes) (b) on all volumes containing the required landmarks.

Accuracy

Comparing the scatter plots of Figure 4.12 it can clearly be seen that the result of the localization is much more stable in Figure 4.12(b) than in Figure 4.12(a). Especially large errors in the region [0 cm, 20 cm] and > 75 cm could almost be eliminated. The problematic regions [20 cm, 30 cm] (shoulder) and [60 cm, 75 cm] (abdomen) can still be identified as a source for larger errors but the overall amount of errors in these regions was also lowered significantly (cf. CDF in Figure 4.11(a)).

The approach was also compared to the work shown in [43]. As their proposed algorithm is designed for query volumes instead of query slices, the smallest sub volumes (4.4 cm) were chosen for their algorithm and single slice queries for the radial descriptor in order to have a setting which provides the best possible comparison. This means of course, that the radial descriptor is using less slices and thus less information for the retrieval. Also, the approach of Feulner *et al* is based on landmarks, so that the data set had to be reduced to 17 volumes (12 male, 5 female, 6 547 slices) as these were the only volumes on which the according landmarks were detected. In Figure 4.11 the CDFs for both approaches can be compared. It can be seen that the break even for the radial descriptor is at 4.5 cm. In case of $\varepsilon \leq 4.5$ cm, the approach of Feulner *et al* performs better. Nevertheless $\varepsilon \leq 5$ cm are observed with a probability of 0.9, whereas the approach of Feulner *et al* yields the same probability at 6.5 cm. This means that the amount of larger errors is smaller in the case of the radial descriptor even though only a single slice is used as a query.

Processing speed:

Comparing the processing speed, it could be shown that the extraction process is 1.6× faster than in the case of the the MR-Descriptor, if the full MR-Descriptor is used and even 1.9× faster, if the MR-descriptor is reduced by a sampled PCA to achieve an optimal result. In seconds, this means

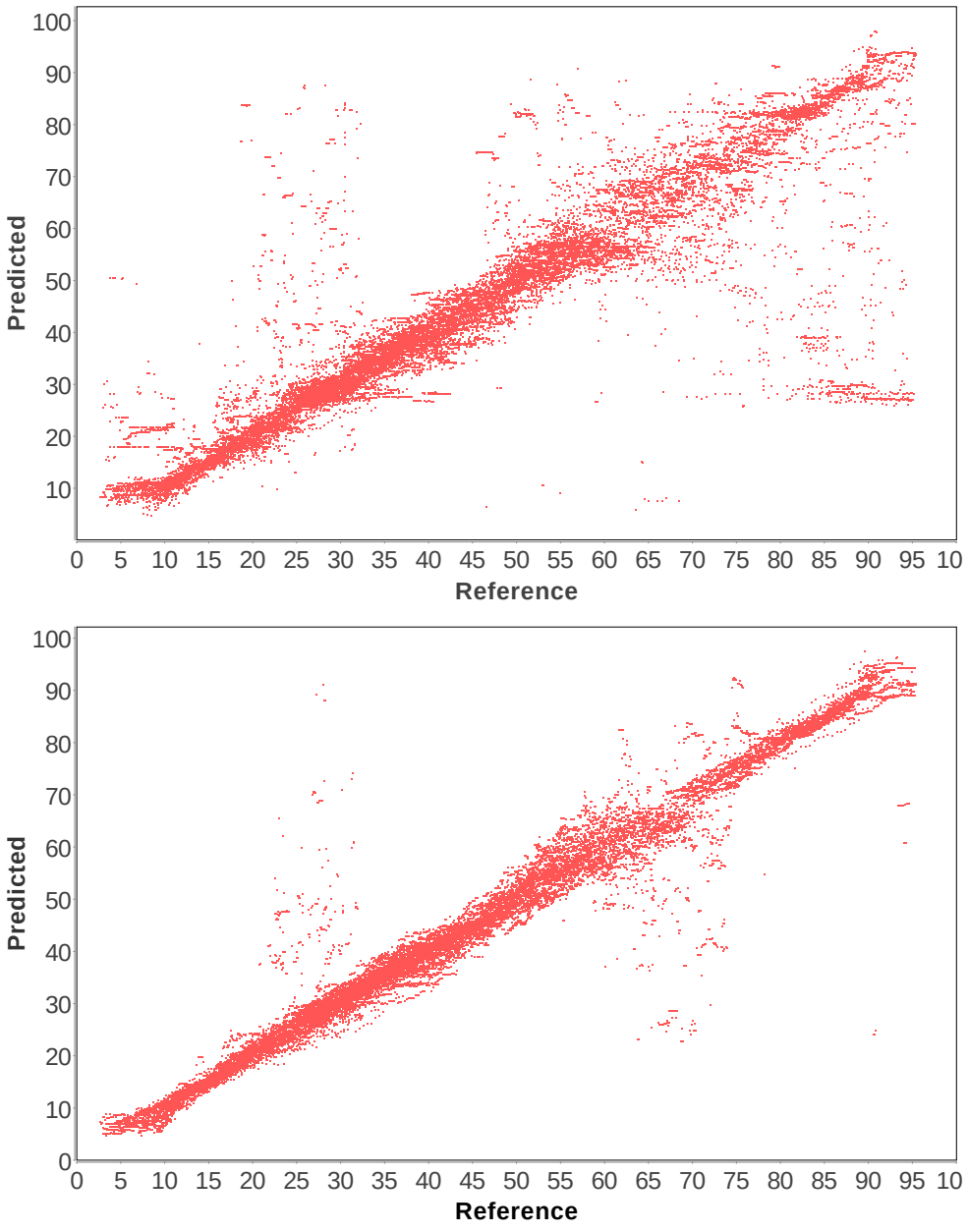


Figure 4.12: Plots comparing the localization quality of the MR-descriptor (top) to the Radial Descriptor (bottom) with each pixel identifying the result of a prediction. The x-component of a pixel denotes the true position of an image, the y-component describes the result of the localization.

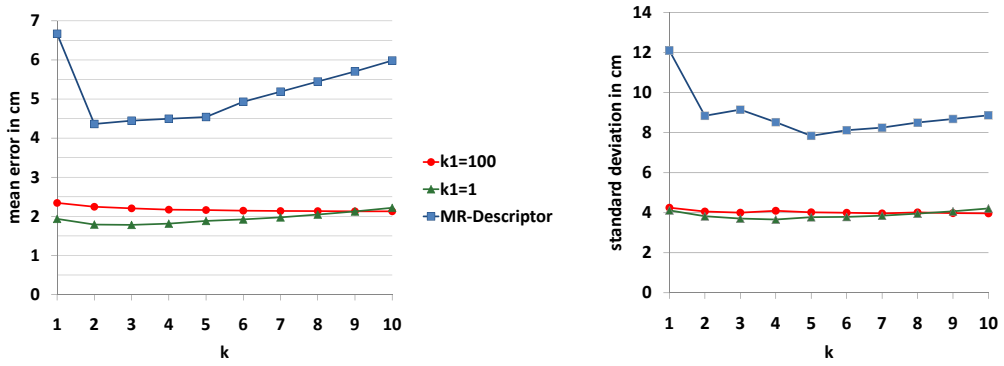


Figure 4.13: Mean error and standard deviation (y-axis) between MR-descriptor and the radial descriptor according to k_2, which is displayed on the x-axis.

that the average time for registering a single unprocessed DICOM image with the radial approach took 0.62 s while the processing of an image with the MR-descriptor required 1.02 s without and 2.03 s with PCA respectively. Testing the impact of the parameters k_1 and k_2 on the runtime did not show a significant difference. This can be explained by the fact that several thousand distance calculations during the search consume much more processing time than the following filtering step.

Impact of k_1, k_2:

Finally, also the impact of k_1 and k_2 on the accuracy and precision was tested. As it can be seen in Figure 4.13, both variables k_1 and k_2 only have a small impact on both the mean error and the standard deviation in this method. In contrary, the MR-Descriptor shows a clear dependence on k. The robustness against k is another advantage as the search process can be simplified accordingly without significant loss of accuracy.

Tilted Slices

Especially in case of head CT scans it is not uncommon that the rotation axis is not paraxial to the x,y-plane but tilted by a certain degree. Reasons for this can for example be that the radiation to which the brain is exposed during a regular head/neck CT scan should be minimized as much as possible. In such cases, the recording plane is tilted to a certain degree θ from the orthogonal plane.

Data Set: As all volume scans of the present data set were recorded as untilted scans, tilted scans had to be simulated by interpolation: Let I_z be a single slice of the volume scan comprising n CT slices, with z denoting the index of the image $z \in [1; n]$. Then the complete volume scan V is first reconstructed from the 2D CT slices.

Afterwards, n tilted slices are computed by projecting the voxel data of V on tilted planes I_z^t. Each I_z^t is located at the same z-position as the corresponding slice $I_z$with a tilt angle θ around the x-axis of the patient's body. Of course, the image qualityof the resulting interpolated slices heavily depends on the resolution along the z-axis of the orinial scan. Higher resoluted scans oviously produce images with more detail, whereas scans with very few slices produce interpolated images with low detail.

After computing the tilted slices I_z^t of a volume scan, the feature extraction process described in Section 4.5.2 is applied without changes, same as the prediction step.

Evaluation: Even though the tilted slices are only interpolated and not the result of real tilted CT scans, the radial descriptor was applicable and produced reasonable position predictions. As expected, the amount of the tilt angle θ correlates to the impact of the prediction quality of the algorithm. It can be seen that the impact to both e_{mean} and σ are almost linearly with increasing tilt angle θ. Yet the impact is just about 5 mm in case of a tilt angle of 30°and less than 2 mm in case of a $\theta \leq 10°$. The precise numbers can be seen in Table 4.5, a diagram visualizing the numbers is given in Figure 4.14.

Table 4.5: Mean error and standard deviation w.r.t. the tilt angle θ.

tilt angle θ	0°	5°	10°	15°	20°	25°	30°
e_{mean} (cm)	1.7	1.8	1.8	1.9	1.9	2.1	2.2
σ (cm)	2.7	2.8	2.9	3.1	3.0	3.3	3.2

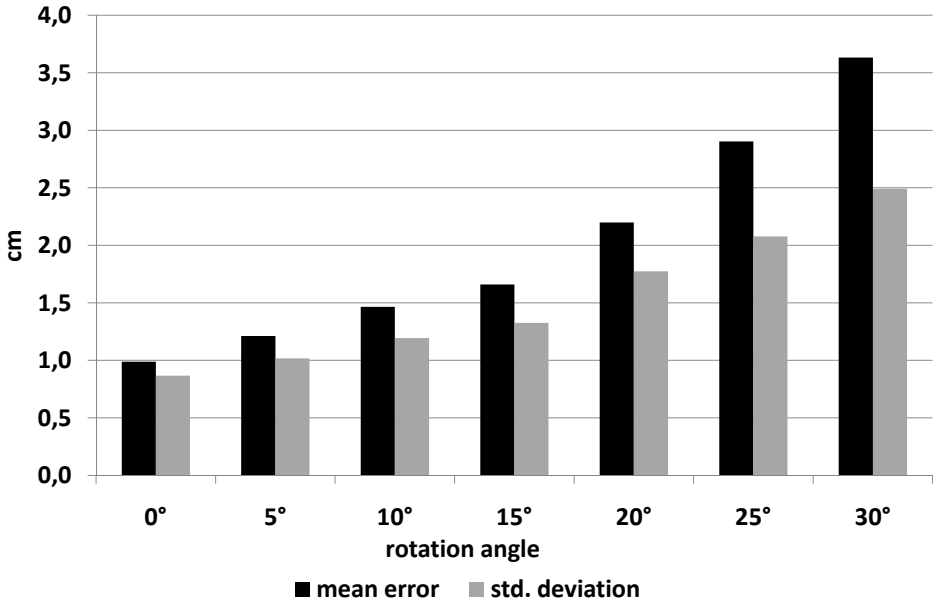


Figure 4.14: Impact of tilt angle to localization

4.7.5 3D Detection

Do demonstrate the improvement of the 3D technique described in Section 4.6, the results are compared to the results obtained from Section 4.5.

The results shown in Figure 4.15 describe that considering two adjacent slices already made a noticeable difference. As expected, the choice of $f(x)$ also becomes more significant with increasing value of m as more and more information has to be aggregated into the final feature vector. It can also

be seen that all functions except f_{Gaussian} outperform the 2D feature vector (black line) for all values of m in terms of both, accuracy and precision.

Nevertheless $f_{\text{inverse-squared}}$ converges to the performance of the 2D feature vector in case of $m \geq 5$ in both accuracy and precision, while f_{Gaussian} increases precision but looses accuracy at $m \geq 8$. This function shows that an improvement in the standard deviation σ does not necessarily have to result in better prediction accuracy, as can be seen for $m \geq 13$. The errors caused by using f_{Gaussian} for example were bigger but the error values were less scattered with likely less small error values - or in other words, f_{Gaussian} provides higher precision by the loss of accuracy.

Nevertheless, all other aggregation functions improve the prediction result. Considering both e_{mean} and the standard deviation σ, the best results were achieved with f_{linear} ((4.11)) and f_{polynome} ((4.10)). Both achieved a mean error of $e_{\text{mean}} = 16.5\,\text{mm}$ with a standard deviation of $\sigma \leq 32.4\,\text{mm}$ compared to $e_{\text{mean}} = 17.6\,\text{mm}$ and $\sigma = 35.9\,\text{mm}$ in the case of pure 2D features. These two function thus achieved an improvement of 6 % and 10 % respectively. Even though the improvement concerning the mean error is appreciated, it is the improvement of the standard deviation that is even more important as it increases the precision of the method and reduces larger errors.

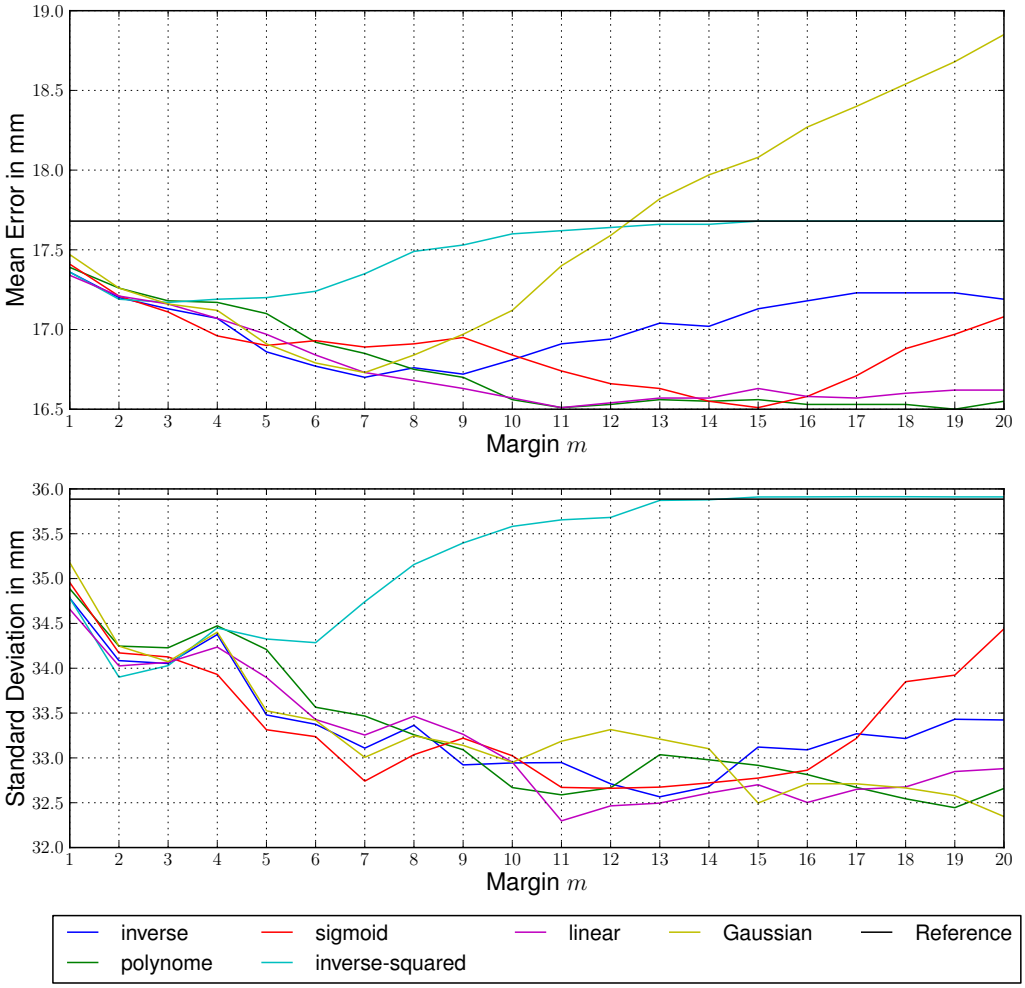


Figure 4.15: Mean error and standard deviation of cross-validation using 3D features compared to the pure 2D features (black line) depending on the margin m and the weighting function $f(x)$.

Chapter 5

Vertebra Detection

5.1 Introduction

Detecting the position and the shape of the spine and single vertebrae is a useful task for various purposes in medical imaging. In spine reconstruction, information being extracted from a body scan is used to build a 3D model of the spine allowing the examination of spinal maladies like scoliosis. Further areas are not directly interested in the particular characteristics of the spine but rather employ the gained information to acquire positioning information within a scan. The position and the form of the spine are well-suited for determining the position of the patient on the table because vertebrae have different shapes in varying parts of the body. Furthermore, bone structures are well suited for automatic detection tasks due to their clear and invariant display in CT scans.

To solve these problems the research community proposed several methods for detecting, segmenting, analyzing and reconstructing meshes of vertebrae and the complete spine being based on a complete 3D body scan. However, there are applications where a limited number of CT slices is available. For example, the slices being contained in a medical record which is sent to a specialist without access to the complete data in the PACS. To provide

valuable information in this or similar applications, the focus in this work is set on detecting the vertebrae in the smallest possible input, a single 2D image.

The task for detecting the position of the vertebrae in 2D CT slices on the transverse plane will now be issued in the following sections. The information being derived by this approach can be employed for localizing the center of the body w.r.t. the position of the patient on the examination table. Additionally, this method is suitable for distinguishing the cervical-, thoracic- or lumbar spine area which yields value information about the body regions of the given image. Also, the method shows reliable results even if larger regions of the considered body scan are available.

To make this method suitable for the named applications it was designed to meet several important requirements: First, it is not necessary to adjust parameters to the characteristics of the given scan. Furthermore, the method works fully automatic, i.e. the system marks the spine in the image without any user interaction. Finally, the method works efficiently without requiring large amounts of main memory which makes it a suitable component for larger imaging systems.

Comparable to the slice localization approach in Section 4, the method for the localization of the vertebra in 2D CT images was developed in two stages. First, the related work of the research community according to this topic is reviewed in Section 5.2, afterwards the first stage is presented in Section 5.3 and demonstrates the general applicability of the employed techniques. Section 5.4 improves this first stage by applying techniques to localize the initial position even better and afterwards refine the position by a dynamical of the result window that indicates the position of the vertebra. The data set and according experiments for both approaches will be presented in Section 5.5

The methods proposed in this chapter are (to be) published in [55] and [52].

5.2 Related Work

In [130, 140, 141, 142], Vrtovec *et al* construct 3D shape models of the spine and analyze the spinal curvature in CT images. Stern *et al* determine the spinal centerline in both CT and MR volume scans in their works [128, 129]. There are also several methods aiming at the detection and segmentation of the spine [79, 81, 150]. Nyúl *et al* [106] proposed methods for detecting the spinal cord and the spinal canal in 3D CT scans by using deformable fences or models. Methods being based on 3D MR scans are proposed by Schmidt *et al* [122], Corso *et al* [30] and Huang *et al* [68]. Though each of these methods are reported to achieve convincing results, all of them require a complete 3D volume scan. Despite the different nature of all approaches, a major disadvantage of the above methods is that they are all dependent on the availability of extended volume scans. While processing the complete scan in order to extract new information often yields good results, there exist use cases where only very small volumes or even single slices of a scan are available. For example, the available slices are taken from a medical record being send to a specialist without access to the complete data in the PACS.

When considering this more challenging setting, the number of related methods is considerably smaller. Rangayyan *et al* [116] for example use a Hough-Transformation to detect the spinal canal. However, the proposed method relies on reducing the the search space to an area that should not be larger than a region comprising the vertebra.

5.3 Static Detection

5.3.1 Introduction

In this section, the first stage of the development of a new method for detecting the position of the vertebrae in 2D CT slices on the transverse plane is proposed. Especially in applications where the volume of the available

scan is rather small or even a single slice, this method generates valuable information which can be employed for navigation and annotation purposes. For example, within the 2D slice the vertebrae is important as an orientation point for determining the center of the body w.r.t. the particular position of the patient on the table. Furthermore, it is possible to classify the detected vertebrae on the image into cervical-, thoracic- or lumbar spine area in order to determine the body region in which the available volume is placed in. Another use of this detector is as a fully automatic preprocessing step for constructing a 3D model of the spine within the available scan volume.

This new method has several characteristics allowing a broad use in various applications. First of all, the method allows reliable vertebrae detection without time-consuming parameter tuning. Since this method is fully automatic and parameterless, it is not necessary to adjust the parameters to a particular volume before receiving usable results. Furthermore, the detector is based on a directed readjustment step which avoids checking for the target region at any possible position on the 2D slice. As a result, the vertebrae is detected very efficiently allowing to apply the detector on large data sets or in interactive applications. A final useful characteristic of this approach is its low memory consumption making the detector a suitable component of more sophisticated imaging systems.

Technically, this method employs four steps to determine the position of the spine. The first step comprises several typical preprocessing steps. Then, possible candidate locations of the vertebrae are detected by extracting interesting pixels from a bone density map. Afterwards, image features are extracted for the candidate regions surrounding the interesting pixels and compared to an annotated sample set in order to select the most promising candidate region for further processing. After the best candidate region is identified, the result region is iteratively readjusted in a refinement step until the method converges to a local optimum.

The rest of this section is organized as follows: The algorithm will be described in Section 5.3.2, followed by the description of the experimental setting and the achieved results in Section 5.5.2.

5.3.2 Algorithm

In this section, algorithm for detecting the vertebrae within a single 2D slice of a CT scan is described. To determine the most relevant region within the a given image, four consecutive steps are performed which will be described in the following subsections. After describing the basic steps of the algorithm, some techniques for improving the processing time are discussed.

Preprocessing

The first step performs several preprocessing steps to make the relevant information easier to detect. Thus, the image is reduced to the region of interest by cutting off empty borders. A border is thereby defined as block of at least 20 lines (columns) where each line(column) contains at least 100 consecutive pixels with an HU value of more than $-600\,\mathrm{HU}$. After detecting the borders on all sides of the body, the region of interest is defined as the outline of the joined borders. Afterwards, the image is scaled to a size of $512\times512\,\mathrm{pixels}$ and all previously applied windowing filter is removed. In order to attenuate the effect of noise pixels which are very likely in CT scans, a Gaussian kernel of size $5\times5\,\mathrm{pixels}$ and $\sigma_{x,y} = 1$ is applied to the image. Figure 5.1 shows the result of the preprocessing step.

Candidate Selection

In the next step, a so-called bone density map is created which is used to identify interesting image regions that might contain the vertebrae. Therefore, the set of bone pixels $D \in [1\ldots512] \times [1\ldots512]$ is determined by selecting pixels having a HU value in a range of 500 – 1000 HU, so that

$$D = \{(x, y) \in I \mid p(x, y) \in [500, 1000]\} \tag{5.1}$$

where I represents the slice under consideration and $p(x, y)$ denotes the HU value of a pixel at location (x, y).

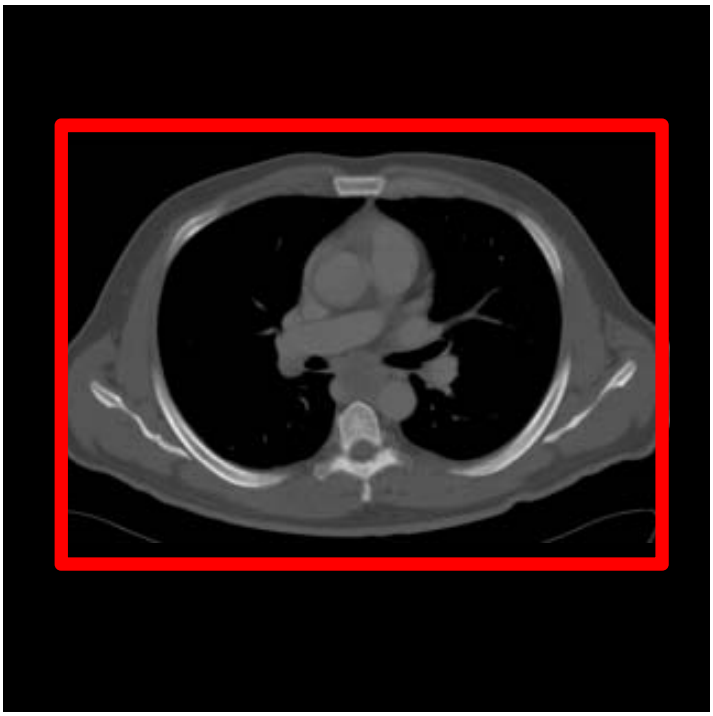


Figure 5.1: Image after pre processing and body detection.

For each bone pixel, the so-called bone density of a pixel is determined by summing up the Euclidean distances to all other bone pixels of the image w.r.t. the pixel coordinates. Formally, the bone density of a pixel at position (x, y) is defined as in (5.2). The densest bone pixel is thus defined as the pixel $(x, y) \in D$ with the lowest value of $density(x, y)$ (cf. (5.3)).

$$density(x, y) = \sum_{d \in D} \sqrt{(x - d_x)^2 + (y - d_y)^2} \tag{5.2}$$

$$(x, y) \in D \mid \forall (u, w) \in D \ : density(u, w) \geq density(x, y) \tag{5.3}$$

The bone density map is constructed by adding an entry for each bone pixel in combination with its bone density. Afterwards, the pixels of the bone density map are ordered in ascending order w.r.t. the bone density. In other words, the pixels denoting the highest bone density are sorted on top of the bone density map.

In order to obtain candidate regions from the bone density map, the first pixel of the list is expanded to a region of 84×68 mm which corresponds to twice the average size of an annotated spine region in the sample database of annotated images. The doubled size of the box is used because the annotation comprises only the vertebrae. Yet, it is required to include the spinous process as well as this region shows a very significant shape w.r.t. the spinal region it is located in.

Regarding the shape of a vertebra, the highest bone density is expected in the part of the vertebral body. As it is desired to extract image features describing the vertebral body as well as the spinous process, the box is not expanded equally in all directions. Instead the box's width is expanded in both directions but the height only downwards in order to raise the chance that the box also covers the complete spinous process. An illustration about the expansion of the box is shown in Figure 5.8 in Section 5.5.2 where the initial point of the bone density map is located at the point in the center of the inner rectangle which represents the average annotation box.

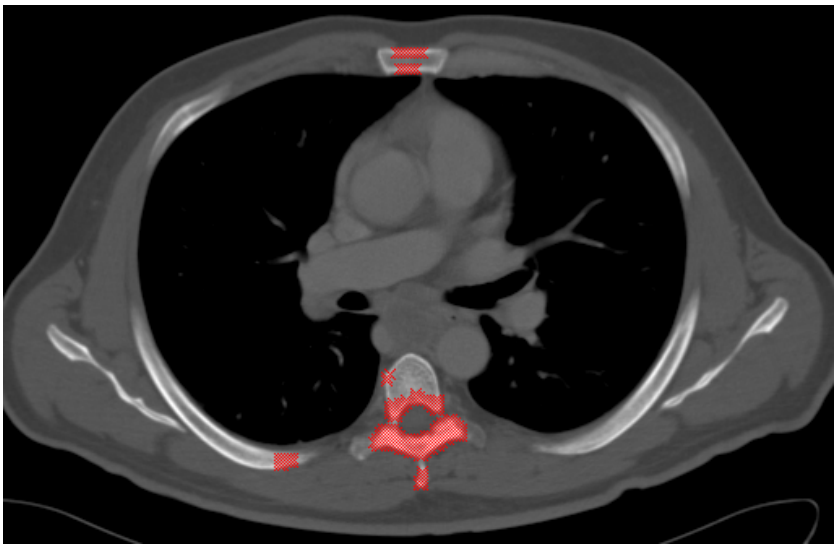
Afterwards, all other entries in the λ-neighborhood N_λ of 42×34 mm (which is the size of the average annotation) of the top entry are removed from the list in order to avoid the generation of regions being too similar to each other (cf. (5.4), $\lambda_x = 21\,\text{mm}$, $\lambda_y = 17\,\text{mm}$). This step is repeated five times in order to obtain the five most promising candidate regions from the bone density map. Figure 5.2(a) illustrates the considered bone pixels and Figure 5.2(b) displays the selected top 4 candidate boxes.

$$N_\lambda(x,y) = \{q \in D \mid d_x^1(p(x,y),q) < \lambda_x \wedge d_y^1(p(x,y),q) < \lambda_y\} \tag{5.4}$$

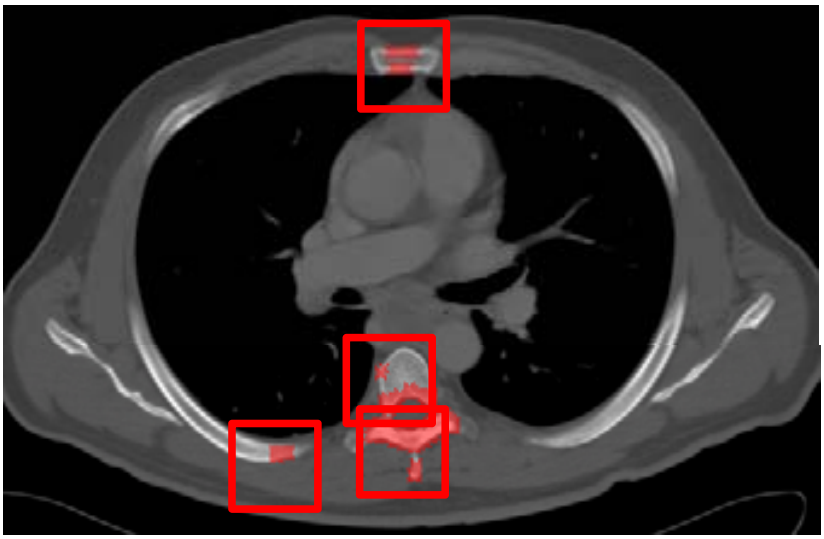
$$d_x^1(p,q) = |p.x - q.x| \;\; ; \;\; d_y^1(p,q) = |p.y - q.y| \tag{5.5}$$

Feature-Based Region Prediction

In the next step, a feature descriptor is derived for each of the remaining five candidate regions. In order to consider the spatial distribution, a candidate



(a)



(b)

Figure 5.2: Bone density map (a) and candidates selected from the bone density map (b).

region is divided into nine disjoint, equally sized sub regions. For each of these sub regions, a feature descriptor is derived and the descriptor for the complete region consists of the concatenation of the descriptors of its nine sub

regions. In this work, with three types of image descriptors were evaluated, i.e. greyscale (HU value) histograms, Haralick texture features and pyramidal histograms of oriented gradients (PHoGs). The performance of each of these descriptors is compared in the next section.

Figure 5.3: Candidate region selected by the feature-based prediction step.

HU Histograms: In the case of HU histograms, 16 uniform intervals in the scale of Hounsfield Units (HU) were distinguished. For each bin in the histogram, the number of pixels having a HU value in the corresponding interval were counted. Thus, 16 values for each sub region creating a feature vector of $16 \cdot 9 = 144$ dimensions for the complete descriptor.

