

GEO 511 Master's Thesis

Implementation of a Spatially-Aware Image Search Engine and Its Evaluation using Crowdsourced Relevance Judgements

Oliver F. M. Zihler

Matriculation Number: 06-729-420

Email: oliver.zihler@uzh.ch

Kloten, 29th August 2013

Faculty Representative: Prof. Dr. Ross Purves

Advisers: Prof. Dr. Ross Purves and Dr. Damien Palacio

GIScience: Geocomputation

Department of Geography

University of Zurich

“The true sign of intelligence is not knowledge but imagination.”

Albert Einstein

Acknowledgments

I would like to thank all people involved in my master thesis.

I especially want to thank Prof. Dr. Ross Purves and Dr. Damien Palacio for vivid discussions, valuable support and crucial help during the whole process of creating this thesis.

I would also like to thank my parents for enabling me my studies and supporting me in all my decisions I made during that time.

Furthermore, I thank Tania Homichonok for taking her time to proofread and for moral support whenever I needed it, as well as Stefan Benz and Roy Weiss for technical support and reading.

Lastly, many thanks go to Mathias Lux for support on implementation details in LiRE, Paul Clough for vital feedback on the experimental setup, and Barry Hunter for contributing the images for the evaluation.

Abstract

Concept- and content-based image retrieval can greatly profit from the incorporation of methods developed in geographical information retrieval in estimating the spatial relevance of an image query containing a spatial part. By merging several concepts and methods found in information, image and geographical information retrieval appropriately, this thesis elaborates a prototype of a spatially-aware image search engine written entirely in Java and capable of indexing and retrieving images in three different dimensions (text, space and image content) to improve retrieval effectiveness of queries with spatial content. The prototype's main algorithm uses textual and spatial indexes to retrieve an initial result list by querying both indexes individually, intersecting the resulting score lists and merging the remaining image relevance scores using CombMNZ. Textual image descriptions (e.g. titles) are stored within a Lucene index, whereas a spatial index inside a PostgreSQL/PostGIS database holds spatial coordinates localising the images. Spatial query footprints are provided via the online tools Yahoo! Placemaker and GeoNames. After merging the initial term and spatial result lists, re-ranking using a new approach based on pseudo-relevance feedback of the images' low-level features (colour, texture) together with agglomerative hierarchical clustering is applied to improve the ordering of this combined list. The content index utilizes JDC global features (CEDD and FCTH) implemented in the freely available, content-based image retrieval library LiRE. A secondary refinement algorithm allows a user to query for similar images to retrieve images by individually querying each of the three indexes, or any combination thereof. Evaluation is conducted using traditional measures like P@10, MAP@10 and NDCG@10, whereas relevance judgements needed for each image to assess the systems performance are gathered through the crowdsourcing platform CrowdFlower. Results reveal that both an incorporation of spatial access methods as well as an additional re-ranking using image content and hierarchical clustering can statistically significantly (text-spatial: p-value < 5% significance level, text-spatial content re-ranking: p-value < 1% significance level) improve retrieval performance compared to a textual baseline. However, no statistically significant difference could be observed between text-spatial and text-spatial content re-ranking, although the latter tends to improve retrieval performance as well in terms of P@10, MAP@10 and NDCG@10. Another outcome is that relevance judgements obtained through crowdsourcing can be used as a viable source for the evaluation of a spatially-aware image search engine if certain measures for quality insurance are taken.

Zusammenfassung

Konzept- und inhaltsbasierte Bildersuche kann durch Methoden der geographischen Informationssuche merklich profitieren, wenn es darum geht, die räumliche Relevanz einer räumlich verorteten Bildabfrage abzuschätzen. Die vorliegende Arbeit entwickelt einen Prototyp einer raumbewussten Bildersuchmaschine, basierend auf einer Vereinigung verschiedener Konzepte und Methoden der normalen und geographischen Informations- und Bildersuche. Die Suchmaschine ist komplett in Java geschrieben und fähig, Bilder in drei Dimensionen (Text, Raum und Bildinhalt) zu indexieren und zu suchen. Damit soll die Sucheffizienz für Abfragen mit räumlichem Inhalt verbessert werden. Der Hauptalgorithmus des Prototyps nutzt textbasierte und räumliche Indizes zur Gewinnung einer ersten Resultatliste durch individuelle Abfrage beider Indizes. Die dadurch erhaltenen beiden Resultatlisten werden dann verschnitten und mittels CombMNZ anhand ihrer Relevanzschätzungszahl verbunden. Bildbeschreibungen (z.B. Titel) werden innerhalb eines Lucene Indexes gespeichert, während räumliche Koordinaten des Aufnahmeorts eines Bildes innerhalb einer PostgreSQL/PostGIS Datenbank Platz finden. Die für räumliche Abfragen benötigten geometrischen Repräsentationen werden durch Onlinedienste wie Yahoo! Placemaker und GeoNames zur Verfügung gestellt. Nachdem textuelle und räumliche Resultatlisten verbunden wurden, wird ein neuer Ansatz zur Neuordnung dieser nun verschmolzenen Liste angewendet. Dieser Ansatz basiert auf Pseudorelevanzfeedback, welches wiederum auf Bildmerkmalen (Farbe, Textur) sowie agglomerativem, hierarchischem Clustering aufbaut. Das Ziel ist es, eine bessere Anordnung einer möglicherweise suboptimalen ersten Resultatliste zu erzielen. Der Index für Bildmerkmale speichert globale JDC Merkmale (CEDD und FCTH), welche in der frei verfügbaren, inhaltsbasieren Bildersuchbibliothek LiRE implementiert sind. Ein weiterer, sekundärer Verfeinerungsalgorithmus erlaubt es einem Suchmaschinennutzer ähnliche Bilder zu einem Beispielbild zu suchen, indem alle drei Indizes entweder individuell oder in Kombination mithilfe dieses Bildes abgefragt werden können. Die Evaluierung dieses Prototyps folgt traditionellen Techniken, basierend auf P@10, MAP@10 und NDCG@10. Allerdings wird die Beurteilung der Relevanz eines Bildes zur Prüfung der Systemeffizienz durch die Crowdsourcingplattform CrowdFlower bewerkstelligt. Resultate zeigen eine statistisch signifikant bessere Suchleistung auf, sowohl bei einer Einbindung räumlicher Algorithmen, als auch einer zusätzliche Neuordnung mittels Merkmalen des Bildinhalts zusammen mit hierarchischem Clustering, verglichen mit einer Suchmaschine, die rein textuelle Beschreibungen zur Abfrage verwendet (textuell-räumlich: p-Wert < 5% Signifikanzniveau, textuell-räumlich mit Neuordnung basierend auf dem Bildinhalt: p-Wert < 1% Signifikanzniveau). Allerdings konnte keine statistische Signifikanz zwischen einerseits textuell-räumlich und andererseits textuell-räumlich mit anschließender Neuordnung mittels Bildmerkmalen beobachtet werden, auch wenn letztere im Allgemeinen zu einer besseren Suchleistung tendiert, bezogen auf P@10, MAP@10 und NDCG@10. Des Weiteren konnte gezeigt werden, dass durch crowdsourcing beschaffte Relevanzprüfungen eine

geeignete Quelle zur Evaluation einer raumbewussten Bildersuchmaschine sind, sofern gewisse qualitätssichernde Massnahmen ergriffen werden.

Резюме

Поиск изображений по описанию и содержанию может значительно выиграть от использования методов, разработанных в географическом информационном поиске, при оценке степени соответствия запроса изображения, содержащего пространственную составляющую. Путем слияния некоторых концепций и методов, используемых в географическом информационном поиске и поиске изображений, в данной дипломной работе был разработан прототип пространственно ориентированного механизма поиска изображений, написанный полностью в Java и способный индексировать и извлекать изображения в трех измерениях (текст, пространство и содержание изображения) с целью повышения эффективности запросов с пространственным содержанием. Главный алгоритм данного прототипа использует текстовые и пространственные индексы для извлечения предварительного перечня изображений: производится индивидуальный запрос обоих индексов, полученные перечни изображений пересекаются, а частично совпадающие в результате этого изображения сливаются с помощью CombMNZ. Текстовые описания изображений (напр., наименования) хранятся в Lucene индексе, в то время как пространственный индекс, основанный на PostgreSQL/PostGIS, содержит пространственные координаты изображений. Определение области пространственного запроса осуществляется через онлайн инструменты, такие как Yahoo! Placemaker и GeoNames. После слияния предварительных текстового и пространственного перечней изображений, с целью улучшения систематизации полученного в результате такого комбинированного перечня производится переранжирование с использованием нового подхода, основанного на псевдорелевантной оценке характеристик изображений низшего уровня (цвет, текстура) совместно с иерархической кластеризацией. Контент - индекс использует глобальные функции службы JDC (CEDD и FCTH) с помощью находящейся в свободном доступе LiRE – библиотеки поиска изображений по содержанию. Вторичный уточняющий алгоритм позволяет пользователю запрашивать схожие изображения с целью получения изображений путем индивидуального запроса каждого из трех индексов или любой их комбинации. Оценка производится путем использования традиционных мер, таких как $P@10$, $MAP@10$ и $NDCG@10$, в то время как оценки соответствия, необходимые для каждого изображения, чтобы оценить производительность системы, собираются через краудсорсинговую платформу CrowdFlower. Результаты выявили, что и внедрение методов пространственного доступа, и дополнительное переранжирование с использованием содержания изображения и иерархической кластеризации могут статистически значительно (текстово-пространственный: p -значение $< 5\%$ уровня значимости, текстово-пространственное переранжирование содержания: p -значение $< 1\%$ уровня значимости) повысить показатели поиска по сравнению с текстовым критерием. Тем не менее, между текстово-пространственным и текстово-пространственным

переранжированием содержимого не было выявлено статистически значимой разницы, хотя последнее и имеет тенденцию к улучшению показателей поиска по отношению к $P@10$, $MAP@10$ и $NDCG@10$. Одним из результатов также является то, что оценки соответствия, полученные путем краудсорсинга, могут быть использованы в качестве эффективного источника для оценки пространственно ориентированного механизма поиска изображений, при условии принятия определенных мер гарантии качества.

Contents

Acknowledgments	iii
Abstract	iv
Zusammenfassung	v
Резюме	vii
1 Introduction	1
1.1 Context.....	1
1.2 Scope and Overview	2
1.3 Thesis Structure.....	3
2 State of the Art of IR, GIR and CBIR	4
2.1 Digital Library.....	5
2.1.1 Image and Metadata.....	5
2.1.2 Semantic Analysis of Images.....	6
2.2 Information Extraction Process Flow.....	7
2.2.1 Textual Information Extraction.....	7
2.2.1.1 Tokens and Terms	7
2.2.1.2 Vector Space Model.....	9
2.2.1.3 Okapi BM 25.....	10
2.2.2 Spatial Information Extraction.....	10
2.2.2.1 Spatial Features	11
2.2.3 Image Content Extraction	12
2.2.3.1 Global Features	13
2.2.3.2 Local Features	15
2.2.3.3 Global vs. Local Features: Briefly Compared.....	16
2.3 Indexes.....	16
2.3.1 Indexes for Terms and Image Content.....	16
2.3.1.1 Hash Table.....	17
2.3.2 Indexes for Spatial Features.....	18

2.3.2.1 R-Tree	19
2.3.3 Hybrid approaches.....	20
2.4 Information Retrieval Process Flow	21
2.4.1 Query Formulation and Feature Extraction	21
2.4.2 Matching, Similarity and Relevance.....	22
2.4.2.1 Textual Relevance and Similarity.....	22
2.4.2.2 Geographical Relevance and Similarity	24
2.4.2.2.1 Point-based Relevance Ranking	25
2.4.2.2.3 Content Relevance and Similarity	27
2.4.3 Retrieval	28
2.4.3.1 Result List Fusion.....	29
2.4.3.2 Learning Techniques	32
2.5 Information Visualisation Process Flow.....	36
2.5.1 Popular User Interfaces for Image Retrieval.....	36
2.5.2 User Interfaces in the Context of GIR	37
2.5.3 New Interfaces for Image Retrieval	38
2.6 Assessing a Search Engine's Performance.....	39
2.6.1 User- and System-Centred Evaluations.....	39
2.6.2 Evaluating a System's Ability to Estimate Relevance	41
2.6.2.1 Precision and P@n.....	41
2.6.2.2 AP and MAP	42
2.6.2.3 Normalized Discounted Cumulative Gain	42
2.6.2.4 Student's paired-samples <i>t</i> -test	44
2.6.3 Evaluation Goals.....	45
2.7 Research Gaps and Research Questions	46

3 Design and Implementation.....	49
3.1 Technology and Design Principles.....	49
3.1.1 Development Facilities.....	49
3.1.2 Design Considerations.....	49
3.2 SPAISE Components for Indexing Images and Metadata.....	51
3.2.1 Term Index Implementation.....	52
3.2.2 Spatial Index Implementation.....	52
3.2.3 Content Index Implementation.....	53
3.3 SPAISE Components for Image Search, Retrieval and Presentation	54
3.3.1 Main Retrieval Algorithm.....	54
3.3.1.1 Extracting Features from a Query	54
3.3.1.2 Retrieving Result Lists	58
3.3.1.2.1 Retrieving Result List from Term Index.....	58
3.3.1.2.2 Retrieving Result List from Spatial Index.....	59
3.3.1.3 Combining Result Lists	64
3.3.1.4 Re-Ranking Final Result List with Image Features	65
3.3.1.4.1 Basic Re-Ranking Algorithm	65
3.3.1.4.2 Reducing Noise through Clustering	65
3.3.1.4.3 System Implementation Details of the Re-Ranking Algorithm.....	70
3.3.2 Query-by-Example Refinement Algorithm	70
3.3.2.1 Term Query from Image.....	71
3.3.2.2 Spatial Query from Image.....	72
3.3.2.3 Low-Level Features Query from Image	72
3.3.2.4 Retrieval and Combination in Query by Example.....	73
3.3.3 Facilities for User Interaction with the System.....	74
3.3.4 System Architecture and Interactions	79

4	Evaluation.....	82
4.1	Creating a Test Collection.....	82
4.1.1	Creating a Corpus of Documents.....	82
4.1.2	Creating Topics for Relevance Judgements	83
4.2	Experimental Setup	85
4.2.1	SPAISE Hardware	85
4.2.2	SPAISE configurations	85
4.2.3	Indexed Images and Metadata	87
4.2.4	Image Pool	87
4.2.5	Tasks	87
4.2.6	Platform	88
4.2.7	Participants	88
4.2.8	Experiment Realisation.....	89
4.3	Pre-processing of Raw Relevance Judgements.....	89
4.3.1	Aggregating Relevance Judgements.....	89
4.3.1.1	Central Tendency: Arithmetic Mean, Median and Mode.....	89
4.3.1.2	Pre-Processing Procedure	90
4.3.2	Assessing Trustworthiness of CS Judges.....	90
4.4	System- and User-Centered Evaluations.....	91
4.4.1	Data Quality Assessment.....	92
4.4.2	Performance Assessment.....	94
4.4.2.1	Indexing and Retrieval Performance of the Main Algorithm	94
4.4.2.2	Performance Evaluation of the Systems' Ability to Estimate Relevance	95
4.4.3	Qualitative Evaluations	101
4.4.3.1	Topic-Wise Analysis using P@10	101
4.4.3.2	Analysis of CS RJs comments.....	103

