LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR INFORMATIK
LEHR- UND FORSCHUNGSEINHEIT
FÜR DATENBANKSYSTEME

Dissertation

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

Similarity Search Applications in Medical Images

Marisa Petri

München, den 18.07.2011

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

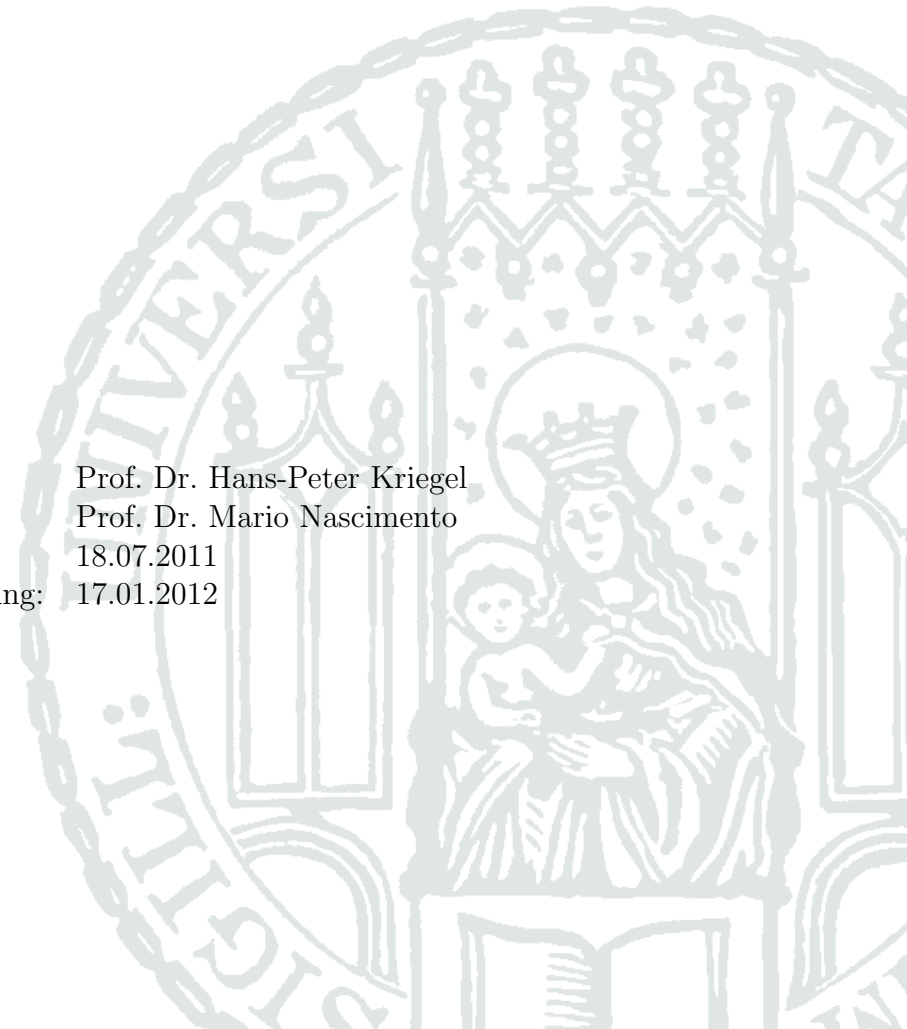
INSTITUT FÜR INFORMATIK
LEHR- UND FORSCHUNGSEINHEIT
FÜR DATENBANKSYSTEME

Dissertation

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

Similarity Search Applications in Medical Images

Marisa Petri



Erstgutachter:	Prof. Dr. Hans-Peter Kriegel
Zweitgutachter:	Prof. Dr. Mario Nascimento
Abgabetermin:	18.07.2011
Tag der mündlichen Prüfung:	17.01.2012

Erklärung

Hiermit versichere ich, dass ich diese Dissertation selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 18.07.2011

.....
Marisa Petri

Acknowledgements

This work was financed by the German Federal Ministry of Economics and Technology under the grant number 01MQ07020 within the research program THESEUS MEDICO. The responsibility for this publication lies with the author.

I would like to thank everybody who supported me in the creation process of this thesis in the past four years.

My greatest thanks goes to my supervisor, Prof. Dr. Hans-Peter Kriegel, for enabling me to work on this interesting area of research and for his valuable advice in any matter related to my work. Your group has provided me with a great research, working and teaching environment.

Second, I would like to thank Dr. Matthias Schubert, who is blessed with extraordinary mentoring skills and a wealth of ideas, and who has co-initiated and actively participated in the THESEUS MEDICO project.

I also thank my co-worker and project partner in almost any work-related field, Franz Graf, for his uncomplicated cooperation and his countless technical tips.

The same dedication goes to the remaining DBS research group, starting with our unforgettable secretary Susanne Grienberger, continuing with the scientific staff of research and teaching assistants or assistant professors, climaxing with our system administrator Franz Krojer, whom I frequently occupied with exploding data sizes or mysterious system environment bugs.

Furthermore, I express my thanks to the THESEUS MEDICO group for multiple meetings, joint reports, the blessing and curse of a concerted prototype and an El Dorado of medical data. The MEDICO core team, compiled from a cooperation of SIEMENS CT, the University Hospital Erlangen, the Fraunhofer Institut für Graphische Datenverarbeitung (IGD), the Deutsches Forschungsinstitut für Künstliche Intelligenz (DFKI) and my university, Ludwig-Maximilians-Universität München (LMU), is currently lead by Dr. Sascha Seifert In addition, I supervised the student assistants Michael Gruber, Dr. Robert Forbrig, Sebastian Pölsterl and Michael Shekelyan, who assisted me in matters of implementation and data annotation.

Same holds for the students for whom I supervised student projects and

theses: Justin Heesemann, Manuel Klette, Bernhard Meyer, Florian Schlegel, Michael Shekelyan, Michael Stockerl, Annika Tonch and Christoph Wagner were a source of inspiration and a help with the test of the resulting ideas.

Finally, I would like to thank Dr. Karsten Borgwardt for acting as my first mentor in the field of computer science and the close cooperation in the field of graph mining.

And of course: A large thank you to my family and friends for giving me the free space I needed for completing this thesis, but also for providing me with the occasional much-needed distraction. And in case of my husband Tobias: double that, thanks for last-minute proof-reading and I forgive you for re-arranging my family name from Thoma to Petri, thus leaving me without any publications under my current name.

Abstract

The advances in medical imaging have enabled the large-scale generation of medical image data in hospitals and private clinics. While taking a high-resolution scan of a troubling body part has been rather exceptional in the past, it is now common practise for a large variety of patient conditions.

The research project THESEUS MEDICO aims at the re-organization of medical picture archiving and communication systems (PACS) such that they can be efficiently queried for more advanced image meta-data than a small set of common identifiers. As a complement to keyword-based retrieval queries, MEDICO also offers a variety of content-based image retrieval (CBIR) applications.

Two use cases employing CBIR techniques for answering clinical queries on Computed Tomography (CT) scans have been developed in the course of this thesis. The first application provides a standardized body coordinate system using image-based landmarks and further image information for automatically providing an anatomical context for any given scan region and for greatly accelerating retrieval queries. The second application is a more direct form of CBIR, enabling the specification of a visual query template which is used for retrieving similar image patterns.

This thesis introduces a variety of newly-developed similarity search components and it evaluates the generated medical search framework at the example of the two search applications.

Zusammenfassung

Durch die Fortschritte in der medizinischen Bildgebung fällt in Krankenhäusern und privaten Arztpraxen in zunehmendem Maße medizinisches Bildmaterial an. Im Gegensatz zu früher ist es mittlerweile bei vielen Krankheitsbildern üblich, hochauflösende Aufnahmen einer auffälligen Körperregion zu verordnen.

Eines der Ziele des Forschungsprojektes THESEUS MEDICO ist es, die im klinischen Alltag üblichen Bildarchive (PACS, *picture archiving and communication systems*) derart zu strukturieren, dass sie effizient nach komplexeren Bildeigenschaften als den bisher üblichen Kennungen durchsucht werden können. Als Ergänzung zu einer Stichwort-basierten Suche bietet MEDICO auch eine Auswahl an bildbasierten Abfragemöglichkeiten (CBIR, *content-based image retrieval*).

Im Zuge dieser Dissertation wurden zwei Anwendungsfälle von bildbasierten Anfragen auf Computertomographie (CT) Aufnahmen entwickelt. Die erste Anwendung erstellt ein standardisiertes Koordinatentystem für den menschlichen Körper, welches automatisch einen anatomischen Kontext zu einer beliebigen Aufnahmeregion bestimmen und somit Anfragezeiten drastisch reduzieren kann. Dies wird sowohl durch die Suche nach charakteristischen Punkten als auch durch direkte Bildähnlichkeitssuche ermöglicht. Der zweite Anwendungsfall stellt eine direktere Form des CBIR dar und erlaubt die visuelle Ähnlichkeitssuche in einer Datenbank auffälliger Bildregionen mit manuell definierten Bildausschnitten.

Diese Dissertation stellt eine Reihe neuentwickelter Ähnlichkeitssuchkomponenten vor und sie evaluiert das entwickelte medizinische Bildsuchesystem am Beispiel der beiden vorgestellten Anwendungsfälle.

Short Table Of Contents

1 Preliminaries on Medical Image Queries **1**
1.1 Introduction to Medical Image Retrieval 2
1.2 Summary and Thesis Structure 8

I Feature Transformation and Distance Learning **9**

2 Discriminative Subgraph Mining **13**
2.1 Introduction 13
2.2 Near-Optimal Feature Selection in Frequent Subgraphs 16
2.3 Related Work 30
2.4 Experimental Evaluation 31
2.5 Summary and Outlook 43

3 Similarity Estimation using Bayes Ensembles **45**
3.1 Introduction 45
3.2 L_p-norms and Problem Definition 47
3.3 Ensembles of Bayes Estimates 50
3.4 Optimizing the Feature Space for Bayes Estimates 54
3.5 Related Work 58
3.6 Experimental Evaluation 60
3.7 Summary 64

4 Multi-Instance Distance Measures **67**
4.1 Introduction 67
4.2 Combination of Instance Distances 68
4.3 Instance Weighting Methods 71
4.4 Indexing-based Distance Measures 81
4.5 Experimental Evaluation 85
4.6 Summary 94

II Similarity Search in Medical Image Repositories 95

5 Region of Interest Queries in CT Scans 99
5.1 Introduction . . . 100
5.2 Related Work . . . 104
5.3 Workflow Overview . . . 105
5.4 Interpolation using Matching Points . . . 109
5.5 Slice Localization via Instance-Based Regression . . . 115
5.6 An Online Retrieval Algorithm . . . 125
5.7 Experimental Validation . . . 128
5.8 Summary . . . 141

6 Medical Content-Based Image Retrieval (CBIR) 143
6.1 Introduction . . . 144
6.2 Combining Semantic and Similarity Search . . . 145
6.3 Search Infrastructure . . . 153
6.4 Related Work . . . 156
6.5 Experimental Evaluation . . . 159
6.6 Summary . . . 165

7 Discussion and Outlook 167
7.1 Practical Barriers . . . 167
7.2 List of Scientific Contributions . . . 168
7.3 Summary . . . 170

A List of Abbreviations 171

List of Figures 173

List of Tables 175

List of Algorithms 177

Bibliography 179

Contents

1 Preliminaries on Medical Image Queries **1**
1.1 Introduction to Medical Image Retrieval 2
1.1.1 Other Image Retrieval Applications 2
1.1.2 Query by Medical Image 4
1.1.2.1 Challenges in Medical Image Retrieval 4
1.1.2.2 Query for Image 6
1.1.2.3 Partial Image Retrieval 7
1.2 Summary and Thesis Structure 8

I Feature Transformation and Distance Learning **9**

2 Discriminative Subgraph Mining **13**
2.1 Introduction 13
2.2 Near-Optimal Feature Selection in Frequent Subgraphs 16
2.2.1 Combinatorial Optimization Problem 16
2.2.2 Feature Selection and Submodularity 17
2.2.3 gSpan 18
2.2.4 Definition of CORK 20
2.2.5 Computation of CORK 21
2.2.6 Pruning gSpan's Search Space via CORK 22
2.2.7 CORK for Multi-Class Problems 26
2.2.8 Using Pre-Mined Subgraphs 28
2.3 Related Work 30
2.3.1 Discriminative Subgraph Mining 30
2.3.2 Related Work on Correspondences 31
2.4 Experimental Evaluation 31
2.4.1 Datasets 32
2.4.2 Comparison to Filter Approaches 34
2.4.2.1 Pearson's Correlation 34
2.4.2.2 Delta Criterion 35

2.4.2.3 Information Gain . . . 35
2.4.2.4 Sequential Cover . . . 36
2.4.2.5 Filter Results . . . 36
2.4.3 Effect of Target Sizes . . . 38
2.4.4 Experimental Runtime Analysis . . . 39
2.4.5 Impact of a Tolerance Threshold for Correspondences . . . 40
2.4.6 Comparison to Wrapper Approaches . . . 42
2.5 Summary and Outlook . . . 43

3 Similarity Estimation using Bayes Ensembles 45
3.1 Introduction . . . 45
3.2 L_p-norms and Problem Definition . . . 47
3.3 Ensembles of Bayes Estimates . . . 50
3.3.1 Bayes Estimates and Bayes Ensemble Distance . . . 50
3.3.2 Training Bayes Ensemble Distances . . . 53
3.4 Optimizing the Feature Space for Bayes Estimates . . . 54
3.5 Related Work . . . 58
3.5.1 Metric Distance Learning . . . 58
3.5.2 Non-Metric Distance Learning . . . 60
3.6 Experimental Evaluation . . . 60
3.6.1 Nearest Neighbor Classification . . . 61
3.6.2 Precision and Recall Graphs . . . 61
3.6.3 Examination of the BED Components . . . 62
3.7 Summary . . . 64

4 Multi-Instance Distance Measures 67
4.1 Introduction . . . 67
4.2 Combination of Instance Distances . . . 68
4.2.1 Average Linkage (AvgLink) . . . 68
4.2.2 Minimum Distance (MinDist) . . . 69
4.2.3 Half the Sum of Minimum Distances (HMD) . . . 69
4.2.4 Sum of Minimum Distances (SMD) . . . 69
4.2.5 Hausdorff Distance (HD) . . . 70
4.2.6 Convolution Distance (CD) . . . 70
4.2.7 Maximum Mean Discrepancy (MMD) . . . 70
4.3 Instance Weighting Methods . . . 71
4.3.1 Weighting Distance Measures . . . 71
4.3.1.1 Weighted Average Linkage . . . 71
4.3.1.2 Weighted Convolution Distance . . . 72
4.3.1.3 Weighted HMD / SMD . . . 72
4.3.1.4 Obstacles in Distance Weighting . . . 73

4.3.2 Using Instance Weights for Instance Selection 73
4.3.3 Instance Weight Computation 74
4.3.3.1 Wilcoxon Rank Sum Test 74
4.3.3.2 Pearson Correlation 76
4.3.3.3 Other Weighting Strategies 76
4.3.4 Limits of Supervised Instance Weighting 78
4.3.5 Other Methods for Instance Selection 79
4.3.5.1 k-SMD 79
4.3.5.2 MILES 80
4.3.5.3 Integrated Region Matching (IRM) 80
4.4 Indexing-based Distance Measures 81
4.4.1 Single-object Indexing 81
4.4.2 Instance Indexing 82
4.4.2.1 Average k-Minimum Linkage Classification . . . 82
4.4.2.2 Geometric k-Minimum Linkage Classification . 83
4.4.2.3 Global k-Minimum Distance Classification . . . 83
4.4.3 Alternative Accelerations on Multi-Instance Retrieval . . 84
4.5 Experimental Evaluation 85
4.5.1 Classification Settings 85
4.5.2 Used Datasets 85
4.5.2.1 Musk Datasets 85
4.5.2.2 Conf Datasets 85
4.5.2.3 Caltech Dataset 87
4.5.2.4 Stock4B Data Set 88
4.5.3 Results 88
4.5.3.1 Ranking Quality 88
4.5.3.2 Retrieval Runtime 91
4.6 Summary 94

II Similarity Search in Medical Image Repositories 95

5 Region of Interest Queries in CT Scans 99
5.1 Introduction 100
5.2 Related Work 104
5.3 Workflow Overview 105
5.3.1 Example-based Query Definition 106
5.3.2 Concept-based Query Definition 108
5.3.3 ROI retrieval for a standardized height range 108
5.4 Interpolation using Matching Points 109
5.4.1 Interpolation Functions 109

5.4.2 Height Atlas Definition . . . 111
5.4.2.1 Model by Example (EXMP) . . . 112
5.4.2.2 Manually Aligned Examples (ALIGN) . . . 112
5.4.2.3 Learning a Standardized Height Atlas . . . 112
5.4.2.4 Collecting the Atlas Distribution . . . 114
5.5 Slice Localization via Instance-Based Regression . . . 115
5.5.1 Regression Features . . . 117
5.5.1.1 Spatial Pyramid Kernel . . . 117
5.5.1.2 Preprocessing . . . 119
5.5.1.3 Used Image Patch Features . . . 120
5.5.2 Instance-Based Regression . . . 121
5.5.2.1 Single-Feature Regression . . . 121
5.5.2.2 Multi-Represented Regression . . . 122
5.5.3 Query Acceleration by using a Spatial Index . . . 123
5.5.4 Generating Reliability Weights . . . 124
5.6 An Online Retrieval Algorithm . . . 125
5.6.1 ROI Query Processing . . . 125
5.6.2 Handling Special Cases . . . 127
5.7 Experimental Validation . . . 128
5.7.1 Atlas Accuracy . . . 129
5.7.2 Validation of Regression Queries . . . 131
5.7.2.1 Regression Ground Truth . . . 131
5.7.2.2 Regression Quality . . . 131
5.7.2.3 Speed-up via RCA and Indexing . . . 136
5.7.3 Precision of ROI Queries . . . 137
5.7.4 Runtime of ROI Queries . . . 140
5.8 Summary . . . 141

6 Medical Content-Based Image Retrieval (CBIR) 143
6.1 Introduction . . . 144
6.2 Combining Semantic and Similarity Search . . . 145
6.2.1 Query by Concept . . . 145
6.2.2 Query by Scribble . . . 147
6.2.2.1 Used Image Descriptors . . . 148
6.2.2.2 Image Feature Combination . . . 150
6.2.3 Combined Search . . . 151
6.3 Search Infrastructure . . . 153
6.3.1 The PACS . . . 154
6.3.2 The Semantic Database . . . 155
6.3.3 The Image Feature Database . . . 155
6.4 Related Work . . . 156

6.4.1 Query by Concept . . . 156
6.4.2 Query by Scribble . . . 157
6.4.3 Combined Search . . . 158
6.5 Experimental Evaluation . . . 159
6.5.1 Datasets . . . 159
6.5.2 Visual Similarity Ranking Performance . . . 160
6.5.3 Benefit of the Combined Search . . . 163
6.6 Summary . . . 165

7 Discussion and Outlook **167**
7.1 Practical Barriers . . . 167
7.2 List of Scientific Contributions . . . 168
7.2.1 Improved Data Integration . . . 168
7.2.2 Flexible Paths to Similarity Search . . . 169
7.2.3 Definition and Solution of two CBIR Use Cases . . . 169
7.2.4 Solutions for Efficient Image Retrieval . . . 169
7.3 Summary . . . 170

A List of Abbreviations **171**

List of Figures **173**

List of Tables **175**

List of Algorithms **177**

Bibliography **179**

Chapter 1

Preliminaries on Medical Image Queries

In hospitals and medical centers, the use of modern medical imaging options has increased enormously over the past decades. Any type of medical image is usually stored in a centralized picture archiving and communication system (PACS). Since these PACS are very user-specialized and manufacturer-dependent, they usually only fulfill a minimum amount of standardization [117, 50], and thus, they only permit a limited variety of retrieval queries.

Computed Tomography (CT) is a typical example of an increasingly frequented examination method. A CT scan can be considered to be a stack of 2D photographs taken from within the human body. Depending on the used image resolution and the size of the scanned body range, the disk space for one CT scan ranges between few megabytes and multiple gigabytes. Current PACS usually store these datasets as uncompressed image material, solely structured by a set of acquisition and patient identifiers.

Recently, multiple research groups intensified their efforts for re-organizing conventional PACS such that they can also be queried for more advanced image meta-data or additional manual or automatically-generated annotations linked the image. The common goal of these inquiries is to facilitate keyword-based retrieval queries, like the request for a list of all scans for patients related with a specific viral infection. This is also one of the targets of Theseus MEDICO, a research project sponsored by the German Federal Ministry of Economics and Technology. Besides providing a framework enabling the user-friendly specification of standardized keyword queries, MEDICO also offers a variety of content-based image similarity search applications.

This thesis introduces two major use cases, which employ content-based image retrieval (CBIR) techniques for answering clinical queries on CT scans. The first application provides a standardized body coordinate system using

image-based landmarks and further image information for automatically providing anatomical context information of a given scan region and for greatly accelerating retrieval queries. The second application provides a more direct form of CBIR by enabling the specification of a visual query template, which is used for retrieving similar image patterns.

Both use cases require a search framework allowing both efficient image indexing (both in the sense of image processing and spatial indexing) and context-sensitive query processing. The search framework integrated into the MEDICO system makes use of state of the art similarity concepts and spatial indexing systems. Additionally, it provides a variety of newly-developed or specialized similarity search components, which will be presented in the first part of this thesis. The second part describes the application of these methods in the two medical use cases and provides a thorough evaluation of the generated search framework.

1.1 Introduction to Medical Image Retrieval

Medical image retrieval is a special case of information retrieval. In this thesis, we will focus on similarity aspects of medical image queries. The following sections will hence give a short overview on existing image retrieval applications and the special challenges of medical image retrieval. This chapter will close with a guide to the structure of the thesis.

For the interested reader missing a comprehensive survey on similarity search and information retrieval we refer to [181] and [78] for guides on similarity search and data mining as well as to [110, 23] for introductions to information retrieval.

1.1.1 Other Image Retrieval Applications

Image retrieval is a fast-growing area within the similarity search community. [39] The leading web search engines (google, bing, yahoo and others) all support some form of image search. Query specification is usually text-based and uses meta-information associated to the potential result images. The content of the images is mostly used as a result filter for user-defined attributes like color or size.

Most search engines also offer a query-refinement option which focuses the result set on images which are supposed to be similar to a chosen candidate. A real query-by-example, where the user can actively provide a query template image, however, is mostly offered by specialized commercial vendors of stock

photography.[1] Since the search mechanisms behind those services are predominantly based on color characteristics, they are not applicable in the medical domain, which primarily features intensity-based grey-value images.

Another large field of application for image similarity search is facial recognition. Various implementations are already in use, e.g. for detecting suspects via matching mug shots to surveillance videos. This area of research is highly specialized and it usually involves the computation of biometric markers defined by heavy use of expert knowledge. Certainly, expert information is also required for queries on medical images. However, facial recognition is closer to the field of object tracking than to similarity search. The goal is to reliably recognize a given facial template under any number of lighting conditions or disguise events and not to raise an alert whenever similar-looking persons are detected. Therefore, even though this technique also involves the human body, the philosophy of exact object matching can only be directly applied to the medical imaging domain in special cases.

Other applications for image retrieval are provided by interesting-point detection approaches. They are based on the detection of characteristic points within an image. The interesting points are then described such that they can be matched to similar image regions within other images, or similar regions within the same image. The most prominent examples for interesting-point procedures are SIFT [106] and SURF [10]. They are usually employed in object tracking applications or for image stitching procedures, where multiple images are to be combined to form a panoramic image. This approach also follows an exact-matching scheme, and therefore, its medical use is mostly located in tracking applications like automatic cartographic computations for camera-based internal examinations.

Another area of image retrieval actually involves adult photography. The most conservative reader will agree that there is a market for similarity queries in erotic image material. Today's research, however, mostly focuses on the protection of minors by automatically categorizing potentially offensive pictures, e.g. via skin detection approaches [91, 11] or hand-tracking [43]. Even though there are multiple applications of similarity search in dermatology, [174, 90] they again rely heavily on expert knowledge.

We thus conclude that image retrieval is very application-specific and that it takes a large amount of careful consideration to choose among the algorithmic components to be used.

[1] For examples see `http://www.fotofinder.com` or `http://www.ideeinc.com/`

1.1.2 Query by Medical Image

CBIR can be a valuable tool for radiologists or other physicians in the clinical routine. Finding similar patient cases based on visual correlations facilitates the comparison of a current patient's status with earlier patient histories. Even though automatic suggestions can hardly replace the valuable practise of personally conferring with other medical specialists, computational support systems will be able to recommend the best-suited specialist for a given problem. Additionally, an up-to-date retrieval system can enable the identification of ongoing studies which may be suitable for participation of the current patient.

1.1.2.1 Challenges in Medical Image Retrieval

For a general review on medical content-based image retrieval (CBIR), please refer to Müller and Deserno [114]. They emphasize the challenges posed by the large variety of possible image types (ranging from 1D measurements like electrocardiograms to collections of videos of 3D volume data taken in multiple image modalities). Consequently, a flexible framework for clinical image retrieval must also be able to deal with multiple modalities and support a broad range of query types. A major focus in the preparation of this thesis was therefore the exploration of *various types of object representations*. This thesis will test retrieval options for simple real-valued feature vectors (Chapter 3), multi-instance features (Chapter 4) and graph-based object representations (Chapter 2). The practical realizations of the medical use cases in Part II additionally explore methods of multi-represented objects.

Besides habitual obstacles like rejection by the users [4] and technical problems like varying image quality and characteristics or legal barriers [114, 2], CBIR in medicine is challenged by the *semantic gap*. The semantic gap summarizes the general observation that image properties which can be efficiently represented electronically do not necessarily correspond to an actual medically relevant content. Chapter 5 will therefore introduce an approach for closing the gap between well-structured but self contained medical ontologies and a subgroup of image-based queries.

An additional constraint on any CBIR application is the *high annotation effort*. As image annotations usually require time-costly drawing interactions, the annotations of the same finding by multiple users or even by the same user vary greatly. Very early, this problem inspired semi-automatic annotation approaches as in [14], however, the time costs could not yet be satisfyingly solved. The exact segmentation of a three-dimensional object like the liver within a CT scan can still take an experienced annotator up to 20 minutes. In order to minimize this obstruction on 3D annotations, various CBIR systems only offer regions of interests (ROIs) with a minimum bounding box (abbreviated

as MBR for minimum bounding rectangle) representation. [35, 138]. MBRs are very convenient for visual examination, since any 3D ROI is usually loaded and visually presented in a bounding box. However, capturing a ROI as an MBR will lead to the inclusion of false positive (background) regions compared to a precisely selected region of arbitrary complexity. Chapter 6 will present solutions for resolving the diluted adequacy of MBR representations for 3D queries.

Furthermore, even if there is a suitable collection of annotations to be queried, the actual performance of any retrieval system is very hard to validate. The quality assessment of a query's result set necessitates an additional annotation effort for defining an ideally expected result set. As an additional complication, this ideal result set is going to vary strongly, depending on the medical query context. This thesis therefore explores various ways of training similarity measures with respect to varying ground truth annotations in the Chapters 2, 3 and 5.

In addition to the goal of retrieval accuracy, clinical queries are mostly asked under enormous time-pressure. The daily routine of a radiologist does not allow for long waiting times of a query system as it is already restricted by loading times due to slow intranet connections. *Efficient query processing* is thus an integral part of any medical retrieval system. Chapter 6 exploits database filtering techniques for speeding up similarity queries, while the retrieval approach in Chapter 5 uses spatial index structures. Moreover, one of the main benefits of the retrieval approach of Chapter 5 is the reduction of loading times of 3D images.

During this thesis we also investigated alternative ways of query acceleration: in [53], we examined solutions of all-nearest-neighbor queries and the presentation in [52] summarized effects of spatial index structures on the new storage medium of SSD (solid state disk) flash storage. In order to keep the focus of this thesis as concise as possible in this multi-disciplinary field, these results will not be presented in the following.

Finally, there is the practical problem of *distributed sources of information*. By far not all patient-specific data in a hospital is stored in the PACS. A variety of additional database systems like the Radiology Information System (RIS), or the Laboratory Information System (LIS) may contain relevant information for an image-related query. Together with any number of specialized decision support systems and additional hospital data collections, these form an immense cloud of data sources, which are only partially and incompletely connected by a large variety of standards like HL7 [50] or DICOM (Digital Imaging and Communications in Medicine) [117]. Hence, the success of future holistic medical software strongly depends on the progress of further standardization and the ability to efficiently connect various types of data sources.

One way of connecting inhomogeneous databases is surveyed in Chapter 6 at the example of combining image-based similarity search and a semantic query system.

It is therefore hardly surprising that medical CBIR today is mainly focused on special cases like photographs of skin lesions [6] or the cervix [175], where they can be optimized for their given use case.

Currently, medical PACS only support a small choice of query options. According to the DICOM standard, [117] it is possible to list examinations per patient with additional filter options on standardized meta-information like the date of examination or the picture modality. Depending on the PACS vendor, these result lists show carefully-chosen meta-information, optimally featuring a representative screenshot. As soon as an image has been selected by the user, it usually has to be completely loaded from the server. For 3D images, the system offers a preview, usually in the form of a 2D topogram.

1.1.2.2 Query for Image

A feature which is missing from most PACS are actual image-related queries:

1. *concept-based retrieval queries*, requiring a textual list of requested image properties, or

2. *similarity queries by example*, enabling the user to provide a template image for which to find similar images.

For concept-based queries, it is easy to see why current radiology systems do not support them: they require a duplicate annotation overhead for first outlining the region of interest and then for tagging it with an exhaustive list of descriptive expressions. However, in the wake of the success of the semantic web and an increasing interest in taking standardization efforts, [142] this form of image queries is still the most common in medical information systems research. Semantic queries as proposed in [94, 137] have the advantage of being well-defined and self-contained, i.e. they do not need to actually analyze the raw image material.

It is for this reason that the idea of structured reporting is finally arriving in the medical routine. Medical experts analyzing any clinical examination, usually file a report about their observations and their professional diagnosis. In structured reporting, the examining experts have to obey a specific and standardized report structure for recording their findings. The main obstacle of this new reporting technique is to ensure that the additional information gain by structured reporting outweighs the additional time required for generating the report. [37] In addition to the improved query possibilities, this way of

reporting offers the great potential of visual links between a single report item and its actual position in the described image. A wealth of ontologies for categorizing anatomical properties, [127] visual findings [103] or diseases [171, 86] is available. The Theseus MEDICO Ontology provides links between these ontologies in order to facilitate the annotation and query process. [137]

A real query-by-image is harder to specify. In order to provide a general query system, it must be able to deal with both 3D and 2D queries, potentially even with collections of multiple query regions. It is therefore not sufficient to test various types of image descriptors on their suitability [44], but a generic image query system also needs to support various types of query representation which are to be queried as efficiently as possible. The image query framework of the Theseus MEDICO system therefore integrates various descriptor types in a generic plug-in fashion for multiple image modalities and query scenarios. Additionally, it provides various ways of training application-specific similarity measures and optimizing retrieval times.

1.1.2.3 Partial Image Retrieval

Another retrieval option not offered on the market so far is the efficient retrieval of partial image volumes. As modern radiological scanners produce increasingly large images, the communication times for loading these volumes from the PACS quickly grows to several minutes in a system under heavy communication load. In many cases, it is not necessary to load the complete volume but only a region of interest (ROI) taking only a fraction of the scan, e.g. an excerpt of the body region that the current user is specialized in. This problem can be solved by raster databases, however, the queried coordinates are rarely pre-defined but only available as an anatomic concept or an example ROI. Consequently, a partial volume retrieval system requires a more sophisticated localization estimation procedure.

Anatomical positioning is not a new problem in the field of medical imaging. The research category *image registration* comprises methods for mapping the medical images of various patients or various acquisition times to each other. [108, 82] This may happen by using a template image or by direct intensity-based image alignment. However, most image registration approaches are computationally demanding and runtime extensive. Furthermore, they require the availability of the complete image material, thereby limiting their use in partial image retrieval to a pre-processing step.

The Theseus MEDICO framework contains a retrieval system for CT scans which makes use of a fast, registration-like anatomical mapping approach. The main advantage of this new approach w.r.t. exhaustive registration techniques is that it can be quickly computed on previously un-processed images, and that

it can choose the optimal retrieval policy based on the amount of pre-processed data already available. As a helpful by-product, this retrieval system also provides a general body height atlas, which summarizes the expected distribution of anatomically relevant landmarks and organs. On the one hand, such an atlas offers the possibility to query a 3D volume by anatomical concepts. On the other hand, it can be used as a convenient means of fully-automatic image annotation, by displaying information about the physiological context of a currently visible image excerpt.

1.2 Summary and Thesis Structure

We conclude that similarity search in medical images is a wide field touching a large diversity of scientific research areas. This thesis cannot provide a complete evaluation of all required components, thus it is primarily focused on the choice of the similarity concepts used for image retrieval and its practical application. It is structured as follows:

Part I presents methods on feature transformation and distance learning for various types of object representations. The most complex representation type handled here are graphs: Chapter 2 surveys a method on discriminative subgraph mining, which integrates a feature selection criterion into the subgraph mining process. In contrast, Chapter 3 deals with conventional real-valued feature vectors and ways how to transform them into a more meaningful space and to consequently improve the notion of object similarity. In Chapter 4, we review existing distance measures on multi-instance objects and explore ways of how to improve their precision and efficiency in large retrieval datasets.

The practical part of this thesis, Part II, demonstrates two concrete applications of similarity search in medical image databases, which have been integrated into the Theseus MEDICO prototype. The first application in Chapter 5 presents a flexible image retrieval system which allows the user to load sub-regions of a volume by either verbally specifying an anatomical region of interest or by asking for a similar anatomical scope already opened in a template volume. Chapter 6 gives a general overview on the similarity search components of the MEDICO prototype at the example of a classical query-by-example for three-dimensional sub-volume queries, which can be complemented by adding semantically-defined filtering constraints.

The final chapter summarizes the methods and algorithms proposed in this thesis and concludes with an outlook on future research paths opened by its findings.

Part I

Feature Transformation and Distance Learning

Part I of this thesis presents the advances in algorithmic foundations of similarity search gained in the course of the practical investigations of medical image queries presented in Part II. As the academic field of similarity search is widely spread, this category holds any subject between the contribution of new machine learning constructs and similarity concepts to retrieval-related discoveries like query optimization or database indexing approaches.

In order to keep the scope of this thesis at a manageable degree of multidisciplinarity, its theoretical part is focused on findings about feature transformation and distance learning for various types of data objects, mostly skipping over advances in the field of spatial indexing. A flexible representation of similarity queries is a demanding but worthwhile. Medical images have been represented by real-valued feature vectors [44], sets of feature vectors [59], trees, [105] or graphs [61].

Therefore, the following chapters will cover a number of optimizations on the similarity among three different feature types. Chapter 2 introduces the feature mining and feature selection approach *CORK* [153] in the domain of graphs. The *BED* algorithm [54] presented in Chapter 3 is a flexible similarity learning framework for real-valued feature vectors. Finally, Chapter 4 presents a survey on multi-instance distance measures, which are needed for multiple types of image representations.

Chapter 2

Discriminative Frequent Subgraph Mining with Optimality Guarantees

The goal of frequent subgraph mining is to detect subgraphs that frequently occur in a dataset of graphs. In classification settings, one is often interested in discovering *discriminative* frequent subgraphs, whose presence or absence is indicative of the class membership of a graph.

In this chapter, we survey an approach to feature selection on frequent subgraphs, called *CORK* (correspondence-based quality criterion), that combines two central advantages. First, it optimizes a submodular quality criterion, which means that we can yield a near-optimal solution using greedy feature selection. Second, our submodular quality criterion can be integrated into gSpan, the state-of-the-art tool for frequent subgraph mining, and help to prune the search space for discriminative frequent subgraphs even *during* frequent subgraph mining.

This work was published in [152] after positive feedback on a workshop contribution in [18]. An extended journal version, generalizing the method to multi-class problems and augmenting it by an alternative feature selection pipeline was later published in [153].

2.1 Introduction

In a graph classification problem, we are given a set of n training graphs $\mathcal{G} = \{G^{(1)}, \ldots, G^{(n)}\}$ with class labels $\{G^{(i)}, y^{(i)}\}_{i=1}^{n}$, $y^{(i)} \in \{1, \ldots, K\}$. Given these training examples, our task is to train a classifier for correctly predicting the labels of unclassified test graphs.

Such graph classification algorithms have a wide variety of real world applications. In biology and chemistry, for example, graph classification quantitatively correlates chemical structures with biological and chemical processes, such as active or inactive in an anti-cancer screen, toxic or non-toxic to human beings [100]. This makes graph classification scientifically and commercially valuable (e.g. in drug discovery).

In computer vision, images can be abstracted as graphs, where nodes are spatial entities and edges are their mutual relationships. Such models can be used to identify the type of foreground objects in an image. An example from medical imaging was presented in [61].

In software engineering, a program can also be modeled as a graph, where program blocks are nodes and flows of the program are edges. Static and dynamic analysis of program behaviors can then be carried out in these graphs. For instance, anomaly detection of control flows is essentially a graph classification problem.

Recent research in graph classification comprises three branches:

1. first, the family of *frequent pattern approaches* [95, 45, 30]. Each graph is represented by its frequent subgraphs, i.e. its set of subgraphs that occur in at least $\sigma\,\%$ of all graphs in the database for a manually-specified threshold σ. This frequent pattern approach is also referred to as the (frequent) substructure or fragment approach, and we will use these terms interchangeably.

2. second, the family of approaches that consider *all subgraphs* of a certain type in a graph [92, 159, 139]. For instance, the graph kernels by [92, 139] belong to this class and they count common walks and subtree patterns in two graphs, respectively.

3. third, the family of wrapper approaches that select informative subgraphs for classification during the training phase. Typical instances of this family are the boosting approach by [101] and the lasso-approach by [154].

In this work, we are concerned with the first of these three families, the family of frequent subgraph approaches. There are two reasons for adapting frequent subgraphs in graph classification. First, it is computationally difficult to enumerate all of the substructures existing in a large graph dataset, while it is possible to mine frequent patterns due to the recent development of efficient graph mining algorithms. Second, the discriminative power of extremely infrequent substructures is small due to their limited coverage in the dataset. Therefore, it is a promising approach to use frequent substructures as features in classification models.

However, the vast number of substructures poses three challenges.

1. Redundancy: Most frequent substructures only differ slightly in structure and co-occur in the same graphs.

2. Statistical significance: Frequency alone is not a good measure of the discriminative power of a subgraph, as both frequent and infrequent subgraphs may be uniformly distributed over all classes. Only frequent subgraphs whose presence is statistically significantly correlated with class membership are promising contributors for classification.

3. Efficiency: Very frequent subgraphs are not useful for classification due to lack of discriminative power. Therefore, frequent subgraph based classification usually sets an extremely low frequency threshold, resulting in thousands or even millions of features. Given such a tremendous number of features, any runtime or memory-intensive feature selection algorithm will fail.

Consequently, we need an efficient algorithm to select discriminative features among a large number of frequent subgraphs. In [152], we introduced a near-optimal approach to feature selection among frequent subgraphs generated by gSpan [177] for two-class problems. Our method greedily chooses frequent subgraphs according to the *submodular* quality criterion CORK (Correspondence-based Quality Criterion). The use of a submodular function in a greedy approach ensures a solution close to the optimal solution [118]. We furthermore showed that CORK can be integrated into gSpan, the state-of-the-art tool for frequent subgraph mining.

Other approaches use heuristic strategies for feature selection (such as [30, 57]) or do not provide optimality guarantees [101, 132, 131, 154, 176, 89]. We will present an overview on related algorithms in Section 2.3.1.

This chapter will present the idea of near-optimal feature selection in subgraph patterns and introduce improvements for future use. We will first formalize the optimization problem to be solved (Section 2.2.1) and then, we will summarize the essential ingredients of our graph feature selector: first, submodularity and its use in feature selection (Section 2.2.2); second, gSpan, the method to find frequent subgraphs (Section 2.2.3). We will review our selection criterion CORK for two-class problems in Section 2.2.4, detail its computation in Section 2.2.5 and explain its integration as additional pruning criterion into pattern growth based graph miners like gSpan in Section 2.2.6.

Many applications for graph learning actually define more than the commonly-used two classes: Biological molecules can be categorized into a wide catalog of functional or structural classes, social network communities are involved with various topics and process flows can be analyzed with respect to multiple attributes. We will thus generalize CORK to multi-class problems in Section 2.2.7.

Finally, for increasing the flexibility of our algorithm, in Section 2.2.8, we will also provide an extension for using the proposed pruning approach on pre-mined graphs. After a review of related work in Section 2.3 we thoroughly evaluate the proposed algorithms in Section 2.4 on 11 real-world datasets and conclude with a discussion and outlook in Section 2.5.

2.2 Near-Optimal Feature Selection in Frequent Subgraphs

We formalize the given dataset as a collection of graphs $\mathcal{G} = \cup_{i=i}^{K} \mathbf{K}_i$ that each belong to one of the K classes $\mathbf{K}_i$. In this thesis we exclude overlapping classes, however, the proposed selection approach can be easily extended to graphs with multiple labels.

As a notational convention, the *vertex set* of a graph $G \in \mathcal{G}$ is denoted by $V(G)$ and the *edge set* by $E(G)$. A label function, l, maps a vertex or an edge to a label. A graph G is a subgraph of another graph G' if there exists a subgraph isomorphism from G to G', denoted by $G \sqsubseteq G'$. Accordingly, G' is called a super-graph of G ($G' \sqsupseteq G$). Due to its importance for this chapter, we here recite the definition of a subgraph isomorphism.

Definition 1 (Subgraph Isomorphism) *A subgraph isomorphism is an injective function $f : V(G) \rightarrow V(G')$, such that*

1. $\forall u \in V(G)$, $l(u) = l'(f(u))$, *and*
2. $\forall (u, v) \in E(G)$, $(f(u), f(v)) \in E(G')$ *and* $l(u, v) = l'(f(u), f(v))$,

where l and l' are the label function of G and G', respectively. f is called an embedding of G in G'.

Given a graph database $\mathcal{G}$, we denote by $\mathcal{G}_{G_1}$ the number of graphs in $\mathcal{G}$ of which G is a subgraph and by $\mathcal{G}_{G_0}$ the number of graphs in $\mathcal{G}$ of which G is *not* a subgraph. $\mathcal{G}_{G_1}$ is called the *(absolute) support*. A graph G is *frequent* if its support is no less than a minimum support threshold, σ. Hence, the frequent graph is a relative concept: whether or not a graph is frequent depends on the value of σ and on the number of elements $|\mathcal{G}|$ contained in $\mathcal{G}$.

2.2.1 Combinatorial Optimization Problem

Feature selection among frequent subgraphs can be defined as a combinatorial optimization problem. We denote by $\mathcal{D}$ the full set of features, which in our

case will correspond to the frequent subgraphs generated by gSpan. When using these features to predict the class membership of individual graph instances, clearly, only a subset $\mathcal{E} \subseteq \mathcal{D}$ of features will be relevant. We denote the relevance of a feature set for class membership by $q(\mathcal{E})$, where q is a quality criterion measuring the discriminative power of $\mathcal{E}$. It is computed by restricting the dataset's representation to the features in $\mathcal{E}$. We then formulate feature selection as:

$$\mathcal{D}^{\ddagger} = \arg\max_{\mathcal{E} \subseteq \mathcal{D}} \; q(\mathcal{E}) \qquad \text{s.t.} \quad |\,\mathcal{E}\,| \leq s \tag{2.1}$$

where $|\cdot|$ computes the cardinality of a set and s is the maximally allowed number of selected features.

The optimal solution of this problem would require us to search all possible subsets of features exhaustively. Due to the exponential number of all feature combinations this approach is prohibitive for large feature sets like frequent subgraphs. The common remedy is to resort to heuristic alternatives, the solutions of which cannot be guaranteed to be globally optimal or even close to the global optimal solution. Hence, the key point in this chapter is to employ a heuristic approach which *does* allow for these quality guarantees, namely a greedy strategy which achieves *near-optimal* results.

2.2.2 Feature Selection and Submodularity

Assume that we are measuring the discriminative power $q(\mathcal{E})$ of a feature set $\mathcal{E}$ in terms of a quality function q. A near-optimality solution is reached for a *submodular* quality function q when used in combination with greedy feature selection. Greedy forward feature selection consists in iteratively picking the feature that – in union with the features selected so far – maximises the quality function q over the prospective feature set. In general, this strategy will not yield an optimal solution, but it can be shown to yield a near-optimal solution if q is submodular:

Definition 2 (Submodular set function) *A quality function q is said to be* submodular *on a set $\mathcal{D}$ if for $\mathcal{E}' \subseteq \mathcal{E} \subseteq \mathcal{D}$ and $X \in \mathcal{D}$:*

$$q(\mathcal{E}' \cup \{X\}) - q(\mathcal{E}') \geq q(\mathcal{E} \cup \{X\}) - q(\mathcal{E}) \tag{2.2}$$

Thus, the quality increase for incorporating X into a set $\mathcal{E}'$ is higher than (or equal to) the increase that can be reached by adding X to any superset $\mathcal{E}$ of $\mathcal{E}'$.

If q is submodular and we employ greedy forward feature selection, then we can exploit the following theorem from [118]:

Theorem 3 *If q is a submodular, non-decreasing set function on a set $\mathcal{D}$ and $q(\emptyset) = 0$, then greedy forward feature selection is guaranteed to find a set of features $\mathcal{E}^\dagger \subseteq \mathcal{D}$ such that*

$$q(\mathcal{E}^\dagger) \geq \left(1 - \frac{1}{e}\right) \max_{\mathcal{E} \subseteq \mathcal{D}:\ |\mathcal{E}|=s} q(\mathcal{E}) \ , \tag{2.3}$$

where s is the number of features to be selected.

As a direct consequence, the result from greedy feature selection achieves at least $\left(1 - \frac{1}{e}\right) \approx 63\%$ of the score of the optimal solution to the feature selection problem. This property is referred to as being *near-optimal* in the literature (e.g. [73]).

2.2.3 gSpan

If we found a useful submodular criterion for feature selection on frequent subgraphs, we could yield a near-optimal solution to problem (2.1). But how do we determine the frequent subgraphs in the first place? For this purpose, we use the frequent subgraph algorithm gSpan [177], which we will outline in the following.

The discovery of frequent graphs usually consists of two steps. In the first step, we generate frequent subgraph candidates, while in the second step, we check the frequency of each candidate. The second step involves a subgraph isomorphism test, which is NP-complete. Fortunately, efficient isomorphism testing algorithms have been developed, making such testing affordable in practice. Most studies of frequent subgraph discovery pay attention to the first step; that is, how to generate as few frequent subgraph candidates as possible, and as fast as possible.

The initial frequent graph mining algorithms, such as AGM [85], FSG [102] and the path-join algorithm [156], share similar characteristics with the Apriori-based itemset mining [1]. All of them require a join operation to merge two (or more) frequent substructures into one larger substructure candidate. To avoid this overhead, non-Apriori-based algorithms such as gSpan [177], MoFa [16], FFSM [84], and Gaston [121] adopt the pattern-growth methodology, which attempts to generate candidate graphs from a *single* graph. For each discovered graph G, these methods recursively add new edges until all the frequent supergraphs of G have been discovered. The recursion stops once no more frequent graph can be generated.

gSpan introduced a sophisticated extension method, which is built on a depth first search (DFS) tree. Given a graph G we label the root, i.e. the starting vertex of the DFS tree, as v_0, and the last visited vertex as v_n. v_n is

also called the *rightmost vertex*. Consequently, the straight path from v_0 to v_n is the *rightmost path*. Figure 2.1 shows an example. The darkened edges form a DFS tree. The vertices are discovered in the order v_0, v_1, v_2, v_3, thus v_3 is the rightmost vertex. The rightmost path is (v_0, v_1, v_3).

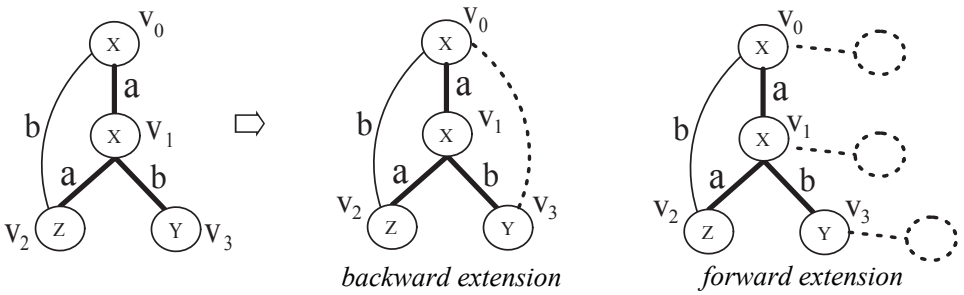


Figure 2.1: gSpan: Rightmost Extension

This method, called rightmost extension, restricts the extension of new edges in a graph as follows: For a given graph and a DFS tree, a new edge e can be added between the rightmost vertex and other vertices on the rightmost path (*backward extension*), or it can introduce a new vertex originating from a vertex on the rightmost path (*forward extension*). As we do not allow duplicate connections, the only legal backward extension candidate of the graph in Figure 2.1 is (v_3, v_0). The forward extension candidates can be edges from v_3, v_1, or v_0 introducing a new vertex. Since there may be multiple DFS trees for one graph, gSpan establishes a set of rules to select one of them as representative so that the backward and forward extensions will only take place in one DFS tree. One of those rules is the restriction of newly generated edges to the vertices along the rightmost path. Another rule, the minimality test, checks whether the currently examined graph has not been treated before. For a detailed description of gSpan, see [177].

Algorithm 1 outlines the pseudocode of gSpan. $G \diamond_r e$ denotes that an edge e extends graph G via rightmost extension. Step 2 is the minimality test, where $\text{DFS}(G)$, the canonical form of graph G [177] is compared to the edge order of G. Therefore, G is only proceeded at the first encounter. The embedding of the candidate set of rightmost extended supergraphs $G \diamond_r e$ and the maintenance of their support $|\{G' \in \mathcal{G} \,|G \diamond_r e \sqsubseteq G'\}|$ is generated in steps 7 and 8. Only frequent candidates are then tested on their validity as actual frequent subgraphs and may be themselves rightmost extended in a further call of gSpan (step 10).

Once we have determined the frequent subgraphs using gSpan, a natural way of representing each graph G is in terms of a binary indicator vector of length $|\mathcal{S}|$:

Algorithm 1 gSpan

Input: Graph G, graph dataset $\mathcal{G}$, threshold σ, set of subgraphs $\mathcal{S}$

```
1: function gSpan(G, 𝒢, σ, 𝒮)
2:     if G ≠ dfs(G) then
3:         return 𝒮                                  ▷ G is not minimal
4:     end if
5:     insert G into 𝒮
6:     C ← ∅                                         ▷ initialize candidate map
7:     scan 𝒢 for all edges e such that G can be rightmost extended to G ⋄_r e
8:     insert all G ⋄_r e into C and record their support
9:     for all G ⋄_r e ∈ C : |{G' ∈ 𝒢 | G ⋄_r e ⊑ G'}| ≥ σ do
10:        call gSpan(G ⋄_r e, 𝒢, σ, 𝒮)      ▷ crawl frequent supergraphs of G
11:    end for
12:    return 𝒮
13: end function
```

Output: The set of frequent subgraphs $\mathcal{S}$.

Definition 4 (Indicator vector) *Given a graph G_i from a dataset $\mathcal{G}$ and a set of frequent subgraph features $\mathcal{S}$ discovered by gSpan. We then define an indicator vector $v^{(i)}$ for G_i as*

$$v_d^{(i)} = \begin{cases} 1 & \text{if } \mathcal{S}_d \sqsubseteq G_i \quad (\mathcal{S}_d \text{ is a subgraph of } G_i) \\ 0 & \text{otherwise} \end{cases}, \tag{2.4}$$

where $v_d^{(i)}$ is the d-th component of $v^{(i)}$ and $\mathcal{S}_d$ is the d-th graph in $\mathcal{S}$.

2.2.4 Definition of CORK

We now define our feature selection criterion q for two-class problems. It will be generalized to multi-class problems in Section 2.2.7.

Definition 5 *Let $\mathcal{G}$ be a dataset of binary vectors, consisting of two disjunct classes $\mathcal{G} = \mathcal{A} \cup \mathcal{B}$. Let $\mathcal{D}$ denote a set of features of the data objects in $\mathcal{G}$, represented by indicator vector $v^{(i)}$ for graphs $G_i \in \mathcal{G}$.*

As we aim to separate the two classes, we pay specific attention to pairs of inter-class instances with the same pattern in the given feature set. These instance pairs are *correspondences*:

Definition 6 (Correspondence) *A pair of data objects $(v^{(i)}, v^{(j)})$ is called a* correspondence *in a set of features indicated by the feature indices $\mathcal{U} \subseteq$*

$\{1, \ldots, |\mathcal{D}|\}$ *(or, w.r.t. a set of features* $\mathcal{U}$*) iff*

$$(v^{(i)} \in \mathcal{A}) \wedge (v^{(j)} \in \mathcal{B}) \wedge \forall d \in \mathcal{U} : (v_d^{(i)} = v_d^{(j)}), \tag{2.5}$$

where $v_d^{(i)}$ *is the value of feature* d *in vector* $v^{(i)}$*.*

Our quality criterion consequently punishes the number of correspondences remaining for feature set $\mathcal{D}$.

Definition 7 (CORK) *We define a quality criterion* q*, called* CORK (Correspondence-based Quality Criterion)*, for a subset of features* $\mathcal{E}$ *as*

$$q(\mathcal{E}) = (-1) * \textit{number of correspondences in } \mathcal{E} \tag{2.6}$$

Theorem 8 *q is submodular.*

Proof For q to be submodular, adding feature $X \in \mathcal{D}$ to a feature set $\mathcal{E}' \subseteq \mathcal{E} \subseteq \mathcal{D}$ has to increase $q(\mathcal{E}')$ at least as much as adding feature X to $\mathcal{E}$ increases $q(\mathcal{E})$. This law of diminishing returns is obviously fulfilled if removing a correspondence from $\mathcal{E}$ by adding feature X also results in a correspondence being eliminated in $\mathcal{E}'$ by adding feature X.

Let us first state that an instance pair $(v^{(i)}, v^{(j)})$, that is a correspondence in $\mathcal{E}$ must also be a correspondence in $\mathcal{E}'$. Note that the opposite is not necessarily true.