Haralick Texture Features: As it was desired to match against image patterns of the vertebrae, the algorithm was also tested with the well known Haralick texture features [63] which also proved valuable information in Section 4.4. For the proposed method, all 13 Haralick features for five different distance values $(1, 3, 5, 7, 11)$ were created. This computation is done for each of the 9 subregions separately. The resulting feature vector contains $9 \cdot 13 \cdot 5 = 585$ features. As stated in Haralick *et al* [63], some of the features are highly correlated. This means that the resulting feature vector contains a lot of redundant information in its full representation. In order

to minimize these redundancies, a principal component analysis (PCA) was applied and thus the dimensionality of the texture descriptor was reduced to 30 dimensions.

Histograms of Oriented Gradients To extract the gradient features for a candidate region, important edges P_{edge} were extracted by applying the well known Canny operator [28] C. Important edges are defined by all locations, where the Canny operator computes values greater than zero (cf. (5.6)). In the next step, the angle of the gradient $G(x, y)$ is computed at the locations of important edges (cf. (5.7)).

$$P_{\text{edge}} = \{(x, y) | C(x, y) > 0\} \tag{5.6}$$

$$G(x, y) = \arctan \frac{\partial y}{\partial x} \; ; \quad \text{where } (x, y) \in P_{\text{edge}} \tag{5.7}$$

Afterwards, a 12 bin histogram is built over all values of $G(x, y)$ within the complete candidate region and for each of its nine sub regions. The histograms are then serialized generating a $(9 + 1) \cdot 12 = 120$ dimensional vector. This representation is referred to as PHoG (pyramid histograms of oriented gradients).

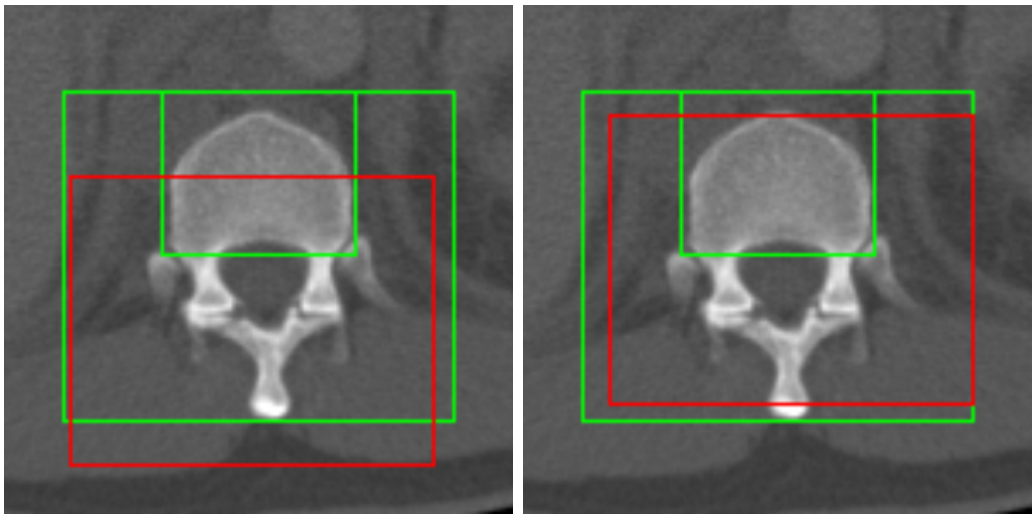
Candidate Selection To determine the most promising of the five selected candidate regions, a sample set of manually annotated regions containing the vertebrae is employed. Each sample in the database is described by the same feature descriptor as the candidate region. To compare the d-dimensional feature descriptors the Manhattan distance (5.8) is used.

$$dist_{Manhattan}(u, v) = \sum_{i=1}^{d} |u_i - v_i| \tag{5.8}$$

The most promising candidate region is determined by calculating the nearest neighbor of each candidate in the sample set. The candidate region having the smallest Manhattan distance to its nearest neighbor in the sample set, is then selected as result region (cf. Figure 5.3).

5.3.3 Refinement

After selecting the best of the initial candidate regions, the algorithm proceeds with an iterative optimization step moving the result to an local optimal position. In each step, new candidate regions are created by moving the position of the currently best candidate region by 5 mm to the top, bottom, left and right. For each of these new candidates, the distance to the nearest neighbor in the sample set is computed and the one with the smallest Manhattan distance is selected. The algorithm terminates, if there is no new candidate region having a smaller distance to its nearest neighbor than the result region in the previous step. Figure 5.4 shows the result region before (cf. Figure 5.4(a)) and after (cf. Figure 5.4(b)) the refinement step.



(a) (b)

Figure 5.4: Image before (a) and after (b) the refinement step. The green boxes denote the (doubled) annotation and the red boxes indicate the current result box of the algorithm.

5.3.4 Performance Tuning

In order to improve the processing speed of the algorithm, it is important to accelerate the computation of the bone density map, as it has a runtime of $O(n + m^2)$ with n being the amount of pixels of the image ($512 \times 512 = 262\,144$ in the case of the images in this data set) and m being the amount of pixels indicating bone structure. One way to improve the calculation speed is to rescale the image by a factor of 0.5 first. This way the amount of ordinary pixels and bone pixels can be significantly reduced. A further optimization employs the observation that the analyzed images are recorded while patients are lying on their back. Thus, it can be assumed that the vertebral column is located in the mid third of the image. Thus, the processing time is reduced to $O\left(\frac{1}{12}n + \frac{1}{16}\delta^2 m^2\right)$ with $\delta \in [0; 1]$ denoting the amount of bone pixels within the mid third of the image compared to the total amount of bone pixels.

5.4 Weighted Detection with Dynamic Resize

5.4.1 Introduction

In this section, the second stage of the development of a new method for detecting the position of the vertebrae in 2D CT slices on the transverse plane is proposed. Regarding the observations and experiences that were made during the work of the static detection in Section 5.3 there were some issues that motivated the improvement of the method.

Some of these issues that were experienced in stage one and were addressed in stage two now were for example that – according to radiologists – almost all CT scans are recorded with the patient being in a dorsal position. The intention was, that this information should be regarded either during the creation of the bone density map or at the time of the candidate selection (cf. Section 5.3.2).

Technically, this extension to the method described in Section 5.3 employs

five steps. In the first step, the input image is preprocessed. Then, the system extracts relevant pixels and weights each pixels. Both, the search mask for determining the relevant pixels and the weighting functions are learned from an image repository containing slices with spine annotations. In the next step, candidate locations in the selected area are extracted. Afterwards, image features for each candidate are extracted and the best candidate locations are selected for employing instance-based learning. In a final step, the annotation box around the best candidate location is fitted to the boundaries of the given vertebra.

The contributions of this method are an improved search mask for the localization of candidate locations of the vertebrae, as well as a new weighting method on the search mask that supports the interesting point detection. Also, an improved algorithm to define the region of the vertebrae more precisely and finally one improved and one new quality measurement for the validation. These quality measures were needed as the former quality measure in (5.18) was not applicable for the case of the refined result boxes.

The rest of this section is organized as follows: The algorithm will be described in Section 5.4.2, followed by the description of the experimental setting and the achieved results in Section 5.5.3.

5.4.2 Algorithm

The detection of vertebrae consists of the five major steps, which are: image preprocessing, region extraction and weighting, candidate generation, candidate selection and refining the best candidate. An overview of the parameters that will be used in the following section can be found in Table 5.2.

Preprocessing

In the first step, the original CT image I is rescaled to a unified width and height of 512×512 pixels. Afterwards, a 2D Gaussian kernel $G(x, y, \sigma)$ with

$\sigma_{x,y} = 1$ is applied to I. The Gaussian blur has the effect of reducing noise in the CT slice. This noise can be caused by a high resolution along the z-axis of the body, which results in very thin CT slices or in case of low-dose CTs. With the given trend to both higher resolution and lower x-ray doses (low-dose CTs), this step will become more and more important in the near future.

Region Extraction and Weighting

On a transversal CT slice, the spine is always located in the lower middle region of the image if the scan was recorded in a dorsal position of the patient (which is the case in a huge majority of the cases, according to some radiologists). Thus, the image I is limited to the sub region which possibly could contain the spine by applying a search mask ρ_{sm}.

In contrast to the method in Section 5.3, where the search space was set heuristically to $\frac{1}{3}$ of the patient's body, ρ_{sm} is determined empirically based on a training database DB. As a result, the relevant region of the image can further be restricted without losing relevant information.

In particular, the training database DB consists of a set of CT volume scans $V_j \in DB$ where each volume scan is represented by an ordered set of images $I_{i,j} \in V_j$ ($i \in [1, n]$). Additionally, each vertebral body of a volume scan is annotated with a paraxial bounding box representing the ground truth. Thus, each $I_{i,j} \in V_j$ refers to a set $M_{i,j} = \{MBR(I_{i,j})\}$ that contains the minimum bounding rectangles (MBR) which are generated by the intersection of the annotation boxes of V_j with the image $I_{i,j}$. The cardinality of $M_{i,j}$ can thus be $[0 \dots m]$ where m is usually no more than 3. $|M_{i,j}| = 0$ occurs, if the CT slice displays the section between two vertebral bodies, so that only a spinal disc is visible. $|M_{i,j}| = 1$ is the obvious case where the slice shows exactly one vertebral body. $|M_{i,j}| > 1$ occurs, if the rotation of the spine is large enough so that a slice along the transversal plane shows not only one single vertebral body but also parts of a preceding or succeeding vertebral body.

To define the search mask ρ_{sm}, the union U of all sets of MBRs $M_{i,j}$ of all volumes $V_j \in DB$ is created. Afterwards, the convex hull of the set U is computed using Graham's scan algorithm [56]. ρ_{sm} is then defined by:

$$\rho_{sm} = ConvexHull(U) \tag{5.9}$$

$$U = \{\cup M_{i,j} \mid M_{i,j} = \cup MBR(I_{i,j} \in V_j) \wedge V_j \in DB\} \tag{5.10}$$

This step requires that DB is large and diverse enough. Otherwise, ρ_{sm} will be too selective and too small so that no correct points of interest can be selected in the following steps. Building the convex hull around the set of ROIs is used to avoid that ρ_{sm} is overfitting to the database.

Candidate Generation

The method aims to detect points within the image that are candidate locations for a vertebra. Compared to its surrounding, a vertebra itself is a very locale bone structure. Thus, a bone density map is created for all suitable bone pixels $b_{x,y} \in \rho_{sm}$. Suitable bone pixels are all pixels $b_{x,y}$ with an HU-value in a certain HU window $[\beta_{lower}, \beta_{upper}]$. For this algorithm, the window was lowered to a more sensitive HU range for compact bones, so that $\beta_{lower} = 300\,\text{HU}$ and $\beta_{upper} = 1000\,\text{HU}$ [118]. Spongy bones, which can be observed in an HU range of 50 – 200 HU can also be observed in the inner part of the vertebral bodies of elderly patients. Nevertheless, the outer part of the vertebral bodies is typically compact and thus in a higher HU range, so that pixels with less than 200 HU need not be taken into account in this case. Furthermore, a too low value for β_{lower} can lead to an increased rate of false candidates. The set D of suitable bone pixels is thus defined as

$$D = \{(x, y) \mid p(x, y) \in \rho_{sm} \wedge p(x, y) \in [\beta_{lower}, \beta_{upper}]\} \tag{5.11}$$

where $p(x, y)$ denotes again the HU value at location (x, y) and p_x, p_y the x- and y-coordinates respectively.

This is almost the same definition as for the bone density map in (5.2) (Section 5.3.2). The differences are the restriction to the search mask ρ_{sm} and

the employed distance metric. In the former definition of the density map, the L_2 distance was employed. This was changed to the L_1 metric due to better computation performance. Same as in Section 5.3.2, the bone pixels with the smallest accumulated distances $D(x, y)$ have the highest bone density because they are closer to other bone pixels than the pixels with larger accumulated distance values.

As it was experienced in Section 5.3, this rather simple distance map still leads to false candidate detections in the area of the sternum, the clavicle and the hips. In these cases, very dense bone structures extend into the search mask and might be selected as false candidate locations. A further refinement of the candidate region is not an applicable solution as the mask will either overfit or crop other true candidates from the search mask. For this reason, it is proposed to apply a weighting function w which is also derived from the training database DB:

$$w(x, y) = arg\ max(1,\ log(|R(x, y)|)) \tag{5.12}$$

$$R(x, y) = \{R_i \in ROI_{DB} | (x, y) \in R_i\} \tag{5.13}$$

where ROI_{DB} is the set of all annotation ROIs in the database DB and $R(x, y)$ the set of all ROI covering the location (x, y).

Thus, $w(x, y)$ can be regarded as a measure for the likelihood that location (x, y) displays a true candidate. The logarithm function is applied as a damping function in order to reduce the impact of pixels with large values of $|R(x, y)|$ and thus to avoid overfitting. In general, any kind of monotone damping function with a co-domain of $[1, \infty[$ would be applicable. The weighting function is then applied to the distance map D by a pixel wise multiplication (5.14) building the weighted distance map D_w. An illustration of w can also be seen in Figure 5.5.

$$D_w = D \circ w \tag{5.14}$$

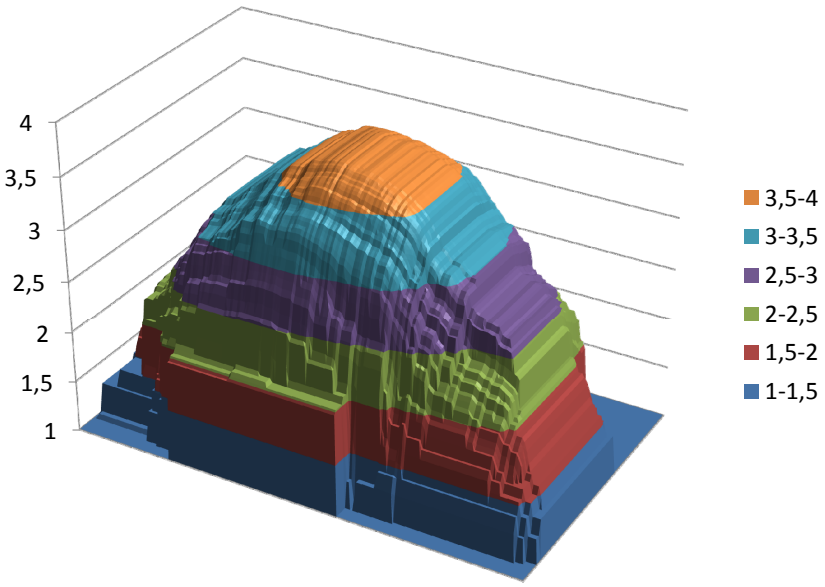


Figure 5.5: Illustration of the weighting function w. The height value denotes the value of $w(x, y)$. The bottom right of the illustration represents the bottom line of ρ_{sm}.

Candidate Selection

After the computation of the weighted bone density map D_w, the algorithm detects the η most promising candidate locations for the position of the vertebrae. This set will be denoted by C_{cand}. Simply extracting the η densest locations from D_w is not feasible, because it is very likely that all of the η locations are positioned within a small region neighboring the global minimum of D_w.

This step is comparable to the candidate selection described in Section 5.3.2. In difference to Section 5.3.2, the λ neighborhood is now represented by a circle with radius λ which is smaller than the rectangle in the previous approach.

$$N_\lambda(p) = \{q \in D_w \mid dist_{euclidean}(p, q) < \lambda\}; \tag{5.15}$$

In order to obtain the η candidates, all pixels of D_w are ordered in list in

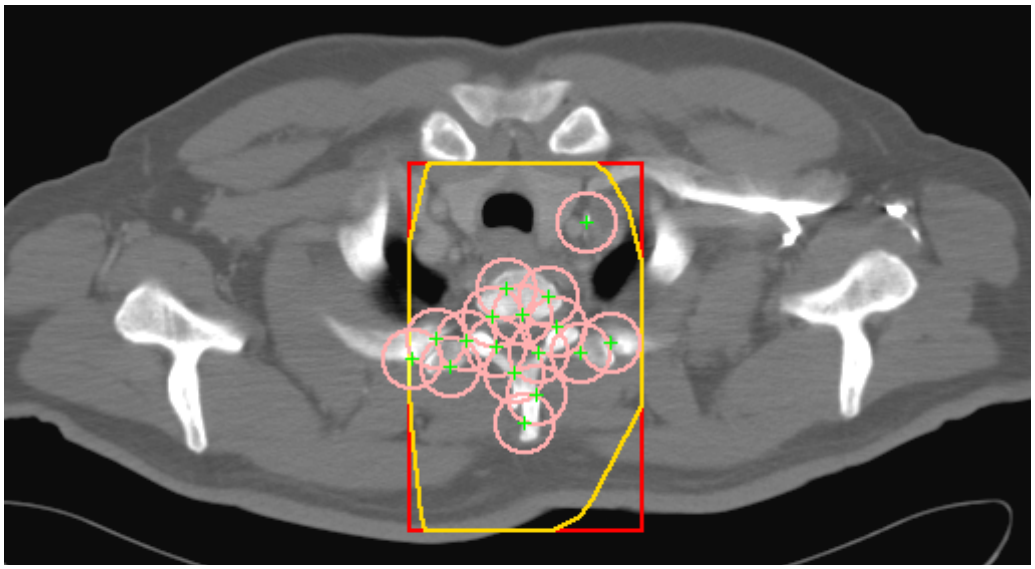


Figure 5.6: The search mask ρ_{sm} (yellow line), and the densest pixels in $D(x, y)$ (green dots) including the λ neighborhood marked by circles.

ascending order of their bone density. Afterwards, the first pixel of the list is removed and put into C_{cand}. All pixels in the λ neighborhood of this pixel are then removed from the list. This procedure is repeated η times candidates. The locations of the densest pixels is illustrated in Figure 5.6.

Feature-Based Region Prediction

After the extraction of the candidate locations, the image features from areas around each candidate location are extracted and based on an instance-based learner, the most promising location is selected.

Same as in Section 5.3.2, the features are extracted from an ROI ϕ_{box} which is is centered at each location in C_{cand}. The size of ϕ_{box} also remains twice the size of an average vertebral body in order to capture the information of the vertebral body including surrounding tissue and spinous process. ϕ_{box} remains sub-divided into equally sized sub-regions from which the features are extracted and serialized to build the feature vector $\phi_{desc(i)}$ with i denoting

the type of feature.

In the experiments different kinds of well known image features were evaluated. Same as in the previous approach, HU-Histograms with 16 bins ($\phi_{desc(1)}$), Haralick texture features [63] ($\phi_{desc(2)}$) and PHoGs ($\phi_{desc(3)}$) were evaluated. Additionally to these three image features binary histograms ($\phi_{desc(4)}$) and resized image regions (thumbnails) ($\phi_{desc(5)}$) were evaluated

After building a ϕ_{desc} for each location in C_{cand}, an instance-based learning is applied for the features. This is done by determining the nearest neighbor for each $\phi_{desc} \in C_{cand}$ in the training database DB_{feat} containing the feature descriptions of the annotations by using the L_1 metric. The location of the feature vector with the smallest distance to database annotation is finally selected as the best candidate position of the vertebrae.

Refinement

At this stage, ϕ_{box} of the selected candidate describes the area which was used for feature extraction and thus, it is about twice as large as the actual vertebral body. This issue is addressed in this section, where the size of ϕ_{box} is reduced to fit the result box as tight as possible to the vertebral body. The dynamic refinement algorithm reduces the width and height of ϕ_{box} under consideration of the bone density at the borders of ϕ_{box}. A standardized scale factor is not applicable in this case, because the size of the vertebrae increases from the thoracic vertebrae to the last lumbar vertebrae. In the following, the method for relocating the top border downwards is described. The other borders are relocated respectively and the refinement process is applied for each border separately.

First, an ROI of κ_{block} pixel rows is aligned at the inner top row of ϕ_{box}. This ROI is moved downwards until each pixel row of the ROI contains at least κ_{row} pixels with an HU value greater than κ_{HU} or until the lower border of ϕ_{box} is reached. Thus, the y-location of the ROI can be defined as:

$$ROI.y = \{min(y) \in \phi_{box} | \forall row_i \in ROI : |p(x, y_i) > \kappa_{HU}| > \kappa_{row}\} \quad (5.16)$$

The resulting region is denoted by κ_{box}. If κ_{box} is empty (e.g. if the above ROI was moved beyond the opposite side), the values κ_{block}, κ_{HU}, κ_{row} and κ_{col} respectively are softened. If κ_{box} was refined too little so that the area decrease is less than $\kappa_{\%}$ compared to ϕ_{box}, the values are hardened and the refinement is restarted in a second iteration. If the second iteration also results in κ_{box} being either too small or too large, κ_{box} defaults back to the size of ϕ_{box}. Finally, it is ensured that κ_{box} has a minimum size of $\kappa_{limit} \times \kappa_{limit}$ pixel which is achieved by rescaling the width/height accordingly if width or height fall below the limit. The default values used for the refinement including softening and hardening are shown in Table 5.1. Illustrations of the ROIs are shown in Figure 5.7. The pseudo code of the refinement process can be found in Algorithm 1.

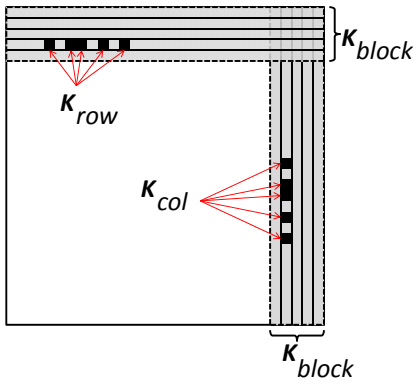


Figure 5.7: Parameters used in the refinement process. The column and row on the left/bottom are omitted for simplification.

The approach proposed by Rangayyan *et al* [116] was also evaluated in order to detect the spinal canal by using a Hough transformation. The intuition was, that this information could be used to refine the position even further. Unfortunately the method turned out to be very unstable on the given data so that no positive contribution of the method could be observed.

Table 5.1: Values for the refinement procedure. The upper two rows show the default values, the lower rows show the deltas for each value which are applied if the values are *soften*ed and *harden*ed respectively.

κ_{block}	8 px	κ_{row}	8 px	κ_{col}	17 px
κ_{HU}	250 HU	$\kappa_{\%}$	0.04	κ_{limit}	60 px
κ_{block}	$+2/-2$ px	κ_{row}	$-3/+3$ px	κ_{col}	$-3/+3$ px
κ_{HU}	$-50/+100$ HU				

Algorithm 1 Pseudocode for the refinement process. The delta of the values applied in the $soften/hardenParameters()$ are displayed in Table 5.1.

$\kappa_{refined} \leftarrow refine(\kappa_{box})$
if $isEmpty(\kappa_{refined})$ **then**
  $softenParameters()$
  $\kappa_{refined} = refine(\kappa_{box})$
else if $area(\kappa_{refined}) > \kappa_{\%} \cdot area(\kappa_{box})$ **then**
  $hardenParameters()$
  $\kappa_{refined} = refine(\kappa_{box})$
end if
$\kappa_{refined}.height \leftarrow$ increase if $width < \kappa_{limit}$
$\kappa_{refined}.width \leftarrow$ increase if $height < \kappa_{limit}$
$result \leftarrow \kappa_{refined}$

5.5 Evaluation

5.5.1 Data Set

In the experiments for the methods shown in Section 5.3 and Section 5.4, real world thorax CT scans of 34 different male and female patients with a total of 9 239 images were used, consuming more than 10 GB disk space. The scans were recorded with different CT scanners with different resolutions along the z-axis. Thus the scans are very heterogeneous w.r.t. resolution (along the x- and y-axis), contrast media, patients' body size, gender and age.

Table 5.2: List of parameters used including their meaning and measures.

α_s	standard annotation box
α_d	α_s doubled in height and width
α_v	α_s doubled in height only
$\phi_{desc(i)}$	Type of feature
ρ_{box}	ROI used for feature extraction
λ	neighborhood radius (pixels)
η	maximum amount of candidates
κ_{box}	result ROI after candidate selection
$\kappa_{refined}$	refined ROI
κ_{block}	minimum of rows/columns for borders
$\kappa_{row}, \kappa_{col}$	minimum number of pixels per row/column
κ_{HU}	lower bound for pixel detection (HU)
$\kappa_{\%}$	minimum area difference to κ_{box} (%)
κ_{limit}	minimum width/height of final ROI
$\xi_{overlap}$	area overlap
$\xi_{distance}$	distance between centers of the ROIs

Some patients also show signs of implants which may cause disturbances and artifacts on the image.

In order to obtain a reliable ground truth, the data set was annotated by a clinician using the MEDICO tool [124]. For the ground truth, each vertebral body of the spine was annotated by a single 3D paraxial bounding box enclosing the vertebral body and a tag that identified the vertebral body (C1-C7, T1-T12, L1-L5). The spinous processes were not included into the annotation boxes because this would have created comparatively large bounding boxes and also quite large overlaps between the annotations of consecutive vertebrae especially in the area of the lumbar vertebrae. Even though that only the vertebral bodies were annotated by enclosing boxes, it was not possible to completely avoid an overlap between annotation boxes. This can be explained by the fact that the annotation boxes are paraxial but

the spine describes a curve along the z-axis of the body. Thus, the vertebrae might be rotated to a certain degree.

In total, there were 393 bounding boxes with sizes from 22.9×17.2 mm (36×29 px) to 89.6×53.7 mm (204 × 120 px) and an average size of 43×36 mm. In the experiments, a cross validation on the patients' volume scans was employed so that it was ensured that the query image was never compared with other images of the same scan of this patient in order to avoid a bias towards the patient from which the query image was obtained.

5.5.2 Static Detection

Quality Measure

In order to evaluate the performance of the proposed algorithm and the according features, an overlap O between the manually annotated box B_m and the box B_a was defined, with B_a being determined by the presented algorithm. The intuitive definition of an overlap

$$O_{simple} = \left(area(B_m) \cap area(B_a)\right) / area(B_m) \tag{5.17}$$

does not cope with the problem that the annotation boxes are varying in size. Yet, this is the case as vertebral bodies are smaller in the upper part of the spine than in the area of the lumbar spine. In contrast to the varying size of the annotated boxes, the size of the result regions have a fixed size. The idea to use the largest annotated box was rejected due to the large maximum values encountered in the annotation (89.6×53.7 mm) which would lead to impractical results of very large overlaps which would always suggest a large overlap.

Using the average box size as target size of the algorithm led to the problem that O_{simple} could never reach a value of 100 % in regions being larger than the annotation even if the found region was overlapping the annotation perfectly. To solve this problem, the overlap was defined by the ratio of the

intersecting area relatively to the smaller of both areas (5.18):

$$O = \frac{area(B_m) \cap area(B_a)}{argmin(area(B_m), area(B_a))} \tag{5.18}$$

Because the annotated box only contains the vertebral body itself but not the very characteristic spinous processes, it was decided to enlarge the box from which the features are extracted by a certain factor (cf. Figure 5.8).

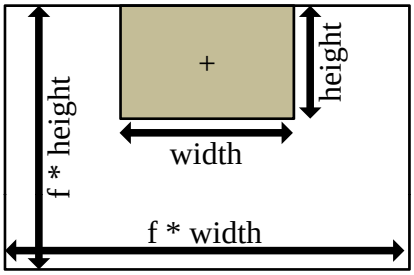
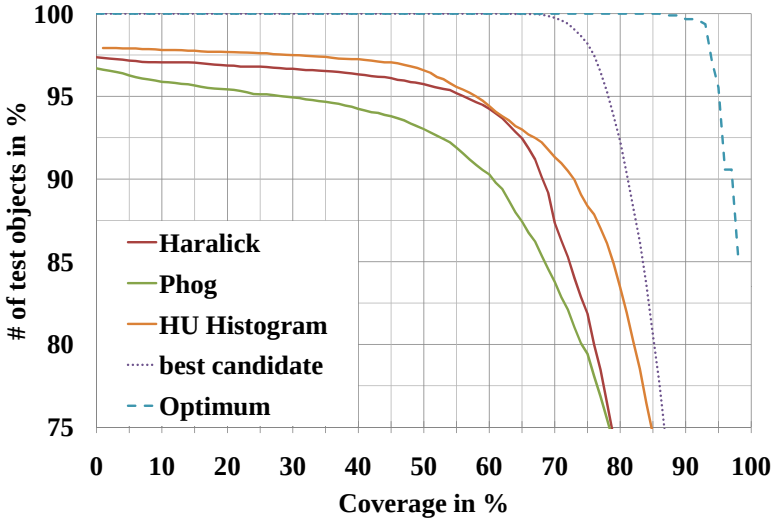


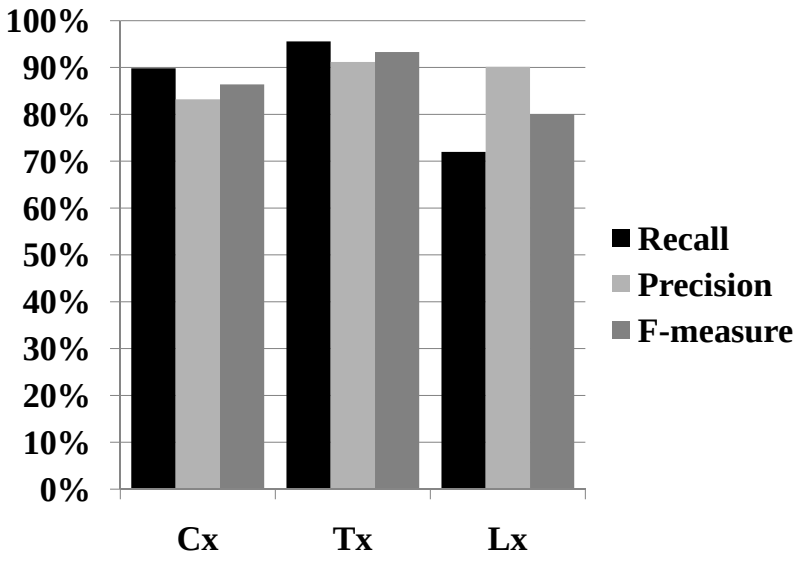
Figure 5.8: Illustration of the enlargement of the average annotation box (inner box) by a factor f. The plus sign in the middle of the inner box denotes the center of the enlargement process during the candidate selection.

Results

Feature Type Selection In the first set of experiments, the suitability of the three feature representations (HU-Histogram, Haralick texture features, PHoGs edge features) for the detection method was evaluated. Figure 5.9(a) shows the results of these experiments. It can be seen that the orange line indicating the HU histogram features clearly outperforms the other feature descriptors as the histogram features always provide a higher overlap value than the other feature types in up to 60 % of the cases. Exactly 60 % overlap could be observed in 94 % of the cases for both HU Histograms and Haralick features. However, the largest overlaps were only achieved by the HU Histograms (>88 % overlap in 75 % of the cases compared to 83 % overlap in case of the Haralick features).



(a)



(b)

Figure 5.9: (a): Performance of different feature representations and the quality of the candidate selection. (b): Precision, recall and F-measure of the proposed algorithm using HU-histograms when classifying images into regions of the cervical- (Cx), thoracic- (Tx) and lumbar (Lx) spine.

Candidate selection. Figure 5.9(a) also shows the impact of the bone density map and the initial candidate selection. The dotted purple line indicates the performance that could be achieved if the candidate selection would always choose the correct candidate from the candidate set. The dashed blue line indicates a perfect match. Due to the fixed size of the box found by the algorithm and the different size of the true annotated box, there cannot be a 100 % match in all cases.

The difference of the dotted best candidate line to the orange line of the histogram features indicates the error being caused by the process selecting the best candidate region. It was observed that if the best candidate region would have been selected in all cases, the coverage of 100 % was achieved in up to 65 % of the cases and a coverage of 87 % was achieved in up to 75 % of the cases. The difference between the dotted and the dashed line indicates the error which is caused by the generation of the candidate boxes.

Classification of the spine. Another experiment was conducted to check whether the HU Histograms would also be suitable to distinguish between cervical-, thoracic- and lumbar spine. Thus, the vertebral bodies were separated into three classes (Cx: cervical-, Tx: thoracic-, Lx: lumbar spine) and afterwards classified each detected box into one of the three classes. To determine the class label of a new image, the label of the closest neighbor of the result region in the sample set was selected.