5	Discussion	106
5.1	Research Question 1	106
5.1.1	Performance Analysis	106
5.1.2	Performance of Online Location Retrieval Tools	106
5.1.3	Performance of Spatial Footprints and Algorithms	107
5.2	Research Question 2	108
5.2.1	Performance Analysis	108
5.2.2	Problem Identification	108
5.2.3	Suggestions on Improving TSCR	109
5.3	Research Question 3	110
5.3.1	Dealing with a Crowd	110
5.3.2	Task Design and Data Pre-Processing	111
6	Conclusions	112
6.1	Achievements	112
6.2	Implications	112
6.2.1	Implications on Retrieval Methods	112
6.2.2	Implications on Crowdsourcing Evaluations	113
6.3	Future Work	114
	Bibliography	116
	Appendix A	127
	Appendix B	130
	Appendix C	131
	Appendix D	132
	Appendix E	138
	Appendix F	140
	Appendix G	141
	Appendix H	146
	Appendix I	150
	Appendix J	153

List of Figures

Figure 1: Illustration of the stages needed to build up and search an index.....	4
Figure 2: An image and its metadata.....	5
Figure 3: Creating an index from an image’s title and description.....	8
Figure 4: Different text document retrieval methods.....	9
Figure 5: A two-dimensional vector space.....	9
Figure 6: Extracting image content.....	13
Figure 7: Basic structure of a hash table and assignment of terms to an array cell.....	18
Figure 8: Functionality of an R-Tree.....	19
Figure 9: Visualisation of the intersection distance.....	28
Figure 10: Illustration of CombMNZ.....	30
Figure 11: Activity diagram of the agglomerative hierarchical clustering algorithm.....	34
Figure 12: Illustration of agglomerative hierarchical clustering.....	35
Figure 13: Illustration of single linkage.....	36
Figure 14: Exemplary popular image search engines on the Web.....	37
Figure 15: GUI of SPIRIT.....	37
Figure 16: Another example of a GIR GUI.....	38
Figure 17: Image Search GUI developed during the Tripod project.....	38
Figure 18: Novel ISE interfaces to enhance user experiences.....	39
Figure 19: Example of an AP calculation.....	42
Figure 20: Basic concept and functionalities needed in the intended SPAISE.....	49
Figure 21: Basic overview of the classes involved in the creation of indexes.....	51
Figure 22: Activity diagram of the main algorithm.....	54
Figure 23: Module for looking up and retrieving a geometric spatial footprint.....	55
Figure 24: Core index search facilities for term and spatial queries.....	58
Figure 25: Overview of packages and classes involved in querying the term index.....	59
Figure 26: Overview of spatial similarity functions.....	59
Figure 27: Illustration of the linear near relationship.....	61
Figure 28: Visualisation of the “north of” spatial relationship.....	62
Figure 29: Overview of the classes needed for processing spatial queries.....	63
Figure 30: Classes concerned with fusing possibly various scores into a single score.....	64
Figure 31: Basic steps of the proposed re-ranking algorithm.....	66
Figure 32: Distance matrix and resulting dendrogram.....	67
Figure 33: Recursive collection of EIs for final re-ranking from an initial set of CIs.....	68
Figure 34: Re-ranking without clustering vs. Re-ranking with clustering.....	69
Figure 35: Overview of packages and classes involved in re-ranking.....	70

Figure 36: Secondary algorithm designed to refine an initial search result.....	71
Figure 37: Packages and classes involved in querying the content index.....	73
Figure 38: Screenshot of the GUI after submission of a query.	74
Figure 39: Query input in the form of <theme><spatial relationship><location>.	74
Figure 40: Representation of retrieved images.....	75
Figure 41: Hover functionality on user interaction with image thumbnails.....	75
Figure 42: An image window.....	76
Figure 43: A map with a green query footprint and retrieved images represented as dots.....	77
Figure 44: Interface for search result refinement.	77
Figure 45: Components used to build up the GUI's main parts.....	78
Figure 46: Main classes involved in the MVC architecture.	79
Figure 47: Main functionality synthesised as classes in Controller.....	80
Figure 48: Distribution of images around UK and Ireland.....	82
Figure 49: A topic defined for evaluation of a SPAISE.	84
Figure 50: Percentage of assumed valid retrieved judgements per query/topic.....	92
Figure 51: Calculation times extracted from the main algorithm.....	94
Figure 52: Histograms for P@10 values of the three systems (T, TS and TSCR).....	97
Figure 53: Histograms for AP@10 values of the three systems (T, TS and TSCR).....	98
Figure 54: Histograms for NDCG@10 values of the three systems (T, TS and TSCR).	99

List of Tables

Table 1: The Pansofsky-Shatford facet matrix to systematically describe images.	6
Table 2: Steps needed for pre-processing documents before indexing.....	7
Table 3: NLP steps required to extract spatial features.....	11
Table 4: Different possibilities for encoding spatial information as geometric objects.....	12
Table 5: A selection of global features.....	14
Table 6: A selection of local features.....	15
Table 7: Examples of index data structures and complexities.	17
Table 8: Different possible index structures for spatial indexing.....	18
Table 9: Calculation of cosine similarity values from tf-idf.....	23
Table 10: Comb Fusion algorithms.....	29
Table 11: Learning techniques to enhance retrieval quality.....	32
Table 12: Different methods for calculating the distance between clusters.....	35
Table 13: Different types of retrieval tasks.....	40
Table 14: Example NDCG@10 calculation.	44
Table 15: Performances of different global image descriptors provided by LiRE.....	53
Table 16: Number of each spatial relationship used in the topics.....	84
Table 17: Hardware used to conduct experiments.....	85
Table 18: Evaluation settings of the systems.....	86
Table 19: Various key performance indicators for indexing.....	87
Table 20: Definition of the four-point relevance scale.	88
Table 21: Countries allowed participation in the evaluation.....	88
Table 22: Correlation analysis between median CS RJs and a trusted ranking.....	93
Table 23: Possible mappings from (1, 2, 3, 4) to (0, 1).....	96
Table 24: Statistical values calculated for the different variations of P@10.....	96
Table 25: Statistical values calculated for the different variations of AP@10.....	98
Table 26: Statistical values calculated for the different variations of AP@10.....	99
Table 27: Summary of all mean measures for all the systems.	99
Table 28: Paired-samples <i>t</i> -test applied to each pair of systems.....	100
Table 29: Comparison of topic performance in terms of P@10 for T, TS and TSCR.	101
Table 30: Summary of query types and systems suited best to assess them.....	113

List of Abbreviations

Term	Meaning	Term	Meaning
<i>AM</i>	Arithmetic mean	<i>OOP</i>	Object oriented programming
<i>AP</i>	Average precision	<i>OSM</i>	Open Street Map
<i>CBIR</i>	Content-based image retrieval	<i>P@n</i>	Precision at rank n
<i>CI</i>	Candidate image	<i>POS</i>	Part-of-speech recognition
<i>CS</i>	Crowdsourcing, crowd-source(d)	<i>PRF</i>	Pseudo-relevance feedback
<i>DP</i>	Design pattern	<i>RJ</i>	Relevance judgements
<i>EI</i>	Example image	<i>RQ</i>	Research question
<i>GIR</i>	Geographical information retrieval	<i>SCE</i>	System-centred evaluation
<i>GN</i>	GeoNames	<i>SD</i>	Standard deviation
<i>GPS</i>	Global positioning system	<i>SIFT</i>	Scale invariant feature transform
<i>HCI</i>	Human-computer interaction	<i>SPAISE</i>	Spatially-aware image search engine
<i>IDE</i>	Integrated development environment	<i>SURF</i>	Speeded up robust features
<i>IR</i>	Information retrieval	<i>T</i>	System using only a term index
<i>ISE</i>	Image search engine	<i>TBIR</i>	Context- or text-based image retrieval
<i>JDBC</i>	Java database connectivity	<i>Tf-idf</i>	Term frequency – inverse document frequency
<i>MAP</i>	Mean average precision	<i>TS</i>	System using term and spatial indexes
<i>MDIP</i>	Most desired image pair	<i>TSCR</i>	TS with content-based re-ranking
<i>MVC</i>	Model view controller	<i>UCE</i>	User-centred evaluation
<i>NDCG</i>	Normalized discounted cumulative gain	<i>UIN</i>	User ('s) information needs
<i>NEI</i>	Named entity interpretation	<i>UML</i>	Unified modelling language
<i>NER</i>	Named entity recognition	<i>WGS 84</i>	World Geodetic System 1984
<i>NEV</i>	Named entity validation	<i>XML</i>	eXtensible markup language
<i>NLP</i>	Natural language processing	<i>YPM</i>	Yahoo! Placemaker

1 Introduction

1.1 Context

The advent of the internet and the emerging of the Web 2.0 have led to a vast amount of images being exchanged by people over their personal websites and between friends using Instant Messaging Services, or by sharing them via social networks like Flickr or Facebook. Additionally, web search engines like Google, Yahoo or Bing, make it possible to search the internet for specific images, mostly using text input, but also by providing an example image, and then matching these user inputs to indexed images. However, to date there exists no globally accepted way of how to search for images most effectively. Many different approaches have been introduced to retrieve images, some of which operate on metadata assigned to an image like titles or descriptions (called context- or *text-based image retrieval*, TBIR, Purves et al. 2010), others directly access the colour values and intensities an image is composed of (its low-level features) by extracting colour, shape and texture information (called *content-based image retrieval* CBIR, Enser 2000). The first techniques use text for retrieval, which is a problem researched on for several decades already in the context of digitising text libraries in *information retrieval* (IR). Therefore, much more research interest in the last decade has been in the creation of appropriate methods for CBIR. CBIR can only be conducted if an example image is available. To assess the relevance of an image relatively to a query image, the same low-level features need to be extracted and matched against each other. Traditionally, a certain similarity measure is used in this matching procedure, and the more similar an image is to a query image, the more relevant it is. However, whereas TBIR can assign a certain *meaning* to an image by using words understandable for human beings, no such meaning can be directly extracted from the low-level features an image is composed of. Therefore, also assessing the *relevance* of an image compared to another image is limited to abstract features no human is able to actually understand. Consider e.g. the riots that recently took place in many Arabic countries. People may want to find images of these riots. Low-level features may retrieve images that have a crowd of people shown in it, but this crowd may at the same time be cheering for a president, a rock band at a concert, or by any other gathering. This circumstance is described and termed by Smeulders et al. (2000) as *Semantic Gap*:

"The lack of coincidence between the information one can extract from the visual data and the interpretation that the same data has for a user in a given situation."

Thus, CBIR, which only operates on the low-level features of an image, has only limited practical value for some few specialised applications (Enser et al. 2007). The retrieved images need to match the user's mental image of what he or she is looking for. Else, the retrieved images may only have little or no value to the user. Above mentioned example of crowds reveals that queries may contain explicit *locational* information. A rioting crowd may be extracted, but if someone is looking for riots in Egypt, riots in Tunisia or Yemen may also look very similar and may therefore be considered relevant by a

CBIR system. Although this problem could be solved through the use of title and descriptions assigned to an image, such allocations may not be available. Moreover, such an assignment needs to be conducted explicitly by a human being. However, nowadays mobile phones and cameras often automatically assign GPS coordinates to an image when it is taken. Such coordinates are able to more or less unambiguously identify the location of where the image was taken within some meters. Therefore, they provide the means to efficiently distinguish an image of riots in Egypt from an image of riots in Tunisia. The research field concerned with spatially relevant information is *geographical information retrieval* (GIR). GIR is an extension of the field of information retrieval (IR) (Baeza-Yates and Ribeiro 1999). The intention is to improve the quality of retrieved information and the access to (unstructured) documents found on the internet (Jones and Purves 2008). However, to date, no complete integration of GIR into the TBIR/CBIR field has been undertaken, although focusing on geographically relevant images certainly makes sense. Different papers suggest around 13% to 23% (summarised in Palacio et al. 2011) of queries submitted to traditional search engines contain spatially relevant information. The incorporation of an additional spatial dimension can therefore provide tools to increase the meaning of an image and therefore help minimise the semantic gap occurring with CBIR in the context of spatial queries. The question is: *how* can we bring these different research fields together to support each other in the context of retrieving images with both thematic *and* spatial relevance and to minimise the impact of the semantic gap when using raw CBIR techniques?

1.2 Scope and Overview

This thesis follows an interdisciplinary approach to merge different findings of TBIR, CBIR and GIR particularly for the retrieval of thematically and spatially meaningful images. To assess the effectiveness of such an approach, new ways of evaluation need to be derived. Therefore, besides implementing a fully functional prototype of a *spatially-aware image search engine* (SPAISE), it will be investigated how to cost effectively and easily collect *relevance judgements* (RJs) through *crowdsourcing* (CS) platforms like CrowdFlower. These RJs will then be used to evaluate the implemented prototype's retrieval performance by applying prominent performance measures of IR. The aim of this thesis is not to evaluate possible pre-processing steps used in IR and GIR to retrieve spatial information from texts, but to directly focus on the implementation of methods from various research fields to make appropriate use of the extracted features in form of GPS coordinates. Furthermore, no new CBIR methods shall be introduced, but use will be made of existing frameworks and software libraries. The idea is to develop an effective way of *merging* established approaches. The overarching research question for this work therefore can be summarised to:

Overarching Research Question

How can methods from GIR and CBIR efficiently be combined for the purpose of retrieving spatially relevant images and also effectively favouring thematically highly relevant images while discarding images with minor relevance to the submitted query; and how can this performance be assessed?

1.3 Thesis Structure

In chapter State of the Art of IR, GIR and CBIR, a comprehensive literature review of TBIR, CBIR and GIR shall be given to extract research gaps and formulate *research questions* intended to answer the overarching research question. This section also provides tools needed to build a SPAISE. Its implementation will then be detailed in Design and Implementation. The implementation is followed by chapter Evaluation, where additional theory not presented in the state of the art is introduced, which is important for understanding the system's evaluation. Afterwards, Discussion intends to reveal the effectiveness of the proposed approaches for merging the three fields of TBIR, CBIR and GIR through a thorough discussion of the research questions, and Conclusions shall provide new recommendations and suggestions on implementing and evaluating a SPAISE together with an outlook of what could be researched on in the future as a result of the findings here.

All UML (*Unified Modelling Language*) class and activity diagrams are created using *Microsoft Visio 2013*. All other illustrations are designed using *Adobe Photoshop CS 5.1*, *Microsoft PowerPoint 2010* or *Microsoft Visio 2013*.