In the following, let x be the index of feature X in $\mathcal{D}$. Whenever adding a feature X to $\mathcal{E}$ removes the above correspondence from $\mathcal{E}$, this means that $v_x^{(i)} \neq v_x^{(j)}$, since the other features in $\mathcal{E}$ must match. Therefore, the two formerly corresponding feature patterns for $(v^{(i)}, v^{(j)})$ cannot match in $\mathcal{E}' \cup \{X\}$ either. Thus, if a feature X eliminates a correspondence from $\mathcal{E}$, this very correspondence (possibly together with further correspondences) is also removed from $\mathcal{E}'$, and we satisfy the submodularity condition of Equation 2.2. □

This submodular criterion can be turned (by adding the constant $|\mathcal{A}| \cdot |\mathcal{B}|$) into a submodular set function fulfilling the conditions of Theorem 3.

2.2.5 Computation of CORK

The CORK value for one feature X in a dataset of the classes $\mathcal{A}$ and $\mathcal{B}$ can be computed as the number of inter-class pairs of objects that both contain X (with $\mathcal{A}_{X_1}$ instances in $\mathcal{A}$ and $\mathcal{B}_{X_1}$ instances in $\mathcal{B}$) or that both do not contain X ($\mathcal{A}_{X_0}$ and $\mathcal{B}_{X_0}$ objects).

$$q(\{X\}) = -\left(\mathcal{A}_{X_0} \cdot \mathcal{B}_{X_0} + \mathcal{A}_{X_1} \cdot \mathcal{B}_{X_1}\right) \tag{2.7}$$

For feature sets CORK can be efficiently computed by recursively dividing the dataset into equivalence classes:

Definition 9 (Equivalence Classes) *Given a two-class dataset $\mathcal{G} = \mathcal{A} \cup \mathcal{B}$ represented as binary indicator vectors over the feature set $\mathcal{U}$. Let $\mathcal{P} \subseteq 2^{\mathcal{U}}$ be the set of all unique binary indicator vectors occurring in $\mathcal{G}$ with $|\mathcal{P}| = l$. Then the* equivalence class *of an indicator vector $v^{(i)} \in \mathcal{G}$ is defined as the set*

$$\{v^{(j)} | v^{(j)} \in \mathcal{G} \ \wedge \forall \ d \in \mathcal{U} : \ v_d^{(i)} = v_d^{(j)}\} \tag{2.8}$$

Each of these unique indicator vectors $\mathcal{P}_c$ forms an equivalence class $\mathbf{E}_c (c \in \{1, ..., l\})$ containing all graphs of with an indicator vector equal to $\mathcal{P}_c$.

We denote by

$$\mathcal{A}_{\mathcal{P}_c} = \left| \{v^{(i)} \in \mathcal{A} \mid \forall d \in \mathcal{U} : \ v_d^{(i)} = \mathcal{P}_c[d]\} \right| \tag{2.9}$$

the number of instances of equivalence class $\mathbf{E}_c$ in $\mathcal{A}$ and by

$$\mathcal{B}_{\mathcal{P}_c} = \left| \{v^{(i)} \in \mathcal{B} \mid \forall d \in \mathcal{U} : \ v_d^{(i)} = \mathcal{P}_c[d]\} \right| \tag{2.10}$$

the number of instances of equivalence class $\mathbf{E}_c$ in $\mathcal{B}$.

In each greedy iteration step, those equivalence classes can be efficiently split into hits and misses. The CORK score for a feature set $\mathcal{U} \subseteq \{1, \ldots, |\mathcal{D}|\}$ can thus be calculated by adding up the correspondences of all occurring equivalence classes $\mathbf{E}_c$ in $\mathcal{U}$:

$$q(\mathcal{U}) = (-1) \cdot \left(\sum_{\mathcal{P}_c \in \mathcal{P}} \mathcal{A}_{\mathcal{P}_c} \cdot \mathcal{B}_{\mathcal{P}_c} \right) \tag{2.11}$$

We can now use q for greedy forward feature selection on a pre-mined set $\mathcal{S}$ of frequent subgraphs in $\mathcal{G}$ and receive a result set $\mathcal{S}^\dagger \subseteq \mathcal{S}$ of discriminative subgraphs with a guaranteed quality bound. However, the success of $\mathcal{S}^\dagger$ strongly depends on the choice of the minimum support σ. If σ is chosen too low, we can quickly generate too many features for the selection step to finish in a reasonable runtime. Setting σ too high can cause the loss of all informative features. In the following, we will introduce a selection approach which directly mines only discriminative subgraphs, which is *nested in gSpan* and which can act independently from a frequency threshold.

2.2.6 Pruning gSpan's Search Space via CORK

gSpan exploits the fact that the frequency of a subgraph $S \in \mathcal{S}$ is an upper bound for the frequency of all of its supergraphs $T \sqsupseteq S$ (all subgraphs containing S) when pruning the search space for frequent subgraphs. We will show

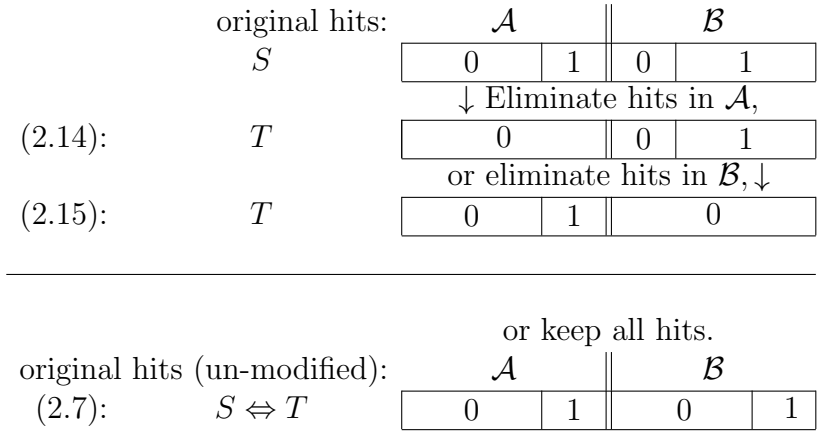


Figure 2.2: Possible change scenarios for the number of hits of supergraphs T for given hit distributions of $S \sqsubseteq T$: Hits ("1") can change into misses ("0"). The resulting extreme cases are illustrated for eliminating all hits from $\mathcal{A}$ (2.14) or from $\mathcal{B}$ (2.15), or for the case where keeping all hits is the best choice as in (2.7)

how to derive an upper bound for the CORK-values of all supergraphs of a subgraph S, which allows us to further prune the search space.

Let us emphasize that this technique can also be applied in other graph miners which employ a kind of hierarchical subgraph pattern growth [16, 84, 121] or Apriori-based join [85, 102, 84]. The only necessary pre-condition for including CORK as pruning step is a supergraph relation ($T \sqsupseteq S$) for patterns mined at a later stage.

Theorem 10 *Let $S, T \in \mathcal{S}$ be frequent subgraphs, and T be a supergraph of S. Let $\mathcal{A}_{S_1}$ denote the number of graphs in class $\mathcal{A}$ that contain S ('hits'), $\mathcal{A}_{S_0}$ the number of graphs in $\mathcal{A}$ that do not contain S ('misses') and define $\mathcal{B}_{S_0}$, $\mathcal{B}_{S_1}$ analogously. Then*

$$q(\{T\}) \leq q(\{S\}) + \max \left\{ \begin{array}{c} \mathcal{A}_{S_1} \cdot (\mathcal{B}_{S_1} - \mathcal{B}_{S_0}) \\ (\mathcal{A}_{S_1} - \mathcal{A}_{S_0}) \cdot \mathcal{B}_{S_1} \\ 0 \end{array} \right\} \tag{2.12}$$

Proof We note that the gSpan pruning criterion is also valid for each class:

$$\mathcal{A}_{S_1} \geq \mathcal{A}_{T_1} \wedge \mathcal{B}_{S_1} \geq \mathcal{B}_{T_1} \ . \tag{2.13}$$

If we want to asses how many correspondences may be eliminated by T, we can take into account, that T can never create new hits but can only decrement the number of hits in both classes. Naturally, the best improvement for S is

made, when T eliminates all hits in one of the two classes and maintains the hits in the other class. This is illustrated in the first two cases of Figure 2.2. When all hits of T disappear from $\mathcal{A}$, $\mathcal{A}_{S_0}$ increases by $\mathcal{A}_{S_1}$ and thus:

$$q(\{T\}) = -\left(\overbrace{(\mathcal{A}_{S_0} + \mathcal{A}_{S_1})}^{|\mathcal{A}|} \cdot \mathcal{B}_{S_0} + 0 \cdot \mathcal{B}_{S_1}\right) = -|\mathcal{A}| \cdot \mathcal{B}_{S_0} \tag{2.14}$$

The same holds for the elimination of all hits from $\mathcal{B}$:

$$q(\{T\}) = -\left(\mathcal{A}_{S_0} \cdot \overbrace{(\mathcal{B}_{S_0} + \mathcal{B}_{S_1})}^{|\mathcal{B}|} + \mathcal{A}_{S_1} \cdot 0\right) = -\mathcal{A}_{S_0} \cdot |\mathcal{B}| \tag{2.15}$$

Finally, we observe a third scenario when T does not cause any change at all, i.e. $q(\{T\}) = q(\{S\})$. This provides an additional bound if the decrease of hits in any class results in more correspondences than for S alone (cf. the last case in Figure 2.2). Our maximal CORK value of T is thus

$$q(\{T\}) \leq \max\left\{\begin{array}{c} -|\mathcal{A}| \cdot \mathcal{B}_{S_0} \\ -\mathcal{A}_{S_0} \cdot |\mathcal{B}| \\ q(\{S\}) \end{array}\right\} \overset{\text{eq. 2.7}}{=} q(\{S\}) + \max\left\{\begin{array}{c} \mathcal{A}_{S_1} \cdot (\mathcal{B}_{S_1} - \mathcal{B}_{S_0}) \\ (\mathcal{A}_{S_1} - \mathcal{A}_{S_0}) \cdot \mathcal{B}_{S_1} \\ 0 \end{array}\right\} \tag{2.16}$$

□

We can now use inequality (2.12) to provide an upper bound for the CORK values of supergraphs T of a given subgraph S and exploit this information for pruning the search space in a branch-and-bound fashion.

Inequality (2.12) can be directly applied in the first iteration of greedy selection. For later iterations of greedy selection, we derive a similar bound on a set of features.

The bound of Equation 2.12 then extends to:

$$q(\mathcal{U} \cup \{T\}) \leq q(\mathcal{U} \cup \{S\}) + \sum_{\mathcal{P}_c \in \mathcal{P}} \max\left\{\begin{array}{c} \mathcal{A}_{\mathcal{P}_c \cup \{S_1\}} \cdot \left(\mathcal{B}_{\mathcal{P}_c \cup \{S_1\}} - \mathcal{B}_{\mathcal{P}_c \cup \{S_0\}}\right) \\ \left(\mathcal{A}_{\mathcal{P}_c \cup \{S_1\}} - \mathcal{A}_{\mathcal{P}_c \cup \{S_0\}}\right) \cdot \mathcal{B}_{\mathcal{P}_c \cup \{S_1\}} \\ 0 \end{array}\right\} \tag{2.17}$$

The main difference to (2.12) is that in later iterations of greedy selection, we only have to consider those graphs which are part of a correspondence (rather than all graphs). We can thus define an additional pruning bound for subgraph enumeration:

Definition 11 (CORK Upper Bound) *Given a subgraph set $\mathcal{U}$ and a subgraph S. The CORK value of any supergraph T of S ($T \sqsupseteq S$) cannot exceed the bound $MAX_{CORK}(\mathcal{U}, S)$:*

$$MAX_{CORK}(\mathcal{U}, S) = q(\mathcal{U} \cup \{S\}) + \sum_{\mathcal{P}_c \in \mathcal{P}} \max \left\{ \begin{array}{c} \mathcal{A}_{\mathcal{P}_c \cup \{S_1\}} \cdot \left(\mathcal{B}_{\mathcal{P}_c \cup \{S_1\}} - \mathcal{B}_{\mathcal{P}_c \cup \{S_0\}}\right) \\ \left(\mathcal{A}_{\mathcal{P}_c \cup \{S_1\}} - \mathcal{A}_{\mathcal{P}_c \cup \{S_0\}}\right) \cdot \mathcal{B}_{\mathcal{P}_c \cup \{S_1\}} \\ 0 \end{array} \right\} . \tag{2.18}$$

Algorithm 2 gSpan$_{\text{CORK}}$

Input: Graph set $\mathcal{G}$, optional threshold σ.

1: **function** gSPAN$_{\text{CORK}}$($\mathcal{G}$, $\sigma = 0$)
2: $\quad\mathcal{S}^\dagger = \emptyset$
3: $\quad S$ = best subgraph for $q(\mathcal{S}^\dagger \cup \{S\})$ ▷ gSpan call
4: $\quad$**if** $q(\mathcal{S}^\dagger \cup \{S\}) > q(\mathcal{S}^\dagger)$ **then**
5: $\quad\quad\mathcal{S}^\dagger = \mathcal{S}^\dagger \cup \{S\}$ ▷ S is an improvement
6: $\quad\quad$**goto** 3
7: $\quad$**end if**
8: $\quad$**return** $\mathcal{S}^\dagger$
9: **end function**

Output: Set of discriminative (frequent) subgraphs $\mathcal{S}^\dagger$.

The new feature mining process is defined in Algorithm 2 as greedy forward selection approach:[1] We initialize the set of selected subgraphs as an empty set $\mathcal{S}^\dagger$ and follow a recursive operation. In step 2, we require the next best subgraph S with $q(\mathcal{S}^\dagger \cup \{S\}) = \max_{T \in \mathcal{S}} q(\mathcal{S}^\dagger \cup \{T\})$. It can be obtained by running gSpan, always maintaining the currently best subgraph S according to q. Whenever in the course of mining, we reach a subgraph T with $\text{MAX}_{\text{CORK}}(\mathcal{S}^\dagger, T) \leq q(\mathcal{S}^\dagger \cup \{S\})$, we can prune all branches originating from T. Else, the candidate subgraph S might still be replaced by any of T's children. As long as the resulting subgraph S actually improves $q(\mathcal{S}^\dagger)$, it is accepted as a discriminative feature and we start looking for the next best subgraph.

In contrast to the definition in Equation 2.1, this setting does not require a selection threshold s for the maximal number of features (subgraphs) since it automatically terminates when no new discriminative subgraph is found. In our experiments, we further noticed that on most datasets, CORK provides such a strong bound that it is even possible to omit the support threshold σ

[1] An implementation of gSPAN$_{\text{CORK}}$ is available at `http://www.dbs.ifi.lmu.de/~thoma/pub/sam2010/sam2010.zip`.

and still receive a discriminative set of (not necessarily frequent) subgraphs within a reasonable amount of time.

2.2.7 CORK for Multi-Class Problems

So far, we have restricted our attention to settings with two classes. Now we will show how to extend gSpan$_{\text{CORK}}$ to multi-class problems. The key challenges here are to extend CORK's definition for handling multiple classes, and to then prove that this multi-class CORK (mcCORK) is still submodular and that it can still be integrated into gSpan.

Definition 12 (pairwise CORK) *Assume we are given a graph dataset* $\mathcal{G} := \cup_{i=1}^{K} \mathbf{K}_i$ *with* K *disjunct classes.* $q_{i,j}(\mathcal{U})$ *shall denote the CORK value restricting the dataset to classes* $\mathbf{K}_i$ *and* $\mathbf{K}_j$ *for a feature set* $\mathcal{U}$*. Then* pairwise *multi-class CORK* ($mcCORK_{pw}$) *is defined as*

$$mcCORK_{pw}(\mathcal{U}) := \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} q_{i,j}(\mathcal{U}) = (-1) \cdot \sum_{\mathcal{P}_c \in \mathcal{P}} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \mathbf{K}_{i,\mathcal{P}_c} \cdot \mathbf{K}_{j,\mathcal{P}_c} \ , \tag{2.19}$$

i.e. as the sum over CORK values for all pairs of classes, where $\mathbf{K}_{i,\mathcal{P}_c}$ *is the number of matches of pattern* $\mathcal{P}_c$ *for* $\mathcal{U}$ *in class* i *and* $\mathbf{K}_{j,\mathcal{P}_c}$ *is the number of* $\mathcal{P}_c$*'s matches in class* j*, respectively.*

Note that we restrict our definition to non-overlapping class labels. Of course, if a graph G belongs to multiple classes, $q_{i,j}(\mathcal{U})$ can be modified such that G is not considered when calculating the overall occurrences per equivalence class. This can be achieved using an additional counter for each equivalence class which is raised whenever a hit also belongs to another class and which is later subtracted from the equivalence class count. However, as structured output is not the focus of this work, we will pause this line of thought for now.

Since pairwise CORK requires a quadratic runtime in the number of classes, we now show the ranking equivalence of pairwise CORK with the linear variant *1-vs.-rest* CORK.

Definition 13 (1-vs.-rest CORK) *Assume we are given a graph dataset* $\mathcal{G} := \cup_{i=1}^{K} \mathbf{K}_i$ *with* K *disjunct classes.* $q_i(\mathcal{U})$ *shall denote the CORK value for a dataset consisting of class* $\mathbf{K}_i$ *and its complement* ($\mathbf{K}_{\neg i} = \cup_{j=1, j\neq i}^{K} \mathbf{K}_j$) *as artificial second class for a feature set* $\mathcal{U}$*. Then* 1-vs.-rest *multi-class CORK*

($mcCORK_{1vr}$) is defined as

$$mcCORK_{1vr}(\mathcal{U}) := \sum_{i=1}^{K} q_i(\mathcal{U}) = (-1) \cdot \sum_{\mathcal{P}_c \in \mathcal{P}} \sum_{i=1}^{K} \mathbf{K}_{i,\mathcal{P}_c} \cdot \mathbf{K}_{\neg i,\mathcal{P}_c} \quad . \tag{2.20}$$

Lemma 14 *1-vs.-rest CORK and pairwise CORK result in the same ranking of feature sets.*

Proof As the classes i to K are disjunct and since CORK does not use relative hit frequencies, the pairwise approach can be reduced to 1-vs.-rest as follows:

$$\begin{aligned}
\mathrm{mcCORK}_{\mathrm{1vr}}(\mathcal{U}) &= (-1) \cdot \sum_{\mathcal{P}_c \in \mathcal{P}} \sum_{i=1}^{K} \mathbf{K}_{i,\mathcal{P}_c} \cdot \mathbf{K}_{\neg i,\mathcal{P}_c} \\
&= (-1) \cdot \sum_{\mathcal{P}_c \in \mathcal{P}} \sum_{i=1}^{K} \left(\mathbf{K}_{i,\mathcal{P}_c} \cdot \left(-\mathbf{K}_{i,\mathcal{P}_c} + \sum_{j=1}^{K} \mathbf{K}_{j,\mathcal{P}_c} \right) \right) \\
&= (-1) \cdot \sum_{\mathcal{P}_c \in \mathcal{P}} \left(\sum_{i=1}^{K} \sum_{j=1}^{K} \mathbf{K}_{i,\mathcal{P}_c} \cdot \mathbf{K}_{j,\mathcal{P}_c} \; - \; \sum_{i=1}^{K} \mathbf{K}_{i,\mathcal{P}_c}^2 \right) \\
&= (-1) \cdot \sum_{\mathcal{P}_c \in \mathcal{P}} \left(2 \cdot \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \mathbf{K}_{i,\mathcal{P}_c} \cdot \mathbf{K}_{j,\mathcal{P}_c} \right) \\
&= 2 \cdot \mathrm{mcCORK}_{\mathrm{pw}}(\mathcal{U})
\end{aligned}$$

□

We next show the submodularity of this multi-class extension of CORK.

Theorem 15 *mcCORK is submodular.*

Proof Both pairwise and 1-vs.-rest mcCORK are sums of pairwise CORK values. As pairwise CORK was shown to be submodular in Theorem 8, mcCORK is a sum of submodular functions. As submodular functions are closed under addition, mcCORK is also submodular. □

For the standard application of CORK-based greedy feature selection, we can hence replace two-class CORK by multi-class CORK, and perform multi-class feature selection with the same optimality guarantees. The question that remains to be answered is whether we can still perform nested feature selection with CORK in multi-class settings, that is whether we can integrate multi-class CORK into gSpan. For this purpose, we require a bound akin to equation (2.18). Since this bound is computed for all encountered frequent subgraphs, we define the bound for the faster 1-vs.-rest mcCORK variant.

Theorem 16 *Let $MAX_{CORK(i)}(\mathcal{U}, S)$ denote the CORK upper bound for the subgraph set $\mathcal{U}$ and a subgraph S for class $\mathbf{K}_i$ and its complement $\mathbf{K}_{\neg i} = \cup_{j=1,j\neq i}^{K} \mathbf{K}_j$. Then*

$$mcCORK_{1vr}(\mathcal{U} \cup \{T\}) \leq \sum_{i=1}^{K} MAX_{CORK(i)}(\mathcal{U}, S) \ , \tag{2.21}$$

where T is any supergraph of S ($T \sqsupseteq S$).

Proof mcCORK($\mathcal{U} \cup \{T\}$) is a sum of pairwise CORK values $q_i(\mathcal{U} \cup \{T\})$, each of which can be upper-bounded by $\mathrm{MAX}_{\mathrm{CORK}(i)}(\mathcal{U}, S)$. As a consequence, the sum of these upper bounds

$$\sum_{i=1}^{K} \mathrm{MAX}_{\mathrm{CORK}(i)}(\mathcal{U}, S) \tag{2.22}$$

provides an upper bound for the sum of pairwise CORK values

$$\sum_{i=1}^{K} q_i(\mathcal{U} \cup \{T\}) \ , \tag{2.23}$$

that is an upper bound for $\mathrm{mcCORK}_{\mathrm{1vr}}(\mathcal{U} \cup \{T\})$. □

Inequality (2.21) can be used for pruning subtrees in gSpan's DFS search tree, if the upper bound on mcCORK in this subtree is less than the subgraph with maximum mcCORK score encountered so far.

2.2.8 Using Pre-Mined Subgraphs

The gSpan$_{\mathrm{CORK}}$ algorithm introduced in Section 2.2.6 is intended to speed up subgraph enumeration procedures which aim at generating features for classification. However, some datasets already allow for fast subgraph enumeration even without explicitly giving additional pruning criteria such as CORK. Furthermore, one could choose to use an alternative kind of enumeration, not necessarily targeting frequent subgraphs [95, 140, 159]. We now show that given an enumeration of subgraphs, we can convert Algorithm 2 into an offline approach depicted in Algorithm 3.

We first require a conversion of the subgraph enumeration into the canonical form of DFS Codes, such that the subgraphs can be sorted in the same lexicographical order as used by the gSpan traversal (step 3). Then we use this sorting to form a mapping $\mathcal{N}$ of each subgraph at sorting position i to the first

Algorithm 3 Offline_Select$_{\text{CORK}}$

Input: List of subgraphs $\mathcal{S}$ with occurrence patterns $v^{(i)}_{\text{index of } S}$ for all $i \in \{1, \ldots, |\mathcal{G}|\}$

1: **function** OFFLINE_SELECT$_{\text{CORK}}(\mathcal{S})$
2: Generate DFS Codes for the graphs of $\mathcal{S}$
3: Sort $\mathcal{S}$ lexicographically in ascending order
4: $\mathcal{N}$ = integer array of size $|\mathcal{S}|$ ▷ map siblings
5: Fill $\mathcal{N}$ s.t. $\mathcal{N}[i]$ is the position of the next element in $\mathcal{N}$ of which $\mathcal{S}[i]$ is not a prefix
6: $\mathcal{S}^\dagger = \emptyset$
7: $S =$ NULL ▷ next best subgraph
8: $i = 0$
9: **while** $i < |\mathcal{S}|$ **do**
10: **if** $q(\mathcal{S}^\dagger \cup \{\mathcal{S}[i]\}) > q(\mathcal{S}^\dagger \cup \{S\})$ **then**
11: $S = \mathcal{S}[i]$
12: **end if**
13: **if** $\text{MAX}_{\text{CORK}}(\mathcal{S}^\dagger, \mathcal{S}[i]) \leq q(\mathcal{S}^\dagger \cup \{S\})$ **then**
14: $i = \mathcal{N}[i]$ ▷ prune the children of $\mathcal{S}[i]$
15: **else**
16: i++
17: **end if**
18: **end while**
19: **if** $q(\mathcal{S}^\dagger \cup \{S\}) > q(\mathcal{S}^\dagger)$ **then**
20: $\mathcal{S}^\dagger = \mathcal{S}^\dagger \cup \{S\}$
21: **goto** 7 ▷ next DFS Code traversal
22: **end if**
23: **return** $\mathcal{S}^\dagger$
24: **end function**

Output: Set of discriminative subgraphs $\mathcal{S}^\dagger$.

subgraph index $> i$ which does not have the DFS Code of $\mathcal{S}[i]$ as a prefix (step 5). If $\mathcal{S}$ is the result of a gSpan run, $\mathcal{N}$ simply points from any DFS Code to the next DFS Code with a lower or equal number of edges. For treating other enumerations, an actual prefix test may become necessary. We now know that all elements of $\mathcal{S}$ from $i+1$ to $\mathcal{N}[i]$ are children of $\mathcal{S}[i]$ in the DFS Search Tree traversal, and thus supergraphs of $\mathcal{S}[i]$. While now traversing $\mathcal{S}$, looking for the next best subgraph according to CORK, in step 14 we skip those graphs if they can be pruned according to the CORK Upper Bound (2.18).

Using pre-mined subgraphs instead of applying the nested approach of Algorithm 2 can be a strong runtime advantage over gSpan$_{\text{CORK}}$ if

1. the number of frequent subgraphs is relatively low, since then the complete enumeration can be faster than repeated enumerations of bounded DFS code trees,

2. or if the frequent subgraphs are especially large, thus they repeatedly slow down the DFS code minimality test.

2.3 Related Work

In this work, we combine two components to achieve our goal of efficient feature selection among frequent subgraphs with quality guarantees: i) frequent subgraph mining and ii) a submodular quality function. We review related work on both of these components in the following.

2.3.1 Discriminative Subgraph Mining

Discriminative frequent subgraph mining has evolved into a major direction in graph mining research over recent years. We here summarize prominent contributions to this branch of graph mining.

LeapSearch [176] speeds up subgraph mining by heuristically exploiting the fact that structurally similar subgraph patterns tend to have similar frequencies and statistical significance scores, resulting in orders of magnitude speed-up in comparison with state-of-the-art methods.

gBoost [101, 132], is a nested boosting approach, which repeatedly mines a set of frequent subgraphs while optimizing an LPBoost problem. This becomes feasible by iteratively refining pruning bounds which restrict the search space. In [131] Saigo *et al.* propose a faster version of gBoost using partial least squares regression on frequent subgraphs (gPLS).

The MoSS subgraph mining approach by Borgelt *et al.* [17] explicitly mines subgraphs which are frequent in the target class and infrequent in the control class. In [89] Jin *et al.* propose COM, a method for discriminative mining frequent subgraphs based on co-occurrence patterns. Using only one subgraph mining cycle, they iteratively grow a set of rules from the subgraphs mined so far, which is also designed for identifying a target class. Comparatively to MoSS they also use a minimum support threshold for rules involving the target class and a maximum support threshold for rules with patterns matching the control class.

An excellent wrapper approach to the problem of discriminate frequent subgraph mining was published by Koji Tsuda [154]. He uses the LASSO algorithm for mining salient features while exploiting pruning criteria on the used search path. Our approach differs from Tsuda's in two ways: Our feature

selection method is a filter method and hence independent from the choice of the classifier and we can provide optimality guarantees for our solution.

Another class of discriminative pattern mining approaches for graph mining was proposed by [186] and [57] who use a decision-tree-like classifier. For a given dataset, [57] iteratively mine for the most meaningful feature according to the information gain, and split this dataset into two separate problems. They proceed until the subproblems are solved or are of a smaller size than a given threshold.

2.3.2 Related Work on Correspondences

While we here present the first integration of a submodular quality function into the frequent subgraph mining process, there is related work on the quality function we employ. Correspondences were referred to as inconsistencies in Dash *et al.* [38] and used to define another, non-submodular quality criterion. In [19], Boros *et al.* derived CORK from families of Hamming distance measures as

$$\theta(\mathcal{U}) = \sum_{v^{(i)} \in \mathcal{A},\ v^{(j)} \in \mathcal{B}} \begin{cases} 1 & \text{if } \exists\ d \in \mathcal{U} :\ v_d^{(i)} \neq v_d^{(j)} \\ 0 & \text{else} \end{cases} \tag{2.24}$$

They recognized its beneficial greedy selection properties and evaluated other, more involved submodular set functions on small datasets with at most 125 features. We examined whether any of these other submodular set functions could be integrated into gSpan for efficient subgraph mining. However, it turned out that only CORK can be represented in terms of equivalence classes which allows for its efficient computation.

2.4 Experimental Evaluation

In this section, we conduct experiments to examine the effectiveness and efficiency of CORK in finding discriminative frequent subgraphs. After introducing the used graph datasets we will compare CORK to a number of other filter approaches. We first use the number of features selected by CORK as parametrization for all filters and later analyze how the competitors perform for a larger variety of selected features. We continue with a runtime analysis of the nested algorithm gSpan$_{\text{CORK}}$, followed by an improvement recommendation involving an additional threshold. We conclude the experimental section with a comparison to some of the wrapper approaches introduced in Section 2.3.1.

Table 2.1: Topologies of used graph sets; if available with their PubChem ID.
$|\mathcal{G}|$: size of the dataset (also called n)
$|V(G)|$: average number of vertices per graph
$|E(G)|$: average number of edges per graph
$|\mathcal{L}_V|$: number of vertex labels
$|\mathcal{L}_E|$: number of edge labels
K: number of classes

Dataset $\mathcal{G}$	PubChem	$\lvert\mathcal{G}\rvert$	$\lvert V(G)\rvert$	$\lvert E(G)\rvert$	$\lvert\mathcal{L}_V\rvert$	$\lvert\mathcal{L}_E\rvert$	K
NCI1	NCI-H23	4 117	29.8	32.3	43	3	2
NCI33	UACC257	3 298	30.1	32.6	39	3	2
NCI41	PC-3	3 108	30.2	32.8	28	3	2
NCI47	SF-295	4 068	29.8	32.4	44	3	2
NCI81	SW-620	4 812	29.1	31.6	44	3	2
NCI109	OVCAR-8	4 149	29.5	32.1	44	3	2
NCI145	SN12C	3 911	29.6	32.1	37	3	2
NCI330	P388 in CD2F1	4 608	24.9	26.6	47	3	2
DD		1 178	284.3	715.7	82	1	2
DD6C		664	357.9	909.7	63	1	6
AIDS		5 621	27.6	29.7	44	4	3

2.4.1 Datasets

To evaluate our algorithm, we employed the 11 real-world datasets summarized in Table 2.1:[2]

- Anti-cancer screen datasets (NCI): we use 8 datasets collected from the PubChem website as in [159]. They are selected from the bioassay records for cancer cell lines. Each of the anti-cancer screens forms a classification problem, where the class labels on these datasets are either active or inactive in a screen for anti-cancer activity. The active class is extremely rare compared to the inactive class. For a detailed description, please refer to [159] and the website, `http://pubchem.ncbi.nlm.nih.gov`. Each dataset can be retrieved by submitting queries in the above website.

  In order to have a fair comparison on those unbalanced datasets, each dataset has been re-sampled by forming 5 data subsets with balanced classes, where excessive instances from the larger class have been removed.

[2] All datasets (overall size 23.4 MB) are available at `http://www.dbs.ifi.lmu.de/~thoma/pub/sam2010/data.zip` .

- Dobson and Doig (DD) [47] molecule data set: it consists of 1 178 proteins, which can again be divided up into two classes: 691 enzymes and 487 non-enzymes. The vertices of an extracted graph represent the C_α atoms of the amino acids of the corresponding protein. Together with all distinct special conformations, they sum up to 82 vertex labels and are connected if they are at least within 6 Å of each other in the 3D protein structure. In order to retrieve edge labels, discretizing those distances would be possible, but prone to arbitrary thresholding. Consequently, edge labels are omitted. Even in this compacted form, with an average size of 285 vertices and 716 edges, these proteins are larger and more densely connected than the molecules from the NCI screening.

- EC-number groups for DD (DD6C): We furthermore use the DD dataset for differentiating the examples of the enzymes class into their EC numbers [5], a hierarchical categorization system for enzymes. We distinguish between the 6 basic classes, thus transferring the dataset DD into a new dataset DD6C consisting of 664 enzymes that could be mapped to an EC number. Among the remaining enzymes 25 could not be mapped and 2 caused duplicate matches and were thus excluded from DD6C. The topology of this new dataset reveals that the non-enzymes in the original DD dataset appear to be smaller on average than the enzymes which also appear in the DD6C dataset. We thus consider the DD6C problem as harder than the DD problem, not only because of the additional classes, but also because of less pronounced variations between the classes. The class distribution is summarized in Table 2.2.

- AIDS antiviral screen data (AIDS): it contains the activity test information of 43 850 chemical compounds. Each chemical compound is labeled as either active (CA), moderately active (CM) or inactive (CI) with respect to the HIV virus. Among these compounds, 423 belong to CA, 1 081 are of CM, and the rest is in Class CI. This dataset is publicly available on the website of the Developmental Therapeutics Program (`http://dtp.nci.nih.gov/docs/aids/aids_data.html`). As with the NCI datasets, we have transformed this data into a slightly more balanced form of 10 splits, combining the active (CA) and moderately active (CM) compounds with samples of the inactive compounds (CI). The average number of compounds per split is shown in Table 2.1.

In the experiments on these datasets, our CORK procedure selected between 11 and 66 subgraphs of sizes varying between 2 and 12 vertices (=atoms or amino acids), approximately 5% of which contain cycles. This means that subgraph mining procedures restricted to sub-classes of graphs like trees [95]

Table 2.2: DD6C class distribution: Number of enzymes of the DD dataset by EC number.

EC	Name	Count
1	Oxidoreductases	145
2	Transferases	175
3	Hydrolases	214
4	Lyases	66
5	Isomerases	37
6	Ligases	27

or graphs of restricted size [169, 124, 159, 140], which have been developed for less complex outputs and faster runtimes, would not enable us to produce results similar to those of gSpan, the graph mining approach we use.

2.4.2 Comparison to Filter Approaches

CORK is a filter method. Hence in the first experiment, we assessed whether CORK selects subgraphs that generalise well on classification benchmarks, comparing it to state-of-the-art filter methods for subgraph selection.

We use 10-fold cross-validation for classification. Each dataset is partitioned into ten parts evenly. Each time, one part is used for testing and the other nine are combined for frequent subgraph mining, feature selection and model learning. In our current implementation, we use LIBSVM [26] to train a C-SVM classifier based on the selected features. C is optimised within a range of seven values $\{10^{-6}, 10^{-4}, 10^{-2}, 1, 10^2, 10^4, 10^6\}$ / (size of the dataset) by cross-validation on the *training* dataset only. We employ a linear kernel on the selected graph features, and normalise the resulting kernel matrix KM via $\mathrm{KM}_{\mathrm{norm}}(a,b) = \frac{\mathrm{KM}(a,b)}{\sqrt{\mathrm{KM}(a,a)\ \mathrm{KM}(b,b)}}$. We repeat the whole experiment 10 times and we report average results from these 10 runs.

We compare CORK to four state-of-the-art filter methods. Three of them are rankers using Pearson's Correlation Coefficient, the Delta Criterion which is closely related to MoSS [17] and Information Gain as a ranking criterion, and the fourth comparison partner is the Sequential Cover method [45].

2.4.2.1 Pearson's Correlation

Pearson's Correlation Coefficient (PC) is commonly used in microarray data analysis [158, 51], where discriminative genes for phenotype prediction need to be selected from thousands of uninformative ones. In order to deal with the

vast amount of available features, the induced quality criterion is calculated for each feature independently and a pre-defined number of the top-scoring features are selected. The selection criterion is the squared correlation between the occurrence pattern $v_j^{(i)}$ of feature index j and the class label pattern $y^{(i)} \in \{1, \ldots, K\}$. It computes as

$$q_{\mathrm{PC}}(j) = \frac{\sum_{i=1}^{n}(v_j^{(i)} - \bar{v}_j)(y^{(i)} - \bar{y})}{\sqrt{\sum_{i=1}^{n}(v_j^{(i)} - \bar{v}_j)^2 \sum_{i=1}^{n}(y^{(i)} - \bar{y})^2}} \ , \tag{2.25}$$

where $\bar{v}_j = \frac{1}{n}\sum_{i=1}^{n} v_j^{(i)}$ and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y^{(i)}$.

2.4.2.2 Delta Criterion

The difference among subgraph frequencies in different classes is another popular feature selection criterion. For instance, the MoSS mining approach by Borgelt *et al.* [17] is designed for pharmacological screenings which specifically aim for characterizing the positive class. Thus, the idea is to accept only subgraphs which are frequent in the positive group, and infrequent in the complement. From this, we derive the following delta criterion as

$$q_{\mathrm{delta}}(S) = \max\left(\mathcal{A}_{S_1} - \mathcal{B}_{S_1}, \mathcal{B}_{S_1} - \mathcal{A}_{S_1}\right) \ , \tag{2.26}$$

which can be used as a ranker criterion, in a similar way as PC. We extend it to multi-class by taking the difference between the number of hits in the class with the maximum frequency and the remaining average hit count per class:

$$q_{\text{delta MC}}(S) = \max_{i \in \{1,\ldots,K\}} \left(\mathbf{K}_{i,S_1} - \frac{1}{K-1} \sum_{j=1, j\neq i}^{K} \mathbf{K}_{j,S_1} \right) \tag{2.27}$$

2.4.2.3 Information Gain

As a final ranking method, we compare CORK to the Information Gain (IG), an entropy-based measure, which is frequently used in feature selection [180, 126]:

$$q_{\mathrm{IG}}(S) = \sum_{i \in \{0,1\}} \sum_{j=1}^{K} p(S = i, C = \mathbf{K}_j) \log_2 \frac{p(S = i, C = \mathbf{K}_j)}{p(S = i) \cdot p(C = \mathbf{K}_j)} \ , \tag{2.28}$$

where C is the class variable of the tested objects.

2.4.2.4 Sequential Cover

Algorithm 4 outlines the sequential cover method (SC). Frequent graphs are first ranked according to their relevance measure such as information gain, Fisher score, or confidence. In this experiment, we use confidence as the relevance measure:

$$q_{\text{conf}}(S) = \max_{i \in \{1,\ldots,K\}} \frac{\mathbf{K}_{i,S_1}}{\sum_{j=1}^{K} \mathbf{K}_{j,S_1}} \tag{2.29}$$

If a top-ranked frequent subgraph covers some of the uncovered training instances, it will be accepted and removed from the feature set $\mathcal{S}$. The algorithm terminates if either all instances are covered or $\mathcal{S}$ becomes empty. SC can be executed multiple times to make several covers on the instances.

Algorithm 4 Sequential Cover (SC)

Input: Set of frequent subgraphs $\mathcal{S}$, training dataset $\mathcal{G}$

function SC($\mathcal{S}, \mathcal{G}$)
  Sort subgraphs in $\mathcal{S}$ in decreasing order of the chosen relevance measure
  while $\mathcal{G} \neq \emptyset \wedge \mathcal{S} \neq \emptyset$ **do**
    S = first subgraph of $\mathcal{S}$
    $\mathcal{S} = \mathcal{S} \setminus \{S\}$
    if S covers at least one graph in $\mathcal{G}$ **then**
      $\mathcal{S}^\dagger = \mathcal{S}^\dagger \cup \{S\}$
    end if
    for all graph $G \in \mathcal{G}$ covered by S **do**
      $\mathcal{G} = \mathcal{G} \setminus \{G\}$
    end for
  end while
  return $S^\dagger$
end function

Output: Selected set of subgraphs $\mathcal{S}^\dagger$

2.4.2.5 Filter Results

The results of the filter experiments are displayed in Table 2.3. Note that for better comparability, the number of selected features for all experiments was determined via CORK. Potential disadvantages for the other selection approaches are addressed in the next section. Table 2.3(a) shows the number of selected subgraphs $\left|\mathcal{S}^\dagger\right|$ among frequent subgraphs of $\sigma = 10\,\%$, together with the average area under the receiver operating characteristic (ROC) curve

Table 2.3: Classification quality of filter methods (PC = Pearson's Correlation, Delta = the Delta criterion, IG = Information Gain, SC = Sequential Cover, CORK = Correspondence-based Quality Criterion). The number of features $\left|\mathcal{S}^{\dagger}\right|$ was set by CORK selection on frequent subgraphs with $\sigma = 10\,\%$; best results are shown in bold.

(a) Classification AUC values (and standard deviation (Std)) for the 8 NCI graph datasets and on the two-class DD graphs.

		PC		Delta		IG		SC		CORK	
Dataset	$\left\|\mathcal{S}^{\dagger}\right\|$	AUC	Std	AUC	Std	AUC	Std	AUC	Std	AUC	Std
NCI1	57	0.682	0.052	0.724	0.025	0.712	0.024	0.690	0.026	**0.769**	0.023
NCI33	53	0.682	0.053	0.718	0.027	0.698	0.027	0.681	0.029	**0.759**	0.028
NCI41	49	0.681	0.058	0.722	0.023	0.748	0.028	0.732	0.037	**0.763**	0.027
NCI47	56	0.714	0.052	0.728	0.022	0.698	0.026	0.687	0.025	**0.779**	0.024
NCI81	64	0.668	0.068	0.711	0.022	0.731	0.022	0.720	0.024	**0.770**	0.022
NCI109	56	0.699	0.061	0.716	0.026	0.749	0.025	0.719	0.028	**0.774**	0.023
NCI145	55	0.684	0.070	0.717	0.029	0.733	0.035	0.698	0.027	**0.773**	0.029
NCI330	66	0.692	0.044	0.699	0.027	0.676	0.028	0.660	0.025	**0.769**	0.023
DD	15	0.605	0.051	**0.800**	0.038	0.674	0.048	0.694	0.039	0.778	0.038

(b) Multi-class average pairwise AUC estimates ($\widehat{\mathrm{AUC}}_{\mathrm{pw}}$) and classification accuracies (both with standard deviation (Std)).

			PC		Delta		IG		SC		CORK	
Dataset	$\left\|\mathcal{S}^{\dagger}\right\|$	Val.	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
DD6C	14	$\widehat{\mathrm{AUC}}_{\mathrm{pw}}$	0.719	0.018	0.703	0.015	0.715	0.009	0.715	0.027	**0.723**	0.018
		Accuracy	0.341	0.047	0.324	0.033	0.323	0.099	0.355	0.044	**0.359**	0.050
AIDS	55	$\widehat{\mathrm{AUC}}_{\mathrm{pw}}$	0.829	0.001	0.829	0.001	0.829	0.001	0.829	0.002	**0.832**	0.006
		Accuracy	0.733	0.001	0.733	0.001	0.733	0.001	0.733	0.001	**0.735**	0.005

(AUC) and its standard deviation (Std) over 100 conducted experiments. We observe that in all but one dataset, CORK detects the best feature combination for the two-class classification problems at hand.

Table 2.3(b) compares the filter selectors on the two multi-class datasets DD6C and AIDS by their average pair-wise AUC estimate

$$\widehat{\mathrm{AUC}}_{\mathrm{pw}}(\mathcal{G},\mathcal{U}) = \sum_{a=1}^{K} \frac{\left|\left\{d^{\mathcal{U}}_{a,b}(G) = a \mid G \in \mathbf{K}_a, b \in \{1,\ldots,K\} \setminus a\right\}\right|}{(K-1)\cdot|\mathcal{G}|} \tag{2.30}$$

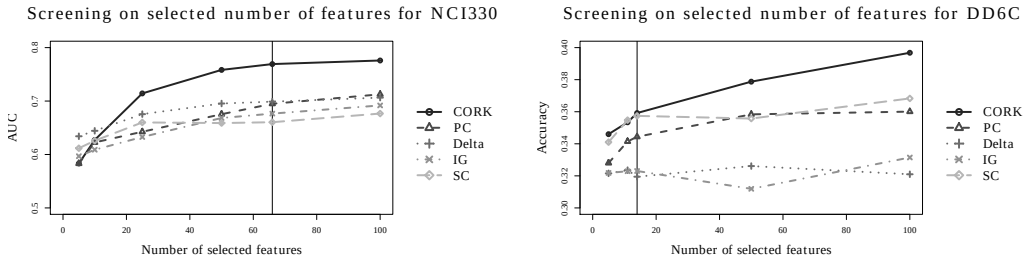
as the fraction of pairwise inter-class decisions in the dataset $\mathcal{G}$ where the decision function $d^{\mathcal{U}}_{a,b}$ of the SVM deciding between the classes a and b votes for the correct class based on the selected subgraphs $\mathcal{U}$. For further orientation, we provide the classification accuracy. As can be seen, CORK performs best for both datasets, although there are no significant differences in accuracy compared to other methods.

It is not surprising that in the vast space of interdependent features spanned by frequent subgraphs, feature combinations are more valuable than the simple ranking approach we used with Pearson's Correlation, the Delta method and the Information Gain. The Sequential Cover method takes into account that all instances should be covered by the selected set of features, yet, can never compete with CORK. We have been rather surprised by the mightiness of the Delta method since it actually scored better than Pearson Correlation.

However, the complexity of the graph classification problem obviously requires the consideration of the various features' interdependence. CORK respects this interdependence by iteratively picking the subgraph feature which optimally complements the set of features selected so far (in terms of resolving correspondences).

2.4.3 Effect of Target Sizes

The number of selected features $\left|\mathcal{S}^{\dagger}\right|$ is an important parameter in feature selection. CORK suggests an automatic bound for the number of selected features, however, the selection procedure can be terminated earlier or restarted for determining fewer or more features. In order to demonstrate the fairness of our evaluation, Figure 2.3 displays screenings over the number of selected features for the tested filter approaches on the two-class problem NCI330 and the multi-class problem DD6C. We see that the number of subgraphs selected by CORK does not represent the optimal number of features for any of the criteria or datasets. However, in all cases, the larger the feature sets get, the smaller the increases in accuracy by adding more features. Moreover, CORK returns the best results for all tested feature sizes above the recommended number of features.



(a) NCI330 dataset (two classes) (b) Multi-class dataset DD6C

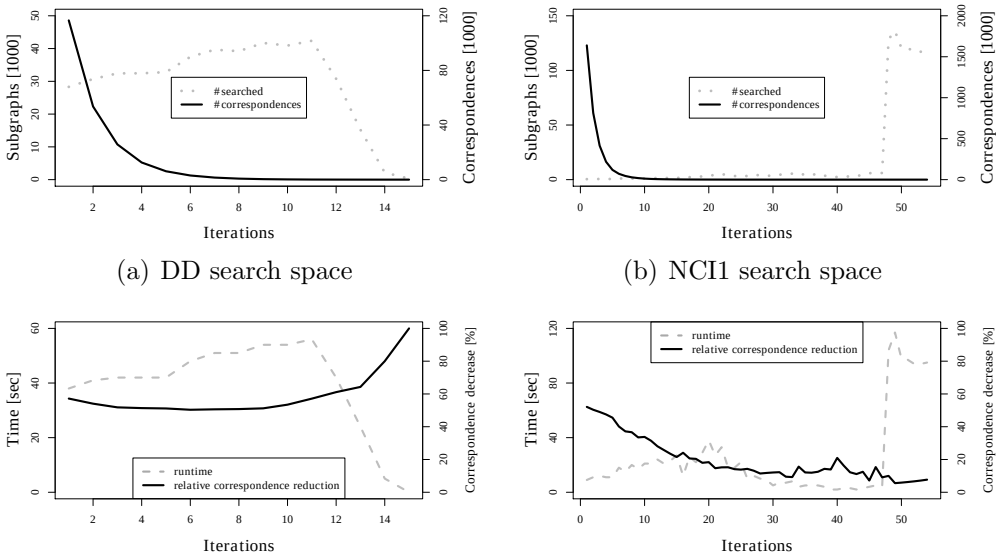
Figure 2.3: Screening of the feature quality over the number of selected features $\left|\mathcal{S}^{\dagger}\right|$ for CORK selection, Pearson's Correlation (PC), the Delta method, Information Gain (IG), and Sequential Cover Selection (SC). The vertical line marks the number of features originally chosen by CORK.

2.4.4 Experimental Runtime Analysis

In our third experiment, we evaluated the runtime performance of nested feature selection, i.e. features are acquired *during* mining, as opposed to un-nested feature selection which takes place *after* mining. We run nested CORK on two complete datasets (the DD dataset and the NCI1 screening in Figure 2.4) and record the number of correspondences and the number of subgraphs examined per iteration. Since previous mining experiments have been handled on training subsets, the number of iterations is slightly elevated ($16 > 15$ and $64 > 57$) as opposed to Table 2.3. All experiments were run on a machine with two Intel Xeon 5160 3.00 GHz Dual-Core processors and 4 GB of main memory.

In the DD experiment (Figures 2.4(a) and 2.4(c)), we observe that in the beginning, we achieve a steep decrease in the number of correspondences, whilst enumerating a comparable number of subgraphs for each of the first 10 iterations and thus maintain an almost constant runtime per iteration. In the end, CORK prunes a larger percentage of the enumerated subgraphs and the iterations speed up. The enumeration stops when all instances from the two classes are separated.

This attractive behaviour can be observed if there exists a (small) subset of subgraph features that eliminates all correspondences. In the other, inseparable case, CORK alone is not able to fully separate the two classes. This does not present a problem in un-nested feature selection, as the procedure simply ends when no new useful features can be identified. However, in the gSpan-nested setting, it may happen, that the complete DFS search tree has to be searched in order to discover that there is no better subgraph. This is illustrated in Figures 2.4(b) and 2.4(d), where the search space cannot be completely resolved, with 33 correspondences remaining.



(a) DD search space (b) NCI1 search space

(c) DD runtimes (d) NCI1 runtimes

Figure 2.4: Nested feature mining experiments on the complete datasets DD and NCI1 (σ is set to 10%): each iteration corresponds to one selected feature. **Upper plots:** number of subgraphs (in 10^3) enumerated for the selection of one feature (dotted-grey, left scale) and number of correspondences (in 10^3) present at each iteration (black, right scale). **Lower plots:** runtime per iteration (dashed-grey, left scale) percentaged decrease in the number of correspondences due to the current feature (in black, right scale).

A way out of this problem is to allow CORK to terminate even if not all correspondences have been resolved, i.e. to introduce a *tolerance threshold* on the number of remaining correspondences.

2.4.5 Impact of a Tolerance Threshold for Correspondences

In our fourth experiment, we assessed the impact of employing a tolerance threshold t that leads to the termination of CORK, i.e. CORK feature selection ends once the number of correspondences falls below t. As demonstrated in Section 2.4.4, in later iterations on inseparable datasets, expensive subgraph mining results in relatively few resolved correspondences. In order to improve the effectiveness of CORK and to prevent over-fitting by meaningless features, we define a tolerance threshold t on the number of correspondences that lead to the termination of the nested mining procedure.

We used the same setting as for the validation runs in Section 2.4.2. For

Table 2.4: Nested CORK versus the two variants of un-nested CORK feature selection ("naïve": no pruning structure, "offline": the pruning approach of Algorithm 3) with varying tolerance thresholds t. The un-nested runtimes are omitting the time needed for the initial enumeration of frequent subgraphs (20 minutes for DD, one minute for NCI33).

(a) DD dataset

	DD Screening			time [min, s]		
t	$\lvert\mathcal{S}^\dagger\rvert$	AUC	Std	nested	naïve	offline
10000	5	0.745	0.036	3'27"	9'28"	23"
1000	8	0.761	0.039	6'01"	15'23"	39"
100	11	0.772	0.039	8'57"	18'41"	56"
10	13	0.776	0.037	10'09"	19'20"	1'01"
0	15	**0.778**	0.037	10'36"	19'28"	1'01"

(b) NCI33 dataset

	NCI33 Screening			time [min, s]		
t	$\lvert\mathcal{S}^\dagger\rvert$	AUC	Std	nested	naïve	offline
10000	10	0.679	0.032	1'21"	1'27"	3"
1000	18	0.707	0.031	3'43"	2'10"	7"
100	31	0.738	0.028	10'06"	2'34"	16"
10	54	0.765	0.023	21'19"	2'48"	30"
0	54	0.765	0.023	23'33"	2'48"	30"

showing the effect of the tolerance threshold, we also compare the runtimes of the nested selection approach $\text{gSpan}_{\text{CORK}}$ to the un-nested variant $\text{offline\_select}_{\text{CORK}}$ and the naïve approach of applying CORK as a common forward feature selection criterion on a pre-mined subgraph set without additional pruning. All CORK selection runs are stopped as soon as they result in less than t correspondences. The results are displayed in Table 2.4.

For the DD dataset (2.4(a)) this summary shows a slight advantage in accuracy of the lower tolerance thresholds 100 and 10, however, the additional runtime does not seem to be worth such an improvement over the quicker alternative of using a threshold of 1000 correspondences. The by far lower runtimes of the nested and offline experiments in comparison to the naïve approach demonstrate the pruning power of MAX_{CORK} over the conventional un-nested variants.

Note that in Table 2.4(a) the runtimes of the nested approach are not only

better than those of naïve forward selection, but they are also competitive to the quick offline variant, since the naïve and offline approaches omit the time necessary to first enumerate the set of frequent subgraphs. When thus counting the enumeration times, gSpan$_{\text{CORK}}$ is the fastest selection approach.

This effect is due to the large number of 110 131 frequent subgraphs for the DD dataset. For datasets which contain fewer frequent subgraphs, like the 2 893 subgraphs for the NCI33 molecule collection in Table 2.4(b), the offline approach and even naïve forward selection can be faster. We also point out the difference in the AUC value between the Tables 2.4(b) and 2.3(a): The CORK evaluation of Table 2.3(a) was achieved by testing offline_select$_{\text{CORK}}$ on a pre-mined set of frequent subgraphs for the *complete* dataset. Of course, we separated the training instances from the test instances in the selection and training phase, however, the frequency bound for the mining step can cause variation in the number of frequent subgraphs between the *complete* and the *training* graphs only (gSpan$_{\text{CORK}}$) and can thus influence the classification performance.

In our experiments, the offline approach has always been faster than the naïve variant. We thus conclude that this algorithm is a useful example of how the gSpan pruning structure can be exploited even after mining has been completed.