The result of this experiment can be seen in Figure 5.9(b) and Table 5.3. It can be seen that both precision and recall are greater than 70 % for all three classes with the precision being greater than 90 % in case of the thoracic- and lumbar spine and recall being greater than 90 % in case of the cervical- and thoracic spine.

Candidate boxes As shown in the experiment above, the selection of the correct candidate box is crucial for the performance of the algorithm. Thus, the impact of the considered number of candidate regions (n_{cand}) to the

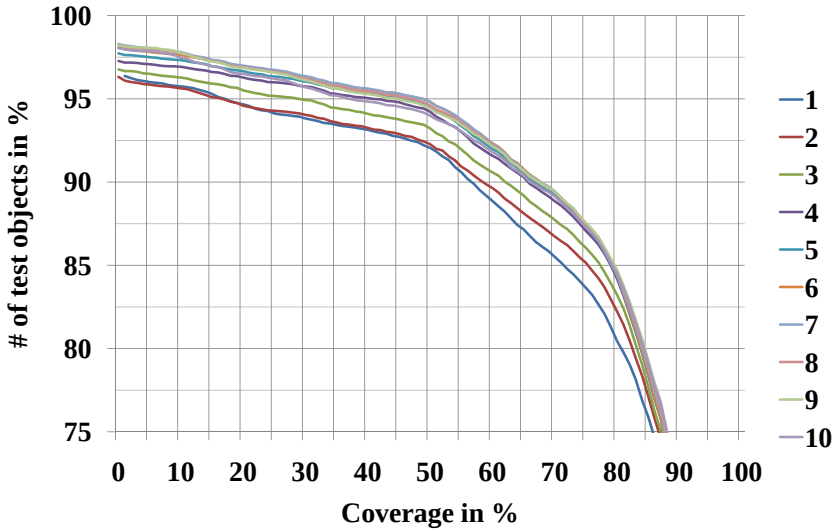
Table 5.3: Results from the classification of the found regions into the cervical- (Cx), thoracic- (Tx) and lumbar spine (Lx).

	Recall	Precision	F-measure
Cx	90 %	83 %	86 %
Tx	96 %	91 %	93 %
Lx	72 %	90 %	90 %

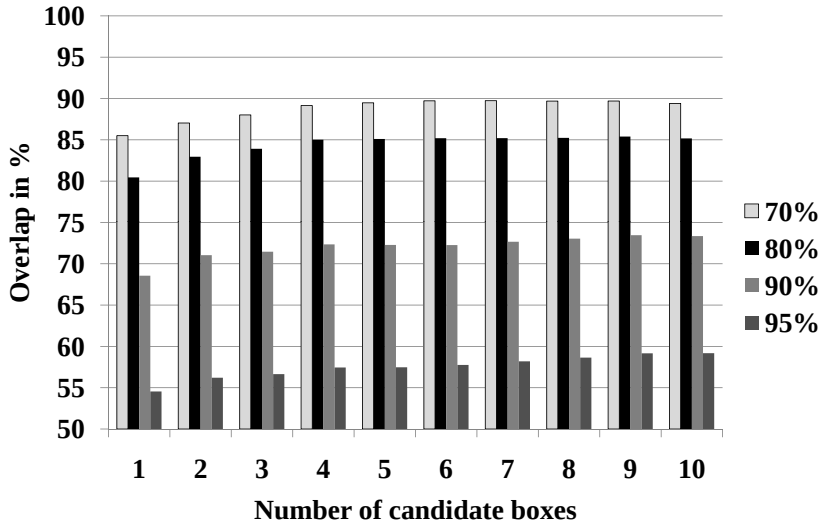
performance of the algorithm was evaluated in this step.

The result of this evaluation can be seen in Figure 5.10. The fact that the lines in Figure 5.10(a) are rather close to each other indicates that the amount of n_{cand} does not have a very large impact for $n_{cand} > 3$. The results displayed in Figure 5.10(b) support this assumption as there is no significant difference in the performance for $n_{cand} > 4$. Thus, all further experiments were conducted with n_{cand} set to 5. The reason for choosing a comparatively small number is that for each candidate region, 1NN query to the sample set has to be performed. Therefore, limiting the amount of candidate regions directly affects the time needed for the 1NN queries. Unfortunately, due to the large dimensionality of the employed feature descriptors, it is not possible to employ conventional index structures to improve the runtime performance of nearest neighbor queries in the sample set. Therefore a full table scan is required for each such query. As a result, increasing the number of considered candidate regions increases the search time linearly.

Size of candidate boxes Another important factor having a large impact on the performance is the size of the box from which the image features are extracted. As mentioned before, the region from which the image features are extracted should be larger than the average annotation box because a region which only shows the vertebral body does not contain a lot of information and does not include the very characteristic shape of the spinous process. On the other hand, if the area is too large, the vertebrae will only cover a small part of the result region. Thus, a trade-off between accuracy and

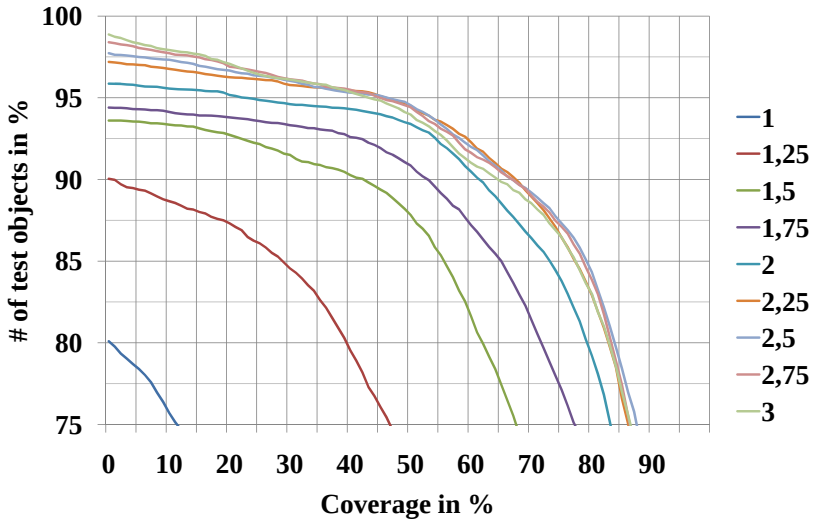


(a)

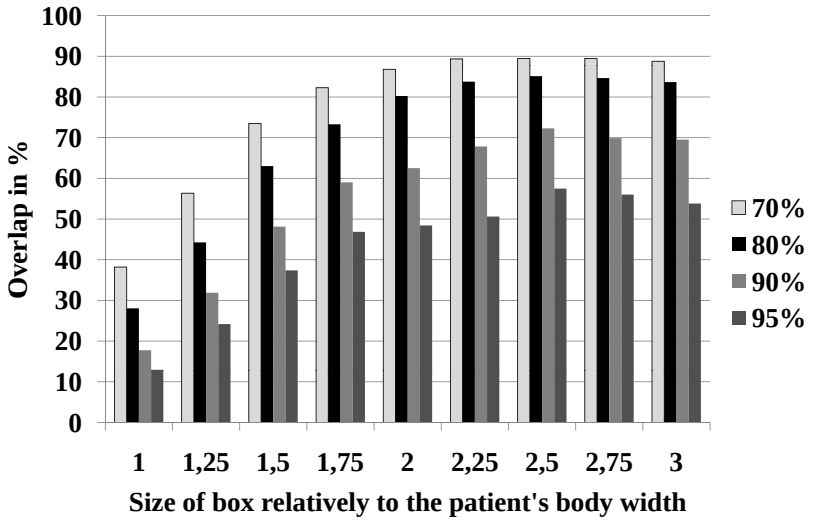


(b)

Figure 5.10: Impact of the number of candidate boxes (1-10) being selected from the bone density map on the classification performance. The bars marked as 70 %, 80 %, 90 % and 95 % indicate the amount of test cases achieving the given overlap.



(a)



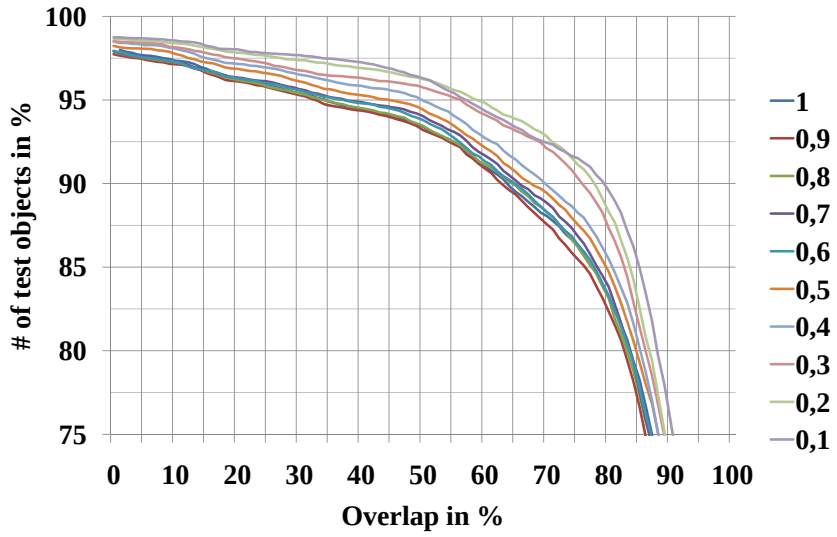
(b)

Figure 5.11: Impact of the size of the area from which features are extracted on the classification performance. The size of the box is regarded relatively to the average annotation box. The bars marked as 70 %, 80 %, 90 % and 95 % indicate the amount of test cases achieving the given overlap.

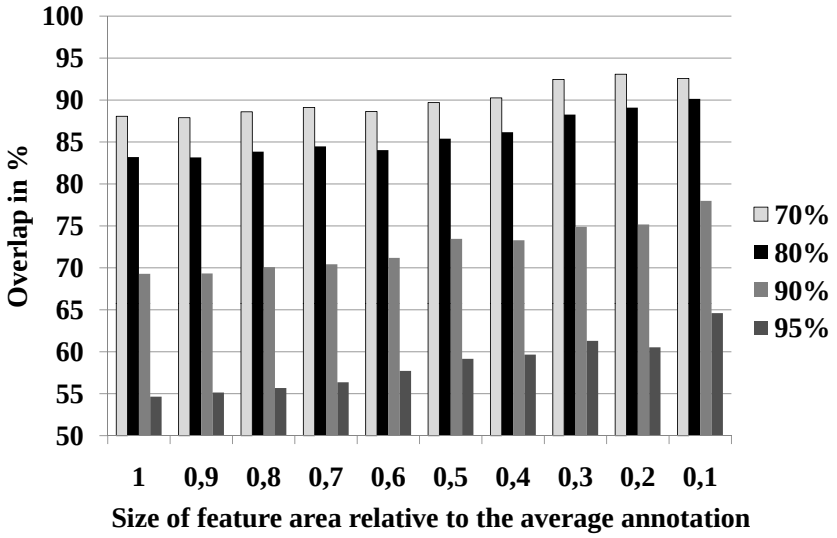
selectivity of the result region has to be found. To evaluate this trade-off, the average annotation was enlarged by a factor $f$between 1 and 3. The result can be seen in Figure 5.11. The diagrams show a clear correlation between the performance of the algorithm and the enlargement of the annotation box with a slight maximum at about $f = 2.5$. The performance decrease for $f > 2.5$ is caused by the effect that the result region might be extended beyond the borders of the of the lower part of the image. Also the 3×3 grid from which the features are extracted is becoming rather rough in this case which makes small important structures less significant.

Size of the inspected area of the image Another source for errors is the bone density map. Especially if the resolution of the CT-scan along the z-axis is very high, it might happen that there is almost no bone structure in the area where the vertebrae is expected because the slice is exactly between two vertebrae and thus only shows the intervertebral disk and the spinous process. In these situations, there is a comparatively high bone density along the ribs or at the sternum, which of course generates false hits.

In Section 5.3.4, it is mentioned, that for performance reasons, it is possible to limit the search for bone pixels to the mid third of the detected image area. This step has not only the effect that the amount of pixel operations is being considerably reduced but also that the possible locations of the spine are limited to the most relevant area as well. Therefore, there exists another trade-off that must be dealt with: The algorithm could scan the complete image and thus provide highest flexibility at the cost of both runtime and accuracy. If the scanned area is very small, the processing speed is better but the classification might be less robust in cases where the vertebrae is outside the considered area. For example, if the bounding box detection fails, it might occur that the actual position of the spine is outside the considered area. Figure 5.12 shows the evaluation of this experiment and proves the assumption that the performance increases with a decreasing size of the scanned area. In order to choose a reasonable trade off between flexibility, processing time and accuracy, a factor of $\frac{1}{3}$ was chosen for the evaluation of the algorithm.



(a)



(b)

Figure 5.12: Impact of limiting the area being analyzed for bone pixels on the classification performance. The size of the analyzed area is regarded relatively to the patient's body width. The bars marked as 70 %, 80 %, 90 % and 95 % indicate the amount of test cases achieving the given overlap.

Table 5.4: Execution time

	HU Histograms	Haralick features	PHoGs
Execution time	38 min	57 min	34 min

Speed and Memory The processing time for all 9 239 images was 38 min on an AMD Athlon 2.59 GHz, which is about 240 ms per image in average in the case of HU Histograms (Haralick: 57 min, PHoG: 34 min, cf. Table 5.4). The settings used for this measurement comprised the complete process chain including the image preprocessing, bone density, candidate selection, nearest neighbor search (with the features kept in memory), refinement and validation. The algorithm was implemented in Java 1.6 with ImageJ [117]. Also the memory footprint of the algorithm did not exceed the standard size of 32 mb for the heap of the Java VM.

The comparatively sparse amount of resources needed for this algorithm also suggests that the method could also be applied in a large fashion with several parallel execution paths.

5.5.3 Weighted Detection with Dynamic Resize

Quality Measure

In Section 5.3, the area overlap between the detected area and the doubled annotation box α_d was used. Since the previous method does not use any refinement step, the result boxes have fixed sizes so that the size of the compared boxes is very similar. In this method, the predicted region can be much smaller than α_d due to the refinement. Thus it can happen that an ROI which is completely contained in the annotation box can still be strongly displaced. Nevertheless, as it is contained in α_d the yielded result will still show a complete cover and thus a perfect hit.

To overcome this problem, a smaller annotation box is employed in this case. The new annotation box around the vertebral body is only extended

vertically to the bottom by a factor of 2, building the new ROI α_v. The extension to the bottom is necessary to make sure that the spinal process is part of the annotation. Nonetheless, the new annotation boxes are only half of the size of the boxes employed before.

Formally, the overlap is now measured as follows:

$$\xi_{overlap} = \frac{area(\alpha_v) \bigcap area(\kappa_{refined})}{argmin(area(\alpha_v), area(\kappa_{refined}))} \tag{5.19}$$

As a second quality measure, the distance $\xi_{distance}$ between the centers of the ROIs α_v and $\kappa_{refined}$ is computed. $\xi_{distance}$ describes the spatial derivation of the search compared result to the true position.

$$\xi_{distance} = \sqrt{(p.x - q.x)^2 + (p.y - q.y)^2} \tag{5.20}$$

$$p = center(\alpha_v) \quad ; \quad q = center(\kappa_{refined}) \tag{5.21}$$

This measure is proposed especially for the case, that the sizes of α_v and $\kappa_{refined}$ have very different sizes, so that one ROI is covered completely by the other. In such a case, it is more preferrrable that the center of the ROIs are close to each other which indicates a better result than the same $\xi_{overlap}$ with larger $\xi_{distance}$.

Evaluation

Same as in Section 4 and Section 5.3 the evaluation was done by applying a leave-one-patient-out validation, where a complete CT scan was defined as the source for query slices and all other scans were used as training data sets. In the following, the method proposed in this section (named **EVD**) will be compared to the method proposed in Section 5.3 (named **VD**).

Comparison to VD and quality metric Comparing EVD to VD shows a significant improvement throughout the complete cumulative distribution function (CDF) which can be seen in Figure 5.13(a). Especially in the region of $70 - 80\%$ area overlap, an improvement of 15% can be observed. Also,

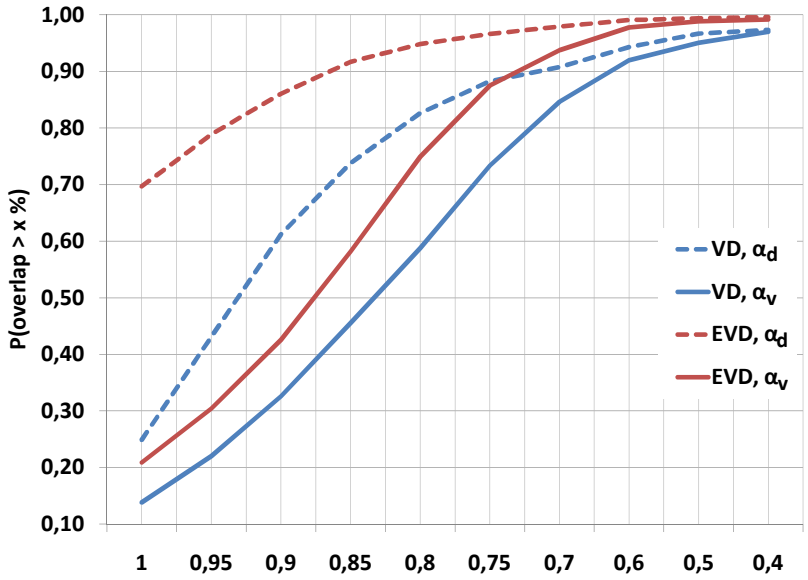
$P(overlap > x\%) > 0.9$ is now reached in $x = 72\%$ of the cases compared to $x = 62\%$ using VD. Also the distance between the centers of the detected ROIs to the true ROI is reduced significantly as can be seen in Figure 5.13(b): The detected ROIs now deviate less than 17 mm using EVD from the true position in 90% of the cases compared to 28 mm when using VD.

In Figure 5.13 both algorithms are compared using the former quality metric as well as the new overlap metric. This figure also illustrates the impact of the new way the area overlap is calculated.

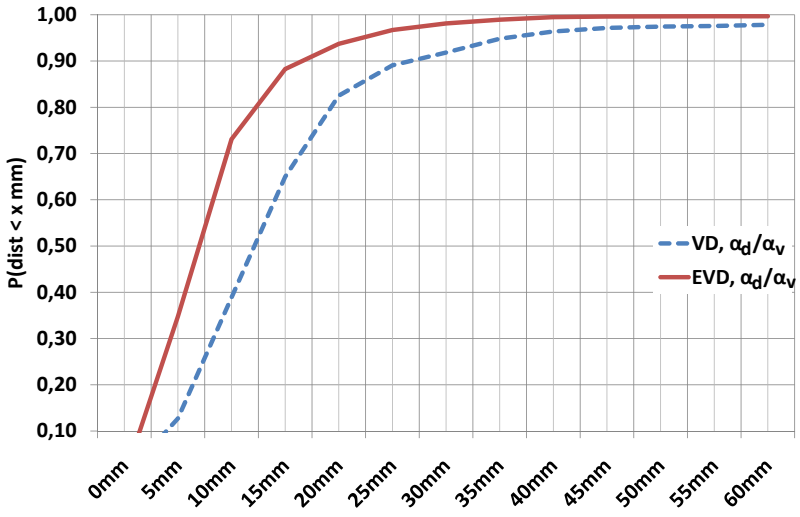
Search mask (ρ_{sm}) By applying the new region extraction, the search space in the image is reduced to an average of less than 9% of the original image's area. This is comparable to the simple approach proposed in Section 5.3, where the search space was reduced to the mid third of the patients body, which caused an average reduction of computation time to less than 21% without employing any knowledge in the database.

Refinement One of the major concerns of VD was the size of the detected ROI which was defined by the size of an average annotation box that was doubled in width and height. The refinement method introduced in Section 5.4.2 addresses this issue and is able to reduce the width, height and area of the ROI to an average of 62% (width) and 80% (height) which results in an average area decrease of 49%. Also, the 2-step refinement fails in just less than 1% of the cases where it fall backs to the size of the unmodified size of ϕ_{box}.

Figure 5.14 shows the positive impact of the refinement process on both area overlap and distance deviation. Figure 5.14(b) shows the large impact on the distance deviation in the first two columns of the diagram. Using EVD, the probability to achieve < 35 mm distance deviation is now 0.35 ($P(dist < 5\,\text{mm}) = 0.35$) compared to 0.11 using VD. Also $P(dist < 10\,\text{mm})$ was raised from less than 0.5 to more than 0.72.

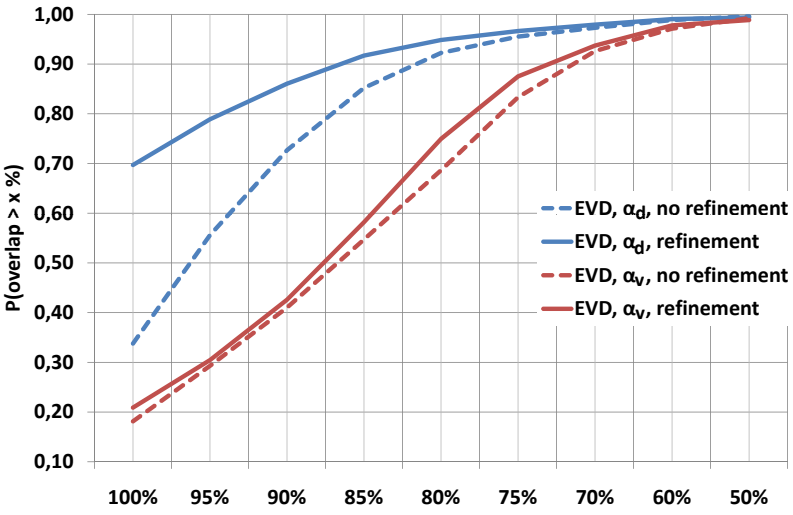


(a)

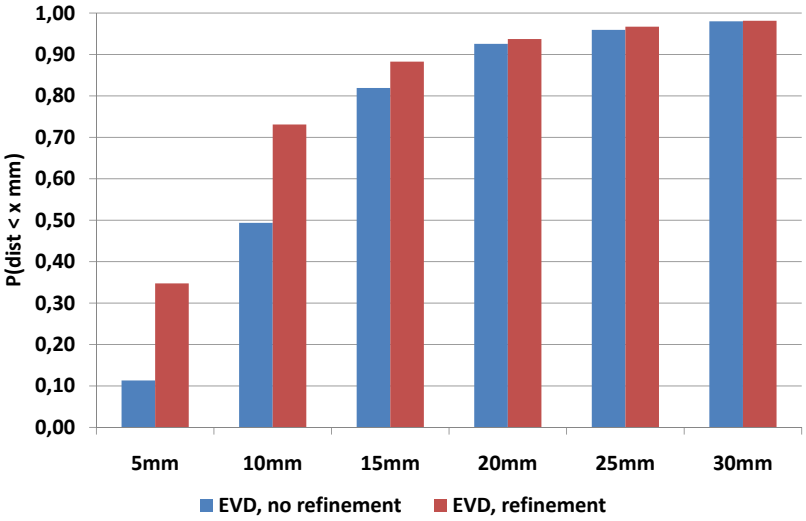


(b)

Figure 5.13: Comparing VD and EVD by using the new overlap measure (a) and distance deviation (b). In Figure (a) the difference of the dashed blue line to the solid blue line shows the impact of the stricter quality measure.



(a)

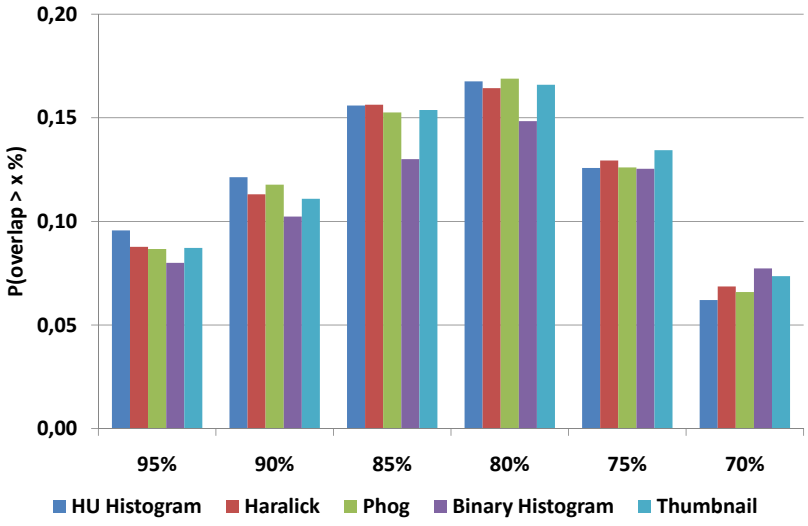


(b)

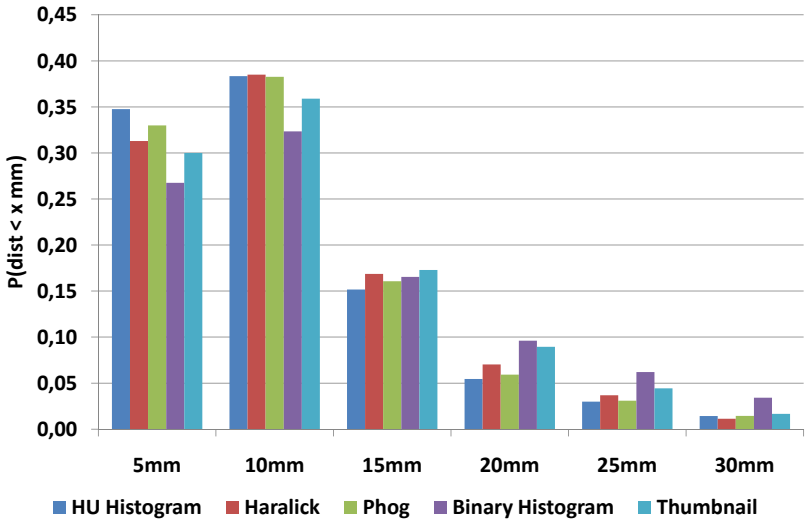
Figure 5.14: (a): Impact caused by the refining process as well as by the tighter area overlap measured by area overlap . and by distance deviation. (a): Impact on the area overlap comparing the new area overlap (red lines) with the are overlap proposed for VD (blue lines).

Feature Descriptors / Candidate selection: The evaluation of different feature descriptors ($\phi_{desc(1)}, \ldots, \phi_{desc(5)}$) mentioned in Section 5.4.2 revealed the slightly superior performance of HU-histograms ($\phi_{desc(1)}$) compared to other feature descriptors which have been proposed in the literature before. The according diagrams can be seen in Figure 5.15 where it can be seen that thumbnail features perform worst. The best features are HU histograms, followed by Haralick texture features.

An evaluation of λ and η which are both affecting the selection of candidates proved a strong stability with respect to the values for both parameters. Diagrams of the comparison of the feature descriptors and for the parameters λ and η are shown in Figure 5.16 and 5.17.

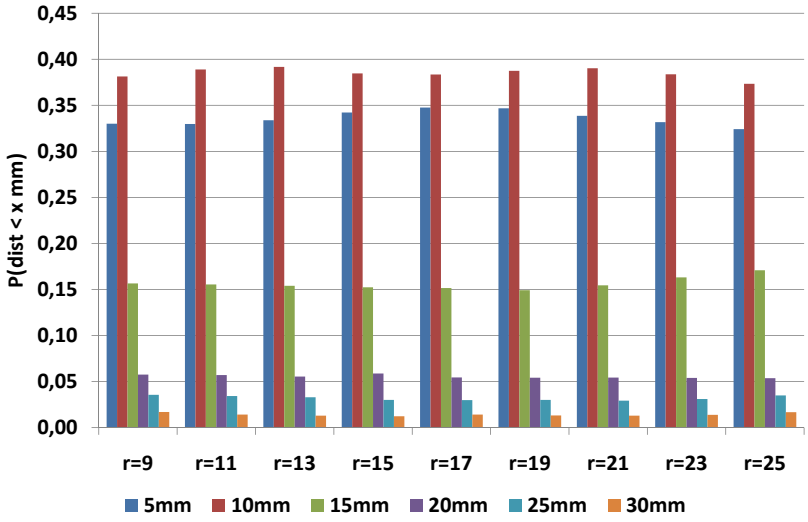


(a)

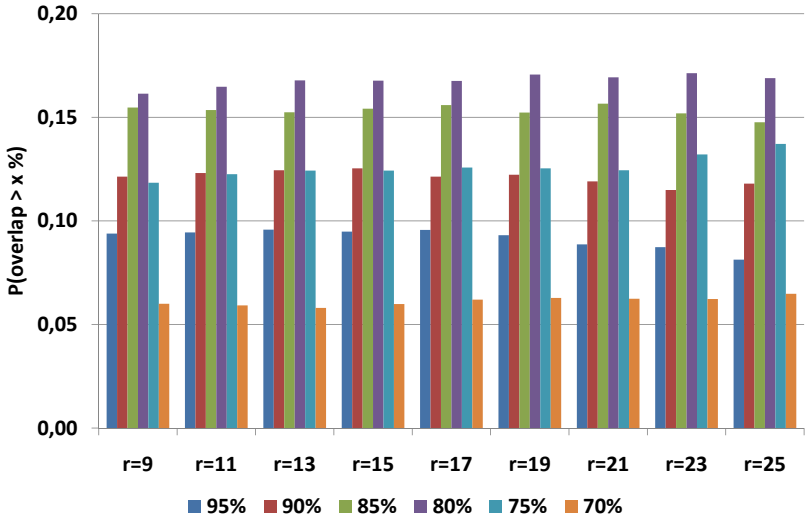


(b)

Figure 5.15: Comparison of feature types regarding both quality measures (Area overlap (a) and distance deviation (b)).

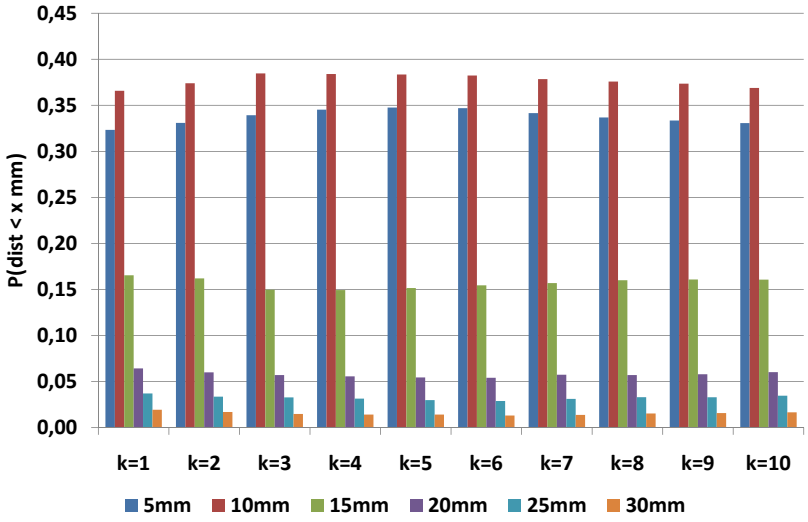


(a)

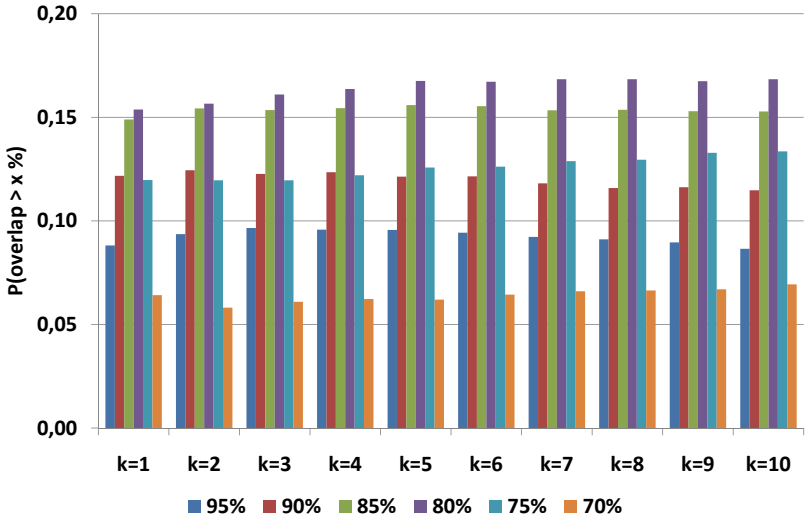


(b)

Figure 5.16: Impact of the amount of candidates (η, displayed on the x-axis) on the detection rate measured by the distance to the ground truth annotation Figure (a) and the area overlap (b).