2 State of the Art of IR, GIR and CBIR

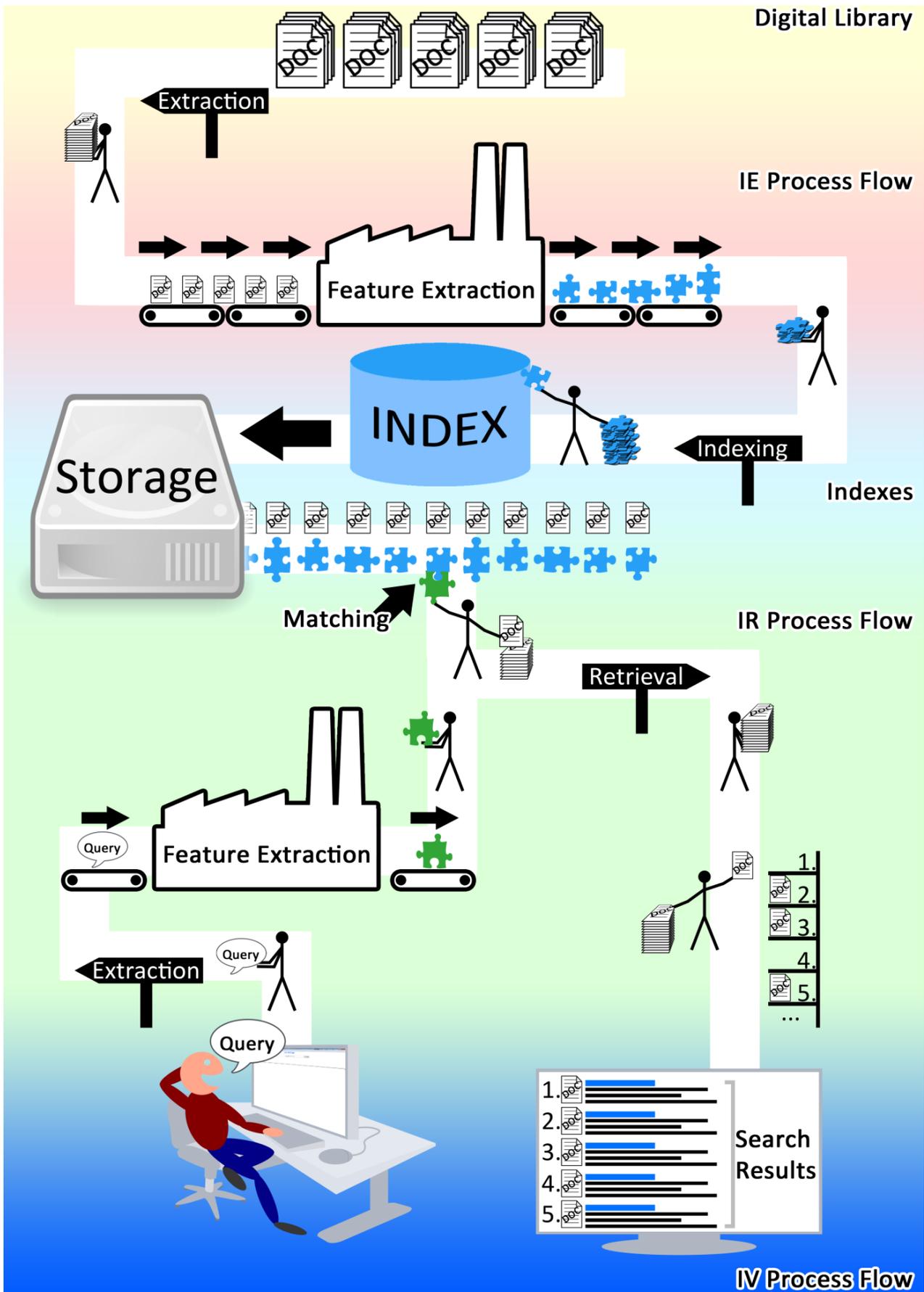


Figure 1: Illustration of the stages needed to build up and search an index.

This chapter gives an in-depth overview of the state-of-the-art concepts related to CBIR and GIR, which are both central for the system to be created. The overview is broken up into the 5 stages depicted in Figure 1: *Digital Library, Information Extraction (IE) Process Flow, Indexes, Information Retrieval (IR) Process Flow, and Information Visualisation (IV) Process Flow* (Palacio et al. 2011). Figure 1 is inspired by Yuen-C and Shin (2009). The description follows a hierarchical top-down approach, where the detail of description increases with each subsection until a certain level of elaboration needed to understand the vocabulary, procedures and methods used in the implementation chapter is achieved. The interested reader may refer to the provided literature for further details.

2.1 Digital Library

Any search endeavour begins with collecting a set of unstructured documents. Documents can, for example, be texts and articles on a website, as is the case with many GIR systems (e.g. Purves et al. 2007), or images, which can also be found on those websites. Naturally, websites are not the only way to obtain pictures, but the fact that many internet-based companies provide facilities for uploading images makes the internet the prime source nowadays. GIR and CBIR consequently mainly focus on web-based search strategies (see e.g. Jones and Purves 2008 for a GIR or Arampatzis et al. 2013 for a CBIR example). Although small image collections containing some few hundreds of items can be searched through fairly quickly, more elaborated methods are needed in the case of thousands and millions of images. Before structuring such collections appropriately, a brief characterisation of images and their associated metadata shall be given.

2.1.1 Image and Metadata

From a technical point of view, an image is simply a data structure holding physical attributes (also called *primitive or low-level features*, Eakings and Graham 1999). The main primitive features are colour, texture, shape and their spatial and spatial-temporal distribution (see Figure 2, Enser 2000). Primitive features themselves do not have any inherent *meaning*. They are nothing more than bits representing an image. What assigns meaning to a picture is a human's *interpretation* of how these features are assembled and in which context they were created or retrieved (e.g. an artwork, a CCTV surveillance image, a holiday image etc.). Meaning therefore is neither a well-defined nor an objectively quantifiable attribute like colour or spatial distribution of shapes in the image (Enser 2000), but a property assigned by humans through combining objective and subjective knowledge in a socio-cognitive process (Heiddorn 1999). Because such a meaning cannot be derived from the primitive features themselves, it is a well-known technique to manually (and also increasingly automatically) assign title, descriptions, keywords/tags or other *textual* descriptions, which then textually represent

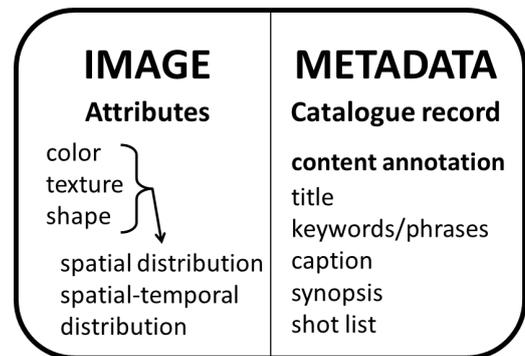


Figure 2: An image and its metadata.

an image (Enser 2000). A prominent example in the case of assigning geographically relevant tags is the *Tripod project* (An overview of published papers can be found on tripod.shef.ac.uk/publications). The question remains, how to describe images properly and in a systematic way so that it is utilisable for retrieval?

2.1.2 Semantic Analysis of Images

Some form of pattern or template, with which an image can be described formally, is the so-called *Panofsky-Shatford facet matrix*, introduced by Shatford (1986) as a generalisation of Panofsky (1982, c1955)'s work. The matrix can be found in Table 1.

Facets	Specific of	Generic of	About
Who? Animate and inanimate; concrete objects and beings	Individually named persons, animals, things, ...	Kinds of persons, animals, things	Mythical beings (generic/specific); Abstractions manifested or symbolized by objects or being
What? What are the objects beings doing? (Actions, events, emotions)	Individually named events	Actions, conditions	Emotions, Abstractions manifested by actions, events
Where? Locale, site place; geographic, cosmographic, architectural	Individually named geographic location	Kind of place, geographic or architectural	Places symbolised (generic/specific); Abstractions manifested by locale
When? Time; linear or cyclical	Linear time; dates or periods	Cyclical time; seasons, time of the day	Emotions or abstractions, symbolised by or manifested by time

Table 1: The Panofsky-Shatford facet matrix to systematically describe images.

This matrix is not an actual classification scheme, but can be used to identify and classify the kinds of subjects a picture contains (Shatford 1986). It is therefore well-suited for the task of assigning meaning to an image systematically. Classification of subjects of a picture is divided into four facets: *who*, *what*, *where*, and *when*. Each of these basic facets can then be subdivided into smaller aspects: *specific of*, *generic of*, and *about*. By analysing an image with this template, the danger of overlooking certain aspects is reduced. This matrix has been extensively used to classify image queries (e.g. Armitage and Enser 1997). However, the “*where*” facet has only recently (Purves et al. 2010) been systematically evaluated with the aim of improving TBIR and tackling the *semantic gap* from a geographic point of view. This description resulted in the proposal of a concept ontology to describe the where/specific cell of the matrix, hierarchically splitting up a photograph into scene types together with their relationships, qualities, elements and related activities.

A digital library may be analysed and described thoroughly and systematically using this matrix and retrieving images would thus solely rely on text. The problem remains that most data on the web is unstructured and that textual assignments may only be sparse, noisy, and inconsistent. Even in the case of a systematic assignment of meaningful words to each image, users would still need to *know* the actual set of acceptable keywords to query a system only based on textual annotations (Jones and

Purves 2008). Another way to describe images, however, is to analyse its *low-level features*. The research field concerned with extracting, indexing and retrieving low-level features from images is that of *content based image retrieval* (CBIR). Additionally, many queries are geographically relevant (Palacio et al. 2011). Therefore, an important task is to set images and queries into a geographical context.

2.2 Information Extraction Process Flow

Different aspects or *features* of images can be regarded for extraction and indexing, which can later on be used for retrieval. The next part explains the state of the art for extracting features from images, focusing on *how* and *what* can be extracted in the context of TBIR, CBIR and GIR. This corresponds to the second stage of Figure 1: *information extraction (IE) process flow*.

2.2.1 Textual Information Extraction

The first feature possible to extract from an image is the *text* assigned to it. Images may show a title, descriptions and tags describing what is visible in the image, where it was taken, what event it was (e.g. a wedding, birthday, etc.) and other information, sometimes not even connected to the image (e.g. an ironic statement about what can be seen in the image). Although these few examples already show how vital it is to analyse the content of texts, this thesis ignores the theoretical part of such analyses to most extents and directly focuses on the practical part of extracting information from texts for storing and retrieval purposes.

2.2.1.1 Tokens and Terms

Step	Description	Example	Consequences
1) Collecting and Parsing Text	<ul style="list-style-type: none"> - Document in digital form. - For images: title and descriptions, locations. 	<p><i>Title: "Blue house"</i></p> <p><i>Description: "A blue house built in the year 2002. It isn't the nicest building in the city, but its blue colour is unique in Kansas, U.S.A."</i></p>	-
2) Tokenisation	<ul style="list-style-type: none"> - Splitting up text into words (<i>tokens</i>). - Removing unwanted characters, e.g. punctuations. 	<p>"Blue", "House", "A", "blue", "house", "built", "in", "the", "year", "2002", "it", "isn't", "the", "nicest", "building", "in", "the", "city", "but", "its", "blue", "colour", "is", "unique", "in", "Kansas", "U.S.A."</p>	-
3) Linguistic Pre-Processing	<ul style="list-style-type: none"> - Normalising tokens (e.g. "aren't", "are not" to "arent"). 	<p>"Blue", "House", "A", "blue", "house", "built", "in", "the", "year", "2002", "it", "isnt", "the", "nicest", "building", "in", "the", "city", "but", "its", "blue", "colour", "is", "unique", "in", "Kansas", "U.S.A."</p>	<ul style="list-style-type: none"> - Needs to be conducted in the same way for both indexing and query processing.
4) Stop word removal	<ul style="list-style-type: none"> - Removing words with little meaning (<i>stop words</i>, e.g. "a/an", "for", "by", etc.). 	<p>"Blue", "House", "blue", "house", "built", "year", "2002", "isn't", "nicest", "building", "city", "its", "blue", "colour", "unique", "Kansas", "U.S.A."</p>	<ul style="list-style-type: none"> - Decreases storage costs. - Removes meaning (e.g. "Pub in York" becomes "Pub", "York").
5) Normalisation	<ul style="list-style-type: none"> - Removing superficial differences (e.g. U.S.A, U-S-A and USA → usa). - Optional: lowercase conversion of tokens. 	<p>"blue", "house", "blue", "house", "built", "year", "2002", "isnt", "nicest", "building", "city", "its", "blue", "colour", "unique", "kansas", "usa"</p>	<ul style="list-style-type: none"> - Needs to be conducted in the same way for both indexing and query processing. - Lower case conversion may alter a word's type.
6) Stemming	<ul style="list-style-type: none"> - Reducing words to their <i>stem</i>. - See e.g. Manning et al. (2008) for an overview of stemmers. - Related: lemmatisation (see e.g. Palacio et al. 2010) 	<p>"blue", "hous", "blue", "hous", "built", "year", "2002", "isnt", "nicest", "build", "citi", "it", "blue", "colour", "uniqu", "cansa", "usa"</p>	<ul style="list-style-type: none"> - Stemming may increase the number of retrieved documents, but decrease the number of relevant retrieved documents.

Table 2: Steps needed for pre-processing documents before indexing.

In information retrieval (IR), text based documents are of primary interest and many methods have been proposed for effective document retrieval. These techniques can also be used to textually retrieve images. Table 2 demonstrates common first steps to analyse texts found in TBIR (e.g. Enser 2000) and GIR (e.g. Purves et al. 2007, Palacio et al. 2011, Brisaboa et al. 2010). The steps are, if not otherwise specified, based on Manning et al. (2008) and an illustrating example is provided in Table 2 for each step. To understand why sentences are not stored as a whole in a text, the notion of an *index* needs to be introduced, which will be explained in further detail in chapter 2.3 Indexes. Words tokenised from a sentence are stored in such an index. An index in its simplest form can be regarded as a dictionary at the end of a textbook, which shows for each important term the pages of the book where this term occurs. Figure 3 visualises the steps from Table 2 and additionally depicts storage of the words in an index.

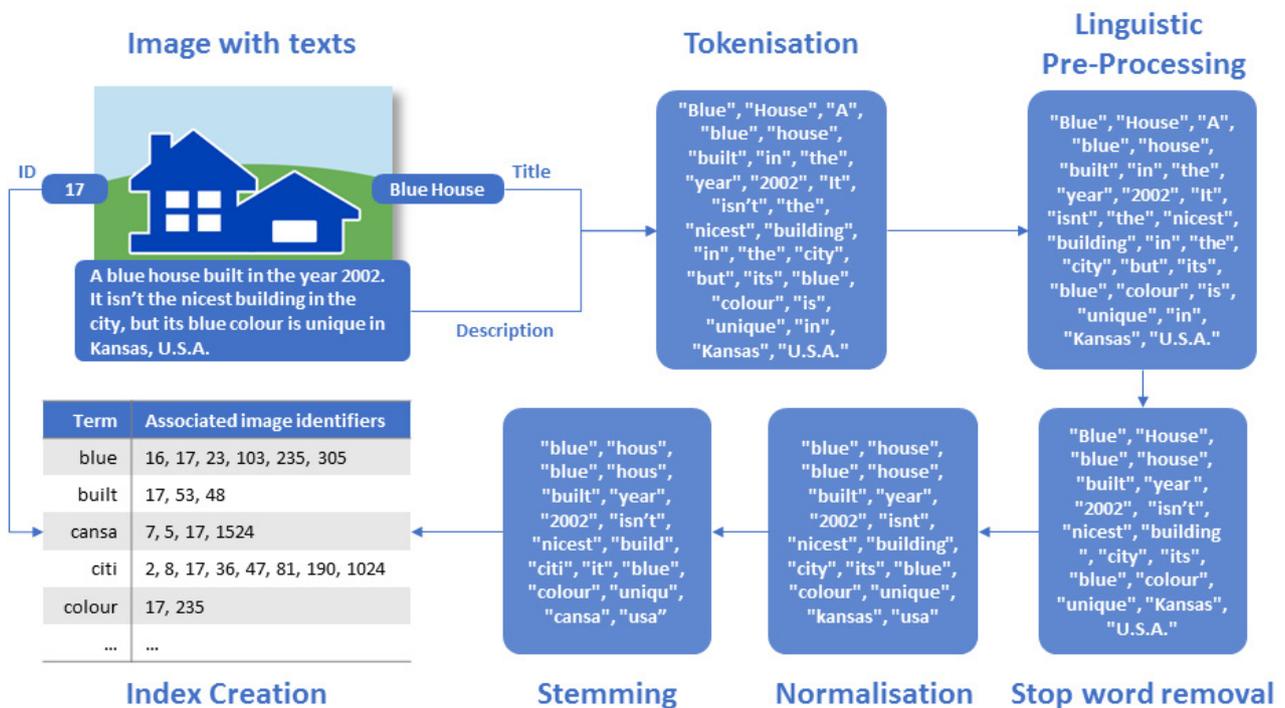


Figure 3: Creating an index from an image's title and description.