2.4.6 Comparison to Wrapper Approaches

The last experiment compares CORK to state-of-the-art wrapper approaches. These wrapper approaches allegedly outperform filter-based approaches in graph mining [154], hence we wanted to get a feeling for the difference in performance. We used the same experimental setup as in Section 2.4.2 and compare CORK to LAR-LASSO and the decision-tree based classifiers of [57] (Table 2.5).

In [57], a query is classified by either directly using the feature tree formed by the subgraph mining routine (M^{b}T), or by building a decision tree on the selected features (DT M^{b}T). We compare the published experiments on the NCI screenings to ours in Table 2.5. Note, however, that the experiments of [57] have been conducted on the complete graph sets, while ours are resulting from balanced subsets of the whole dataset. CORK usually scores better than the model-based search tree approaches M^{b}T and DT M^{b}T, even though these employ by far more subgraphs than CORK. Let us note, that on average those two feature selectors perform slightly better than the simple ranker approach also employing Information Gain (cf. Tables 2.3(a) and 2.5). Information Gain can be submodular, given certain pre-conditions [96]. This, however, is not the case here, since subgraphs are neither independent nor do they represent a

Table 2.5: Classification AUC values (with standard deviation (Std)) on the 8 NCI graph datasets and of the DD graphs (CORK = Correspondence-based Quality Criterion, M^{b}T and DT M^{b}T = Model based search tree approaches – results taken from [57], LAR-SVM = features selected (the same number $|\mathcal{S}^\dagger|$ as CORK) by LAR-LASSO evaluated via SVM). The frequency threshold σ is set to 10%.

		Filter		Wrapper				
		CORK		M^{b}T AUC			LAR-SVM	
Dataset	$\lvert\mathcal{S}^\dagger\rvert$	AUC	Std	$\lvert\mathcal{S}^\dagger\rvert$	M^{b}T	DT M^{b}T	AUC	Std
NCI1	57	0.769	0.023	77	0.685	0.74	0.805	0.021
NCI33	53	0.759	0.028	344	0.743	0.745	0.792	0.024
NCI41	49	0.763	0.027	376	0.765	0.763	0.802	0.025
NCI47	56	0.779	0.024	587	0.708	0.727	0.809	0.023
NCI81	64	0.770	0.022	685	0.696	0.723	0.792	0.021
NCI109	56	0.774	0.023	605	0.699	0.746	0.808	0.022
NCI145	55	0.773	0.029	491	0.747	0.752	0.807	0.022
NCI330	66	0.769	0.023		n.a.		0.797	0.020
DD	15	0.778	0.038		n.a.		0.789	0.039

subset of features mined previously. Thus, our less complex selection criterion still leads to higher quality results.

CORK cannot yet fully compete with the LAR-LASSO wrapper approach by [154]. The nested variant gSpan$_{\text{CORK}}$, however, seems to be more successful in matters of runtime on the Dobson & Doig problem, consisting of significantly larger graphs (see Table 2.1). This observation suggests that CORK pruning may be a useful alternative for datasets of large graphs. In addition, the selection runtimes of offline_select$_{\text{CORK}}$ (between 30 and 60 seconds) are constantly below the runtime of LAR-LASSO (1 to 15 minutes). Furthermore, CORK as a filter method is useful when searching for features irrespective of a specific classifier.

2.5 Summary and Outlook

In this chapter we have proposed a supervised feature selection approach for multi-class classification problems using frequent subgraphs. Since we use a submodular selection criterion, we can provide optimality guarantees for the set of selected features obtained by greedy forward selection. Additionally, we have explained how to integrate this criterion directly into the subgraph

mining process by exploiting an upper bound for pattern-growth extension miners like gSpan. Moreover, we show how to use this bound on a set of pre-mined subgraphs, allowing for more flexibility in the choice of the type of subgraph used.

Similar to information theoretic criteria used for decision trees, CORK measures the quality of a set of features by means of its ability to separate target classes. In our experiments on classification benchmark datasets, the features selected by CORK reach the best accuracies among the filter methods. Among the wrapper methods, CORK outperforms M^bT and DT M^bT in all but one cases. The LAR-LASSO method still achieves a more accurate classification, however, CORK has runtime advantages on pre-mined patterns and large subgraphs.

A strategy to further improve the runtime of our approach is to store the DFS search tree for a set of previously mined frequent subgraphs [154]. When restricting the mining procedure to a fixed minimum support value, this entails much shorter mining times, since gSpan effectively only has to be called once per feature selection step and not several times. Still, the feasibility of this approach obviously depends on the size of the DFS tree that has to be stored.

The goal of future research is to find optimality guarantees for the horizontal leap search strategy for pattern mining proposed in [176], and to speed up CORK by employing this search strategy while maintaining its attractive theoretical properties. Another exciting question is whether our results on the optimality of supervised feature selection can be transferred to techniques for unsupervised feature selection on frequent subgraphs [21] (S. Nijssen, personal communication (2008, 2009)).

Finally, with regard to the overall scope of this thesis, we would like to explore imaging applications of the graph theoretic insights gained in this work.

Chapter 3

Similarity Estimation using Bayes Ensembles

Similarity search and data mining often rely on distance or similarity functions. Queries using these functions should detect instances which are considered to be similar on an intuitive level. Mostly, the underlying object representations, e.g. image features or laboratory measurements, do not reflect this intuition when being queried with standard distance measures like L_p norms. This problem is also called the *semantic gap*. To bridge this gap between feature representation and object similarity, the distance function has to be adjusted to the current application context or the current user.

In [54], we have therefore proposed a probabilistic framework for estimating a similarity value based on a Bayesian setting. Our framework provides a train-able distance function for real-valued feature vectors. This function consists in an ensemble of weak Bayesian learners, each corresponding to a dimension of an implicit feature space. In order to find this implicit feature space with independent dimensions of maximum meaning for the current context, we apply a space transformation based on eigenvalue decomposition.

In our experiments, we demonstrate that our new method shows promising results compared to related Mahalanobis learners on several test datasets w.r.t. nearest-neighbor classification and precision-recall-graphs.

3.1 Introduction

Learning similarity functions is an important task for image retrieval and data mining in general. In data mining, distance measures can be used in various algorithms for classification and clustering. In order to improve classification, learned distance measures can be plugged into any instance-based learner like a k-NN (k-nearest neighbor) classifiers. Though clustering is basically an un-

supervised problem, learning a similarity function on a small set of manually annotated objects is often sufficent to guide clustering algorithms into grouping semantically similar objects.

Adaptive similarity measures provide a powerful tool to bridge the semantic gap between object representations and user expectations. In most settings, the similarity between two objects cannot be described by a standardized distance measure fitting all applications. Instead, it is often a matter of application context and personal preference. Thus, two objects might be similar in one context while they are not in another. For example, assume an image collection of various general images of persons, vehicles, animals, and buildings. In this context, a picture showing a red Ferrari will be considered as quite similar to a picture of a red Volkswagen. Now, take the same images and put them into a different context like a catalogue of rental cars. In this more specialized context, both pictures will most likely be considered as dissimilar. An important assumption in this work is that there is no exact value specifying object similarity. Instead, we consider object similarity as the probability that a user would label the objects as similar.

Learning a distance or similarity function requires a general framework for comparing objects. In most established approaches to similarity learning, this framework is provided by using Mahalanobis distances or quadratic forms. In general, a Mahalanobis distance can be considered to be the Euclidean distance in a linear transformation of the original feature space. Thus, Mahalanobis distances are metric distance functions guaranteeing reflexivity, symmetry and the triangular inequality. Furthermore, the computed dissimilarity of two objects might be increased infinitely. We are going to argue that these mathematical characteristics are unnecessarily strict and sometimes even against intuition when trying to construct a similarity measure.

As an example, it is known from cognition science that humans do not distinguish dissimilar objects to an infinite degree. A human would not care whether object o_1 is *more dissimilar* to the query object q than object o_2 after having decided that both objects o_1, o_2 have nothing in common with the query object q. On the other hand, in most feature transformations, it is possible that two different objects are mapped to the same feature representation. Thus, even if we can guarantee that two objects having a zero distance are represented by the same feature description, we have no guarantee that the corresponding objects should be considered to be maximally similar as well.

Hence, inspired by an approach of [97], we describe similarity in a different way by considering it as the probability that an object o is relevant for a similarity query object q. The core idea of our similarity estimation approach is to consider each feature as evidence for similarity or dissimilarity. Thus, we can express the implication of a certain feature dimension i to the similarity

of objects o and q as a probability $p(\text{SIMILAR}(o,q) \mid (o[i]-q[i]))$. To calculate this probability, we employ a simple one-dimensional Bayes estimate (BE).

However, in order to build a statement comprising all available information about object similarity, we do not build the joint probability over all features. We argue that in most applications, considering a single feature is not sufficient to decide either similarity or dissimilarity. Thus, to derive a joined estimation considering all available features, we average the probabilities derived from each BE. Our new estimate is basically an ensemble of weak Bayesian learners. Therefore, we call our new dissimilarity function Bayes Ensemble Distance (BED). A major benefit of BED is that dissimilarity is very insensitive to outlier values in a single dimension which is a drawback of classical L_p-norm based measures. The major factors to successfully employing an ensemble of learners are the quality and the independence of the underlying weak classifiers. Therefore, we will introduce a new optimization problem that derives a linear transformation of the feature space, allowing the construction of more descriptive BEs. To conclude, the following sections will provide:

1. A discussion about L_p-norms and Mahalanobis distances for modelling object similarity.
2. A new framework for similarity estimation that is built on an ensemble of Bayes learners.
3. An optimization method for generating a linear transformation of the feature space that is aimed at deriving independent features which are suitable for training high quality weak classifiers.

The rest of this chapter is organized as follows. In Section 3.2, we discuss L_p norms and Mahalanobis distances for modeling object similarity. Our framework for modeling object similarity is described in Section 3.3. In Section 3.4, we introduce an optimization problem to derive an affine transformation that allows the training of more accurate Bayes estimates. Section 3.5 briefly reviews related similarity learners. Afterwards, Section 3.6 illustrates the results of our experimental evaluation comparing our new method with related metric learners on several UCI classification datasets and two image retrieval datasets. Finally, we will close Section 3.7 with a summary and some directions for future work.

3.2 L_p-norms and Problem Definition

The task of similarity learning is to find a function, mapping a pair of objects o_1, o_2 to a similarity value $\text{SIM}(o_1, o_2)$ describing how strongly the first object

resembles the other one in the best possible way. To train this function, it is necessary to have training examples representing the notion of similarity which underlies the given application. Let us note that there might be various notions of similarity on the same dataset depending on the application context or even the current user. Thus, a similarity learner should be as flexible as possible to adjust to any given example set.

Basically, there are two categories of examples used for learning similarity functions. The first type is providing class labels to a training set indicating that objects with equal labels are similar and objects with different labels are considered as dissimilar. Most machine learning approaches in metric learning use class labels because most of the proposed methods in this area aim at improving the accuracy of instance-based learners. One important advantage of this type of labeling is that there is a large variety of classification datasets available. Additionally, having n labeled objects results in $\frac{n \cdot (n-1)}{2}$ labeled object pairs. Finally, in classification datasets, the labeling is usually quite consistent because the classes are mostly reproducible by several persons.

As a drawback of this approach, it is required to find a universal set of classes before learning a similarity function. Thus, this type of training examples is difficult to use when learning similarity measures for similarity search. The second type of user feedback is direct relevance feedback providing a similarity value for a set of object pairs. Using relevance feedback allows to determine a degree of similarity for each pair and thus, the similarity information is not necessarily binary. Additionally, relevance feedback does not require to define explicitly known classes and is thus more attractive for similarity search systems.

A drawback of relevance feedback is that labelling a sufficiently large set of object pairs with similarity scores is usually much more strenuous than labelling objects with classes. Furthermore, it is often much more difficult to generate a consistent labelling because there rarely are well-defined criteria for object similarity.

After describing the labels of our examples, we will now formalize our object descriptions, i.e. the feature vectors. A feature is a type of observation about an object and the corresponding feature value describes how an object behaves w.r.t. this type of observation. Mathematically, we will treat a feature F as a numerical value $x_F \in \mathbb{R}$. Considering a predefined number of features d leads to a feature vector $x \in \mathbb{R}^d$. Formally, a training example in our setting is a triple (x_1, x_2, y) where $x_1, x_2 \in \mathbb{R}^d$ are two d-dimensional feature vectors and $y \in [0, l]$ is a similarity score, i.e. $y = l$ represents maximum similarity whereas $y = 0$ describes maximum dissimilarity. In case of class labels, we assign 1 to similar and 0 to dissimilar objects. The most common approach for describing object similarity is to sum up the differences of feature values which is the

basis of L_p-norm-based similarity. Given two feature vectors $x_1, x_2 \in \mathbb{R}^d$, the L_p-norms are defined as:

$$L_p(x_1, x_2) = \left(\sum_{i=1}^{d} |x_{1,i} - x_{2,i}|^p \right)^{\frac{1}{p}} \tag{3.1}$$

For $p = 2$, the L_p-norm is called Euclidean distance which is the most common distance metric in similarity search and distance-based data mining. Semantically, we can interpret the L_p-norm as an evidence framework. Each feature represents an observation about an object and the difference of feature values determines how similar two objects behave with respect to this observation. Since a single observation is usually not enough to decide similarity, all observations are combined. By summing up over the differences for each observation, the L_p-norm describes the degree of dissimilarity of two objects. The parameter p determines the influence of large difference values in some dimensions to the complete distance. For $p \to \infty$, the object distance is completely determined by the largest object difference in any dimension. Let us note that the exponent $\frac{1}{p}$ is used for normalization reasons only. Therefore, it is not required in algorithms that require a similarity ranking.

Given a specialized application context, the standard L_p-norms have several drawbacks:

1. Correlated features are based on the same characteristics of an object and thus, they implicitly increase the impact of this characteristics when calculating the dissimilarity.

2. Not each observation is equally important when deciding about object similarity. When, for instance, deciding between large and small people, the *height* parameter will be more significant than the *weight* parameter.

3. In order to have a large distance w.r.t. an L_p-norm, it is sufficient to have a considerably large difference in any single feature. Correspondingly, a small dissimilarity requires that both vectors display small difference values in each feature. On the other hand, to decide dissimilarity, any single feature is sufficient. This effect is a serious drawback because object similarity might not necessarily always depend on the same set of features. Having an extraordinarily large difference w.r.t. a single rather unimportant feature could thus prevent two otherwise identical objects from being found in a similarity query. Thus, we argue that dissimilarity as well as similarity should be decided based on a combination of several features.

To solve the problems (1) and (2), the Euclidean distance has been extended to the Mahalanobis distance or quadratic form. The idea of this approach is to employ an affine transformation of the original feature space which is applied within the distance measure itself:

$$D_{\text{Mah}}(x_1, x_2) = \left((x_1 - x_2)^T \cdot A \cdot (x_1 - x_2)\right)^{\frac{1}{2}} \tag{3.2}$$

In order to make D_{Mah} a metric, the transformation matrix $A \in \mathbb{R}^{d \times d}$ has to be positive definite. In this case, A implies an affine transformation of the vector space B where the Euclidean distance is equivalent to D_{Mah} in the original space.

$$\left((x_1 - x_2)^T A (x_1 - x_2)\right)^{\frac{1}{2}} = \left((x_1 - x_2)^T B^T B (x_1 - x_2)\right)^{\frac{1}{2}} \tag{3.3}$$

$$= \left((Bx_1 - Bx_2)^T (Bx_1 - Bx_2)\right)^{\frac{1}{2}} \tag{3.4}$$

When properly derived, this matrix A can achieve that the directions in the target space are uncorrelated. Additionally, the directions are weighted by their importance to the given application. There are multiple methods to learn a proper Mahalanobis distance like Fisherfaces [12], RCA [7], ITML [40] or LMNN [164] which are described in Section 3.5.

However, the Mahalanobis distance does not adequately solve the third problem named above because the feature values are only linearly scaled. Thus, all observed difference values are decreased by the same factor. Therefore, by preventing a too large impact in some distance calculations, we would generate too small distance values in others. To conclude, Mahalanobis distances are still equivalent to an Euclidean distance in a transformed data space and thus, these methods are no solution to the third problem mentioned above.

3.3 Ensembles of Bayes Estimates

In the following, we formally describe our method. We start with the definition of Bayes Estimates (BE) and Bayes Ensemble Distance (BED) on the original feature dimensions. Afterwards, we introduce our solution to the problem of correlated features and provide a new way to derive an affine transformation of the feature space that allows the training of a meaningful BED.

3.3.1 Bayes Estimates and Bayes Ensemble Distance

As mentioned above, we want to learn a function having a pair of feature vectors as input and returning a similarity score as output. Similar to the L_p-norm, we describe the comparison between two feature vectors $x_1, x_2 \in$

$\mathbb{R}^d$ by their difference vectors $(x_1 - x_2)$, or $(x_2 - x_1)$. Thus, our method assigns a similarity score to each difference vector. Since both difference vectors should provide the same dissimilarity score, we have to make sure that our similarity function is symmetric with respect to the direction of the input difference vector. As mentioned before, our approach treats each dimension of the input space separately. Thus, we define the Bayes Estimates (BE) for feature dimension i as a Bayes classifier receiving a difference value $x_{1,i} - x_{2,i}$ as input.

This classifier distinguishes object comparisons of similar objects (SIM) from comparisons of dissimilar objects (DIS). Thus, we learn two distribution functions over the difference values for similar objects and dissimilar objects. Additionally, we employ a prior distribution describing whether similarity is less likely than dissimilarity. As a result, we can calculate the conditional probability $P(\text{DIS} \mid x_{1,i} - x_{2,i})$ describing the dissimilarity likelihood for two objects under the condition of the observed difference value in dimension i. Correspondingly, $P(\text{SIM} \mid x_{1,i} - x_{2,i})$ expresses the likelihood that two objects are similar and it can be used as a similarity function. Formally, the Bayes Estimate (BE) for comparing two vectors $x_1, x_2 \in \mathbb{R}^d$ w.r.t. dimension i is defined as:

Definition 17 (Bayes Estimate) *Let $x_1, x_2 \in \mathbb{R}^d$ be two feature vectors. Let p_s and p_d represent a prior distribution describing the general likelihood that objects are considered to be similar. Then, the* Bayes Estimate (BE) *for x_1 and x_2 w.r.t. dimension i is defined as follows:*

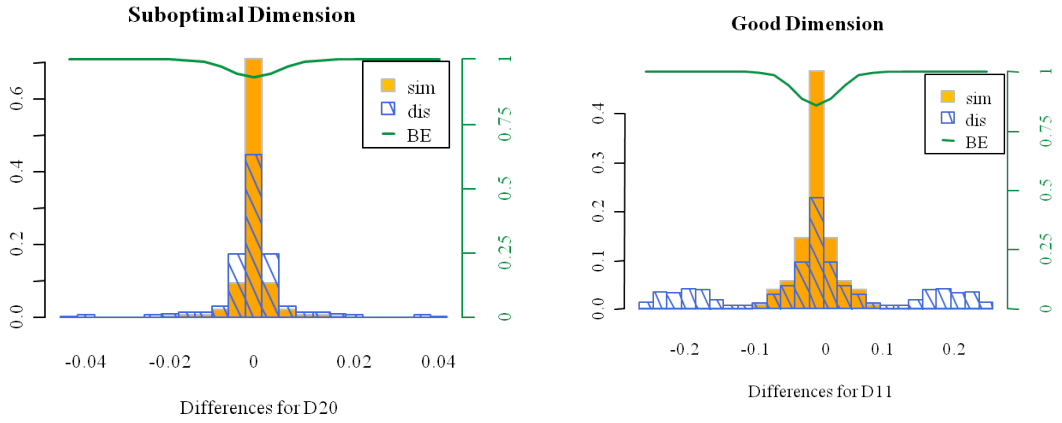
$$\text{BE}_i(x_1, x_2) = \frac{p_d \cdot P((x_{1,i} - x_{2,i}) \mid \text{DIS})}{P_{total}(x_{1,i}, x_{2,i})} \ , \tag{3.5}$$

where $P_{total}(x_{1,i}, x_{2,i})$ is the sum of the similarity and the dissimilarity probabilities ($p_s \cdot P\left((x_{1,i} - x_{2,i}) \mid \text{SIM}\right)$ and $p_d \cdot P\left((x_{1,i} - x_{2,i}) \mid \text{DIS}\right)$) in the i^{th} dimension.

To combine these probabilities, we take the average estimates over all dimensions. Thus, we employ an ensemble approach combining the descriptiveness of all available features. Let us note that this approach is different from building the joint probability for class DIS like in an ordinary Naïve Bayes classifier (NB):

$$\text{NB}(x_1, x_2) = \frac{1}{\text{scale}} \cdot \prod_{i=1}^{d} \text{BE}_i(x_1, x_2) \tag{3.6}$$

The Naïve Bayes approach would imply that in order to be similar, two objects have to be sufficiently similar in each dimension. Correspondingly, dissimilarity would require a sufficiently large difference value in all dimensions. Thus,



(a) Conf-hsv (D 20) (b) Conf-hsv (D 11)

Figure 3.1: Difference distributions for similar (orange bars) and dissimilar (blue, dashed bars) objects in an image retrieval dataset in dimension 20 and 11. The green line corresponds to the second y-axis and it visualizes the Bayes Estimate (BE) for the given difference.

the joint probability could again be determined by a single dimension. By using the average, our method offers a more flexible understanding of similarity: neither a very large difference nor a very small difference in a single dimension can imply similarity or dissimilarity on its own and hence, the resulting distance is more stable against outlier dimensions. Formally, we define the Bayes Ensemble Distance (BED) in the following way:

Definition 18 (Bayes Ensemble Distance) *Let $x_1, x_2 \in \mathbb{R}^d$ be two feature vectors. The* Bayes Ensemble Distance (BED) *for x_1 and x_2 is defined as follows:*

$$\mathrm{BED}(x_1, x_2) = \frac{1}{d} \cdot \sum_{i=1}^{d} \mathrm{BE}_i(x_1, x_2) \tag{3.7}$$

From a data mining point of view, the BED is an ensemble of d weak Bayesian learners, each deriving a probabilistic statement from the corresponding feature. Each learner distinguishes two classes, i.e. similarity and dissimilarity. Let us note that our method does not directly distinguish degrees of similarity. Instead, a quantitative view on object similarity is provided by the average probability that both objects are similar.

An open issue to the use of BED is the type of probability distribution being used to model the Bayes Estimate. To select a well-suited probability

density function, we examined several data sets with respect to their difference vector distribution for similar and dissimilar objects. We built histograms on the observed difference values in each dimension. Remember that all distributions have to be symmetric to the origin because of the pairwise appearance of positive and negative distance values. An example for the histograms derived from two image retrieval datasets is displayed in Figure 3.1 with a badly separable case in Figure 3.1(a) and a partially separable case in Figure 3.1(a). For a description of the used dataset, please refer to Section 3.6.2.

In this dataset and many of the others we examined, we observed a normal distribution for similar objects. Very similar or identical objects will usually display similar feature values. For the distributions describing dissimilarity, we sometimes observed distributions that also resemble a normal distribution but displayed a larger variance. In cases having well separated classes, the dissimilarity distribution is often split into three components, one for large positive, one for low negative difference values and a bulk distribution centered around the origin. Thus, the dissimilarity resembled a mixture model having two symmetric components of equal weight where the first has a positive mean value and the second component has a negative mean value. In our experiments, we employed Gaussians as basis distribution. However, the general method is applicable for any other type of distribution function, e.g. exponential power distributions.

3.3.2 Training Bayes Ensemble Distances

Training BEDs consists of determining the distribution parameters for each dimension, e.g. mean and variance for a Gaussian. Furthermore, it is often useful to determine prior probabilities for similarity and dissimilarity.

In case that the examples are provided with class labels, it is easy to decide whether an object comparison is counted for the similar class (SIM) or for the dissimilarity class (DIS). If both objects belong to the same class, the observed difference value contributes to the SIM distribution. If both objects belong to different classes, the observed difference vector contributes to the distribution describing DIS.

For small data sets, it is possible to consider all possible difference vectors occurring in the training set. However, this approach is not feasible for large datasets because the number of difference vectors is increasing with the squared number of training vectors. Thus, it is often advisable to select a subset of the difference vectors instead of employing all available samples. To find this subset, random sampling is applicable. In our experiments, we adapt the idea of target neighbors from [164] and select the difference vectors corresponding to the k-nearest neighbors of the same class and the k-nearest neighbors belonging

to any other class for each training object. We employed the Euclidean distance to determine the target neighbors.

In case of labeled pairs, selecting examples is usually not an option because each object comparison has to be manually labeled and thus, it is rather unlikely that there will be too many examples for efficient training. However, labeling object pairs allows to distinguish several degrees of similarity $y \in [0..1]$, e.g. the label could indicate a similarity of 0.8 or 0.1. To employ these more detailed labels, we propose to proceed in a similar way as in EM clustering and let the training example contribute to both distributions. In order to consider the class labels, we weight the contribution to the similar distribution by y and the contribution to the dissimilar distribution by $1 - y$. This way, undecidable comparisons having a label of 0.5 would equally contribute to both distribution functions, whereas a comparison having a label of 1.0 would exclusively contribute to the similar distribution.

In many applications, using a prior distribution can improve the accuracy of similarity search and object classification. Especially when using BED for nearest neighbor classification, we can assume that we know how many objects belong to the same class and how many objects belong to any other class. In these cases, we can determine the frequency $|c_i|$ of examples for each class $c_i \in C$ in the training set and easily derive the prior probability for similarity:

$$p_s = \frac{\sum_{c_i \in C} |c_i|^2}{(\sum_{c_i \in C} |c_i|)^2} \qquad \text{and thus:} \qquad p_d = 1 - p_s \tag{3.8}$$

In other words, we know that there are $|c_i|^2$ comparisons of similar objects within each class c_i. Dividing the amount of these comparisons by all possible comparisons computes the relative frequency of p_s. Since we only distinguish two cases, we can calculate p_d as $1 - p_s$.

In case of relevance feedback, directly determining the relative portion of similarity in the training objects is also easily possible. However, depending on the selection of the object pairs to be labeled it is often very unlikely that the label distribution is representative for the distribution on the complete database. Thus, it is often more useful to manually assign a value for the occurrence of each class.

3.4 Optimizing the Feature Space for Bayes Estimates

Employing BED on the original dimensions ensures that neither similarity nor dissimilarity can be decided based on the difference value in a single di-

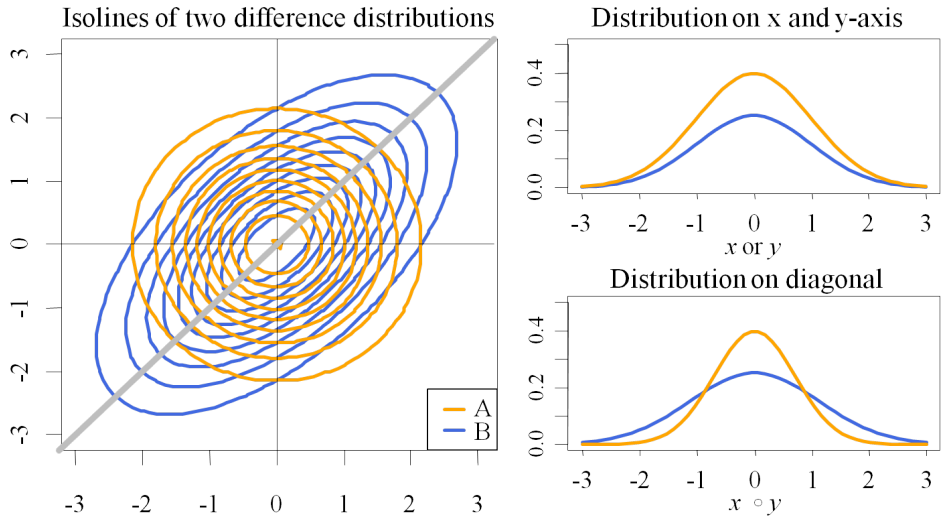


Figure 3.2: Idealized distributions of similar (orange) and dissimilar (blue) objects in a bivariate dataset. Isolines of multivariate Gaussians (left) and projections onto different hyperplanes (right).

mension. Additionally, the importance of each dimension is indicated by the distinction of both distribution functions. However, correlated features still pose a problem for the performance of BED.

First of all, the advantage of using an ensemble of learners strongly depends on their statistical independence. Additionally, it can occur that the single BEs in the original dimension are not very informative. However, there often exist projections of in the data space allowing a good separation of the distribution function. An example is illustrated in Figure 3.2. In the displayed case, the distributions of similar and dissimilar objects are modeled as multivariate Gaussians. If we consider the projection of both distributions onto the x-axis, we cannot decide between the two distributions at all. Projecting the Gaussians onto the main diagonal enables a clearer separation. In this example, it can be assumed that the BE on the main diagonal has a much stronger predictive quality. To conclude, analogously to the Euclidean distance, BED can be improved by a linear transformation of the input space which decreases feature dependency and provides features allowing meaningful similarity estimation.

Formally, we want to find a set of base vectors $W = [w_1, \ldots, w_{d^*}]$ for transforming each original vector $x \in \mathbb{R}^d$ into another d^*-dimensional feature space where each new dimension allows to build a better BE. Since we want to have independent learners, we additionally require that $w_i \perp w_j$ for $i \neq j$.

To determine the suitability of a dimension to train a useful BE, we need to

find a criterion that is independent of the used type of distribution function. A certain dimension in the feature space is useful in case that the distance values between similar objects are in average smaller than the distance values of dissimilar objects. Let us note that the mean value for both distributions has to be zero regardless of the underlying density function. Since distance values always occur in pairs of negative and positive values, the mean is always zero in each dimension. Now, a direction is well-suited if the distance values being observed for similar objects are grouped closer to the origin than the values being observed for dissimilar objects. To quantify this intuition, we calculate the variance of the samples for both distributions SIM and DIS in dimension i and build the difference between both values:

$$\begin{aligned} q_i &= \frac{1}{n} \cdot \left(\sum_{x_d \in \text{DIS}}^{n} \left(x_{d,i}^2 - 0\right) - \sum_{x_s \in \text{SIM}}^{n} \left(x_{s,i}^2 - 0\right) \right) = \\ &= \frac{1}{n} \cdot \left(\sum_{x_d \in \text{DIS}}^{n} x_{d,i}^2 - \sum_{x_s \in \text{SIM}}^{n} x_{s,i}^2 \right) \end{aligned} \tag{3.9}$$

If q_i is large, the difference values between similar objects are generally grouped more closely around zero than the difference values between dissimilar objects in dimension i. If q_i converges to zero, dimension i will usually not allow the training of a useful BE.

To describe the variance along all possible linear projections in the space of distance values, we can build the covariance matrix for similar and dissimilar difference vectors. We define Σ_{SIM} as

$$(\Sigma_{\text{SIM}})_{i,j} = \sum_{x_s \in \text{SIM}} (x_{s,i} - 0) \cdot (x_{s,j} - 0) = \sum_{x_s \in \text{SIM}} x_{s,i} \cdot x_{s,j} \ . \tag{3.10}$$

Σ_{DIS} is built correspondingly on the difference vectors of dissimilar objects.

Our task is to find a set of orthogonal dimensions for which the difference between the variance of the dissimilar distribution and the variance of the similar distribution is as large as possible. Formally, we can define the following optimization problem:

$$\begin{aligned} &\text{Maximize } L(w_i) = w_i^T \Sigma_{\text{DIS}} w_i - w_i^T \Sigma_{\text{SIM}} w_i = w_i^T \cdot \left(\Sigma_{\text{DIS}} - \Sigma_{\text{SIM}}\right) w_i \\ &\quad \text{s.t. } w_i \perp w_j \end{aligned} \tag{3.11}$$

The following eigenvalue equation solves this problem:

$$\lambda w = \left(\Sigma_{\text{DIS}} - \Sigma_{\text{SIM}}\right) \cdot w \ . \tag{3.12}$$

To integrate the learned affine transformation into the training of BED, we can either transform all feature vectors before training and testing by W or integrate the transformation directly into the BE distance by rotating each difference vector before it is processed.

A final aspect of this space transformation is that it allows to reduce the number of considered dimensions. One way to do this, is to select a fixed number of features d^* and to keep only the top d^* dimensions w.r.t. the quality q_i. Alternatively we could determine a threshold τ and keep only those dimensions offering a quality q_i which is better than τ. The choice of τ, however, would be completely dependent on the training dataset, thus we decided to test an optional extension which enables the explicit naming of the desired dimension d^* of the implicit feature space.

To conclude, the training of a BED is summarized in Algorithm 5.

Algorithm 5 BED Training

Input: Difference vectors $\text{SIM} \subseteq \mathbb{R}^d$ of similar object pairs and dissimilar object pairs $\text{DIS} \subseteq \mathbb{R}^d$, optional target dimension d^*.

function TRAIN_BED(SIM, DIS, $d^* = 0$)
  Derive Σ_{SIM} and Σ_{DIS} as in (3.10)
  Compute $W \in \mathbb{R}^{d \times d}$ by solving (3.12)
  Get weights $q \in \mathbb{R}^d$ using (3.9) in the rotated feature space $W^T X$
  if $d^* > 0$ **then** ▷ reduce dimension?
    $W^* \leftarrow \emptyset, q^* \leftarrow \emptyset$ ▷ initialize new parameters
    for all $i \in \{1, \ldots, d\} :\ q_i \in$ top d^* weights of q **do**
      $W^* \leftarrow [W^*, w_i]$ ▷ keep component i
      $q^* \leftarrow [q^{*T}, q_i]$ ▷ and the decision weight
    end for
    $(W, q) \leftarrow (W^*, q^*)$
  end if
  return (W, q)
end function

Output: BED parameters for trained BED(x_1, x_2) of (3.13).

The new BED is then defined as follows:

Definition 19 (Trained Bayes Ensemble Distance) *Let $x_1, x_2 \in \mathbb{R}^d$ be two feature vectors. Let $W \in \mathbb{R}^{d \times d^*}$ be the rotation matrix and let $q \in \mathbb{R}^{d^*}$ be the trained dimension weights. Then, the* Trained Bayes Ensemble Distance

(BED) *for x_1 and x_2 is defined as follows:*

$$\mathrm{BED}(x_1, x_2) = \left(\sum_{i=1}^{d^*} q_i \right)^{-1} \cdot \sum_{i=1}^{d^*} q_i \cdot \mathrm{BE}_i(W^T x_1, W^T x_2) \tag{3.13}$$

3.5 Related Work

In this section, we briefly review existing approaches to similarity learning. We will focus on the field of related feature transformation techniques.

3.5.1 Metric Distance Learning

Most distance learning methods use the Mahalanobis distance, represented by a semi-definite matrix. The shared principle among all of those approaches is to ensure that the relations among a dataset's objects are transformed such that they best represent an underlying characteristic of the data.

In the following, we give a short summary of existing metric learning approaches. For detailed surveys, see [179, 178]. The main idea of unsupervised approaches is to reduce the feature space to a lower-dimensional space in order to eliminate noise and enable a more efficient object comparison. The criteria for selecting such a subspace are manifold. Principal Component Analysis (PCA) [68] builds an orthogonal basis aimed at best preserving the data's variance, Multidimensional Scaling (MDS) [34] seeks the transformation which best preserves the geodesic distances and Independent Component Analysis (ICA) [33] targets a subspace that guarantees maximal statistical independence. ISOMAP [151] by Tenenbaum *et al.* is a non-linear enhancement of the MDS principle, in identifying the geodesic manifold of the data and preserving its intrinsic geometry. Other unsupervised approaches (e.g. [129, 13]) try to fulfill the above criteria on a local scale.

Among supervised approaches, the first to be named is Fisher's Linear Discriminant (FLD) [62]. It maximizes the ratio of the between-class variance and the within-class variance using a generalized eigenvalue decomposition. This method has been extended by Belhumeur *et al.* [12] to the Fisherfaces approach. It precedes FLD with a reduction of the input space to its principal components and can thus filter unreliable input dimensions. BED and especially the target function L share several important ideas with Fisherfaces. However, FLD assumes that the data is partitioned into classes which are modeled using the Gaussian distribution function, whereas BED does not require explicit object classes. Furthermore, the BED is not determined to the use of Gaussian functions. Instead BEDs employ the difference vectors

and always try to distinguish the two basic statements of object similarity and object dissimilarity which can be modeled by an arbitrary symmetric density function. Both methods generate covariance matrices of difference vectors representing similarity (in FLD: the within-class scatter matrix) and dissimilarity (in FLD: the between-class scatter matrix). However, in FLD the matrices are built based on the difference vectors w.r.t. a mean value whereas BED directly employs object-to-object comparisons. Where FLD tries to find dimensions where the ratio between the variances of dissimilarity and similarity are as large as possible, BED maximizes the difference between the variances of the dissimilarity and the similarity distributions.

With RCA [7], Bar-Hillel *et al.* focus on the problem of minimizing within-*chunklet* variance. They argue that between-class differences are less informative than within-class differences and that class assignments frequently occur in such a way that only pairs of equally-labelled objects can be extracted. These pairs are extended into chunklets (sets) of equivalent objects. The inverse chunklet covariance matrix is used for calculating the Mahalanobis distance. This step should usually be preceded by dimensionality reduction. The main difference between BED and RCA is that RCA does not build a distribution function for object comparison corresponding to dissimilarity. Correspondingly, RCA only requires examples for comparison between the objects of the same class. As a result, the optimization which is provided by RCA is not aimed at distinguishing both classes of difference vectors. Instead, RCA is mostly based on a whitening transformation of a matrix which is similar to the within-class-scatter-matrix of FLD.

NCA [67] proposed by Goldberger *et al.* optimizes an objective function based on a soft neighborhood assignment evaluated via the leave-one-out error. This setting makes it more resistant against multi-modal distributions. The result of this optimization is a Mahalanobis distance directly aimed at improving nearest-neighbor classification. The objective function is, however, not guaranteed to be convex.

With Information-Theoretic Metric Learning (ITML) [40], Davis *et al.* propose a low-rank kernel learning problem which generates a Mahalanobis matrix subject to an upper bound for inner-class distances and a lower bound to between-class distances. They regularize by choosing the matrix closest to the identity matrix and introduce a way to reduce the rank of the learning problem.

LMNN (Large Margin Nearest Neighbor) [164] by Weinberger *et al.* is based on a semi-definite program for directly learning a Mahalanobis matrix. They require *k-target neighbors* for each input object x, specifying a list of objects, usually of the same class as x, which should always be mapped closer to x than any object of another class. These k-target neighbors are the within-class k-

nearest neighbors. Hence, the loss function consists of two terms for all data points x: the first penalizes the distance of x to its k-target neighbors and the second penalizes close objects being closer to x than any of its target neighbors. In [165], they propose several extensions, involving a more flexible handling of the k-target neighbors, a multiple-metric variant, a kernelized version for datasets of larger dimension than size and they deal with efficiency issues arising from the repeated computation of close objects. Nonetheless, LMNN requires a specialized solver in order to be run on larger datasets.

3.5.2 Non-Metric Distance Learning

In order to be metric, a distance has to fulfill the metric axioms (i.e. self-similarity, symmetry, triangle inequality). In fact, several recent studies have shown that these axioms (triangle inequality above all) are often not conform with the perceptual distance of human beings [135, 155] and thus not suitable for robust pattern recognition [87].

Most of the approaches learning a non-metric function as distance function only use fragments of the objects for the similarity calculation between them (e.g. [150, 87]). This can be useful for image retrieval and classification, where only small parts (not a subset of features) of two images can yield to perception of similarity, but is not applicable for object representations in general.

Another class of non-metric distance learners are Bayesian Learners as used in [112], which are also designed for the special case of object recognition in images. In this work, we do not want to restrict similarity to images, but rather present a more general view on a broad range of applications.

3.6 Experimental Evaluation

In this section, we present the results of our experimental evaluation. As comparison partner we selected the methods that are closest to our approach: Relevant Component Analysis (RCA) and Fisher Faces (FF). Let us note that RCA requires only chunks of data objects having the same class and no explicit class set. However, since we used datasets having class labels, we provided RCA with the complete set of training objects for each class as a chunk. Furthermore, we compared Bayes Estimate Distance (BED) to the standard Euclidean distance (Eucl) to have a baseline method. We evaluated all methods on several real-world datasets to test their performance for classification and retrieval tasks. All methods were implemented in Java 1.6 and tests were run on a dual core (3.0 Ghz) workstation with 2 GB main memory.

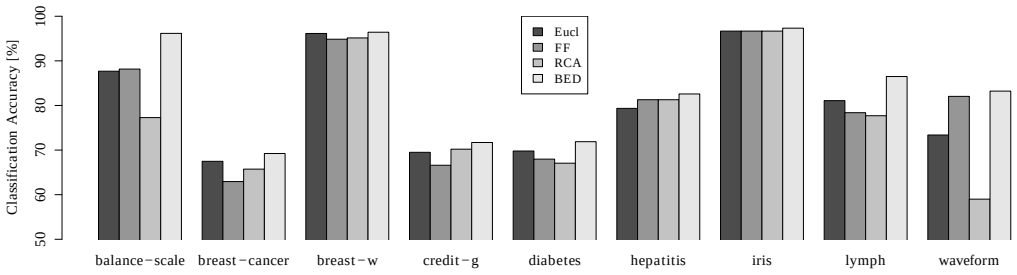


Figure 3.3: NN-Classification results on several UCI datasets.

3.6.1 Nearest Neighbor Classification

As mentioned before, our similarity learner can be applied for different applications. A first, well-established method is improving the quality of nearest neighbor classification. For the classification task, we used several datasets from the UCI Machine Learning Repository [119]. Evaluation on the datasets was performed using 10-fold cross-validation and all 4 distance measures were used for basic nearest neighbor classification. To train BED, we employed sampling based on the target neighbors. In other words, we took the difference vectors of all training objects to the k-nearest neighbors within the same class and the k-nearest neighbors in all other classes. To find a suitable value for k, we screened over a small set of numbers between 5 and 20s.

The results for k-NN classification are shown in Figure 3.3. BED displays the largest accuracy in all 9 datasets. We observe strong variations in the relative accuracies of the FF and RCA approach w.r.t. Eucl and BED. BED leads to classification results which are up to 8% better than the best of Eucl, FF and RCA. Thus, we can state that BED can improve the results of instance-based learners.

3.6.2 Precision and Recall Graphs

We employed two image datasets for testing the performance of our new distance measures for retrieval applications. The *Conf* dataset was created by ourselves and contains 183 images of 35 different motives. Please refer to Section 4.5.2.2 for a sample of the contained images. The *Flowers* dataset was introduced in [122] and consists of 1 360 images of 17 different types of flowers. From these two datasets, we extracted color histograms (based on the HSV color space), facet features [31] and Haralick features [79]. The characteristics of the resulting feature datasets can be seen in Table 3.1.

We measured the retrieval performance on these datasets using Precision-Recall graphs. We posed a ranking query for each image and measured the

Table 3.1: Image Retrieval Datasets

Dataset	Instances	Attributes	Classes
Conf-hsv	183	32	35
Conf-facet	183	24	35
Conf-har	183	65	35
Flowers-hsv	1360	32	17
Flowers-facet	1360	24	17
Flowers-har	1360	65	17

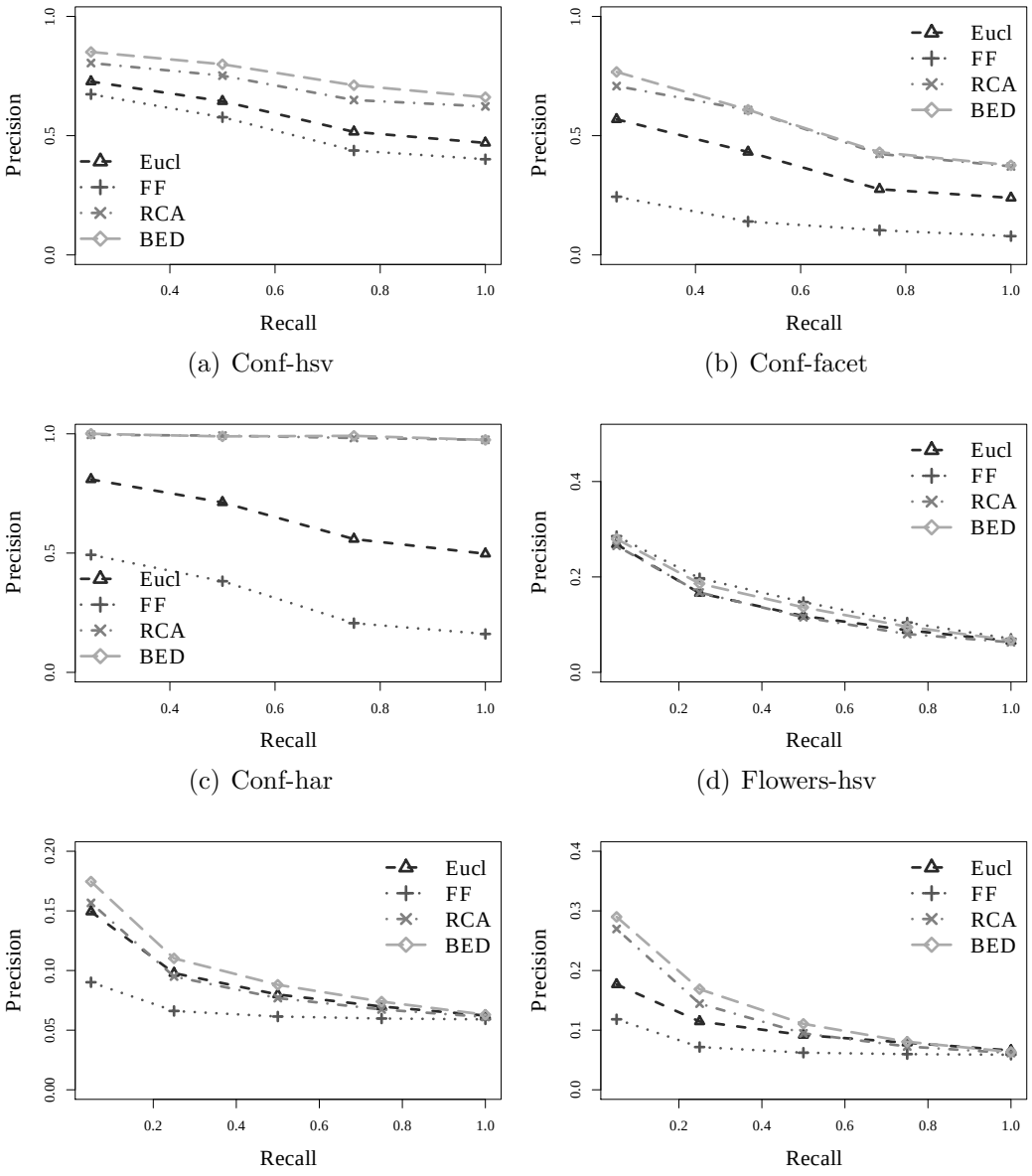
precision of the answer resulting from the remaining database for several levels of recall. In the retrieval task, we employed very large numbers of difference vectors for training, to adjust BED to achieving reasonable precision values for large levels of recall.

On the *Conf* dataset, BED shows an impressive boost of the retrieval quality using hsv-color-histograms (Figure 3.4(a)), while it still leads to slightly better results using facet or Haralick features (see Figures 3.4(b) and 3.4(c)) in contrast to RCA. FF does not appear to be well-suited for these datasets, as it performs even worse than the Euclidean Distance. On the Flowers dataset, retrieval quality can again be improved by BED when using Facet and Haralick features respectively (see Figures 3.4(e) and 3.4(f)). On the feature dataset consisting of the hsv-color-histograms of *Flowers*, Fisherfaces lead to a better Precision-Recall-Graph (Figure 3.4(d)) than the other approaches. Note, however, that this is the only retrieval experiment where FF performed better than the Euclidean distance. Thus, we can state the BED is suitable for retrieval tasks as well as for data mining tasks.

3.6.3 Examination of the BED Components

In our last experiment, we examine the performance of BEDs compared to their separated components. We trained BEs on the original dimensions (only BE) of the feature space. Furthermore, we wanted to find out whether the learned eigenvalue decomposition can be used for learning a Mahalanobis distance improving classification results. To create such a transformation, we additionally multiplied each eigenvector w by its inverse eigenvalue. The comparison was performed for several retrieval datasets which all displayed similar results. An example precision-recall graph of the *Conf-hsv* dataset is presented in Figure 3.5

Using the BED without the rotation still increases the retrieval performance compared to the plain Euclidean distance on the same feature space.



(a) Conf-hsv (b) Conf-facet

(c) Conf-har (d) Flowers-hsv

(e) Flowers-facet (f) Flowers-har

Figure 3.4: Precision-Recall graphs on the Conf and Flowers dataset.

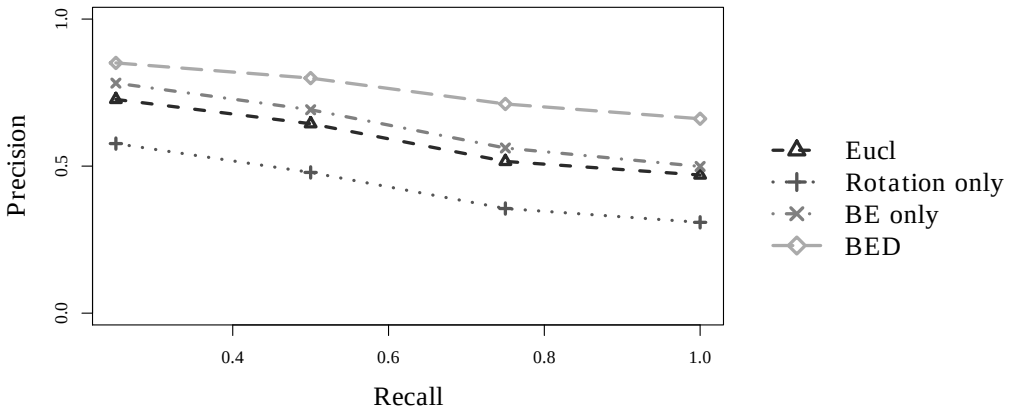


Figure 3.5: Different versions of BE on Conf-hsv.

Thus, even without an affine transformation, the BED is capable of improving the retrieval quality. A second very interesting result is that the rotation component of BEDs does not yield any performance advantage when used as Mahalanobis learner. Though the learned directions do optimize the BEs being observed in the new dimensions, they seem to be unsuitable for improving the results obtained by the Euclidean distance.

3.7 Summary

This chapter has introduced Bayes Ensemble Distance (BED) as an adaptable dissimilarity measure. BED is applied to the difference vector of two feature vectors. For each dimension, BED independently determines the likelihood that both objects are dissimilar employing a simple Bayesian learner called Bayes Estimate (BE). The results of the BEs are combined by computing a weighted average prediction.

That way, the derived similarity score is less dependent on outlier values in some of the dimensions. Since BED is dependent on the spatial rotation of the data space, it is possible to optimize the vector space in order to derive a feature space allowing the training of more descriptive and independent BEs.

In our experimental evaluation, we have demonstrated that BEDs can largely increase the classification accuracy of instance-based learning. Additionally, we have demonstrated the suitability of BED for retrieval tasks.

For future work, we plan to investigate efficiency issues when using BED for information retrieval in order to actually use it in medical image retrieval. Furthermore, we plan to apply the idea of BEs to structured objects like graphs. Finally, as by far not all feature differences are normally distributed,

we would like to explore alternative distribution functions on their applicability in our proposed learning scheme.

Chapter 4

Multi-Instance Distance Measures

Images are often represented by a set of representative regions or hotspots. The elements of such a set usually are described by the same feature type, only representing different excerpts of an image. As the number of represented regions is mostly unknown for a new image, this poses a special challenge for any similarity or distance measures applied on a database of images represented by sets of sub-features.

An object representation consisting of a collection of data objects is called a multi-instance object. Prominent examples for this feature type are interesting-point or salient point procedures like SIFT [106] and SURF [10]. They automatically discover a set of likely recognizable coordinates within a given image and generate multi-dimensional real-valued feature vectors for each point.

For other applications, images are first segmented (either manually or (semi-) automatically) into various image subregions, which again are collected as an unordered set of sub-image representations. [116] Chapter 6 will introduce a use case of medical image retrieval, where an object consists of a sequence of 2D images, each of which is represented by the same type of image descriptors. The corresponding query scenario consequently employs a multi-instance view on the examined query objects.

It was therefore necessary to investigate this special group of object representations for suitable distance measures and ways of improving query processing times.

4.1 Introduction

We will first clarify the notation and terms used in this chapter.

Definition 20 (Multi-Instance Object, Instances) *A* multi-instance object $A \subseteq \mathcal{F}$ *on a domain* $\mathcal{F}$ *is a collection of data objects* $a_i \in \mathcal{F}$ *of the same type. The objects* $a_i \in \mathcal{F}$ *are called the* instances *of* A*,* $i \in \{1, \ldots, |A|\}$*, with* $|A|$ *being the number of instances in* A*. In favor of a simplified notation, the index* i *will be omitted if the multi-instance relation* $a_i \in A$ *is evident.*

Both, instances and multi-instances are thus objects, however, the instances usually only represent part of an object.

Definition 21 (Distances) *A distance function* $d_I : \mathcal{F} \times \mathcal{F} \mapsto \mathbb{R}$ *representing the degree of dissimilarity between two instances in* $\mathcal{F}$ *is denoted as an* instance distance measure*. A distance function* $d : 2^{\mathcal{F}} \times 2^{\mathcal{F}} \mapsto \mathbb{R}$ *between two multi-instance objects* $A, B \subseteq \mathcal{F}$ *is a* multi-instance distance measure*.*

In the following, we will assume that d_I is symmetric, i.e. $\forall\ a, b \in \mathcal{F} : d_I(a, b) = d_I(b, a)$.

4.2 Combination of Instance Distances

In this setting, the inter-instance distances are the only information available, thus for this thesis, all multi-instance distance measures will be combinations of distances among the objects' instances.

4.2.1 Average Linkage (AvgLink)

A straightforward way to measure the distance between two sets is the mean of all pairwise instance distances, also known as *Average Linkage*:

$$d_{\text{Avg}}(A, B) = d_{\text{AL}}(A, B) = \frac{\sum_{a \in A} \sum_{b \in B} d_I(a, b)}{|A|\,|B|} \tag{4.1}$$

This distance measure is appropriate when all instances within an object are equally important and all instances are drawn from the same distribution. For more complex objects, which may be based on several distributions, this approach is likely to fail: low instance distances among similar instance groups of two multi-instance objects are bound to be cancelled out by the higher cross-distribution distances. Thus, even though the two objects have similar subsets, they will be considered to be dissimilar.