(a)



(b)

Figure 5.17: Impact of the neighborhood size (λ, displayed on the x-axis) on the detection rate measured by the distance to the ground truth annotation Figure (a)) and the area overlap (b).

Part III

Medical Sensor Data

Chapter 6

Introduction

In the previous chapter, the importance of medical imaging was emphasized as the amount of imaging data which is currently produced by clinicians is overwhelmingly huge but yet far from being fully exploited. Another field in the medical domain that suffers from the same problem is medical sensor data. Even though imaging devices could also be regarded as sensors, this chapter does not focus on imaging techniques but addresses topics concerning physical activity of a person or patient.

It is commonly agreed on the fact that physical activity is a major factor in medical prevention, diagnosis and also in therapy. Yet the potentials of physical activity seems to be far less disclosed as it is for example the case of medical imaging. One problem for the exploitation of physical activity is of course that the recording of data is more time consuming than in the case of medical imaging. Per definition, physical activity is defined by certain motion or behavior over a certain period of time. This statement poses a set of problems:

First of all, motion per se is usually a 4 dimensional measure as it is usually the case that a 3 dimensional coordinate or change in coordinates is tracked in dependency of time. This implies of course that - in order to track an activity - hardware devices are needed for the tracking itself. This might

not be a problem if the activity is monitored and recorded in a controlled environment like in a medical rehab center or a clinic. However if activity should be recorded in a long term manner in an uncontrolled environment or in a patient's home environment with least impact to his daily life as possible, the requirements to the recording devices are much stricter in terms of size, robustness and runtime.

The second problem shows up when the data should be analyzed and interpreted. Comparable to medical imaging where a variety of imaging modalities exists, there is a large variety, how physical activity can be measured. This involves for example the position of the sensor at the body (e.g. at the leg, foot, arms or at the belt to name just a few), the measurands that are recorded (e.g. position, acceleration, pulse, skin temperature, etc.) and if the data is recorded by a single sensor or a multiple sensor systems.

Depending on the data that is recorded, a large variety of algorithms has already been developed in the past. Yet those algorithms are very specialized to the combination of recorded data and problem statement. In the case of this work, a newly developed miniaturized 3d accelerometer was used to record data at a comparatively low rate (25 Hz) while being mounted at the ankle of the test person. The location of the sensor was suggested by the use case that the location of the sensor should be as unobtrusive as possible. Also it must not feel disturbing to the person wearing the sensor even if the sensor is worn over a long time.

In Section 7 of this work, the issue was to evaluate existing work for the analysis of the data, so that a classification of activities would be possible. As it turned out that existing methods did not yield the required performance, the knowledge of existing works was combined and extended with new techniques to achieve satisfying results.

But even if a method succeeds in a given task, there is still a long way to a valid prototype. Usually a new method is tested and evaluated by using one (or more) of the established data mining frameworks as they offer a large repertoire of methods that come into consideration for solving the

problem. Yet, if such methods have to be integrated into a prototype, they usually do not offer convenient ways for an easy integration of the newly developed methods into the prototype. This means that each time, the proposed solution (a.k.a the model) is changed, some efforts have to be made to port and integrate the solution into the prototype. The more often the solution is changed, the more efforts are needed to update the integration. This leads either to the compromise of an increasing amount of cost or a reduced update rate - which are both unsatisfying solutions.

In Section 8 a software is presented which is based on an established software platform. Instead of reinventing yet another data mining framework or another demo, this work provides an integration technique for existing data mining frameworks and combines them with the power of the open platform system on which the work is based. The advantage is that algorithm development and prototype development occur in the same environment without loosing the ability to use algorithms from established frameworks. This is done mainly by providing different interfaces (like a scientist interface or a prototype-user interface) for different uses without the need to change the underlying platform.

Chapter 7

Activity Recognition

7.1 Introduction

Physical activity is not only strongly conclusive in nowadays diagnosis, yet it also plays a major role in medical prevention as well as in therapy. For example it can be important in order to prepare for certain clinical treatments to measure and rate the fitness of a patient. Also in cases of rehabilitation after surgeries or accidents it is of great importance for the recovery of the patient that he performs a controlled amount of physical activity. In case of prevention, it is also widely agreed on the fact, that physical activity is a large impact factor in oder to avoid slipped disks. Even though it is widely agreed upon the importance of physical activity, the medical potential is not yet fully exploited.

Possibly, the most important reason for this is the fact that it is very hard to detect and measure the amount and intensity of physical activity automatically and most important objectively. In order to monitor the physical activity in a long term manner, systems are needed that provide a run time of 24h per day and several days successively. Also such systems should be as small and unremarkable as possible in order to avoid stigmatization of a patient – which would instantly lower the acceptance of such systems.

Yet the requirements are not only constraint to the hardware which is recording the activity. Even if the activity signals are recorded, clinicians require methods for proper aggregation and visualization of the data as it is unfeasibly to interpret the huge amount of raw data of a long term study in a reasonable amount of time. Yet such methods must not make diagnoses totally of their own but act as a supporting tool for the clinicians, driving the process of manual interpretation into computer aided diagnosis. A first step towards this aim is the automatic detection and classification of activity on such data, so that a clinician can gain a quick overview about a patient's activity during a certain amount of time without the need of having to interpret raw data.

The following chapter describes the current state of an approach that was designed for this use case of the automatic classification of physical activity. The chapter is started by Section 7.2 which discusses related work, followed by Section 7.3 which describes the feature extraction process. Section 7.4 describes the classification method which was employed on the features. The experiments and evaluation are described in Section 7.5

7.2 Related work

The detection and classification of human activity by using data that was obtained from sensor devices is subject of several research activities. Yet there is no optimal solution for this task. An issue that might seem simple is for example the position of the sensor at the body. The authors of [112, 138, 10] have shown that the positioning can causes major implications to the detection of physical activity. Yet the findings are contrary as the positioning also depends on the type of activity that should be detected.

Nevertheless, most activity recognition techniques have a common pipeline. This pipeline usually performs some cleaning of the raw data by applying noise reduction techniques like median- or average filters [74, 77]. If the sensor data is obtained from accelerometers, it might be desired to remove the earth gravity from the raw signal as shown in [74] and [6].

Afterwards relevant features are extracted from the resulting signal. In order to apply common machine learning algorithms it is common practice to discretize the continuous data stream into time frames of data with equal length. The length of the time frames thereby varies from 1 – 10 sec [78, 88], depending on the activity that should be recognized or detected.

After the discretization of the data, the actual features are derived from the time frames. Common feature types can roughly be categorized in time domain, frequency domain, wavelet (time-frequency) and heuristic features. Time domain features are comparatively easy to compute as they can be derived directly from the data. The authors of [88, 110, 143] for example propose to use arithmetic mean and variance of the signal as features. Another characteristic feature is given by the peaks in the amplitudes of the accelerometer data. In [88] the average time between peaks is used as a feature, whereas the amount and average intensity of the peaks are used in [143]. In cases where frequency domain features should be used, it is necessary to transform the raw data into the frequency domain first. This is usually done by applying a Discrete-Fourier-Transform (DFT). The authors of [132]and [10] are then using the spectral energy of the signal as a feature. In [10] the authors propose to use the entropy of the frequency domain to differentiate between activities with very similar energy signals. In [105], time and frequency domains are not regarded separately but combined by applying a wavelet transform to the original signal. The coefficients of the transform are then proposed as features for further processing.

Heuristic features are usually computed by applying statistical and mathematical methods that involve more than a single axis of an accelerometer. The authors of [74, 6, 149, 77] propose to use the signal magnitude area (SMA) of all axes of the accelerometer as a feature in order to differentiate between dynamic and static activities. The inter-axis correlation is used amongst other features in [10] to differentiate activities concerning more than one body part.

The classification of physical activities based on the extracted features is the next step in the processing chain. In [74, 105], threshold based techniques are used. More sophisticated methods apply Decision Trees [10, 74, 65], Hid-

den Markov Models [143], Gaussian Mixture Models [6], k-Nearest-Neighbors Classifier [110, 65], Naïve Bayes Classifier [65] or Support Vector Machines [84].

7.3 Feature Extraction

Prior to extracting any features from the raw data, there is first the need to pre-process the recorded data. The accelerometers used for this work are recording data in a range of ±2 g with a rate of 25 Hz while the sensor is worn at the ankle of the patient. The limitation to ±2 g is suitable for most of the recorded activities. Yet if an activity which involves very strong accelerations or decelerations is recorded, the signal is clamped to the domain of ±2 g even though the actual signal would be much larger. In [25] for example, signals up to ±12 g were observed.

7.3.1 Signal Reconstruction

In order to compensate the loss of information caused by the technical limitation, the first preprocessing step aims at reconstructing the original signal in areas where the signal seems to be cut-off.

In order to detect such samples, the raw signal is scanned for consecutive maximum/minimum values which indicate a cut-off. Afterwards, the signal is reconstructed by considering T samples before and after this segment. For each of these T samples, the average slopes are computed. Afterwards both values are combined to obtain the estimated slope Δ_{total}:

$$\Delta_{before} = \frac{1}{T} \sum_{i=s-T}^{s} (x_{i+1} - x_i) \tag{7.1}$$

$$\Delta_{after} = \frac{1}{T} \sum_{i=e}^{e+T} (x_i - x_{i+1}) \tag{7.2}$$

$$\Delta_{total} = \frac{\Delta_{before} + \Delta_{after}}{2} \tag{7.3}$$

With x_i being a signal value at time i and s, e defining the start and end indices of the segment that should be reconstructed. Afterwards the peak signal is reconstructed by replacing all x_i in the cut-off segment by interpolated values. In cases where the cut-off segment has even length, a global extremal value within this segment is ensured by increasing one of the middle values. This procedure is illustrated in Algorithm 2.

Algorithm 2 Peak Reconstruction

$h \leftarrow \lfloor (e - s + 1)/2 \rfloor$
for $i = 1 \rightarrow h - 1$ **do**
  $x_{(s+i)} \leftarrow x_{(s+i)} + \sqrt{i}\Delta_{total}$
  $x_{(e-i)} \leftarrow x_{(e-i)} + \sqrt{i}\Delta_{total}$
end for
if $isOdd(e - s + 1)$ **then**
  $x_{(s+h)} \leftarrow x_{(s+h)} + \sqrt{h}\Delta_{total}$
else if $(|\Delta_{before}| > |\Delta_{after}|)$ **then**
  $x_{(s+h+1)} \leftarrow x_{(s+h+1)} + \Delta_{total}$
else
  $x_{(s+h)} \leftarrow x_{(s+h-1)} + \Delta_{total}$
end if

7.3.2 Segmentation

In order to apply classification techniques it is often desirable to obtain time frames with equal lengths building feature vectors with the same dimensionality. Some activities like jogging, walking or biking show a very periodic pattern when they are executed over a longer period of time. Especially in the case of biking, this is very comprehensible as pedaling causes a very cyclic movement and thus a very repetitive signal as long as the person keeps moving with the same speed. Using the standard approach, such a segment of the data would be split up into several separate time frames which are then classified. Depending on the length of the chosen time frames, this

causes a repetitive classification of the same signal. Given the information that a certain segment (which is assumed to be much larger than the time frames) contains only periodic information, it is sufficient to classify only a single time frame of this segment and thus classify the whole segment in a transitive way. However the information whether a signal is periodic or not needs to be computed. Obviously it is essential that this computation must be computationally cheap in order to make advantage of this procedure.

In this work, the autocorrelation $\rho(S, t_1, t_2)$ of a signal S at times t_1 and t_2 is used to identify the periodic patterns:

$$\rho(S, t_1, t_2) = \frac{E[(S_{t_1} - \mu_{t_1})(S_{t_2} - \mu_{t_2})]}{\sigma_{t_1}\sigma_{t_2}} \tag{7.4}$$

Hereby, S_i defines the subsequence of the recorded signal starting at time i. μ_{t_1}, μ_{t_2} and σ_{t_1}, σ_{t_2} are the mean and variance values of the subsequences starting at the according indices t_1 and t_2. The length of the subsequences is limited to a certain length. In order to detect the existence and the length of a periodic pattern in the data, Algorithm 3 is employed on the data. The required parameters are shown in Table 7.1.

Basically, the algorithm starts to search for a t_2 where $\rho(S, t_1, t_2) \geq \tau$. If τ is not exceeded within the search range of max samples, the algorithm has not found a periodic pattern for the subsequence so that a new detection is executed for a new subsequence at $t_1 + x$ with $x > 0$. Otherwise if $\rho(S, t_1, t_2) \geq \tau$, the first periodic pattern to the subsequence at t_1 was found at t_2. The distance between t_1 and t_2 is called the $shift$. The algorithm then recalculates the autocorrelation repeatedly as long as (7.5) holds. As soon as the equation no more holds, the end of the last segment is returned for which the equation did hold.

$$\rho(S, t_1, t_1 + k \cdot shift) \geq \tau \ ; \ k \in \mathcal{N}^+ \tag{7.5}$$

Afterwards t_1 is set to a position after t_2 and the process is started again.

The accelerometer in this work records a data series for each axis so that this procedure is applied for all three axes separately. Finally only those

Table 7.1: Input parameters for the algorithm used to identify periodic patterns in the recorded data.

X	Recorded data for one axis
t_1	Starting index
τ	Minimum value for the autocorrelation
max	Maximum seek range measured from t_1

Algorithm 3 Identification of periodic patterns

Require: X, t_1, τ, max
  $shift = 0$; $\rho = 0$
  while $\rho < \tau$ **and** $shift < max$ **do**
    $shift \leftarrow shift + 1$
    $\rho \leftarrow \rho(X, t_1, t_1 + shift)$
  end while
  $t_2 = t_1$
  if $\rho \geq \tau$ **then**
    while $\rho \geq \tau$ **do**
      $t_2 \leftarrow t_2 + shift$
      $\rho \leftarrow \rho(X, t_1, t_2)$
    end while
  end if
  return t_2

segments are marked as periodic signals, where the algorithm described above detected periodic signals in each axis. At the end of the segmentation process, the complete data series is marked with segments which contain periodic data or aperiodic data.

7.3.3 Feature Extraction

The aim of this step is to derive feature vectors from the data segments. This feature vector represents a concatenation of 5 different features which will be

explained in the following. Unless mentioned otherwise, a reference to the term acceleration can be regarded equivalent to the term deceleration within the next sections.

Auto Regression Coefficients (ARC)

Autoregressive (AR) models are used in signal processing and statistics in order to model certain kinds of random processes. The value x_t of an autoregressive model at time t is thereby defined by

$$x_t = \sum_{i=1}^{p} a_i \cdot y_{(t-i)} + \epsilon_t \tag{7.6}$$

with p denoting the order of the AR model, a_i denoting the i-th AR coefficient, $y_t \in Y$ the value of the process at time t and ε_t denoting the white noise at time t. Given the data Y, the values of a_i can be estimated in various ways, for example by using the common least squares method. In this work, AR models with an order of 3 were computed for each axis separately. The resulting $3 \times 3 = 9$ coefficients represent the first 9 values of the final feature vector.

Signal Magnitude Area (SMA)

The signal magnitude area is a simple, yet common measure in the area of activity classification. The SMA is defined by the normalized sum of absolute values over all axes recorded from the accelerometer divided by the length of the regarded segment. In case of a 3 axes accelerometer with the signal obtained from the axes denoted by x, y and z, the SMA of a segment with length N is defined by:

$$SMA = \frac{1}{N} \sum_{i=1}^{N} (|x_i| + |y_i| + |z_i|) \tag{7.7}$$

As there is no normalization regarding the maximum peak values, larger amplitudes of the recorded signals obviously have direct impact on the value

of the SMA. Thus, activities with large accelerations will usually produce higher values for the SMA than slower activities with less accelerations.

Tilt Angle (TA)

The 3d accelerometer in this work is intended to be worn always in the same position at the ankle. Therefore, the tilt angle of the sensor also describes the tilt angle of the lower leg. The position and movement of the lower leg are used as an additional description of the activity that is currently performed by the patient. The value used for the feature vector describes the angle between the positive vector of the earth's gravitational field and the z-axis of the accelerometer: $v = arccos(z)$.

Average Peak (AP)

Another feature that is proposed in this work is the intensity of the average peak values of a segment S. In contrast to the tilt angle, the average peak is calculated for each axis separately. In the following, the term peak refers to both negative and positive peaks. To reduce the effect of outliers, S is first convolved (smoothed) with a kernel k of size 3 ($k = 1/3, 1/3, 1/3$).

In order to determine the peaks of a segment, the algorithm first starts with the determination of an absolute global peak s_{max} with respect to the absolute values of the signals in S:

$$s_{max} = \{s_t \in S \mid \forall s_i, s_t \in S \wedge t \neq i : |s_i| \leq |s_t|\} \tag{7.8}$$

With s_t and s_i denoting the intensity values of the samples at time points t and i respectively. Afterwards the threshold value τ_{max} is initialized with $\tau_{max} = 0.98$

In the second step, the algorithm identifies all samples S_{peaks} with an intensity larger or equal than the absolute value of $\tau_{max} \cdot s_{max}$:

$$S_{peaks} = \{s_{t_i} \in S \mid |s_{t_i}| \geq \tau_{max} \cdot |s_{max}|\} \tag{7.9}$$

After this, S_{peaks} may contain neighboring samples belonging to a single peak. Therefore S_{peaks} is filtered in the next step. If subsequent samples are less then 10 samples apart from each other, only the larger one is retained while the other one is discarded.

If this procedure detect less than 3 peaks τ_{max} is lowered by 0.02 and the peak detection is repeated until either 3 or more peaks were identified or until $\tau_{max} = 0.7$.

If more than 3 peaks were detected ($|S_{peak}| \geq 3$), the feature value for this axis is determined by the mean value of all samples in S_{peak}. Otherwise, if the latter condition is reached ($\tau_{max} = 0.7$), it is assumed that S does not contain significant peaks and the feature value is assigned $+1$ if the mean value of the samples in S_{peak} is greater than 0 or -1 otherwise. The process of peak detection is outlined in Algorithm 4.

Surrounding Segmentation Rate (SSR)

During the evaluation of the segmentation process, it was experienced that different activities show a very different ratio between periodic and aperiodic segments. This lead to the assumption, that the information about the ratio is also an informative feature for the differentiation between different activities. To compute the SSR, a time frame of 60 s surrounding the current segment is examined. The ratio of signals located in periodic segments then realizes the value of the SSR feature.

Features from Periodic Segments

Due to the information that the signal is periodic it is sufficient to extract a single feature vector for the complete segment. After the classification of this single feature vector, the result can be transferred to the complete segment. This has the advantage that the computation for the feature vectors has only to be done once for a complete segment. Another advantage is that it is possible to represent large amounts of raw data with just a single feature

Algorithm 4 Peak Detection

Require: S
  $S \leftarrow$ convolve S with (1/3, 1/3, 1/3)
  $s_{max} \leftarrow \arg\max_{s_i \in S} S$
  $S_{peaks} \leftarrow \{\}$
  $\tau_{max} = 1$
  while $|S_{peaks}| < 3 \wedge \tau_{max} > 0.7$ **do**
    $\tau_{max} \leftarrow \tau_{max} - 0.02$
    $S_{peaks} \leftarrow \{s_i \in S \mid |s_i| \geq |s_{max}| \cdot \tau_{max}\}$
    for all $s_i, s_j \in S_{peak}$ **do**
      if $|i - j| \leq 10$ **then**
        $S_{peaks} \leftarrow S_{peaks} - \arg\min\{|s_j|, |s_i|\}$
      end if
    end for
  end while
  $avg \leftarrow$ average of all $s_i \in S_{peaks}$
  if $|S_{peaks}| \geq 3$ **then**
    return avg
  else
    return $avg/|avg|$
  end if

vector. For example, a periodic segment with a length of 1 min consists of 4 500 samples[1]. After feature extraction, a single feature vector with just 15 values remains which results in a compression factor of 300.

Features from Aperiodic Segments

Aperiodic segments require a more detailed examination as it is not feasible to extract a single feature vector for the complete segment as it is likely that it consists of different activities. In this case, the segment is split into sub

[1] $1\,\text{min} \rightarrow 60\,\text{s} \cdot 25\,\text{Hz} \cdot 3\,\text{axes} = 4\,500$

segments with a size of 80 samples. Afterwards, a feature vector is computed for each sub segment. If the last segment is larger than 40 samples, another feature vector is computed for this sub segment as well.

7.3.4 Linear Discriminant Analysis

Before applying the classification process, a Linear Discriminant Analysis (LDA) is applied to the features with the aim to decrease the scatter within a class and to increase the inter class distance of the feature vectors. This step is necessary because the features were extracted from the raw signals ignoring the fact that different people show a different flow of movement even if they perform the same physical activity. Different flows of motion between different persons can for example be caused by different body heights or the the speed at which the activity is performed. Also a trained person might show a different flow of motion compared to an untrained person. Last but not least, the position of the sensor is also an impact factor. Even if the position is restricted to the ankle, the rectangular sensor can still be rotated by 180° in each axis and of course the sensor can change the position over time if it is not fixed accurately.

7.4 Classification

7.4.1 Classifying Features

The classification is used to assign class labels to the feature vectors and thus to the segments or sub segments from which the feature vectors were extracted. In this work, Naïve Bayes classifiers were applied to perform the classification. In contrast to other works, there is not just a single classifier but one classifier for features extracted from periodic segments and one for feature vectors from aperiodic segments. Except the differentiation of the input data, both classifiers are trained and used in the exactly same way. In the end, almost all samples from the raw data are assigned a class label. Samples for which no feature vector was extracted (for example because they are at the end of an aperiodic segment), remain unlabeled in this step.

7.4.2 Reclassification

The aim of this step is to classify all samples which have not yet been assigned a class label and also to compensate classification errors from the step above. If for example the "*biking*" is recognized over a period of several minutes with some seconds "*elliptical training*" in between, it is very likely that a classification error occurred. The reclassification is executed for each sample s_i of the data set in the following way:

First, a weighted class label histogram $H_{classes}$ is created for the period of $s_i \pm d_{max}$ samples, with $d_{max} = 375$ which corresponds to $\pm 15\,\mathrm{s}$. Samples that are not yet assigned a label are temporarily assigned the class *unclassified*. Depending on the distance to s_i, a sample s_j contributes $d_{max} + 1 - |i - j|$ to $H_{classes}$ if the sample does not belong to the class *unclassified*.

The second step is the actual reclassification: In case of s_i being an unclassified sample, s_i is assigned the class with the highest value in $H_{classes}$. Otherwise the class label is only changed, if the class with the highest value

in $H_{classes}$ has a ratio of more than 50 % compared to all other classes in $H_{classes}$.

7.5 Experiments

7.5.1 Data

The data used for the experiments in this work were obtained from different sensors donated from the Sendsor GmbH. All sensors are identical in construction and record data at a rate of 25 Hz by using a built-in 3D accelerometer. The accelerometer records the acceleration in all 3 axes in a range of ± 1 g with a resolution of 128 units ($\hat{=}1/128$ g).

Different people recorded their physical activities independently of each other in their home environment. The speed and intensity of the activity was not restricted. Also the only requirement was to wear the sensor upright at the ankle of a foot without specifying the top or bottom of the sensor. The test persons were all in the age between 20 and 35, with different gender, different fitness levels and different body height, so that a rather diverse data set was obtained. In total, 22 different sequences with a total duration of more than 10 hours were obtained for 5 different activities (cf. Table 7.2).

Table 7.2: Data set description

Activity	# of Sequences	Duration (hh:mm:ss)
elliptical trainer	3	00:55:07
walking	6	02:42:09
inline skating	3	01:55:50
jogging	6	02:37:13
biking	4	02:32:46
total	22	10:43:05

7.5.2 Evaluation

All the following evaluations were performed by executing a Leave-One-File-Out validation. This type of validation was preferred over a regular cross-validation in order to avoid having samples and segments from one sequence in the test and training data. The over all performance is compared to a recent work shown in [77].

Feature Evaluation

The amount of features that are used and described in appropriate literature is very large. To avoid picking a random set of features which possibly includes features that do not contribute to the detection rate, a set of 8 features shown in Table 7.3 was implemented and evaluated by using a Forward-Backward-Search [152] that avoids an exhaustive search through all possible feature combinations. This evaluation was done separately for periodic and aperiodic

Table 7.3: Candidates for the feature selection process. The last two columns indicate the features that were used in [77] and in this work.

Name	Abbrev.	Features	in [77]	here
AR Coefficients	ARC	9	✓	✓
Signal Magnitude Area	SMA	1	✓	✓
Tilt Angle	TA	1	✓	✓
Average Peak	AP	3	-	✓
Surrounding Segm. Rate	SSR	1	-	✓
Inter Axis Correlation	IAC	3	-	
Arithmetic Mean	Mean	3	-	
Variance	VAR	3	-	

segments. In the end, this evaluation showed that a combination of 5 features (ARC, SMA, TA, AP, SSR) from the candidate set performs best.

Classification

The decision for the Naïve Bayes classification was the result of an extensive comparison of 32 classifiers provided by WEKA [61] including two artificial neuronal nets (ANN) (one from [61] and a more sophisticated ANN from the Encong framework[2]). All the classifiers were tested with their default parameters. ANNs were tested with 10 neurons and 5 output neurons (according to the amount of activities/classes).

The result of this evaluation showed a superior performance of 97.18 % in case of the comparatively easy Naïve Bayes classifier. The second best result of 96.67 % was obtained from using sequential minimal optimization (SMO) [111], followed by 94.88 % by a normalized Gaussian radial basis function network. A more detailed listing of the classification results of the Naïve Bayes classifier can be seen in Table 7.4.

Table 7.4: Classification rate by using the Naïve Bayes classifier without reclassification.

Inline	Elliptical Tr.	Jogging	Walking	Biking	Average
98.20 %	97.76 %	97.79 %	95.29 %	97.57 %	97.18 %

Signal Reconstruction

The impact of the signal reconstruction step described in Section 7.3.1 showed the largest impact in the activities *inline skating* and *walking* where the classification rate was improved by 2 %. In the *jogging* class, the performance was increased by about 0.5 % while the effect at *biking* was negligible and the *elliptical trainer* class even showed a slightly decreased classification rate (0.5 %). Over all, the classification was increased by 0.9 %, so that the signal reconstruction was retained.

[2] http://www.heatonresearch.com/encog

Segmentation

The impact of the segmentation process described in Section 7.3.2 was most significant for the activities *inline skating*, *walking* and *biking*. In these classes the classification rate increased each by more than 5 %. In that case, the SSR feature could of course not be used, so that the feature vector only consisted of 14 instead of 15 features. In average, an increase of the classification rate of 4.47 % was observed (from 92.71 % to 97.18 %).

Linear Discriminant Analysis (LDA)

Excluding the application of the LDA to the features had a very large impact to the classification of the *elliptical trainer* data where the classification decreased to about 50 % without an LDA while the classification of the remaining activities remains almost unchanged. Without the LDA, the *elliptical trainer* data was mis-classified either as *walking* or as *biking* which show indeed very similar flows of motion.

Reclassification

Applying the reclassification as described in Section 7.4.2 showed a positive effect on all activities so that the over all classification rate could be raised by another 1.63 % to a final accuracy of 98.85 %. Yet it should be kept in mind that if changes of activities occur within less than a minute, the reclassification step could also decrease the classification as it uses data of the surrounding 0.5 min to classify yet unclassified samples but also to reclassify samples.

Comparison to Reference

Last but not least, the method proposed in this chapter is compared to a reimplementation of the work of Khan *et al* [77] (referred to as KLLK in the following) where an accuracy of 97.9 % is reported. However, the sensor used

in KLLK is not mounted at the ankle but at the chest so that very different signals are recorded by the accelerometer.

Using the data set described above, KLLK did not yield the expected performance, so that a resilient propagation algorithm instead of a back propagation algorithm for the training of the neuronal nets as this produced better results. After this change, KLLK achieved an accuracy of 97.1 % and 93.6 % in case of the elliptical trainer and jogging classes respectively. In the cases of inline skating and walking the performance was about 80 % and biking was only classified correctly in less than 75 % of the cases, so that an average accuracy of 85.31 % was achieved.

This means that the algorithm proposed in this work performs about 13 % better than KLLK, regarding all classes. Also, the stability of the classification is much higher for the proposed algorithm with a minimum of 96.83 % in case of walking compared to 80.1 % and 74.94 % in case of KLLK. An overview comparing both algorithms can be seen in Table 7.5.

Table 7.5: Comparison of the classification results for the test data using KLLK to the work proposed in this chapter.

	skating	ell.trainer	jogging	walking	biking	average
KLLK	80.85 %	97.10 %	93.56 %	80.10 %	74.94 %	85.31 %
this work	99.37 %	98.73 %	99.36 %	96.83 %	99.94 %	98.85 %
Δ	18.52 %	1.63 %	5.8 %	16.73 %	25.00 %	13.54 %

Chapter 8

Knowing: A Generic Data Analysis Application

8.1 Introduction

Supporting the data mining process by tools was and still is a very important step in the history of data mining. By the support of several tools, scientists are nowadays able to apply a diversity of well known and established algorithms on their data for quick comparison and evaluation. In the past years, several data mining frameworks like ELKI [2], MOA [21], Weka [61] or RapidMiner [96] have been presented and established (among many others). Although all frameworks perform data mining in their core, they all have different target groups:

WEKA and MOA provide both algorithms and graphical user interfaces (GUIs). By using these GUIs, the user can analyze data sets, configure and test algorithms and visualize the outcome of the according algorithm for evaluation purposes without needing to do some programming. As the GUI cannot satisfy all complex scenarios, the user still has the possibility to use the according APIs to build more complex scenarios in his own code. RapidMiner integrates WEKA and provides powerful analysis and reporting functionalities

which are not covered by the WEKA GUI itself. RapidMiner also provides an improved GUI and also defines an API for user extensions. Both RapidMiner an WEKA provide some support to external data bases. The aim of ELKI is to provide an extensible framework for different algorithms in the field of clustering, outlier detection and indexing with the main focus being set on the comparability of algorithm performance. Therefore, single algorithms are not extensively tuned to performance but tuning is done on the application level for all algorithms and index structures. Same as the other frameworks, ELKI also provides a GUI so that no programming is needed for the most basic tasks. The framework also provides an API that supports the integration of user-specified algorithms and index structures.

The above frameworks are providing support for the process of quick testing, evaluating and reporting very well and all define APIs in different depths so that programmers can incorporate own algorithms into the systems. But even though all frameworks are based on Java, none of them makes use of a standardized plug-in system like OSGi[1], Java Plugin Framework (JPF) or Java Simple Plugin Framework (JSPF). This has the disadvantage that each implementation of an algorithm is specifically adapted to the according framework without being interchangeable.

In cases where the requirements enforce a rapid development from data mining to a representative prototype or to an early release of the software, these unstandardized plug-in systems can cause a significant delay which is caused by the time which is needed to incorporate the algorithms into the prototype.