Image identifiers can be regarded as pages in a dictionary at the end of a book, wherein the term occurs. Besides stemming, there is also a technique called "lemmatisation". It is similar to stemming but uses more knowledge about a word (e.g. the lemma of the word "forgotten" would be "forget", the "base" word of "forgotten". See e.g. Palacio et al. 2012).

However, terms alone may not be enough information for an effective retrieval of relevant documents. Thus, it is vital to also assess the *relative importance* of terms and documents to the whole document collection. Many different approaches for textual information retrieval exist, as Figure 4 shows. The models are categorised into *mathematical basis* and *properties of the model*. Kuroпка (2004) gives a thorough examination of each model. Only the *Vector Space Model* (VSM) is looked into in greater detail, because it is a widely used model in IR.

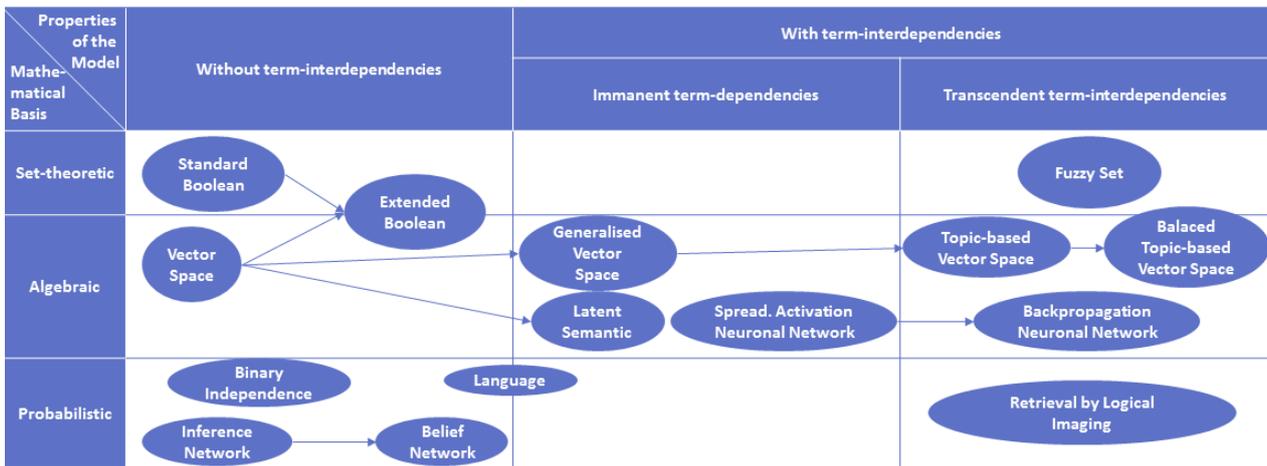


Figure 4: Different text document retrieval methods.

Adapted from Kuroпка (20.07.2012).

2.2.1.2 Vector Space Model

The following descriptions are based on Manning et al. (2008). In VSM, a set of documents is represented as vectors in a vector space. It is an algebraic model (Figure 4). The vector \vec{V} derived from a document d can be denoted as $\vec{V}(d)$. Each term in this vector is represented by a component w_t . Different ways of computing these components w_t exist. An often used method is *tf-idf weighting*. A set of documents of a collection then becomes a set of vectors in a vector space, where an axis is assigned to every term/dimension (see Figure 5). For example, if a document consists only of the two terms “gossip” and “jealous”, the corresponding two-dimensional vector would have two components:

$$\vec{V}(d) = \begin{pmatrix} w_{gossip} \\ w_{jealous} \end{pmatrix}$$

VSM therefore supports the so-called *bag-of-words model*, where the exact ordering of the terms in a document is ignored. Thus, the “rabbit is faster than the snail” is considered identical to “the snail is faster than the rabbit”. Although semantically not always correct, it is still intuitive to assume that two documents with a similar bag-of-words representation are also similarly relevant to an input query. This is especially true in comparison to a vector of a document without any term in common with a vector of another document.

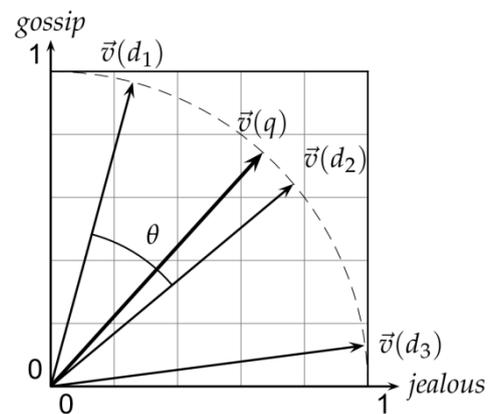


Figure 5: A two-dimensional vector space.

Term frequency and weighting: tf-idf. Each term of the documents in the collection needs to be assigned a weight representing its importance to a particular document if it is not to be used in a simple Boolean true/false query. A commonly used weighting method is *tf-idf*, see Formula II. Tf-idf assigns a high weight to a term if it occurs frequently within a particular document but only rarely in the whole collection of documents. Two components are needed:

- 1) *Term frequency* (tf) describes how many times a term occurs in a document. A document gets a higher score if a term occurs more often in the same document. This corresponds to the intuitive understanding that a document is more important than another if a term occurs more often in that document.
- 2) However, terms that occur in many documents of the collection should be lessened in importance, because the high frequency of that term may be systematic (e.g. “auto” in a collection of documents about the auto industry). To scale the importance of a term that occurs frequently in many documents, the *inverse document frequency* (idf) of this term is calculated. For the calculation of idf, *document frequency* (df) of a term is needed, which represents the number of documents in the collection containing this term. *Idf* is calculated using the logarithm of the number of documents in the collection (N) divided by df (see Formula I). tf-idf therefore assigns a weight to term t in document d that is

- 1) Highest, when a term occurs many times within a small number of documents.
- 2) Lower, when the term occurs fewer times in a document or occurs in many documents.
- 3) Lowest, when the term occurs in almost all documents.

I

$$idf_t = \log \frac{N}{df_t}$$

II

$$tf-idf_{t,d} = tf_{t,d} \cdot idf_t$$

Using such a weighting procedure, each term contributes differently to the retrieval. Therefore, besides extracting terms from textual documents, such parameters may also be directly calculated and stored in association to terms. It will be shown in 2.4.2.1 Textual Relevance and Similarity how these components are used for the actual retrieval.

2.2.1.3 Okapi BM 25

Besides VSM, which is part of the algebraic models, the so-called okapi BM 25 model is often encountered in the literature (Robertson 1995). Classified as a probabilistic model (Figure 4), it is a bag-of-words retrieval function (or more of a function family) like VSM, which also uses weighting based on tf-idf to calculate a relevance score of a document. An advantage of BM 25 is that it takes into account the length of the documents. Detailed descriptions on okapi BM 25 can be found in e.g. Manning et al. (2008). In the context of GIR, BM 25 is used, for example, in the textual index of SPIRIT (Purves et al. 2007).

2.2.2 Spatial Information Extraction

Before spatial information can be extracted from texts, they need to be appropriately processed. GIR systems thus scan texts for occurrences of geographic content and convert them to geographic

features. These features can then be assessed in the context of geographic relevance, similar to the weightings extracted for terms. The process of finding and extracting locations from texts and assigning coordinates to these places is called *geocoding*. It can only be accomplished through *geo-parsing*, where *Natural Language Processing* (NLP) is used to identify geographic references in texts (Clough et al. 04.01.2004). A location that is found, disambiguated and assigned a coordinate (e.g. a GPS coordinate with latitude and longitude in degrees) is then *geo-referenced*. It knows its geographic location (Hill 2006), which can then be exploited in further processing steps. NLP platforms like OpenNLP, but also LingPipe, MetaCarta and OpenCalais all provide possibilities for spatial named entity recognition and extraction from textual documents. Palacio et al. (2011) and Brisaboa et al. (2010) give overviews of what NLP can do for GIR, summarised in Table 3.

Term	Description
Part-of-Speech (POS) tagging and Named Entity Recognition (NER)	Two ways of linguistic analysis: <ul style="list-style-type: none"> - POS: Process of sequentially labelling tokens with syntactic labels (e.g. nouns, verbs, etc.). - NER: process of finding mentions of predefined categories (e.g. names of locations).
Named Entity Validation (NEV)	<ul style="list-style-type: none"> - Knowledge-based resources validate candidate named entities (e.g. spatial features). - Real candidates are distinguished from false ones and <i>geo-referenced</i> (e.g. by assigning a coordinate). - Requires <i>Gazetteer</i> lookup (geographical dictionary containing location names, alternative location names, population, coordinates, etc.).
Named Entity Interpretation (NEI)	<ul style="list-style-type: none"> - Last step: find relations between tokens and collect meaningful token groups. - Uses knowledge-based resources for disambiguation and association of representations to spatial features and to analyse spatial relationships.

Table 3: NLP steps required to extract spatial features.

Online tools exist that encapsulate the whole geo-parsing and -coding process, e.g. *Yahoo! Placemaker* (YPM, now incorporated as PlaceFinder and PlaceSpotter in Yahoo! BOSS Geo Services, see developer.yahoo.com/boss/geo) and *GeoNames* (GN, geonames.org). These platforms allow querying for place names and receiving a set of information about the locations as an XML document, e.g. coordinates of the location as well as the country it is located in, etc.

2.2.2.1 Spatial Features

After extraction and disambiguation of place names, they can be assigned a geometrical representation. Such representations have advantages over place names, because they are unambiguous and persistent. Another name for these spatial features is *spatial* or *geographic footprints* (Frontiera et al. 2008). They can be encoded in varying levels of detail. Table 4 illustrates some of the most common geometrical representations used in GIR systems. YPM can retrieve an MBR as well as centroid (point) coordinates according to a location name, whereas GN can only provide centroid representations. Both systems additionally return information about the country, city or district the queried location is situated. GN also offers population information about a location. GN is

used e.g. in Brisaboa et al. (2010) for the purpose of extracting spatial features from text. YPM’s MBR retrieval capabilities provide e.g. in Martins and Calado (2010) a spatial footprint for relevance assessment. However, no research has evaluated the performance of these services in the context of a SPAISE. Chapter 2.4.2.2 Geographical Relevance and Similarity gives insights into relevance estimations with geometrical representations for querying.

Term	Visualisation	Description
Point		<ul style="list-style-type: none"> - Simplest representation of a location, e.g. the centre point of a country. - Disregards possible area associated with a spatial entity (points are infinitesimal).
MBB/MBR		<ul style="list-style-type: none"> - Minimum Bounding Box/Rectangle. - Approximates area by a rectangular box, minimally and completely surrounding the spatial entity. - Often adds area not part of the actual shape of the spatial entity. - Other MB approximations: MB Ellipses, MB N-corner convex polygons (according to Cai 2011).
Convex Hull		<ul style="list-style-type: none"> - A convex hull of a set X of points is the smallest convex set that contains this set X (Berg et al. 2008). - “Rubber band stretched around a set of points”. - Often adds additional area, but not necessarily as much as an MBR. - Estimates a spatial feature more accurate than an MBR.
Polygon		<ul style="list-style-type: none"> - Highest level of detail possible. - Can approximate actual area of the spatial feature. - Higher storage costs than MBR or convex hull.

Table 4: Different possibilities for encoding spatial information as geometric objects.
Summarised from Frontiera et al. (2008) and Cai (2011).

2.2.3 Image Content Extraction

Besides extracting textual information from descriptions *assigned* to images, features of an image’s *content* itself can be extracted. Figure 6 shows a typical image content/low-level features extraction processing chain. In CBIR, the idea is to extract *low-level features* directly from an image. As mentioned before, these low-level features cover *colour*, *texture* and *shape*, as well as their *spatial* and *spatial-temporal* distribution (e.g. Enser 2000, Liew and Law 2008, Datta et al. 2008). They can be extracted directly from the pixels an image is made of. First, each image of an image collection is run through an algorithm that extracts low-level features. Then, similar to the aforementioned textual information extraction, an index is created and all the extracted low-level features as well as an identifier referencing the actual image these low-level features belong to are stored within the index. A low-level feature is intended to capture certain visual properties of an image, either globally for the entire image or locally for a small group of pixels (Datta et al. 2008, Deselaers et al. 2008). Therefore, a distinction can be drawn between *global features* (describing the whole image) and *local features* (describing

many small parts of an image). For additional reading and overviews, valuable information can be found in e.g. Smeulders et al. (2000), Datta et al. (2008) or Deselaers et al. (2008).

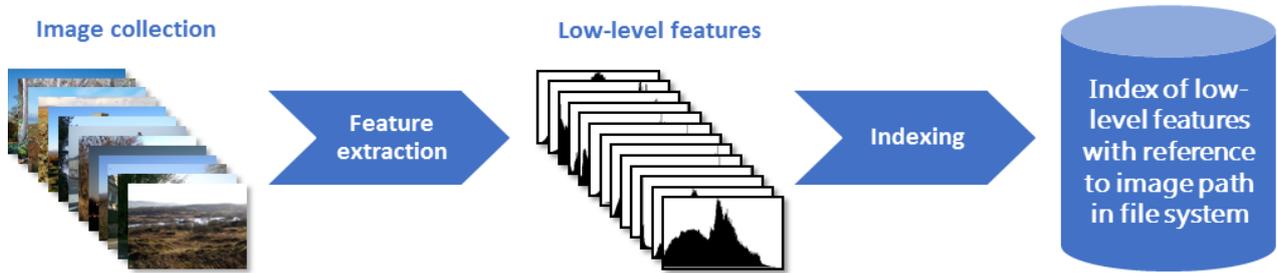


Figure 6: Extracting image content.

Adapted and altered according to Iqbal et al. (2012).

2.2.3.1 Global Features

Global features describe colour, texture and shape globally for the whole image. For example, an image may be segmented into sub-images. Then, for each sub image, the average colour components (e.g. red, green and blue) can be computed. Therefore, the overall image is represented by one vector of colour components, where one dimension of the vector corresponds to a certain sub image location. The advantage of global features is their rather low computational expenses. However, this low complexity comes at the cost of less discriminative power compared to e.g. local features (Datta et al. 2008). Various global features, each of which may be better for one task or another, can be found in the literature. Table 5 only shows a selection to give an insight into the possibilities.

A brief note on MPEG-7 features (see Table 5): Several visual descriptors form the MPEG-7 (*Moving Picture Experts Group*) standard. MPEG focuses on standardising computationally inexpensive and comparable features that also efficiently use the available memory. Overviews of features can be found in e.g. Sikora (2001), Manjunath et al. (2001), or Skarbek (2001).