4.2.2 Minimum Distance (MinDist)

For some kinds of data, only small subsets of all the available instances may be useful at all. This leads us to the other extremal case, where we completely focus on the minimal distance between all pairs of instances, the *minimum distance* or *single link*:

$$d_{\text{MinDist}}(A, B) = d_{\text{SL}}(A, B) = \min_{a \in A, b \in B} d_I(a, b) \tag{4.2}$$

This approach is strongly prone to noise, since in the end, the distance value depends on one instance pair only. Furthermore, it is not well suited for the comparison of objects which consist of several characteristic distributions.

4.2.3 Half the Sum of Minimum Distances (HMD)

In order to attenuate the dependency on one particular instance per object, we can average over the minimum distances of every instance in A to the instances in B, thus receiving

$$d_{\text{HMD}}(A, B) = \frac{1}{|A|} \sum_{a \in A} \min_{b \in B} d_I(a, b) \ . \tag{4.3}$$

This distance measure is the asymmetric combination of all minimum linkages of A to B. Even if symmetry is not required, e.g. when searching for the most similar objects B in a database D, this approach still suffers from variance in the size of the objects in D – more instances raise the probability of a close link, fewer instances lower the probability.

4.2.4 Sum of Minimum Distances (SMD)

The solution to the symmetry problem is the *Sum of Minimum Distances* (SMD) which combines both directions of the HMD (hence the bulky name: *half-SMD*).

$$d_{\text{SMD}}(A, B) = \frac{1}{2} \left(d_{\text{HMD}}(A, B) + d_{\text{HMD}}(B, A) \right) \ . \tag{4.4}$$

Note that this differs from another definition of SMD, frequently used in the literature, [98, 42]

$$d_{\text{SMD alt.}}(A, B) = \frac{1}{|A| + |B|} \left(\sum_{a \in A} \min_{b \in B} d_I(a, b) + \sum_{b \in B} \min_{a \in A} d_I(a, b) \right) \ , \tag{4.5}$$

by allowing the same distance contribution to both objects instead of maintaining a disequilibrium between the two distance sets.

4.2.5 Hausdorff Distance (HD)

Another MinDist-related measure is the Hausdorff distance. It is defined as the larger of the two maximum minimum links between two datasets, i.e. the instances of two multi-instance objects.

$$d_{\mathrm{HD}}(A,B) = \max\{\max_{a\in A}\min_{b\in B} d_I(a,b), \max_{b\in B}\min_{a\in A} d_I(a,b)\} \tag{4.6}$$

Another possibility is to take only the maximum distance of query object A to a candidate object B in the database:

$$d_{\mathrm{HD\ asym.}}(A,B) = \max_{a\in A}\min_{b\in B} d_I(a,b) \tag{4.7}$$

This family of distance measures is well-suited for compact geometric objects. By identifying the weakest (i.e. largest) minimal link, we can better incorporate the dissimilarity of two objects than in SMD. However, if the objects are allowed to contain noise instances or if we are also interested in partial matches, these measures will always result in too large distances.

4.2.6 Convolution Distance (CD)

The notion of similarity is frequently associated with kernel methods. A direct way of using kernel functions $k : \mathcal{F}\times\mathcal{F} \mapsto \mathbb{R}^+$ for multi-instance distance calculations is the use of a convolution:

$$d_{\mathrm{CD}}(A,B) = 1 - \frac{\left(\sum_{a_i\in A, b_j\in B} k(a_i,b_j)\right)^2}{\left(\sum_{a_i,a_j\in A} k(a_i,a_j)\right)\left(\sum_{b_i,b_j\in B} k(b_i,b_j)\right)} \tag{4.8}$$

It results in a distance value in $[0,1]$ which makes it attractive for further processing. Moreover, it can be flexibly fine-tuned by selecting the best-suited kernel function for a given problem.

4.2.7 Maximum Mean Discrepancy (MMD)

Another kernel-based distance measure is the Maximum Mean Discrepancy (MMD). [71] It reduces the distance problem among sets to the difference between the mean instance values, possibly in a different feature space. Using

a kernel function $k : \mathcal{F} \times \mathcal{F} \mapsto \mathbb{R}^+$, the MMD can be written as:

$$d_{\mathrm{MMD}}(A, B) = \left[\frac{1}{|A|^2} \sum_{a_i, a_j \in A} k(a_i, a_j) - \frac{2}{|A|\,|B|} \sum_{a_i \in A, b_j \in B} k(a_i, b_j) + \frac{1}{|B|^2} \sum_{b_i, b_j \in B} k(b_i, b_j) \right]^{\frac{1}{2}} \tag{4.9}$$

This method can be applied on two-sample problems, where the distributions of two sample sets are to be analyzed and may thus be considered to be an approximation of the Average Linkage distance. However, this approach is prone to carry the same disadvantages mentioned earlier in Section 4.2.1 when dealing with objects formed from several distributions.

4.3 Instance Weighting Methods

In all distance measures introduced so far, every instance within an object is allowed the same contribution to the overall distance. For some objects, however, we may be able to identify instances of higher and lower importance for the comparison to other objects. These notions of importance may be converted into weights, which can be incorporated into the previously-mentioned distance measures.

Definition 22 (Instance Weights) *We call a multi-instance object $A \subseteq \mathcal{F}$* weighted*, if it is has assigned an* instance weight *$v_i \in \mathbb{R}$ (with $v \in \mathbb{R}^{|A|}$) to every instance $a_i \in A$.*

For convenience of notation, we will denote the instance weights of a second multi-instance object $B \subseteq \mathcal{F}$ as $w \in \mathbb{R}^{|B|}$.

4.3.1 Weighting Distance Measures

The direct way to use instance weights is to include them into the chosen multi-instance distance measure.

4.3.1.1 Weighted Average Linkage

Including weights into the average linkage distance (Section 4.2.1) can be achieved by simply extending the instance distances by their instances' weights

and by then normalizing by the sum of used weighting combinations:

$$d_{\mathrm{WAvg}}(A,B) = \frac{\sum_{i=1}^{|A|}\sum_{j=1}^{|B|} v_i w_j d_I(a_i,b_j)}{\sum_{i=1}^{|A|}\sum_{j=1}^{|B|} v_i w_j} \tag{4.10}$$

This weighting procedure should assert a fair contribution of each instance to the global distance in accordance with its weight.

4.3.1.2 Weighted Convolution Distance

The inclusion of weights in the convolution distance also can be done in a very straightforward way:

$$d_{\mathrm{WCD}}(A,B) = 1 - \frac{\left(\sum_{a_i\in A, b_j\in B} v_i w_j k(a_i,b_j)\right)^2}{\left(\sum_{a_i,a_j\in A} v_i v_j k(a_i,a_j)\right)\left(\sum_{b_i,b_j\in B} w_i w_j k(b_i,b_j)\right)} \tag{4.11}$$

For other multi-instance distance measures, such an inclusion of weights is less evident.

4.3.1.3 Weighted HMD / SMD

There are basically two ways of weighting the sum of minimum distances: only taking the weights of the query-instances looking for their minimum distance partners

$$d_{\mathrm{WHMD}_1}(A,B) = \frac{\sum_{a_i\in A} v_i \min_{b_j\in B} d_I(a_i,b_j)}{\sum_{i\in 1}^{|A|} v_i}\ , \tag{4.12}$$

or combining the weights of both of the minimum distance pairs' partners:

$$d_{\mathrm{WHMD}_2}(A,B) = \frac{\sum_{a_i\in A} \min_{b_j\in B} v_i w_j d_I(a_i,b_j)}{\sum_{(i,j)\text{: MinDist pairs}} v_i w_j}\ . \tag{4.13}$$

For the second variant, in many cases, the weights w of the second multi-instance object B may suggest different minimum distance pairs than the optimum assignment. Consequently, for the above distance definition, we are

confronted with a third distance variant: the minimum distance partner $b_{j,\text{MIN}}$ can be determined without the inclusion of any weights, only using the weight for forming the overall distance:

$$d_{\text{WHMD}_{2'}}(A,B) = \frac{\sum_{a_i \in A} v_i w_{j,\text{MIN}} \min\limits_{b_j \in B} d_I(a_i, b_j)}{\sum\limits_{(i,j)\text{MinDist pairs}} v_i w_j} \,. \tag{4.14}$$

This alternative, however, is probably counter-productive, since the choice of the minimum distance partner can be influenced by noise. The idea of weights is to dispose of meaningless pairs. Thus, we should choose the minimum partners in correlation with the overall distance measure and discard variant (4.14).

4.3.1.4 Obstacles in Distance Weighting

The combinations of distance measures with instance weights introduced so far raise the question about how to balance the weights' distribution over the objects. The straightforward approach of normalizing the weight vector w to sum to 1 is bound to result in a bias of lower distances to objects with a larger number of instances. Not normalizing it at all calls for a balanced way of proceeding as with weighted average linkage.

The most important question is: how do we determine the weights? Does a large weight generally indicate a stronger influence of an instance than a low weight? In the cases of average linkage and the convolution distance, this is definitely true. In case of a distance measure using minimum distances like the SMD, one should rather invert this connotation in order to not disturb the beneficial minimal linkage effect.

4.3.2 Using Instance Weights for Instance Selection

All the difficult questions of weight inclusion can be omitted, if we decide to only use the weights as an instance filter. We discriminate between 3 ways of weight-based instance selection:

1. *Global Thresholding*: We set a global weight threshold t to a value in the range of the actual instance weights and discard any instance below (or, if the lower weights are perceived as meaningful, above) t.

2. *Frequency-based Thresholding*: An object's instances' weights are treated as a ranking, and only the top t, or, to be more flexible, the top $t\,\%$ instances are included in the ensuing distance measures.

3. *Object-adapted Thresholding*: The instance weights of an object are analyzed by a given probabilistic statistic and this statistic determines the appropriate weight threshold t, depending on the weights' distribution.

Global thresholding requires the weighting method to return globally comparable weights. Thus, size-dependent-statistics like the Wilcoxon Test (see Section 4.3.3.1) are not applicable. Furthermore, we must assert that for every multi-instance object, we can retain at least one instance, else it will be impossible to match this object to any other object.

This requirement is fulfilled by frequency-based thresholding. The main question here is: do we want a fixed number of instances or a fixed fraction? The first approach obviously opens possibilities for easier object handling, when all objects have the same number of instances. However, for many objects, the number of instances is a helpful feature itself, and ought not to be discarded. This observation speaks in favor of the fractional approach, selecting the top $t\,\%$ instances.

Object-adapted thresholds can be defined on a variety of statistics. The fractional approach is just one of them, comparable to a quantile-based selection procedure. Other possibilities include gradient-based thresholds or Gaussian mixture models. Such an approach can also be tightly coupled with the process of weight generating.

4.3.3 Instance Weight Computation

The importance of an instance for its object depends on its ability to discern between similar and dissimilar objects as well as on its affinity to the other instances of the object. The first property can be judged by statistical tests, measuring how often the current instance is consistent with the overall object similarity.

The other requirement is more difficult to meet. If an object consists of many, highly distinctive instances, we only need some of them in order not to over-estimate the similarity between two well-described objects. If, on the other hand, an object only consists of barely helpful instances, we still cannot afford to discard them all. We may even have to artificially assign higher weights than for instances of clearly described objects.

4.3.3.1 Wilcoxon Rank Sum Test

The Wilcoxon rank sum test is a non-parametric test on whether or not two samples $X, Y \subseteq \mathcal{O}$ originate from the same distribution on ordinal values $\in \mathcal{O}$, i.e. values that can be compared. Originally proposed by [170] in 1945 for equally-sized samples, it was extended to unbalanced samples [109] in 1947.

The null hypothesis, stating that X and Y are equally distributed, is verified by a t-test on the two samples' ranks, that is: their positions in the samples' sorted concatenation of size $|X| + |Y|$. This transformation has the great advantage, that the samples do not have to be normally distributed and the whole test is less prone to outliers.

In our instance weighting approach, we define X and Y to be two instance distance distributions in $\mathbb{R}$. In the interest of covering the more meaningful distances, we focus on minimum instance distances which have proven to be a valuable component in the more successful multi-instance distances. For any instance to be weighted, X contains instance distances of *similar* instances and Y contains instance distances of *dissimilar* instances. The decision of which objects are similar is supervised on a database of multi-instance objects labelled with class attributes: multi-instance objects of the same class ($\in \mathcal{C}^+$) are considered to be similar, multi-instance objects belonging to another class ($\in \mathcal{C}^-$) are supposed to be dissimilar.

An instance $a_i \in A \subseteq \mathcal{F}$ is then rated according to the minimum instance distance distribution X of objects of the same class and the distribution of minimum instance distances to objects of other classes Y:

$$X = \left\{ \min_{b_j \in B} d_I(a_i, b_j) \mid B \in \mathcal{C}^+ \setminus A \right\} \tag{4.15}$$

$$Y = \left\{ \min_{b_j \in B} d_I(a_i, b_j) \mid B \in \mathcal{C}^- \right\} . \tag{4.16}$$

The decision value for a_i is a p-value on the divergence of the mean rank of the distances of the smaller set of X and Y, from the expected rank

$$\frac{1}{2} \min\{|X|, |Y|\} \cdot (|X| + |Y| + 1) \tag{4.17}$$

according to the normal distribution. Due to the ranking procedure, these p-values are comparable among each other for instances belonging to the same class. An inter-class comparison of p-values should take into account the dependency of the p-value to the sample sizes.

A major drawback of the Wilcoxon test is its independence from scale. It is helpful for dealing with outliers, however, instances with very low minimum distances get the same p-value as instances with minimum distances which are multiple times higher than the previous values but happen to be ordered similarly. Additionally, the assumption that an instance must have at least one suitable minimum distance partner in any other multi-instance object of the same class in order to be meaningful, is again a strong constraint on the nature of the given dataset.

4.3.3.2 Pearson Correlation

Inspired by a feature selection approach introduced in Section 2.4.2.1, we also assessed the importance of an object's instance via the class-based Pearson Correlation. The covariance is a measure for linear dependence of two distributions $X, Y \subseteq \mathbb{R}$ with expected values μ_X, μ_Y:

$$\mathbf{Cov}(X, Y) = \mathbf{E}\left((X - \mu_X)(Y - \mu_Y)\right) \ . \tag{4.18}$$

It can be seen as a combined linear variance measure of two distributions. If A and B are independent, their values' strongest divergences from their means will not coincide, thus cancel out: $\mathbf{Cov}(A, B) = 0$. A positive covariance tells us, that A and B are correlated in the same direction.

If we normalize the covariance by the two distributions' standard deviations σ_X and σ_Y, we receive the *Correlation Coefficient*

$$\rho_{X,Y} = \frac{\mathbf{Cov}(X, Y)}{\sigma_X \sigma_Y} \ , \tag{4.19}$$

which scales in $[-1, 1]$. We still have $\rho_{X,Y} = 0$ for independent distributions, but now we can compare several correlations without having to re-scale.

The estimate for the correlation of two samples $X = \{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^n$ and $Y = \{y_1, y_2, \ldots, y_n\} \in \mathbb{R}^n$ is the *Pearson Correlation Coefficient*:

$$\hat{\rho}_{X,Y} = \frac{\sum_{i=1}^{n}(x_i - \mu_X)(y_i - \mu_Y)}{\sqrt{\sum_{i=1}^{n}(x_i - \mu_X)^2 \sum_{i=1}^{n}(y_i - \mu_Y)^2}} \ , \tag{4.20}$$

where $\mu_X = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\mu_Y = \frac{1}{n}\sum_{i=1}^{n} y_i$.

For testing whether or not the minimum distances of an instance a_i to its own class $\mathcal{C}^+$ are lower than the minimum distances to objects of the other class $\mathcal{C}^-$, we can use an approach popular in feature selection. [75] We concatenate those distances and test their correlation to a vector containing 1 for the distances of $\mathcal{C}^+$ and -1 for the distances of $\mathcal{C}^-$.

A resulting value close to -1 is then an indicator of a strong negative correlation of the minimum distances with its class indicators and thus a useful instance.

4.3.3.3 Other Weighting Strategies

In addition to the weighting schemes proposed above, we also tested a number alternative weighting approaches.

Entropy Entropy-based criteria are frequently used in feature selection. [38, 75] Usually, it involves the discretization [49] of the possible feature values

and the entropy calculation on the class-wise probability per discretized value. In our tested setting, we discretized the sets of nearest-neighbor or ϵ-range instances into instances resulting from the same class and instances belonging to another class.

Good Count Triplets [65] suggested a method for faster weight calculation by restricting the training data set to the closest objects within and out of the class of the object to be trained. For training an object A, they first generate triplets (A, B^+, B^-), containing close objects of A of the same and close objects of another class, and perform their optimization procedure on the set of resulting minimum distance pairs per instance a_i: $\big(\min_{b_j \in B^+} d_I(a_i, b_j), \min_{b_j \in B^-} d_I(a_i, b_j)\big)$. A refined optimization problem is formulated in [64].

A simple way of generating an instance weight for a_i from those triplets is to raise a counter $c(a_i)$ by one, whenever the minimum instance distance of the within-class partner is lower than that of the outer class partner. Else, the counter is lowered by one. The resulting weight is then $\frac{c(a_i)}{\text{number of triplets}}$, a value in $[-1, +1]$, categorizing a_i the more useful the closer the weight is to $+1$.

Triplet Distance Differences The same triplets can also be used for determining the average distance difference of within-class distances with outer-class distances. This distance can be directly used as a pruning weight in instance selection, since it defines a ranking over the instances of a multi-instance object. Using it for weighted distances in cross-object comparison, requires an additional normalization.

Difference Among Average Distances Another variant is to take an instance's average minimum distance avg_{d^+} to objects of the same class $\mathcal{C}^+$ and compare it to the average minimum distance avg_{d^-} to objects of the complement $\mathcal{C}^-$. This alone should return comparable results as the above method. It might, however, be further improved by including the distance variances into the test: An instance may be accepted, if

$$\text{avg}_{d^+} + \text{stdev}_{d^+} < \text{avg}_{d^-} - \text{stdev}_{d^-} \ , \tag{4.21}$$

where stdev_{d^+} is the standard deviation of minimum distance pairs within the same class and stdev_{d^-} is the standard deviation of minimum distance pairs drawn from other classes, respectively.

4.3.4 Limits of Supervised Instance Weighting

The described instance weighting approaches manage to define a measure of reliability to the instances of an object with a known class affiliation. Such a measure could be used as an interesting tool for the visual interpretation of a multi-instance object derived from an image. However, the scope of this work is to capture the similarity of objects with a so far unknown classification. Unfortunately, multi-instance objects proved to be complicated w.r.t. weighting instances in an unsupervised or semi-supervised way.

When computing weights in the proposed supervised fashion, the training error is drastically decreased for any of the weighting approaches of Section 4.3.3 for both the distance weighting schemes of Section 4.3.1 and the instance selection variants of Section 4.3.2. In addition to a very good adaptation to the training dataset, the tested weighting procedures provide a good interpretability of the instances' relevances.

However, in a real-world test case, the class information will not be available for the query object. Therefore, we cannot follow the same instance weight computation strategy for the multi-instance query object as for the objects stored in the training database.

We tested three solutions for generating weights for the query object:

Constant Weights In the first setting, we kept the weights of the query instances at a constant value. The choice of the constant depends on whether or not the instance weights of the training set have been normalized. When using weight-induced instance selection, of course, no instances can be discarded from the query object.

Intermediate Classification The second setting generates an intermediate class label for the query object by using the training set. This may happen either without using any weights at all, or by using weights generated with the *Constant Weights* approach sketched above. After the intermediate class label is available, the instance weights are generated by pretending it to be the ground truth.

Multi-Hypothesis Vote As the above approach is very dependent on a correct intermediate class label, we also tested a variant forming a consensus classification. We generate instance weights for all possible class hypotheses on the query object and keep the set of instance weights for the hypothesis with the strongest indication of relevance. The decision on relevance can either be generated from the weights if they have not yet been normalized, or from a confidence measure on the ensuing classification of the query object. This approach only works for a finite set of discrete class labels.

Unsurprisingly, the worst results were generated by the intermediate classification approach, which has a strong tendency to emphasize the very bias effects which we are trying to remove by using instance weights. By deciding between multiple hypothesis, this problem should be avoided. Yet, we observed that the best-performing weighting strategy was the use of constant query weights. Obviously, when in doubt, multi-instance query weight labelling should rather be omitted than artificially forced.

This conclusion alone would not harm the whole concept of instance weighting. However, the overall results of the constant weights approach were hardly any better than those of the basic, untrained multi-instance distance measures, and they were extremely dependent on the dataset. The most plausible explanation for this observation is that the datasets we used for testing were too small for obtaining weights which are sufficiently regularized from the machine learning point of view. However, also exemplary tests on larger datasets did not lead to the expected improvement.

After an exhaustive number of experiments on various test sets and a multitude of strategy configurations, we therefore closed the research efforts in this direction and investigated alternative methods of multi-instance distance measures.

4.3.5 Other Methods for Instance Selection

Instance selection does not necessarily have to be based on instance weighting. Converting the multi-instance problem into another problem oftentimes results in the implicit selection of instances.

4.3.5.1 k-SMD

The sum of minimum distances can be further refined by restricting the number of minimum links to the k best, i.e. the k-lowest minimum distances. This turns the HMD introduced on page 69 into the k-HMD:

$$d_{k\text{-HMD}}(A,B) = \text{avg}\{d_{\text{MIN}}(a_i,B) \mid a_i \in A \wedge d_{\text{MIN}}(a_i,B) \leq k\text{-minimum distance of an } a_j \in A \text{ to } B\} \; . \tag{4.22}$$

This asymmetric distance measure can then be combined to a symmetric variant as in Equation (4.4), the k-SMD.

As a side effect, this transformation of using only the k-minimum distances per object allows to treat the multi-instance problem as a multi-represented, or multi-modal problem, which can be tackled by a series of alternative distance measures such as MUSE [97].

The advantage of k-SMD is its flexibility w.r.t. noise instances, which usually will provide larger distance contributions. However, only taking the k lowest minimum distance pairs cannot ensure that all relevant distributions forming a multi-instance object are represented. In addition, we now have to deal with a parametrized distance measure.

4.3.5.2 MILES

Another transformation was used by [29] in MILES (Multiple-Instance Learning via Embedded Instance Selection). Their approach converts each instance of a training database D_{train} into a real-valued feature for all objects in D_{train}, and it applies a specialized learning algorithm on the resulting feature vectors. These new feature vectors can, of course, also be treated by a variety of conventional learning algorithms, such as an SVM or a common k-nearest neighbor classifier. The instance-feature conversion used in [29] is to derive a probability $P(x|A)$ for every instance $x \in \mathcal{F}$ occurring in D_{train}, that it represents a multi-instance object A based on a classical radial basis function (RBF)

$$P(x|A) = \max_{a_i \in A} \exp\left(-\frac{d_I(a_i, x)^2}{\sigma^2}\right) , \tag{4.23}$$

where σ is a a predefined, constant scaling factor.

Our experiments on MILES showed fine training errors which could be converted to comparatively stable and good cross-validation test errors. When, however, excluding the query objects' representations from the training set, thereby only representing the query objects by foreign instance representatives, we ran into similar problems as with the instance weighting approaches proposed in Section 4.3.

A related, but kernel-based learning approach was proposed by Gärtner *et al.* in [66]. With *MissSVM*, [184] convert the problem of multi-instance distance learning to a semi-supervised learning problem, which is solved via a modified SVM formulation. It is, however, restricted to 2-class problems and does not perform best among its competitors. MILES, for instance, always scores better.

4.3.5.3 Integrated Region Matching (IRM)

Another variant of instance selection is not to select the instances themselves but a set of instance pairs between the multi-instance objects to be compared. In [161], Wang *et al.* describe Integrated Region Matching (IRM), a multi-instance distance measure especially-designed for images represented by a set of segmentations.

They first use the relative size of the image segments as image-driven instance weights v_i and w_j for the two objects A and B. Then, they iteratively generate a matrix of significance values $s_{i,j}$ of all instance pairs (a_i, b_j) as a combination of the corresponding instance weights and the actual instance distances $d_I(a_i, b_j)$. In order to avoid comparisons between instance pairs which are not relevant, many of the instance pairs are labelled with a significance value of 0, i.e. the resulting significance matrix is sparse. The final distance measure is then a variant of the weighted average linkage distance:

$$d_{\mathrm{IRM}}(A, B) = \sum_{i=1}^{|A|} \sum_{j=1}^{|B|} s_{i,j} d_I(a_i, b_j) \ . \tag{4.24}$$

In the datasets we tested so far, we never had additional information like the relative segment size which could be exploited to infer image-driven instance weights. Therefore, this thesis will not present any tests on the IRM distance measure. For an example of a successful application in an image retrieval problem, see Nascimento *et al.* [116].

4.4 Indexing-based Distance Measures

Since all of the distance measures mentioned so far require at least a quadratic runtime in the number of instances, for a large number of object comparisons as required by database-retrieval queries, a speed-up of distance computation becomes necessary.

4.4.1 Single-object Indexing

If a type of multi-instance objects generally consists of a large number of instances, it can be a good idea to index one or both of the multi-instance objects to be compared. Using a sped-up version of all-k-nearest-neighbor queries (e.g. as in [182] or our own approach in [53]) we can decrease the runtime for k-nearest-neighbor-based multi-instance distance measures such as the SMD or the Hausdorff distance. This advantage, however, is only effective, if the number of instances per object is really high. So far, we have not yet encountered a data set which actually needs objects of such a large size. Usually, they can be easily transformed into more compact data objects without loss of descriptiveness.

4.4.2 Instance Indexing

One solution suited for classification tasks is to simply index a database containing all instances of a multi-instance database. In the following, we introduce a strategy of combining multiple instance similarity queries for the instances of a query object into a useful classification scheme.

4.4.2.1 Average k-Minimum Linkage Classification

In this approach, we derive the k-nearest neighbors for all query instances and form a histogram H of the resulting instances' class labels. The winner class is then chosen as the predicted class label for the query object.

Algorithm 6 k-Minimum Linkage Classification

Input: multi-instance query object A, nearest neighbor parameter k, class-labeled instance database $D_I \subseteq \mathcal{F}$

1: $H \leftarrow$ empty histogram; ▷ Initialize a class-wise histogram
2: **for** instances $a_i \in A$ **do**
3: KNNs $\leftarrow \{b_i \in D_I \mid b_i$ is a $k-$nearest neighbor of a_i in $D_I\}$;
4: **for** $j = 1;\ j \leq$ KNNs.LENGTH; j++ **do**
5: RAISE$(H, b_j.\text{CLASS}, j)$; ▷ Account for this instance's match
6: **end for**
7: **end for**
8: $w \leftarrow$ NULL; $b \leftarrow -\infty$; ▷ Determine the winner class
9: **for** $i = 1;\ i \leq H.\text{LENGTH};\ i$++ **do** ▷ For all class weights
10: **if** $b < H[i]$ **then**
11: $b \leftarrow H[i]$; ▷ Update maximum weight
12: $w \leftarrow H[i].\text{CLASS}$; ▷ Update winner class
13: **end if**
14: **end for**
15: **return** w;

Output: The predicted class label w for A.

Algorithm 6 describes this procedure. The *average k-minimum linkage classification* is retrieved by setting function RAISE$(H, b_j.\text{CLASS}, j)$ in step 5 s.t. it raises the bin of H for the given class label $b_j.\text{CLASS}$ by one regardless of the ranking position j.

Databases consisting of unequally-distributed classes can be specially handled by weighting the histogram counts of H with the inverse class frequencies. When doing so, however, remember to also account for statistical effects of the k-parameter.

In this setting, all instances of the query or the instance database D_I are handled as equals. The main disadvantage is the complete loss of the multi-instance structure of the objects in the training database, however, the resulting classification experiments show that the accuracies are almost competitive to conventional multi-instance distances, whereas runtimes can be drastically decreased.

4.4.2.2 Geometric k-Minimum Linkage Classification

This procedure can be modified such that it also includes the instances' ranks into the decision finding process. In a new setting, we therefore change the $\text{RAISE}(H, b_j.\text{CLASS}, j)$ method of step 5 in Algorithm 5 s.t. it adds $\frac{1}{j}$ to the bin of $b_j.\text{CLASS}$ in H.

4.4.2.3 Global k-Minimum Distance Classification

Another way to predict the class of a given query object is to derive the k-nearest neighbors of any of the instances in the query to any of the instances in the database and to form the class histogram from these matches. Given a good index structure, this method is even faster than the two previous propositions, since the number of candidates which remain to be tested can be decreased with the number of instances which have already been tested.

This speed-up requires a ranking functionality of the index with a function GETNEXT(query, maxDist), which stops the retrieval of k-nearest neighbors as soon as the next-nearest neighbor's distance to the query is larger than a given maxDist. During every classification query, we know the global k-minimum distance for all instances queried so far, usually by using a priority queue, and we can use it to provide an ever decreasing bound for the maxDist parameter of the GETNEXT function. In each step of querying the next instance's k-nearest neighbors, we can therefore skip the call of retrieving the next-nearest neighbor whenever the instance's minimum distance partner is farther away than the currently k-best distance partner. When using a hierarchical tree-like structure like the X-Tree [15], this bound can already be effective at a high pruning level, therefore not requiring the exact distance computation to any candidates.

Note that if $k = 1$, this method is equivalent to a 1-nearest-neighbor classifier using the MinDist introduced in Section 4.2.2.

4.4.3 Alternative Accelerations on Multi-Instance Retrieval

We also explored alternative indexing approaches which maintained the multi-instance structure of the database objects. By carefully deriving lower and upper bounds for all encountered multi-instance candidate objects from the single-instance distance rankings of a multi-instance query object, we can compute a valid HMD ranking with a limited number of instance queries. This query framework requires some overhead for organizing an additional index, referencing any indexed instance's parenting multi-instance object. In addition, the retrieval strategy has to be optimized such that the complete multi-instance distance is computed instead of the instance-based bounds, whenever computing the bounds becomes more computationally expensive. All of these issues can be solved in a rather straightforward way.

In practise, we found the framework to be successful w.r.t. the early exclusion of candidate nearest neighbors. However, in order for the bounds to be effective, we needed to compute rather large single-instance rankings. The overall runtime advantage in comparison to a complete sequential scan on multi-instance objects was therefore marginal. On very large datasets, this framework provided a larger runtime gain, yet, the retrieval times were too high to be accepted in any real-world application.

The most common way of handling multi-instance objects in imaging is to use *visual words*. The bag-of-words approach originates in text retrieval and is based on counting word or term frequencies within a document and in a global context. Using normalization procedures like the *term frequency-inverse document frequency*, these retrieval approaches usually generate large and sparse feature vectors which can still be efficiently handled, e.g. using an inverted file structure.

With the increased use of salient-point-based features like SIFT [106] and SURF [10], this approach has been transferred to the imaging community by defining visual words. One bag of visual words represents a cluster of similar instances generated across a training dataset. Any multi-instance object (an image represented by various instances) is then represented by frequency counts of the clusters which are closest to the object's instances. [141]

In [123], Nistér and Stewénius proposed the vocabulary tree as an index structure for quickly retrieving objects with visual word characteristics similar to a query object. As the k-means clustering strategy behind this approach was not suitable for really large databases, we extended the vocabulary tree with the BIRCH [183] clustering method in a Diploma thesis. [80] The retrieval times showed a clear improvement w.r.t. previous visual word approaches, however, the accuracies of the tested SIFT and SURF descriptors were not satisfying.

Therefore, further work is needed for a publication of this otherwise successful strategy.

4.5 Experimental Evaluation

In our setting, we validate multi-instance distance measures by using them for classification of a multi-class database $D \subseteq \left(2^{\mathcal{F}}\right)^*$, which can be partitioned into disjoint sets $\mathcal{C}_1, \ldots, \mathcal{C}_c \subseteq \left(2^{\mathcal{F}}\right)^*$ of the classes 1 to c.

4.5.1 Classification Settings

The classification problem at hand is to predict the class of every object $A \in D$, such that it is equal to the real class of $A : C(A)$. Our experiments have been conducted via y repetitions of x-fold cross-validation. If not stated otherwise, $y = x = 10$. We have usually used a k-nearest-neighbor classifier.

4.5.2 Used Datasets

The multi-instance distance measures summarized in this survey have been tested on a variety of data sets.

4.5.2.1 Musk Datasets

The musk data sets have first been introduced in [46]. They consist of two 2-class problems on multi-instance data. They represent biological molecules via sets of 166-dimensional feature vectors. Objects of the positive class possess at least one instance each possessing a desired property. The classes of the musk1 data set are more balanced than the musk2 data set, which consists of more instances per object than the musk1 data. The datasets are summarized in Table 4.1.

A major flaw of this data set is its simplicity. Since one representing instance of the positive class is sufficient to indicate that the complete object is positive, this problem cannot be regarded to be a real multi-instance problem, but it is rather a special case of a one-class problem (in- and out-class).

4.5.2.2 Conf Datasets

The conf data set is an in-house collection of rather similar images, taken from slightly different viewpoints. Figure 4.1 shows a number of example images. Very similar objects are combined to form a total of 35 classes to conf35. These

Table 4.1: Summary of the musk1 and musk2 datasets.

Data set	$\lvert\mathcal{C}_1\rvert$	$\lvert\mathcal{C}_2\rvert$	# instances	$\frac{\#\text{ instances}}{\lvert\mathcal{C}_1\rvert+\lvert\mathcal{C}_2\rvert}$
musk1	45	47	476	5
musk2	63	39	6598	65

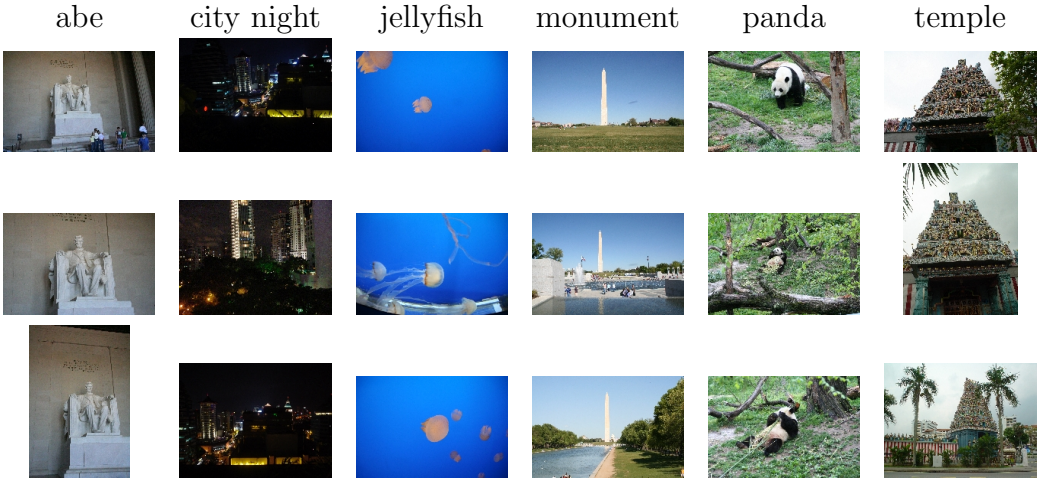


Figure 4.1: Example images of the conf dataset taken from 6 classes.

classes have also been summarized to form a more general class assignment of 8 classes (conf8).

For generating image descriptors, each image has been down-scaled to a maximal width or height of 200 pixels in order to be efficiently processed by the SIFT (Scale-Invariant Features Transform) Key point Detector by [106, 107]. This results in a set of so-called *interesting points* which are localized at distinctive regions of the image and are represented by a 128-dimensional feature vector describing the specific image region's gradient distribution. The number of interesting points (the instances) varies with the size and entropy of an image. Their distribution is outlined in Table 4.2.

Note that for color images, we usually observe very good results when using color-based image descriptors. Especially for this very homogeneous dataset, we have noted better classification results using simple color histograms with lower dimensionalities than the 128-dimensional SIFT descriptors. However, the goal of the investigations of this thesis was the application similarity search in medical images, which are mostly represented in grey-values. Therefore, we use SIFT descriptors for validating the distance measures introduced in this chapter.

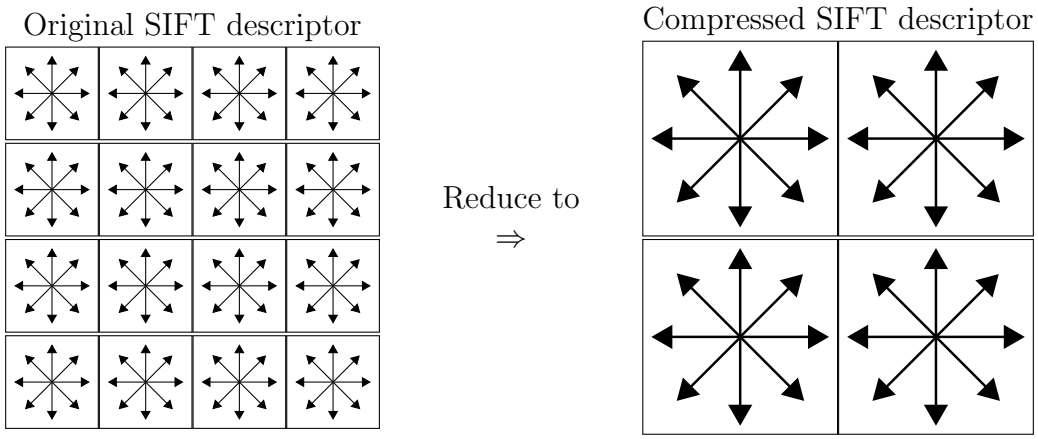


Figure 4.2: Reduction of SIFT descriptors from 128 to 32 dimensions: the gradient histograms represented by 8 values each of the 16 original image quadrants are summarized into 4 quadrants.

32D-SIFT Still, due to the high dimensionality of the SIFT descriptor's feature vectors, various attempts of dimensionality reduction have been made (e.g. [93, 10]). In a number of testing trials, we have discovered a low-loss reduction of the SIFT descriptor to 32 dimensions. The original descriptor recursively divides the image patch to be described into 16 sub-windows by first dividing the region of interest into 4 quadrants. Each sub-window is then described by an 8-dimensional Histogram of Gradients [36] which is being formed according to a Gaussian weighting kernel applied over the complete image patch.

We found that joining those 16 groups again such that they form back into the 4 original quadrants bears hardly any loss in matters of predictiveness. Thus, all image datasets can also be represented by 32-dimensional, summarized SIFT vectors. The compression step is illustrated in Figure 4.2.

4.5.2.3 Caltech Dataset

The Caltech 101 data set was first presented in [58]. It is an image dataset of 9 144 images ordered into 102 object categories presenting various objects. For the multi-instance experiments of this thesis, we restrict the tests on this dataset to a small subset easyCT, containing 100 images of 10 image categories. The complete dataset was only used for runtime tests, but never for a qualitative evaluation of multi-instance distance measures. The interested reader should note that since 2007, there is also a Caltech 256 dataset available, containing 30 607 images of 257 categories. [72] Besides an increased number of images, it poses an additional challenge by providing more versatile lighting

Table 4.2: Summary of the tested image datasets using SIFT descriptors (of 182 and 32 dimensions). c: number of classes, $|D|$: number of images.

Data set	$\|D\|$	c	$\frac{\|D\|}{c}$	# instances	$\frac{\#\text{ instances}}{\|D\|}$
conf35	183	35	5	38 589	211
conf8	183	8	23	38 589	211
easyCT	100	10	10	50 648	506
Stock4B	1 743	80	22	335 583	193

and image background conditions.

As the Caltech 101 dataset already contains rather small images of approximately 300 pixels width, we refrained from a further downscaling step before extracting SIFT features. Due to the smaller image extension of 200 pixels used in the conf dataset, the number of interesting points per image in the easyCT is more than doubled.

4.5.2.4 Stock4B Data Set

Another image data set was made available to the LMU by STOCK4B [172], an image stock agency in Munich. The Stock4B dataset consists of 1 743 images, loosely ordered into 80 categories. These categories, however, only represent a small fraction of 1 300 boolean attributes which have been assigned to matching images. The missing categories will not be taken into account for the experiments of this thesis.

We again derived SIFT features on images scaled to a maximum width or height of 200 pixels. Consequently, the number of interest points per image is comparable to the conf datasets.

4.5.3 Results

The following experiments display accuracy and runtime screenings for the basic distance measures introduced in Section 4.2 and the instance indexing approaches of Section 4.4.2. Since none of the instance weighting strategies introduced in Section 4.3 proved to be valuable in practical applications, the corresponding experiments are omitted.

4.5.3.1 Ranking Quality

The results of the experiments of the basic multi-instance distance measures are summarized in Table 4.3 (on page 90). The datasets musk1, musk2, conf8,

conf35, easyCT and Stock4B have been tested by k-NN classification for a range of $k \in \{1, 2, 3, 5\}$ for the distance measures Average Linkage (AvgLink), Minimum Distance (MinDist), Half the Sum of Minimum Distances (HMD), Sum of Minimum Distances (SMD), the asymmetric Hausdorff distance (HD asym.), Hausdorff Distance (HD) and the Convolution Distance (CD) for the linear kernel (dot) and an RBF kernel (RBF). The RBF kernel is defined as

$$k_{\mathrm{RBF}}(a, b) = \exp\left(-\frac{(a-b)^T \cdot (a-b)}{2\sigma^2}\right), \tag{4.25}$$

for real-valued feature vectors $a, b \in \mathcal{F} = \mathbb{R}^d$, in a d-dimensional space. The RBF kernel's σ parameter has been estimated in a comparable manner as in [70]: we sample a large number of instance pairs from the database and take the median Euclidean distance of the sample as an estimate for σ. Naturally, this is only a heuristic which can be further improved by excessive parameter screening. Experiments for Maximum Mean Discrepancy (MMD) are omitted: due to its assumption that multi-instance objects are drawn from a single distribution, it performed worse than the comparable Convolution Distance with an RBF kernel in any experimental setting.

Table 4.3 shows the best classification accuracies (Acc: number of correctly classified objects / database size) per dataset and distance measure, together with the k parameter that returned the best classification. All experiments but the tests of the conf and Stock4B datasets are averaged over 10 repetitions of 10-fold, stratified cross-validation. This setting was chosen for historical reasons in order to allow a fair comparison to the trained distance measures in Section 4.3. The actual accuracies of a leave-one-out setting are slightly higher.

We see that for most datasets, the best experiments were achieved using the Sum of Minimum Distances (SMD), whereas Hausdorff Distance (HD) and Half the Sum of Minimum Distances (HMD) performed best for only one dataset each. While the relative performance of SMD and HMD is rather stable over most tested datasets, HD appears to be only suited for the musk datasets and not for the SIFT-based test sets. The same observation holds for the asymmetric Hausdorff distance (HD asym.) and Average Linkage (AvgLink).

The group of Convolution Distances (CD) appears to be well-suited for the musk datasets, it has mediocre validation values on the conf datasets and it is not appropriate for the less homogeneous Caltech (easyCT) and Stock4B datasets.

In general, the 32-dimensional SIFT descriptors perform slightly worse than the larger, 128-dimensional SIFT descriptors, but this is not always the case. Overall, the variations in accuracy are rather strong within a set of cross-validation experiments: the standard deviations of the accuracy range from

Table 4.3: Accuracies of Basic Multi-Instance Distance Measures together with the optimal $k \in \{1, 2, 3, 5\}$ in k-nearest neighbor classification. The Convolution Distance (CD) is tested both for the scalar product (dot) and an RBF Kernel (σ parameters listed in the last column). The best-performing distance measures per dataset are highlighted in bold.

	AvgLink		MinDist		HMD		SMD		HD asym.		HD		CD (dot)		CD (RBF)		
Dataset	Acc	k	Acc	k	Acc	k	Acc	k	Acc	k	Acc	k	Acc	k	Acc	k	σ
musk1	0.842	2	0.895	2	0.863	1	**0.896**	2	0.853	1	0.831	2	0.819	2	0.823	2	1055.8
musk2	0.746	3	0.775	3	0.763	3	0.795	3	0.754	3	**0.804**	1	0.755	3	0.758	1	1146.6
conf35*	0.186	1	0.827	1	0.868	1	**0.910**	2	0.062	3	0.230	5	0.648	1	0.724	1	16.8
conf35_32D*	0.179	3	0.788	1	0.800	1	**0.897**	1	0.078	5	0.305	3	0.646	3	0.749	1	554.7
conf8*	0.299	1	0.860	1	0.879	1	**0.911**	2	0.160	2	0.374	2	0.764	3	0.774	1	16.6
conf8_32D*	0.331	2	0.867	2	0.808	1	**0.901**	1	0.123	2	0.379	5	0.753	3	0.806	2	554.7
easyCT	0.234	2	0.241	3	0.356	3	**0.427**	1	0.147	1	0.191	5	0.387	1	0.240	1	16.4
easyCT_32D	0.240	5	0.225	5	0.334	2	**0.438**	1	0.136	5	0.159	2	0.361	1	0.419	1	554.7
Stock4B†	0.043	5	0.403	1	**0.478**	1	0.414	3	0.037	2	0.108	2	0.378	1	0.352	1	16.8
Stock4B_32D†	0.031	5	0.380	1	0.414	1	**0.419**	1	0.049	1	0.099	2	0.274	1	0.373	1	555.5

*10x5-fold CV
†2x10-fold CV

Table 4.4: Accuracies of Indexing-based Multi-Instance Distance Measures. Evaluated via 10x10-fold cross-validation (in case of conf, only 5-fold) for a screening of $k \in \{1, 2, 3, 5, 10, 25\}$.

	Avg. k-Min. Link.		Geom. k-Min. Link.		k-Min. Dist.	
Dataset	Acc	k	Acc	k	Acc	k
musk1	0.864	2	**0.870**	3	0.851	25
musk2	**0.816**	10	0.802	25	0.784	25
conf35_32D	**0.889**	2	0.898	5	0.838	10
conf8_32D	0.854	3	0.861	3	**0.890**	2
easyCT_32D	**0.556**	1	**0.556**	1	0.252	25

1 % (in musk1) to 10 % (in musk2) to more than 30 % in many experiments on noisy image-based datasets.

Table 4.4 validates the accuracies of the instance-indexing-based similarity retrieval approaches introduced in Section 4.4.2. In order to provide realistic runtime tests in the following section, the experiments are restricted to the musk datasets and the dimensionally reduced image datasets. We see that the performance of instance indexed retrieval is mostly competitive to the actual multi-instance distance measures presented in Table 4.3. Even though the top performers among the multi-instance distances are rarely excelled, on average, this flat classification scheme completely ignoring the multi-instance structure of the tested datasets appears to be sufficient for most applications.

4.5.3.2 Retrieval Runtime

We now want to verify that the indexing-based retrieval approaches are actually faster than the real multi-instance rankings conducted with a sequential scan. The experiments listed in Table 4.4 have been generated using instance indexes organized as X-Trees [15]. Since the X-Tree is actually only suited for medium-dimensional data, even the retrieval queries on the image datasets with only 32 dimensions are not optimally sped up, let alone the 166-dimensional musk datasets. As, however, experiments with other index structures which are better suited for high dimensions like the VA-File [163] did not result in better runtimes, we decided to stay with the X-Tree.

Figure 4.3 show the query times per query object for four differently-sized datasets. The grey bars display the number of multi-instance objects and the total number of instances of the datasets, whereas the lines are query runtime measurements. Note the logarithmic scale of the y-axes for correctly

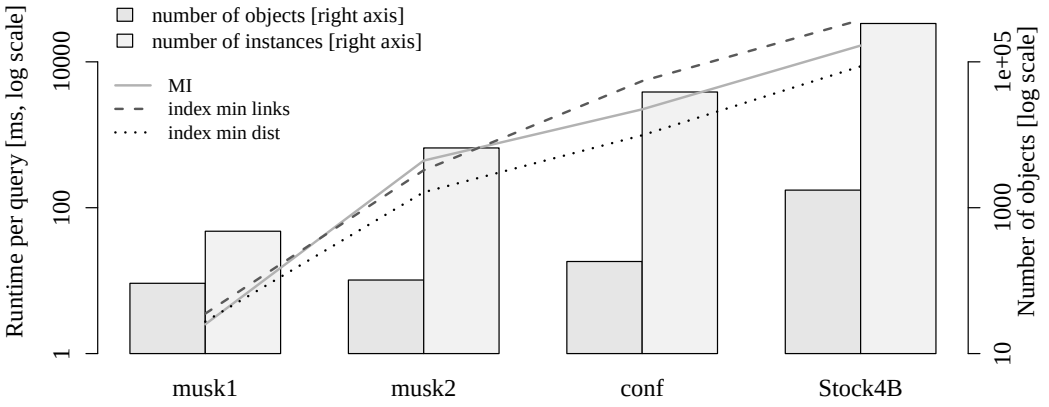


Figure 4.3: Runtimes per query for two types of multi-instance classification queries on datasets of varying size. The continuous line represents multi-instance (MI) distance measures; dashed and dotted lines represent runtimes of instance-indexing approaches.

interpreting the results. We differ between three types of retrieval settings in the runtime measurements:

MI The MI line summarizes runtimes of sequential scans using multi-instance distance measures requiring the quadratic number of instance distance computations, i.e. $|A|\,|B|$ instance distances for two multi-instance objects $A, B \subseteq \mathcal{F}$. Note that both the Sum of Minimum Distances (SMD) and Hausdorff Distance (HD) actually require twice the computation time of the other multi-instance distance measures, however, for expensive distance computations like the high- to medium-dimensional features vectors used in this setting, the instance distances can be cached for avoiding a duplicate computation.

index min links The dashed line summarizes the query runtimes of the Average k-Minimum Linkage and the Geometric k-Minimum Linkage Classification experiments. By using the X-Tree on comparatively high-dimensional datasets, there is nothing left of the runtime improvement reached by avoiding the quadratic all-against-all complexity of the sequential multi-instance scan. In fact, runtimes on the larger image datasets are even higher than the pure multi-instance approach.

index min links Finally, the dotted line displays the query runtimes of the Global k-Minimum Distance Classification. Using this strategy, we finally see that index-based retrieval *is* faster than the scan-based retrieval

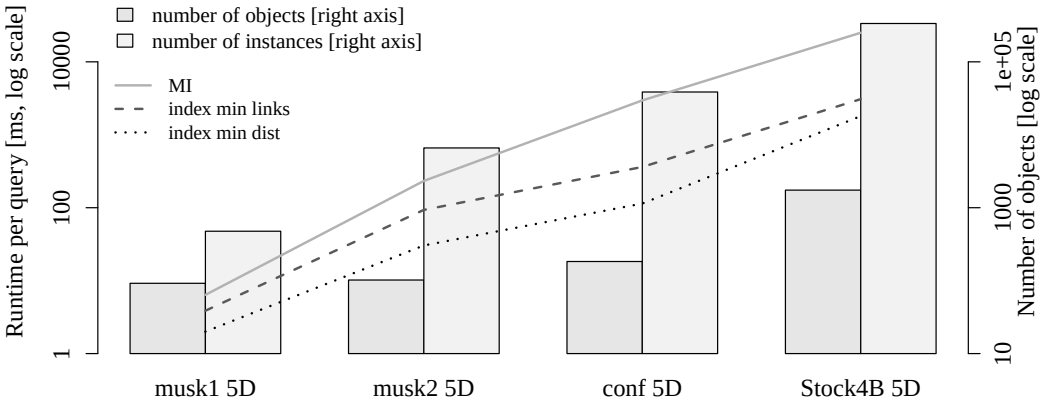


Figure 4.4: Runtimes per query for lower-dimensional datasets with the same settings as in Figure 4.3.

using real multi-instance distance measures. In general, this comes at the price of slightly lower class prediction accuracies.

All experiments were executed in main memory. Therefore, you might argue that indexing-based queries will in the end profit more on larger databases than sequential scan experiments. However, only an increased profit is not sufficient for an application in the real world, if a user has to wait too long for the classification of one image.

A much stronger argument in favor of the proposed indexing approaches is their clear superiority to scan-based multi-instance distances in lower-dimensional spaces. When reducing the tested datasets to 5-dimensional feature vectors, the retrieval times are strongly improved for both query approaches, however, the instance-index methods profit by far more from the dimensionality reduction. The runtimes are displayed in Figure 4.4. Similar observations can be made on completely artificial datasets.

Note, however, that the accuracies of the underlying classification tasks suffer enormously from the reduced information content of those simplified datasets. Even when employing supervised dimensionality reduction methods like Relevant Component Analysis (RCA) [7], the multi-distributed character of multi-instance objects just fails to be adequately conserved into the new, lower dimensional feature space. Image features usually *are* high-dimensional, therefore, the attempt of reducing the complexity of an image descriptor can only succeed in special cases as the approach introduced in Section 5.5.3.

4.6 Summary

This chapter provided a survey over existing multi-instance distance measures and suggested a number of improvements w.r.t. accuracy and runtime. Multi-instance objects represent an entity as a set of sub-entities, the so-called instances. Since multi-instance objects are a common form of image descriptors, the author of this thesis thought it worthwhile to investigate this special field of object representation.

The main approach of improving multi-instance distance measures was based on computing instance weights either for weighting the instances' contributions to a global multi-instance distance or for selecting meaningful instances from a dataset. After testing several weighting procedures and instance weighting and selection methods, however, we had to realize that the idea of instance weighting suffers from various obstacles related to finding the right weights for the query instances.

Therefore, the experimental validation of this chapter is focused on a general assessment of basic multi-instance distance measures on a variety of real-world datasets and the exploration of runtime improvement possibilities. We conclude that the qualitative performance of multi-instance distance measures depends strongly on the properties of the dataset. It appears that minimum distance-based measures have an advantage of stability over geometrically-inspired measures like the Hausdorff distance. The overall best-performing distance was the Sum of Minimum Distances (SMD), which also showed to be successful in a content-based image retrieval application described in Chapter 6.

In matters of runtime, the proposed improvement strategies of Section 4.4.2 succeeded in lower-dimensional datasets, which are suited for an efficient retrieval framework based on spatial indexes. As, however, most image descriptors are high-dimensional, the value of the discoveries presented in this chapter lies more within the theoretical implications than in an actual practical use for medical imaging.

The second part of this thesis will thus be dedicated to the examination of practical applications in medical image search instead of a further investigation of the theoretical background of similarity search.