Knowing (*Know*ledge Engineer*ing*) aims at providing a framework that bridges the gap between the data mining process and the process of rapid prototype development. This is achieved by using the standardized OSGi architecture[2] so that algorithms can be packed in OSGi resource bundles. This offers the possibility to either create brand new algorithms from scratch as well as the possibility of importing existing algorithms from other data

[1]OSGi: http://en.wikipedia.org/wiki/OSGI
[2]Eclipse Equinox: http://www.eclipse.org/equinox/

mining tools. In the latter case, the imported algorithms are wrapped and packed together into a separate bundle. Such bundles are then registered as independent service providers to the *Knowing* framework. In either case, algorithms are wrapped into Data Processing Units (DPU) which can be configured via GUI controls.

The advantage of these OSGi compliant bundles is that they are not restricted for a use in *Knowing* but can be used in any OSGi compliant architecture like the Eclipse Rich Client Platform (RCP) or the NetBeans RCP. This means that *Knowing* does not provide yet another plug-in system. Instead *Knowing* provides the possibility to use the DPUs contained in the system but also to use them in any other OSGi compliant architecture. As dependencies between resource bundles have to be modeled explicitly, it is much easier to extract certain bundles from the system than in other systems.

This loose coupling is not only an advantage in case where algorithms should be ported between completely different systems but also if the GUI should be changed from a data mining view to a prototype view for the productive system. *Knowing* itself is based on the established and well known Eclipse RCP[3] system, so that the GUI can be changed very easily in order to change the data mining view into a system or end user view.

This can be done by either using the resource bundles containing the DPUs, or by directly extending *Knowing* itself. As Eclipse Equinox itself is designed as an RCP using OSGi, it is comparatively easy to unregister the original *Knowing* interface and replace it with an interface representing the final application.

In this scenario, the *MedMon* system is presented which itself extends *Knowing*. In the developer stage, it can easily be switched between the scientific data mining view and the views which will be presented to the end users later on. As *MedMon* is intended to be used by different target groups (physicians and patients), it is desired to use a single base system for all views and only deploy different user interface bundles for each target

[3]Eclipse RCP: http://www.eclipse.org/platform/

group. This way, the data mining process can seamlessly be integrated into the development process by reducing long term maintenance to a minimum as only a single system with different interface bundles has to be kept up to date and synchronized instead of a special data mining tool, a physician tool and the patient tool.

Thus, the *Knowing* framework provides: a standardized plug-in system based on OSGi, a pluggable GUI based on Eclipse RCP, a growing amount of wrappers for algorithms of common data mining frameworks.

The advantage for the user lies in the improved possibility to integrate and exchange algorithms as well as to easily perform pre- and post-processing of data. Thus, *Knowing* does not not provide yet another data mining tool with an open API but provides an exchange format for algorithms and algorithm wrappers to combine different tools for faster prototype development. Besides this, *Knowing* offers a simple, yet powerful user interface, a bundled embedded database as data storage, extensible data mining functionality, extension support for algorithms addressing different use cases and a generic visualization of the results of the data mining process. Details of the architecture will be given in Section 8.2. A demo system dealing with temporal sensor data will be explained in Section 8.3.

8.2 Architecture

In the following the architecture of the *Knowing* framework is described which consists of a classical three tier architecture comprising data storage tier, data mining tier and GUI tier:

8.2.1 Data Storage

The data storage tier of *Knowing* provides the functionality and abstraction layers to access, import, convert and persist the source data. The data import is accomplished by an import wizard using service providers, so that importing

data is not restricted to a certain format.

In the case of the *MedMon* application for example, a service provider is registered that reads binary data from a 3D accelerometer[137] which is connected via USB. The data storage currently defaults to an embedded Apache Derby database[4] which is accessed by the standardized Java Persistence API (JPA & EclipseLink). This has the advantage that the amount of data being read is not limited by the computers memory and that the user does not have to set up a separate data base server on his own. However, by using the JPA there is the possibility to use more than 20 elaborated and well known data base systems which are supported by this API[5]. An important feature in the data storage tier arises from the possibility to use existing data to support the evaluation of newly recorded data, e.g. to apply certain parts of the data as training sets or reference results.

8.2.2 Data Mining

This tier includes all components needed for data mining and data analysis. OSGi bundles containing implemented algorithms are available fully transparently to the system after the bundle is registered as a service provider.

Algorithms are either implemented directly or wrapped in Data Processing Units (DPUs). Following the design of WEKA, DPUs represent filters, algorithms or classifiers. One or more DPUs can be bundled into an OSGi resource bundle which is registered into the program and thus made available in the framework. Bundling algorithms enforces a pluggable and modular architecture so that new algorithms can be integrated and removed quickly without the need for extensive dependency checks. The separation into bundles also provides the possibility of visibility borders between bundles so that separate bundles remain independent and the danger of building an unmaintainable system where everything depends on everything else is very

[4]Apache Derby: http://db.apache.org/derby/

[5]List of supported databases:
http://wiki.eclipse.org/EclipseLink/FAQ/JPA

low.

The modularity also provides the possibility to concatenate different algorithms into processing chains so that algorithms can act both as sources and targets of processed entities. Raw data for example first could pass one or more filtering components before being processed by a clustering component.

Creating a processing chain (model) of different, concatenated algorithms and data-conditioning filters is supported by GUI controls, so that different parameters or concatenations can be tested easily. After a model has proved to fit the needs of a use case, the model can be bundled and later be bound to other views in the GUI so that porting, adapting and integration costs are minimized to binding components and models together without porting and adapting algorithms etc from different APIs.

This architecture provides the possibility to integrate algorithms from other sources like [2, 21, 61], so that existing implementations can be reused without having to re-implement all algorithms from scratch. This also provides the possibility to replace components by different implementations quickly if performance or licensing issues require to do so.

In the data mining part of the application, *Knowing* not only supports plain Java but also relies on the use of the Scala programming language. Scala is a functional and object oriented programming language which is based on the Java Virtual Machine, so that it seamlessly integrates into *Knowing*. The advantage of Scala in this part of the application lies in the easy possibility of writing functional code shorter than in regular Java code. By using the Akka actor-model[6] it is easy to create processing chains which are executed in a parallel way so that *Knowing* can make use of multi core systems.

[6]Project Akka: http://akka.io/

8.2.3 User Interface

Using the well established Eclipse RCP and its powerful concept of views enables developers to easily replace the view of the data mining scientists with different views for end users or prototypes. Thus, the task of porting data mining algorithms and the data model to the final application is replaced by just switching the view component and binding model and GUI components together.

8.2.4 Modularity

As mentioned before, *Knowing* is based on Eclipse and is organized in different bundles. This brings the great advantage that data minders and developers can take two different ways towards their individual goal: If they start a brand new RPC based application, they can use *Knowing* out of the box and create the application directly on top of *Knowing*. The more common case might be that an RPC or OSGi based application already exists and should only be extended with data mining functionality. In this case, only the appropriate bundles are taken from *Knowing* and integrated in the application.

8.3 MedMon

A prototypical implementation based on *Knowing* is *MedMon*. *MedMon* (*Med*ical *Mon*itoring, cf. Figure 8.1) is motivated by following a real-world use case where the convalescence of patients should be monitored by analyzing their daily physical activity as presented by the works of [136] and [137]. Among others, features like quality, intensity and amount of physical activity are diagnostically strongly conclusive as they have major influence on medical prevention, convalescence and therapy.

Physical activity in this case includes various types of motion like walking, running and cycling. The task is to perform data mining on long-term

temporal sensor data which is provided by people wearing a little 3D sensor which is recording and storing acceleration data in all three axes with a frequency of 25 Hz. When the sensor is connected to a computer, the data is parsed and transferred to the *Knowing* framework, where it is stored in a database. *Knowing* is able to deal with different types of time series which are not limited to the medical field but can be applied to different types of scenarios where time series data is being produced and needs to be analyzed. Analyzing the data in this use case means the application of clustering and classification techniques in order to detect motion patterns of activities and thus to separate the different types of motions. Available algorithms as well as additionally implemented techniques for data mining and the pre-conditioning of the temporal data (e.g. filtering of specific information, dimensionality reduction or removing noise) can efficiently be tested and evaluated on the data and furthermore applied to the data by taking advantage of the OSGi compliant architecture (cf. Section 8.2). By using the standardized OSGi plug-in system, well-known data mining tools are integrated so that the re-implementation of already tested algorithms can be avoided. The requirement of a quick migration of the final data mining process chain to a prototype system is accomplished by using different graphical views on a common platform. Thus, neither the process model nor the algorithms needs to be ported. Instead, only a different view of the same base model needs to be activated to enable the prototype. Finally, *MedMon* provides a generic visualization model to present the results of the data mining process to the user.

By using the *MedMon* application, the user can import 3D acceleration data from the hardware sensor into a database. Working with *MedMon*, the user is enabled to switch between different roles (cf. Figure 8.1 and 8.2). The prototype allows several views on the recorded data and the results of the data mining process: Such views are for example the *data mining view*, where DPUs can be combined to processing chains and which allows to employ newly developed algorithms; the *physician view*, which provides a more detailed view on the data for multiple users' activities as well as the possibility to add and modify electronic health records; and the *patient view*, which displays

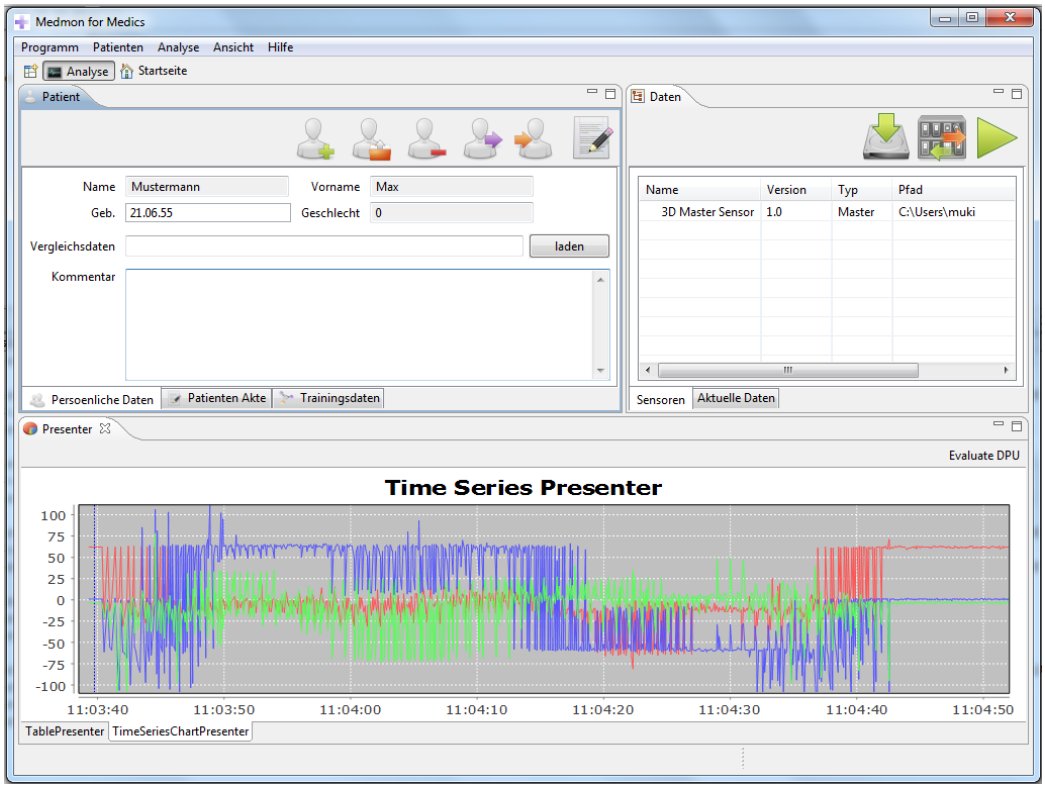


Figure 8.1: The *MedMon* prototype GUI, using components of *Knowing*.

only a very brief summarization of the patient's daily activity in order to give feedback to the user about his achieved activity pensum each day (this view is currently planned).

In the presented use case, the daily activity can be analyzed, processed and long-term data analysis can be performed by using an aggregated view of the results of the data mining process from the physician view and the patient view.

The *MedMon* prototype system is not limited to medical applications but provides a valuable tool for scientists having to deal with large amounts of time series data. The source code of the *Knowing* framework and the *MedMon* prototype in its current state are available via GitHub[7].

[7]GitHub project page: https://github.com/knowing/

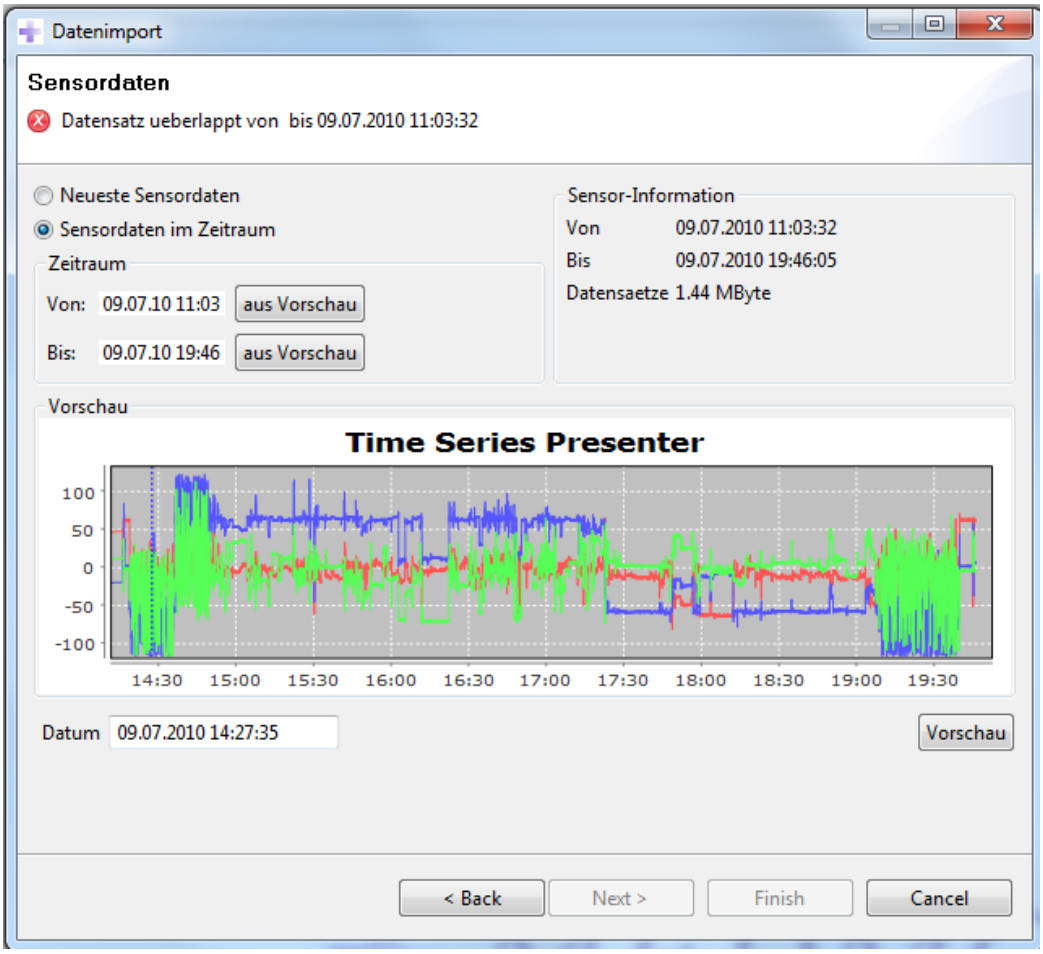


Figure 8.2: The Sensor Import Wizard of *MedMon*.

Part IV

Indexing

Chapter 9

Introduction

In the previous chapters, features of different kinds have been introduced. Features and Feature vectors thereby always played a central role in the algorithms which is not very surprising as all kinds of signals (time series, images, etc.) that have been the subject of the observation first have to be transformed into a representation, so that the according algorithms can be applied. In these cases, the feature vectors were usually represented as vectors in an N-dimensional space. The features which are building the vector thereby were either floating point values or integers. In either case, the feature extraction process aimed at extracting relevant subsets of the original data.

Yet, this process has a very convenient side effect in that the data which an algorithm has to treat is also reduced. In the case of image features for example, a 512×512 image consists of 262 144 pixels. In other words, the original data space comprises more than 250 000 dimensions.

Creating a feature extraction process for obtaining the so called relevant subsets can thus also be seen as a transformation of the very high dimensional original data space into a space with much lower dimensionality. If for example a 144 dimensional feature vector is extracted from a 512×512 image, this can also be seen as dimensionality transformation and (even more important) reduction step that reduces the data space by a factor of more than 1 800.

This does not only have the effect that the resulting feature vector can act as a tool for solving certain problems, but it also means that for such a problem, the data could be compressed by an impressing factor of more than 1 800. Even though the data space was already compressed so far, it is not a secret that feature spaces with more than some dozen dimensions can hardly be indexed in a way, so that the application of an index structure really enhances nearest neighbor queries compared to a sequential scan over all the feature vectors. Of course, there exist some specialized solutions like for example BOND [34] that can index several hundred dimensions. Nevertheless most of such approaches are not generally applicable and pose some restrictions. Such a restriction can for example contain the constraint to only a certain kind of data distribution. This also leads to the fact that especially image retrieval and image similarity tasks often use either no index structure at all and iterate across the complete data set or employ queries and hashing techniques [80, 69, 98, 7].

This is acceptable in use cases like search engines where for example 20 very similar but not necessarily the most similar images w.r.t the query image should be returned. Yet there are use cases, where hashing is not acceptable or simply not desired – like in medical imaging. Assuming that the data set to be processed does not fit to any specialized solutions, the task of similarity search leads into a dilemma where one can either perform an exact but slow search (which is usually realized by a full scan over the data) or an approximate and fast search.

This problem motivated the works which are presented in the following sections. In 2002, BOND [34] was presented by using a column storage data base instead of a row storage model. This promising approach was followed and extended and finally ended in the work presented in BeyOND. Also, the growing availability of solid state disks (SSDs) with dramatically different access times have started to show some impact on the design of data bases and index structures, as some paradigms that have been driving the according models no longer hold for these kind of storage devices.

The techniques and methods proposed in the following chapters, take

advantage of these changes. These works are published in [18] and [39]. Several findings of Section 10 were also influenced by previous works done in [17] and [16] which are not part of this thesis.

Chapter 10

BeyOND – unleashing BOND

10.1 Introduction

It is common opinion, that similarity search in high-dimensional data is inherently difficult. The reasons for this finding, however, are not that widely agreed upon. For example, it has been stated that high-dimensional similarity search facilitated by partitioning or clustering based data structures cannot beat the sequential scan [144]. This has been backed but also relativized by some mainly theoretical studies [20, 14, 66, 4, 46, 66]. The essence of these studies for research on data structures is: it depends on the characteristics of the data distributions whether an index-based method is more suitable than a sequential scan-based method or vice versa. This may not seem impressively enlightening but, surprisingly enough, this key message has been neglected in many research contributions over the last decade [3, 75, 15, 72, 36, 5]. Thus, it still appears to be well worth noting that nearest neighbor search is meaningful if and only if the nearest neighbor of the arbitrary query object is sufficiently different from its farthest neighbor. This is in general the case whenever a data set exhibits a natural structure in clusters or groupings of subsets of data, e.g., when the data is generated by several different distributions. It is, however, not well studied which impact the relation of relevant versus irrelevant attributes in a data space has. How do data

structures behave, if the grouping of data is evident only in subspaces (built by the "relevant" attributes) of the original data space whereas "irrelevant" attributes do not contribute to discerning the different data groups from each other. Furthermore, if there exist several clusters within a data set, some attributes can be relevant for some clusters (i.e., useful for separation of these clusters) and at the same time irrelevant for other clusters. These important differentiations have been elaborated recently in [67, 19]. While much effort has been spent on studying possibilities to facilitate efficient similarity search in high-dimensional data, scarcely ever the question arose how to support similarity search when the similarity of objects is based on a subset of attributes only. Aside from fundamentally studying the behavior of data structures in such settings, this is a practically highly relevant question. It could be interesting for any user to search, e.g., in a database of images represented by color-, shape-, and texture-descriptions, for objects similar to a certain image where the similarity is related to the shape of the motifs only but not to their color or even the color of the background. An online-store might like to propose similar objects to a customer where similarity can be based on different subsets of features. While in such scenarios, meaningful subspaces can be suggested beforehand [83, 64], in other scenarios, possibly any subspace could be interesting. For example, for different queries, different regions of interest in a picture may be relevant. Since there are 2^d possible subspaces of a d-dimensional data set, it is practically impossible to provide data structures for each of these possible subspaces in order to facilitate efficient similarity search. Another application where efficient support of subspace similarity queries is required are many subspace clustering algorithms [86] that rely on searching for clusters in a potentially large number of different subsets of the attributes. If efficient support of subspace range queries or subspace nearest neighbor queries were available, virtually all subspace cluster approaches could be accelerated considerably. Note that this problem is essentially different from the feature selection problem [59, 86]. Established index structures [120] are designed and optimized for the complete data space where all attributes contribute to partitioning, clustering etc. For these data structures, the space of queries facilitated by the index structure must be fixed prior to the

construction of the index structure. Approaches addressing the problem of subspace similarity search *explicitly* are [99, 85]. There, the authors propose an adaptation of the VA-file [144] to the problem of subspace similarity search. The basic idea of these approaches is to split the original VA-file into d *partial* VA-files, where d is the data dimensionality, i.e. one file for each dimension is obtained, each containing the approximation of the original full-dimensional VA-file in that dimension. Based on the information of the partial VA-files, upper and lower bounds of the true distance between data objects and the query are derived. Subspace similarity queries are processed by scanning only the relevant files in the order of relevance, i.e. the files are ranked by the selectivity of the query in the corresponding dimension. As long as there are still candidates that cannot be pruned or reported using the upper and lower distance bounds, the next ranked file is read to improve the distance approximations or (if all partial VA-files have been scanned) the exact information of the candidates accessed to refine the exact distance. Another approach to the problem is proposed in [93], although only ε-similarity range queries are supported. The idea of this method is based on multiple pivot-points to derive lower and upper bounds for distances. The bounds are computed in a preprocessing step for a couple of pivot points. Essentially, this approach allows to sequentially scan the database reading only the information on lower and upper bounds and to refine the retrieved candidates in a post processing step. A bottom-up combination of one-dimensional indices and a top-down search in a full-dimensional index structure, restricted according to the query, are discussed in [16, 17].

As opposed to all these approaches, BOND [34] is essentially also a search strategy for the full-dimensional space enhancing the sequential scan. It is, however, quite naturally possible to restrict a query to a given subspace, since the basic idea of BOND is to use a column store (as it might be known from NoSQL database systems). BOND ranks the columns according to their potential impact on distances and prunes later columns if their impact becomes too small to change the query result. By the design of this method, subspace queries can be *implicitly* facilitated with the same architecture. However, BOND is motivated by the application of metrics for image retrieval and,

thus, requires certain properties of a data set which restricts the application considerably:

1. The first proposed metric is only applicable to normalized histogram data.
2. Using Euclidean distance, still the length of each vector is required for pruning columns with low impact.
3. An enhanced Euclidean distance metric provides tighter pruning bounds, but requires Zipfian distributed data (like color or gray scale histograms) and certain resolve order of the columns in the database.

In this section, the focus is set on extending BOND by loosening the restrictions of its use for data sets and by improving the pruning power. In the following, BOND and its deficiencies w.r.t. these aspects will be described (Section 10.2). Afterwards the proposed extensions will be described in Section 10.3. The extensions will be demonstrate and the improved performance will be shown in Section 10.3.3.

10.2 BOND revisited

Processing multi-step queries using a filter-refinement framework, traditional index approaches resolve the data of feature vectors row-wise (horizontally) in order to obtain their exact representation. The main advantage of BOND is that feature vectors are resolved column-wise (vertically) so that the values of a feature vector v are obtained successively. Thus, the resolved part of the feature vector is known exactly whereas the unresolved part has to be approximated. This approach is inherently different from traditional tree-indexing approaches where a feature vector is either completely approximated or completely available. In order to avoid possibly unnecessary IO-operations, traditional tree-indexing techniques aim at avoiding to resolve as many feature vectors as possible which are not part of the result set. On the contrary,

BOND starts with resolving all feature vectors column by column and tries to approximate the remaining part of the feature vector. As soon as the approximation yields a sufficiently high pruning power, false candidate feature vectors can be pruned from the candidate set, so that the remaining dimensions of these feature vectors need not be resolved. BOND supports regular k-NN queries on the full data set as well as on weighted subspaces. Nevertheless, the pruning bounds deteriorate in case of subspace queries.

The main goal of the pruning statistics used in BOND is to tighten the approximations of the yet unresolved parts of the feature vector in order to be able to prune false candidates from the candidate set as soon as possible before resolving additional columns for this vector.

In the rest of the paper, the notation of [34] is followed, where $q \in \mathbb{R}^d$ denotes a d-dimensional query vector and $v \in \mathbb{R}^d$ denotes an arbitrary d-dimensional feature vector of the database. Furthermore, any database vector v can be split into a resolved part $v^- \in \mathbb{R}^m$ and an unresolved part $v^+ \in \mathbb{R}^{d-m}$, so that $v = v^- \cup v^+$. The variable $m \in [1, d]$ denotes the amount of columns that have been resolved so far. The distance $S(q, v)$ between q and v can thus be approximated by a composition of the exact distance plus the approximation:

$$S_{approx}(q, v) = S(q^-, v^-) + S(q^+, v^+) \tag{10.1}$$

Assuming a k-nearest neighbor (k-NN) query, the resulting distance bounds are then used to refine the candidate set in a traditional way, where all candidates are pruned if their lower distance bound is greater than the kth smallest upper bound. The distance $S(q^-, v^-)$ between the known parts of q and v can be computed precisely. Concerning the unknown part (v^+), an approximation for the lower and upper distance bounds to the query vector q needs to be created. The computation of $S(q^+, v^+)$ of course depends on whether the upper or lower bound has to be computed.

The basic approach of BOND uses the application scenario of histogram data, where the length of each data vector can safely be assumed to be 1. Relaxing this condition, an extension of the basic approach assumes the unit

hypercube $[0,1]^d$ as the data space. This extension is based on the Euclidean distance and does not rely on any distribution or assumption of the data set, as it only depends on the query vector:

$$S_1(a,b) = \sum_i (a_i - b_i)^2 \tag{10.2}$$

$$S_2(a,b) \leq \sum_i \max\{a_i, 1-a_i\}^2 \tag{10.3}$$

Yet the advantage of the independence from the data distribution and resolve order is paid with the loss of pruning power, as the obtained bounds are:

$$S_{upper}(q,v) \leq S_1(q^-, v^-) + S_2(q^+, v^+) \tag{10.4}$$

$$S_{lower}(q,v) \geq S_1(q^-, v^-) + 0 \tag{10.5}$$

The weakness of these bounds is obvious, especially for the lower bound which assumes the distance 0 for the remaining, unresolved subspace and the upper bound only takes the query vector into account and does not make any assumptions on the database vector. A second extension relies on the precomputed length of each feature vector v, which is stored in the database additionally to the values of v, and a skewed Zipfian distribution of the data set. This method is used as a reference as it provided the best results in the original paper. In this case, a large number of distance computations and IO-operations can be saved compared to the sequential scan. However, the upper and lower distance bounds computed by this method quickly lose their pruning power if the data distribution changes. Also this method strictly requires a certain resolve order of the columns in the database, which is not optimal in the case of other distributions or in case of correlated dimensions. Changing the resolve order however is not an option, because this would invalidate the proof for the correctness of the pruning bounds.

10.3 BeyOND BOND

One of the main limitations of BOND is the dependence on the data distribution. The distance approximations proposed in [34] work well as long

as the data follows a skewed Zipfian distribution like in the case of color histograms and if the database columns are resolved in decreasing order of the query feature values. If either of the conditions is not fulfilled, BOND quickly degenerates, i.e. the complete data set needs to be resolved to answer the query. Thus, BeyOND extends the original idea of BOND in order to supply a query system that allows an efficient execution of k-NN queries on data sets that follow an arbitrary or unknown distribution, so that the following restrictions are removed:

1. BeyOND does not depend on the data distribution, so any distance metric can be employed that provides valid upper and lower distance approximations.

2. The values v_i of the feature vectors are no more restricted to $v_i \in [0, 1]$.

3. BeyOND does not rely on a specific resolve order of the query vector, so more sophisticated resolve techniques can be applied to further increase the pruning power.

Removing the first and third restriction also disables the possibility to use the improved distance approximations of the original work. Thus, the weak approximations shown in (10.4) and (10.5) had to be improved.

In the following, it is described how BeyOND combines the concepts of BOND and the VA-file [144] by introducing *Sub Cubes* (Section 10.3.1), supported by minimum bounding rectangles (MBRs) for certain sub cubes (Section 10.3.2), based on a BOND-style column-store architecture. This way, it is still possible to resolve the data set in a column wise manner. A restriction that remains in BeyOND, however, is the embedding into a hyper cube, so that the minimum and maximum values of each dimension need to be known.

10.3.1 Sub Cubes

The first proposed extension is to pick up the idea of the VA-file [144] by splitting the cube once in each dimension. Thus the hyper cube describing the feature space is partitioned into 2^d pairwise disjunct sub cubes. Each sub cube can be identified by the according Z-Order ID (Z-ID), which is stored as a memory-efficient bit-representation. This Z-ID is stored additionally to the values of each feature vector. The locations of the split positions in each dimension are stored in separate arrays, so that quantile splits are also supported. Assuming that the feature vectors are composed of 8 byte double values, this means that the memory consumption of a feature vector increases by a value of $\frac{1}{64}$ bytes per dimension. It would also be possible to increase the split level of the cubes even further. Nevertheless, each additional split also increases the size of the Z-IDs to $\frac{sd}{8}$ bytes, where s denotes the amount of splits. This leads to a trade-off between additional memory consumption from larger Z-IDs and tighter approximations for the upper and lower bounds of the distance computation due to smaller sub cubes. An evaluation about the impact of additional split levels is shown in the evaluation (Section 10.3.3). Given a Z-ID of a feature vector and the coordinate arrays containing the split positions, it is a computationally cheap task to recreate the coordinates of the according sub cube, so that the MBRs of potentially 2^d sub cubes need not be kept in memory but can be quickly recomputed on demand.

The sub cubes provide the advantage that the upper and lower distance approximations need not be computed w.r.t. the complete hyper cube that encloses the feature space but only between the cubes containing the query vector and the feature vectors of the database. Thereby, the following two cases need to be considered: Let Z_q and Z_v be the Z-IDs of the query vector q and a vector v of the database.

$Z_q = Z_v$ indicates that both q and v share the same sub cube, so the upper bound of the distance approximation can be lowered to the borders of this sub cube ((10.8)). The lower distance remains 0 for all unresolved dimensions.

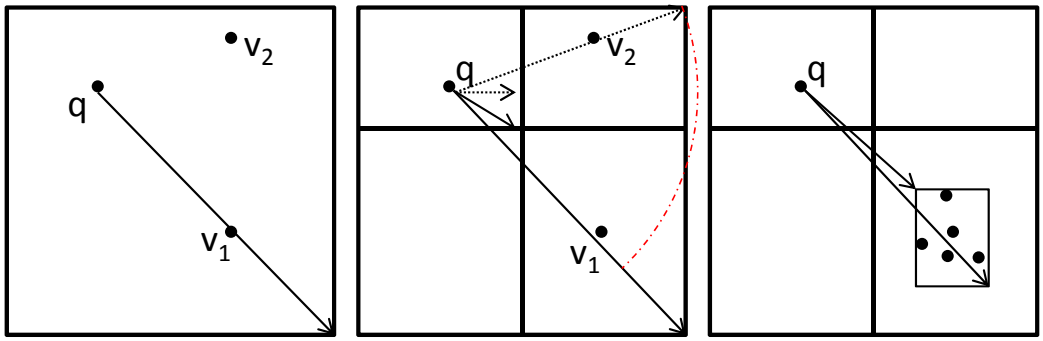


Figure 10.1: Improvement of the upper/lower distance approximation in case of no split (left), one split (center) and split with MBRs (right).