Feature name	Description
1) Tamura texture features (Tamura et al. 1978)	<ul style="list-style-type: none"> - Three features <i>coarseness</i>, <i>contrast</i> and <i>directionality</i>, important to human perception. - <i>Coarseness</i>: most fundamental feature often referred to as <i>texture</i>. - <i>Contrast</i>: stretching or shrinking of the grey scale of an image. - <i>Directionality</i>: global property over a given region.
2) Colour Histograms CH (e.g. Swain and Ballard 1991)	<ul style="list-style-type: none"> - Obtained by discretising colours of an image into a discrete colour space (e.g. <i>red</i>, <i>green</i> and <i>blue</i> (RGB) or <i>hue</i>, <i>saturation</i> and <i>value</i> (HSV)). - Counting the number of times each discrete colour occurs in the image. - Invariant to translations and rotations around the viewing axis and to changes in scale or occlusion. - Suitable for representing three-dimensional objects with only a small number of histograms.
3) Auto Colour Correlogram AAC (Huang et al. 1997)	<ul style="list-style-type: none"> - Colour correlograms express how the spatial correlation of pairs of colours changes with distance. - The auto-correlogram of an image captures the spatial correlation between identical colours (\neq colour histogram: captures only colour) \rightarrow effective in discriminating images, eliminates major drawbacks of classical colour histograms. - Efficient computation.
4) Scalable Colour Descriptor SCD, MPEG-7 (e.g. Sikora 2001)	<ul style="list-style-type: none"> - Describes colour distribution over the whole image. - Uses a colour histogram in a uniformly quantised HSV colour space with 255 bins. - Rough (16 bits) to high-quality (1000 bits) histogram representations possible.
5) Colour Layout Descriptor CLD, MPEG-7 (e.g. Sikora 2001)	<ul style="list-style-type: none"> - Describes spatial distribution of colour in an arbitrarily-shaped region. - Local and global spatial colour distribution describable. - Suitable for high-speed retrieval and browsing. - Much more compact than a colour histogram approach. - Clusters colours of regions into small numbers of representative colours and values. - Suitable for sketch-based retrieval, content filtering and visualisation.
6) Edge Histogram EH, MPEG-7 (e.g. Sikora 2001)	<ul style="list-style-type: none"> - Non-homogenous, compact, scale invariant texture descriptor, 240 bits size. - Captures spatial distribution of edges (similar to CLD). - Supports both rotation-sensitive and -invariant matching. - Extraction through image division into 16 non-overlapping, equally sized blocks. - Edge information is calculated in five categories for each block (vertical, horizontal, 45°, 135°, and non-directional edge). - Each image block is represented by a five bin histogram.
7) Colour and Edge Directivity Descriptor CEDD (Chatzichristofis 2008 #141)	<ul style="list-style-type: none"> - Combines colour and texture information in a 432 bit histogram. - Splits an image into a number of blocks. - HSV histogram generated through fuzzy-linking \rightarrow rule based three-input fuzzy system finally generates a 24 bin quantised histogram of colour information. - MPEG-7 Edge Histogram texture information classifies each block into one or more of 6 texture classes \rightarrow 144 bins histogram. - Gustafson-Kessel (Gustafson and Kessel 1978) fuzzy classifier quantifies the 144 CEDD factor values to the interval of [0, 7], limiting the length of the descriptor to 432 bits. - See also FCTH.
8) Fuzzy Colour and Texture Histogram, FCTH (Chatzichristofis and Boutalis 2008b)	<ul style="list-style-type: none"> - Combines colour and texture information in a 576 bit histogram. - Colour information extraction as in CEDD. - Texture information extraction: Each image block is transformed with Haar Wavelet (Haar 1910) and a set of texture elements are exported \rightarrow inputs into third fuzzy system \rightarrow converts 24 bins histogram to a 192 bins histogram. - Gustafson-Kessel fuzzy classifier quantifies the 192 FCTH factor values to the interval of [0, 7], limiting the length of the descriptor to 576 bits. - See also CEDD.
9) Joint Composite Descriptor JCD (e.g. Chatzichristofis and Arampatzis 2010)	<ul style="list-style-type: none"> - Joins CEDD and FCTH \rightarrow Combines colour information and texture areas of both \rightarrow suitable for natural colour images, more effective than MPEG-7 descriptors. - Consist of 7 texture areas (each area consisting of 24 colour areas).

Table 5: A selection of global features.

2.2.3.2 Local Features

Feature name	Description
1) Scale Invariant Feature Transform (SIFT, Lowe 1999)	<ul style="list-style-type: none"> - Invariant to scaling and rotation. - Partially invariant to change in illumination and 3D camera viewpoint. - Localised in spatial and frequency domains → reduces probability of disruption by occlusion, clutter, noise. - Highly distinctive → high matching probability of a feature in large databases. - Computation involves: <ol style="list-style-type: none"> (1) Scale-space extrema detection. (2) Key point localisation. (3) Orientation assignment. (4) Key point descriptor. - Feature computation covers entire image → object recognition possible.
2) Gradient Location and Orientation Histogram (GLOH, Mikolajczyk and Schmid 2005)	<ul style="list-style-type: none"> - Extension of SIFT. - Changes location grid and uses Principle Component Analysis to reduce size. - Increases robustness and distinctiveness of the SIFT descriptor. - Can outperform SIFT in many cases in the obtained paper.
3) Histogram of Oriented Gradients (HOG, Dalal and Triggs 2005)	<ul style="list-style-type: none"> - Idea: describe local object appearance and shape within an image through the distribution of intensity gradients/edge directions. - Divides an image into small cells. - For each cell, a histogram of gradient directions/edge orientations is computed → descriptor: combination of these histograms. - Advantages: invariant to geometric and photometric transformations, except object orientation. - Suited for human detection in images.
4) Speeded Up Robust Features (SURF, Bay et al. 2006)	<ul style="list-style-type: none"> - Similar to SIFT. - Faster and more robust against image transformations than SIFT. - Discrete image correspondences are searched via (1) a selection of interest points at distinctive locations in the image and (2) the representation of the neighbourhood of each interest point by a feature vector. - Does not use colour. - Invariant detectors and descriptors have a good compromise between feature complexity and robustness to commonly occurring deformations (like SIFT).

Table 6: A selection of local features.

Generally speaking, *local feature extraction* computes a set of features for every pixel or small sub-image using its neighbourhood (e.g. average colour values across a small block centred on that pixel or sub-image (Datta et al. 2008)). Such local features are capable of recognising objects and faces, but are highly complex to calculate and require much computational power (Arampatzis et al. 2013, Deselaers et al. 2008). Methods for extracting local features are for example SIFT (*Scale Invariant Feature Transform*, Lowe 1999) features, SURF (*Speeded Up Robust Features*, Bay et al. 2006), GLOH (*Gradient Location and Orientation Histogram*, Mikolajczyk and Schmid 2005) or HOG (*Histogram of Oriented Gradients*, Dalal and Triggs 2005). Further local features comprise e.g. shape context, cross correlation, steerable filters, spin images, differential invariants, complex filters, and moment invariants (see e.g. Mikolajczyk and Schmid (2005) for an overview and comparisons).

2.2.3.3 Global vs. Local Features: Briefly Compared

Global features are generally noisier than local features. The biggest problem with global features is that they rank the *whole* collection. This is in contrast to textual techniques, where documents matching no query keyword are not retrieved (Arampatzis et al. 2013). On the other hand, local features provide slightly better retrieval effectiveness than global features (Aly et al. 2009), because they represent images with multiple points in the feature space. Global features are only single-point representations. Local features are therefore more robust, but come at the expense of computationally more complex calculations (Arampatzis et al. 2013). Local-feature techniques have high-dimensional feature spaces and need nearest neighbours approximation to perform points matching (Popescu et al. 2009). As a consequence, global features are still more popular in general CBIR systems due to their reduced computational greediness. However, ranking whole image databases doesn't make global features a very applicable solution, either. Therefore, there cannot be named one feature or feature class that fulfils all needs, but each extractable feature is more or less applicable for a certain task. Furthermore, all of these image features, as described in the introduction, cannot solve the semantic gap on their own. Thus, an on-going research endeavour is to define methods and procedures to appropriately incorporate these low-level features into ISEs. A comprehensive overview and comparison of the performance of a large variety of visual descriptors can be found in Deselaers et al. (2008), although since then, further features have been introduced. General overviews of low-level features and CBIR can also be found in Rui and Huang (1999), Smeulders et al. (2000) or Datta et al. (2008).

2.3 Indexes

Very important for any information retrieval system is the way in which the extracted features are stored. An index enables efficient filing and fast retrieval. If no index is used, in the worst case, all documents need to be searched through to find the desired one (linear search time). The purpose of indexes is to *decrease search time* to a bearable minimum (e.g. logarithmic or even constant search time). Many different indexes and underlying data structures exist, and some will be introduced hereafter.

2.3.1 Indexes for Terms and Image Content

A number of different textual index structures in the literature are *inverted indexes* (Zobel and Moffat 2006), *signature files* (Faloutsos and Christodoulakis 1984), and *suffix arrays* (Manber and Myers 1993), but almost all current web search engines and text information retrieval systems are based on inverted indexes (Ounis et al. 2011). According to Manning et al. (2008), inverted indexes are the most effective way of storing documents for fast text retrieval. Index structures for indexing terms are also in use for the purpose of indexing low-level features extracted from images (Lux and Chatzichristofis 2008). Consequently, they are not examined separately. An inverted index, as the name already implies, stores a feature and corresponding document identifiers inversely (Manning et al. 2008).

Think again of a dictionary at the end of a book, where for each word of interest, the pages are written next to the word. For example, the word combination “Linked-List” found in a computer science book may show the pages “102, 104, 239”. The exact same principle applies to inverted indexes and is depicted exemplarily in Figure 3. Each image has a distinct *identifier* (a book page in the above example), and each term of the index *points* to all the identifiers of images containing the term in either their title, description, or both. Such an index structure greatly decreases storage size because each term is only saved once for an image collection. A term index can additionally store further statistical information about terms and documents, e.g. *document frequency* and *term frequency* (see chapter 2.4.2.1 Textual Relevance and Similarity).

Data structures used to store such indexes typically are either hash tables or tree structures, e.g. balanced binary search trees like AVL-trees or B-/B+ trees, the latter being especially popular within *database systems* (DBS, Ounis et al. 2011). Table 7 shows typical complexities of these data structures denoted in big O notation. The next section briefly explains hash tables, because it is one of the most efficient and often used data structures for inverted indexes, also implemented within *Lucene*. For in-depth analyses of binary search trees, see Goodrich et al. (2004). Additionally, for B-/B+ trees, Elmasri and Navathe (2011) offer detailed descriptions.

Data structure	Insert		Search		Delete		Space	
	Avg.	WC	Avg.	WC	Avg.	WC	Avg.	WC
Hash table	$O(1)$	$O(n)$	$O(1)$	$O(n)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$
AVL-tree	$O(\log n)$	$O(n)$	$O(n)$					
B-Tree	$O(\log n)$	$O(n)$	$O(n)$					

Table 7: Examples of index data structures and complexities.

Avg. denotes average complexity, WC represents worst case complexity. Composed from Goodrich et al. (2004) and Elmasri and Navathe (2011).

2.3.1.1 Hash Table

Descriptions are based on Goodrich et al. (2004). The fast expected insertion and retrieval times of $O(1)$ for hash tables (Table 7) come at the main drawback of its storage size N , which is typically larger than the actual number of elements n stored to be efficient. A hash table normally uses an array at its base. A container holding a *(key, value)* pair can be assigned to each cell of the array. A key in text indexes may be represented by a stemmed word (see Figure 7). A value could be an image identifier. A key represents an array cell number. Thus, a key made of characters needs to be converted to a number. Mapping a key to an integer value is accomplished by a *hash function*. The resulting *hash code* can easily exceed the range of possible array cell numbers (being larger or even negative) dependent on the hash function. As a result, the second step involves normalising or *compressing* the hash code to an actual array cell number. This is accomplished through the use of *compression maps*. After compression, the hash code represents an array cell number and the *(key, value)* pair is simply assigned to this cell. Unfortunately, such a calculation may produce equal cell numbers for storage of a *(key, value)* pair. Therefore, there exist measures not further specified here to avoid overwriting of

existing (*key, value*) pairs. To retrieve a term's image identifiers, a query term has to undergo the same key deriving processing chain to calculate the array cell in which the identifiers (values) are stored.

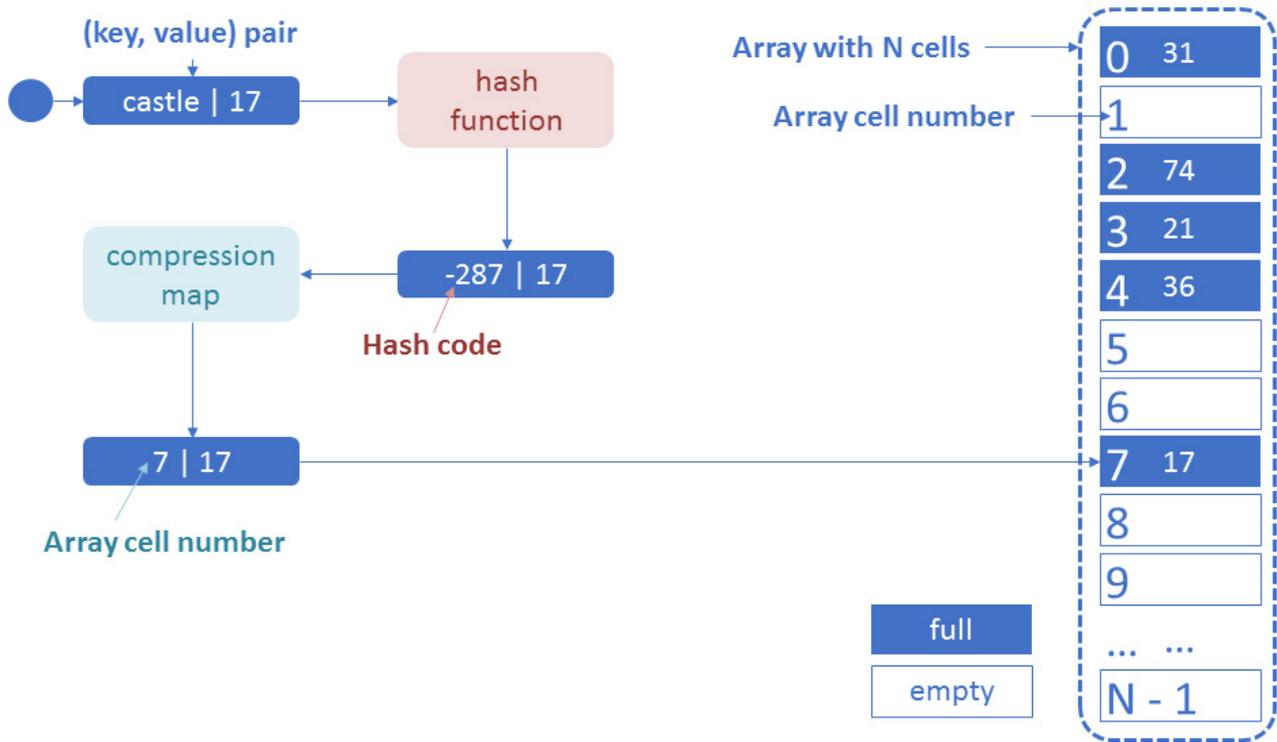


Figure 7: Basic structure of a hash table and assignment of terms to an array cell.