Part II

Similarity Search in Medical Image Repositories

The theoretical knowledge on similarity search and image retrieval is steadily widened and adapted to new application areas. Consequently, the feature transformation and distance learning approaches introduced in this thesis can very well be applied on standard imaging benchmarks (e.g. [81, 58, 134, 144]) and depending on their suitability, they will achieve good results. However, even benchmarks which are explicitly directed to medical image retrieval tasks like this year's ImageCLEF task (`http://www.imageclef.org/2011/medical`) are abstracted too far from the medical routine for actually being able to function as an effective test for a real-world medical content-based image retrieval (CBIR) system.

The challenges in medical CBIR range from very large, unstructured datasets over complicated availability issues to a low quantity of adequately-annotated training images. Even though modern hospitals quickly accumulate large image repositories, in order to guarantee the privacy protection of their patients they rarely share this data with non-medical research groups. Consequently, the research project THESEUS MEDICO enjoys a special role in the field of image retrieval, since it is actually able to use parts of a big real-world dataset of 3D images collected at the Imaging Science Institute of the University Hospital Erlangen. This data, however, has been thoroughly pseudonymized such that valuable fields of meta-information like details on the used contrast agent or even the patient's age have been removed. As an additional implication, the images are bare of any additional medical information, because any supplementary repositories of the radiological information system (RIS) like textual radiology reports or laboratory measurements may not be directly connected to those pseudonymized volumes.

As mentioned before, CBIR is not yet very common in medicine. Even in well-defined and technically advanced retrieval applications like cervicographic imaging, the concept of example-based retrieval queries has not yet been accepted by the users. [4] Therefore, it was a very demanding task to associate any useful labelling information to the dataset for defining a practically-relevant use case of image similarity search.

We defined our first CBIR application around the only piece of meta-information available: the body position of the given 3D volume. Since the examined patients have varying body sizes and proportions, 3D scans of the human body are not taken with respect to a standardized body coordinate system but with respect to a scanner-specific real-world offset. The only information available about the actual anatomical type of the scanned body region is thus a short textual tag in the images' meta-data and even that may be wrong. [74] Chapter 5 therefore tackles the problem of automatic anatomical localization in Computed Tomography (CT) scans by defining a standardized body coordinate system for providing anatomical information about the scope

of a volume. The MEDICO protoype can exploit this information for automatically presenting corresponding body regions and for greatly speeding up volume retrieval queries.

Additionally, another goal of the MEDICO project was to provide the functionality for actual instance-based image queries. In order to distinguish our new approach from retrieval systems which have been available at the start of the project or which were expected to be on the market too soon, we decided to support 3D queries, i.e. sub-volumes of a three-dimensional intensity grid representing a CT scan. On the one hand, this posed the challenge of defining new 3D descriptors which can be efficiently queried in a large repository of annotations. On the other hand, these 3D annotations to be queried first had to be generated and labeled according to a medically relevant notion of similarity. Hence, we defined our second use case as similarity retrieval task on lesions visible in CT scans. This application is described in Chapter 6 and it is integrated into the MEDICO protoype as a generic and versatile query option.

In course of the problem-solving process for those two use cases, we also tested the solutions for distance learning and feature transformation introduced in the first part of this thesis. However, in most areas of the underlying similarity search components, we had to develop alternative solutions due to scalability or precision constraints. Hence, the bi-partite nature of this thesis.

Chapter 5

Region of Interest Queries in CT Scans

The first retrieval application on medical image databases described in this thesis is a special type of volume retrieval queries. Depending on the required degree of precision, medical images – and especially CT scans – can become rather large data objects. In order to guard patient privacy and to guarantee a centralized data store, all volumes of a hospital are usually stored in a central Picture Archiving and Communication System (PACS). A user asking to see a specific volume thus has to wait for the data to be loaded from this server. Especially if there are multiple requests for large data transfers at the same time, today's hospital I/O environments are frequently exhausted and the waiting times become a real obstactle to the clinical routine.

Additionally, the user frequently is not interested in the complete image volume, but primarily wants to see a certain subregion, a Region of Interest (ROI). This ROI may be a general body region like the head, a smaller body structure like an organ, a specific 2D view of a volume intersection or the same body part, already opened in another scan. Two possible query scenarios are displayed in Figure 5.1. Using a standard PACS, the user needs to load the complete volume and then has to manually navigate to the location of the actual ROI.

The purpose of this work is to save time required for both actions: loading the volume and finding the ROI. If the system knew where to find the ROI, this ROI could be pre-loaded and readily displayed to the user for examination, while the rest of the volume is either left on the server or loaded with a lower priority than the ROI. The problem of this approach is that CT scans are not normalized to a unified coordinate system, but they vary in image resolution, in the body size and proportion of the examined patient and in the context of the actually visualized body part. There already are multiple image registration

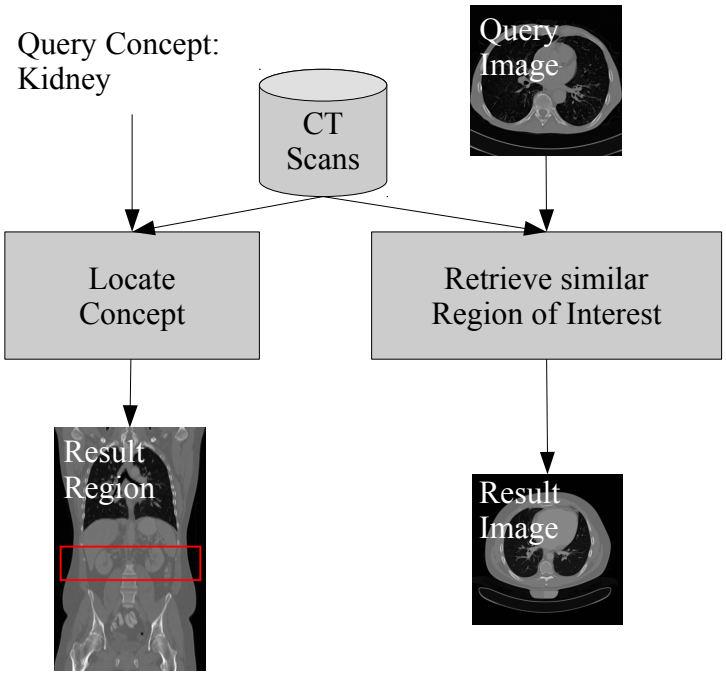


Figure 5.1: Two examples for ROI queries: to the left, the user is interested in the region of the body showing the kidneys. To the right, the user wants to see the same body region already opened in a template or example scan.

approaches trying to map patient images into a standardized template space, however, these are expensive to compute and usually they are specialized to a local body part.

In this chapter, we review efficient methods of registering a CT scan into a standardized height atlas, which enables the mapping of two volumes along their height axis. This includes a landmark-based registration approach as well as the similarity search application of [55], mapping single slices into a standardized height space. Thus, we are able to query both annotated datasets and databases which have not yet been subject to any kind of pre-processing. [25] Both methods can be applied on flexibly-defined ROIs and they are carefully evaluated in matters of precision and runtime.

5.1 Introduction

Radiology centers all over the world currently collect large amounts of 3D body images being generated with various scanning modalities like CT, PET-CT, MRT or sonography. Each of these methods generates a three dimensional image of the human body by transforming the echo of a different type of signal

allowing a radiologist to examine the inner parts of a human body. In the following, we will particularly focus on CT body scans. However, the methods proposed in this chapter are generally applicable to other types of scans as well.

Technically, the result of a CT scan is stored as a stack of 2D images representing 3D slices of the human body, i.e. each slice is considered to have a certain thickness. The scans in a radiology center are stored in a centralized picture archiving and communication system (PACS) and they are transferred via LAN to the workstation of a physician. In commercial PACS, querying CT scans is currently restricted to retrieving complete scans being annotated with certain meta information like patient name, date and type of the examination. Therefore, each time a CT scan is queried, the complete scan, potentially comprising several thousand high-resolution images, has to be loaded from the image repository. For example, the data volume of a thorax scan being generated by a modern scanner comprises around 1 GB of data. Considering that several physicians will simultaneously query a PACS, the loading time of a single CT scan is up to several minutes depending on network and server traffic. Additionally, after loading the CT scan, it is necessary to manually navigate to the region of interest (ROI) the physician needs to examine.

However, in many cases it is not necessary to display the complete scan. For example, if a physician wants to see whether a certain liver lesion has improved between two scans, the user primarily requires the portion of both scans containing the liver. Therefore, the physician loses up to several minutes by loading unnecessary information and searching for the liver within both scans. Thus, a system retrieving the parts of both scans containing the liver, would save valuable time and network bandwidth.

Parts of a CT scan can be efficiently loaded by raster databases [9] as long as the coordinates of the ROI are specified. However, in the given context, the ROI is rather defined by the image content. In other words, the coordinates of organs and other anatomical regions may strongly vary because of differences in the patients' heights or in the scanned body region. Thus, raster coordinates cannot be used to align to CT scans w.r.t. the image content.

We therefore propose to use an implicit, standardized coordinate system which can be used to define a concept-based query (like the region containing the liver or the fifth lumbar vertebra) or to standardize an example-based query (comprising an ROI which is currently highlighted by the user in an example scan) into the implicit coordinate system. The result of an ROI query contains the part of the scan showing the hereby-defined, standardized query ROI in one or multiple result scans.

The most established approach to answer these types of queries is based on landmark detection. [137] A landmark is an anatomically unique location

in the human body which is well-detectable by pattern recognition methods. To use landmarks for query processing, it is first of all necessary to detect as many landmarks as possible in the example scan and all result scans. Let us note that landmark detection employs pattern recognition methods and thus, there is a classification error, i.e. some of the predicted landmark positions are error-prone. Furthermore, it can happen that some of the landmarks are not detectable due to disturbances while recording the scan. Additionally, a set of landmarks cannot be completely invariant for various patients due to varying size and proportion. Neither will it be invariant for different scans of the same patient as there will be distortions due to movement or varying body conditions (state of digestion, breathing artifacts, loss or gain of weight).

Since these two sources of uncertainty (detection error and imperfect invariance) cannot be completely eliminated in the ensuing volume alignment step it must be regarded with care. However, having detected a sufficiently large number of landmarks, it is possible to align both scans and afterwards select the area from the target scan corresponding to the query. A method using the principle of landmark-based volume alignment will be presented in Section 5.4. Our approach is stabilized by using an additional standardized coordinate system, which combines the landmark information of a large database of CT scans. This combination results in an assessable set of standardized landmark positions which can be also used for aligning volumes with non-complementing sets of landmarks.

An important aspect of this approach is that landmark detection should be done as a preprocessing step. Thus, the example scan of an example-based query and the target scans need to be annotated with the landmark position to allow efficient query processing. However, this causes a problem when allowing example scans not being stored in the same PACS. In this case, the query might not have any landmarks or it is not labelled with the same set of landmarks. If the example scan and the result scan are taken by CT scanners from different companies, the positioning systems might not be compatible. Another problem of the landmark approach is the scope of the scan. CT scans are often recorded for only a small part of the body. Thus, it cannot be guaranteed that the scanned body region contains a sufficiently large set of align-able landmarks for either the example scan and the set of queried scans in the PACS. To conclude, a fixed and comparably small set of landmarks is often not flexible enough to align arbitrary scans.

Consequently, we propose a more flexible approach being based on similarity search on the particular slices of a CT scan. [55] This method does not rely on any time-consuming preprocessing steps, but it can be directly applied on any example and result scan. Whereas landmark-based approaches can only align scans with respect to a limited amount of fixed points to be matched, our

new approach can generate the positions in the scan to be matched on the fly. Thus, we can even align scans being labelled with different types of landmarks or scans not having any detectable landmarks at all.

The key idea behind our method is to map single slices of a CT scan to a standardized height model describing the relative distances between concepts w.r.t. the height axis of the human body. The height model is independent of the individual size and proportions of a particular patient. Let us note that it is possible to use width and depth axes as well. However, the height axis is the predominantly used navigation axis for CT scans.

By mapping single slices to the model, we can better adjust to limited information about the scan and we are independent from the distribution of predefined landmark positions. Our prediction algorithm employs instance-based regression for assigning a standardized height coordinate to a single query slice. In order to efficiently answer k-nearest neighbor queries we use Relevant Component Analysis [7] for dimension reduction of the input space and then we index our training database using an X-Tree [15].

Landmark-independent ROI queries differ from landmark-based queries as follows: Example-based queries employ instance-based regression to determine the query ROI in the standardized height model instead of interpolating it from a set of landmarks. Then, we need to identify the ROI in each non-annotated target scan corresponding to the query ROI in the height model. Let us note that this second step is more complicated, since we cannot directly determine the slice belonging to a particular height value. One solution to this problem would be to label all available slices with the height value in the model. However, labelling all DICOM images in an average PACS would cause an enormous overhead in preprocessing. Since the majority of images will never be involved in answering an ROI query, we pursue a different strategy. Instead of preprocessing each image in the PACS, we assign standardized height values for a given slice on the fly. To make this type of processing efficient, we propose a query algorithm that alternates regression and interpolation steps until the queried ROI is found in the result scan.

Let us note that although the solutions proposed in this chapter are very problem-oriented, the solution principle can be extended to other data as well. For example, a similar processing scheme can be applied to video streams (e.g. procedure timing in surveillance videos) or text mining (e.g. news tickers, twitter streams, age classification in Internet forums).

The rest of the chapter is organized as follows. Section 5.2 surveys methods that are related to our approach or parts of it. In Section 5.3, we formalize the two types of ROI queries and give an overview of our system. Section 5.4 describes interpolation methods for aligning CT scans to a generalized height model followed by an algorithm for learning a body atlas from annotated exam-

ple scans. Afterwards, Section 5.5 introduces our method for predicting height values for particular CT slices. Section 5.6 presents the query algorithm for ROI queries without any pre-processing, and the results of our experimental evaluation are shown in Section 5.7. The chapter concludes with a brief summary and ideas for future work in Section 5.8.

5.2 Related Work

In medical imaging, there are various localization or registration approaches. Most of them are domain specific, like the Talairach space brain atlas [149], the MNI space [56] or a more recent thorax atlas [160]. Nevertheless, as these atlases are very specific to their domain, they were not designed to cover the entire body and they can thus hardly be used for general ROI queries.

Position mapping via landmark-detector-based approaches like the THESEUS MEDICO system presented in [137] are more appropriate for our purpose. This prototype provides an image parsing system which automatically detects 22 anatomically relevant landmarks, i.e. invariant points, and 9 organs. [136] It is thus possible to query the database directly for ROIs which are equivalent to these automatically-annotated image regions. However, general queries for arbitrarily defined ROIs are not yet supported.

A more general, landmark-based interpolation approach for mapping a volume into a standardized height space has been proposed by [76]. However, it is very patient-specific and dependent on the used landmarks. Another approach that uses partial volumes as query is described in [59]. It localizes query volumes with sizes ranging from 4 cm to more than 20 cm by comparing the partial volume with an implicit height atlas based on Haar-like features. In [55], we presented an alternative method such that only a single query slice is needed in order to achieve comparable results. We will examine this method in Section 5.5.

Section 5.5.3 introduces an iterative interpolation and regression approach. In contrast to established regression methods, [83, 120] we enhance our model with newly generated information after each iteration in order to refine the final model until convergence is reached.

We experimented with several regression methods from the Weka machine learning package [77]. However, simple approaches like linear regression did not yield a sufficient prediction accuracy and more complicated approaches like support vector regression using non-linear kernel functions could not cope with the enormous amount of training data. Therefore, we decided to employ instance-based regression which is robust and sufficiently fast when employing techniques of efficiently computing the k-nearest neighbors (k-NN). In partic-

ular, we employ k-NN queries being based on the X-Tree [15]. Let us note that there are multiple other index structures [133] for speeding up the same type of query. We decided to employ the X-Tree because it represents an extension of the standard R*-Tree [99] which is better suited for higher dimensionalities.

Current database systems like RasDaMan [9] already support conventional region of interest queries in raster data like CT scans. Nevertheless, the system needs to know the coordinate system in which the query is applied in order to navigate to the requested region. As we do not know the complete coordinate systems of the patients' CT scans in advance and since patients differ in height and body proportions, and thus, locations along the z-axis are not standardized, a globally fixed coordinate system will not be available in our setting. Therefore, our new approach represents a way to bridge the gap between the coordinates in the query scan and the coordinate system of the result scan.

5.3 Workflow Overview

In this section, we specify the proposed query process and give an overview of the proposed system.

Definition 23 (Volume Dataset, Standardized Height Space)
A volume dataset *consists of n volumes $v_i \in \mathbb{N}^{x(i)\times y(i)\times z(i)}$ with $i \in \{1,\ldots,n\}$. Each integer value within the grid of a volume is termed a* voxel. *Conseqeuntly, $x(i), y(i)$ and $z(i)$ are referred to as voxel dimensions. The* standardized height space *$H \subset \mathbb{R}$ is an interval $[h_{min}, h_{max}]$ with $h_{min}, h_{max} \in \mathbb{R}$ representing the extension of the human body in the z-axis.*

Note that H is *not* equivalent to a patient's real-world height in millimeters, since the standardized height space is designed to be independent of any given volume v_i.

Definition 24 (Mapping Functions h and s)
A mapping function *$h_i : \mathbb{N} \to H$ maps slices of volume v_i to a standardized height value $h \in H$. Correspondingly, the* reverse mapping function *$s_i : H \to \mathbb{N}$ maps a position h in the standardized height space to a slice number s in v_i.*

As a link between these two spaces, we define matching points:

Definition 25 (Matching Points) *A triple $p = (\mathbf{s}_{i,p}, \mathbf{h}_p, \mathbf{w}_p) \in \mathbb{N} \times H \times \mathbb{R}$ of a slice number $\mathbf{s}_{i,p} \in \{0,\ldots,z(i)-1\}$ in volume v_i, its corresponding height value $\mathbf{h}_p$ in H and a reliability weight $\mathbf{w}_p \in \mathbb{R}$ is called a* matching point. *By P_i, we denote the set of all available matching points of v_i.*

Table 5.1: Notation of frequently used parameters.

n	number of volumes
$v_i \in \mathbb{N}^{x(i)\times y(i)\times z(i)}$	one volume ($i \in \{1,\ldots,n\}$)
$H \subset \mathbb{R}$	target space / height model
$\mathbf{h}_j$	one height value in H
$\mathbf{s}_{i,j}$	one slice number of v_i in $\{0,\ldots,z(i)-1\}$
$\mathbf{w}_j$	reliability weight
$p = (\mathbf{s}_{i,p}, \mathbf{h}_p, \mathbf{w}_p)$	matching point
P_i	set of matching points of v_i
$h_i(s)_{P_i}$	mapping function of $\mathbb{N} \to H$ using a set P_i
$h_i^{\mathrm{REG}}(s)$	regression function of $\mathbb{N} \to H$
$s_i(h)_{P_i}$	interpolation function of $H \to \mathbb{N}$ using a set P_i
$(\hat{\mathbf{s}}_{i,\mathrm{lb}},\ldots,\hat{\mathbf{s}}_{i,\mathrm{ub}})$	slice range in v_i
$[h_{\mathrm{lb}}, h_{\mathrm{ub}}]$	interval in H
k	k-nearest neighbors (k-NN) parameter
$F(s_{i,j}) : \mathbb{N} \to \mathcal{F} = \mathbb{R}^d$	feature transformation of slice j of v_i with $d \in \mathbb{N}$
T_A	training set for atlas
T_R	training set for regression

Table 5.1 displays an overview of the most frequently used parameters including some additional notations that will be introduced in the following sections.

In our system, a region of interest (ROI) query results in the retrieval of a consecutive sequence of CT slices $(\hat{\mathbf{s}}_{i,\mathrm{lb}},\ldots,\hat{\mathbf{s}}_{i,\mathrm{ub}}) \subseteq \{0,\ldots,z(i)-1\}$ for a query range $[h_{\mathrm{lb}}, h_{\mathrm{ub}}] \subseteq H$ in the standardized height space from a volume v_i. Here, lb and ub are mnemonics of lower and upper bound, respectively. This query range can be specified by either giving an example range in another example volume v_e or by defining an anatomical concept with a known standardized height range.

Figure 5.2 illustrates the complete workflow of query processing for concept-based queries and example-based ROI queries where the user provides an example ROI.

5.3.1 Example-based Query Definition

Definition 26 (Example-based ROI Query) *In* example-based ROI queries*, the queried ROI is described by the content of an example set of consecutive slices* $(\hat{\mathbf{s}}_{e,lb},\ldots,\hat{\mathbf{s}}_{e,ub}) \subseteq \{0,\ldots,z(e)-1\}$ *in volume* v_e*.*

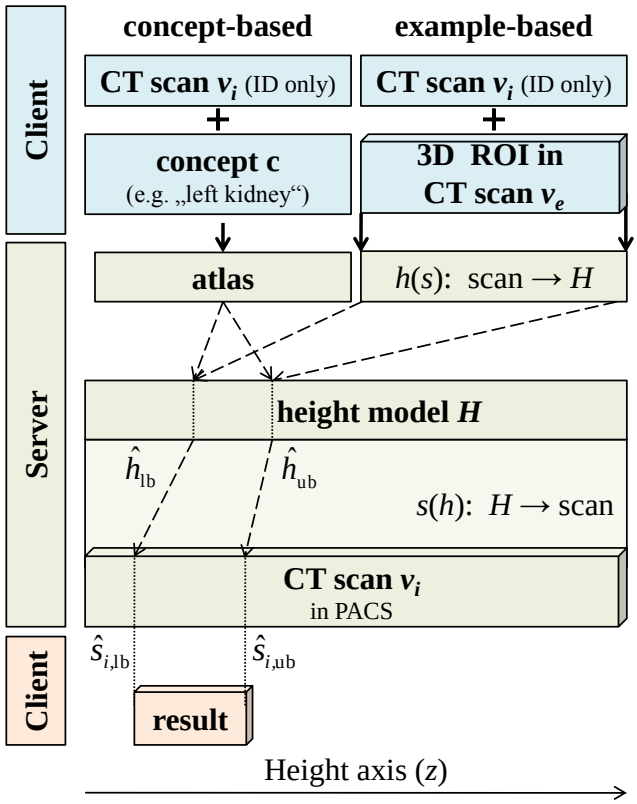


Figure 5.2: Workflow of ROI retrieval for concept-based and example-based queries.

In practise, the user specifies an ROI query on the client computer by marking a region in an example scan v_e. Additionally, the queried scan v_i has to be identified for the server.

The query interval $[h_{\text{lb}}, h_{\text{ub}}] \subset H$ of an example-based ROI query can be determined using mapping function h_e on the slices of the given example ROI. This chapter contains two ways of implementing h_e: Section 5.4 describes an interpolation approach, which requires an existing mapping of v_e to an anatomical atlas of body landmarks. This mapping defines matching points P_e with $p = (\mathbf{s}_{e,p}, \mathbf{h}_p, \mathbf{w}_p) \in P_e$ mapping the slice $\mathbf{s}_{e,p}$ of v_e of landmark p to its standardized atlas position $\mathbf{h}_p$ and weight $\mathbf{w}_p$.

In Section 5.5, we describe a regression approach which derives the lower and upper bounds h_{lb} and h_{ub} via an instance-based regression function $h_e^{\text{REG}}(s)$. This function derives image features for the lower and upper slice of the example ROI. Thus, it is not necessary to transfer the complete marked subset of slices of v_e to the server. Instead, it is sufficient to transfer a scale-reduced version of the first and the last slice of the subset. After receiving the slices,

the server performs a feature extraction step generating image descriptors for both slices. As an alternative, the client computer can also directly compute the required image descriptors and only transfer the descriptors.

5.3.2 Concept-based Query Definition

Definition 27 (Concept-based ROI Query) *In* concept-based ROI queries*, the queried ROI is described by the named concept $c \in C$, e.g. an organ like the heart or the liver. C is the set of all considered query concepts.*

Concept-based queries require a global body atlas which contains a probability distribution $P_c(H = h)$ for the standardized heights of each concept $c \in C$. This allows to specify a query range $[h_{\text{lb}}, h_{\text{ub}}] \subseteq H$ as

$$\left[\underset{h}{\text{argmax}}\{P_c(H \leq h) \leq \tau\}, \underset{h}{\text{argmin}}\{P_c(H \geq h) \leq \tau\}\right] \tag{5.1}$$

for any concept c and a threshold quantile τ. (By default, we use $\tau = 0.05$).

Section 5.4.2.4 describes our way of obtaining the probability distribution of such a standardized body atlas.

5.3.3 ROI retrieval for a standardized height range

With the query interval $[h_{\text{lb}}, h_{\text{ub}}]$ given, the server applies $s_i(h)$ to determine the result set of corresponding CT slices $(\mathbf{s}_{i,\text{lb}}, \ldots, \mathbf{s}_{i,\text{ub}}) \subseteq \{0, \ldots, z(i) - 1\}$ in the queried volume v_i. Afterwards, the CT slices from $\mathbf{s}_{i,\text{lb}}$ to $\mathbf{s}_{i,\text{ub}}$ are transferred to the client computer. Let us note that $s_i(h)$ returns an approximative slice range $(\hat{\mathbf{s}}_{i,\text{lb}}, \ldots, \hat{\mathbf{s}}_{i,\text{ub}})$, which is usually extended by the amount of slices corresponding to 90 % of the expected prediction error in order to compensate for the inaccuracy of the used mapping functions.

Comparable to the query specification of a query-by example, we examine two ways of implementing $s_i(h)$. If volume v_i is already registered into a height atlas, the interpolation approach of Section 5.4 can also be used for the mapping $s : H \mapsto \mathbb{N}$. Else, v_i must first be transformed into the standardized height space H. The straightforward way would be to map all slices of v_i to H with the regression approach introduced in Section 5.5. The resulting set of $z(i)$ matching points P_i would then be used for interpolating a compromise result range – a necessary measure, since the matching points are prone to contradict each other due the a given uncertainty of $h_i^{\text{REG}}(s)$. This approach, however, is rather expensive, since it requires the slice retrieval and feature computation for all slices of v_i.

A more efficient way of aligning volume v_i to the height model H is an algorithm described in Section 5.6. Its aim is to find the trade-off between a low number of matching points and a high reliability of the resulting slice range.

5.4 Interpolation using Matching Points

If a set of matching points P_i is available for a given volume v_i, they can be used for defining the functions $s : H \mapsto \mathbb{N}$ and $h : \mathbb{N} \mapsto H$ for mapping positions in the standardized height space H to a specific slice number of v_i and back. The following section will present some interpolation functions which are suitable for implementing s and h, whereas Section 5.4.2 introduces several ways for obtaining standardized H coordinates of a set of landmarks which can be used for defining such a set of matching points.

5.4.1 Interpolation Functions

The first interpolation approach maps a scan v_i to the height model H based on a set of available matching points P_i. Inspired by a setting of Haas *et al.* [76], we tested various non-linear interpolation approaches for dealing with varying body proportions and imprecise matching points.

The function $h_i(\mathbf{s}_{i,q})_{P_i}$ maps slice number $\mathbf{s}_{i,q}$ of volume v_i to a height value $\mathbf{h}_q \in H$. The dependency of $h_i(\mathbf{s}_{i,q})_{P_i}$ on the scan v_i is determined by the set of matching points P_i and the slice spacing δ_i describing the thickness of a slice in the target space H. We approximate δ_i as the median slice spacing over all pairs of matching points in P_i:

$$\hat{\delta}_i = \text{median}_{(p,p')\in P_i, \mathbf{s}_{i,p}\neq\mathbf{s}_{i,p'}} \left|\mathbf{h}_p - \mathbf{h}_{p'}\right| / \left|\mathbf{s}_{i,p} - \mathbf{s}_{i,p'}\right| \ . \tag{5.2}$$

Let us note that we use the median in order to achieve a higher stability against outliers caused by unreliable matching points. Our interpolation functions for predicting the standardized height of a slice $\mathbf{s}_{i,q}$ are based on a linear term $\delta_i \cdot \mathbf{s}_{i,q}$ with a potentially non-linear offset. The mapping $h_i(\mathbf{s}_i, q)_{P_i}$ is thus defined by the following interpolation function

$$h_i(\mathbf{s}_{i,q})_{P_i} = \delta_i \cdot \mathbf{s}_{i,q} + \frac{\sum_{p\in P_i} \mathbf{w}_p \cdot \text{REL}_p(\mathbf{s}_{i,q}) \cdot (\mathbf{h}_p - \delta_i \cdot \mathbf{s}_{i,p})}{\sum_{p\in P_i} \mathbf{w}_p \cdot \text{REL}_p(\mathbf{s}_{i,q})} \ , \tag{5.3}$$

where $\text{REL}_p(\mathbf{s}_{i,q})$ is a relevance term for the matching point p w.r.t. the input slice $\mathbf{s}_{i,q}$. Its flexible definition allows the realization of various characteristics of the matching points' local influence.

The simplest relevance function is equivalent to a *weighted linear interpolation*. Relevance function (LIN) is defined as

$$\mathrm{REL}_p^{\mathrm{LIN}}(\mathbf{s}_{i,q}) = 1 \ . \tag{5.4}$$

When defining more sophisticated relevance functions, we aim to weight matching points which are close to the query slice $\mathbf{s}_{i,q}$ higher than far away matching points. This way, the influence of closer matching points is increased in comparison to potentially irrelevant matching points. The relevance term is thus used in a kernel smoothing manner.

We introduce the *inverse absolute difference* (ABS) between the slice number of the matching point $\mathbf{s}_{i,p}$ and the query slice $\mathbf{s}_{i,q}$. In order to limit the maximal possible relevance, we additionally require a stabilizing parameter t:

$$\mathrm{REL}_p^{\mathrm{ABS}}(\mathbf{s}_{i,q}) = \min\left(t, |\mathbf{s}_{i,q} - \mathbf{s}_{i,p}|^{-1}\right) \ . \tag{5.5}$$

In our experiments, we usually set $t = 1$. A variant of this function is the *squared inverse difference* (SQR):

$$\mathrm{REL}_p^{\mathrm{SQR}}(\mathbf{s}_{i,q}) = \min\left(t, (\mathbf{s}_{i,q} - s_{i,p})^{-2}\right) \ . \tag{5.6}$$

Since we can also accumulate knowledge about the deviation σ_p of the underlying distribution function of a matching point p, we also test a *radial basis function* as a relevance term (RBF):

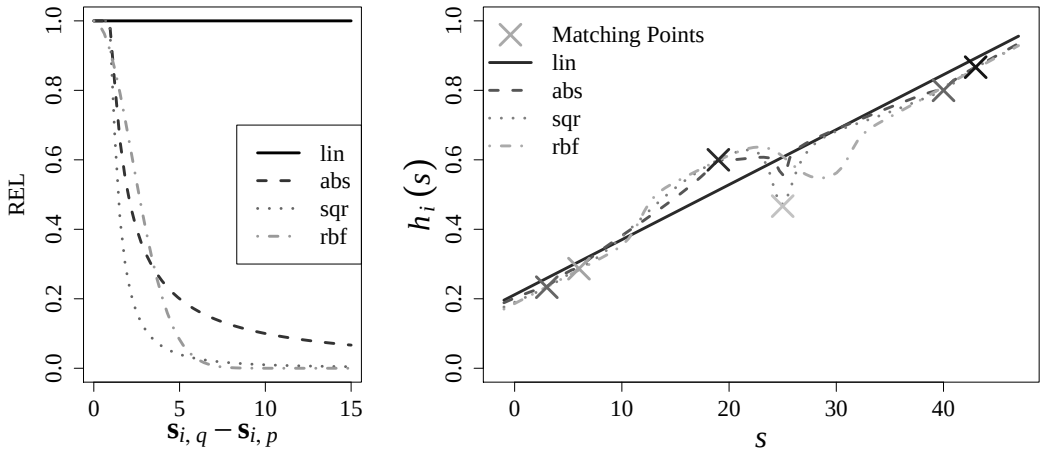
$$\mathrm{REL}_p^{\mathrm{RBF}}(\mathbf{s}_{i,q}) = \exp\left(-\frac{(\mathbf{s}_{i,q} - \mathbf{s}_{i,p})^2}{\sigma_p^2}\right) \ . \tag{5.7}$$

The relevance terms are visualized in Figure 5.3(a). Their effects on the mapping function of a toy dataset of matching points P_i are shown in Figure 5.3(b).

We also considered to use other interpolation families like bilinear interpolation, locally-weighted polynomial regression [32] or other kernel smoothing schemes. Since, however, our experiments showed a clear advantage of the simpler interpolation schemes LIN and ABS in comparison to the more locally adaptive SQR and RBF approaches, we concluded that the given use case does not require any more complex interpolation procedures.

For mapping model positions in H to slices in $\mathbb{N}$ we transform the interpolation function $h_i(\mathbf{s}_{i,q})_{P_i}$ to a *reverse interpolation model* $s_i(\mathbf{h}_q)_{P_i}$:

$$s_i(\mathbf{h}_q)_{P_i} = \frac{\mathbf{h}_q}{\delta_i} - \frac{\sum_{p \in P_i} \mathbf{w}_p \cdot \mathrm{REL}'_p(\mathbf{h}_q) \cdot \left(\frac{\mathbf{h}_p}{\delta_i} - \mathbf{s}_{i,p}\right)}{\sum_{p \in P_i} \mathbf{w}_p \cdot \mathrm{REL}'_p(\mathbf{h}_q)} \ . \tag{5.8}$$



(a) Relevance terms REL for slice distances $\mathbf{s}_{i,q} - \mathbf{s}_{i,p}$.

(b) Interpolation functions on a toy dataset. The matching points' colors correspond to their weights (the darker the point the higher the weight).

Figure 5.3: Relevance terms and exemplary behaviour of the proposed interpolation family.

The transformation inverts the spacing δ_i and swaps the mapping spaces $\mathbb{N}$ and H. Additionally, a relevance function REL$'$ now uses position differences in the model scale instead of slice differences like the relevance functions REL.

The following section will describe how the proposed interpolation approaches can be used for training an atlas of globally-available matching points.

5.4.2 Height Atlas Definition

The accuracies of the above query specifications are mainly dependent on the quality of the matching points P_i of the given volume v_i. A comfortable way to generate these matching points is to use a standardized height atlas $(\mathbf{h}, \mathbf{w}) \in H \times \mathbb{R}$ which maps standardized heights and reliability weights to a pre-defined set of concepts C. In medical imaging, these concepts are usually landmarks, which can be quickly detected and are located at anatomically meaningful and physically stable regions in the body.

There are two principal ways to generate such an atlas: using the real-world landmark positions of a template or example volume or by forming a consensus model from a set of manually-aligned scans. Since these approaches are biased, we additionally propose a refinement algorithm, which improves an existing atlas by iterative adaption to a larger dataset.

5.4.2.1 Model by Example (EXMP)

A Model by Example atlas (EXMP) is formed using a single, well-selected example volume. Our example scan is a publicly available full body scan (`http://pubimage.hcuge.ch:8080/`, MELANIX) showing a 1.75 m tall woman. We use the landmark detector of Seifert *et al.* [136] which annotates each scan with up to 22 landmarks for defining a set of concepts C. In order to account for inaccuracies of the detection process, we average the output of the landmark detector over varying resolution scales as $\mathbf{h}$ and derive the weights $\mathbf{w}$ from the variation of the detected coordinates for each landmark.

5.4.2.2 Manually Aligned Examples (ALIGN)

Another way of atlas generation is to use various versions of detected landmark positions in a manually standardized volume set (ALIGN). [55] Our annotation procedure of a scan v_i consists in defining standardized height values $\mathbf{h}_0$ and $\mathbf{h}_{z(i)-1}$ for the scan's lower slice $\mathbf{s}_{i,0}$ and its upper slice $\mathbf{s}_{i,z(i)-1}$. Any remaining height values are linearly interpolated between these two fix points.

The dataset used for the (ALIGN) atlas consists of 33 CT scans (12 female, 21 male patients) which were manually mapped into a standard $[0, 1]$ scale (0=feet, 1=head). On average, 16 of 22 landmarks were detected by the landmark detector of [136]. The standardized height $\mathbf{h}_c$ of a concept c is defined as the median of the annotated height values, and the weight $\mathbf{w}_c$ is formed from a combination of its inverse standard deviation and the number of scans c occurs in. This model is less biased than EXMP, but it can be subject to annotation imprecision.

5.4.2.3 Learning a Standardized Height Atlas

The atlases EXMP and ALIGN can both be further improved using an expectation maximization (EM) like procedure (c.f. Algorithm 7) starting with an initial atlas $M_{\text{seed}} = (\mathbf{h}, \mathbf{w}) \in H^{|C|} \times \mathbb{R}^{|C|}$. The algorithm requires an interpolation function h and slice annotations $(\mathbf{s})_{i,c} \in \mathbb{N}^{n \times |C|}$ for the set of detectable landmarks or concepts C within a database of n volumes. Note that there will not be a concept annotation for every $\mathbf{s}_{i,c}$, since not all volumes will show all concepts and there may be false negatives of the detector.

In iteration step t, the concept slice positions $\mathbf{s}$ are combined with the current atlas M_t to generate a set of induced matching points $\hat{P}_i = P_{i,M_{t-1}} = (\mathbf{s}_i, \mathbf{h}, \mathbf{w})$ for each training scan v_i. Applying $h_i(\mathbf{s}_{i,c})_{P_{i,M_{t-1}}}$ to determine a new standardized height value in H for each scan v_i and each concept c, we then generate a new atlas model from step 4 to 11 in a similar way as the ALIGN model: for every concept c, a new standardized height value $\hat{h}_c$ is defined as

Algorithm 7 Atlas Refinement

Input: $\mathbf{s} \in \mathbb{N}^{n\times|C|}$: slice numbers for n volumes and $|C|$ concepts (allowing missing values), $M_{\text{seed}} \in H^{|C|} \times \mathbb{R}^{|C|}$: atlas of the concepts' positions and reliabilities, h: interpolation function, ϵ_A: minimum model improvement

1: **function** ADAPT_MODEL($\mathbf{s}, M_{\text{seed}}, h$)
2: $M_0 \leftarrow M_{\text{seed}}$; $\quad \text{err}_0 \leftarrow \text{Loss}_{\text{MSE}}(\mathbf{s}, M_{\text{seed}}, h)$
3: **for** $t \in 1 : t_{\max}$ **do**
4: $\hat{P}_i \leftarrow P_{i,M_{t-1}}$ ▷ Generate matching points from $\mathbf{s}$ and M_{t-1}
5: **for** $c \in 1 : C$ **do** ▷ Generate new model using h
6: $\hat{\mathbf{h}}_c \leftarrow \text{median}_{i\in\{1,\dots,n\}} \left\{ h_i(\mathbf{s}_{i,c})_{\hat{P}_i \setminus \{p_{i,c}\}} \right\}$
7: $\hat{\mathbf{w}}_c^{\text{sd}} \leftarrow 1/\left(\text{stdev}_{i\in\{1,\dots,n\}} \left\{ h_i(\mathbf{s}_{i,c})_{\hat{P}_i \setminus \{p_{i,c}\}} \right\}\right)$
8: $\hat{\mathbf{w}}_c^{\text{occ}} \leftarrow \text{size}\left(\{i \in \{1,\dots,n\} |\, \mathbf{s}_{i,c} \text{ was found}\}\right)$
9: **end for**
10: $\hat{\mathbf{w}} \leftarrow \text{normalized } \hat{\mathbf{w}}^{\text{sd}} + 0.5 \cdot \text{normalized } \hat{\mathbf{w}}^{\text{occ}}$ ▷ Combine weights
11: $M_t \leftarrow (\hat{\mathbf{h}}, \hat{\mathbf{w}})$ ▷ File the new model
12: $\text{err}_t \leftarrow \text{Loss}_{\text{MSE}}(\mathbf{s}, M_t, h)$ ▷ Evaluate
13: **if** $\text{err}_{t-1} - \text{err}_t < \epsilon_A$ **then** ▷ No sufficient improvement
14: **return** M_{t-1} ▷ Previous model
15: **end if**
16: **end for**
17: **return** $M_{t_{\max}}$
18: **end function**

Output: New height atlas adapted to the dataset $\mathbf{s}$

the median standardized height as returned by the given interpolation function $h_i(\mathbf{s}_{i,c})_{\hat{P}_i\setminus\{p_{i,c}\}}$. Note that the matching points $p_{i,c} = (\mathbf{s}_{i,c}, \mathbf{h}_c, \mathbf{w}_c)$ are of course excluded from the re-computation of concept c (step 6).

In order to form new relevance weights $\hat{\mathbf{w}}$, we aim to attest a high relevance to concepts which mostly map to the same standardized position and which can be detected frequently. The concept weights $\hat{\mathbf{w}}_c$ (step 10) are thus constructed from a combination of the inverse standard deviation of the predicted standardized heights (step 7) and the relative number of occurrences (step 8).

The quality of the new model is evaluated using a least squares error function: $\text{Loss}_{\text{MSE}}(\mathbf{s}, M_t, h)$:

$$\text{Loss}_{\text{MSE}}(\mathbf{s}, M_t, h) = \sum_{i=0}^{n} \sum_{c\in C} (h_i(\mathbf{s}_{i,c})_{P_{i,M_t}\setminus\{(\mathbf{s}_{i,c},\mathbf{h}_c,\mathbf{w}_c)\}} - \mathbf{h}_c)^2 \tag{5.9}$$

If the new model is a significant improvement, the algorithm proceeds with the next iteration $t+1$, else it returns the previous model M_{t-1}. The maximum

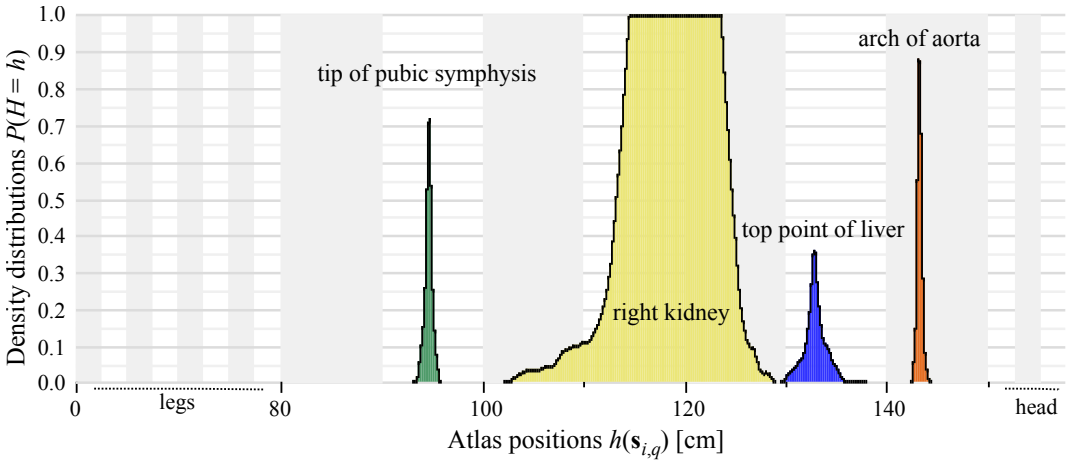


Figure 5.4: Empirical density distributions of three landmarks and the organ boundaries of the right kidney when mapped to the atlas scale via $h_i(\mathbf{s}_{i,q})$ using ABS interpolation based on an EXMP seeding.

number of iterations $t_{\max}$ should be limited in case there are unexpected effects of the optimization procedure.

5.4.2.4 Collecting the Atlas Distribution

We have now seen various ways to define a standardized height atlas $(\mathbf{h}, \mathbf{w})$ for a set of concepts C. This atlas can be used for defining matching points P_i for a volume v_i for which a set of concepts could be successfully detected. One question still remains: How to define the retrieval range $[h_{\text{lb}}, h_{\text{ub}}]$ for a concept-based query as defined in Section 5.3.2?

This range can be extracted for a given confidence threshold if a standardized height distribution $P_c(H = h)$ for every concept c is available. In fact: Algorithm 7 already uses these distributions for re-computing the atlas positions $\mathbf{h}_c$. The H coordinates $h_i(\mathbf{s}_{i,c})_{\hat{P}_i \setminus \{p_{i,c}\}}$ of the matching points generated in step 6 define the distribution of concept c.

In order to efficiently compute the quantile heights needed for deriving a standardized query height range for a concept, it would be beneficial to approximate these distributions as a distribution function like a normal distribution. When, however, generating the concept distributions for various concepts, we quickly discarded this idea, since the concepts' distributions are too dissimilar. Density distributions of three exemplary landmarks as generated by Algorithm 7 on a repository of 371 CT volumes are displayed in Figure 5.4. The figure additionally shows the density distribution of the right kidney (in yellow). The landmark and organ detectors used here were provided

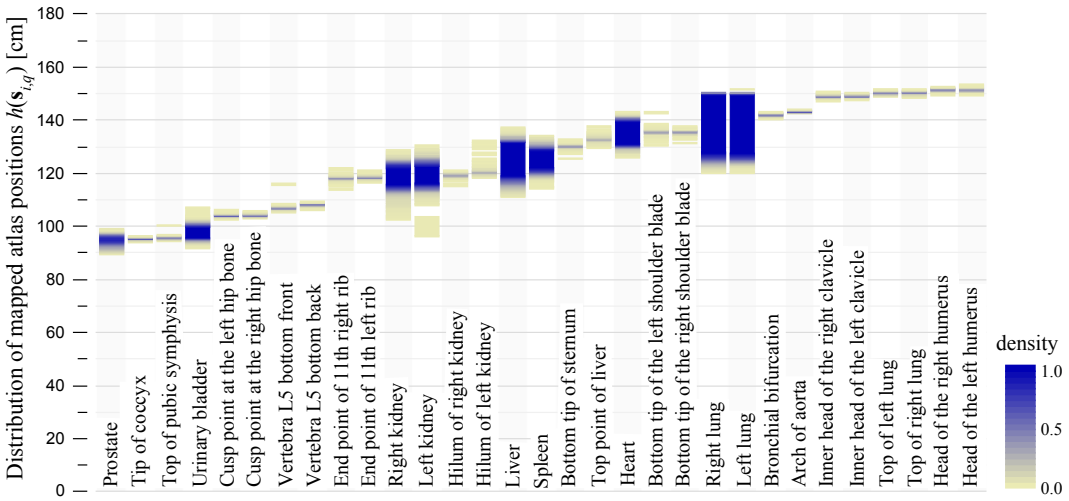


Figure 5.5: The distribution of landmark and organ boundaries when mapped to the atlas scale via $h_i(\mathbf{s}_{i,q})$ using ABS interpolation based on an EXMP seeding.

by Seifert *et al.* [136]. Evidently, the organ boundaries cannot be modeled in the same way as the landmark boundaries.

The THESEUS MEDICO framework thus supports concept distributions in the form of height distribution histograms. This flexible setting also enables the description of non-point concepts, which are required for handling organ height distributions.

Figure 5.5 displays the observed density distributions in H for the 22 landmarks and 9 additional organs' boundaries collected in a run of Algorithm 7 with the EXMP mapping as seed model M_{seed} using ABS interpolation.

5.5 Slice Localization via Instance-Based Regression

This section introduces our method for mapping a single slice into the standardized height scale H. In order to answer ROI queries on a scan v_i in the way introduced in Section 5.4, it is necessary to have at least two matching points as a set P_i. For the experiments within the THESEUS MEDICO project, we use the landmark detector of [136], which detects up to 22 landmarks within the thorax region. In roughly a third of our tested scans, however, the detector does not return any or not enough matching points. The reason why the detector failed to find landmarks were the following: The image quality is too fuzzy for the detector, the body region covered by the scan is not big enough or only

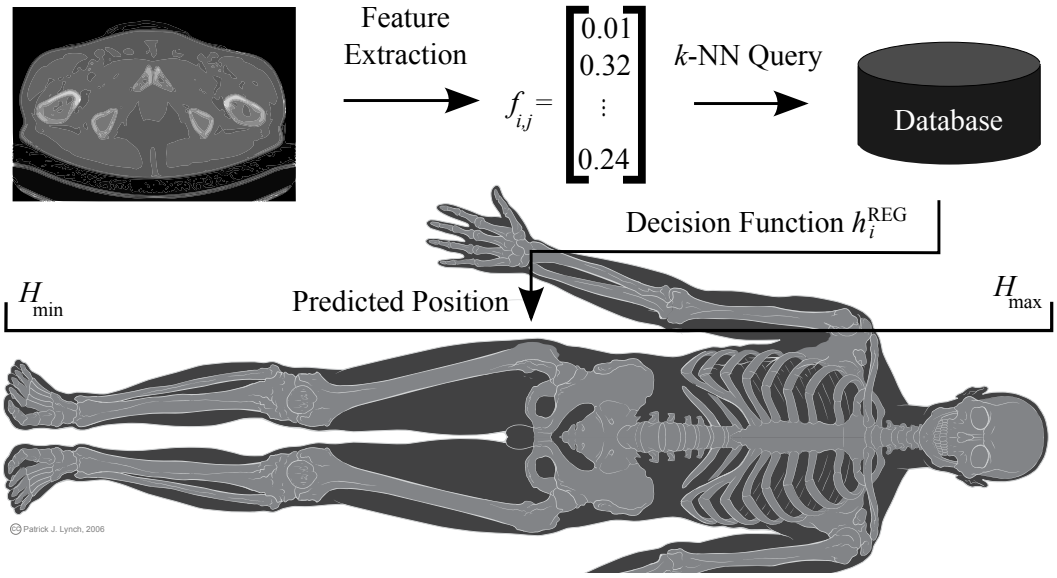


Figure 5.6: Overview of content-based matching point generation using instance-based regression. (The human model visualization is taken from Patrick J. Lynch, medical illustrator and C. Carl Jaffe, MD, cardiologist at `http://commons.wikimedia.org/wiki/File:Skeleton_whole_body_ant_lat_views.svg`)

a single slice is available. Further drawbacks of employing landmark detection for generating matching points are the complexity and availability of reliable detectors and that their runtimes are not suitable for interactive query processing. In order to allow instant query processing on arbitrary scans, a faster and more flexible method should be employed that can efficiently generate a matching point for any given slice in the queried scan.

We thus use the instance-based regression technique we proposed in Emrich *et al.* [55] for generating matching points if none are available. This approach maps a single slice $\mathbf{s}_{i,q}$ into the standardized height scale H using 2D image features for a pattern-based regression. To train the regression function, we employ a training set of CT scans T_R where each slice is labelled with a height value $h \in H$. An overview of this approach using instance-based regression can be see in Figure 5.6.

An example-based query as defined in Section 5.3.1 using this second kind of matching points transfers a scale-reduced version of the first and the last slice of the query ROI to the server and predicts the coordinates $[h_{\text{lb}}, h_{\text{ub}}]$ by instance-based regression instead of landmark-based interpolation.

5.5.1 Regression Features

We tested various feature types on their suitability for the slice localization problem. A wealth of image features has already been tested in the field of similarity search and information retrieval [44] also on medical images, [2] however, every use case requires its own specialized feature type.

In the regression problem at hand we have to discard the third dimension of the single slices' source volumes. Even though this causes a loss of information, it clearly facilitates the problem. In 2D, there are two general approaches of generating image descriptors: *global* image descriptors generate one descriptor per image, whereas *patch-based* image descriptors form a set of descriptors describing a set of sub-images, the number of which is usually unknown.

Examples for global image descriptors are thumbnails or grey-value histograms. They typically result in a single, d-dimensional feature vector can be comfortably handled by most distance measures of classical retrieval systems.

A patch-based descriptor first identifies the *image patches*, the image subregions which are to be described in more detail. Then, those patches are all represented by any given (global) image descriptor. The resulting unordered set of image descriptors forms a *multi-instance object* which must be handled by special distance measures (see also Chapter 4). The best-known patch-based descriptors are SIFT [107] and SURF [10]. Due to their high degree of detail, they have been shown to be very effective in the fields of scene-recognition and object tracking. For the usage in image retrieval, however, they tend to be too inflexible in matters of descriptiveness and too demanding in their runtime requirements.

Thus, a third group of image descriptors has evolved. Comparable to patch-based descriptors, they also compartmentalize the image into subregions, however, they do so following a strict pattern. This ensures a constant number of patches, which can be handled more easily by the retrieval query pipeline. Among this group, we received good results when applying a *spatial pyramid kernel* as in [104] for obtaining locally sensitive features.

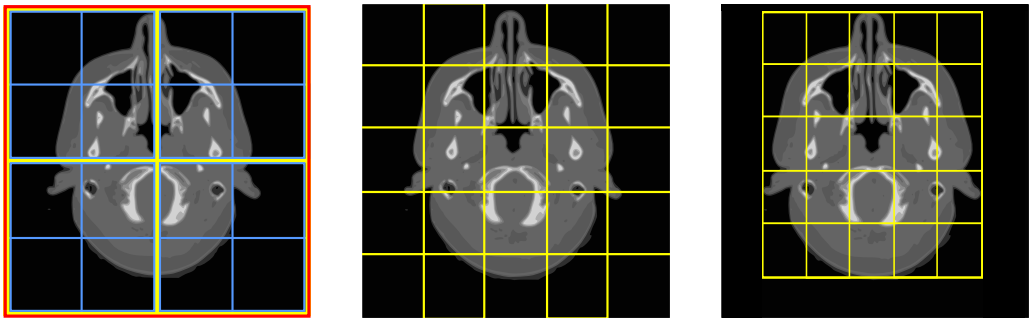
5.5.1.1 Spatial Pyramid Kernel

A spatial pyramid kernel recursively divides the image into disjoint subregions. In the original implementation [104], the division resembles the quad tree [60] partitioning: the first sub-descriptor is generated from the complete image, then it is divided into four disjoint, equally-sized subregions by a split along the x and y axes which can then be further sub-divided (c.f. Figure 5.7(a)). The resulting image features are normalized and concatenated into a single feature vector. In order to ensure a better comparability among the various levels of the partitioning, the authors of [104] additionally down-sample the

image resolution of the higher levels in the division hierarchy (featuring fewer divisions).

For our regression problem, we use the Pyramid of Histograms of Orientation Gradients (PHOG), derived from Bosch *et al.* [20]. They employ a Histogram of Oriented Gradients (HOG) [36] as image region descriptor and they skip the down-scaling step for higher levels.