$Z_q \neq Z_v$ implies that q and v are located in different cubes, so the lower distance approximation can be raised to the minimum distance of q to the sub cube containing v ((10.9)). The upper distance approximation is again computed w.r.t. the bounds of the hyper cube containing v using (10.8). Compared to approximating the upper distance w.r.t. the complete hyper cube, this decreases the upper bound when both sub cubes share a common plane, which is the case in $d-2$ out of $d-1$ cases (cf. Figure 10.1).

$$S_2^{'}(a,b) \leq \sum \max\{|a_i - c_{b_i}^{lower}|, |a_i - c_{b_i}^{upper}|\}^2 \tag{10.6}$$

$$S_2^{''}(a,b) \geq \sum \begin{cases} 0, \text{ if } a_i \in [c_{b_i}^{lower}, c_{b_i}^{upper}] \\ \min\{|a_i - c_{b_i}^{lower}|, |a_i - c_{b_i}^{upper}|\}^2 \end{cases} \tag{10.7}$$

$$S_{upper}^{'}(q,v) \leq S_1(q^-, v^-) + S_2^{'}(q^+, v^+) \tag{10.8}$$

$$S_{lower}^{'}(q,v) \geq S_1(q^-, v^-) + S_2^{''}(q^+, v^+) \tag{10.9}$$

where $c_{b_i}^{upper}$ ($c_{b_i}^{lower}$) denotes the upper (lower) bound of the sub cube c containing the feature vector b in dimension i.

10.3.2 MBR Caching

In high-dimensional data sets that do not strongly cluster, the majority of the 2^d sub cubes is occupied by at most one feature vector. In the few cases that a sub cube is occupied by more feature vectors, it is proposed to evaluate tightening the approximation of the sub cubes. Therefore, the set of sub cubes is iterated which are occupied by more than one feature vector. Then, the MBR for the according set of feature vectors is computed and the resulting MBR is stored in a priority queue (PQ) which is sorted in descending order w.r.t. the ranking function

$$f(\text{MBR}) = \frac{V_{\text{sub cube}} \cdot \mathit{card}(\text{MBR})}{V_{\text{MBR}}} \tag{10.10}$$

where card(MBR) denotes the number of feature vectors contained in the according MBR and V denotes the volume of the sub cube or MBR.

As the resulting MBRs cannot be derived from any fixed values similar to the case of the split positions, at least 2 d-dimensional coordinates are required to define each MBR, so that each MBR requires $d \cdot 16$ bytes (again assuming 8 byte double coordinates). Even though this seems to be a quite large overhead, an MBR can be shared among all feature vectors of the respective set. Thus, the memory increase is reduced to $\frac{d \cdot 16}{\mathit{card}(\text{MBR})}$ per feature vector comprised by the MBR. As the MBR is associated with the respective Z-ID, not even an additional memory pointer is required for the feature vector.

In order to define an upper limit for this additional memory consumption, the size of the MBR queue PQ is limited to 1% of the amount of total feature vectors in the database. Combined with the ranking function (10.10), it is ensured that only a limited amount of MBRs is held in memory that contain a large amount of feature vectors on the one hand and also a significantly smaller volume compared to the surrounding sub cube on the other hand. This threshold has to be chosen as a trade-off between pruning power and additional memory consumption. Alternatively, the threshold can also be chosen in absolute values if the maximum amount of memory should be limited. In any case, the threshold should be chosen low enough so that

either all MBRs can be kept in memory or it should be ensured that only those MBRs are read from disk that approximate a fairly large amount of feature vectors, so that the time needed to load the MBRs is still smaller than resolving the respective feature vectors.

In order to use the tighter approximation provided by the MBRs, the variables $c_{b_i}^{lower}$ and $c_{b_i}^{upper}$ in (10.6) and (10.7) need to be filled with the coordinates of the MBR instead with those of the sub cube, so that this second extension integrates seamlessly into the computation of the distance approximations.

10.3.3 Experiments

Data Sets

In the experiments, the proposed techniques were evaluated on three data sets which are summarized in Table 10.1:

First, 27-dimensional color histograms extracted from the ALOI data set [49] comprising 110,250 feature vectors (ALOI) were used. This data set poses the hardest challenge as BOND is expected to perform best on this data set as the color histograms follow a Zipfian distribution.

Second, a 20-dimensional synthetic clustered data set was used, comprising 500 000 feature vectors organized in 50 clusters, each following a 20-dimensional Gaussian distribution (CLUSTERED).

Finally, a data set from the area of medical imaging was used, containing 10 715 feature vectors with 110 dimensions (PHOG). The features were provided from the work of [40] and represent gradient histograms which were extracted from medical computer tomography images. The features were already reduced in dimensionality by applying a principal component analysis and the dimensions are ordered by decreasing value of the eigenvalues.

Table 10.1: Data sets used in the evaluation.

Name	Cols.	Rows	Distribution
ALOI	27	110 250	Zipfian
CLUSTERED	20	500 000	50 clusters
PHOG	110	10 715	gradient histograms

Evaluation

In the experiments 50 k-NN ($k = 10$) queries were submitted to the database. During the processing the amount of feature vectors that were pruned after a data column was resolved and the distance approximations were recomputed was measured. The measurements shown in the Figure 10.2, 10.3 and 10.4 represent the averaged cumulative amount of feature vectors that were pruned after a column was resolved. The area under the curves can thus be regarded as the amount of data that does not need to be resolved from disk, whereas the area above the curve indicates the amount of data that needs to be taken into account for further refinement from disk and for computation of the distance approximations. Thus, better approximations of the upper and lower distance bounds yield better pruning power, so that more feature vectors can be pruned at an early stage of the computation.

In the ideal case, only a few columns have to be resolved until the k final nearest neighbors remain in the data set. Also, the final aim of the algorithm is to prune as many feature vectors as possible at a very early stage of the algorithm so that further data columns of this feature vector do not have to be resolved.

In order to measure the impact of the VA-file approach and the MBR caching, the following tests were performed: Both distance approximations of the original implementation of BOND using the improved distance approximation (BOND) and the simple approximation (BOND Euclidean) were evaluated. Then the contribution of the VA-File-approach was tested by measuring the pruning power of a 1- and a 2-level VA-file (BOND+VA+median,

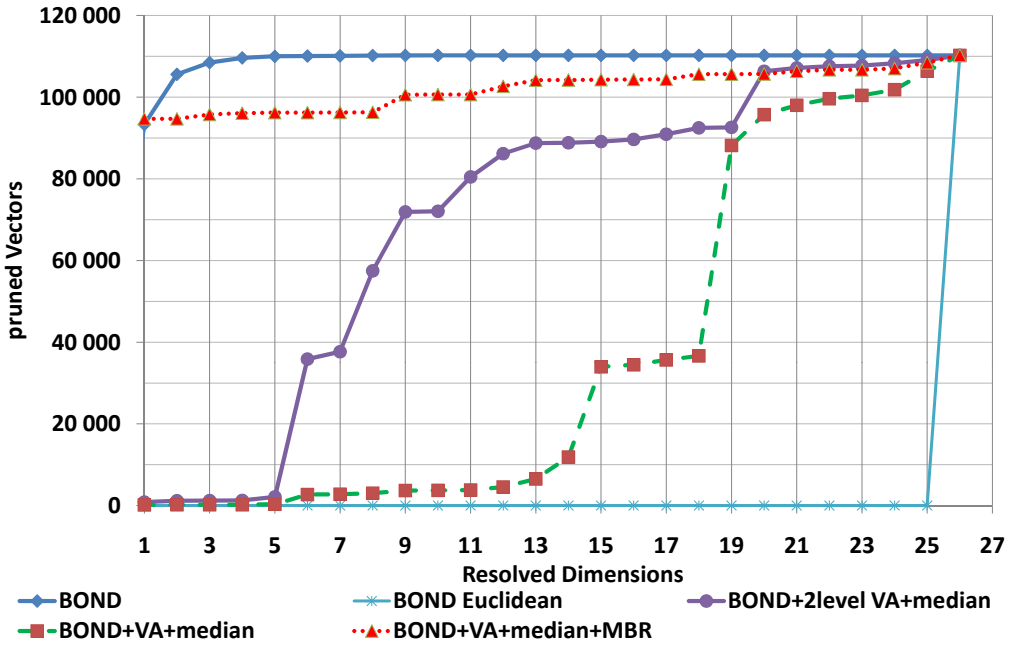


Figure 10.2: Pruning power on ALOI.

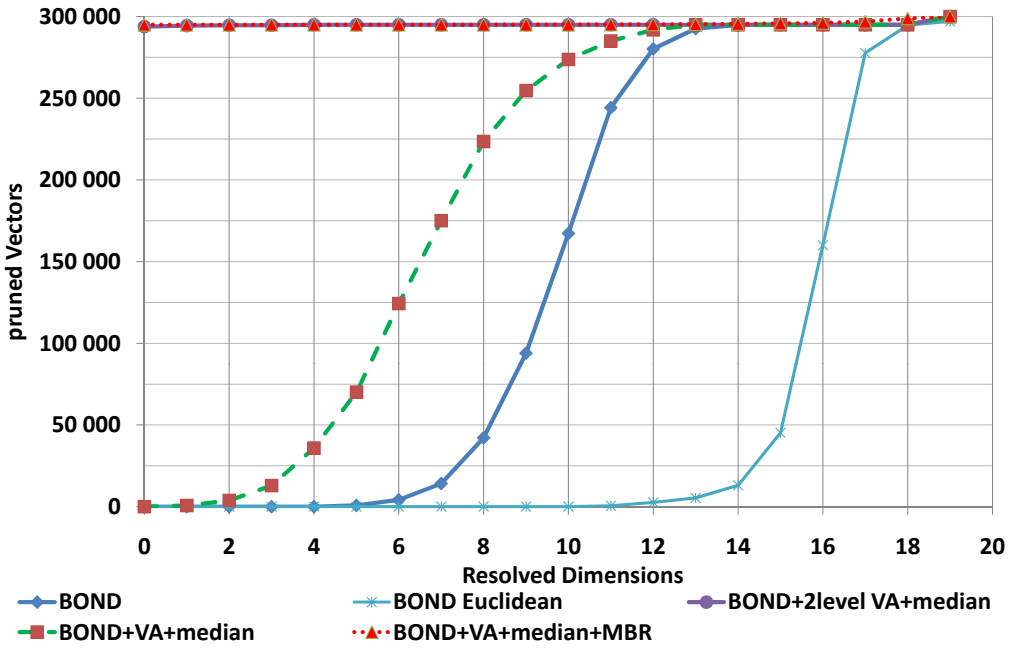


Figure 10.3: Pruning power on CLUSTERED.

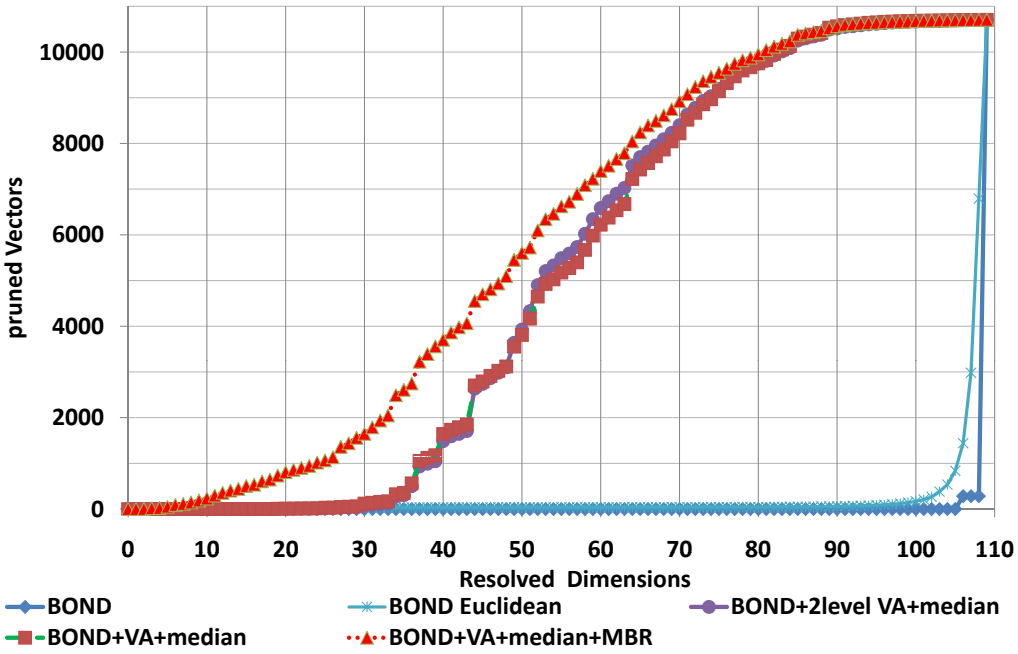


Figure 10.4: Pruning power on PHOG.

BOND+2level VA+median). Finally, the additional impact of MBR caching was tested (BOND+VA+median+MBR).

Comparing the ALOI data set with the other data sets, it can be seen that the original BOND performs as expected on histogram-like data sets. Nevertheless, BOND resolves about half of the data on the CLUSTERED data set and almost all columns on the PHOG data set. This shows the strong dependence on the data distribution, which is addressed in the approach of BeyOND.

In the first step, it is proposed to refine the simple Euclidean distance approximation by using the sub cubes that were derived from the Z-ID being saved additionally to the feature vector. While the improvement in the ALOI data set is clearly visible, the impact on the CLUSTERED and PHOG data sets is much higher (cf. Table 10.2, rows 1-3). Here, the amount of resolved dimensions is lower using the CLUSTERED and PHOG data sets compared to the ALOI data set.

Table 10.2: Pruning power of the sub cube approach. The columns show the data set, the amount of splits per dimension and the amount of resolved columns (in percent), where more than 25%, 50% and more than 90% of the candidates were pruned.

Data set	Splits	25%	50%	90%
ALOI	1	16 (59%)	19 (70%)	23 (85%)
CLUSTER	1	7 (35%)	8 (40%)	10 (50%)
PHOG	1	45 (41%)	58 (53%)	80 (73%)
ALOI	2	7 (26%)	9 (33%)	21 (75%)
CLUSTER	2	1 (5%)	1 (5%)	1 (5%)
PHOG	2	45 (41%)	55 (50%)	79 (72%)

The intuitive approach to add more splits per dimension and thus decrease the size of the sub cubes performs well with ALOI and CLUSTERED. Nevertheless, the improvement with the PHOG data set (cf. Table 10.2, rows 3-6) is negligible, while obviously the CLUSTERED data set takes most advantage from the quadratic growth of additional sub cubes ($2^d \rightarrow 4^d$), which poses a very good approximation of the clusters.

The second improvement pre-computes the MBRs in the case a sub cube contains more than a single feature vector, the MBR would be small enough and the maximum amount of MBRs is not reached yet. More sophisticated methods to determine the maximum amount of MBRs could regard the vector distribution within the cube, a minimum volume decrease, etc. In this case, 1% of the amount of feature vectors was used as a limit for the number of created MBRs for the sub cubes with the largest volume decrease. Also, each dimension was just split once. The result can be seen in Figure 10.2, 10.3 and 10.4, indicated by the dotted line, and in Table 10.3. Using the ALOI data set, the initial pruning power in dimension 1 is even comparable to the original BOND method. The CLUSTERED data set performs best as before, as 98% of the data could be pruned at once. PHOG again poses the hardest challenge. Yet, there is still an improvement compared to the basic sub cube

Table 10.3: Pruning power of Sub Cubes + MBRs. The columns show the data set, and the amount of resolved columns (in percent), where more than 25%, 50% and more than 90% of the candidates were pruned.

Data set	25% pruned	50% pruned	90% pruned
ALOI	1 (4%)	1 (4%)	10 (37%)
CLUSTER	1 (5%)	1 (5%)	1 (5%)
PHOG	37 (34%)	50 (45%)	77 (70%)

Table 10.4: Total amount of data viewed. The columns show the data set and the amount of data resolved in case of 1 and 2 splits per dimensions and the combination of 1 split and cached MBRs.

Data set	1 split	2 splits	1 split + MBR
ALOI	66.9%	38.3%	7.7%
CLUSTER	34.1%	1.6%	1.4%
PHOG	52.6%	52.3%	45.4%

approach (with 1 or 2 splits per dimension).

Table 10.4 shows the total amount of data that was resolved from the data set. It can be seen that in case of ALOI and PHOG, it is more profitable to extend the original idea of BOND with a 1-level VA-file (1 split per dimension) using the technique of MBR caching instead of simply adding more layers (2 or more splits per dimension) which generates more sub cubes. Using the CLUSTERED data set, there is almost no difference between the approaches of more splits and MBR caching. Nevertheless, the solution of a single split combined with MBRs offers more flexibility regarding the choice of MBRs and the control of additional memory consumption than simply adding more splits.

Chapter 11

Impact of Solid State Drives on Spatial Indexing

11.1 Introduction

Similarity search and spatial proximity queries are an important query type in spatial, temporal and multimedia databases. In general, the task is to find all spatially close neighbors to a query object in a database of d-dimensional feature vectors. Example applications for this type of query might be to select all sensors within a 5 mile diameter around some seismic distortion or find the top-k similar songs in an audio database. For processing this type of queries the simplest solution is to scan the complete database. However, for large databases this leads to an enormous overhead in distance computations as well as in I/O. To avoid this overhead, the database community proposed spatial index structures [47, 22] organizing the database in order to avoid comparing the query object to each feature vector in the database. The most prominent spatial index structure is the R-Tree [58] and its extension the R*-Tree [12]. Though many further extension of its principles have been proposed in the last twenty years, the R-Tree is still the most used method in the area. Additionally, it is implemented in several standard database

systems like MySQL[1], PostgreSQL via GiST[82] or Oracle[2].

Despite its wide use, the R-Tree or related data structures do not solve the problem of efficient similarity search by guaranteeing a logarithmic processing time for similarity queries. Instead, it only provides an average logarithmic search time. Factors having a negative impact on the search time include high dimensionality and an inappropriate data distribution.

Besides the dimensionality, the hardware underlying the R-Tree plays an important role when determining the performance advantage compared to the sequential scan. A major difference between both methods exists due to the predominant type of I/O-operations employed in both methods. The sequential scan basically reads the complete database, ideally employing a single seek on the disk. Thus, the transfer rate of the device is very important, whereas the seek time and latency are rather negligible. Searching on a hierarchical index like the R-Tree on the other hand is largely determined by random access I/O to single node pages. Thus, the cost of a similarity query on an R-tree is strongly dominated by the number of accessed nodes because accessing the page usually requires much more time than transferring its contents into main memory. Thus, in [23] the authors state that the selectivity of a query in a hierarchical index should be less than 5% in order to clearly outperform the sequential scan. In this case, considering only the CPU time of both methods would still favor the tree because the amount of distance computations is still several times smaller than for the scan. However, since the search is I/O bound and sequentially reading the complete dataset is faster than reading 5% of the data with random access operations, the scan still yields a performance advantage. Nevertheless, the threshold of 5% is subject to various system parameters like latency, seek time and transfer rate of the underlying storage system. Since the performance characteristics of available background storage devices have significantly changed within the last ten years, the current threshold should be considerably different as well. Additionally, unlike the simple scan an index structure usually utilizes the

[1] http://dev.mysql.com, R-Tree indexing for 2 dimensions
[2] http://www.oracle.com, Oracle Spatial: 2 to 4 dimensions

disk caches to a larger extent than the sequential scan.

As a result the used hardware and system characteristics have a tremendous effect on the performance behaviour of a spatial index structure like the R-Tree. However, when discussing the performance benefits of spatial index structures, most performance evaluations presume the operation on a conventional single user workstation system without considering using modern hardware components and a multi processor server workload.

In this chapter, the real-time performance improvement of the R*-Tree will be examined under various aspects by using new hardware, i.e. flash solid-state-drives (SSDs). As a first result of the examination, it will be shown that simulating the system workload is mandatory in order to observe the influence of different background storages due to cache utilization. It is argued that the use of SSDs for storing spatial indexes is quite a realistic architecture because the size of the available SSDs is now sufficiently large even for very large index structures. Furthermore, datasets in large spatial or multimedia search systems are have to perform more read that write queries indicating that change operations do occur considerably less often than search queries. Thus, using SSDs is feasible in spite of the limited writing capacity of flash based storage modules. Another important factor is the affordable price of flash SSDs, making the use of dedicated storage devices for indexing rather inexpensive. Finally, the most important reason for using an SSD for spatial indexing is its enormous improvement in access time compared to conventional hard drives. While the transfer rate of a modern hard disk drive (HDD) is quite comparable to an SSD, the access time of the SSD is up to two orders of magnitude faster. Thus, hierarchical index structures should significantly benefit more from this new storage device than scan-based methods. The contributions of this chapter are:

- An examination of the query-time behaviour of the R*-Tree depending on the server workload for background storages using an HDD and an SSD.
- A discussion of the implications of the SSD's characteristics on tuning

the page size of the R*-Tree.

- A comparison of the effects of increasing dimensionality to the same index run on both storage systems.

The rest of this chapter is organized as follows: In Section 11.2 related work on using SSDs in databases is being discussed. Section 11.3 dicusses the changes in the access path over the last two decades. In Section 11.4, the testing environment for the experiments is formalized with results being shown in displayed in Section 11.5. They demonstrate the effect of the hardware advances to the performance of the R*-Tree w.r.t. system workload, page size and dimensionality. the chapter concludes with a summary and ideas for future work.

11.2 Related Work

Publications on SSDs usually focus on their main weakness: a limited number of write operations which are relatively slow compared to reads. Lee *et al* [90] compared the transaction performance of standard SQL I/O operations on HDDs and SSDs and have found SSDs to be faster. However, the runtime advantage of SSDs decreases with an increasing number of users due to imperfect handling of write operations. As write operations are crucial for the creation and updates of index structures, there has already been research on how to design write structures for indexing on SSDs. In [146, 147] Wu *et al* proposed a method to speed up the construction and maintenance of B-Trees or R-Trees directly by using the flash translation layer (FTL). In [92] Li *et al* introduced the FD-Tree, a B^+-Tree derivate of three index layers specialized to the use in flash discs.

In addition, various types of queries on SSDs have been analyzed. For join operations, the writing problem prohibits an effect of the full advantage of fast reads. Thus, [125, 134, 37] have developed methods for fast join processing on flash devices.

Write-independent queries profit stronger from exchanging HDDs by SSDs. In [51] Goetz Graefe tested query runtimes of the B-Tree on SSDs as opposed to HDDs and concluded that SSDs are not only faster but they also induce a lower optimum page size. In [100] Nath and Kansal use a cost model for the automatic adaption of flash-specialized B^+-Tree types and their parametrization (e.g. page sizes) to the varying access costs of different flash devices. It depends on the tree's height, the read and write access cost of the disk and the node's *utility* [57], the logarithm of records within a node.

11.3 Changes in the access path

In this section, the advances in computer architecture will be briefly reviewed that have the strongest influence on the performance of spatial index structures. For most types of similarity queries (apart from special applications like similarity joins), it still holds that the performance bottleneck derives from the I/O-operations. Thus, the focus will be on changes w.r.t. the access path of the indexed data.

11.3.1 Caching

Commonly used cost models for estimating the I/O-costs of index structures usually only regard caching strategies implemented directly into the proposed method (like LRU-page buffers). But besides these explicit caching strategies, all methods are implicitly using caching strategies provided by the underlying operating system (OS) (unless explicitly coded differently) in order to avoid time consuming I/O operations.

A typical disc read request proceeds according to the following pattern: The process requests a certain part of a file to be read from disc. This request is directed to the file system driver which first checks the cache manager and virtual memory for the page. If the page is found (cache hit), the data is returned immediately from the cache without needing to start a time

consuming I/O operation. On the other hand, if the data could not be found in the cache (cache miss), the disc driver requests the data to be read directly from the device and thus from the next cache layer, the disc cache which is implemented directly on the according device having a size of 8 – 128 MB (depending on type and manufacturer). If this request also results in a cache miss, the data must finally be read from disc, causing the expected I/O cost of seek- and transfer time of the requested blocks of data.

Another issue is that OS and disc cache managers analyze file access patterns to a certain amount and start reading ahead data into the caches in order to improve access speed for future reads.

11.3.2 New Storage Media

Thus, the most important part of the access path remains the used storage medium of the data to be indexed. With the rise of SSDs as a new possibility to store and access data some years ago, it was the aim to examine the performance characteristics of common HDDs and SSDs in the context of spatial index structures. Traditionally, three parameters are of crucial importance in this scenario:

- Seek Time: The time to find the requested blocks on the medium.
- Latency: The time until the storage medium can access the requested blocks.
- Transfer Rate: The time to transfer the requested blocks to the processor.

Hard Disk Drives

Commonly used hard Disk Drives (HDDs) store data on rapidly rotating platters with magnetic surfaces. The data is accessed by positioning the head of the right platter and then transfering the data. The seek time is the time

required to position the head on the target platter and the correct track, whereas latency is the time passing until the platter is rotated to the position of the sector containing the requested data block. Just like the transfer rate, the average latency is therefore dependent on the rotational speed of the drive. In the last two decades, especially the transfer rate of HDDs increased ($\sim$ factor 40), whereas the seek time and latency only improved little ($\sim$ factor 3).

Solid State Disks

In contrast to HDDs, flash Solid State Disks (SSDs) do not have any mechanically moving parts but use NAND flash memory chips to store the data. Each flash chip is divided into several flash blocks consisting of several flash pages. Operations on the drive are performed on page level. Due to these characteristics the only latency for a read operation derives from the mapping between logical block addresses and flash pages. This results in access times which are typically two orders of magnitude faster than the ones from HDDs. However, the structural characteristics only remove the seek process and boost the latency time of a disk, thus flash SSDs are not (yet) able to outperform HDDs w.r.t. the transfer rate until now.

Theoretical Implications

When accessing a large amount of (unfragmented) data sequentially, the main parameter of interest is the transfer rate, since seek time and latency only occur once. Thus this parameter has the highest influence on scan-based methods like the sequential scan or the VA-File [144]. On the other hand a hierarchical index structure has to access the data independently and in a random access manner. In this case, the seek time and the latency are more important for the performance of the index structure. This suggests that hierarchical spatial indexes benefit by far more from the use of a flash SSD than the sequential scan does. To confirm this assumption, in a first step the

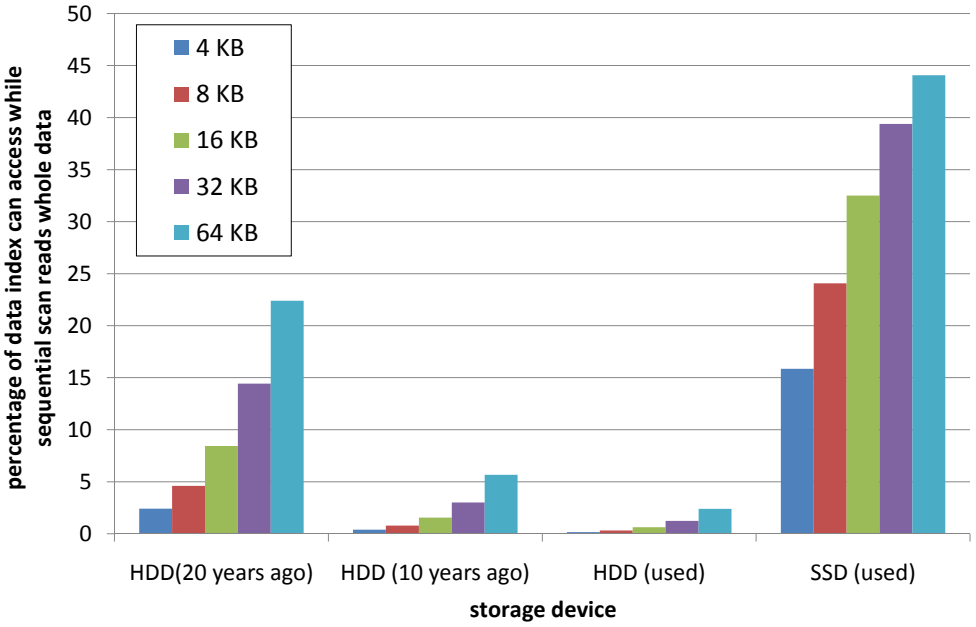


Figure 11.1: Fraction of data a spatial index structure can read before the sequential scan becomes faster

performance of the sequential scan and spatial index structures are examined from a theoretical point of view. Therefore the cost model from [23] is used to calculate the percentage of data which can be accessed by an index structure (via random access) while the sequential scan reads the whole dataset. Like the authors of [23] state, a storage utilization of the index of 50 % is assumed. Besides the two storage devices used in the experimental section (summarized in Table 11.1), two HDDs were included into the calculation. One as it was used 20 years ago (latency + seek time = 20 ms; transfer rate = 5 MB/s) and one as it was used 10 years ago (latency + seek time = 15.3 ms; transfer rate = 32 MB/s). Figure 11.1 illustrates the fraction of the data a spatial index can maximally access before the sequential scan becomes faster. For each storage device, several page sizes are compared as this has an impact on the break-even point. Also note that the optimal page size for an index is mainly dependent on the characteristics of the indexed dataset. The figure shows that the advances in the HDD technology of the last decades makes it harder for an index structure to perform faster than the sequential scan. This

fraction is nowadays far below the commonly assumed 5 % rate (cf. [23]).

With the new technology of SSDs this trend is reversed. Following the above considerations, a spatial index theoretically still performs better on an SSD than the scan even if the accessed amount of data is far above 20% for a common page size ($\geq$ 8 KB). Thus, on SSDs a spatial index should outperform the sequential scan even if a large amount of the data has to be visited as it is the case for data which is hard to index (e.g. high dimensional). However, the real-time performance of a spatial query usually depends on several other system characteristics like the cache utilization. Therefore, the performance advantage of an SSD is empirically examined compared to a modern HDD in the next section.

11.4 Evaluation

11.4.1 Datasets

For the majority of the experiments, a random test database was created. Since tree-based spatial index structures are most challenged by poorly clustering datasets, uniformly distributed datasets were chosen. Clustered datasets were not simulated in order to avoid overfitting of the distributions to the used index. All experiments involve 1 million 10-dimensional feature vectors unless explicitly stated otherwise, i.e. $N = 10^6$, $d = 10$.

11.4.2 Hardware

To support the assumptions, several settings were tested on two storage devices, one of each class. The HDD was a *Western Digital Caviar Blue (WD2500AAJS)* SATA Drive with 8 MB cache, 250 GB memory and 7200 rpm. The SSD was a *Corsair P128 (CMFSSD-128GBG2D)* SATA II drive with 128 MB cache and 128 GB memory. For further specifications see Table 11.1. All experiments were run on a machine with two Intel Xeon 5160 3.00 GHz

Table 11.1: Performance characteristics of used devices

	HDD	SSD
Avg. seek time	8.9 ms	none
Avg. latency	4.2 ms	0.09 ms
Transfer rate	93.5 MB/s	94.7 MB/s

Dual-Core processors and 4 GB of main memory.