2.3.2 Indexes for Spatial Features

According to Ounis et al. (2011), the main approaches for spatial indexing can be classified into *space-filling curves*, *grid files* and *tree-based methods* (*space-* and *data-partitioning*) as shown in Table 8. Of those, R-trees, the basic data structures for plenty of multidimensional indexes implemented in DBS mainly for the purpose of supporting spatial access methods, are exemplary presented in more detail.

Data structure	Description
Space-filling curves	<ul style="list-style-type: none"> - Points close to each other in space are close to each other on a curve (Ounis et al. 2011). - See Böhm et al. (1999), Gaede and Günther (1998), Samet (2006).
Grid file	<ul style="list-style-type: none"> - Splits space into a non-periodic grid. - One or more cells of the grid refer to a small set of points. - See Nievergelt et al. (1984).
Space-partitioning	<ul style="list-style-type: none"> - Divide space into disjoint tiles. - Examples: quad-trees (Finkel and Bentley 1974) or kd-trees (Bentley 1975). - Disadvantage: store objects that span across borders between tiles twice.
Tree-based methods	<ul style="list-style-type: none"> - Divide spatial objects into disjoint subsets. - Examples: <i>R-tree</i> family (Guttman 1984).
Data-partitioning	<ul style="list-style-type: none"> - Widely used data structures for spatial index structures. - Advantage: store objects only once. - Disadvantage: use overlapping subareas → multiple visits of sub trees possible. - <i>R* trees</i> or <i>R+ trees</i> solve some of the R-trees drawbacks.

Table 8: Different possible index structures for spatial indexing.

2.3.2.1 R-Tree

The following descriptions are based on Shekhar and Chawla (2003). B-trees can only be used for one-dimensional space. Spatial objects however are at least two- or three-dimensional. R-trees, first mentioned in Guttman (1984) as a natural extension of B-trees, are height-balanced trees that can handle multidimensional, spatial objects. An object in an R-tree is represented by its MBR. An R-tree has the following properties:

- 1) Every *leaf node* contains between m and M index records, unless it is the *root* (where $m \leq M/2$).
- 2) For each index record $(I, \text{tuple-identifier})$ in a leaf node, I is the MBR that spatially contains the k -dimensional data object represented by the indicated tuple.
- 3) Every *non-leaf node* has between m and M children, where $m \leq M/2$, unless it is the root.
- 4) For each entry $(I, \text{child-pointer})$ in a non-leaf node, I is the MBR that spatially contains the rectangles in the child node.
- 5) The root node has at least two children, unless it is a leaf.
- 6) All leaves appear on the same level.
- 7) All MBRs have sides parallel to the axis of a global coordinate system.
- 8) The maximum number of levels is $\lfloor \log_m N \rfloor - 1$ (N = total number of entries of the tree).

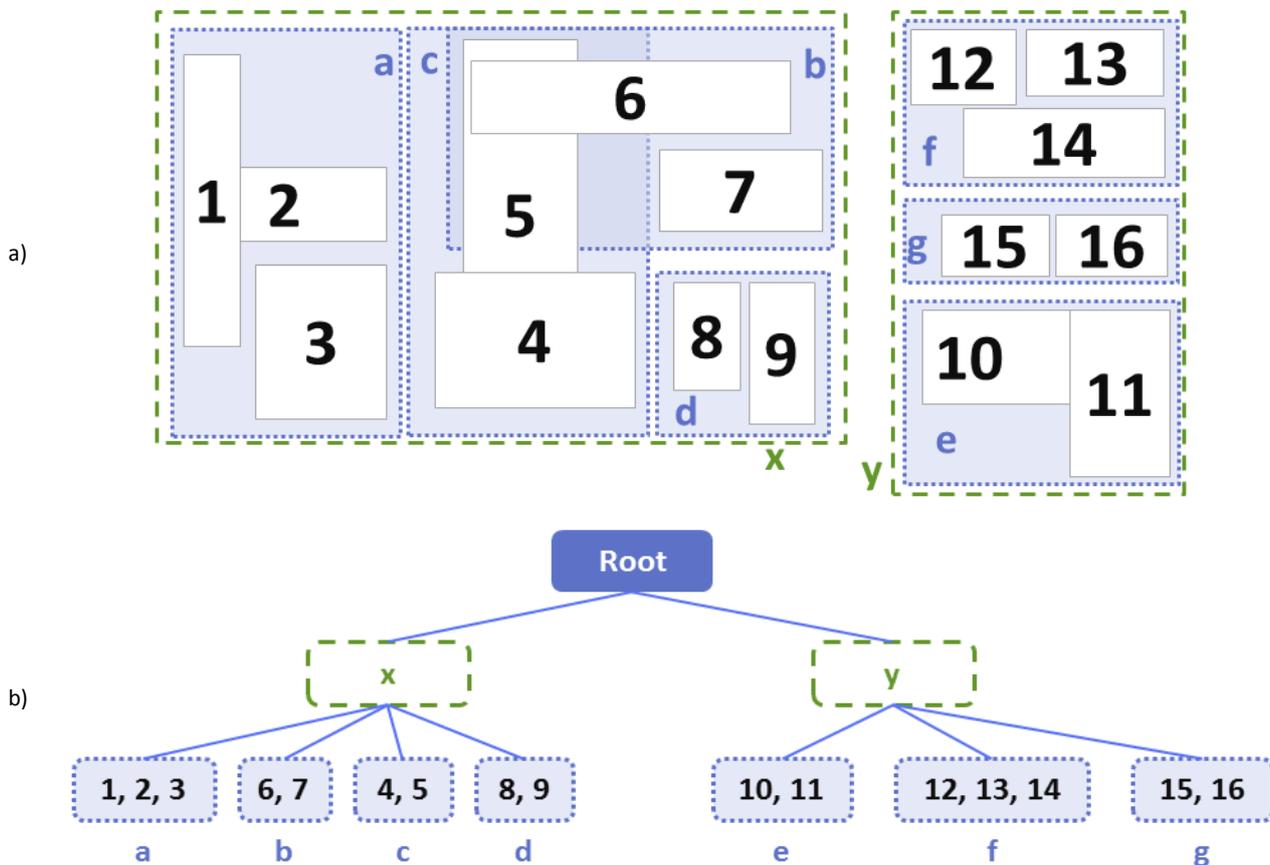


Figure 8: Functionality of an R-Tree.

Spatial objects assigned to nodes of the R-tree are shown in a). Leaves are coloured light blue, and objects are white. In b), the resulting R-tree with its root, intermediary nodes (non-leaf nodes, x and y) and leaves (a - g) containing the MBR of the spatial objects (1 – 16), is shown.

Figure 8 a) displays such a set of spatial objects (their MBRs) in a two-dimensional space. Here, a tree node can have three entries maximum. Queries, insertions and removals are processed recursively. In Figure 8 b), the resulting tree structure is depicted, which can be used as the base of a spatial index.

2.3.3 Hybrid approaches

To tackle the problems arising from multidimensional indexing and retrieval tasks that are common in GIR, index structures *combining* spatial and textual features have been introduced. The main purpose of these structures is to increase retrieval speed and make the two geographical dimensions, topic and space, easier accessible. Vaid et al. (2005) experimented with two schemes where either first the textual and then the spatial, or the spatial first and then the textual dimension are indexed. These schemes are implemented in SPIRIT (Purves et al. 2007). The first scheme is called *spatial primary index ST*. Here, the space of a geographical coverage of place names found in documents is divided into a set of regular grid cells. For each cell, an inverted index is constructed. The documents in this inverted index are only those, whose footprints intersect with the corresponding spatial cell. The other approach is the inversion of ST, the *text primary spatio-textual index TS*. A pure text index structure is modified so that the list of documents for each term is associated with a spatially-grouped set of documents that contain that term. Such a hybrid index can reduce query time, but index sizes increase noticeably. A similar approach is used in Brisaboa et al. (2010), where spatial and inverted indexes are combined into one index structure using an ontology of geographic space. This ontology structures space hierarchically into four levels (continent, country, region, populated place). All these data structures can solve pure textual queries, pure spatial queries, text queries with place names as well as text queries over a geographic area. However, there are authors who intentionally keep separated indexes for text and geographic aspects. Martins et al. (2005) provides separated indexes for:

- 1) more efficient query processing in the case of mono-dimensional queries,
- 2) the possibility to independently update both indexes and specifically optimise either of the index structures, and
- 3) to enable experiments on different combination strategies for the two dimensions.

Similarly, Gaio et al. (2008) (see also Palacio et al. 2011, Palacio et al. 2012) build their document search engine for *Virtual Itineraries in the Pyrenees* (PIV) on separate indexes, incorporating yet another geographic dimension, time, so that in the end, three indexes for topic, space, and time are supported and different combination strategies can be tried out. The question remains whether the smaller retrieval time of combined indexes weighs more or less than an increased flexibility to adjust and replace an existing index with higher retrieval times. As this brief introduction to hybrid indexes implies, much research is going on in this field and to date, barely any method is operationally implemented.

2.4 Information Retrieval Process Flow

For any retrieval process to be effective, it has to be possible to formulate queries, extract the right features from the query, and retrieve the most matching documents. Matching documents then need to be presented in an adequate way to the user. For each of the three dimensions - text, space, and image content - different methods to formulate corresponding queries are required, which are introduced hereafter.

2.4.1 Query Formulation and Feature Extraction

In GIR, a query can be characterised by a triplet of <theme><spatial relationship><location> (Jones and Purves 2008). An example is <Churches><in><Scotland>. However, such a query formulation requires users to have *geographic knowledge* not necessarily present. An approach avoiding such problems is to provide the user with the possibility to *sketch* an area on a map, where they want to search for a theme (see 2.5.2 User Interfaces in the Context of GIR). However, such an approach assumes users to already have a spatial interest on formulating a query and consequently makes such an interface a specialised tool. Advantages of such an approach, however, are that no geometry needs to be provided for a place name. Furthermore, the place name does not need to be disambiguated. Disambiguation may especially pose a problem in the case of hard to define areas like “Scottish Highlands”. The first part, <theme>, is typically submitted in form of *free text queries* (Manning et al. 2008). Retrieving images, however, requires other approaches if they are not accompanied by textual metadata. Datta et al. (2008) summarise various ways to pose a query for the purpose of retrieving images. To be able to search for images in CBIR, a query has to involve an *example image*. Low-level features corresponding to those extracted from collection images and stored in the content index, are then extracted from this image. The problem such a *query-by-example* approach faces is that a user is assumed to have an image at hand as an input query, which is barely the case. Therefore, there exist systems that let the user *sketch* what should be retrieved. However, this requires the user to *know* what he is looking for and certain drawing skills as well. For this reason, users nowadays are still more used to textual query formulation. Chapter 2.4.3 Retrieval will deal with ways to adequately incorporate example images into a query. In general, all the term, spatial, and low-level features extracted from an input query need to be exactly the same as those of the indexed images. Otherwise they cannot be compared. Chapter 2.2 Information Extraction Process Flow gives an insight into possible features extractable from text and images. Before we can retrieve images, the notion of *matching* a query’s features to indexed documents’ features needs to be defined for each of the three dimensions (text, space, and image content). “Matching” already implies some sort of *similarity* comparison between the features of two images. This is where the previously encountered term *relevance* comes into play, a close relative to *similarity* in IR.

2.4.2 Matching, Similarity and Relevance

Extracted features are used to evaluate *relevance*, which is the relationship between a *user's information needs* (UINs) and the resources/documents available to meet those needs (Frontiera et al. 2008). Relevance is a subjective concept impossible for a system to measure directly (Blair 1979). Therefore, it is implemented as a *matching function* evaluating the *similarity* between system representations of queries and available information resources/documents (Frontiera et al. 2008). This is the *core part of retrieval*, because the extracted and indexed features seen in the chapters before define to a great extent the type of similarity that can be assessed. A comprehensive review of the meaning of relevance is not the topic of this thesis. However, the interested reader may refer to Hjørland (2009), who gives an in-depth analysis. The main problem can be summarised to what a system can algorithmically calculate on the one hand and the concept of relevance a user has in mind on the other hand, similar to the semantic gap in CBIR. Whereas relevance is only meaningful in relation to goals and tasks, a system does not have any goals or tasks. Therefore, only users can judge retrieved documents to be *relevant*, and this relevance is *subjective* for each user. Systems can only *estimate* relevance from document features. These features, however, are defined by users to *mimic* the task of measuring actual relevance. A query's features are *matched* against the indexed features, and those documents whose features show the best accordance with the query's features are considered to be more relevant than those spotting only minor or no similarities at all. As an example, a query for "Bridges north of London" could simply be solved by retrieving images having the word "Bridges" assigned in their textual descriptions and that were taken "north of London". Such Boolean queries (Manning et al. 2008), where a document either matches or does not match a query, will *not* assign any relevance to an image compared to other pictures in the collection. Thus, every image having these words assigned would be considered equally relevant to the query. A matching score, on the other hand, weighs indexed document differently according to a user query and allows the retrieval of a *sorted result* or *ranked list*. Here, the highest scored documents represent the highest estimated relevance to a query and are, therefore, most likely those that can fulfil an UIN. The following part is concerned with practical implementations. Furthermore, the assessment of relevance through similarity measures is introduced.

2.4.2.1 Textual Relevance and Similarity

The contents of this section are, if not otherwise specified, based on Manning et al. (2008). An established way of quantifying the similarity between two documents d_1 and d_2 is to compute the so-called *cosine similarity* of their vector representations $\vec{V}(d_1), \vec{V}(d_2)$ (Formula III).

$$\text{III} \quad \text{similarity}(d_1, d_2) = \cos(\theta) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)| |\vec{V}(d_2)|}$$

The numerator represents the *dot product* of the vectors $\vec{V}(d_1)$ and $\vec{V}(d_2)$ (Formula IV), whereas the denominator is calculated using the product of their *Euclidean lengths* (Formula V).

$$IV \quad \vec{V}(d_1) \cdot \vec{V}(d_2) = \sum_{i=1}^n V(d_1)_i V(d_2)_i$$

$$V \quad |\vec{V}(d_1)| |\vec{V}(d_2)| = \sqrt{\sum_{i=1}^n V(d_1)_i^2} \cdot \sqrt{\sum_{i=1}^n V(d_2)_i^2}$$

The denominator of Formula III normalises the length of the vectors $\vec{V}(d_1)$ and $\vec{V}(d_2)$ to the *unit vectors* $\vec{v}(d_1) = \frac{\vec{V}(d_1)}{|\vec{V}(d_1)|}$ and $\vec{v}(d_2) = \frac{\vec{V}(d_2)}{|\vec{V}(d_2)|}$, so that the *cosine similarity* can be rewritten to Formula VI:

$$VI \quad \text{similarity}(d_1, d_2) = \cos(\theta) = \vec{v}(d_1) \cdot \vec{v}(d_2)$$

Looking back at Figure 5 reveals that this similarity is actually the cosine of the angle θ between two document vectors (here in a two-dimensional vector space). Geometrically, the similarity is highest when $\cos(\theta)$ is 1. This happens when the angle θ between the two vectors is 0, meaning that the two vectors fall together. A value of zero is achieved if query and a document do not share a single word. An illustrating example of the whole procedure is given in Table 9.