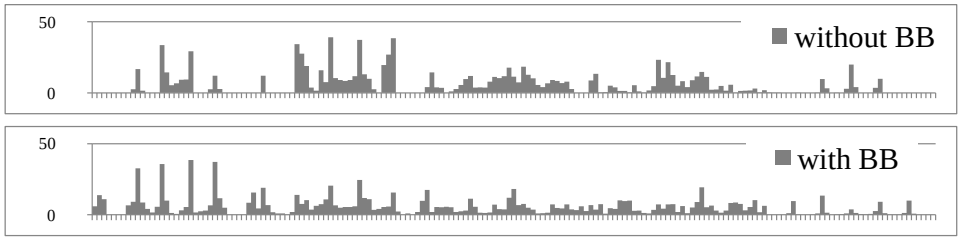
The original PHOG descriptor, however, suffers from two problems: First, the dimensionality of the resulting feature vector increases exponentially with the number of recursive sub-divisions. Since two recursive divisions as in Figure 5.7(a) showed to provide an insufficient level of detail, we would require a descriptor formed from $(1+4*(1+4*(1+4))) = 85$ regions. Second, the center split for every dimension is not well-suited for medical images, which usually bear some axial symmetries but are rarely perfectly centered. Thus, even a slight shift of the split axis can have a large effect on the resulting descriptor.



(a) Original pyramid kernel using 21 regions. [20]

(b) Modified pyramid kernel using 25 regions.

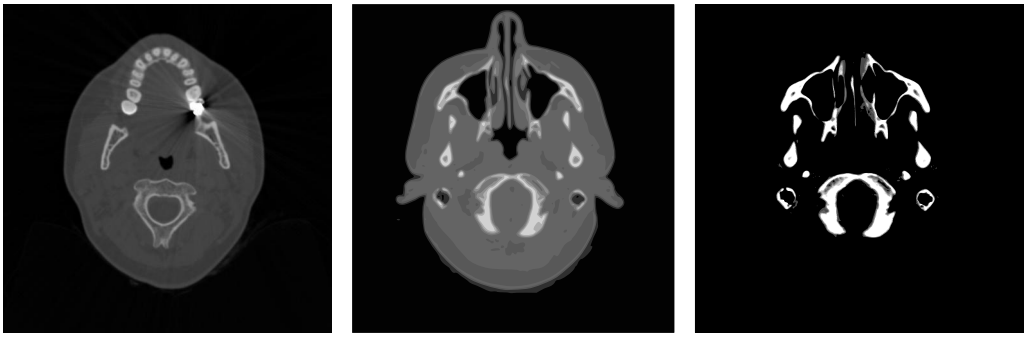
(c) Modified pyramid kernel after BB detection.



(d) PHOG descriptor for (b) (top, complete image) and (c) (bottom, using a Bounding Box). Each plot displays the feature vector resulting for the given image.

Figure 5.7: Modified spatial pyramid kernels and the impact of Bounding Box (BB) detection on the resulting feature vectors.

We thus use a modified version of the PHOG descriptor, which applies an uneven number of recursive splits per axis. This mildens the centering problem. Additionally, we found one split to be sufficiently descriptive for a



(a) Metallic artifact. (b) Full HU window. (c) Bone window.

Figure 5.8: Examples for image distortion (in (a)) and varying grey-value windows for an example head scan ((b) and (c)).

split dimension of 5 splits per axis. Thus, our new PHOG descriptor is formed using 25 disjoint, equi-sized regions as in Figure 5.7.

5.5.1.2 Preprocessing

Before applying our chosen image descriptor, however, we have to consider a couple of pre-processing steps. Since a CT scan is not a common photograph, a number of obstacle may disturb the resulting images. In [8], Barrett *et al.* present a survey on the variety of image distortions like movement artifacts or disturbances by metallic objects. Many of these sources of error are already treated by the hospital software, yet, some distortions cannot be completely removed.

Figure 5.8(a) shows an exemplary metallic artifact caused by a dental inlay. Such an artifact strongly distorts the resulting image features. Thus, our strategy is to use as many data points as possible in our regression approach in order to collect a great variety of special cases and to incorporate them into the resulting decision function.

Grey Value Windowing Additionally, the perception of medical images can be dramatically altered with the choice of the visible image value range. In DICOM images, [117] the medical image standard, the grey value range of CT scans is usually defined by the Hounsfield (HU) scale. [22] The Hounsfield scale describes the attenuation coefficient as measured by radiological images, with HU $= 0$ being calibrated to the attenuation of water and HU $= -1000$ corresponding to the attenuation of air. The resulting pixel range of a CT scan is thus normally represented with 12 bit numbers spanning the HU values in $[-1024, 3071]$. Since the human eye cannot distinguish 4096 grey values,

this means that either a lot of granularity is lost when displaying an image on the complete HU range (as in Figure 5.8(b)), or the used grey value range is directly restricted to a grey value window of interest (c.f. the bone window with HU values $[50, 530]$ in Figure 5.8(c)).

In our first experiments, we especially tested the bone window on its suitability for the slice localization task, since the bones can be assumed to be the most invariant body element. Various body regions like the abdomen, however, showed to require more information than the bone structure in order to be sufficiently resolved. Thus, we discarded the windowing approach as an additionaly pre-processing step.

Bounding Box (BB) Detection Another method of image adjustment showed to be more helpful: in Figure 5.7(b) on page 118, we see that the applied grid of the spatial pyramid kernel contains various empty patches. As already mentioned, medical images are not necessarily perfectly centered, and they frequently display differently sized patients or measured regions. It is thus sensible to only apply the image descriptor to the actual bounding box (BB) of the visible body region. This way, we can avoid an excess of empty image patches and the generated image descriptors can be better compared to each other.

In the MEDICO framework, the bounding box borders of an image are discovered from the original outer border of the image to its center. A border (upper, lower, left or right) is defined, once at least 4 % consecutive pixel rows (or columns) contain at least 20 % pixels with a HU value ≥ -600. This way, regions containing mostly air are excluded from the image feature generation.

The effect of this BB detector is visualized in Figure 5.7(c). The resulting feature vector in Figure 5.7(d) when using the detector strongly differs from the feature vector generated without the BB detector.

5.5.1.3 Used Image Patch Features

Among a number of tested image patch descriptors, two have proven to be especially well-suited for the slice localization task.

Haralick Texture Features Our first patch representation are Haralick texture features. [79] We compute all 13 Haralick features for five different window distance values $(1, 3, 5, 7, 11)$. The resulting patch feature vector thus contains $13 \cdot 5 = 65$ features. Haralick *et al.* [79] already stated that some of the features are highly correlated. Hence, the resulting feature vector contains a lot of redundant information. In our original publication, we tackled this effect by applying Principal Component Analysis (PCA) on the resulting image features.

Our current experiments show that the more advanced feature transformation approach using Relevant Component Analysis (RCA), which will be described in Section 5.5.3, is even better suited for eliminating redundant information.

Histograms of Oriented Gradients (HOG)s The second patch representation is a Histogram of Oriented Gradients (HOG) [36] as also used in the SIFT descriptors [107]. For obtaining a HOG, we first detect all relevant edges with a Canny operator [24] and then determine the gradients' angles for all pixels featuring a relevant edge. The resulting angles are then summarized to a HOG consisting of 7 equally-sized bins.

By aggregating the various image patches of a spatial pyramid kernel as proposed in Section 5.5.1.1, we receive our version of a Pyramid of Histograms of Orientation Gradients (PHOG), differing only in the kernel setting from the approach proposed by Bosch *et al.* in [20].

Since our query algorithm requires multiple online feature extraction steps per query, we down-scale the images before feature extraction for speeding up feature generation. We use a 100x100 pixels resolution for the PHOG descriptors and 200x200 pixels for the spatial pyramid of Haralick features.

5.5.2 Instance-Based Regression

In this section, we explain the use of our proposed regression scheme. We will first deal with the regression problem using one object representation form and then extend it to a multi-represented or multi-modal regression scheme.

5.5.2.1 Single-Feature Regression

We denote by $F(\mathbf{s}_{i,j}) : \mathbb{N} \to \mathbb{R}^d$ the feature transformation of the $\mathbf{s}_{i,j}^{\text{th}}$ slice of volume v_i to the final, d-dimensional feature space. Our task is thus to map a d-dimensional feature vector $f_{i,j}$ corresponding to the j^{th} slice of the scan v_i to the height model H. The problem $h : \mathbb{R}^d \to H$ could be handled with any given regression function. However, in our experiments, the majority of standard regression methods either required extensive training times on our large datasets of up to 900 000 training examples or they did not yield an acceptable prediction quality.

Therefore, we employ an instance-based approach to regression: we determine the k-nearest neighbors (k-NN) of the given feature vector in the training set T_R, consisting of image features $r \in \mathbb{R}^d$ with existing standardized height labels $h(r) \in H$, w.r.t. Euclidean distance. In order to avoid distorting effects of self-matches, the training set T_R for any query slice $\mathbf{s}_{i,j}$ in scan v_i may not contain any of the other slices of v_i. This leave-one-out scheme is necessary due

to a high self-similarity of slices originating from the same scan. Afterwards, the height of slice $\mathbf{s}_{i,j}$ is predicted using the following decision function:

$$h_i^{\mathrm{REG}}(\mathbf{s}_{i,j}) = \mathrm{median}\left\{h(r) \mid r \in T_R \wedge r \in k\text{-NN of } F(\mathbf{s}_{i,j})\right\} \ . \tag{5.10}$$

The precision of $h_i^{\mathrm{REG}}(\mathbf{s}_{i,j})$ can be further improved by allowing only slices $r \in T_R$ originating from disjunct volumes to form the k-NN candidate set. Again, this effect is due to high within-volume similarities. Constraining the contribution of each template volume in T_R to at most one target value decreases the probability that the matches contained in the candidate set are only similar to the query slice $\mathbf{s}_{i,j}$ due to similarities between the parenting volumes and not due to an actual height accordance.

5.5.2.2 Multi-Represented Regression

This instance-based regression setting also offers interesting possibilities for the usage of multiple image features. We can directly extend the learner to base its prediction on a mixture of l input spaces. In our application, we face the problem that certain feature representations are less suited for certain regions of the body, while they provide excellent results in certain other regions. PHOG descriptors, for instance, are well-suited for areas with a rich bone structure resulting in various edges. However, they are less descriptive in the abdomen area.

Thus, combining two or more image representations with a common ensemble approach is not very promising. Neither is the direct concatenation of various representations into a single, combined feature vector. Instead, we propose to use an automatic *representation selection* approach.

We want to select the image feature representation that most probably offers the best prediction quality for the current input image. Thus, we first predict the position of the current input slice in each of the l available feature representations $F^f(\mathbf{s}_{i,j})$ with $f \in \{1, \ldots, l\}$ and afterwards assign a reliability value to each representation's prediction. In instance-based learning, the reliability is closely coupled with the variance of the positions within the k-NN candidate set of each representation

$$\mathrm{var}(f, \mathbf{s}_{i,j}) = \mathrm{var}\left\{h(r) \mid r \in T_R^f \wedge r \in k\text{-NN of } F^f(\mathbf{s}_{i,j})\right\} \ , \tag{5.11}$$

with T_R^f containing training elements of representation f of the multi-represented dataset T_R.

For large variances, the k-nearest neighbors are localized in different parts of the body and thus, representation f does not yield a consistent statement about the slice's position. If the labels of the k-nearest neighbors point to

similar positions, $\mathrm{var}(f, \mathbf{s}_{i,j})$ is small and representation f offers a coherent prediction.

Our new decision function for multi-represented objects is thus formed as the prediction corresponding to the representation $f_{i,j}^{\min}$ providing the smallest positional variance for a given target slice $\mathbf{s}_{i,j}$ of volume v_i:

$$f_{i,j}^{\min} = \underset{f \in \{1,\ldots,l\}}{\operatorname{argmin}} \{\mathrm{var}(f, \mathbf{s}_{i,j})\} \ , \tag{5.12}$$

$$h_i^{\mathrm{REG}}(\mathbf{s}_{i,j}) = \mathrm{median} \left\{ h(r) \mid r \in k\text{-NN of } F^{f_{i,j}^{\min}}(\mathbf{s}_{i,j}) \text{ in } T_R^{f_{i,j}^{\min}} \right\} \ . \tag{5.13}$$

Note that the same restrictions apply for the k-NN candidate set derived from the various representation datasets T_R^f as for the single-feature prediction decision rule (5.10). Furthermore, the k-NN queries required for a prediction only have to be performed once per representation and since the different representations are considered to be independent of each other, they can be computed in parallel.

5.5.3 Query Acceleration by using a Spatial Index

Although instance-based regression does not suffer from extensive training times, the cost for large example datasets has to be spent at prediction time. However, the prediction rule does only require to process a single kNN query per representation. This calls for the use of optimization methods for this well-examined problem.

In order to allow efficient query processing, we transform the high-dimensional feature space of the proposed image features into a lower-dimensional space which can be indexed by suitable spatial index structures. For the tests of this work, we use an X-Tree [15], which is well-suited for data of medium dimension.

We reduce the input dimensionality d in a supervised way employing Relevant Component Analysis (RCA) [7] with the goal of maintaining the principal information of the original feature vectors $r \in \mathbb{R}^d$. RCA transforms the data into a space minimizing the co-variances within subsets of the data, which are supposed to be similar, the so-called *chunklets*.

Chunklets can be defined by matching a set of class labels or by using clusters. In our setting, we chose to use a purely supervised chunking procedure: we sort the data points used for training the feature transformation according to their height labels and retrieve a pre-defined number (150 chunklets performed well) of equally-sized data subsets.

We also tested alternative grouping approaches. Usually, our training datasets show varying support for the different body regions. Thus, using

fixed bin widths (e.g. 5 mm steps) for grouping the height labels cannot be recommended. We also tried to further improve our grouping procedure by applying an additional similarity-based subclustering to chunklets already generated in a supervised manner. Our experiments, however, did not indicate any significant improvement of this approach.

Let us note that in case of high-dimensional datasets with a relatively small number of training instances, the chunklet sizes generated by our proposed approach may become too small for retrieving an informative within-chunklet covariance matrix. In our system, we require at least 10 instances per chunklet. If this number cannot be ensured, either the number of chunklets needs to be decreased, or we can artificially increase the number of chunklets by allowing chunklets to overlap with their neighboring chunklets according to their height label order. This workaround works fine up to a certain degree of information recycling, however, it never became necessary in the experiments published in this thesis.

For our datasets, using a 10-dimensional feature representation turned out to be a viable trade-off between prediction time and accuracy. On the average, a query took 20 ms while yielding an average prediction error of only 1.98 cm.

5.5.4 Generating Reliability Weights

When using positions $\hat{h}_{i,j}^{\mathrm{REG}}$ computed with $h_i^{\mathrm{REG}}(\mathbf{s}_{i,j})$ as matching points for answering ROI queries, we are also interested in how reliable they are. One way to determine a position's reliability was just presented in Section 5.5.2.2. We could again use the variance of the k-nearest neighbors, with a low variance indicating a reliable prediction. [55] However, in our setting, the best predictions could often be observed with $k = 1$ or $k = 2$. Since building a deviation on 1 or 2 samples does not make any sense, we had to develop an alternative approach for approximating the prediction quality.

Therefore, we perform an additional pre-processing step assigning a weight to all instances r in the training database T_R. The weight $w(r)$ of instance r is determined in a leave-one-out run of $h_i^{\mathrm{REG}}(r)$ on T_R: we derive a predicted height value $\hat{h}_r^{\mathrm{REG}}$ and compare it to the true position $h(r)$. Finally, we determine the weight $w(r)$ as follows:

$$w(r) = 0.1 / \left(0.1 + \left|\hat{h}_r^{\mathrm{REG}} - h(r)\right|\right) \tag{5.14}$$

The reliability of a predicted value $\hat{h}_i^{\mathrm{REG}}(\mathbf{s}_{i,j})$ is now approximated by the average weight $w(r)$ over all k-nearest neighbors r of the queried instance $\mathbf{s}_{i,j}$.

5.6 An Online Retrieval Algorithm

In the following, we define a method for retrieving an ROI in a volume v_i for which no matching points are yet available. As mentioned before, the first step of an ROI query is to determine the query interval $[h_{\text{lb}}, h_{\text{ub}}]$ in the standardized space H. Depending on the query type, this information may be retrieved from an example-based query via landmark-driven height interpolation (Section 5.4) or instance-based regression as proposed in the previous section. Alternatively, $[h_{\text{lb}}, h_{\text{ub}}]$ can be determined in a concept-based query (Section 5.3.2) by looking up an anatomically meaningful body part or a set of anatomic regions as query interval from a given atlas.

Once such a query interval is defined, we need to collect a set of at least two matching points $P_i \subseteq \{0, \ldots, z(i) - 1\} \times H \times \mathbb{R}$ for being able to interpolate from the standardized height space H to the volume space of a slice $\mathbf{s}_{i,j} \in \mathbb{N}$ of volume v_i with an interpolation function $s_i(h)_{P_i}$. If a set of matching points is already given – e.g. via a successful landmark retrieval which could be mapped to a standardized height atlas – $s_i(h)_{P_i}$ can be directly applied on h_{lb} and h_{ub}. The resulting slice numbers $\hat{s}_{\text{lb}} = s_i(h_{\text{lb}})_{P_i}$ and $\hat{s}_{\text{ub}} = s_i(h_{\text{ub}})_{P_i}$ define the result range $(\hat{s}_{\text{lb}}, \ldots, \hat{s}_{\text{ub}})$ of ROI slices in v_i.

In many cases, however, no landmark detector is available or it does not return the minimally required number of two landmarks. If this is the case, we propose to use the instance-based height regression approach $h_i^{\text{REG}}(s)$ of Section 5.5 for manually generating matching points.

5.6.1 ROI Query Processing

Naturally, the quality of the mapping $s_i(h)_{P_i}$ directly depends on the quality of the matching points $p \in P_i$. Having a large set of matching points increases the mapping quality because it increases the likelihood that reliable matching points being close to h_{lb} and h_{ub} are available. Furthermore, having more matching points decreases the impact of low-quality matching points. However, increasing the amount of matching points is connected with generating costs for feature transformation, dimension reduction and regression.

Thus, we want to employ a minimal number of matching points while attaining a high interpolation quality. The core idea of our method is to start with a minimal set of matching points and to measure the quality of the induced mapping function. As long as this quality is significantly increasing, we select slices in the query scan and induce additional matching points using the regression method $h_i^{\text{REG}}(s)$ proposed in Section 5.5. This process is illustrated in Figure 5.9.

Once a result range $(\hat{s}_{\text{lb}}, \ldots, \hat{s}_{\text{ub}})$ has been determined, we validate its qual-

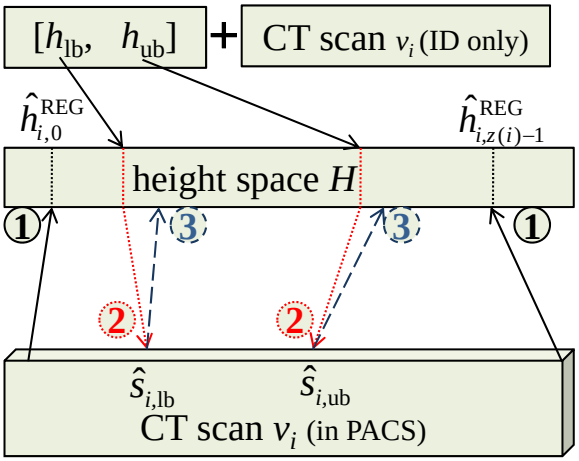


Figure 5.9: First steps of Algorithm 8: a query range $[h_{\text{lb}}, h_{\text{ub}}]$ is to be found in a scan v_i. In the initial step (1), the seed slices forming the matching points P_i are selected and mapped to H with h_i^{REG}. In step (2), P_i is used for interpolating a result range $(\hat{s}_{\text{lb}}, \ldots, \hat{s}_{\text{ub}})$ in v_i. Step (3) validates this result range using h_i^{REG} and decides whether a new range should be tested.

ity using the same mechanism of manually generating matching points via regression $h_i^{\text{REG}}(\mathbf{s})$:

Definition 28 (Validation Error) *The* validation error *of a prediction* $\hat{s}_c$ *for a value* $\mathbf{h}_c$ *is defined as* $|h_i^{\text{REG}}(\hat{s}_c) - \mathbf{h}_c|$.

Since $h_i^{\text{REG}}(\hat{s}_c)$ is fixed during query processing, the only possible way to reduce the error is to improve the quality of the matching points. This can happen by either updating their weights $\mathbf{w}_j$ or by adding further matching points. Even though it is sensible to update weights in special cases, the core component of our algorithm involves the second improvement variant.

For a given query interval $[\mathbf{h}_{\text{lb}}, \mathbf{h}_{\text{ub}}]$, our method proceeds as follows (see also Algorithm 8): We select g equally-spaced seed slices $\mathbf{s}_i \subset \{0, \ldots, z(i)-1\}$ to generate an initial set of g matching points by predicting their positions as $\hat{\mathbf{h}} \in H^g$ using instance-based regression $h_i^{\text{REG}}(\mathbf{s}_i)$. After deriving the weights $\hat{\mathbf{w}} \in \mathbb{R}^g$ obtained in the regression procedure as described in Section 5.5.4 we can induce an initial set of matching points $P_i = (\mathbf{s}_i, \hat{\mathbf{h}}, \hat{\mathbf{w}})$, thereby completing the $\text{INIT}(v_i, h_i^{\text{REG}})$ step of Algorithm 8. We are now free to make our first prediction of the result range.

We interpolate $\hat{s}^*_{\text{lb}} = s_i(h_{\text{lb}})_{P_i}$ and $\hat{s}^*_{\text{ub}} = s_i(h_{\text{ub}})_{P_i}$ in the query scan using the current set of matching points P_i with s_i, one of the interpolation methods of Section 5.4. Next, we employ $h_i^{\text{REG}}(\mathbf{s})$ on the borders of the predicted slice

Algorithm 8 Online ROI Query

Input: v_i: Query volume, $[h_{\text{lb}}, h_{\text{ub}}]$: query interval in H, h_i^{REG}: height regression function $\mathbb{N} \to H$, s_i: interpolation function $H \to \mathbb{N}$, ϵ: tolerated result range deviation

1: **function** ONLINE_ROI_QUERY($v_i, [h_{\text{lb}}, h_{\text{ub}}], h_i^{\text{REG}}, s_i, \epsilon$)
2: $P_i = (\mathbf{s}_i, \hat{\mathbf{h}}, \hat{\mathbf{w}}) \leftarrow \text{INIT}(v_i, h_i^{\text{REG}})$ ▷ Initialize P_i
3: $\{\text{err}_{\text{lb}}, \text{err}_{\text{ub}}\} \leftarrow \{\infty, \infty\}$ ▷ Errors for lb and ub
4: $\{\hat{s}_{\text{lb}}, \hat{s}_{\text{ub}}\} \leftarrow \text{NULL}$ ▷ Resulting slice numbers
5: **while** $\text{err}_{\text{lb}} > \epsilon$ **or** $\text{err}_{\text{ub}} > \epsilon$ **do**
6: $\{\hat{s}^*_{\text{lb}}, \hat{s}^*_{\text{ub}}\} \leftarrow \{s_i(h_{\text{lb}})_{P_i}, s_i(h_{\text{ub}})_{P_i}\}$ ▷ Interpolation
7: $\{\hat{h}^{\text{REG}}_{\text{lb}}, \hat{h}^{\text{REG}}_{\text{ub}}\} \leftarrow \{h_i^{\text{REG}}(\hat{s}^*_{\text{lb}}), h_i^{\text{REG}}(\hat{s}^*_{\text{ub}})\}$ ▷ Regression
8: $\{\text{err}^*_{\text{lb}}, \text{err}^*_{\text{ub}}\} \leftarrow \left\{\left|\hat{h}^{\text{REG}}_{\text{lb}} - h_{\text{lb}}\right|, \left|\hat{h}^{\text{REG}}_{\text{ub}} - h_{\text{ub}}\right|\right\}$
9: **if** $\text{err}_{\text{lb}} > \text{err}^*_{\text{lb}}$ **then** ▷ New lower bound
10: $\hat{s}_{\text{lb}} \leftarrow \hat{s}^*_{\text{lb}}$; $\text{err}_{\text{lb}} \leftarrow \text{err}^*_{\text{lb}}$
11: **end if**
12: **if** $\text{err}_{\text{ub}} > \text{err}^*_{\text{ub}}$ **then** ▷ New upper bound
13: $\hat{s}_{\text{ub}} \leftarrow \hat{s}^*_{\text{ub}}$; $\text{err}_{\text{ub}} \leftarrow \text{err}^*_{\text{ub}}$
14: **end if**
15: Get weights $\hat{w}^{\text{REG}}_{\text{lb}}, \hat{w}^{\text{REG}}_{\text{ub}}$ of new matching points
16: $P_i.\text{APPEND}\Big((\hat{s}^*_{\text{lb}}, \hat{h}^{\text{REG}}_{\text{lb}}, \hat{w}^{\text{REG}}_{\text{lb}}), (\hat{s}^*_{\text{ub}}, \hat{h}^{\text{REG}}_{\text{ub}}, \hat{w}^{\text{REG}}_{\text{ub}})\Big)$ ▷ Extend P_i
17: **end while**
18: **return** $(\hat{s}_{\text{lb}}, \ldots, \hat{s}_{\text{ub}})$
19: **end function**

Output: Result range $(\hat{s}_{\text{lb}}, \ldots, \hat{s}_{\text{ub}})$

range $(\hat{s}^*_{\text{lb}}, \ldots, \hat{s}^*_{\text{ub}})$ and determine the validation error estimate (step 8). If the lower or upper bound ($\hat{s}^*_{\text{lb}}$ or $\hat{s}^*_{\text{ub}}$) has been improved compared to the minimal error observed so far, we update the corresponding resulting slice number variables for the lower or upper bound ($\hat{s}_{\text{lb}}$ or $\hat{s}_{\text{ub}}$). Finally, we augment the set of matching points P_i by the regression prediction $h_i^{\text{REG}}(s)$ for the boundaries of the target range $\hat{s}^*_{\text{lb}}$ and $\hat{s}^*_{\text{ub}}$. The algorithm terminates if the validation errors on both sides of the target range is less than the tolerance threshold ϵ. In our implementation, ϵ is set to the expected regression error of the training database T_R.

5.6.2 Handling Special Cases

For simplicity reasons, this algorithm omits a number of special cases. Since the derivation of matching points via regression is expensive due to the over-

head of feature generation, the algorithm has to ensure that no slice number of v_i is tested multiple times. The MEDICO implementation thus tests a neighboring slice if a slice prediction $\hat{s}^*_{\text{lb}}$ or $\hat{s}^*_{\text{ub}}$ does not occur for the first time in step 6.

Moreover, the search procedure should stop, once there is no more change to the predicted slice bounds $\hat{s}^*_{\text{lb}}$ or $\hat{s}^*_{\text{ub}}$, as this usually means that the volume is not well enough resolved for perfectly matching the target range.

It is also beneficial to test for both bounds whether a new matching point generated for the opposite bound is better suited w.r.t. the adapted validation errors $\left|\hat{h}^{\text{REG}}_{\text{lb}} - h_{\text{ub}}\right|$ or $\left|\hat{h}^{\text{REG}}_{\text{ub}} - h_{\text{lb}}\right|$. This test is helpful in case of a strongly erroneous initialization or if the query range is especially small. Additionally, if only one bound has been established in an acceptable quality, but it remains stable over a couple of iterations, one should refrain from trying to further improve this bound by costly regression calls and only update the opposite bound.

Furthermore, a number of exceptions should be handled: both s_i and h^{REG}_i can be mapped outside of their allowed ranges. In the case of s_i, this may be an indication that the query range is not contained in the volume. Repeated range violations should thus terminate the algorithm with an indication of a mismatch or a partial match. If $h^{\text{REG}}_i(s)$ goes astray, this can either be noise in the regression function or it can be a reason for down-weighting the current set P_i and for seeking further matching points. We specifically down-weight the set of existing matching points, if the predicted slice number for the lower bound $\hat{s}^*_{\text{lb}}$ is larger than the predicted slice number of the upper bound $\hat{s}^*_{\text{ub}}$.

In our implementation, a down-weighting step simply multiplies all weights $\mathbf{w}_j$ of P_i with a constant factor 0.1. This way, the collected information – some of which may be accurate – is not completely discarded, but it has a smaller effect on any future predictions.

5.7 Experimental Validation

In the following, we present the results of our experimental evaluation by measuring the quality of the retrieval system and by demonstrating the improved query time of our complete system. All of our experiments were performed on subsets of a repository of 4 479 CT scans provided by the Imaging Science Institute of the University Hospital Erlangen for the THESEUS MEDICO project. The scans display various subregions of the human body, starting with the coccyx and ending with the top of the head.

For generating a ground truth of height labels, we used the landmark detector of [136] annotating each scan with up to 22 landmarks. This restricted

the dataset to 2 476 scans where enough landmarks could be detected. The complete repository contains more than a million single CT slices comprising a data volume of 520 GB.

The MEDICO prototype is implemented in JAVA 1.6 and stores the scans and their annotations in a MySQL database. To simulate the distributed environment of a radiology center, we employed the LAN and the workstations in our lab consisting of common workstations of varying type and configuration being connected by a 100 Mb Ethernet.

5.7.1 Atlas Accuracy

For validating the quality of our atlas generation method, we evaluate all combinations of atlas seedings and interpolation functions.

We use 371 scans of our repository providing at least 20 landmarks for training a reliable atlas. The atlas, built within a 4-fold cross-validation setting, is validated in a leave-one-out testing scheme for each landmark. We measure the average localization errors $\left|\mathbf{h}_j - h_i(\mathbf{s}_{i,j})_{P_i \setminus p_j}\right|$ and $\left|\mathbf{s}_{i,j} - s_i(\mathbf{h}_j)_{P_i \setminus p_j}\right|$ of a single landmark in volume v_i forming the matching point $p_j = (\mathbf{h}_j, \mathbf{s}_{i,j}, \mathbf{w}_j)$ when predicted with the interpolation functions h_i or s_i by using only the remaining matching points $P_i \setminus p_j$. The results are displayed in Figure 5.10 for the atlases EXMP and ALIGN (dashes), as well as for atlases refined with Algorithm 7 (boxes). As a baseline we also added a third seed atlas RAND, consisting of 22 random standardized height positions $\mathbf{h}$ and random weights $\mathbf{w}$, drawn from a uniform distribution. We transformed the error values to the range of the initial EXMP mapping covering a height range of 0-175 cm by determining the scaling factor via linear interpolation between the different atlases. This enables us to provide a meaningful and adequate comparison of the different mappings in a real-life-scale.

Figure 5.10 shows that ABS and LIN interpolation are comparably successful on trained atlases with average errors around 1 cm. Additionally, the locally more adaptive mappings SQR and RBF (and other, more complex interpolation approaches not shown here) have a stronger susceptibility to outliers. Atlases based on EXMP seedings converge to comparable errors as those based on ALIGN seedings. However, the EXMP seedings really profit more from the atlas refinement algorithm than the ALIGN seedings, which were hardly improved at all (cf. the original seed error displayed in dashes). Note that there are no original errors for the random (RAND) seedings, as it does not offer an anatomically realistic model and thus, it cannot be properly evaluated. The models resulting from Algorithm 7 for ABS and LIN interpolation, however, are anatomically sound and competitive w.r.t. the anatomically meaningful seedings ALIGN and EXMP. In contrast, the more complex interpolation approaches

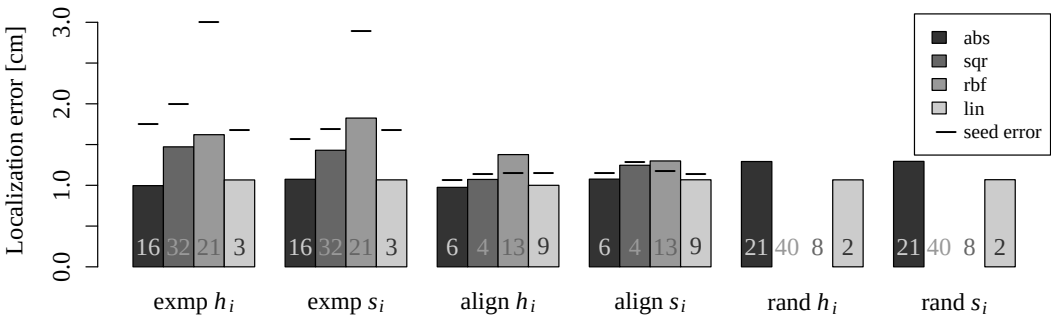
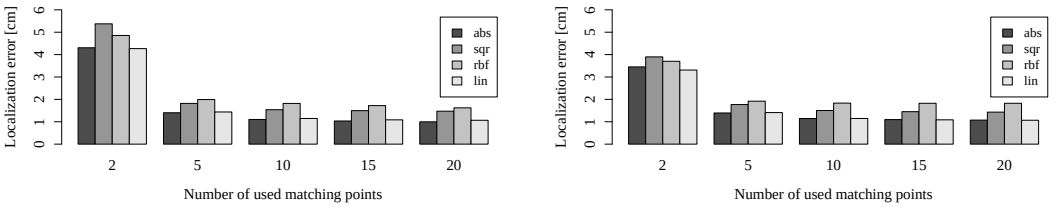


Figure 5.10: Leave-one-out localization errors [in cm] of h_i (scan to atlas) and s_i (atlas to scan positions) for the seeding approaches EXMP, ALIGN and RAND using 20 matching points. The number of iterations required for training Algorithm 7 is displayed at the bottom of the bars.



(a) Deviation for h_i (scan to atlas)

(b) Deviation for s_i (atlas to scan positions)

Figure 5.11: Leave-one-out localization errors [in cm] when using a variable number of matching points in h_i and s_i for the seeding approach EXMP.

SQR and RBF failed to generate a useful mapping from the RAND seeding: they only reordered the landmarks into disjoint sub-clusters and got stuck in local minima. Thus, the bars for this experimental setting are missing.

Figure 5.10 also displays the number of iterations required by Algorithm 7. The number of iterations vary rather strongly from 2 to 40 for the various seedings. Training times accordingly vary between less than two 2 and 17 minutes. In general, the ALIGN training times were the shortest, except for linear interpolation (LIN). LIN usually converged the earliest with by far the worst training error (Loss_{MSE}, not shown here). Since, however, the localization error displayed in this section does not punish strong outliers with a quadratic term, LIN is still competitive to the other interpolation approaches.

In fact, the most surprising result of these experiments is the good performance of the simple weighted linear interpolation LIN. Let us note that for the pre-dominant spacing in our database of 5 mm, an average precision of 1 cm is only a deviation of two slices.

We now examine how many matching points are needed. We use the same

experimental setting as before and only restrict the number of allowed matching points used for the leave-one-out test. Figure 5.11 demonstrates the effect of the number of available matching points for the EXMP models. Other seedings show the same trend: a strong decrease of the prediction errors when using five instead of two matching points. Using even more matching points results in further but smaller prediction improvements. A number of five matching points appears to be a good compromise between the effort of generating matching points and a good localization.

5.7.2 Validation of Regression Queries

In the following section, we first examine the used image features on their suitability for k-NN regression. Afterwards, we describe the beneficial effects of reducing the original image feature space using RCA.

5.7.2.1 Regression Ground Truth

For our regression experiments, we have to provide height labels $h(r)$ as ground truth for all entries of the required regression database, i.e. for each scan in the training dataset T_R. There are two methods for generating these labels. The first is to manually mark the highest and the lowest point in all scans of a database and to linearly interpolate the height values. [55] This procedure has been used for generating the ALIGN height atlas evaluated in the previous section. We refer to this method as *manual labelling*.

Since instance-based regression profits from a larger database, we also use an automatic labelling method. It assigns height labels to the slices of a volume with the interpolation model $h : \mathbb{N} \to H$ introduced in Section 5.4 as (5.3) based on detected landmarks which can be mapped to standardized positions as matching points. In our regression experiments, we use ABS interpolation with an atlas refined from the ALIGN atlas with Algorithm 7. The matching point types are again the 22 landmarks of [136], marking meaningful anatomical points. They could be detected in 2 526 of our CT scans. These landmarks are time-expensive to compute and their computation fails in the remaining 1 953 scans of our dataset. We will refer to the height labels generated with this interpolation procedure as *automatic labelling*.

5.7.2.2 Regression Quality

Single Feature Performance In our first test, we measure the regression performance of the original image descriptors, which have not yet been transformed by RCA. We first examine a manually annotated dataset of 60 CT scans of 45 patients with a total of 27 646 slices. The average leave-one-out

prediction errors for the tested feature types are displayed in Table 5.2. Let us note that leave-one-out in these experiments means that only slices from other scans than the query scan are accepted as k-nearest neighbors in order to exclude distorting effects of within-scan similarities.

We tested both the Haralick texture features and the Pyramid of Histograms of Orientation Gradients (PHOG) with our version of a 5×5 spatial pyramid kernel. As a baseline, we also tested thumbnail descriptors, representing the interpolated pixel values after re-scaling the input images to a grid of 16×16 pixels.

Additionally, in grey, we display the results for a new 2D image descriptor especially developed by our group for the purpose of slice regression. [69] This new image descriptor results in feature vectors of length 384 and it specifically targets the pixel distributions characteristic for bone and air material using radial image compartments. The performance of this new descriptor is clearly better than that of any of the other image features. However, the time required to generate the new descriptor is more than 25 times the runtime required for generating the PHOG descriptor, which is also rather expensive with about 0.02 ms per slice. In the interest of a fast query processing, any further experiments generated with this new descriptor are thus omitted from this thesis even though they usually also show a better accuracy. As soon as the feature generation can be sufficiently sped-up, we will exchange the current retrieval pipeline in the MEDICO prototype for this improved descriptor.

Table 5.2 shows that the best results among the remaining descriptors could be generated with the PHOG image descriptor after applying the preprocessing step of bounding box detection (BB). Furthermore, BB detection has a much more favorable effect on the PHOG and Thumbnail features than on the Haralick descriptors. The Haralick texture features thus appear to be less dependent on the exact geometry of the resulting spatial pyramid kernel than the gradient-based PHOG features or the grey-value-based Thumbnail features.

PHOG image features thus appear to be best-suited for the regression task at hand. The strong error variations between different kinds of preprocessing and different k-nearest-neighbor parameters, however, suggest that the performance of any feature type is rather dependent on the dataset. Furthermore, the relatively strong performance of the simple Thumbnail features may raise doubts about the suitability of the selected image features. As an additional base-line image feature, we thus also tested grey-value histograms without any spatial image information. The leave-one-out error for those histograms ranges between 11 and 15 cm and is thus considerably worse than the above results. When testing alternative datasets, we reached the same conclusions about the suitability of the selected image features.

Still, our group continues the research for custom-designed image descrip-

Table 5.2: Leave-one-out validation (LOO) errors [in cm] of k-NN slice mapping $h_i^{\text{REG}}(\mathbf{s}_{i,j})$ for various image feature types on a manually-labelled dataset of 60 CT scans with 27 646 slices. Each feature type was tested once without (noBB) and once with the preprocessing step of bounding box detection (BB). Best results are marked in bold.

	$k=1$		$k=2$		$k=3$		$k=5$	
Feature type	noBB	BB	noBB	BB	noBB	BB	noBB	BB
Haralick	5.52	4.38	5.37	4.53	5.12	4.29	5.08	4.18
PHOG	5.91	3.49	6.19	3.63	6.01	**3.35**	5.93	3.55
Thumbnails	7.38	5.12	7.05	5.08	6.60	4.75	9.25	6.86
Ref. [69]	3.66	3.16	3.67	2.95	3.52	2.85	3.91	**2.82**

tors for our instance-based regression task. One example is the expensive descriptor of Ref. [69] mentioned previously.

Combined Feature Performance When testing k-nearest-neighbor parameters between 1 and 5 on the 60 manually annotated CT scans of the previous experiments, we found varying values of k to perform the best for different image representations. This observation collides with the requirements of our multi-represented regression approach presented in Section 5.5.2.2, which uses the variances in the candidate sets of the single features' predictions in its decision function. Even though it is possible to determine the reliability of the separate representations' predictions with different k parameters, representations requiring large k would obviously be underpriviledged due to higher variances of the candidate sets.

Thus, we use a global k-parameter for our feature combination approach. We combine the information of three image representations: the Haralick features are extracted without bounding box preprocessing (noBB) and the PHOG and Thumbnails features are extracted from detected bounding boxes (BB). Note that even though the Haralick features generally perform better when using the preprocessing step (BB), the version omitting the preprocessing step is more stable in some body regions not perfectly covered by the other descriptors. The combination results in an overall leave-one-out error of 2.89 cm for the best $k = 4$, which is lower than the errors of any of the single representations listed in Table 5.2 (again discarding the results of Ref. [69], which also profits from our combination approach).

Figure 5.12 explains this effect by discretizing the average error per annotated body region. Each interval on the x-axis represents an annotated body

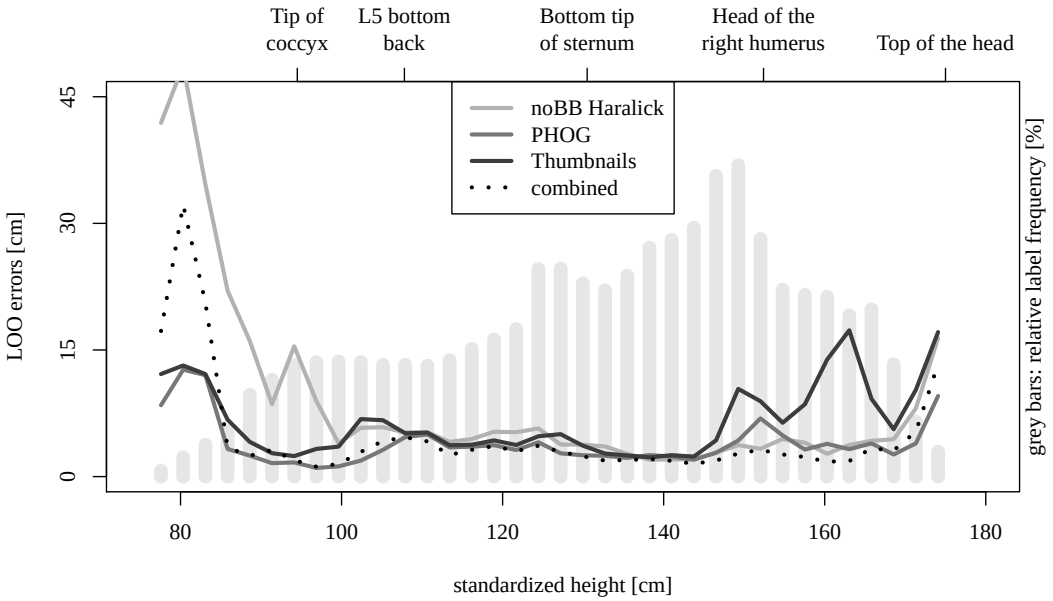


Figure 5.12: Effect of multi-represented regression: the advantages of the single image representations in different body regions are combined.

range of 3 cm. In order to facilitate the mapping of the x-axis to the body's anatomy, the top scale lists some landmark positions as expected in the atlas of the MELANIX height space. The grey bars in the background display the data abundance, whereas the solid lines show the average leave-one-out error per image range of the single features. Our instance-based regression approach requires a sufficient number of training examples for the body height to be predicted. Thus, the elevated average leave-one-out errors at the lower body region below the coccyx and the region near the top of the head are hardly surprising. For all descriptors, we observe errors below 15 cm for body areas with a broad coverage (with an average of at least one example per volume). The observed qualitative variance is reduced by using the combined features (dotted line). Nevertheless, the figure also shows that the combined approach requires a sufficient number of training examples as well.

For more experiments on feature combination, see the results section in [55], our corresponding regression publication. The following experiments all use the single-represented PHOG image descriptors in order to limit the number of experiments to be presented. All techniques analyzed from now on, however, are also applicable on the multi-represented regression approach.

Gathering More Training Data with Automatic Labels Before evaluating our dimension reduction method, we need to analyze the regression

Table 5.3: Leave-one-out validation (LOO) errors [in cm] of k-NN slice mapping $h_i^{\mathrm{REG}}(\mathbf{s}_{i,j})$ for various database sizes of n CT scans with m slices. The last column lists the best-performing k parameters.

Ground Truth	n	m	Error [cm]	Time / Query [ms]	Best k
manual	29	12 529	4.18	13	1
automatic	29	12 529	3.95	13	3
automatic	100	46 063	2.96	44	8
automatic	150	72 190	2.33	65	10
automatic	300	143 284	2.15	142	3
automatic	500	235 159	1.82	220	3
automatic	1000	440 226	1.58	339	1

performance for automatically-labelled datasets. We will first show that the atlas-based labelling approach yields comparable results as manual labels. This allows us to increase the size of the training set in order to further improve the quality of our height predictions.

We thus restrict our training dataset of 60 volumes to the 29 volumes with 12 529 slices for which at least two of the 22 landmarks could be detected with the detector of [136]. The average leave-one-out prediction error of this data subset is listed in the first row of Table 5.3.

The next row of Table 5.3 displays the error of the same dataset, which has been labelled automatically. Our experiments show that the average registration error of 4.18 cm of the manual labelling is even lowered to 3.92 cm when using the automatic labelling. Thus, we can safely test our regression method on larger datasets, which have been automatically annotated. This allows to fully exploit the strength of our instance-based regression approach. For smaller databases, alternative regression approaches should be considered, however, with the wealth of information available, our lazy learner is very hard to beat.

We observe a steady improvement of the empirical errors down to 1.58 cm for increasing database sizes, however, this comes at the price of longer runtimes. For a dataset of 500 volumes consisting of 235 159 slices, a single query performed as sequential scan in main memory requires 220 ms. The additional cost of keeping the complete training database in main memory poses a further drawback. The following section will therefore evaluate our method of runtime optimization by using an efficient indexing scheme.

As an interesting side observation, note the variation of the best-performing k-nearest neighbor parameter for growing datasets. For each experiment, we screened various k parameters between 1 and 20 and we display the error of the

Table 5.4: LOO regression errors [in cm] for RCA-transformed data with query times [in ms] and the best-scoring k-nearest neighbor parameters in X-Trees representing 2 103 scans with 901 278 slices.

Dimension	Error [cm]	Time / Query [ms]	Best k
5	2.68	6	15
10	1.98	20	7
25	1.33	209	1
50	1.15	2 598	1

best-performing k-paramtrization. For very small datasets, smaller k are more successful than large k, as there is little choice of plausible nearest-neighbor candidates. In medium-sized datasets, the best k parameters are rather high, i.e. the consensus of many nearest-neighbors is more stable than just a few nearest neighbors. For larger databases of more than 300 volumes, this effect is reversed, since there is now a sufficient choice of very similar nearest neighbors, which will only be distorted by forming a consensus regression vote. This effect can also be observed for databases formed from dimensionally-reduced feature vectors, however, due to an unavoidable loss of information caused by the compression, the peak of large best k parameters is delayed to even larger databases than 1000 volumes.

5.7.2.3 Speed-up via RCA and Indexing

In order to speed up regression, we index the training data in an X-Tree [15] after reducing the dimensionality via RCA. We tested the target dimensions $5, 10, 25$ and 50. Using an index, we could now employ the complete dataset of 2 476 scans. We used a subset of 373 scans (= 163 376 slices) as training set for the RCA and tested the performance on the remaining 2 103 scans (901 278 instances). Table 5.4 shows the average leave-one-out (LOO) errors and query runtimes (excluding the time for feature generation) for the indexes generated from the test set.

As can be been seen in Table 5.4, the curse of dimensionality causes the X-Tree to lose much of its effectiveness for increasing dimensions. Additionally, the error does only moderately increase for smaller descriptor dimensionalities. Based on these observations, we consider the 10 dimensional data set as the best trade-off, having a prediction error of 1.98 cm and a query time of 20 ms. We use this dataset for all following experiments.

Note that the runtimes displayed in Table 5.4 are actual disk-based retrieval measurements. Since, however, we tested 100 000 predictions in a row per ex-

periment, the underlying caching mechanism of the used hard disk and of the operating system caused a decrease in I/O costs. A single prediction query for 10-dimensional image features will thus most likely require more time than 20 ms. For a more comprehensive analysis of this effect and additional observations on the use of hierarchical index structures in the advancing medium of solid state disks (SSDs), please refer to our publication in [52].

The total runtime required for feature generation is combined from the time of the actual feature generation for a down-scaled version of the query slice (20 ms) and the time required for RCA transformation (0.1 ms). Thus, our selected query configuration results in a total prediction time of 40 ms.

Next, in order to validate the performance of the proposed ROI query workflow we will first analyze the accuracy of the retrieved ROIs and then proceed with an examination of retrieval times.

5.7.3 Precision of ROI Queries

For validating the precision of a complete ROI Query, we could again use automatically detected landmarks for defining a ground truth of lower and upper bounds. However, we cannot guarantee for the correctness of these matching points.

Therefore, we generated a new set of annotation points with five new landmark types: "lower bound of coccyx", "sacral promontory", "lower plate of the twelfth thoracic vertebra", "lower xiphoid process" and "cranial sternum". An annotation example is shown in Figure 5.13. These landmarks were hand-annotated by a medical expert for providing a set of markers which have been verified visually. The ground-truth positions $h_j(v_i)$ of an annotated landmark j in a volume v_i are defined by the manual labelling scheme linearly interpolating between two user-defined fix points.

In Table 5.5, we show the results of predicting all visible intervals with ROI queries formed by pairs of these landmarks in a dataset of 33 manually annotated volumes. As not all landmarks were visible in all volumes, only 158 of the 330 theoretically-possible intervals could be tested. Since the annotation error – the deviation of these markers from their expected positions – is at 2.58 cm, we cannot expect the queries to produce more reliable predictions.

Still, the ranges predicted with the landmark-based interpolation approach (again using the 22 landmarks of Seifert *et al.* [136] with a model trained from the EXMP atlas via ABS interpolation) reduce this error to less than 2 cm. Our manually-annotated landmarks are thus well-captured by the non-linear interpolation approach.

Using Algorithm 8 with varying grid sizes g for the initial matching points P_i also provides good predictions. We observe, however, that using a larger

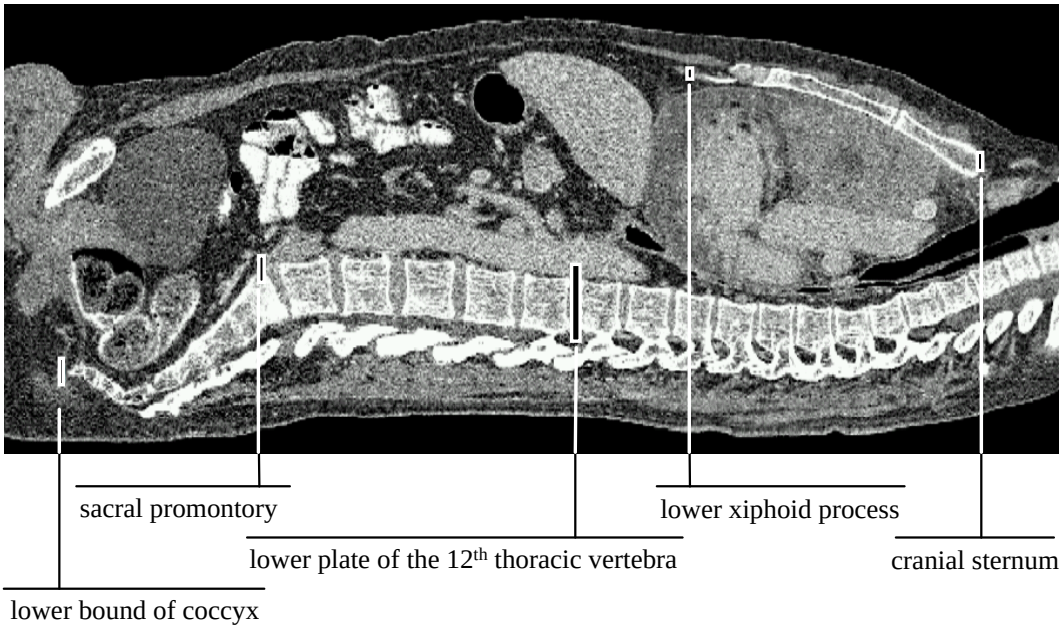


Figure 5.13: Annotation example for manually-defined thorax landmarks on a sagittal body cross section.

number of seed points only mildly improves the accuracy of the predictions, but it greatly increases the number of matching points being generated by regression (q). We conclude that two seed points are sufficient for our simple optimization scheme. Any more sophisticated optimization procedures should rather involve an intelligent screening of the proposed result range $(\hat{\mathbf{s}}_{\mathrm{lb}}, \ldots, \hat{\mathbf{s}}_{\mathrm{ub}})$ than use more seed points.

In Figure 5.14 we see the cumulative distribution function $F(\text{error} \leq x\,\text{cm})$ for the analyzed query intervals. The 'Annotation' bars show the performance of the annotated ground truth landmarks, and the 'Algorithm 8' bars represent our ROI query algorithm using two seed points. Again, the interpolation-based retrieval approach performs best. Additionally, there is almost no difference between the quality of the ground truth and our algorithm. The probability that the total prediction error $(\mathrm{err}(\hat{\mathbf{s}}_{\mathrm{lb}}) + \mathrm{err}(\hat{\mathbf{s}}_{\mathrm{ub}}))$ is at most 2 cm is almost 50 %. Again, with a height spacing of 5 mm, this means that in half of the cases, the retrieved range deviates by only two slices for each the lower and upper bound. When thus extending the returned query range by our predefined safety range, most returned subvolumes will completely contain the requested ROI.

Concerning the weighting procedure introduced in Section 5.5.4 for prioritizing the matching points generated by Algorithm 8, let us remark that our

Table 5.5: Average deviation [in cm] of the result ROI of Algorithm 8 from the manually marked ROIs ($\text{err}(\hat{\mathbf{s}}_{\text{lb}}) + \text{err}(\hat{\mathbf{s}}_{\text{ub}})$) with the average number of regression queries q and the runtime per query.