11.4.3 Software

The test data was inserted and stored in a persistent R*-Tree of the ELKI framework [2]. Each accessed node in the tree results in one access to the underlying storage system. Correspondingly, the experiments showing the results of the sequential scan were programmed to access the storage device only once. An important aspect of the following results is that the implemented search system is running on top of an operating system allowing concurrent processes. In this work, *openSUSE 10.3 (X86-64)* was used for all following tests. Additionally, most experiments were also evaluated on Windows XP to test whether a different operating system would cause significantly different results. However the results between both operating systems were quite comparable and thus, the results presented in this section were all measured on the LINUX system. As mentioned in the previous section there are multiple caching systems for background storages. Thus the experiments include the caching mechanism provided by the underlying hardware and operating system. Even though using a disk cache is a realistic assumption, having an otherwise idle system is definately not. Thinking of an index structure as a component of a database server, it is a very unrealistic assumption that the only process currently running is the search preocess itself. An experiment with just a single active process leads to an unrealistically large amount of available main memory which the operating system will transparently use for caching parts of the index or the dataset. Furthermore, if the system is only occupied with the test program, it is also quite likely that

its caches will be exclusively used for parts of the index. However, assuming the search process as part of a larger server system, the system will share the resources with other processes as well. Considering the sequential scan, it is possible to scan a large data file in a consecutive way if there are no other concurrent processes accessing the disk as well. However, on a real database server, it is rather unlikely that there is no other process or thread requesting to access the disk as well. Thus, in order to provide fair answering times, most systems will interrupt large scans causing multiple disk accesses for the sequential scan as well. Thus, cache utilization, available main memory and concurrent reads will be limited by the system load caused by other database server functions and other processes. A further aspect limiting the resources for a single query is the concurrent processing of several user requests.

To conclude, in order to make sure that the tests are performed under more realistic conditions, it is required to limit the available resources for answering a query and to simulate a server workload consisting of multiple concurrent processes. To achieve this result, main memory was allocated and locked to make sure that the test system only had access to 1 GB of main memory. Furthermore, to simulate concurrent queries being answered at the same time, the test program was multi threaded for answering multiple queries at the same time. The effect of the number of parallel queries will be discussed in the following section. The query performance was measured based on the average answering times of k-nearest neighbor queries with the number of retrieved neighbors k set to 10.

11.5 Experimental Results

In this section, the results of the experiments are presented in which similarity queries based on a flash SSD and HDD background storage were executed. The first test starts with measuring the utilization of the background storage when scaling the system load. Afterwards, the impact of the changed performance characteristics on tuning the page size of the R*-Tree is measured. In a

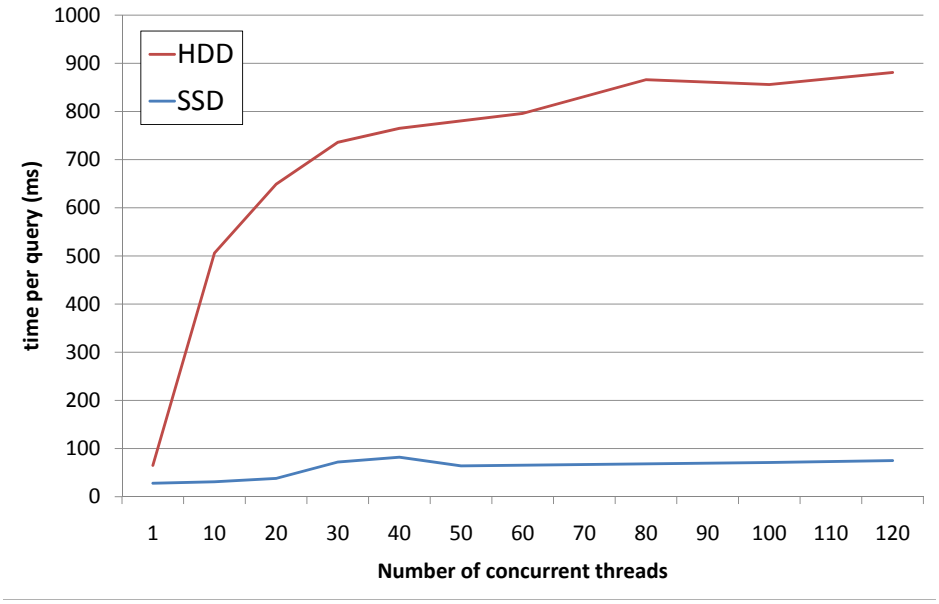


Figure 11.2: Query time for an increasing number of parallel queries and databases ($k = 10$, $d = 10$, $N = 10^6$, uniformly distributed data)

final set of experiments, the influence of a changed size of k and different dimensionalities of the dataset will be evaluated.

11.5.1 System Load and Storage Device Utilization

As mentioned in the previous section, it is expected that the workload of the database server has an important influence on the effect of the background storage and thus the query times. If the system load is rather low, the database server is expected to spend more resources like caches and main memory to answering the query. Thus, the performance of the background storage should have a smaller impact. To simulate different levels of work load, 5 000 queries were performed to an R*-Tree comparing both devices while changing the number of concurrent threads processing the queries. Each R*-Tree stores one million 10-dimensional feature vectors. It is assumed that each thread has its own instance of the R*-Tree, which simulates queries on different index structures. The results of this experiment can be seen in

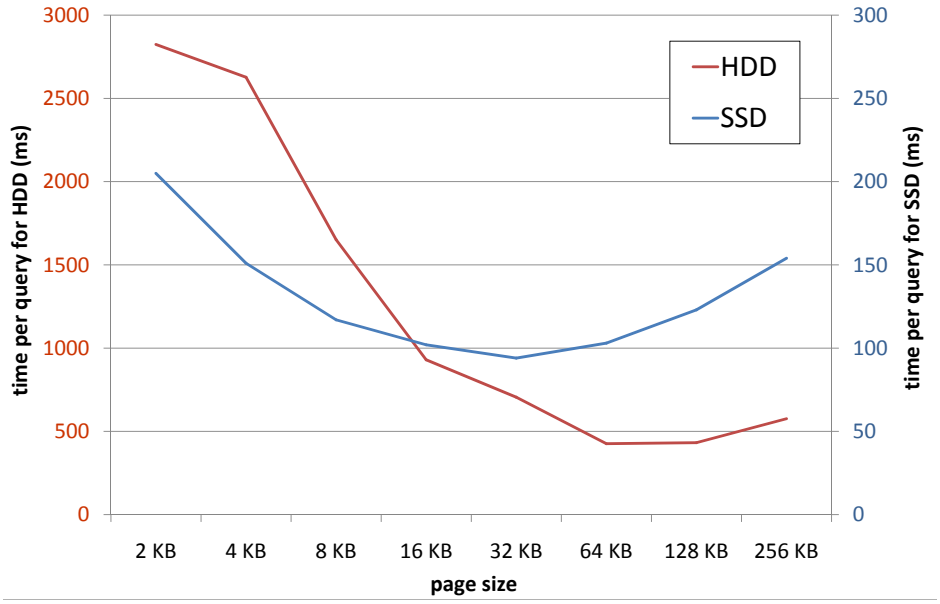


Figure 11.3: Query time for different page sizes on HDD and SSD ($k = 10$, $d = 10$, $N = 10^6$, uniformly distributed)

Figure 11.2.

While the performance advantage of the SSD-based tree is rather small for a single query thread, the gap between both devices rapidly increases with the number of simultaneous threads. The rather small difference for a limited number of threads can be explained due to the good cache utilization for a small level of concurrency. However, with an increasing number of concurrent threads, it becomes more and more unlikely that a cache hit for the data in of a dedicated thread occurs so that the amount of cache misses for a dedicated thread will increase. Thus, the effect of the cache is strongly decreasing and for a number of 100 threads the impact of the storage device can clearly be observed. For more than 100 concurrent threads, the average answering time of a query is about one order of magnitude faster on the SSD than on the HDD. For the tests w.r.t. data dimensionality and the number of retrieved nearest neighbors k 100 concurrent thread were used in the following.

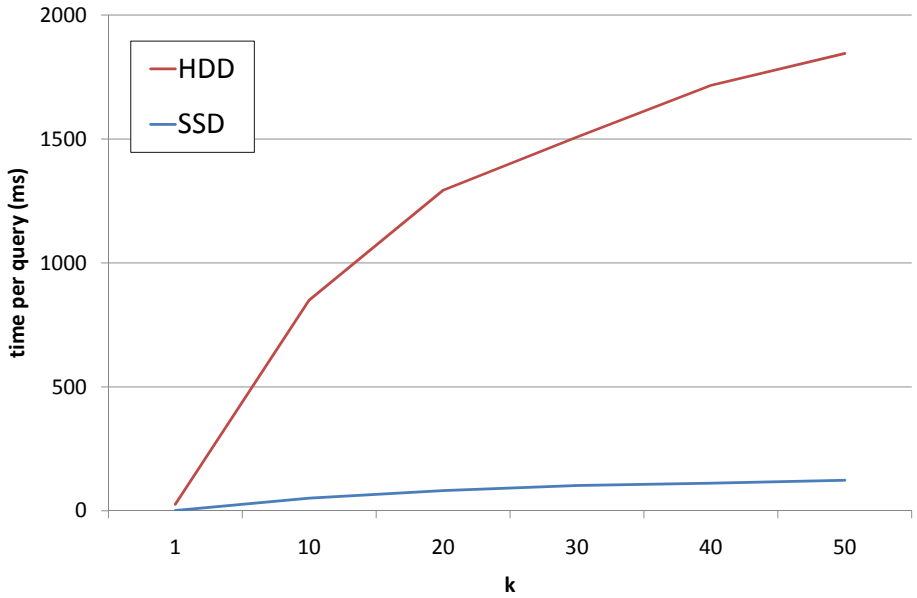


Figure 11.4: Query time on for increasing k on HDD and SSD ($d = 10$, $N = 10^6$, uniformly distributed)

11.5.2 Impact of the Page Size

In this section, the impact of the page size used for the background storage of the R*-Tree indices will be evaluated. Selecting a suitable page size can have a large impact on the query performance of a spatial index structure. The impact of the page size can be explained as follows. A large page size reduces the overhead of accessing a page on the background storage compared to the transfer time of the page content. In the extreme case the page size is large enough to keep the complete dataset and thus, the R*-Tree degenerates into the sequential scan on the root page. From a CPU-time point of view, small pages are usually more beneficial because their spatial approximations usually have a smaller spatial extension. Therefore, it is less likely that they intersect with the query region. In combination with the smaller amount of stored data objects this leads to a decreased number of distance computations.

In the following, the optimal page size for the R*-tree based on the flash SSD compared to the HDD should be evaluated. Therefore R*-trees for page

sizes varying from 2 kB to 256 kB are generated first. Each tree contains the same dataset of one million 10-dimensional, uniformly distributed feature vectors. To test the query performance, 480 10-nearest-neighbor queries were simultaneously performed by 30 concurrent threads for both storage devices. The measured average processing time per query can be seen in Figure 11.3.

On the HDD the results indicate that the pages should be chosen considerably larger than the 4 kB disk pages of the underlying file system. The best results were achieved with a page size of 64 kB closely followed by 128 kB pages. Due to the significantly shorter access times of the flash SSD, its optimal page size is expected to be smaller than for the hard drive. As expected, the best results where measured when using a page size of 32 kB on the SSD. Due to the comparatively large dimension of the test data, this amounts to a considerably smaller discrepancy between HDD and SSD than an earlier study on B-Trees (256 vs. 2 kB for only 1 dimension) [51]. However, in general the performance of the SSD indicates no strong decrease in performance for 8 to 128 kB.

This is a remarkable observation as it shows a clear difference between the SSD and a classical HDD as the performance of an R*-Tree stored on a HDD depends much more on a suitable page size compared to the situation observed on the SSDs. For example, the query time of ∼2 700 ms on the hard drive based on a system with a page size of 4 kB is more than five times higher compared to the index stored with a page size of 64 kB (∼500 ms). In contrast, the runtimes of the observed optimal page size of 32 kB and the worst page size of 2 kB on the SSD only differ by the factor two. Due to the fast access times of the SSD, its performance is more independent from the choice of the proper page size.

11.5.3 Query Size

In the next experiment, the impact of different query sizes on the runtime of the spatial index was tested on the two different devices. Therefore the runtime for different parameters of k was measured. Figure 11.4 visualizes the

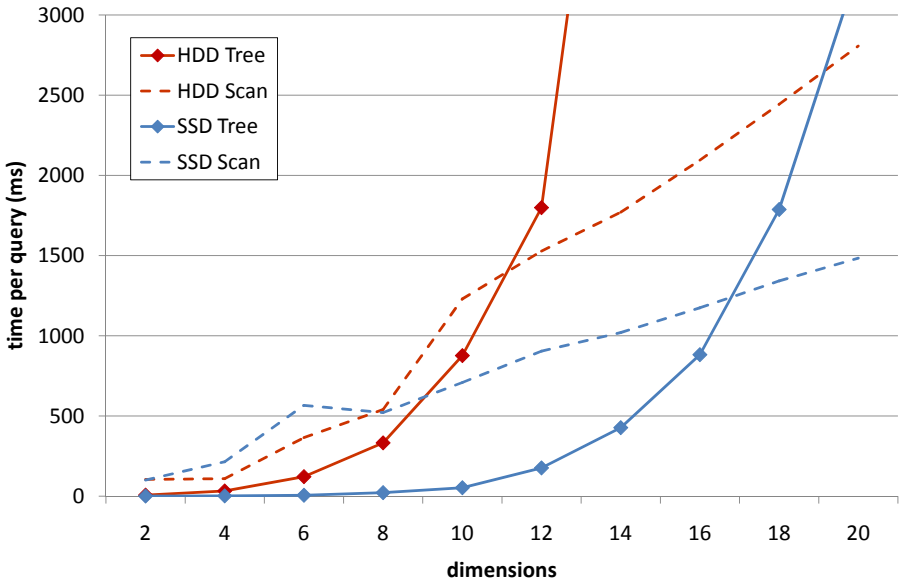


Figure 11.5: Performance of R*-Tree and sequential scan with increasing dimensionality on HDD and SSD ($k = 10$, $N = 10^6$, uniformly distributed)

results of the experiment. In this case it can be observed the total difference in runtime increases for a higher value of k. Interestingly, the query time of the index on the SSD is always an order of magnitude faster than the one on the HDD.

11.5.4 Dimensionality

One of the most interesting questions for spatial index structures is the dimensionality of the data. As the key idea of an index structure is to improve access speed, it is crucial to know, at which dimensionality of the data a sequential scan over all the data should be favored over the index. Generally, with increasing dimensionality of a dataset, the overlap between pages of an R*-Tree increases more and more which leads to a higher rate of pages which have to be visited at query time. This directly leads to a decreased selectivity and this again to an increase in query time.

Section 11.3.2 already gave an indication, that R*-Trees can still perform

better than the sequential scan on SSDs even for a large ratio of read pages. To test this hypothesis test were performed on uniformly distributed data sets with increasing dimensionality. For each test the query time was measured for both the sequential scan and for the R*-Tree on HDD and SSD. The test results can be found in Figure 11.5. As expected, the query time using a sequential scan increases roughly linearly (due to increasing data volume and increased amount of distance computations) in both cases, with the SSD performing faster than the HDD. Since the transfer rate of both media is comparable, this effect is probably caused by interruptions of the sequential scan, resulting in new seeks. Due to the almost non existent seek times in case of the SSD, such interrupts hardly impact the scan on SSD whereas each additional seek causes remarkable impact in case of the HDD.

Comparing the performance of the R*-Tree on both devices shows that the index is around one order of magnitude faster in case of the SSD than on the HDD, regardless of the dimensionality. This is caused by the lower seek and latency times of SSDs which play a central role in the performance of spatial index structures.

The observations of this chapter conclude with a comparison of the query times of the R*-Tree and the sequential scan for each device. On the HDD, the scan outperforms the index if the data consists of more than 11 dimensions. In case of the SSD, the break-even point can be observed when the dimensionality of the data grows larger than 17. This result confirms the theoretical assumptions from the previous sections and shows that spatial index structures can greatly benefit from the use of SSDs.

Part V

Conclusions

Chapter 12

Summary

Knowledge discovery and data mining in large medical data bases are challenging tasks. Supporting clinicians by evaluating and providing tools for computer aided diagnosis, algorithms for medical image computing and methods for data analysis is more than just the automation of daily routines. Research and development in this area should also be regarded as a way to extract and recover knowledge that might otherwise be undiscoverable due to the huge amount and the fast growth of the data. The topics of this thesis focus on data mining and indexing aspects in medical imaging and medical sensor data. The following part of this chapter concludes the thesis and outlines the main contributions of the presented methods.

12.1 Preliminaries

In Part I the three major parts of the thesis are briefly introduced. First, the need for techniques in the field of medical imaging is motivated. Afterwards, an example of long term monitoring patients is outlined to motivate the need of data mining techniques for data which is obtained for activity recognition and classification in long term patient monitoring. As a lot of techniques organize data in feature vectors with high or very high dimensions, the need

for indexing is motivated in this part as well.

12.2 Medical Imaging (Part II)

Part II focuses on the domain of medical imaging and more precisely on Computer Tomography data.

Section 3 introduces the history and the different imaging modalities in the area of medical imaging in more depth and afterwards presents two novel methods for the localization of a single CT slice along the z-axis of the body.

In Section 4 the topic of slice localization is motivated by the need for retrieval methods which are independent of the height of a patient and also based on image similarity only as it cannot be assumed that there is any reliable annotation about the body position in the meta data of an image. The proposed methods are evaluated on a very large set of CT volume scans obtained from clinical daily use. In order to achieve detection rates that are within the constraints stated by an expert medical user, feature vectors first need to be extracted from the CT slices. The vectors are then used as query objects to perform similarity search queries in the data base. Afterwards, the impact of using the information obtained from multiple adjacent slices is evaluated. As a reference, the method is compared to a state of the art method that is attempting to accomplish the same issue by using large volume sets. As a result it can be observed that the proposed method is more stable and more generic than the reference as it does not depend on any landmarks that have to be detected in advance. Furthermore it only requires a single slice for the same task by achieving the same or a better results.

Another topic introduced in Section 5 of the thesis is the localization of the vertebrae on single CT slices. Research on that topic is motivated by the observation that there are a lot of methods operating on the spine and the vertebral bodies. However these algorithm usually have to be initialized with the information about the location of the spine. By providing this new method, there is now the possibility to combine this method with the

algorithms mentioned before in order to determine the location of the spine automatically. This information can for example be used to initialize the afore mentioned algorithms without human intervention.

12.3 Medical Sensor Data (Part III)

Part III of the thesis focuses on the analysis of medical sensor data and a framework for faster prototype development.

Section 7 deals with the classification of physical activity by analyzing samples, recorded by a small 3d accelerometer. Even though the field of activity classification is not completely new, the special setting of this use case (comparatively low recording rate and the special position of the sensor mounted at the ankle) poses a challenging task. After evaluating related work, a new method is proposed that combines known features with newly developed features in order to build a novel feature description. The related approach and the new approach are evaluated on a data set obtained from volunteers who recorded more than 10 h training data in their spare time.

Section 8 describes a software which is based on a common platform that allows the integration of common data mining techniques. This work is motivated by the original aim of the use case in this chapter which is not restricted to the classification of physical activity but also to deliver this information to the end users. However, the user group of end users had to be divided strictly in two separate classes: One group representing attending physicians which require the possibility to obtain a quick overview of the data as well as the possibility to have a very detailed view to the data. The other target group covers the patients wearing the sensors. For this user group, it is desired to have a very simple interface which allows the patient a quick information if he/she has already fulfilled a certain amount of physical activity. The advantages in this approach are that some major data mining frameworks can be integrated and that the view which is presented to an end user can be replaced very easily. In the current state, the software provides

a view for scientists so that algorithm chains can be build visually but also a prototype view for attending doctors who want to classify data obtained from the accelerometers.

Concluding it can be said that the open *Knowing* framework allows faster integration of data mining techniques into the development process so that information and data can be managed more effectively. The integration of *Knowing* in the application of medical monitoring is demonstrated by the *MedMon* application which outlines the bridge between data mining and development. Currently, more well known data mining techniques are added and the UI handling of the framework is continuously improved for a more convenient user experience.

12.4 Indexing (Part IV)

In Part IV, a technique is proposed to enhance nearest neighbor search and indexing in (very) high dimensional data sets.

Section 9 introduces the topic by building the bridge between medical imaging and sensor data to the indexing topic. The section also explains the drawbacks of the BOND method and why it was regarded to be worth to improve the method.

In Section 10 BeyOND is explained in depth as a technique to perform exact nearest neighbor queries in very high dimensional spaces comprising far more than 10 dimensions. This approach differs from hashing methods in so far that BeyOND provides the ability to find the guaranteed k nearest neighbors instead of approximate results only. The method itself is a generalization of BOND which was proposed in 2002 in the work of [34] and combines this work with ideas of the VA-File which was published in [144].

Section 11 addresses the impact of modern hardware on established indexing techniques. The growing popularity of Solid State Disks (SSDs) in the past years also affected the research in the data base community as access

paradigms that have been driving the indexing methods for decades began to change. In the mean time, it is common to apply SSDs together with conventional HDDs to improve data base performance if the performance of classical index structures is a bottleneck in an application. In this work, the performance impact of modern hardware to classical spatial indexing is evaluated. The key contributions of this work shows that spatial indices can now be created on data spaces with a dimensionality that is almost 60 % higher than in the case of classical HDDs. A possibly even larger contribution is the performance evaluation of index structures in cases of heavy or unexpected load situations. Such situations can be observed on large web data base hosting clusters for several reasons. For example, global and seasonal events like Christmas cause an increase of server load used at shopping websites across all over the world at the same time. Local events like very heavy load on single web sites can be caused by criminal intent like in the case of distributed denial of service (DDoS) attacks or due to a sudden and unexpected increase of traffic to the web page. In such cases SSDs are not the solution to the problem, but due to behavior in case of massively increasing parallel requests, they allow a graceful degrade and thus increase the reaction time until a site turns completely unavailable. In case of HDDs, there is no graceful degrade so that the time to react is very short and the point where the site is unavailable is reached rather suddenly as soon as no more system memory is available.

Chapter 13

Future Directions

After having discussed the contributions of the work so far, this section addresses the ideas for further research in the field of medical imaging and sensor data as well as in the domain of indexing high dimensional data.

13.1 Medical Imaging

Even though the work concerning the slice detection in Section 4 has already reached a very high level of accuracy, there are still some topics for future research that have not yet been addressed:

- Currently the 2-stage $k$nn search which is applied for the prediction step involves the complete data base. In order to reduce the search space, it should be evaluated if some clustering techniques could be applied on the data base. The clustering could be used to identify almost identical feature vectors which could be replaced with a single representative. At the current stage, the search does not take a long time, however as the $k$nn search involves the complete data base, the search time scales linearly with the size of the data base.
- With the slice detection showing such convincing results on Computer

Tomography data, there has been a considerable amount of requests asking for an extension of the method to data obtained from MRI scanners. Due to the lack of access to such data it has not yet been possible to evaluate or adapt the technique to such data. However, discussions with experts in this area allow to make an educated guess that the method can be applied to MR data as well.

- The extension of the technique to subset queries is yet far from fully exploited. Combining different adjacent feature vectors can be done in many different ways. For example, the query feature vectors could first be analyzed if they contain some outliers which could be excluded from the query.

Also the topic of the detection of the vertebrae still has some potential for future research:

- An improved selection of the candidate window poses a major and possibly simple way to improve the result even further. This could for example be done by machine learning and pattern recognition algorithms that create an improved ranking for the candidate boxes.

- The shrinking process currently relies on a set of fixed parameters. Even though it currently provides a good baseline, it should be evaluated if a more sophisticated approach could yield better or more stable results.

- After the detection of a vertebra, the next steps could include the classification of the vertebra. First tests have already shown a classification rate of about 80 % with an accuracy of ± 1 vertebra. Combining the vertebra detection with slice localization could improve the accuracy dramatically.

- It should also be evaluated whether algorithms from the field of face detection could also be used to detect the very specific shape of the vertebra on an image.

13.2 Medical Sensor Data

Even though the new way to classify physical activity shows already very convincing results, there are still several topics for future research:

- There is of course the aim to increase the data set in terms of more activities, more test users and more observations per user.

- The current method provides assistance for analyzing and classifying long term activity logs. Yet, it would be a great contribution if some measurement unit would be integrated to quantify the intensity of an activity. By combining the quantity with the intensities of activities, a daily pensum could be defined and controlled much more precisely as it is possible at the current state.

- New units of measurement could also be included in the analysis which could be obtained either directly by the sensor (skin temperature, pulse, etc.) or from external resources like from a mobile phone. For example it would be possible to combine GPS information obtained by a mobile phone in order to estimate means of transportation between different activities. This could for example be used in a recommendation system to improve physical activities in daily routines.

Concerning the *Knowing* and *MedMon* frameworks there is of course still some work to do:

- Future plans cover the implementation of plug-ins to integrate more data mining frameworks and an improvement of the user interface.

- Another topic that could be addressed in the near future is the extension of the computation model to a cloud-computing schema. In cases of slow client computers or very CPU intensive tasks, this could provide a viable alternative to changing the underlying hardware.

- The *MedMon* application which is based on the framework will soon be adapted to requirements posed by attending doctors. Also the improvement of usability on all interfaces is subject to further development.

13.3 Indexing

Same as the topics above, BeyOND still poses several possibilities for enhancement as there are still some restrictions left concerning the data space that should be removed or at least loosened.

- Currently, the resolve order of the columns is independent of the query vector. The question, whether there is a possibility to improve the pruning power by applying a resolve order that also depends on the query feature vector and not (only) on the data set is still unanswered and poses a challenging task for future research.
- Current trends in research dealing with high dimensional data tend to focus on hashing which usually cannot guarantee to find the exact nearest neighbors but only approximate nearest neighbors. However it might be possible to combine the ideas of BeyOND with hashing techniques to improve the approximations used for pruning.
- The impact of the storage layer in the case of BeyOND is also an issue that should be addressed in the future. As data in such an index structure is stored no more row wise but column wise per dimension, a single block on the disc shares information of much more feature vectors. Assuming a 100 dimensional feature vector with double values, a 4 kB block in a row wise storage can store 5 feature vectors ($5 \cdot 100 \cdot 8$ Bytes). In a column store the same 4 kB block shares information of up to 512 vectors (4 kB/8 B). This means that if such feature vector should be resolved completely, a row wise storage accesses a single block, while the column wise storage has to read a single block per dimension.

Image Licenses

Figure 4.1: The body planes schema in Figure 4.1 (p. 24) originates from http://en.wikipedia.org/wiki/File:Human_anatomy_planes.svg, by wikimedia.org user "YassineMrabet". The figure is licenced under the Creative Commons Attribution-Share Alike 3.0 Unported Licence: http://creativecommons.org/licenses/by-sa/3.0/

Figure 4.2: The human model in Figure 4.2 (p. 26) originates from http://commons.wikimedia.org/wiki/File:Skeleton_whole_body_ant_lat_views.svg, Patrick J. Lynch, medical illustrator; C. Carl Jaffe, MD, cardiologist. The figure is licensed under the Creative Commons Attribution 2.5 License 2006: http://creativecommons.org/licenses/by/2.5/

List of Figures

3.1 Röntgenogram 15

4.1 Body planes schema 24
4.2 Slice Localization Schema 26
4.3 Pyramid Kernels and ROI impact 29
4.4 Feature extraction process 39
4.5 Annotation tool 47
4.6 Mapping land marks to height values 48
4.7 PHoG vs. Haralick features 49
4.8 Combined representations 50
4.9 Impact of k and PCA 51
4.10 Boxplot comparing MR-Descriptor and radial descriptor . . . 56
4.11 CDF comparing the two approaches with Feulner *et al* 57
4.12 Scatterplots 59
4.13 Mean error and std. dev according to the two methods 60
4.14 Impact of tilt angle to localization 62
4.15 Mean error and standard deviation in the 3D case 64

5.1 CT slice after pre processing 70
5.2 Bone Density Map and candidate selection 72
5.3 Selected candidate regions 73
5.4 Imapct of refinement. 75

5.5 Illustration of the weighting function w 81
5.6 Search mask ρ_{sm} 82
5.7 Parameters used in the refinement process 84
5.8 Box enlargement 88
5.9 Performance of different feature representations 89
5.10 Impact of number of candidate boxes 92
5.11 Impact of region size from which features are extracted. 93
5.12 Impact of limiting the area being analyzed for bone pixels. . . 95
5.13 Comparing VD and EVD using different overlap metrics . . . 99
5.14 Impact caused by refining and the tighter overlap measure . . 100
5.15 Comparison of feature types 102
5.16 Stability of η 103
5.17 Stability of λ 104

8.1 *MedMon* GUI 137
8.2 The Sensor Import Wizard of *MedMon*. 138

10.1 Improvement of the upper/lower distance approximation . . . 153
10.2 Pruning power on ALOI. 157
10.3 Pruning power on CLUSTERED. 157
10.4 Pruning power on PHOG. 158

11.1 Index vs. sequential scan 168
11.2 Scaling with parallel accesses 172
11.3 Impact of page size. 173
11.4 Impact of k 174
11.5 Impact of dimensionality 176

List of Tables

4.1 Workflow comparison 22
4.2 Parameter setting for both descriptors. 40
4.3 Error measures for the tested feature combinations 53
4.4 Error values of the MR-Descriptor compared to the improved Radial Descriptor. 55
4.5 Mean error and standard deviation w.r.t. the tilt angle θ. . . . 62

5.1 Values for the refinement procedure 85
5.2 Parameters 86
5.3 Classification of the vertebrae 91
5.4 Execution time 96

7.1 Input parameters for pattern detection. 117
7.2 Data set description 124
7.3 Candidates for the feature selection process 125
7.4 Classification rate 126
7.5 Evaluation of classification 128

10.1 Data sets used in the evaluation. 156
10.2 Pruning power of Sub Cubes 159
10.3 Pruning power of Sub Cubes + MBRs 160
10.4 Total amount of data viewed. 160

11.1 Performance characteristics of used devices 170

List of Algorithms

1 Refinement process 85
2 Peak Reconstruction 115
3 Identification of periodic patterns 117
4 Peak Detection 121

Bibliography

[1] MD Abramoff, PJ Magelhaes, and SJ Ram. Image processing with ImageJ. *Biophotonics international*, 11(7):36–42, 2004.

[2] E. Achtert, T. Bernecker, H.-P. Kriegel, E. Schubert, and A. Zimek. ELKI in time: ELKI 0.2 for the performance evaluation of distance measures for time series. In *Proceedings of the 11th International Symposium on Spatial and Temporal Databases (SSTD), Aalborg, Denmark*, 2009.

[3] C. C. Aggarwal. Re-designing distance functions and distance-based applications for high dimensional data. *ACM SIGMOD Record*, 30(1):13–18, 2001.

[4] C. C. Aggarwal, A. Hinneburg, and D. Keim. On the surprising behavior of distance metrics in high dimensional space. In *Proceedings of the 8th International Conference on Database Theory (ICDT), London, UK*, 2001.

[5] C. C. Aggarwal and P. S. Yu. On high dimensional indexing of uncertain data. In *Proceedings of the 24th International Conference on Data Engineering (ICDE), Cancun, Mexico*, 2008.

[6] F. R. Allen, E. Ambikairajah, N. H. Lovell, and B. G. Celler. An Adapted Gaussian Mixture Model Approach to Accelerometry-Based Movement Classification Using Time-Domain Features. In *Proc. 28th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society EMBS '06*, pages 3600–3603, 2006.

[7] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122, 2008.

[8] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems 15*. MIT Press, Cambridge, MA, 2003.

[9] N. Arora, D. Martins, D. Ruggerio, E. Tousimis, A.J. Swistel, M.P. Osborne, and R.M. Simmons. Effectiveness of a noninvasive digital infrared thermal imaging system in the detection of breast cancer. *The American Journal of Surgery*, 196(4):523–526, 2008.

[10] Ling Bao and Stephen Intille. Activity Recognition from User-Annotated Acceleration Data. In Alois Ferscha and Friedemann Mattern, editors, *Pervasive Computing*, volume 3001 of *Lecture Notes in Computer Science*, pages 1–17. Springer Berlin / Heidelberg, 2004.

[11] Herbert Bay, Tinne Tuytelaars, and Luc J. Van Gool. Surf: Speeded up robust features. In Ales Leonardis, Horst Bischof, and Axel Pinz, editors, *ECCV*, volume 3951 of *Lecture Notes in Computer Science*, pages 404–417. Springer, 2006.