Description	Mathematical representations and calculations												
1) Documents d_1 and d_2 of a collection.	$d_1 = \{\text{old, blue, house, car}\}$ $d_2 = \{\text{new, blue, house, tree}\}$												
2) Weights (e.g. tf-idf _i) of d_1 and d_2 .	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>Blue</td> <td>Tree</td> <td>House</td> <td>New</td> <td>Old</td> <td>Car</td> </tr> <tr> <td>1</td> <td>0.9</td> <td>0.6</td> <td>0.02</td> <td>0.06</td> <td>1.6</td> </tr> </table>	Blue	Tree	House	New	Old	Car	1	0.9	0.6	0.02	0.06	1.6
Blue	Tree	House	New	Old	Car								
1	0.9	0.6	0.02	0.06	1.6								
3) Vector representations of d_1 and d_2 using weights (e.g. $w_t = \text{tf-idf}_t$).	$\vec{V}(d_1) = (w_{\text{blue}}, w_{\text{tree}}, w_{\text{house}}, w_{\text{new}}, w_{\text{old}}, w_{\text{car}}) = (1, 0, 0.6, 0, 0.06, 1.6)$ $\vec{V}(d_2) = (w_{\text{blue}}, w_{\text{tree}}, w_{\text{house}}, w_{\text{new}}, w_{\text{old}}, w_{\text{car}}) = (1, 0.9, 0.6, 0.02, 0, 0)$												
4) Cosine similarity between d_1 and d_2 .	$\cos(\theta) = \frac{(1^2) + (0.6^2)}{\sqrt{1^2 + 0.6^2 + 0.06^2 + 1.6^2} \cdot \sqrt{1^2 + 0.9^2 + 0.6^2 + 0.02^2}} = 0.55 (\approx 56.63^\circ)$												

Table 9: Calculation of cosine similarity values from tf-idf.

If a word of the set of words from both documents does not occur in one document, it is assigned a tf-idf weight of 0.

Most queries submitted to search engines on the World Wide Web are *free text queries*. Such queries can be viewed as a set of words like the indexed documents of the collection. Thus, a query can be represented as a vector exactly the same as a document (for a query q , a vector $\vec{v}(q)$ can be constructed as depicted in Figure 5). Formula VI can therefore be rewritten to $\text{similarity}(q, d) = \vec{v}(q) \cdot \vec{v}(d)$, where $\vec{v}(q)$ is the unit vector of the query. A similarity score is then calculated with Formula VI and assigned to each document of the collection. The documents most similar to the query are finally retrieved as an ordered list of scores from highest score to lowest (maximum value 1.0, minimum value 0.0). If no inverted index was used, the query would have to be compared to each of the documents in the collection. The inverted index allows comparing the query to only those documents that contain the query's terms.

2.4.2.2 Geographical Relevance and Similarity

Spatial or *geographical relevance* is defined as a relation between a human's *geographical information needs* and *geo-referenced information objects* (e.g. documents, images, maps, etc. Raper 2007). However, practically, geographical relevance estimation is also implemented as similarity measures. Two fundamental principles underlie most concepts of geographic relevance (Frontiera et al. 2008):

- 1) *First law of geography*: “everything is related to everything else, but near things are more related than distant things” (Tobler 1970).
- 2) *Topology matters, metric refines* (Egenhofer and Mark 1995): “In geographic space, topology is considered to be first-class information, whereas metric properties, such as distances and shapes, are used as refinements that are frequently captured”.

Modelling these two properties of geographic relevance involves the use of geocentric coordinate systems for geo-referencing documents and queries, as well as methods to calculate relationships between geometric objects. Cai (2011) mentions another point important for the human judgement of geographic relevance:

- 3) “The duality of human spatial cognition, where mental imagery and image schemata are both playing a role”. This means that humans mainly use place names and landmarks to refer to geographic locations rather than geographical objects that can be derived from borders of e.g. countries or districts.

As a consequence, GIR systems have to provide both place names (as an input possibility for humans) and geometric representations (for calculating relevance or similarity, respectively), if a geometric approach shall be applied (Frontiera et al. 2008). Geometric approaches approximate geographical relevance by spatial similarity measures defined by *metric* characteristics (e.g. area, perimeter, length, shape, density, etc.), *topological* relationships (e.g. distance, overlap, contain, nearness, adjacency, etc.) and *directional* relationships (e.g. south, east, west, north etc., Frontiera et al. 2008). A spatial footprint, as defined in chapter 2.2.2.1 Spatial Features, can be used to determine these relationships. If the query is a point, point-in-polygon and distance-based near relationships can be calculated. If the footprint is a polygon, intersection of polygons is the most widely used similarity function (e.g. “area of overlap”, Hausdorff Distance, etc., see Larson and Frontiera (2004) for an overview). Thus, there exist many different geometric approaches. However, such approaches may come at the cost of high computational requirements. Therefore, instead of using the most accurate polygon representation, it may be enough to only use a point, MBR or convex hull. MBR approximations, furthermore, seem to be the favourable choice for computing spatial similarity/relevance, although some level of detail may be lost through the use of such representations (Frontiera et al. 2008, Cai 2011).

2.4.2.2.1 Point-based Relevance Ranking

Similar to any other extracted feature (terms or image low-level), the type of representation (e.g. point, polygon, MBR, etc.) to index geographic content determines possible spatial similarity estimation functions (Frontiera et al. 2008). Some geometric and topologic methods to estimate spatial relevance from point locations, as they usually occur in the case of images, will be introduced hereafter. See Larson and Frontiera (2004) and Frontiera et al. (2008) for estimation possibilities where a polygon is associated with the indexed documents.

Topological Relationship: Inside. Inside relationships describe a binary (Boolean) operator between a query spatial footprint and an indexed image's spatial footprint (Purves et al. 2007). If the image's footprint is contained within the query's MBR, it is considered inside and assigned the score 1.0. If it is not contained within the query's footprint, it is considered outside and assigned the score 0.0.

Topological Relationship: Near. The near relationship is one of the most difficult concepts to formalise because of the inherent vagueness of the word "near". Two possible examples from the literature are presented hereafter. The first one is taken from Purves et al. (2007) and describes an *exponential* function in Formula VII.

$$\text{VII} \quad \text{ExponentialNear}(P_q, P_c) = e^{-L \cdot d(P_q, P_c)}$$

In this formula, P_q is the centroid of the query footprint and P_c is the centroid of the indexed document's footprint (which, in the case of an image, is only its point location). $d(P_q, P_c)$ describes the Euclidean distance between the query centroid and the image's point location. The score decays from 1.0 to 0.0 with increasing distance. L controls the rate of decay, or how far away from P_q an image's P_c is still considered to be near and thus, relevant.

A second possible implementation of "near" is based on a formula found in Kamahara et al. (2012) and describes a simple *linear* function in Formula VIII.

$$\text{VIII} \quad \text{LinearDistance}(P_q, P_c) = \frac{d(P_q, P_c)}{R}$$

$d(P_q, P_c)$ represents the distance between the query footprint P_q (represented as centroid of the retrieved footprint) and the image footprint P_c (which is a point location). R describes the radius of how far away from P_q an image's P_c is still considered to be near. To get a score smaller or equal to 1.0, where 1.0 represents closest proximity from P_q , Formula VIII needs to be subtracted from 1.0, resulting in Formula IX.

$$\text{IX} \quad \text{LinearNear}(P_q, P_c) = 1.0 - \frac{d(P_q, P_c)}{R}$$

If R is smaller than $d(P_q, P_c)$, then $LinearNear(P_q, P_c)$ is smaller than 0.0. However, because any $d(P_q, P_c)$ larger than R is not considered inside the relevant range, a *threshold* can be set so that images further away than R are automatically discarded.

The remaining problem is that what is considered to be near something else is subjective to the evaluating person. Furthermore, it depends on the spatial resolution of the spatial object in question (e.g. Zurich is near St. Gall compared to Bale, but Switzerland is also near England compared to the USA). Therefore, setting the parameters right for such relationships is a non-trivial task with not a single or simple solution. A sensible suggestion is to have the distance dependent on the MBR of the retrieved spatial footprint (see Purves et al. 2007).

Directional Relationships: North- , South- , West- and East-of Relation. Formula X can also be found in Purves et al. (2007). Assuming again that P_c and P_q are the centroids of a document and a query footprint, respectively, and that φ is the angle of the vector $\overrightarrow{P_q P_c}$ from the positive x axis with the origin assumed at point P_q , then Formula X is used to describe the north-of relationship:

$$X \quad north - of(P_q, P_c) = \begin{cases} 1 - \frac{|90 - \varphi|}{90} & \text{if } \varphi < 180^\circ \text{ or } \varphi > 0^\circ \\ 0 & \text{if } \varphi \geq 180^\circ \text{ or } \varphi \leq 0^\circ \end{cases}$$

All other directional operators (south-, east-, west-of) are calculated accordingly by adapting the range of degrees. Taking into account the first law of geography (see chapter 2.4.2.2 Geographical Relevance and Similarity), the final score is multiplied by one of the aforementioned near Formulae VII or IX.

Probabilistic Similarity Estimations. As an addition to geometric functions, Frontiera et al. (2008) experimented with probabilistic approaches (see also van Rijsbergen 1979), which involve a similarity score representing the *probability* that a document is relevant to a query. These methods are based on the observation that relevance cannot be known with certainty and should therefore be estimated probabilistically (Maron 1960), which makes it suitable for geographic objects that inherently have some degree of uncertainty (Goodchild 1999). The method uses a linear form of a beforehand trained logistic model estimating probability of relevance as a function of properties of the query-document pair.

Tile-based standardisation. Palacio et al. (2011) use a tile-based standardisation to represent spatial (and temporal) objects homogeneously. The approach bares similarities to stemming and weighting methods used in textual IR (e.g. term frequency in VSM, see 2.4.2.1 Textual Relevance and Similarity). This spatial standardisation, or tiling, allows grid-based or administrative zoning (e.g. district, city, and county) of the territory mentioned in a document collection and a projection of spatial features of the spatial raw index on this segmentation. The frequency of a tile then corresponds to the number of spatial features that intersect it (this applies also to the topical and temporal dimension used in their

work). All dimensions therefore can be homogeneously represented as a tile. This approach is especially well-suited for textual documents containing many extractable geographic references.

2.4.2.3 Content Relevance and Similarity

Despite all efforts made in recent years, there is not yet a universally accepted algorithmic approach for characterising human vision (Datta et al. 2008). Some examples of extractable signatures are provided in chapter 2.2.3 Image Content Extraction. This section deals with the assessment of *image similarity*. Image similarity measures can be grouped into the following classes according to design philosophy (Datta et al. 2008):

- 1) Features as vectors, non-vector representations or ensembles.
- 2) Region-based similarity, global similarity, or a combination of both.
- 3) Similarities computed over linear space or non-linear manifold.
- 4) The role of image segments in similarity computation.
- 5) Stochastic, fuzzy, or deterministic similarity measures.
- 6) Supervised, semi-supervised, or unsupervised learning.

The following descriptions are, if not otherwise specified, based on Smeulders et al. (2000) and are far from being exhaustive. The interested reader may refer to Smeulders et al. (2000) as well as Datta et al. (2008) for more detailed descriptions of low-level similarity measures.

Global features (2.2.3.1 Global Features) often make use of some kind of histogram. These histograms can be seen as low-level feature vectors. The similarity between two images in its general form is defined by the similarity between the two feature vectors \mathbf{F}^q and \mathbf{F}^d of the query image q and an image d of the indexed data set. In its unspecified form in Formula XI, g is a positive, monotonically non-increasing function and d is a distance function.

$$\text{XI} \quad \text{similarity}(\mathbf{F}^q, \mathbf{F}^d) = g \circ d(\mathbf{F}^q, \mathbf{F}^d)$$

A possibility for d could be the intersection distance (Swain and Ballard 1991) described in Formula XII.

$$\text{XII} \quad d_{\cap}(\mathbf{F}^q, \mathbf{F}^d) = \sum_{j=1}^n \min(\mathbf{F}_j^q, \mathbf{F}_j^d)$$

Where \mathbf{F}_j^q and \mathbf{F}_j^d are two histograms of images containing n bins. For easier understanding, Figure 9 visualises Formula XII. It is divided by the whole area of d 's histogram to retrieve a similarity score between 0.0 and 1.0 dependent on d .

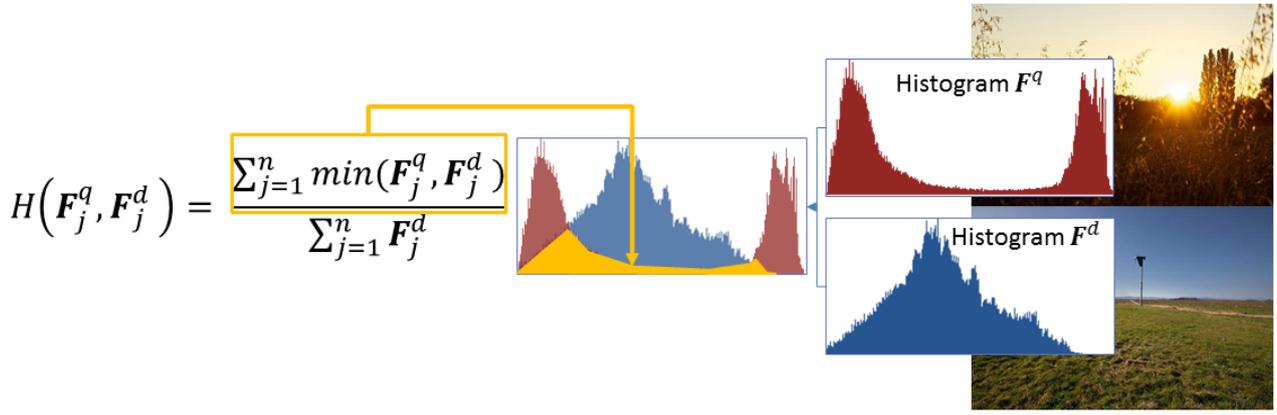


Figure 9: Visualisation of the intersection distance.

Another approach can be seen in Formula XIII, where the distance between two histograms is defined in vector form as:

$$\text{XIII} \quad d_M(\mathbf{F}^q, \mathbf{F}^d) = \sqrt{(\mathbf{F}^q - \mathbf{F}^d)^t \mathbf{M} (\mathbf{F}^q - \mathbf{F}^d)}$$

\mathbf{M} is a matrix expressing the similarity between the bins j and k of two histograms. The advantage of this method is that the similarity can be considered being between values in the feature space, making inclusion of the feature space's metric into the similarity measure possible.

In chapter 2.2.3.1 Global Features, two low-level features based on histograms, FCTH and CEDD, as well as their combination JCD, were introduced. These two measures use the Tanimoto coefficient (Chi et al. 1996, Formula XIV) for assessing similarity of two images.

$$\text{XIV} \quad T_{ij} = t(x_i, x_j) \frac{x_i^T x_j}{x_i^T x_i + x_j^T x_j - x_i^T x_j}$$

More methods can be found in the provided literature. E.g. Datta et al. (2008) summarises popular distance measures.