Annotation:
Error: 2.58 cm

Interpolation:
Error: 1.96 cm
Runtime: 3 ms
(without landmark detection)

ROI prediction with Algorithm 8			
g	Error [cm]	q	Time / Query [ms]
2	2.66	6.8	1 273
5	2.55	9.2	1 951
10	2.43	15.2	3 032
25	2.57	30.0	5 946
50	2.39	55.5	10 081

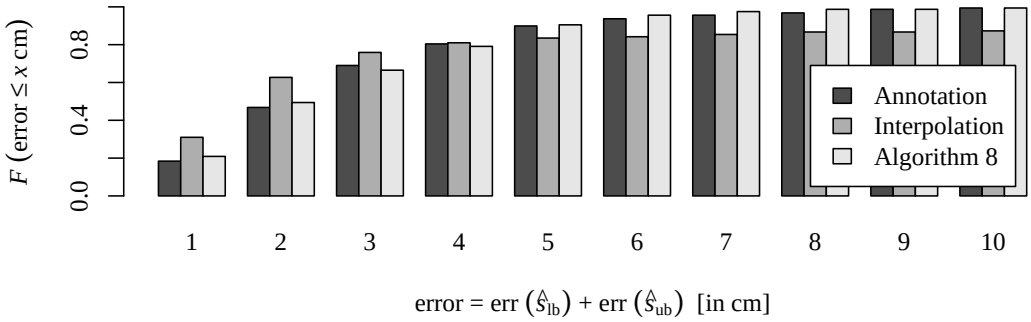


Figure 5.14: Cumulative distribution function: $F(\text{error} \leq x\,\text{cm})$ (the steeper the better). It compares the error distributions of landmark-based ROI queries with Algorithm 8 and it displays the quality of the used annotation points.

experiments show almost no difference in precision or runtime when using default constant weights for the matching points or using the weights coupled with the reliability of the k-nearest neighbors. The weighting procedure of Section 5.5.4 thus does not actually hurt the query process, but we rarely observe any significant improvement, so it can also be omitted from the query pipeline.

We thus conclude that ROI queries can be efficiently answered by using Algorithm 8 with two initial matching points. The query time for grid size $g = 2$ is 1.3 seconds. Thus, our final experiments will show that the benefit of reducing volume queries to a region of interest strongly outweighs this cost.

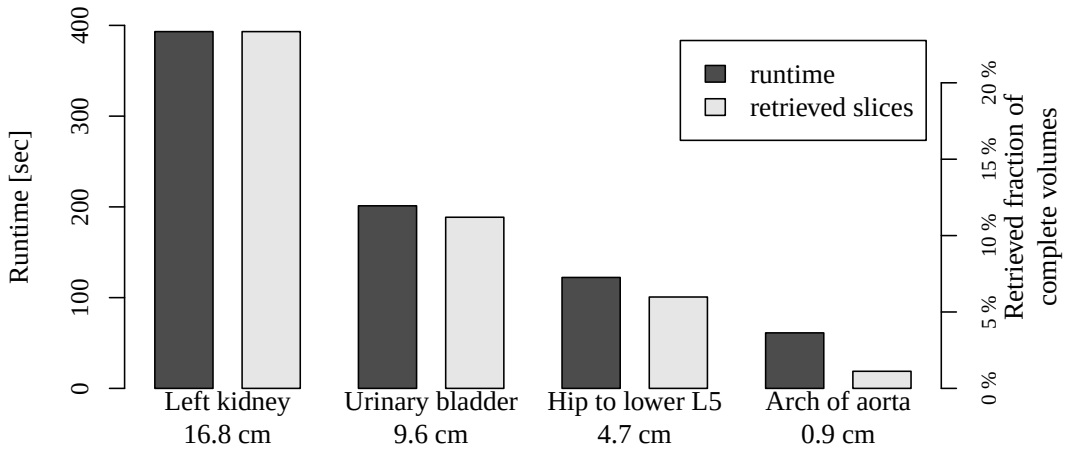


Figure 5.15: Average runtimes (ten repetitions) and volume size reduction using ROI queries with Algorithm 8. Each experiment tests 20 volumes with a total of 12 240 slices. Loading the *complete* dataset takes 1 400 seconds.

5.7.4 Runtime of ROI Queries

While mapping into the height atlas with available matching points takes only 3 ms, the detection of the 22 landmarks of [136] usually requires between 3 to 10 seconds. In our group's 100 Mb Ethernet, it takes 110 ms to load one slice of 512×512 pixels. Thus, this time investment pays off if at least 30 slices can be excluded from the retrieval query. Since the expected number of slices in our repository is 424, it is very likely that in most retrieval queries at least 30 slices can be omitted. Moreover, one artificial matching point via regression can be generated in only 40 ms. Hence, even if Algorithm 8 needs to generate 20 matching points, our retrieval approach is already faster than loading the complete volume, if we can avoid the loading of 8 slices.

Our last experiment therefore simulates some real-world retrieval scenarios. We chose a random set of 20 volumes from the database and tested them against four ROI queries defined in an example scan. Two queries are aimed at organs ("Left kidney" and "Urinary bladder"), one query ranges from the top of the hip bone to the bottom back point of vertebra L5 and the final query only requests the view of the arch of aorta. The four hereby defined query ranges have heights of 16.8, 9.6, 4.7 and 0.9 centimeters.

In Figure 5.15, we display the retrieval times of the resulting ROIs and their fraction of the complete dataset of 12 240 slices. Loading the complete 20 volumes from the server takes 1 400 seconds, whereas transferring only the ROIs induced by the given concepts takes 60 to 400 seconds, including the computation overhead for finding the ROI.

To conclude, employing our system for answering ROI queries saved between $77 - 99\,\%$ of the loading time compared to the retrieval of the complete scan. If a set of matching points is already available, e.g. in the form of detected landmarks, the decrease in retrieval time is even stronger. Thus, in the clinical routine, our system is capable to save valuable time as well as hardware resources.

5.8 Summary

This chapter discussed a retrieval pipeline for processing region of interest (ROI) queries on a repository of CT scans. It allows example-based queries specified by giving an example ROI in another CT scan as well as concept-based queries which define the query range as an anatomical concept. Since CT scans are usually stored as stacks of 2D images representing a horizontal slice in the scan, the answer of an ROI query on a CT scan is a subset of the slices in the target scan representing an ROI which is equivalent to the query ROI. The goal of the ROI retrieval system is to shorten waiting times for the user and to reduce I/O costs within the clinical network.

After query specification, the system defines a query range within a standardized height model of the human body which is designed to be independent of the patient's size or body proportions. Then, the query region in the height model is mapped to a subregion of the queried scan containing the ROI. This indirect mapping approach allows the use of the same retrieval mechanisms for example-based and concept-based queries and it is faster and less I/O-demanding than a complete pixel-based volume alignment to an example scan or a template scan.

Technically, our system is based on an interpolation function using so-called matching points linking a CT scan to the height model. These matching points can be generated by standard landmark detection approaches. By detecting a set of anatomically meaningful landmarks in a large database of CT scans, we learn a standardized height atlas mapping of these landmarks, which also facilitates the query specification by anatomical concept. Still, landmark detection frequently fails due to inappropriate image material (e.g. blurred or incomplete scans). As an alternative, we can guarantee the availability of matching points even for unannotated CT scans, by using content-based image descriptors and regression for generating matching points for arbitrary slices in a scan. Finally, we propose a query algorithm for finding a stable mapping while deriving a minimal amount of matching points.

The experimental section validated the accuracy of all components of our approach on a large database of 4 479 CT scans acquired within the THESEUS

MEDICO project, of which 2 476 scans could be automatically labelled using our atlas-based labelling system. Depending on the method used and the amount of available information, we observe average deviations between 1 and 2 centimeters. We also presented experiments for the reduced transfer volume of ROI queries being processed by our system. We conclude that loading runtimes are greatly improved with acceptable error rates for clinical retrieval.

In future work, our system will be extended to handling ROIs in all three dimensions and we will test further landmark types. Additionally, we would like to test alternative learning approaches for the used image features and the feature transformation by RCA. We also aim to apply our retrieval solution to other types of 3D objects being stored in raster databases and to examine further, more general regression or interpolation problems.

Chapter 6

Medical Content-Based Image Retrieval (CBIR)

The current diagnostic process at hospitals is mainly based on reviewing and comparing images coming from multiple time points and modalities in order to monitor disease progression over a period of time. However, for ambiguous cases the radiologist deeply relies on reference literature or second opinion. Although there is a vast amount of acquired images stored in PACS (picture archiving and communication systems) which could be reused for decision support, these datasets suffer from weak search capabilities. Thus, there is a need for a search methodology which enables the physician to fulfill intelligent search scenarios on medical image databases combining ontology-based semantic and appearance-based similarity search.

This chapter introduces the search capabilities developed within the THESEUS MEDICO project. The MEDICO prototype includes the basic retrieval mechanisms of any common PACS as well as more flexible semantic query specifications for an advanced keyword search of the PACS and additional meta-data like manual or automatically-generated annotations or literature cross-references. Furthermore, it supports a content-based image similarity search scenario by allowing the user to specify a query region of interest (ROI) within a 3D volume which is then submitted to a ranking query within a database of annotated conspicuous image regions. The different query types can be combined by a method we introduced in Seifert *et al.* [138], thus allowing for an even more flexible query specification. Details on the architecture used for the query combination step have been published in Stegmaier *et al.* [146].

6.1 Introduction

The original objective of the MEDICO research project is to improve the search mechanisms currently available in clinical database at the use case of lymphoma patients. Lymphoma is a cancer that originates in the lymphatic cells of the immune system. It is usually noticed in the form of a solid tumor of lymphoid cells and sometimes affects abdominal organs[1]. Due to their comparatively small size, these tumors can only be made visible in computed tomography (CT) scans of a rather fine resolution. As the complete body needs to be monitored in order to rule out any further organic manifestations, the total data load required for a lymphoma patient grows quickly. Our database consists of 4 479 CT scans with a total size of more than 700 GB. The development of the ROI retrieval system described in the previous chapter is thus a logical consequence of the data available.

A subset of 100 CT scans has been semantically annotated by medical experts in a *semantic reporting process* [137]. An image is annotated by first using an *image parsing system* [136] for automatically generating spatial annotations of anatomical landmarks and organs and by then manually verifying the automatic annotations. Next, further manual annotations are added, describing any image regions which are meaningful to the lymph node context or which are medically conspicuous. These annotations are organized in a comprehensive medical ontology [113] and they can be queried with a semantically flexible keyword search.

Furthermore, a medical expert annotated 1293 lesions (973 liver, 130 spleen, 190 kidneys) in 577 CT scans as 3D bounding boxes. These lesions form a database which can be queried by specifying an example template ROI in a CT scan currently opened in the MEDICO system. The resulting list contains the visually most similar lesions contained in the database.

By combining the two search methodologies of semantic queries and visual similarity search, MEDICO provides a content-based image retrieval system that exploits query constraints on anatomical annotations in order to increase the quality of an image search. Recent work [115, 3] tends toward the same direction but often loses track of the global picture. The MEDICO system provides its users with a holistic view on the patient, supporting them with a tool to search for similar-appearing lesions restricted to an individual organ, but additionally including extra-organ disease processes at the lymph nodes. By using a generic framework modeling the logical connections between the various types of queries and databases, our proposed query scheme enables the early elimination of irrelevant search results. Thus, we save time normally

[1] In this work, the spleen is subordinated the abdominal organs, even if physicians consider it to be a lymphatic structure and not an organ.

required for manual result filtering and we additionally gain a speed-up in query processing.

This concludes in a query functionality for *similar* patients, i.e. patients showing similar anatomical and pathological characteristics. Investigating the anamnesis and the successful treatment can then provide good advice for the current patient. The ability to compare images with those obtained in other patients has the potential to provide real-time decision support to practicing radiologists by showing them similar images with associated diagnoses and, where available, responses to various therapies and outcomes.

6.2 Combining Semantic and Similarity Search

We currently provide two complementing search mechanisms: *query by concept* enables the user to query the image database by the use of regular expressions where the terms are coming from the MEDICO-ontology. [113] The second search mechanisms is called *query by scribble*: the query interface provides the user with a drawing tool to define arbitrary regions. In our case, we use it to enclose a reference lesion. Combining these two mechanisms, the query language is tremendously extended w.r.t. classical content-based image retrieval systems (CBIR). Subsequently, we explain the mechanism with the following sample query:

> "Find all patients with similar lesions in the liver and with thoracic lymph nodes enlarged."

The image database can be searched for images containing similar regions based on the visual appearance. This is a close approach to classical content-based image retrieval systems (CBIR). The main advantage of our system is that the CBIR results can be restricted by the *query by concept* (here: *enlarged thoracic lymph nodes*). This mechanism furthermore allows to fully automatically limit the results to lesions within the organ which is currently of interest (here: the liver).

6.2.1 Query by Concept

The semantic annotations are stored in the *Annotation Ontology* (see Figure 6.1), which is part of the MEDICO ontology stack. The arrows labeled with '`mano:`' are used to depict property relations, rectangles represent classes and '`isa`' arrows are inheritance dependencies. The annotation ontology scheme sets the patient to the center. Every patient owns some studies defined by a unique identifier and a specific time period. The MEDICO study is more

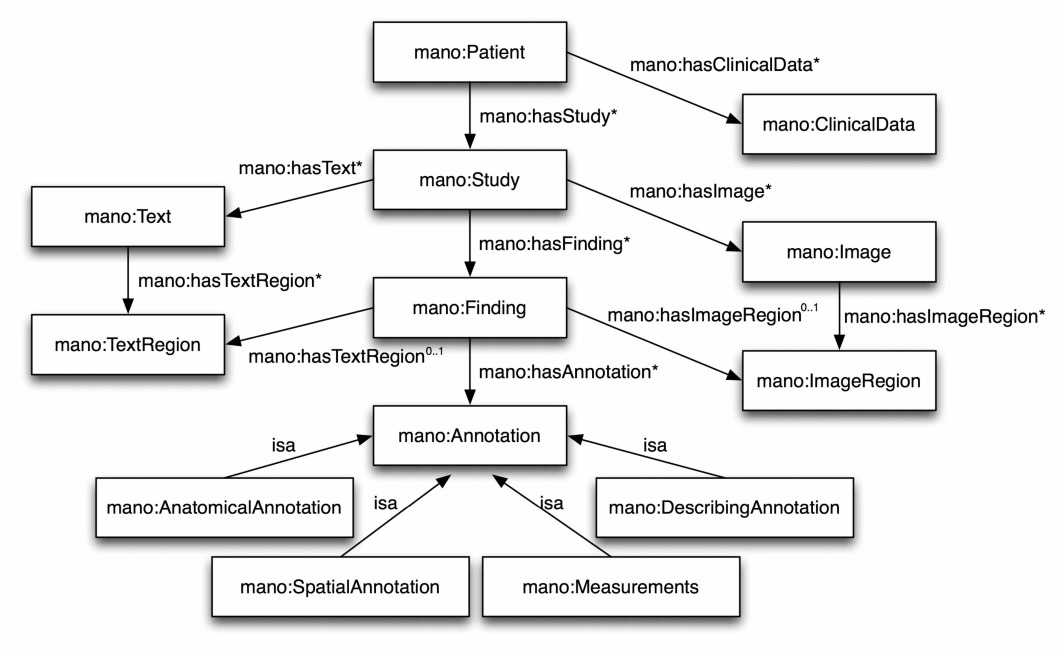


Figure 6.1: MEDICO annotation (`mano`) ontology scheme supporting temporal, multi-modal and text-to-image relations.

than just a DICOM [117] study: it is a container for all annotations from images, texts, clinical data within a given time period. This is the cornerstone to enable temporal queries as well as queries considering multiple modalities.

The annotation ontology scheme is illustrated in Figure 6.1 and the design was driven by the following requirements:

- *Link report text passages with related image regions:* Annotations from images and texts must be stored in the same model which should consider the fact that reports summarize annotations from multiple images.
- *Disease progression:* Changes to anatomy due to a pathology over time should be represented. A combined examination of studies with their pre-studies needs temporal relations.
- *Multi modality:* Diagnosis often needs a synoptic view of images acquired with different modalities, e.g., CT, MRI, US. Therefore, the underlying annotation ontology should link annotations not only across time, but also across different modalities.
- In order to adopt hospitals preferred wording, the stack of used ontologies should be extensible, e.g., some of the hospitals have already made

experience with SNOMED CT [86] or AIM [27], or they use the World Health Organization's system for international classification of diseases (ICD [171]). Therefore, the annotation scheme should not only incorporate RadLex [103] and FMA [127] but also support further ontologies. For the ontology alignment required for mapping between these various ontologies, the KEMM-methodology [168] was developed within the MEDICO project.

An image region is an arbitrarily shaped spatial sub image which is defined as landmark point, triangulated mesh or image mask. The triangulated meshes are currently used to describe organs detected by the image parsing system, [136] and image masks are used to define scribbles.

The class `mano:Finding` relates anatomical annotations, such as *liver*, *spleen* with annotations describing the anatomy, such as *enlarged*, *hypodense*, *jagged margin*. Currently, FMA and the anatomical tree of RadLex are used to define the anatomy, whereas the imaging observation and visual modifier trees of RadLex are used for finding descriptions. If an anatomical term of a finding is missing in the existing vocabulary, spatial annotations allow the user to paraphrase it with spatial relations such as *nearTo* or *inBetween*, e.g., the lymph node *near to* renal hilum.

If the finding is a specific area or volume, we can add a `mano:Measurement` to store the values of the parameter. Any additional information can be archived by `mano:FreeText`. To free the user from selecting the right anatomy term, we added a query expansion mechanism which recursively infers subclasses in the FMA. Thus, the sample query *Thoracic lymph node* results in 90 sub-classes:

Thoracic lymph node → Mediastinal lymph node
→ Aortopulmonary lymph node
→ Pericardial lymph node
→ ...
→ Esophageal lymph node
→ ...

For further restricting query results to patients with lymphoma of a given Ann-Arbor stage, we can furthermore make use of an integrated staging system [185] implemented in OWL DL as a plug-in to the MEDICO-ontology.

6.2.2 Query by Scribble

The goal of MEDICO's visual similarity search is to allow the user to quickly outline a region of interest (ROI) and to ask the system for similar ROIs,

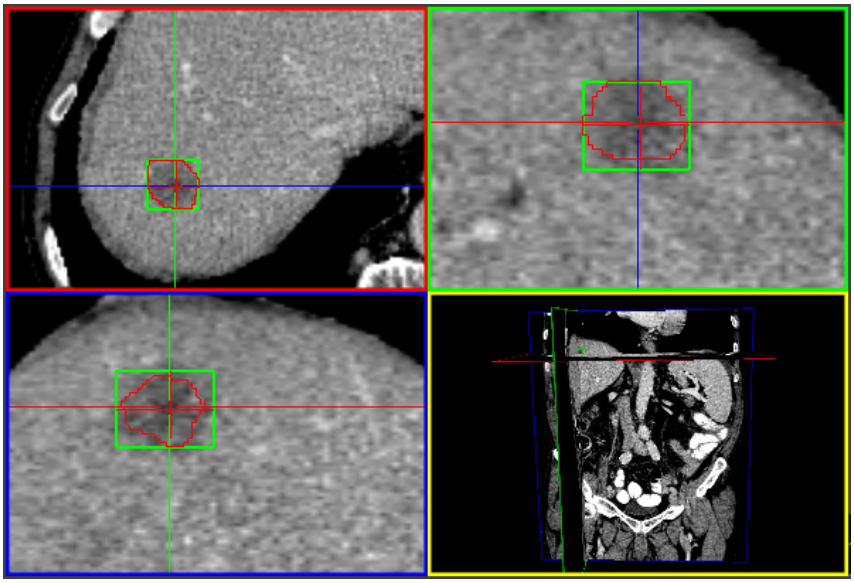


Figure 6.2: Example for a query by scribble selected in three 2D projections (top left: transverse plane, top right: sagittal plane, bottom left: coronal plane) with a 3D projection of the displayed planes (bottom right). The ROI actually used as query is outlined by the green box.

without having to take the time for an exact segmentation. We call such a quick ROI specification a *scribble*. An example scribble is displayed in Figure 6.2. In order to relieve the user from having to specify an exact 3D segmentation of the query region, the ROI defined by a scribble is its minimum bounding box.

6.2.2.1 Used Image Descriptors

We support 3D scribbles, however, in favor of a faster query specification, we expect most queries to be posed as 2D selections. Since 2D image features will have the highest descriptive power due to their maximized image resolution, our first group of image descriptors is based on 2D image features.

Multi-instance Representations One way to generate 3D image descriptors is to represent a 3D bounding box by a collection of 2D image features generated from the 2D ROIs implied by the 3D ROI in the transversal plane. We treat such a collection like a classical multi-instance problem, where one object is represented by an unknown number of instances of a fixed representation. On the one hand, this causes a loss of information, since we discard the slices' order. On the other hand, the multi-instance perspective allows the

comparison of various slice permutations, which can very well contribute to lesion similarity.

Among the tested 2D image descriptors, the best-performing were simple grey-value histograms and the Haralick [79] texture features already introduced in Section 5.5.1.3. The grey-value histograms are formed from 150 bins (HIST) over the given Hounsfield space, and the Haralick descriptor is generated for the 9 subwindows of a 3 by 3 grid imposed on each 2D ROI (HAR). Since one Haralick descriptor for 5 different pixel distance values $(1, 3, 5, 7, 11)$ contains $13 \cdot 5 = 65$ statistics, this amounts to descriptors of sizes 150 and $65 \cdot 9 = 585$ for each slice covered by the lesion's bounding box.

Global Image Descriptors Naturally, we also wanted to use real global image descriptors. A simple and plausible global image feature of a 3D annotation is its real-world size. We thus define a lesion representation (SIZE) consisting of the extension of the lesion's bounding box in all three dimensions, i.e. a vector (w_x, w_y, w_z).

The main problem of deriving a 3D image descriptor on a minimum bounding box annotation is that it is prone to be disturbed by the large background quota of pixels which do not actually belong to the lesion. This obstacle can be mildened in the case of multi-instance representations by using a tolerant distance measure, but it can have very strong effects in a global context. Take for example a grey value histogram over the complete bounding box: the descriptor of any lesion located at the border of an organ is prone to be heavily disturbed by surrounding air or neighboring bone structures.

Wa have already introduced one way to deal with these local effects in Section 5.5.1.1 with the spatial pyramid kernel. We could directly extend this approach to 3D, however, usually the resolution in the z-axis will be by far lower than in along the x- or y-axis, especially if only a 2D query is posed. Additionally, the spatial pyramid kernel is not rotation invariant. This was not a problem for the 2D query use case of slice localization, but in 3D, any minor rotation of a lesion will cause it to be classified as dissimilar from a descriptor derived from an un-rotated version of the same lesion.

We thus decided to use a rotation invariant shell kernel, fitting 3D ellipsoids of decreasing radius within the 3D bounding box of a given ROI. The regions between two succeeding ellipsoids form a shell from which we compute a global image extractor. For now, we only tested a grey value histogram descriptor, however, other decriptors modeling the shell's texture should also be considered as an option.

In addition to the shells, we observed a beneficial effect of including the background pixels not covered by any of the ellipsoid kernel's shells as an additional container of image information. This effect can be explained by the

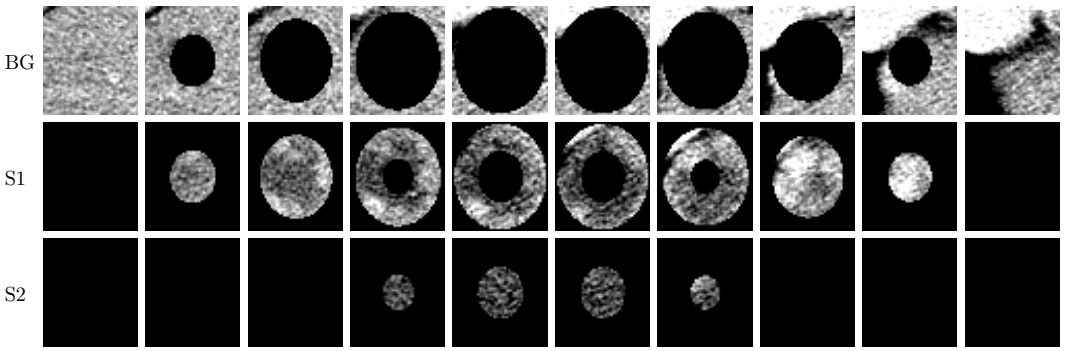


Figure 6.3: Visualization of the ellipsoid shell kernel using two shells (S1 and S2) and a background selection region (BG). Any non-selected pixels are displayed as black.

nature of our validation procedure, where we compare feature-based similarity rankings with manually-assigned pair-wise similarity labels. The annotator is more likely going to qualify a pair of lesions to be similar if their backgrounds are of a similar appearance. The background pixels can thus be qualified as a rather valuable piece of information

Our descriptor (EHIST) is thus defined as the concatenation of s h-dimensional grey value histograms formed from the shells defined by s concentric ellipsoids plus an h-dimensional grey value histogram formed from the remaining background pixels.

In our experiments, the use of two shells and 60-dimensional grey value histograms turned out to perform best. Figure 6.3 visualizes the pixel selection kernels of a 3D ROI. The upper row displays the pixels selected as background information (BG) in the 10 consecutive slices of a hypodense lesion. In the middle row, we see all pixels of the lesion's outer boundary approximation (shell S1) and the bottom row visualizes the selected pixels of the innermost shell (S2).

6.2.2.2 Image Feature Combination

In order to achieve the optimal combination of various types of image features, we would have preferred to use an analogous approach to the multi-represented regression scheme introduced in Section 5.5.2.2. In fact, when assuming an existing labelling of the retrieval dataset consisting of pair-wise similarity labels or a classification of the dataset into various classes, we could again use this recipe: for R representations query the R separate databases and combine the R result sets according to a criterion evaluating their label coherence. However, first, we cannot assume to have a labelled retrieval dataset in a

medical environment. Second, the result sets of various image representations will usually be labelled in versatile ways and will thus be difficult to combine.

Therefore, we decided to use a classical weighted sum of distances over the chosen representations' distances. We require two kinds of weights: the first weight, s_r, represents the standard deviation of distances of representation r for ensuring a comparable distance scaling. Note that this simple distance joining procedure assumes that the single representations' distances follow comparable distributions. This was the case for the distance measures tested in our framework, however, other image representations might require a more sophisticated scaling procedure. Additionally, the weights w_r represent an actual weighting factor to be manually assigned to representation r. The combined distance measure d_{combined} for a set of lesion representations R is:

$$d_{\text{combined}} = \frac{1}{\sum_{r=1}^{R} w_r} \sum_{r=1}^{R} \frac{w_r}{s_r} d_r \ . \tag{6.1}$$

For each representation, we tested various distance measures on their suitability for the image retrieval task. For global descriptors, i.e. d-dimensional, real-valued feature vectors like the ones obtained with the SIZE or the EHIST feature, we used two types of L_p norms (c.f. (3.1)): the Manhattan distance (L_1) and the Euclidean distance (L_2). Note that for retrieval tasks, it is sufficient to compute the squared Euclidean distance, consequently economizing the time usually required for computing the square root.

The multi-instance representations HIST and HAR were evaluated with various of the multi-instance distance measures surveyed in Chapter 4. The best-performing distance measure were the sum of minimum distances (SMD, c.f. Section 4.2.4) and its variant using only the k-lowest pair-wise minimum distances, the k-SMD introduced in Section 4.3.5.1. By also using the L_1 or L_2 norm as instance-distance in the selected multi-instance distances, we actually receive comparable distance distributions for the multi-instance representations as for the global image representations and we can thus apply the feature combination distance d_{combined}.

6.2.3 Combined Search

Even though our combined distance measure is well-suited for lesion comparison, similarity among medical images remains a difficult application. The appearance of a CT image depends on the setting of the image kernel, the time and kind of the applied contrast agent and other factors like previous organ excisions or medical implants. In many cases, this information is not even available to the computer. Therefore, image similarity alone can hardly be a significant indication of a similar patient case.

The MEDICO system thus exploits all available, manually specified and automatically generated meta-information of the query volume for restricting the search space to annotations which are actually relevant. MEDICO can automatically determine the position of the ROI w.r.t. a number of organs and landmarks [137] or within a standardized body atlas [55, 25], as well as any available information on the patient's prior history in the accessible database collection.

Figure 6.4 shows an example scribble and the position of a query by scribble in the combined search workflow.

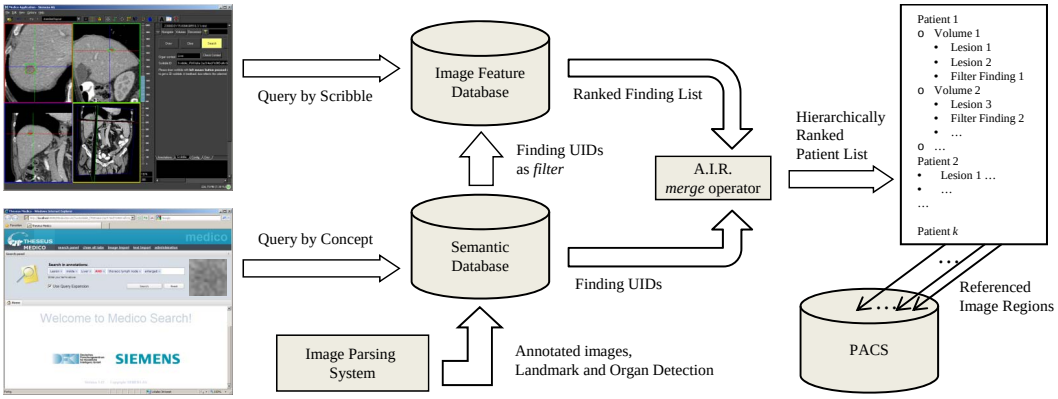


Figure 6.4: Query by Scribble: retrieve similar regions of interest (ROIs) via a quick selection mark on an ROI. Query by Concept: retrieve an image list using semantic filtering criteria, some of which can be automatically generated. Combined Search: use (some of) the output of a Query by Concept as filter list for a Query by Scribble and combine the two result lists.

The main advantage of a *query by concept* over a *query by scribble* is the standardized way of query specification. By using the standardized vocabulary of the MEDICO ontology, queries by concept can be easily combined (with `and`, `or` or `not`) and extended to hypernyms. Thus, they allow a time-efficient query processing.

However, even using a convenient search-as-you-type functionality, the use of the expert vocabulary requires some training. As some terms do not directly induce a unique medical concept, they need to be verified by manually inspecting the underlying type hierarchy. A more serious drawback is the fact that the results of a query by concept are not ordered by relevance. In current keyword-based search engines on the internet, the first hits are expected to be the most relevant. This relevance is usually determined by combining knowledge about user-specific interest profiles with the general demand on the result pages. Such a ranking information is typically not available to a medical search

engine, since the retrieval frequency of a patient should not impose any notion of importance. If, however, a query returns thousands of randomly-ordered results, the user will quickly lose interest in the search engine. A similarity search as used in a query by scribble, however, does provide a ranking of the result set. Additionally, it allows to specify a query without the need of giving a detailed verbal description of the observed ROI.

In order to combine the best properties of the available search modalities, we thus need to decide for the optimal order of query processing. The two basic combination operators for two result sets are the *merge* or *join* operator and the *filter* or *intersect* operator. Since a query by concept is expected to be faster than a more computationally involved query by scribble, the most efficient order is to first pose the query by concept and to then use the resulting output as a *filter* for a query by scribble. This way, the expensive image search only needs to rank relevant results.

Finally, we also use the *merge* operator for including any non-lesion results of the query by concept into the final result set. This way, the unstructured output of the semantic search receives a sensible order and the bare image similarity ranking is enriched by an explanation of why the ranked lesions are presented to the user.

An example of a combined query is presented in the experimental section. Let us note that we also exploit further information available to the MEDICO system: the MEDICO *image parsing system* [136] can initialize a combined query by automatically detecting the organ context of the scribble. The anatomical parsing system requires less than 2 minutes for detecting the most important organs and landmarks – if this step has already been completed, when initializing a query, the user does not have to manually provide the semantic expression for the anatomical location of the scribble as filter criterion for the query by concept.

6.3 Search Infrastructure

MEDICO provides the user with an easy-to-use web-based form to describe a search query. Currently, a search consists of a semantically rich data set composed of DICOM tags, image annotations, text annotations and gray-value based 3D CT images as reference. This leads to a heterogeneous multimedia retrieval environment with multiple query languages for retrieval: DICOM information is stored directly in the PACS, image and text annotations are saved in a triple store and the CT scans are accessible by an additional image search engine performing instance-based similarity search in a relational database.

Apparently, all these retrieval services are using their own query languages

for retrieval (e.g., SPARQL or SQL) as well as different ways of data representation (e.g., OWL, numeric feature vectors). Beside all differences, they form a common, semantically linked, global data set. To fulfill a meaningful semantic search, these interoperability issues had to be solved. Furthermore, it is essential to formulate queries that take the aforementioned diverse retrieval paradigms into account. For this purpose, MEDICO integrates the AIR [145] multimedia middleware framework which implements the MPEG Query Format (MPQF), [48] which is currently the most specific query language for multimedia retrieval. This framework has been especially designed to serve as a mediator between a search interface and an arbitrary amount of back-ends. AIR is able to support both, distributed query processing as well as local query processing.

The current dataset of the MEDICO project consists of 5 493 anonymized volumes (of which 4 479 are CT scans) which were taken of 333 lymphoma patients for monitoring the progress of the disease. The volumes were collected by our clinical partner at Friedrich Alexander University in Erlangen and they sum up to a raw pixel dataset of more than 700 gigabytes. 100 volumes have been semantically annotated with terms from the MEDICO-ontology [113], combining expert knowledge represented in medical ontologies such as the *Foundational Model of Anatomy* (FMA) [128] and RadLex [103]. Another medical expert additionally annotated lesions within liver, spleen and kidney in 574 volumes.

To avoid large efforts in annotating images the MEDICO project recently proposed a *semantic reporting process* [137] which makes use of an *image parsing system* [136] and a semi-automatic *semantic reporting tool*. The image parsing system automatically detects anatomical structures and generates an initial annotation list, whereas the reporting tool allows the radiologist to complement them. This semantic reporting tool provides the user with context-sensitive term suggestion, fast volume navigation by directly jumping to or zooming into an anatomical region and hyperlink report text passages with the appropriate image location.

The complete data store is split into the following three knowledge bases: a *PACS*, a *semantic database* and an *image feature database*.

6.3.1 The PACS

The local PACS containing the raw images as well as any meta-information generated by the used medical scanners is connected via DCM4CHE [41], a strict implementation of the DICOM [117] standard. A DICOM document consists of the actual image data as well as supplementary header information about the image modalities and additional information like the acquisition

time of the scan, a patient's UID or any used contrast agent. Each scan gets assigned a unique DICOM series UID. The system supports basic retrieval queries like a request for all images belonging to a given patient.

6.3.2 The Semantic Database

The *Semantic Database* stores semantic image and text annotations. It is implemented using a Jena text database (Jena TDB, `http://www.openjena.org/TDB/`), which directly supports OWL/RDF and SPARQL. We selected the Jena library because of its good scalability and runtime performance. [147] See the MEDICO ontology in Figure 6.1 for the OWL developed to store the semantic annotations. They are associated to so-called *findings*, which allow the link to a text source or a specific image region. This database enables more complex queries for those findings, fulfilling a given semantic constraint, e.g. a lesion within a pre-defined body region. The findings are associated to the volumes stored in the PACS via series UIDs and patient UIDs.

6.3.3 The Image Feature Database

Instance-based similarity queries are handled by the image feature database. All required data is stored in a relational database (mySQL: `http://dev.mysql.com/`). Its content is sketched in Table 6.1.

Table 6.1: Structure of the image feature database.

	PACS information	semantic image annotations
Data references:	DICOM meta-data: patient UIDs volume UIDs radiology reports	landmarks organs regions of interest (ROIs): lesions
	image features linked to a volume UID or an ROI UID	
Image features:	grey value histograms, histograms of gradients, texture features, tissue classification histograms, size measurements	

In order to minimize the communication overhead with the PACS, this database also contains selected fields of the DICOM meta-data. Additionally, the database contains selected types of text and image annotations also available in the semantic database. These annotations may be landmarks or organs, [137] or manually-specified image regions, i.e. regions of interest (ROIs).

The total number of annotations and the number of volumes associated to such an annotation type is summarized in Table 6.2.

Table 6.2: Statistics on the data store within the image feature database containing 4 479 CT scans.

Data type	number of objects	for the number of volumes
Landmarks	42 902	2 793
Organs	18 048	3 130
Radiology reports	2 722	2 722
Lesions	1 293	574

The main focus of this similarity search application is the fast and selective availability of automatically-generated image features. The SQL tables enable a quick retrieval of candidate lesions via the specification of a filter set of candidate findings or volumes provided by the semantic database. If no filter is specified, our system also supports spatial indexing structures for accelerated ranking queries.

For more information on the query architecture of the MEDICO framework and for details on the supported query processing mechanisms of the AIR query broker please refer to Stegmaier *et al.* [146].

6.4 Related Work

This section gives a brief overview on existing semantic and image similarity query approaches in medical database.

6.4.1 Query by Concept

A real query-by-concept as defined in this chapter is so far not available in standard medical databases. The well-established DICOM standard [117] supports keyword-based search in special fields of meta-information, however, keywords alone are not sufficient to form a flexible query exploiting the advantages of a semantic structure.

In the boom of Web 2.0 technology, semantic search has received special attention in the past years. Especially in medical environments, the hopes about simplifying the complex process of standardization by using medical ontologies have been high. [142] Consequently, concept-based queries [113, 137] and the architectures required for extracting the required information from

medical ontologies already available [168] lead to the first publications of the MEDICO project.

Other research projects like IRMA (the Image Retrieval in Medical Application)[2] try to use existing mechanisms like DICOM Structured Reporting (SR) for adding semantic annotations to images. [166] The main problem here is that these structured documents are only occasionally supported by the leading manufacturers and that it still requires an additional query system organizing the selected conceptual tags.

This is one of the reasons why the research program AIM (Annotation and Image Markup) also uses a comparable approach as the MEDICO project for defining an annotation ontology on medical images. [27, 28] In [130], they followed the strategy of examining web queries of radiologists for identifying relevant search terms and their relations to various types of medical ontologies.

Progress in the field of semantic similarity search, however, has been relatively small. In [94], Korenblum *et al.* introduce a similarity search system which ranks images according to their intersection of keyword annotations. The main drawback of such an approach is the implausible handling of missing annotations: it is highly unlikely that both the objects within the query database and the query image itself have been completely annotated with all relevant keywords, i.e. concepts. Thus, even though a match in any given concept can be taken to be a positive contribution to object similarity, a lack of a correspondence in another concept cannot necessarily be taken as an indication of dissimilarity. Consequently, the MEDICO system refrains from employing a ranking system as in [94], where a high rank of an image or a finding would be rather correlated with its degree of annotation detail than with its similarity to a query template.

6.4.2 Query by Scribble

In the past years, CBIR in medical databases was mostly restricted to special cases like 2D skin lesions [6] or cervicographic images. [175] These specialized systems are usually designed for decision support, i.e. they try to summarize their results in a way equivalent to a classification system. In [162] for instance, Wang *et al.* present an alternative classification system for lesion tissue classification using image patches via support vector machines. A recent summary over retrieval-based classification systems is listed in [2].

However, also the classical idea of image retrieval has advanced in the medical field. [114] A survey on medical image retrieval systems currently available is presented at the begining of this thesis in Section 1.1.2. Our use case is the selection of image subregions for use as a query template.

[2] http://irma-project.org

In Seifert *et al.*, [138] we presented our approach at the example of lesion retrieval, with a lesion being any visually conspicuous part within a volume scan. Napels *et al.* [115] describe similarity retrieval on liver lesions using an electronic representation derived from an exact 2D segmentation of the lesion's center slice. The image descriptors used in their work are rather focused on manually-annotated semantic ROI features than on automatically-generated image features, as they elaborated in [94]. An extension using only automatic image features still requires the exact segmentation of the lesion and it involves the computation of 2D shape descriptors. [173] Other image query systems based on structured reports as proposed in [166] are mostly focused on 2D image annotations and therefore 2D queries as well.

Our query approach uses actual 3D annotations, however due to the high annotation overhead of exact lesion segmentations, we only generate a rough lesion representation by outlining bounding boxes. Since these bounding boxes are not restricted to the target lesions, they also contain background information of the liver or any bounding organs, bones or neighboring air.

Since by applying the 3D ellipsoid image kernel, we are using a quasi-segmentation step in our feature generation approach, we also point out a small sample of existing segmentation techniques in the field of liver lesions: [14] describes a semi-automated watershed approach in 2D. In [143], Soler *et al.* present a deformable model-based 3D liver segmentation approach, which also outlines lesions in order to improve the quality of the organ segmentation. [148] proposed a semi-supervised, iterative growth algorithm based on Gaussian intensity estimates. [111] use a probabilistic boosting tree for detecting and segmenting two types of liver lesions.

6.4.3 Combined Search

As already mentioned, Napel *et al.* [115] use image features and manually-annotated semantic features for ranking liver lesions. Their modality aggregation approach is distance combination using adaptive boosting (AdaBoost [63]). This way, they achieve a good performance on a dataset of 30 exactly segmented 2D lesion annotations. It is hard to judge the effect of overfitting on this rather small test set. Moreover, this approach also depends on a complete semantic annotation of the data, not allowing for missing values which are bound to be abundant in medical environments.

Welter *et al.* [167] propose to use DICOM [117] structured reporting documents for providing a better standardization of various CBIR-based CAD (computer-aided diagnosis) systems. Given a properly-structured database containg both image-based annotations and anatomical or disease-relevant meta-information, combined queries will indeed be easier to handle than the

combination of various knowledge bases as in the MEDICO project. For various reasons (vendor competition, privacy protection and other legal restrictions), however, the availability of such an integrated system is rather unlikely. Therefore, most semantically-motivated image similarity retrieval approaches are based on the exhaustive semantic annotation of a given database as in [94].

6.5 Experimental Evaluation

We validated our search procedure with respect to the quality of the visual similarity ranking, as well as w.r.t. the gain achieved by combining the visual query with automatically-derived and manually-specified semantic queries.

6.5.1 Datasets

A medical expert annotated the 3D bounding boxes of 1293 lesions (973 liver, 130 spleen, 190 kidneys) in 577 CT scans for 92 patients. For verifying the quality of our visual similarity metric, we selected 111 liver lesions with a bounding box volume $\geq 5\,\mathrm{cm}^3$ as validation set V_1 (79 volumes of 26 patients) and a medical expert annotated them with pair-wise similarity scores on a 5-step scale from 0 (completely dissimilar) to 100 (same lesion).

In order to reduce this significant annotation effort, we defined equivalence classes for 21 lesions which occurred multiple times in the dataset in the context of a scan taken another time. We only annotated a set of 62 lesions with pair-wise similarity scores ($= \frac{62 \cdot 61}{2} = 1\,891$) and auto-extended these scores by the known equivalence classes. This way, we lose some precision in the annotation, since only a part of the final $\frac{111 \cdot 110}{2} = 6\,105$ pair-wise similarity scores has been manually validated, however, we can generate a larger data store at lower annotation costs. A screenshot of our similarity annotation tool can be examined in Figure 6.5.

Furthermore, we randomly sampled 60 of the liver lesions of V_1 and combined them with 60 additional spleen lesions and 60 kidney lesions (all $\geq 5\,\mathrm{cm}^3$) as dataset V_2. This way we can evaluate the effect of omitting the automatically-derived location knowledge obtained by the image parsing system.

Additionally, our medical experts annotated 100 CT scans as set V_3 in a *semantic reporting process* [137] for visible radiological findings, mapped into the MEDICO-ontology. [113] This set of volume annotations can be queried by advanced semantic queries like *enlarged thoracic lymph nodes*.

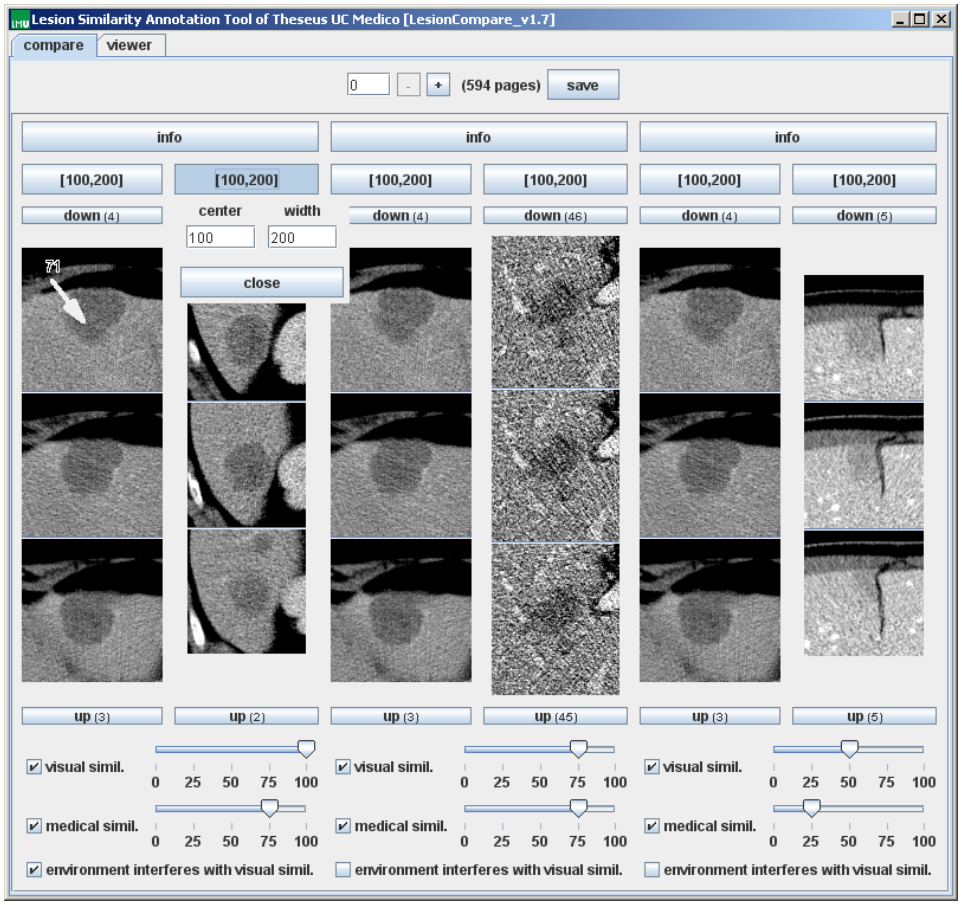


Figure 6.5: Screenshot of the annotation tool used for defining pair-wise lesion similarities. By displaying three consecutive slices which can be scrolled and further visually explored, this tool enables a three-dimensional comparison. The presentation of three pairs at once facilitates the spontaneity of the annotation process.

6.5.2 Visual Similarity Ranking Performance

In contrast to our fast box-annotation scheme, Napel *et al.* [115] proposed a retrieval scheme for liver lesions which requires the exact segmentation of the lesion and an additional, manual specification of 161 semantic properties. They tested their approach on 30 lesion annotations. Our goal is to achieve rankings of a comparable quality over a larger database with a considerably smaller annotation effort (only box annotations, no semantic properties).

In order to minimize the annotation overhead for our large set of lesions, we decided to restrict the validation of the visual similarity search to subset V_1 of 111 liver lesions. For every lesion, we generated a ranking of the remain-

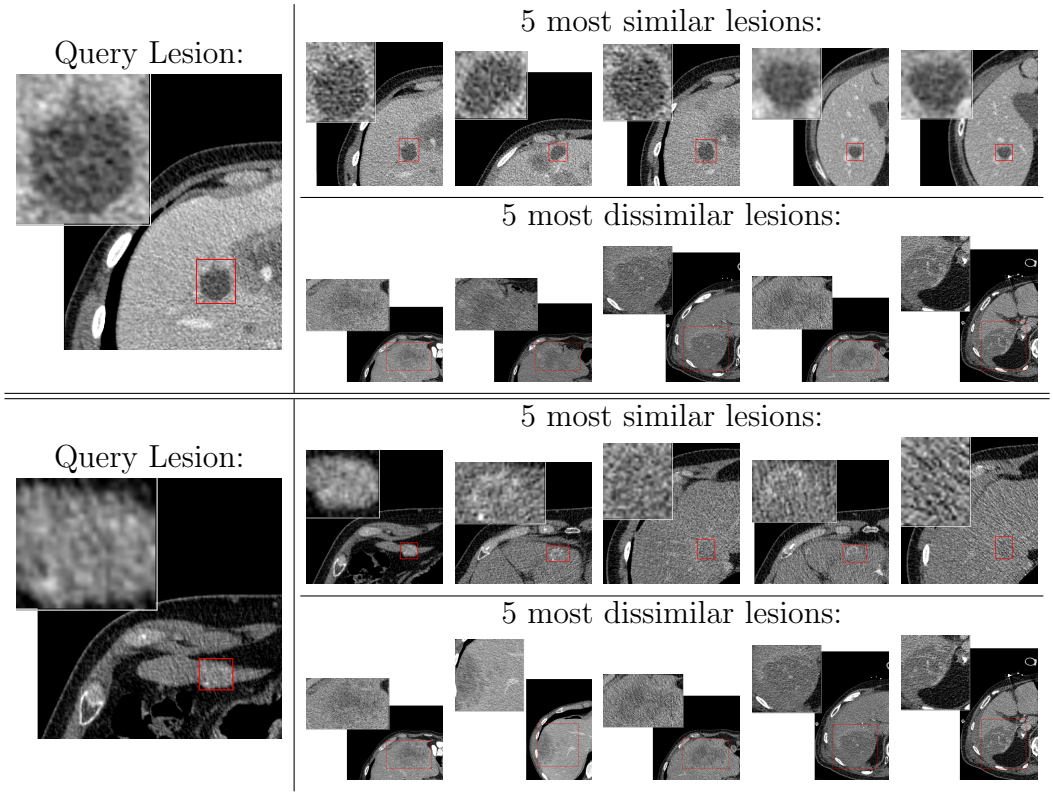
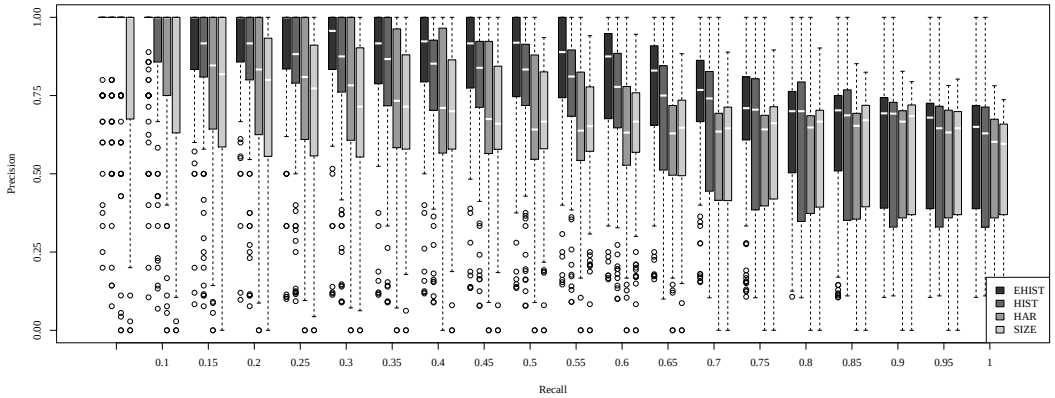


Figure 6.6: Example rankings for two queries by scribble. The annotations are displayed by the red bounding box with a close-up to the top left. These excerpts only show the center slice of the annotations, which may heavily vary in depth.

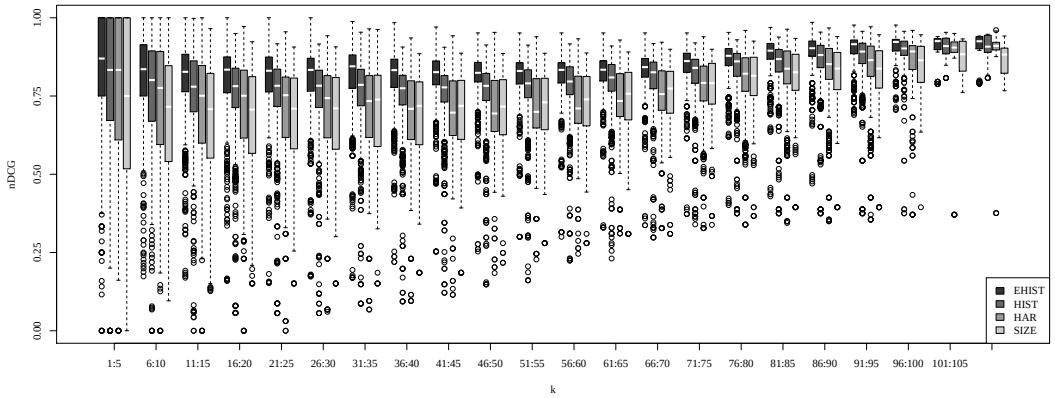
ing lesions according to their automatically-determined visual similarity. The first and the last ranked lesions of two such example rankings are depicted in Figure 6.6. In both cases, the first (and best) match in the top row is actually the same lesion, only originating from CT scans taken on another day.

Figure 6.7 shows the precision-recall (PR) curves (a pair is considered to be relevant for a similarity score ≥ 75) and the normalized discounted cumulative gain (nDCG) [88] aggregated over the complete set of 111 lesions. For both validation statistics, optimal rankings would maintain their maximum y-coordinate (either the precision or the nDCG) of 1 until the maximum value of the x-axis. Obviously, the plots do not display optimal rankings, however, they are clearly better than random, especially for the first k hits, which are the most relevant.

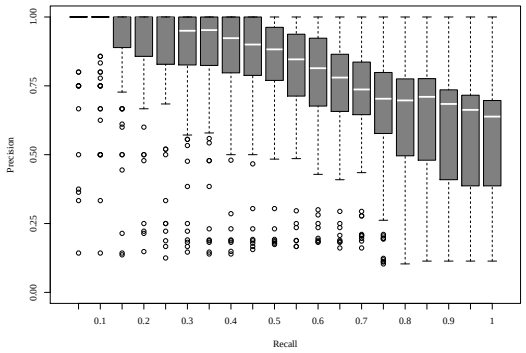
6.7(a) and 6.7(b) display the performance of the single image features: the 3D ellipsoid kernel of grey-value histogram EHIST, slice-wise grey-value histograms HIST, a slice-wise pyramid kernel of Haralick textures HAR, and



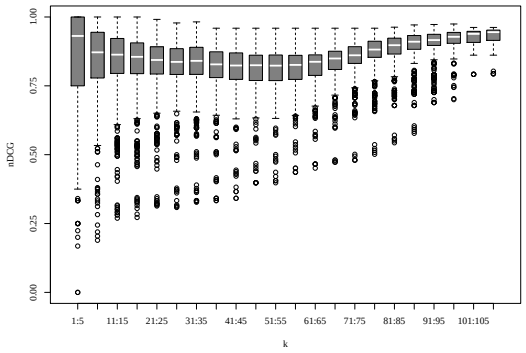
(a) Precision-Recall (PR) curves for four single descriptors.