[12] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-Tree: An efficient and robust access method for points and rectangles. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD), Atlantic City, NJ*, 1990.

[13] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *In NIPS*, pages 831–837, 2000.

[14] K. P. Bennett, U. Fayyad, and D. Geiger. Density-based indexing for approximate nearest-neighbor queries. In *Proceedings of the 5th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), San Diego, CA*, 1999.

[15] S. Berchtold, C. Böhm, H. V. Jagadish, H.-P. Kriegel, and J. Sander. Independent Quantization: An index compression technique for high-dimensional data spaces. In *Proceedings of the 16th International Conference on Data Engineering (ICDE), San Diego, CA*, 2000.

[16] T. Bernecker, T. Emrich, F. Graf, H.-P. Kriegel, P. Kröger, M. Renz, E. Schubert, and A. Zimek. Subspace similarity search: Efficient k-nn queries in arbitrary subspaces. In *Proceedings of the 22nd International Conference on Scientific and Statistical Database Management (SSDBM), Heidelberg, Germany*, 2010.

[17] T. Bernecker, T. Emrich, F. Graf, H.-P. Kriegel, P. Kröger, M. Renz, E. Schubert, and A. Zimek. Subspace similarity search using the ideas of ranking and top-k retrieval. In *Proceedings of the 26th International Conference on Data Engineering (ICDE) Workshop on Ranking in Databases (DBRank), Long Beach, CA*, 2010.

[18] T. Bernecker, F. Graf, H.-P. Kriegel, C. Moennig, and A. Zimek. Beyond - unleashing bond. In *Proceedings of the 37th International Conference on very Large Data Bases (VLDB) Workshop on Ranking in Databases (DBRank), Seattle, WA*, 2011.

[19] T. Bernecker, M. E. Houle, H.-P. Kriegel, P. Kröger, M. Renz, E. Schubert, and A. Zimek. Quality of similarity rankings in time series. In *Proceedings of the 12th International Symposium on Spatial and Temporal Databases (SSTD), Minneapolis, MN*, 2011.

[20] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In *Proceedings of the 7th International Conference on Database Theory (ICDT), Jerusalem, Israel*, 1999.

[21] Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. MOA: Massive online analysis. In *Journal of Machine Learning Research (JMLR)*, 2010.

[22] C. Böhm, S. Berchtold, and D. A. Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys*, 33(3):322–373, 2001.

[23] C. Böhm and H.-P. Kriegel. Dynamically optimizing high-dimensional index structures. In *Proceedings of the 7th International Conference on Extending Database Technology (EDBT), Konstanz, Germany*, 2000.

[24] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 7th ACM International Symposium on Advances in Geographic Information Systems (ACM GIS), Kansas City, KS*, 2007.

[25] C. V. C. Bouten, K. T. M. Koekkoek, M. Verduin, R. Kodde, and J. D. Janssen. A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity. *IEEE Transactions on Biomedical Engineering*, 44(3):136–147, 1997.

[26] David J. Brenner and Eric J. Hall. Computed tomography – an increasing source of radiation exposure. *New England Journal of Medicine*, 357(22):2277–2284, 2007.

[27] Corinna Bürger. *Automatic Localisation of Body Regions in CT Topograms*. VDM Verlag Dr. Müller, Saarbrücken, 1st edition, 2008.

[28] F. John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[29] JK Choi, K. Miki, S. Sagawa, and K. Shiraki. Evaluation of mean skin temperature formulas by infrared thermography. *International Journal of Biometeorology*, 41(2):68–75, 1997.

[30] J. Corso, R. Alomari, and V. Chaudhary. Lumbar disc localization and labeling with a probabilistic model on both pixel and object features. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008*, pages 202–210, 2008.

[31] B. Cowling, L. Lau, P. Wu, H. Wong, V. Fang, S. Riley, and H. Nishiura. Entry screening to delay local transmission of 2009 pandemic influenza a (h1n1). *BMC Infectious Diseases*, 10(1):82, 2010.

[32] A. Criminisi, J. Shotton, D. Robertson, and E. Konukoglu. Regression Forests for efficient anatomy detection and localization in CT studies. *Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging*, pages 106–117, 2011.

[33] Ritendra Datta, Jia Li, and James Ze Wang. Content-based image retrieval: approaches and trends of the new age. In HongJiang Zhang, John Smith, and Qi Tian, editors, *Multimedia Information Retrieval*, pages 253–262. ACM, 2005.

[34] A. P. de Vries, N. Mamoulis, N. Nes, and M. Kersten. Efficient k-NN search on vertically decomposed data. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD), Madison, WI*, 2002.

[35] Deutsches Röntgen Museum. Chronik 100 Jahre Anwendungen der Röntgenstrahlen in der medizinischen Diagnostik. http://www.roentgenmuseum.de/fileadmin/bilder/PDF/ChronikDiagnostik.pdf. Accessed 2001/07/21.

[36] J. Dittrich, L. Blunschi, and M. A. V. Salles. Dwarfs in the rearview mirror: How big are they really? In *Proceedings of the 34nd International Conference on Very Large Data Bases (VLDB), Auckland, New Zealand*, 2008.

[37] Jaeyoung Do and Jignesh M. Patel. Join processing for flash SSDs: remembering past lessons. In *Proceedings of the 5th International Workshop on Data Management on New Hardware (DaMoN), Providence, RI*, pages 1–8, 2009.

[38] K.T. Dussik. Über die Möglichkeit, hochfrequente mechanische Schwingungen als diagnostisches Hilfsmittel zu verwerten. *Zeitschrift für die gesamte Neurologie und Psychiatrie*, 174(1):153–168, 1942.

[39] T. Emrich, F. Graf, H.-P. Kriegel, M. Schubert, and M. Thoma. On the impact of flash SSDs on spatial indexing. In *Proceedings of the 6th International Workshop on Data Management on New Hardware (DaMoN), Indianapolis, IN*, 2010.

[40] T. Emrich, F. Graf, H.-P. Kriegel, M. Schubert, M. Thoma, and A. Cavallaro. CT slice localization via instance-based regression. In *Proceedings of the SPIE Medical Imaging 2010: Image Processing (SPIE), San Diego, CA*, volume 7623, page 762320, 2010.

[41] SA Fausti, DA Erickson, RH Frey, and BZ Rappaport. The effects of impulsive noise upon human hearing sensitivity (8 to 20 khz). *Scandinavian Audiology*, 10(1):21–29, 1981.

[42] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *IEEE CVPR 2004, Workshop on Generative-Model Based Vision*, 2004.

[43] J. Feulner, S. K. Zhou, S. Seifert, A. Cavallaro, J. Hornegger, and D. Comaniciu. Estimating the body portion of CT volumes by matching histograms of visual words. In *Proceedings of the SPIE Medical Imaging 2009 Conference (SPIE), Lake Buena Vista, FL*, 2009.

[44] R. A. Finkel and J. L. Bentley. Quad trees. a data structure for retrieval on composite keys. *Acta Informatica*, 4(1):1–9, 1974.

[45] Bundesamt für Strahlenschutz. Röntgendiagnostik: Häufigkeit und strahlenexposition. http://www.bfs.de/de/ion/medizin/diagnostik/roentgen/haeufigkeit_strahlenexposition.html, Feb 2011. Accessed 2011/07/26.

[46] D. Francois, V. Wertz, and M. Verleysen. The concentration of fractional distances. *IEEE Transactions on Knowledge and Data Engineering*, 19(7):873–886, 2007.

[47] V. Gaede and O. Günther. Multidimensional access methods. *ACM Computing Surveys*, 30(2):170–231, 1998.

[48] M. Gautherie and C.M. Gros. Breast thermography and cancer risk prediction. *Cancer*, 45(1):51–56, 1980.

[49] J. M. Geusebroek, G. J. Burghouts, and A.W.M. Smeulders. The Amsterdam Library of Object Images. *International Journal of Computer Vision*, 61(1):103–112, 2005.

[50] MO Güld, M. Kohnen, D. Keysers, H. Schubert, B.B. Wein, J. Bredno, and T.M. Lehmann. Quality of DICOM header information for image categorization. In *Proceedings of the SPIE Medical Imaging 2002 Conference (SPIE), San Diego, CA*, volume 4685, pages 280–287, 2002.

[51] G. Graefe. The five-minute rule twenty years later, and how flash memory changes the rules. In *Proceedings of the 3rd International Workshop on Data Management on New Hardware (DaMoN), Beijing, China*, pages 1–9, 2007.

[52] F. Graf, R. Greil, H.-P. Kriegel, M. Schubert, and A. Cavallaro. Enhanced Detection of the Vertebrae in 2D CT-Images. In *Proceedings of the SPIE Medical Imaging 2011: Image Processing (SPIE), San Diego, FL*, 2012.

[53] F. Graf, H.-P. Kriegel, S. Pölsterl, M. Schubert, and A. Cavallaro. Position prediction in ct volume scans. In *Proceedings of the 28th International Conference on Machine Learning (ICML) Workshop on Learning for Global Challenges, Bellevue, Washington, WA*, 2011.

[54] F. Graf, H.-P. Kriegel, M. Schubert, S. Pölsterl, and A. Cavallaro. 2d image registration in ct images using radial image descriptors. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI), Toronto, Canada*, volume 6892 of *Lecture Notes in Computer Science*, pages 607–614. Springer, 2011.

[55] F. Graf, H.-P. Kriegel, M. Schubert, M. Strukelj, and A. Cavallaro. Fully automatic detection of the vertebrae in 2D CT images. In *Proceedings of the SPIE Medical Imaging 2011: Image Processing (SPIE), Lake Buena Vista, FL*, 2011.

[56] R. L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Information processing Letters*, 1972.

[57] J. Gray and G. Graefe. The five-minute rule ten years later, and other computer storage rules of thumb. *ACM SIGMOD Record*, 26(4):63–68, 1997.

[58] A. Guttman. R-Trees: A dynamic index structure for spatial searching. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD), Boston, MA*, 1984.

[59] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157 – 1182, 2003.

[60] B. Haas, T. Coradi, M. Scholz, P. Kunz, M. Huber, U. Oppitz, L. André, V. Lengkeek, D. Huyskens, A. van Esch, and R. Reddick. Automatic segmentation of thoracic and pelvic CT images for radiotherapy planning using implicit anatomic knowledge and organ-specific segmentation strategies. *Physics in Medicine and Biology*, 53(6):1751–71, 2008.

[61] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations*, 11(1):10–18, 2009.

[62] T.J. Hall. Aapm/rsna physics tutorial for residents: Topics in us. *Radiographics*, 23(6):1657, 2003.

[63] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Speech and Audio Processing*, 3(6):6103–623, 1973.

[64] X. He. Incremental semi-supervised subspace learning for image retrieval. In *Proceedings of the 13th ACM International Conference on Multimedia (ACM MM), Singapore*, 2005.

[65] Ernst A. Heinz, Kai Kunze, Matthias Gruber, David Bannach, and Paul Lukowicz. Using Wearable Sensors for Real-Time Recognition Tasks in Games of Martial Arts - An Initial Experiment. In *in Proceedings of the 2nd IEEE Symposium on Computational Intelligence and Games (CIG)*, pages 98–102. IEEE Press, 2006.

[66] A. Hinneburg, C. C. Aggarwal, and D. A. Keim. What is the nearest neighbor in high dimensional spaces? In *Proceedings of the 26th International Conference on Very Large Data Bases (VLDB), Cairo, Egypt*, 2000.

[67] M. E. Houle, H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Can shared-neighbor distances defeat the curse of dimensionality? In *Proceedings of the 22nd International Conference on Scientific and Statistical Database Management (SSDBM), Heidelberg, Germany*, 2010.

[68] S.H. Huang, Y.H. Chu, S.H. Lai, and C.L. Novak. Learning-based vertebra detection and iterative normalized-cut segmentation for spinal MRI. *IEEE Transactions on Medical Imaging*, 28(10):1595, 2009.

[69] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing (STOC), Dallas, TX*, 1998.

[70] A.F. Jerant, J.T. Johnson, C. Sheridan, and T.J. Caffrey. Early detection and treatment of skin cancer. *American family physician*, 62(2):357–386, 2000.

[71] LJ Jiang, EYK Ng, ACB Yeo, S. Wu, F. Pan, WY Yau, JH Chen, and Y. Yang. A perspective on medical infrared imaging. *Journal of medical engineering & technology*, 29(6):257–267, 2005.

[72] H. Jin, B. C. Ooi, H. T. Shen, C. Yu, and A. Y. Zhou. An adaptive and efficient dimensionality reduction algorithm for high-dimensional indexing. In *Proceedings of the 19th International Conference on Data Engineering (ICDE), Bangalore, India*, 2003.

[73] B.F. Jones. A reappraisal of the use of infrared thermal image analysis in medicine. *Medical Imaging, IEEE Transactions on*, 17(6):1019–1027, 1998.

[74] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE Transactions on Information Technology in Biomedicine*, 10(1):156–167, 2006.

[75] N. Katayama and S. Satoh. Distinctiveness-sensitive nearest-neighbor search for efficient similarity retrieval of multimedia information. In *Proceedings of the 17th International Conference on Data Engineering (ICDE), Heidelberg, Germany*, 2001.

[76] D.A. Kennedy, T. Lee, and D. Seely. A comparative review of thermography as a breast cancer screening technique. *Integrative cancer therapies*, 8(1):9, 2009.

[77] A. M. Khan, Young-Koo Lee, S. Y. Lee, and Tae-Seong Kim. A Triaxial Accelerometer-Based Physical-Activity Recognition via Augmented-Signal Features and a Hierarchical Recognizer. *IEEE Transactions on Biomedical Engineering*, 14(5):1166–1172, 2010.

[78] Adil Khan, Young-Koo Lee, Sungyoung Lee, and Tae-Seong Kim. Accelerometer's position independent physical activity recognition system for long-term activity monitoring in the elderly. *Medical and Biological Engineering and Computing*, 48:1271–1279, 2010.

[79] Y. Kim and D. Kim. A fully automatic vertebra segmentation method using 3D deformable fences. *Computerized medical imaging and graphics: the official journal of the Computerized Medical Imaging Society*, 2009.

[80] Jon M. Kleinberg. Two algorithms for nearest-neighbor search in high dimensions. In *STOC*, pages 599–608, 1997.

[81] Tobias Klinder, Jörn Ostermann, Matthias Ehm, Astrid Franz, Reinhard Kneser, and Cristian Lorenz. Automated model-based vertebra detection, identification, and segmentation in ct images. *Medical Image Analysis*, 13(3):471 – 482, 2009.

[82] M. Kornacker. High-performance extensible indexing. In *Proceedings of the 25th International Conference on Very Large Data Bases (VLDB), Edinburgh, Scotland*, 1999.

[83] M. Koskela, J. Laaksonen, and E. Oja. Use of image subset features in image retrieval with self-organizing maps. In *Proceedings of the 3rd International Conference on Image and Video Retrieval (CIVR), Dublin, Ireland*, 2004.

[84] A. Krause, M. Ihmig, E. Rankin, D. Leong, Smriti Gupta, D. Siewiorek, A. Smailagic, M. Deisher, and U. Sengupta. Trading off prediction accuracy and power consumption for context-aware wearable computing. In *Proc. Ninth IEEE Int Wearable Computers Symp*, pages 20–26, 2005.

[85] H.-P. Kriegel, P. Kröger, M. Schubert, and Z. Zhu. Efficient query processing in arbitrary subspaces using vector approximations. In *Proceedings of the 18th International Conference on Scientific and Statistical Database Management (SSDBM), Vienna, Austria*, 2006.

[86] H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *IEEE Transactions on Knowledge Discovery and Data Mining*, 3(1):1–58, 2009.

[87] A. Kurjak, N. Vecek, T. Hafner, T. Bozek, B. Funduk-Kurjak, and B. Ujevic. Prenatal diagnosis: what does four-dimensional ultrasound add? *Journal of perinatal medicine*, 30(1):57–62, 2002.

[88] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, 12:74–82, March 2011.

[89] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06), New York, NY, USA*, 2:2169–2178, 2006.

[90] S.-W. Lee, B. Moon, C. Park, J.-M. Kim, and S.-W. Kim. A case for flash memory SSD in enterprise database applications. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD), Vancouver, BC*, 2008.

[91] M.L. Lenhardt, R. Skellett, P. Wang, and A.M. Clarke. Human ultrasonic speech perception. *Science*, 253(5015):82, 1991.

[92] Y. Li, B. He, Q. Luo, and K. Yi. Tree indexing on flash disks. In *Proceedings of the 25th International Conference on Data Engineering (ICDE), Shanghai, China*, pages 1303–1306, 2009.

[93] X. Lian and L. Chen. Similarity search in arbitrary subspaces under Lp-norm. In *Proceedings of the 24th International Conference on Data Engineering (ICDE), Cancun, Mexico*, 2008.

[94] O. Maron and A.L. Ratan. Multiple-instance learning for natural scene classification. In *Proceedings of the 15th International Conference on Machine Learning (ICML), Madison, WI*, 1998.

[95] Peter W. Michor. *Seventy-Five Years or Radon Transform: Proceedings of the Conference Held at the Erwin Schrodinger International Institute for Mathematical Physics in ... Lecture Notes in Mathematical Physics, V. 4).* Intl Pr, 1994.

[96] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler. YALE: Rapid prototyping for complex data mining tasks. In *Proceedings of the*

12th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Philadelphia, PA, 2006.

[97] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1615–1630, 2005.

[98] Bamshad Mobasher, Honghua Dai, Tao Luo, and Miki Nakagawa. Effective personalization based on association rule discovery from web usage data. In *Proc. of the 3rd int. workshop on Web information and data management (WIDM '01)*, pages 9–15, New York, NY, USA, 2001. ACM.

[99] W. Müller and A. Henrich. Faster exact histogram intersection on large data collections using inverted VA-files. In *Proceedings of the 3rd International Conference on Image and Video Retrieval (CIVR), Dublin, Ireland*, 2004.

[100] Suman Nath and Aman Kansal. FlashDB: dynamic self-tuning database for NAND flash. In *Proceedings of the 6th International Conference on Information Processing In Sensor Networks (IPSN), Cambridge, MA*, pages 410–419, 2007.

[101] H. Nishiura and K. Kamiya. Fever screening during the influenza (h1n1-2009) pandemic at narita international airport, japan. *BMC Infectious Diseases*, 11(1):111, 2011.

[102] NobelPrize.org. The Nobel Prize in Medicine 2003. http://nobelprize.org/nobel_prizes/medicine/laureates/2003/, Jul 2011. Accessed 2001/07/21.

[103] NobelPrize.org. The Nobel Prize in Physics 1901. http://nobelprize.org/nobel_prizes/physics/laureates/1901/, Jul 2011. Accessed 2001/07/21.

[104] NobelPrize.org. The Nobel Prize in Physiology or Medicine 1979. http://nobelprize.org/nobel_prizes/medicine/laureates/1979/, Jul 2011. Accessed 2001/07/21.

[105] M.N. Nyan, F.E.H. Tay, K.H.W. Seah, and Y.Y. Sitoh. Classification of gait patterns in the time-frequency domain. *Journal of Biomechanics*, 39(14):2647 – 2656, 2006.

[106] L.G. Nyúl, J. Kanyó, E. Máté, G. Makay, E. Balogh, M. Fidrich, and A. Kuba. Method for automatically segmenting the spinal cord and canal from 3D CT images. In *Computer Analysis of Images and Patterns*, pages 456–463. Springer, 2005.

[107] C.P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proceedings of the Sixth International Conference on Computer Vision*, page 555. IEEE Computer Society, 1998.

[108] DD Pascoe, EF Ring, JB Mercer, J. Snell, D. Osborn, and J. Hedley-Whyte. International standards for pandemic screening using infrared thermography. In *Proceedings of the SPIE Medical Imaging 2010: Image Processing (SPIE), San Diego, CA*, volume 7626, page 76261Z, 2010.

[109] T. Pavlidis and Y.T. Liow. Integrating region growing and edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 225–233, 1990.

[110] Susanna Pirttikangas, Kaori Fujinami, and Tatsuo Nakajima. Feature Selection and Activity Recognition from Wearable Sensors. In Hee Youn, Minkoo Kim, and Hiroyuki Morikawa, editors, *Ubiquitous Computing Systems*, volume 4239 of *Lecture Notes in Computer Science*, pages 516–527. Springer Berlin / Heidelberg, 2006.

[111] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*. MIT Press, 1998.

[112] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, and D. Howard. A Comparison of Feature Extraction Methods for the Classification of Dynamic Activities From Accelerometer Data. *IEEE Transactions on Biomedical Engineering*, 56(3):871–879, 2009.

[113] RJ Pumphrey. Upper limit of frequency for human hearing. *Nature*, 166, 1950.

[114] G. Qian, S. Sural, Y. Gu, and S. Pramanik. Similarity between euclidean and cosine angle distance for nearest neighbor queries. In *Proceedings of the 2004 ACM symposium on Applied computing*, 2004.

[115] J. Radon. Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten. *Berichte Sächsische Akademie der Wissenschaften*, 69:262–267, 1917.

[116] R. M. Rangayyan, H. J. Deglint, and G. S. Boag. Method for the automatic detection and segmentation of the spinal canal in computed tomographic images. *Journal of Electronic Imaging*, 15(3):033007, 2006.

[117] W. S. Rasband. ImageJ. In *U. S. National Institutes of Health, Bethesda, Maryland, USA, http://rsb.info.nih.gov/ij/*, 1997-2009.

[118] Siegfried Reball. *Effekte der Physik und ihre Anwendungen.* Deutsch Harri GmbH, 2004.

[119] M. Sadeghi, M. Razmara, M. Ester, TK Lee, and MS Atkins. Graph-based pigment network detection in skin images. In *Proceedings of the SPIE Medical Imaging 2010: Image Processing (SPIE), San Diego, CA*, volume 7623, page 762312, 2010.

[120] H. Samet. *Foundations of Multidimensional and Metric Data Structures.* Morgan Kaufmann, San Francisco, 2006.

[121] A.F. Scarsbrook, A. Ganeshan, J. Statham, R.V. Thakker, A. Weaver, D. Talbot, P. Boardman, K.M. Bradley, F.V. Gleeson, and R.R. Phillips. Anatomic and functional imaging of metastatic carcinoid tumors. *Radiographics*, 27(2):455, 2007.

[122] Stefan Schmidt, Jörg Kappes, Martin Bergtholdt, Vladimir Pekar, Sebastian Dries, Daniel Bystrov, and Christoph Schnörr. Spine detection and labeling using a parts-based graphical model. In *IPMI'07: Proceedings of the 20th international conference on Information processing in medical imaging*, pages 122–133, Berlin, Heidelberg, 2007. Springer-Verlag.

[123] S. Seifert, A. Barbu, S.K. Zhou, D. Liu, J. Feulner, M. Huber, M. Suehling, A. Cavallaro, and D. Comaniciu. Hierarchical parsing and semantic navigation of full body CT data. In *Proceedings of the SPIE Medical Imaging 2009 Conference (SPIE), Lake Buena Vista, FL*, volume 7259, page 725902, 2009.

[124] Sascha Seifert, Michael Kelm, Manuel Moeller, Saikat Mukherjee, Alexander Cavallaro, Martin Huber, and Dorin Comaniciu. Semantic annotation of medical images. In Brent J. Liu and William W. Boonn, editors, *Proceedings of the SPIE Medical Imaging 2010: Image Processing (SPIE), San Diego, CA*, volume 7628, page 762808. SPIE, 2010.

[125] Mehul A. Shah, Stavros Harizopoulos, Janet L. Wiener, and Goetz Graefe. Fast scans and joins using flash drives. In *Proceedings of the 4th International Workshop on Data Management on New Hardware (DaMoN), Vancouver, BC*, pages 17–24, 2008.

[126] Siemens. Computertomographie Geschichte und Technologie. http://www.medical.siemens.com/siemens/de_DE/rg_marcom_FBAs/files/Hintergrundinfos/CT_Geschichte.pdf. Accessed 2001/07/21.

[127] Jonathan M. Smith and Shih-Fu Chang. Quad-tree segmentation for texture-based image query. In *ACM Multimedia*, pages 279–286, 1994.

[128] D. Štern, B. Likar, F. Pernuš, and T. Vrtovec. Automated detection of spinal centrelines, vertebral bodies and intervertebral discs in CT and

MR images of lumbar spine. *Physics in Medicine and Biology*, 55:247, 2010.

[129] D. Štern, T. Vrtovec, F. Pernuš, and B. Likar. Automated determination of spinal centerline in CT and MR images. In *Proceedings of the SPIE Medical Imaging 2009 Conference (SPIE), Lake Buena Vista, FL*, volume 7259, pages 72594M–1, 2009.

[130] D. Stern, T. Vrtovec, F. Pernus, and B. Likar. Segmentation of vertebral bodies in ct and mr images based on 3d deterministic models. In *Proceedings of the SPIE Medical Imaging 2011: Image Processing (SPIE), Lake Buena Vista, FL*, 2011.

[131] A. Strehl, J. Ghosh, and R. Mooney. Impact of similarity measures on web-page clustering. In *Workshop on Artificial Intelligence for Web Search (AAAI 2000)*, pages 58–64, 2000.

[132] A. Sugimoto, Y. Hara, T. W. Findley, and K. Yoncmoto. A useful method for measuring daily physical activity by a three-direction monitor. *Scandinavian Journal of Rehabilitation Medicine*, 29(1):37–42, Mar 1997.

[133] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118. ACM, 2001.

[134] D. Tsirogiannis, S. Harizopoulos, M. A. Shah, J. L. Wiener, and G. Graefe. Query processing techniques for solid state drives. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD), Providence, RI*, pages 59–72, 2009.

[135] Zhuowen Tu. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. *Computer Vision, IEEE International Conference on*, 2:1589–1596, 2005.

[136] C. Türmer, D. Dill, A. Scholz, M. Gül, T. Bernecker, F. Graf, H.-P. Kriegel, and B. Wolf. Concept of a medical activity monitoring system

improving the dialog between doctors and patients concerning preventions, diagnostics and therapies. In *Forum Medizin 21, Evidenzbasierte Medizin (EbM), Salzburg, Austria*, 2010.

[137] C. Türmer, D. Dill, A. Scholz, M. Gül, A. Stautner, T. Bernecker, F. Graf, and B. Wolf. Conceptual design for an activity monitoring system concerning medical applications using triaxial accelerometry. In *Austrian Society for Biomedical Engineering (BMT), Rostock, Germany*, 2010.

[138] Christoph Simon Türmer. Konzeptionierung eines Aktivitätsmonitoring-Systems für medizinische Applikationen mit dem 3D-Accelerometer der Sendsor GmbH. Diploma Thesis, Technische Universität München, 2009.

[139] Paul A. Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01), Kauai, HI*, pages 511–518. IEEE Computer Society, 2001.

[140] T. Vrtovec, B. Likar, and F. Pernuš. Quantitative analysis of spinal curvature in 3D: application to CT images of normal spine. *Physics in medicine and biology*, 53:1895, 2008.

[141] T. Vrtovec, F. Pernus, and B. Likar. Determination of vertebral pose in 3D by minimization of vertebral asymmetry. In *Proceedings of the SPIE Medical Imaging 2011: Image Processing (SPIE), Lake Buena Vista, FL*, 2011.

[142] T. Vrtovec, D. Tomazevic, B. Likar, L. Travnik, and F. Pernus. Automated construction of 3 D statistical shape models. *Image Analysis & Stereology*, 23(2):111–120, 2004.

[143] Jamie A. Ward, Paul Lukowicz, and Gerhard Tröster. Gesture spotting using wrist worn microphone and 3-axis accelerometer. In *Proceedings*

of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies, pages 99–104, New York, NY, USA, 2005. ACM.

[144] R. Weber, H.-J. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of the 24th International Conference on Very Large Data Bases (VLDB), New York City, NY*, 1998.

[145] JD Whited, BJ Mills, RP Hall, RJ Drugge, JM Grichnik, and DL Simel. A pilot trial of digital imaging in skin cancer. *Journal of Telemedicine and Telecare*, 4(2):108, 1998.

[146] C.-H. Wu, L.-P. Chang, and T.-W. Kuo. An efficient B-tree layer for flash-memory storage systems. In *Proceedings of the 9th International Conference on Real-Time and Embedded Computing Systems and Applications (RTCSA), Tainan, Taiwan*, pages 17–24, 2003.

[147] C.-H. Wu, L.-P. Chang, and T.-W. Kuo. An efficient R-tree implementation over flash-memory storage systems. In *Proceedings of the 11th ACM International Symposium on Advances in Geographic Information Systems (ACM GIS), New Orleans, LA*, pages 17–24, 2003.

[148] L. Xu, M. Jackowski, A. Goshtasby, D. Roseman, S. Bines, C. Yu, A. Dhawan, and A. Huntley. Segmentation of skin cancer images. *Image and Vision Computing*, 17(1):65 – 74, 1999.

[149] Jhun-Ying Yang, Jeen-Shing Wang, and Yen-Ping Chen. Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers. *Pattern Recogn. Lett.*, 29:2213–2220, December 2008.

[150] W. Wee Z. Peng, J. Zhong and J. h. Lee. Automated vertebra detection and segmentation from the whole spineMR images. In *27th Int. Conf. of the Engineering in Medicine and Biology Society IEEE - EMBC '05*, 2005.

[151] Chengcui Zhang, Xin Chen, Min Chen, Shu-Ching Chen, and Mei-Ling Shyu. A multiple instance learning approach for content based image retrieval using one-class support vector machine. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Amsterdam, Netherlands*, pages 1142–1145. IEEE, 2005.

[152] Tong Zhang. Adaptive Forward-Backward Greedy Algorithm for Sparse Learning with Linear Models. In *NIPS'08*, pages 1921–1928, 2008.

Acknowledgements

I would like to express my gratitude to my supporter and doctoral adviser Prof. Dr. Hans-Peter Kriegel who offered me the opportunity to enter his research group and do all this fascinating research within the past years.

I also want to thank all my colleagues and the staff of Prof. Kriegel's group who helped with words and deeds and several discussions through all the time of my research. Besides my colleagues I have received helpful support from several student assistants. Namely I want to thank Robert Forbrig, Robert Greil, Sebastian Pölsterl, Michael Strukelj, Nepomuk Seiler, Christian Mönnig and Michael Weiler for their great work. Also I want to thank Alexander Scholz, Dieter Dill, and Christoph Türmer from the Sendsor GmbH for the productive collaboration and for providing the hardware sensors needed to record data for this work.

Yet, the biggest dept of gratitude must go to Patricia Hurst who supported me mentally through all the time of my research and my family which encouraged me on all my way, wherever it took me.

This research has been supported in part by the THESEUS program which is funded by the German Federal Ministry of Economics and Technology under the grant number 01MQ07020.

About the author

Franz Graf was born on March, 22. 1980 in Bad Tölz, Germany. He finished his secondary education at the Gabriel–von–Seidl Gymnasium in Bad Tölz in 1999 and gathered experience as employed and freelancing web developer.

After his military service in the air force, he began studying computer science at the Ludwig-Maximilians-Universität (LMU) München in 2001. During his studies he was freelancing in web development and small management tasks. In 2006 he was working at Siemens Corporate Technology (CT) in Erlangen with Siemens automation systems as a working student. In 2008 he successfully finished his diploma thesis "*Erkennung von Objekten in Bildern anhand von 3D Modellen*" in Prof. Kriegel's Group.

Afterwards he started in Prof. Kriegels Group as a researcher in the THESEUS Research Project in the Core Technology Cluster (CTC), Workpackage 6 "Machine Learning" and cooperated with Siemens AG, the Imaging Science Institute (ISI) Erlangen, Fraunhofer IDMT, Fraunhofer FIRST and the TU Berlin. During this time, he also initialized cooperations with the Sendsor GmbH and Mayflower GmbH.

His research involved computer vision, indexing, medical imaging and the analysis of sensor data. He also invested some spare time to develop a framework for routing queries in OpenStreetMap data which was used in some student works and publications that he co-authored.