(b) nDCG [88] curves of single descriptors. The k-nearest neighbors validation statistics are summarized in groups of 5.



(c) PR Curves of combined descriptors.



(d) nDCG curves of combined descriptors.

Figure 6.7: Evaluation of the rankings on V_1 via box-plots, displaying the median en-boxed by the first and third quantile. The whiskers represent the farthest non-outliers. (a), (b): rankings based on single image descriptors. (c), (d): rankings for the combined distance measure d_{combined}. The single distance contributions are weighted EHIST : HAR : SIZE = 2 : 1 : 1.

Table 6.3: Confusion matrix of ranking test on V_2 (60 lesions each from the kidneys, the liver and the spleen) displaying the 10-nearest neighbors (excluding the query) for the query organ in the rows.

(a) Confusion matrix: total precision 71.3 %

	Kidneys	Liver	Spleen
Kidneys	525	33	42
Liver	19	362	219
Spleen	10	194	396

(b) Same experiment in percent

	Kidneys	Liver	Spleen
Kidneys	87.5	5.5	7.0
Liver	3.2	60.3	36.5
Spleen	1.7	32.3	66.0

the simple size measure SIZE. We clearly see the improvement of the 3D ellipsoid kernel features EHIST over the multi-instance representation of 2D grey-value histograms HIST, which we employed in [138]. Since the multi-instance approach is also more expensive than the 3D lesion approximation, the choice of which grey-value histogram descriptor to use is obvious.

Therefore, the figures 6.7(c) and 6.7(d) validate the combined distance measure d_{combined} based on the three representations EHIST, HAR, and SIZE, using manual contribution weights 2 : 1 : 1. The resulting rankings result in a mean average precision of 0.78 and an average nDCG value for the 10^{th} retrieved lesion of 0.85.

This ranking procedure does not completely reach the quality of the validation results by Napel *et al.* [115], however, this is due to the rougher annotation quality and due to the larger size (111 instead of 30 lesions) of our dataset.

6.5.3 Benefit of the Combined Search

The quality of the above results gained a lot from our information combination approach. The information about the scribble's anatomic position enables to exclude all entities from the search space which are not localized within the liver. To test our hypothesis, we generated rankings on the dataset V_2 containing 60 lesions each from the liver, the spleen and the kidneys. When querying V_2 without using the semantic information about the organ context of the query lesion, 29 % of the top ten hits originate from foreign organs (cf. Table 6.3). The strongest error contribution arises from the general optical similarity between spleen and liver tissue. The miss-placed lesions appear to be similar for the image descriptors, but they are not useful in the context of a lesion query.

This is a major advantage of the MEDICO query system in comparison to other retrieval systems, where this information has to be filled in manually.

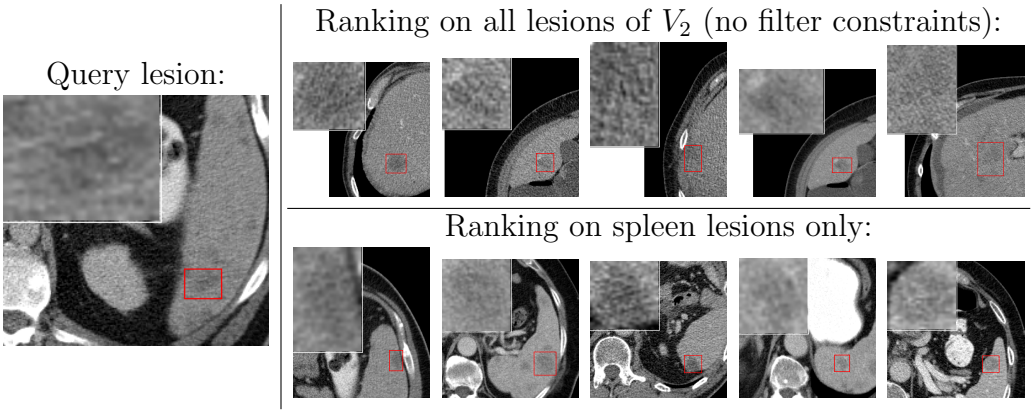


Figure 6.8: Example rankings for a spleen lesion query without (top row) and with (bottom row) organ constraints. Ranking all of V_2 returns only liver lesions in the top 5 hits.

The effect is exemplified by Figure 6.8, where similar spleen lesions will only appear in the top 5 ranks when applying an intelligent context filter.

The MEDICO system furthermore allows to specify manual semantic queries. In our example case the user wants to see all patients with enlarged thoracic lymph nodes. This query can be posed to the set of 100 semantically annotated volumes V_3 and it matches 34 patients in the *Annotation Database*. 10 patients have assigned lesion annotations and 9 of these patients show a total of 35 liver lesions in 26 volumes. The joint datasets of V_1 and V_2 can thus be restricted to a set of 35 instead of taking only the 111 liver lesions of V_1 by requesting a similar patient history. An exemplary ranking is displayed in Figure 6.9.

Besides the obvious benefit of restricting the result set to semantically valid items, the combination with semantic filter properties also speeds up the visual similarity ranking. A single query to V_2 takes 1 330 ms when the database is not cached for a quick main memory retrieval, including the time required for generating the query lesion's features (266 ms). The same query takes only 1 033 ms when adding the organ information "liver" (kidneys: 551 ms, spleen: 296 ms). When restricting the context to patients with enlarged lymph nodes, one query only takes 675 ms.

Naturally, the query process can be greatly sped up by caching the query database, however, in an environment not yet prepared for large-scale main memory storage, this procedure would interfere with other services. Thus, an intelligent filtering of the query database is an important step for a well-performing similarity query.

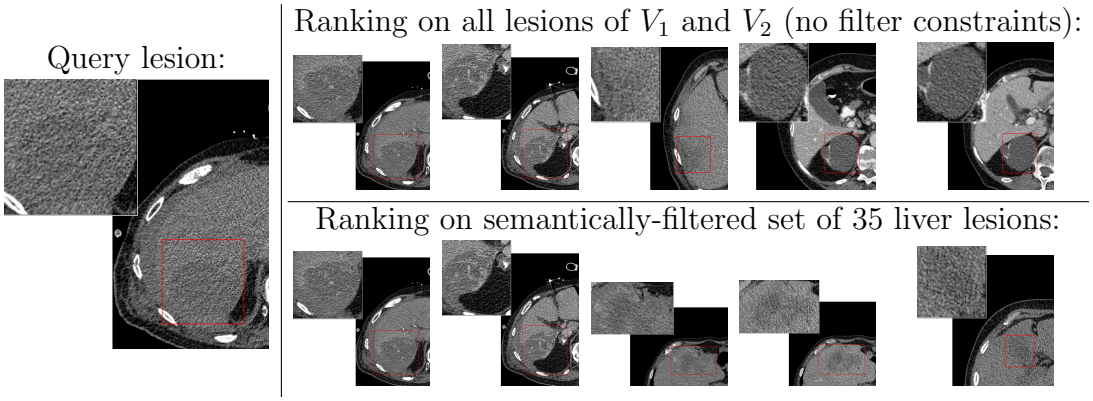


Figure 6.9: Example rankings for a liver lesion query without (top row) and with (bottom row) the semantic constraint "in liver, thoracic lymph nodes enlarged". The first two hits of both rankings each show the query lesion in various stages. When all of V_2 is queried, two kidney lesions are among the top 5 hits.

6.6 Summary

This chapter presented search components of the MEDICO prototype, a comprehensive medical search framework which enables the user to accomplish visual similarity search combined with semantic search based on web 2.0 technology. [138] Our approach significantly extends the search capabilities compared with currently available content-based image retrieval systems and enables the system to answer real-world questions.

The MEDICO prototype allows both semantic queries in the form of a query by concept and instance-based image similarity queries. A query by concept is an extension of a query by semantic keyword, allowing flexible concept combinations and the automatic query expansion of a more general concept to its subclasses. In a query by scribble, our implementation of instance-based similarity search, the user defines an arbitrary region of interest (ROI) within a 3D volume and asks the system for similar ROI annotations. Additionally, the two query types can be combined in order to complement the other query type's advantages and to provide an even more flexible query framework.

In our query scenario of intra-organ lesion search, the result set can be automatically reduced by taking the semantic information about the containing organ of the lesion into account. Another reduction can be achieved by reducing a result set of 111 to 35 liver lesions by including an additional semantic search criterion (*thoracic lymph nodes enlarged*). On the one hand, this approach increases the relevance of instance-based image similarity queries by restricting the result according to a similar patient pathology. On the other

hand, we can exploit this effect for greatly speeding up our similarity queries.

In future work we aim toward improving the quality of the image-based query component by testing further image descriptors and similarity measures. At the moment, we additionally try to improve the quality of our image descriptors by incorporating a lesion segmentation step for detailing the imprecise box annotations. In most cases, rather simple threshold-based segmentation approaches are not sufficient for defining a reliable lesion boundary. By incorporating various improvements like image smoothing kernels, lesion probability assumptions and clustering approaches, we were able to generate visually plausible 3D lesion segmentations. Nevertheless, so far, this new knowledge about whether or not an MBR voxel is within or outside of the lesion, did not sufficiently improve the ranking quality to justify the additional cost required for this fine-segmentation step.

Furthermore, we will look for ways of refining the query combination mechanism and we plan to test our system on larger sets of annotated data. Finally, we are integrating alternative types of medical data sources like laboratory reports into the query system.

Chapter 7

Discussion and Outlook

In this thesis, the author summarized general findings about computer-based similarity search with a special focus on the use in medical imaging. Most paradigms of similarity search also hold in the field of medicine, however, a number of additional constraints need to be considered.

7.1 Practical Barriers

Within the research and implementation work of this thesis, the main obstacles on a successful image similarity search were:

Data Availability Even with a close cooperation to one of the leading radiology centers in Germany it proved to be challenging to acquire a sufficient or appropriate amount of medical image data. This general problem is mostly due to privacy protection measures which are complicated by the numerous image types and formats. The advances in imaging prove the value of these sanctions. As there is hardly any kind of personal data which is more private than medical data, no patient wants to see their laboratory values posted on the web. And faces have been reconstructed from bone surface scans [157] or CT scans [125] for a long time.

Data Standardization In addition to the availability issue of medical data, an even more pressing problem is its state of standardization. Current standards in medical imaging [117, 50] are either insufficient for providing comparable image material or they are ignored by some manufacturers of image scanning technology. Any information retrieval approach developed on medical datasets must thus be robust w.r.t. the manifold nature of the expected input.

Ground Truth Annotations Defining a solid ground truth on medical images showed to be a further hindrance on the way to a useful image search application. A medical expert will rarely compare two images solely based on the visual appearance. Usually some preliminary information about the patient's background is known in order to better place the image material to be examined. It is therefore very difficult to get such an expert to make a clear statement about their perceived impression of similarity if this information is missing.

Appropriate Pattern Mining Originally, the development of new image descriptors was not within the scope of this work. There is already a multitude of image descriptors available in the literature, however, these are usually designed for rather specific retrieval approaches or they are too complex for being suited for an efficient image retrieval. In order to correctly represent the various special cases of medical images handled within this thesis, we had to explore various strategies for generating image descriptors.

Multidisciplinarity The most demanding challenge of our inquiries in the field of medical image queries was the multiplicity of relevant fields of research. Our developed solutions touch various areas of similarity search, data mining, data indexing, machine learning, pattern mining and software development. It therefore proved to be of utmost importance to clearly define self-contained milestones and sub-projects when aiming toward the goal of an integrated and holistic prototype, but to never loose track of the global picture.

7.2 List of Scientific Contributions

Most of the contributions of this thesis have been directly integrated into the THESEUS MEDICO image query framework. The following section gives a brief summary of their main components.

7.2.1 Improved Data Integration

In [146], we introduced the architectural background of the THESEUS MEDICO solution to the integration of various kinds of data sources. The crosslinks between various types of databases can also be implemented manually, however, in order to enable the fast integration of a further data source, we use a generic framework for query processing. Therefore, the relevant query options

of the MEDICO image search components have been standardized according to the MPEG query format (MPQF) [48] within the AIR [145] architecture.

7.2.2 Flexible Paths to Similarity Search

A major focus of this thesis was to ensure the flexibility of the developed similarity search framework. We therefore tested various types of image descriptors and object representations on their applicability within a similarity retrieval system, ranging from classical, real-valued feature vectors [54] over set-valued image descriptors [138] to graphs [153]. In the course of this research, we generated a new machine learning scheme of Bayes Ensembles [54], we investigated feature selection and feature mining approaches in graph applications [153] and we explored two types of feature combination mechanisms in similarity ranking [138] and instance-based regression [55].

7.2.3 Definition and Solution of two CBIR Use Cases

The practical part of this work is mostly centered around two applications of content-based image retrieval (CBIR) in medical data. In the first use case, [25] we exploit visual properties of CT scans for defining a standardized body height atlas. This atlas is employed for automatically providing anatomical information on selected body regions and for accelerating volume retrieval. The second use case is an example-based image query system, allowing the user to define a region of interest (ROI) within a CT scan for which to retrieve the most similar ROIs of a database. [138] In addition, we combine this image-based similarity query option with a semantic query framework, allowing queries for semantically-standardized anatomical keywords [137], disease stages [185] or laboratory measurement ranges.[1]

7.2.4 Solutions for Efficient Image Retrieval

Finally, our research in the field of medical image similarity search always emphasizes the necessity of efficient query runtimes. In order to respect the special requirements of a medical environment, the precision of a search system's answers should not suffer, however, the daily routine in radiology does not allow for long waiting times.

We therefore closely examined the runtime requirements of the feature selection and subgraph mining approach introduced in [153] and provided guidelines for deciding, which selection variant to use under which given conditions. The main gain of the multi-instance query approaches discussed in Chapter 4

[1] Patent pending.

also consisted in a speed-up of similarity retrieval queries. Moreover, we investigated various methods for the special case of all-nearest-neighbor queries in [53] and explored the behaviour of spatial index structures on solid state disks [52]. For the brevity of this thesis, these results are only referenced.

Eventually, we established an algorithm for the accelerated retrieval of subvolumes [25] using spatial indexes and the abstraction layer of a standardized body coordinate system. Additionally, in [138], we exploit information generated by an automatic anatomical image parsing system [136] and results of semantic queries as filters for speeding up image-based similarity queries.

7.3 Summary

This thesis presented parts of the scientific results concerning similarity search in medical image databases of the THESEUS MEDICO research project. It gave an overview on the computational challenges of image-retrieval in a medical environment and presented solutions for two chosen use cases.

Image retrieval in medical applications will remain a very active field of research. Due to a rapidly growing data pool of medical images, the major vendors of computational radiology equipment have intensified their efforts in finding solutions for capitalizing this enormous source of medical information. This will also necessitate the increased use of standardized reporting techniques and the use of new electronic documentation media in hospitals. Both areas of research also fall within the scope of the THESEUS MEDICO project.

In future work, we plan to improve the existing technologies of the two proposed medical CBIR scenarios and to develop new applications for the image retrieval components developed within this thesis. We aim toward tackling new challenges of data collections like irregular time-series' of volumetric images or videos of different modalities. And finally, we are going to look for further, non-medical application domains of the similarity search techniques developed for the use in medical image queries.

Appendix A

List of Abbreviations

2D / 3D	two-dimensional / three-dimensional
BE	Bayes Estimate [54]
BED	Bayes Ensemble Distance [54]
CBIR	Content-based Image Retrieval
cm	centimeters
CORK	Correspondence-based Quality Criterion [153]
CT	Computed Tomography
DFS	Depth-First Search
DICOM	Digital Imaging and Communications in Medicine [117]
GB	gigabytes
gSpan	graph-based Substructure pattern mining [177]
HL7	Health Level 7 [50]
HMD	Half the Sum of Minimum Distances
HU	Hounsfield Unit [22]
I/O	input/output
LOO	leave-one-out
MBR	minimum bounding rectangle
MILES	Multiple-Instance Learning via Embedded Instance Selection [29]
PACS	Picture Archiving and Communication System
RCA	Relevant Component Analysis [7]
ROI	Region of Interest
SIFT	Scale-Invariant Features Transform [106, 107]
SMD	Sum of Minimum Distances
SURF	Speeded Up Robust Features [10]
UID	unique identifier

List of Figures

2.1 gSpan: Rightmost Extension 19
2.2 CORK Supergraph Relations 23
2.3 Threshold Screening 39
2.4 Nested Examples 40

3.1 Examplary Difference Distributions 52
3.2 Bivariate Difference Distributions 55
3.3 Classification Results on Several UCI Datasets 61
3.4 Precision-Recall Graphs on the Conf and Flowers Dataset 63
3.5 Different Versions of BE on Conf-hsv 64

4.1 Example Images of the Conf Dataset 86
4.2 Dimension Reduction of the SIFT Descriptor 87
4.3 Multi-Instance Query Runtimes 92
4.4 Multi-Instance Query Runtimes for Lower Dimensions 93

5.1 Examples for ROI Queries 100
5.2 Workflow of ROI Retrieval 107
5.3 Relevance Terms and Interpolation Functions 111
5.4 Exemplary Landmark and Organ Distributions 114
5.5 Landmark and Organ Atlas 115
5.6 Instance-Based Regression Scheme 116
5.7 Modified Spatial Pyramid Kernels 118
5.8 Image Distortion and Grey-Value Windows 119
5.9 First Steps of Algorithm 8 126
5.10 Seed Atlas and Interpolation Validation 130
5.11 Atlas Validation for the Number of Matching Points 130
5.12 Multi-Represented Regression 134
5.13 Example for Manual Landmark Definition 138
5.14 Cumulative Distribution Function of ROI Queries 139
5.15 Runtimes of ROI Queries 140

6.1 Annotation Ontology Scheme 146
6.2 Query by Scribble Example 148
6.3 Visualization of the Ellipsoid Shell Kernel 150
6.4 Query by Scribble Workflow 152
6.5 3D Similarity Annotation Tool 160
6.6 Example 3D Rankings 161
6.7 Evaluation of 3D Ranking Queries 162
6.8 Example Ranking of a Spleen Lesion Query 164
6.9 Example for the Effect of a Combined Query 165

List of Tables

2.1 Topologies of Used Graph Sets 32
2.2 DD6C Class Distribution 34
2.3 Classification Quality of Filter Methods 37
2.4 Nested Versus Un-Nested Selection 41
2.5 Comparison to Wrapper Approaches 43

3.1 Image Retrieval Datasets 62

4.1 Summary of the Musk Datasets 86
4.2 Summary of the Image Datasets using SIFT Descriptors 88
4.3 Accuracies of Basic Multi-Instance Distance Measures 90
4.4 Accuracies of Index-Based Multi-Instance Distance Measures . . 91

5.1 Notation of Frequently Used Parameters 106
5.2 Slice Regression: Single Features 133
5.3 Slice Regression: Various Database Sizes 135
5.4 Slice Regression: Reduced Dimensions 136
5.5 Quality of ROI Queries 139

6.1 Structure of the Image Feature Database 155
6.2 Content of the Image Feature Database 156
6.3 Confusion Matrix of Lesion Retrieval for Various Organs 163

List of Algorithms

1 gSpan 20
2 $\text{gSpan}_{\text{CORK}}$ 25
3 $\text{Offline\_Select}_{\text{CORK}}$ 29
4 Sequential Cover (SC) 36
5 BED Training 57
6 k-Minimum Linkage Classification 82
7 Atlas Refinement 113
8 Online ROI Query 127

Bibliography

[1] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), Santiago de Chile, Chile*, pages 487–499, 1994.

[2] Ceyhun Burak Akgül, Daniel L. Rubin, Sandy Napel, Christopher F. Beaulieu, Hayit Greenspan, and Burak Acar. Content-based image retrieval in radiology: Current status and future directions. *Journal of Digital Imaging*, 24:208–222, 2011.

[3] Gowri Allampalli-Nagaraj and Isabelle Bichindaritz. Automatic semantic indexing of medical images using a web ontology language for case-based image retrieval. *Engineering Applications of Artificial Intelligence*, 22(1):18–25, 2009.

[4] Sameer Antani, Zhiyun Xue, L. Rodney Long, Deborah Bennet, Sarah Ward, and George R. Thoma. Is there a need for biomedical CBIR systems in clinical practice?: outcomes from a usability study. In *Proceedings of the SPIE Medical Imaging Conference 2011: Advanced PACS-based Imaging Informatics and Therapeutic Applications, Lake Buena Vista, FL, USA*, volume 7967, page 796708, 2011.

[5] Amos Bairoch. The ENZYME database in 2000. *Nucleic Acids Research*, 28(1):304–305, 2000.

[6] Alfonso Baldi, Raffaele Murace, Emanuele Dragonetti, Mario Manganaro, Oscar Guerra, Stefano Bizzi, and Luca Galli. Definition of an automated content-based image retrieval (CBIR) system for the comparison of dermoscopic images of pigmented skin lesions. *BioMedical Engineering OnLine*, 8(1):18, 2009.

[7] Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. Learning distance functions using equivalence relations. In *Proceedings of*

the 20th International Conference on Machine Learning (ICML), Washington, DC, pages 11–18, 2003.

[8] Julia F. Barret and Nicholas Keat. Artifacts in CT: Recognition and avoidance. *Radiographics*, 24(6):1679–1691, 2004.

[9] Peter Baumann, Paula Furtado, Roland Ritsch, and Norbert Widmann. The RasDaMan approach to multidimensional database management. In *Proceedings of the 12th ACM Symposium on Applied Computing (ACM SAC), San Jose, CA*, pages 166–173. ACM, 1997.

[10] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. *Computer Vision and Image Understanding*, 110:346–359, June 2008.

[11] Ruan J. S. Belém, João M. B. Cavalcanti, Edleno Silva de Moura, and Mario A. Nascimento. SNIF: A simple nude image finder. In *Third Latin American Web Congress (LA-Web 2005), Buenos Aires, Argentina*, pages 252–258, 2005.

[12] Peter N. Belhumeur, João P. Hespanha, and David J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[13] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[14] E. Bellon, M. Feron, F. Maes, L. van Hoe, D. Delaere, F. Haven, S. Sunaert, AL Baert, G. Marchal, and P. Suetens. Evaluation of manual vs semi-automated delineation of liver lesions on CT images. *European Radiology*, 7(3):432–438, 1997.

[15] S. Berchtold, D. A. Keim, and H.-P. Kriegel. The X-Tree: An index structure for high-dimensional data. In *Proceedings of the 22nd International Conference on Very Large Data Bases (VLDB), Bombay, India*, 1996.

[16] Christian Borgelt and Michael Berthold. Mining molecular fragments: Finding relevant substructures of molecules. In *Proceedings of the 2nd IEEE International Conference on Data Mining (ICDM), Maebashi City, Japan*, pages 211–218, 2002.

[17] Christian Borgelt, Thorsten Meinl, and Michael Berthold. MoSS: a program for molecular substructure mining. In *Proceedings of the 1st Open Source Data Mining Workshop (OSDM 2005) on Frequent Pattern Mining Implementations in conjunction with 11th ACM SIGKDD, Chicaco, IL, USA*, pages 6–15. ACM, 2005.

[18] Karsten M. Borgwardt, Xifeng Yan, Marisa Thoma, Hong Cheng, Arthur Gretton, Lee Song, Alex Smola, Jiawei Han, Philip S. Yu, and Hans-Peter Kriegel. Combining near-optimal feature selection with gSpan. In *Proceedings of the 6th International Workshop on Mining and Learning with Graphs (MLG), Helsinki, Finland*, 2008.

[19] Endre Boros, Takashi Horiyama, Toshihide Ibaraki, Kazuhisa Makino, and Mutsunori Yagiura. Finding essential attributes from binary data. *Annals of Mathematics and Artificial Intelligence*, 39(3):223–257, 2003.

[20] Anna Bosch, Andrew Zisserman, and Xavier 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*, pages 401–408. ACM, 2007.

[21] Björn Bringmann and Albrecht Zimmermann. One in a million: picking the right patterns. *Knowledge and Information Systems (KAIS)*, 18:61–81, 2008.

[22] Rodney A. Brooks. A quantitative theory of the Hounsfield Unit and its application to dual energy scanning. *Journal of Computer Assisted Tompography*, 1(4):487–493, 1977.

[23] Stefan Büttcher, Charles L. A. Clarke, and Gordon V. Cormack. *Information Retrieval: Implementing and Evaluating Search Engines*. MIT Press, 7 2010.

[24] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[25] Alexander Cavallaro, Franz Graf, Hans-Peter Kriegel, Matthias Schubert, and Marisa Thoma. Region of interest queries in CT scans. In *Proceedings of the 12th International Symposium on Spatial and Temporal Databases (SSTD), Minneapolis, MN, USA*, pages 56–73, 2011.

[26] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[27] David S. Channin, Pattanasak Mongkolwat, Vladimir Kleper, and Daniel L. Rubin. The annotation and image mark-up project. *Radiology*, 253(3):590–592, 2009.

[28] David S. Channin, Pattanasak Mongkolwat, Vladimir Kleper, Kastubh Sepukar, and Daniel L. Rubin. The caBIG annotation and image markup project. *Journal of Digital Imaging*, 23:217–225, 2010.

[29] Yixin Chen, Jinbo Bi, and James Z. Wang. MILES: multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1931–1947, 2006.

[30] Hong Cheng, Xifeng Yan, Jiawei Han, and Chih-Wei Hsu. Discriminative frequent pattern analysis for effective classification. In *Proceedings of the 23rd International Conference on Data Engineering (ICDE), Istanbul, Turkey*, pages 716–725, 2007.

[31] G. Chinga, O. Gregersen, and B. Dougherty. Paper surface characterisation by laser profilometry and image analysis. *Journal of Microscopy and Analysis*, 84:5–7, 2003.

[32] William S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836, 1979.

[33] Pierre Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994.

[34] Trevor F. Cox and Michael A. A. Cox. *Multidimensional Scaling*. Chapman & Hall/CRC, 2nd edition, 2001.

[35] Antonio Criminisi, Jamie Shotton, Duncan Robertson, and Ender Konukoglu. Regression forests for efficient anatomy detection and localization in CT studies. In *Proceedings of the MICCAI 2010 Workshop: Medical Computer Vision (MCV), Beijing, China*, pages 106–117. Springer, 2011.

[36] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05), Washington, DC, USA*, pages 886–893, 2005.

[37] Gary H. Danton. Radiology reporting: changes worth making are never easy. *Applied Radiology*, 39(5):19–23, 2010.

[38] Manoranjan Dash, Huan Liu, and Hiroshi Motoda. Consistency based feature selection. In *Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Kyoto, Japan*, pages 98–109, London, UK, 2000. Springer-Verlag.

[39] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40:1–60, May 2008.

[40] Jason Davis, Brian Kulis, Suvrit Sra, and Inderjit Dhillon. Information-theoretic metric learning. In *Proceedings of the Workshop on Learning to Compare Examples at the 23rd Annual Conference on Neural Information Processing Systems (NIPS), Whistler, BC, Canada*, 2007.

[41] dcm4che open-source community. Open source clinical image and object management (dcm4che), 2011. `http://www.dcm4che.org`.

[42] Da Deng. Content-based image collection summarization and comparison using self-organizing maps. *Pattern Recognition*, 40(2):718–727, 2007.

[43] Enrica Dente, Anil Anthony Bharath, Jeffrey Ng, Aldert Vrij, Samantha Mann, and Anthony Bull. Tracking hand and finger movements for behaviour analysis. *Pattern Recognition Letters*, 27:1797–1808, November 2006.

[44] Thomas Deselaers, Daniel Keysers, and Hermann Ney. Features for image retrieval: an experimental comparison. *Information Retrieval*, 11:77–107, April 2008.

[45] Mukund Deshpande, Michihiro Kuramochi, Nikil Wale, and George Karypis. Frequent substructure-based approaches for classifying chemical compounds. *IEEE Transactions on Knowledge and Data Engineering*, 17(8):1036–1050, 2005.

[46] Thomas G. Dietterich, Richard H. Lathrop, and Tomas Lozano-Perez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89:31–71, 1997.

[47] Paul D. Dobson and Andrew J. Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of Molecular Biology*, 330(4):771–783, Jul 2003.

[48] Mario Döller, Ruben Tous, Matthias Gruhne, Kyoungro Yoon, Masanori Sano, and Ian S Burnett. The MPEG Query Format: On the way to unify the access to multimedia retrieval systems. *IEEE Multimedia*, 15(4):82–95, 2008.

[49] James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised discretization of continuous features. In *Proceedings of the 12th International Conference on Machine Learning (ICML), Tahoe City, CA*, pages 194–202. Morgan Kaufmann, 1995.

[50] e-Health Web Services. Joint initiative on SDO global health informatics standardization, 2011. `http://www.global-e-health-standards.org/`.

[51] Liat Ein-Dor, Or Zuk, and Eytan Domany. Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proceedings of the National Academy of Sciences of the United States of America*, 103(15):5923–5928, 4 2006.

[52] Tobias Emrich, Franz Graf, Hans-Peter Kriegel, Matthias Schubert, and Marisa 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, USA*, pages 3–8, 2010.

[53] Tobias Emrich, Franz Graf, Hans-Peter Kriegel, Matthias Schubert, and Marisa Thoma. Optimizing all-nearest-neighbor queries with trigonometric pruning. In *Proceedings of the 22nd International Conference on Scientific and Statistical Database Management (SSDBM), Heidelberg, Germany*, volume 6187, pages 501–518, 2010.

[54] Tobias Emrich, Franz Graf, Hans-Peter Kriegel, Matthias Schubert, and Marisa Thoma. Similarity estimation using Bayes Ensembles. In *Proceedings of the 22nd International Conference on Scientific and Statistical Database Management (SSDBM), Heidelberg, Germany*, volume 6187, pages 537–554, 2010.

[55] Tobias Emrich, Franz Graf, Hans-Peter Kriegel, Matthias Schubert, Marisa Thoma, and Alexander Cavallaro. CT slice localization via instance-based regression. In *Proceedings of the SPIE Medical Imaging Conference 2010: Image Processing (SPIE), San Diego, CA, USA*, page 762320, 2010.

[56] Alan C. Evans, D. L. Collins, S. R. Mills, E. D. Brownand, R. L. Kelly, and Tricia M. Peters. 3D statistical neuroanatomical models from 305

MRI volumes. In *IEEE Nuclear Science Symposium and Medical Imaging Conference*, 1993.

[57] Wei Fan, Kun Zhang, Hong Cheng, Jing Gao, Xifeng Yan, Jiawei Han, Philip S. Yu, and Olivier Verscheure. Direct mining of discriminative and essential frequent patterns via model-based search tree. In Ying Li, Bing Liu, and Sunita Sarawagi, editors, *Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Las Vegas, NV*, pages 230–238. ACM, 2008.

[58] Li Fei-Fei, Rod Fergus, and Pietro 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.

[59] Johannes Feulner, S. Kevin Zhou, Sascha Seifert, Alexander Cavallaro, Joachim Hornegger, and Dorin Comaniciu. Estimating the body portion of CT volumes by matching histograms of visual words. In *Proceedings of the SPIE Medical Imaging Conference 2009 (SPIE), Lake Buena Vista, FL, USA*, volume 7259, page 72591V, 2009.

[60] Raphael A. Finkel and Jon L. Bentley. Quad Trees: a data structure for retrieval on composite keys. *Acta Informatica*, 4:1–9, 1974.

[61] Benedikt Fischer, Michael Sauren, Mark O. Güld, and Thomas M. Deserno. Scene analysis with structural prototypes for content-based image retrieval in medicine. In *Proceedings of the SPIE Medical Imaging Conference 2008: Image Processing (SPIE), San Diego, CA, USA*, page 6914, 2008.

[62] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[63] Yoav Freund and Robert E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, 1999.

[64] Andrea Frome, Fei Sha, Yoram Singer, and Jitendra Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV), Rio de Janeiro, Brazil*, pages 1–8. Berkeley, 2007.

[65] Andrea Frome, Yoram Singer, and Jitendra Malik. Image retrieval and classification using local distance functions. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Proceedings of the 19th Annual Conference on Neural Information Processing Systems (NIPS), Vancouver, Canada*, pages 417–424. MIT Press, Cambridge, MA, 2007.

[66] Thomas Gärtner, Peter A. Flach, Adam Kowalczyk, and Alex J. Smola. Multi-instance kernels. In *Proceedings of the 19th International Conference on Machine Learning (ICML), Sydney, Australia*, pages 179–186. Morgan Kaufmann Publishers Inc., 2002.

[67] Jacob Goldberger, Sam Roweis, Geoff Hinton, and Ruslan Salakhutdinov. Neighborhood component analysis. In *Proceedings of the 17th Annual Conference on Neural Information Processing Systems (NIPS), Vancouver, Canada*, pages 513–520. MIT Press, 2004.

[68] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[69] Franz Graf, Hans-Peter Kriegel, Matthias Schubert, Sebastian Pölsterl, and Alexander Cavallaro. 2D image registration in CT images using radial image descriptors. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI), Toronto, Canada*, 2011.

[70] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A kernel method for the two-sample-problem. In *Proceedings of the 19th Annual Conference on Neural Information Processing Systems (NIPS), Vancouver, Canada*, 2006.

[71] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A kernel method for the two-sample problem. *The Computing Research Repository*, abs/0805.2368, 2008.

[72] Greg Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[73] Carlos Guestrin, Andreas Krause, and Ajit Paul Singh. Near-optimal sensor placements in gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning (ICML), Bonn, Germany*, pages 265–272, 2005.

[74] Mark O. Güld, Michael Kohnen, Daniel Keysers, Henning Schubert, Berthold B. Wein, Jörg Bredno, and Thomas M. Lehmann. Quality

of DICOM header information for image categorization. In *Proceedings of the SPIE Medical Imaging Conference 2002 (SPIE), San Diego, CA, USA*, pages 280–287, 2002.

[75] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, March 2003.

[76] 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.

[77] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations*, 11(1):10–18, 2009.

[78] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2nd edition, 1 2006.

[79] Robert M. Haralick, K. Shanmugam, and Its'hak Dinstein. Textural features for image classification. *IEEE Transactions on Speech and Audio Processing*, 3(6):610–623, 1973.

[80] Justin Heesemann. Automatic annotation of medical volume datasets. Diplomarbeit, Ludwig-Maximilians-Universität München, Munich, Germany, 3 2010.

[81] Setz Hettich and Stephen D. Bay. The UCI KDD archive. http://kdd.ics.uci.edu, 1999.

[82] Derek L. G. Hill, Philipp G. Batchelor, Mark Holden, and David J. Hawkes. Medical image registration. *Physics in Medicine and Biology*, 46:R1–R45, 2001.

[83] Paul W. Holland and Roy E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-Theory and Methods*, 6(9):813–827, 1977.

[84] Jun Huan, Wei Wang, and Jan Prins. Efficient mining of frequent subgraph in the presence of isomorphism. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM), Melbourne, FL*, pages 549–552, 2003.

[85] Akihiro Inokuchi, Takashi Washio, and Hiroshi Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discoverys (PKDD), Lyon, France*, pages 13–23, 2000.

[86] International Health Terminology Standards Development Organisation. IHTSDO: SNOMED CT, 2011. `http://www.ihtsdo.org/snomed-ct/`.

[87] David W. Jacobs, Daphna Weinshall, and Yoram Gdalyahu. Classification with non-metric distances: Image retrieval and class representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):583–600, 2000.

[88] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[89] Ning Jin, Calvin Young, and Wei Wang. Graph classification based on pattern co-occurrence. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM), Hong Kong, China*, pages 573–582, New York, NY, USA, 2009. ACM.

[90] R. H. Johr. Dermoscopy: alternative melanocytic algorithms – the ABCD rule of dermatoscopy, menzies scoring method, and 7-point checklist. *Clinics in dermatology*, 20(3):240–247, 2002.

[91] Michael J. Jones and James M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.

[92] Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. Marginalized kernels between labeled graphs. In *Proceedings of the 20th International Conference on Machine Learning (ICML), Washington, DC*, pages 321–328, 2003.

[93] Yan Ke and Rahul Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04), Washington, DC, USA*, volume 2, pages 506–513, 2004.

[94] Daniel Korenblum, Daniel Rubin, Sandy Napel, Cesar Rodriguez, and Chris Beaulieu. Managing biomedical image metadata for search and retrieval of similar images. *Journal of Digital Imaging*, pages 1–10, 2010.

[95] Stefan Kramer, Luc De Raedt, and Christoph Helma. Molecular feature mining in HIV data. In *Proceedings of the 7th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), San Francisco, CA*, pages 136–143, 2001.

[96] Andreas Krause and Carlos Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Proceedings of the 21st International Conference on Uncertainty in Artificial Intelligence (UAI), Edinburgh, Scotland*, pages 324–331, 2005.

[97] Hans-Peter Kriegel, Peter Kunath, Alexey Pryakhin, and Matthias Schubert. MUSE: multi-represented similarity estimation. In *Proceedings of the 24th International Conference on Data Engineering (ICDE), Cancun, Mexico*, 2008.

[98] Hans-Peter Kriegel and Matthias Schubert. Classification of websites as sets of feature vectors. In *Proceedings of the IASTED International Conference on Databases and Applications (DBA), Innsbruck, Austria*, 2004.

[99] Hans-Peter Kriegel, Bernhard Seeger, Ralf Schneider, and Norbert Beckmann. The R*-tree: An efficient access method for geographic information systems. In *Proceedings of the International Conference on Geographic Information Systems, Ottawa, Canada*, 1990.

[100] Hugo Kubinyi. Drug research: myths, hype and reality. *Nature Reviews: Drug Discovery*, 2:665–668, 2003.

[101] T. Kudo, E. Maeda, and Y. Matsumoto. An application of boosting to graph classification. In *Proceedings of the 17th Annual Conference on Neural Information Processing Systems (NIPS), Vancouver, Canada*, pages 729–736, Dec. 2004.

[102] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In *Proceedings of the 1st IEEE International Conference on Data Mining (ICDM), San Jose, CA*, pages 313–320, 2001.

[103] Curtis P. Langlotz. RadLex: A new method for indexing online educational materials. *Radiographics*, 26(6):1595–1597, 2006.

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

[105] Thomas M. Lehmann, Daniel Beier, Christian Thies, and Thomas Seidl. Segmentation of medical images combining local, regional, global, and hierarchical distances into a bottom-up region merg ingscheme. In *Proceedings of the SPIE Medical Imaging Conference 2005: Image Processing (SPIE), Lake Buena Vista, FL, USA*, pages 546–555, 2005.

[106] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV), Kerkyra, Greece*, pages 1150–1157, 1999.

[107] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[108] J. B. Antoine Maintz and Max A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.

[109] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1):50–60, March 1947.

[110] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 1st edition, 7 2008.

[111] Arne Militzer, Tobias Hager, Florian Jäger, Christian Tietjen, and Joachim Hornegger. Automatic detection and segmentation of focal liver lesions in contrast enhanced CT images. In *Proceedings of the 20th International Conference on Pattern Recognition (ICPR), Washington, DC, USA*, ICPR '10, pages 2524–2527. IEEE Computer Society, 2010.

[112] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710, 1997.

[113] Manuel Möller and Michael Sintek. A generic framework for semantic medical image retrieval. In *Proc. of the Knowledge Acquisition from Multimedia Content (KAMC) Workshop, 2nd International Conference on Semantics And Digital Media Technologies (SAMT)*, November 2007.

[114] Henning Müller and Thomas M. Deserno. Content-based medical image retrieval. In Thomas M. Deserno, editor, *Biomedical Image Processing*, pages 471–494. Springer Berlin Heidelberg, 2011.

[115] Sandy A. Napel, Christopher F. Beaulieu, Cesar Rodriguez, Jingyu Cui, Jiajing Xu, Ankit Gupta, Daniel Korenblum, Hayit Greenspan, Yongjun Ma, and Daniel L. Rubin. Automated retrieval of CT images of liver lesions on the basis of image similarity: Method and preliminary results. *Radiology*, 256(1):243–252, 2010.

[116] Mario A. Nascimento, Veena Sridhar, and Xiaobo Li. Effective and efficient region-based image retrieval. *Journal of Visual Languages and Computing*, 14(2):151–179, 2003.

[117] National Electrical Manufacturers Association (NEMA). DICOM homepage, 2011. `http://medical.nema.org/`.

[118] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming*, 14:265–294, 1978.

[119] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases, 1998.

[120] T. D. Nguyen. *Robust estimation, regression and ranking with applications in portfolio optimization.* PhD thesis, Massachusetts Institute of Technology, 2009.

[121] S. Nijssen and J. Kok. A quickstart in frequent structure mining can make a difference. In *Proceedings of the 10th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Seattle, WA*, pages 647–652, 2004.

[122] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06), New York, NY, USA*, 2:1447–1454, 2006.

[123] David Nistér and Henrik Stewénius. Scalable recognition with a vocabulary tree. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06), New York, NY, USA*, volume 5, pages 2161–2168, 2006.

[124] Nataŝa Prŵulj. Biological network comparison using graphlet degree distribution. In *Proceedings of the 5th European Conference on Computational Biology (ECCB), Eilat, Israel*, September 2006.

[125] Gérald Quatrehomme, Stéphane Cotin, Gérard Subsol, Hervé Delingette, Yves Garidel, G. Grévin, Martha Fidrich, Paul Bailet, and Amédée Ollier. A fully three-dimensional method for facial reconstruction based on deformable models. *Journal of Forensic Sciences*, 42(2):649–652, 1997.

[126] Predrag Radivojac, Zoran Obradovic, A. Keith Dunker, and Slobodan Vucetic. Feature selection filters based on the permutation test. In *Proceedings of the 15th European Conference on Machine Learning (ECML), Pisa, Italy*, pages 334–346. Springer, 2004.

[127] Cornelius Rosse and José L. V. Mejino, Jr. A reference ontology for biomedical informatics: The Foundational Model of Anatomy. *Journal of Biomedical Informatics*, 36(6):478–500, 2003.

[128] Cornelius Rosse and José L. V. Mejino, Jr. *Anatomy Ontologies for Bioinformatics: Principles and Practice*, volume 6, chapter The Foundational Model of Anatomy Ontology, pages 59–117. Springer, December 2007.

[129] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[130] Daniel L. Rubin, Adam Flanders, Woojin Kim, Khan M. Siddiqui, and Charles E. Kahn. Ontology-assisted analysis of web queries to determine the knowledge radiologists seek. *Journal of Digital Imaging*, 24:160–164, 2011.

[131] Hiroto Saigo, Nicole Krämer, and Koji Tsuda. Partial least squares regression for graph mining. In *Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Las Vegas, NV*, pages 578–586. ACM, 2008.

[132] Hiroto Saigo, Sebastian Nowozin, Tadashi Kadowaki, Taku Kudo, and Koji Tsuda. gBoost: a mathematical programming approach to graph classification and regression. *Machine Learning*, 75(1):69–89, 2009.

[133] Hanan Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, San Francisco, 2006.

[134] Mark Sanderson, Paul Clough, et al. ImageCLEF - image retrieval in CLEF, 2011. `http://www.imageclef.org`.

[135] Simone Santini and Ramesh Jain. Similarity measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):871–883, 1999.

[136] Sascha Seifert, Adrian Barbu, S. Kevin Zhou, David Liu, Johannes Feulner, Martin Huber, Michael Suehling, Alexander Cavallaro, and Dorin Comaniciu. Hierarchical parsing and semantic navigation of full body CT data. In *Proceedings of the SPIE Medical Imaging Conference 2009 (SPIE), Lake Buena Vista, FL, USA*, volume 7259, page 725902, 2009.

[137] Sascha Seifert, Michael Kelm, Manuel Möller, Saikat Mukherjee, Alexander Cavallaro, Martin Huber, and Dorin Comaniciu. Semantic annotation of medical images. In *Proceedings of the SPIE Medical Imaging Conference 2010: Image Processing (SPIE), San Diego, CA, USA*, volume 7628, page 762808, 2010.

[138] Sascha Seifert, Marisa Thoma, Florian Stegmaier, Matthias Hammon, Martin Kramer, Martin Huber, Hans-Peter Kriegel, Alexander Cavallaro, and Dorin Comaniciu. Combined semantic and similarity search in medical image databases. In *Proceedings of the SPIE Medical Imaging Conference 2011: Advanced PACS-based Imaging Informatics and Therapeutic Applications, Lake Buena Vista, FL, USA*, volume 7967, page 796702, 2011.

[139] Nino Shervashidze and Karsten M. Borgwardt. Fast subtree kernels on graphs. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS), Vancouver, Canada*, pages 1660–1668, 2009.

[140] Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. Efficient graphlet kernels for large graph comparison. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, 2009.

[141] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the 9th IEEE International Conference on Computer Vision (ICCV), Nice, France*, pages 1470–1477. IEEE Computer Society, 2003.

[142] Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J. Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J. Mungall, Neocles Leontis, Philippe Rocca-Serra, Alan Ruttenberg, Susanna-Assunta Sansone, Richard H. Scheuermann, Nigam Shah, Patricia L. Whetzel, and Suzanna Lewis. The OBO foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology*, 25(11):1251–1255, November 2007.

[143] Luc Soler, Herve Delingette, Gregoire Malandain, Johan Montagnat, Nicholas Ayache, Christophe Koehl, Olivier Dourthe, Benoit Malassagne, Michelle Smith, Didier Mutter, and Jacques Marescaux. Fully automatic anatomical, pathological, and functional segmentation from CT scans for hepatic surgery. *Computer Aided Surgery*, 6(3):131–142, 2001.

[144] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark, 2011. `http://benchmark.ini.rub.de`.

[145] Florian Stegmaier, Mario Döller, Harald Kosch, Andreas Hutter, and Thomas Riegel. AIR: Architecture for interoperable retrieval on distributed and heterogeneous multimedia repositories. In *Proceedings of the 11th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, pages 1–4, April 2010.

[146] Florian Stegmaier, Mario Döller, Kai Schlegel, Harald Kosch, Sascha Seifert, Martin Kramer, Andreas Hutter, Marisa Thoma, Hans-Peter Kriegel, Matthias Hammon, and Alexander Cavallaro. Generische Datenintegration zur semantischen Diagnoseunterstützung im Projekt THESEUS MEDICO. In *Proceedings of the 41st Jahrestagung der Gesellschaft für Informatik: Datenmanagement und Interoperabilität im Gesundheitswesen, Berlin, Germany*, pages 41–55, 2011.

[147] Florian Stegmaier, Udo Gröbner, Mario Döller, Harald Kosch, and Gero Baese. Evaluation of current RDF database solutions. In *Proceedings of the 10th International Workshop on Semantic Multimedia Database Technologies (SeMuDaTe), 4th International Conference on Semantics And Digital Media Technologies (SAMT)*, pages 39–55, December 2009.

[148] Y. Taieb, O. Eliassaf, M. Freiman, L. Joskowicz, and J. Sosna. An iterative bayesian approach for liver analysis: tumors validation study. In *Proceedings of the 3D Segmentation in the Clinic MICCAI 2008 Workshop: Liver Tumor Segmentation (LTS'08)*, 2008.

[149] Jean Talairach and Pierre Tournoux. *Co-Planar Stereotaxic Atlas of the Human Brain 3-Dimensional Proportional System: An Approach to Cerebral Imaging.* Thieme Medical Publishers, 1988.

[150] Xiaoyuang Tan, Songcan Chen, Zhi-Hua Zhou, and Jun Liu. Learning non-metric partial similarity based on maximal margin criterion. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer*

Vision and Pattern Recognition (CVPR '06), New York, NY, USA, pages 138–145, 2006.

[151] Joshua B. Tenenbaum, Vin Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[152] Marisa Thoma, Hong Cheng, Arthur Gretton, Jiawei Han, Hans-Peter Kriegel, Alex Smola, Le Song, Philip S. Yu, Xifeng Yan, and Karsten Borgwardt. Near-optimal supervised feature selection among frequent subgraphs. In *Proceedings of the 9th SIAM International Conference on Data Mining (SDM), Sparks, NV*, pages 1075–1087, 2009.

[153] Marisa Thoma, Hong Cheng, Arthur Gretton, Jiawei Han, Hans-Peter Kriegel, Alex Smola, Le Song, Philip S. Yu, Xifeng Yan, and Karsten M. Borgwardt. Discriminative frequent subgraph mining with optimality guarantees. *Statistical Analysis and Data Mining*, 3(5):302–318, 2010.

[154] Koji Tsuda. Entire regularization paths for graph data. In *Proceedings of the 24th International Conference on Machine Learning (ICML), Corvalis, OR*, pages 919–926, 2007.

[155] Amos Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.

[156] Natalia Vanetik, Ehud Gudes, and Solomon Eyal Shimony. Computing frequent graph patterns from semistructured data. In *Proceedings of the 2nd IEEE International Conference on Data Mining (ICDM), Maebashi City, Japan*, pages 458–465, 2002.

[157] Peter Vanezis, M. Vanezis, G. McCombe, and Tim Niblett. Facial reconstruction using 3-D computer graphics. *Forensic Science International*, 108(2):81–95, 2000.

[158] Laura J. van't Veer, Hongyue Dai, Marc J. van de Vijver, Yudong D. He, Augustinus A. M. Hart, et al. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:530–536, 2002.

[159] Nikil Wale and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM), Hong Kong, China*, pages 678–689, 2006.

[160] Hongkai Wang, Jing Bai, and Yonghong Zhang. A normalized thoracic coordinate system for atlas mapping in 3D CT images. *Progress in Natural Science*, 18(1):111–117, 2008.

[161] J.Z. Wang, J. Li, and G. Wiederhold. SIMPLIcity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:947–963, 2001.

[162] Luyao Wang, Zhi Zhang, Jingjing Liu, Bo Jiang, Xiyao Duan, Qingguo Xie, Daoyu Hu, and Zhen Li. Classification of hepatic tissues from CT images based on texture features and multiclass support vector machines. In Wen Yu, Haibo He, and Nian Zhang, editors, *Advances in Neural Networks – ISNN 2009*, volume 5552 of *Lecture Notes in Computer Science*, pages 374–381. Springer Berlin / Heidelberg, 2009.

[163] Rober Weber, Hans-J. Schek, and Stephen 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*, pages 194–205, 1998.

[164] Kilian Q. Weinberger, John Blitzer, and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Proceedings of the 19th Annual Conference on Neural Information Processing Systems (NIPS), Vancouver, Canada*. MIT Press, 2006.

[165] Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

[166] Petra Welter, Thomas M. Deserno, Ralph Gülpers, Berthold B. Wein, Christoph Grouls, and Rolf W. Günther. Exemplary design of a DICOM structured report template for CBIR integration into radiological routine. In *Proceedings of the SPIE Medical Imaging Conference 2010: Advanced PACS-based Imaging Informatics and Therapeutic Applications, San Diego, CA, USA*, page 76280B, 2010.

[167] Petra Welter, Jörg Riesmeier, Benedict Fischer, Christoph Grouls, Christiane Kuhl, and Thomas M. Deserno. Bridging the integration gap between imaging and information systems: a uniform data concept for content-based image retrieval in computer-aided diagnosis. *Journal of the American Medical Informatics Association*, 18(4):506–510, 2011.

[168] Pinar Wennerberg, Sonja Zillner, Manuel Möller, Paul Buitelaar, and Michael Sintek. KEMM: A knowledge engineering methodology in the

medical domain. In *Proc. of the 5th International Conference on Formal Ontology in Information Systems (FOIS)*, 2008.

[169] Sebastian Wernicke. A faster algorithm for detecting network motifs. In *Proceedings of the 5th Workshop on Algorithms in Bioinformatics (WABI)*, pages 165–177, 2005.

[170] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[171] World Health Organization (WHO). WHO — international classification of diseases (ICD), 2011. `http://www.who.int/classifications/icd/en/`.

[172] Gudrun Wronski. STOCK4B Bildagentur — sichtbar anders, December 2008. `http://www.stock4b.com`.

[173] Jiajing Xu, Jessica Faruque, Christopher Beaulieu, Daniel Rubin, and Sandy Napel. A comprehensive descriptor of shape: Method and application to content-based retrieval of similar appearing lesions in medical images. *Journal of Digital Imaging*, pages 1–8, 2011. 10.1007/s10278-011-9388-8.

[174] 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.

[175] Zhiyun Xue, Sameer Antani, L. Rodney Long, Jose Jeronimo, and George R. Thoma. Investigating CBIR techniques for cervicographic images. *AMIA Annual Symposium Proceedings*, 2007:826–830, 2007.

[176] Xifeng Yan, Hong Cheng, Jiawei Han, and Philip S. Yu. Mining significant graph patterns by leap search. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD), Vancouver, BC*, pages 433–444, 2008.

[177] Xifeng Yan and Jiawei Han. gSpan: Graph-based substructure pattern mining. In *Proceedings of the 2nd IEEE International Conference on Data Mining (ICDM), Maebashi City, Japan*, pages 721–724, 2002.

[178] Liu Yang. An overview of distance metric learning. Technical report, Department of Computer Science and Engineering, Michigan State University, 2007.

[179] Liu Yang and Rong Jin. Distance metric learning: A comprehensive survey. Technical report, Department of Computer Science and Engineering, Michigan State University, 2006.

[180] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML), Nashville, TN*, pages 412–420. Morgan Kaufmann Publishers Inc., 1997.

[181] Pavel Zezula, Giuseppe Amato, Vlastislav Dohnal, and Michal Batko. *Similarity Search: The Metric Space Approach*. Springer, 1st edition, 11 2005.

[182] Jun Zhang, Nikos Mamoulis, Dimitris Papadias, and Yufei Tao. All-nearest-neighbors queries in spatial databases. In *Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM), Santorini Island, Greece*, pages 297–306, 2004.

[183] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: An efficient data clustering method for very large databases. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD), Montreal, Canada*, pages 103–114, 1996.

[184] Zhi-Hua Zhou and Jun-Ming Xu. On the relation between multi-instance learning and semi-supervised learning. In *Proceedings of the 24th International Conference on Machine Learning (ICML), Corvalis, OR*, pages 1167–1174. ACM, 2007.

[185] Sonja Zillner. Reasoning-based patient classification for enhanced medical image annotation. In Lora Aroyo, Grigoris Antoniou, Eero Hyvönen, Annette ten Teije, Heiner Stuckenschmidt, Liliana Cabral, and Tania Tudorache, editors, *The Semantic Web: Research and Applications*, volume 6088 of *Lecture Notes in Computer Science*, pages 243–257. Springer Berlin / Heidelberg, 2010.

[186] Albrecht Zimmermann and Björn Bringmann. CTC - correlating tree patterns for classification. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM), Houston, TX*, pages 833–836, 2005.