2.4.3 Retrieval

The final step of any search engine is the retrieval and presentation of a sorted ranked list of results, where more relevant documents (higher similarity score) are listed higher than less relevant ones. At least, this is a common way for *one-dimensional* retrieval systems (e.g. only based on one textual index). Typical for GIR systems, however, is the occurrence of more than one result list from different dimensions. In most cases, one dimension is added to normal IR systems, namely *geographic space*. Additionally, it is also possible to have three result lists if the third geographic dimension *time* is added to the system (Palacio et al. 2011), or even more. In this thesis, there are three dimensions: text, space and image content. All dimensions will retrieve different result lists, each holding a potentially different image set and different scores assigned to each image according to the assessed similarity. Cai (2011) argues that for browsing with no special task in mind, a user may want to have several

different relevance levels (i.e. dimensions) to comb through. For search however, better use may be made through *combining* the different result lists. Such methods are summarised in the following section as *fusion algorithms*.

2.4.3.1 Result List Fusion

Literature proposes different approaches to fuse result lists. The aim is to assess the relevance of different dimensions individually and then merge the various dimensions before further processing. Such methods are introduced in the next section and are also referred to as *late fusion* in the literature (Maillot et al. 2007, Arampatzis et al. 2013), because fusion is carried out as a last step of combining result lists of different dimensions.

Combination according to document scores. Result list fusion can be accomplished by the Comb-family of fusion algorithms (Fox and Shaw 1993), summarised in Table 10.

Term	Description
CombMIN	Minimum of all scores of a document
CombMAX	Maximum of all scores of a document
CombSUM	Summation of all scores of a document
CombANZ	CombSUM divided by the number of nonzero scores of a document
CombMNZ	CombSUM multiplied by the number of nonzero scores of a document

Table 10: Comb Fusion algorithms.

Introduced by Fox and Shaw (1993).

In the following part, CombMNZ will be detailed as an example. Before CombMNZ can be applied, result lists need to be normalised to comparable ranges. This procedure is called *score normalisation* and is important when coping with set of scores with different ranges (He and Wu 2008). Min-max normalisation shown in Formula XV is a well-known procedure for this task. It normalises a score into the interval $[0, 1]$. $Score^{min}$ and $Score^{max}$ correspond to the minimum and maximum scores of a ranked list of one dimension (i.e. the textual, spatial or content dimension), and $Score^d$ is one of the scores of an image d of this ranked list *before* normalisation. $MinMaxScore^d$ is the normalised score of image d in the range of 0 and 1.

$$XV \quad MinMaxScore^d = \frac{Score^d - Score^{min}}{Score^{max} - Score^{min}}$$

Formula XVI shows how CombMNZ combines several scores of one image d into one score:

$$XVI \quad CombMNZ^d = NrOfNonzeroScores^d \cdot \sum_{i=1}^N (MinMaxScore_i^d)$$

Where d is a single document that was contained within N score lists and had a *nonzero* score in each of those. An example of applying CombMNZ to combine two lists of scores retrieved with different ranking schemes can be seen in Figure 10 (inspired by Palacio et al. 2011).

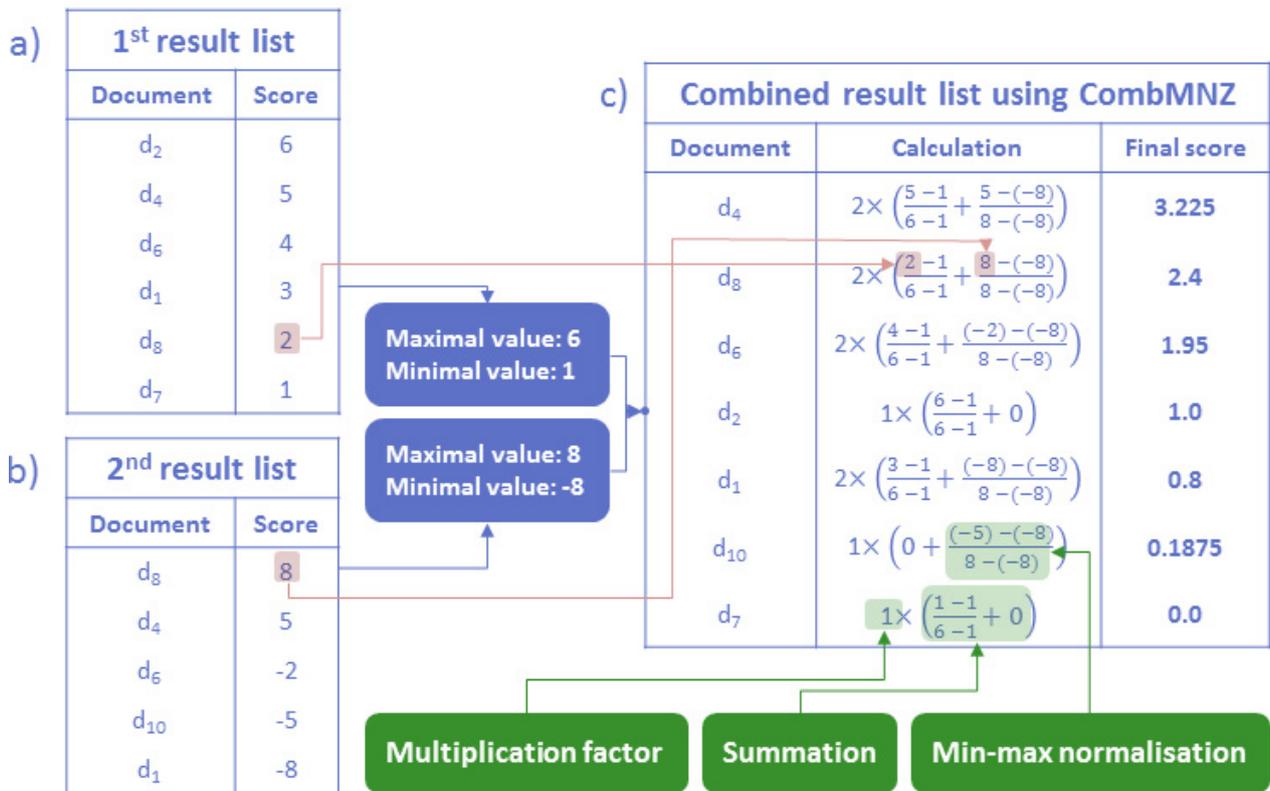


Figure 10: Illustration of CombMNZ.

Tables a) and b) represent two score lists with different ranges of ranks. From each list, minimal and maximal values are extracted and in table c), the lists are combined using CombMNZ with min-max normalised scores.

The multiplication factor before the min-max normalised summation depends on the number of result lists a document was contained in. If it was contained within all lists, this factor is 2 (in this example), as were d_4 , d_8 , d_6 and d_1 . However, a document only occurring in one result list gets only a factor of 1, e.g. d_2 , d_{10} and d_7 . In this case, the score not present in the corresponding result list within the sum of CombMNZ is simply represented by 0 for the corresponding dimension.

The calculation illustrates how the final combined score takes into account two factors (Palacio et al. 2011):

- 1) The *more often* a document is retrieved by an index, the higher its score.
- 2) The higher the initial score was, the higher the combined score is.

Therefore, the multiplication factor boosts documents that occur in *several* result lists more than such that may have an overall better final score, if only summation was applied, but are only relevant for *one* of the dimensions, not both (i.e. no score was retrieved for one of the dimensions). See e.g. d_1 in the combined result list. Although min-max normalisation will result in the second score of the sum to be 0, it is higher weighed than a document like d_{10} that only occurred in one result list (thanks to the higher multiplication factor), which is intuitively correct, because d_1 contributes overall more to an accurate relevance representation if all dimensions are regarded as relevant for the final result list.

Besides this basic version of the algorithm, it is also possible to weigh each dimension before summation (see Palacio et al. 2011), so that e.g. the spatial dimension weighs more than the term dimension.

Combination according to document ranks. Another prominent example is the Borda fusion algorithm family (Borda 1781). These algorithms, initially designed for voting, merge the result lists of a number of search engines by using the documents' *ranks* instead of their scores. Kraft et al. (2001) describes the basic method as follows:

"Each voter ranks a fixed set of c candidates in order of preference. For each voter, the top ranked candidate is given c points, the second ranked candidate is given $c-1$ points, and so on. If there are some candidates left unranked by the voter, the remaining points are divided evenly among the unranked candidates. The candidates are ranked in order of total points, and the candidate with the most points wins the election."

Transferred to IR, a *voter* is a retrieval system that returns a ranked list, and the *candidates* are all the documents in the corpus (e.g. 1,000,000 images). As mentioned in Palacio et al. (2011), for any query, there most likely are more unranked than ranked documents, resulting in an evenly divided number of points among the unranked documents. This leads to many tied documents with no actual relevance to the query. Two problems arise: firstly, from a user perspective, documents considered irrelevant by the retrieval system are attributed credit which is intuitively incorrect. Secondly, from a system perspective, these tied documents introduce a bias in the information retrieval evaluation. The authors therefore apply two changes to Borda fusion:

- 1) The top-ranked document gets n points, where n is the length of the longest result list.
- 2) Unranked documents get no points.

Additional details of result combinations based on document ranks can be found e.g. in Kraaij et al. (2007) and Liu (2007). Both CombMNZ as well as Borda fusion are experimented on in Palacio et al. (2011), and both provide similarly good retrieval results.

Multidimensional Scattered Ranking Methods for GIR. A combination of textual and spatial dimensions is used in SPIRIT (Purves et al. 2007) as well. The applied method is termed *scattered ranking* (van Kreveld et al. 2005). This method favours points close to a query, but also points far away from already ranked points, which minimises redundancy (e.g. consider a query like "castles near Koblenz". If this castle is in immediate vicinity of Koblenz, all these documents are ranked high. The user, on the other hand, may also want to retrieve images of castles located further away, not only those of the same castle).

Additional techniques. Further combination strategies found in the literature comprise (according to Palacio et al. 2011):

- *Parallel filtering*: intersecting the results of various result lists, so that only the documents relevant in all aspects are retrieved (e.g. Palacio et al. 2011).
- *Sequential filtering*: the spatial dimension is processed first and only on the remaining documents, the term-based scoring is applied (e.g. Larson and Frontiera 2004).
- *Linear interpolation*: combination of the different result lists by using a linear function, where weighted harmonic-mean supports a weighted combination (e.g. Martins et al. 2005, Brisaboa et al. 2010).

2.4.3.2 Learning Techniques

Several *learning techniques* have been introduced in recent years. Datta et al. (2008) give an overview of currently used learning techniques in Table 11 in the case of CBIR, which may also often be applied in other fields of IR. The aim of such learning techniques is to improve the retrieval performance and they are often applied *after* retrieving an initial result list. The application of such procedures is especially advisable for low-level feature image retrieval due to the semantic gap. For an in-depth description of techniques mentioned in Table 11, please consult Datta et al. (2008). Some examples of Table 11 important to this thesis shall be analysed in the following subsection.

Augmentation	User involvement	Purpose	Techniques	Drawbacks
Clustering	- Minimal.	<ul style="list-style-type: none"> - Meaningful result visualisation. - Faster retrieval. - Efficient storage. 	<ul style="list-style-type: none"> - Side-information. - Kernel mapping. - K-means. - Hierarchical. - Metric learning. 	<ul style="list-style-type: none"> - Same low-level features. - Poor user adaptability.
	<ul style="list-style-type: none"> - Requires prior training data. - Not interactive. 	<ul style="list-style-type: none"> - Pre-processing. - Fast/accurate retrieval. - Automatic organisation. 	<ul style="list-style-type: none"> - Support Vector Machine (SVM). - MIL. - Statistical models. - Bayesian classifiers. - K-NN. - Decision trees. 	<ul style="list-style-type: none"> - Training introduces bias. - Many classes unseen.
Relevance Feedback	<ul style="list-style-type: none"> - Significant. - Interactive. 	<ul style="list-style-type: none"> - Capture user and query specific semantics. - Refine initial result list accordingly. 	<ul style="list-style-type: none"> - Feature re-weighting. - Region weighting. - Active learning. - Memory/mental retrieval - Boosting. 	<ul style="list-style-type: none"> - Same low level features. - Increased user involvement.

Table 11: Learning techniques to enhance retrieval quality.

Pseudo-Relevance Feedback. Combining the two dimensions - text and space - may increase the relevance of results for GIR systems based on textual documents. For images, however, it may make sense to additionally incorporate its low-level features into the relevance assessment. After all, images are made of such features and assigned textual features are desirable, but not necessarily available. However, CBIR only provides searching by example, not by text. As mentioned before, users prefer to search by text instead (Hsu et al. 2007) and especially web image search is dominated by textual search techniques (Popescu et al. 2009). Another problem in CBIR is the mentioned *noisiness* of global features (2.2.3.1 Global Features), predominantly occurring if there are only few relevant images in the collection. The fact that global CBIR typically ranks *all* images in a collection, furthermore, makes it barely applicable for large image collections (Arampatzis et al. 2013). Contrariwise, local features (2.2.3.2 Local Features) are more robust, but computationally more complex. However, many search engines nowadays are web based and thus should provide computationally inexpensive methods (Popescu et al. 2009). *Relevance feedback* (RF, Table 11) can help overcome such problems. In RF, an initially ranked list is presented to a user for evaluation. The user then actively decides which documents of the ranked list are valuable and which are not. In the next step, the user's choices may be used to *re-rank* the initial list, leading to a result list that hopefully fulfils a UIN better. However, such a procedure adds a lot of active working steps to the user looking for images while it should actually be the system's task to find relevant images. A technique freeing the user from active intervention is *pseudo-relevance feedback* (PRF, Carbonell et al. 1997). PRF does not need a user to give feedback on which retrieved documents are relevant but the *system itself* (blindly) conducts a re-ranking of the initial ranked list. The assumption made is that the top-ranked K (e.g. $K = 10$) images of an initially retrieved ranked list are most likely the most relevant images to a query (Maillot et al. 2007). Therefore, the idea is to *first retrieve images by a secondary medium and then re-rank this initial list using a selection of images from the highest ranked ones* (Arampatzis et al. 2013). If the secondary medium is a simple text query, the aim is to elevate retrieved images that have low textual relevance but are highly relevant in terms of low-level image features. Thus, a content-based re-ranking procedure can be applied on the images of this initial result list according to their relevance to *example images* (EI) taken from the first top- K images. A main advantage is that PRF is only applied on a *subset* of the image collection, reducing the number of images needed to conduct computationally costly CBIR operations. Maillot et al. (2007) propose such a dual approach of textual and content retrieval in the context of the ImageCLEF 2006 photo retrieval task. PRF there uses a static $K = 3$ of the top-ranked images for re-ranking, and both global (colour, texture) and local (SIFT) features are experimented on. However, their approach applying CBIR PRF on an initial textual retrieval *decreases* retrieval performance. In Popescu et al. (2009), a re-ranking method compares each result to other query results and an external, contrastive class of items. The idea is that images visually similar to other retrieved images, but visually dissimilar to images found in the contrastive class, are likely to be good matches for re-ranking. Diversification of images is applied to present different aspects of the query to

the user, but although retrieval efficiency is *increased* through re-ranking compared to a conventional ISE, additional diversification *decreases* the gained performance again (nevertheless, diversity is, as expected, higher in the diversified case). Both methods, with and without diversification, however, have higher retrieval effectiveness than the used conventional ISE. In their work, K equals 30% of the initially retrieved top-ranked images. A very recent example can be found in Arampatzis et al. (2013). Their novelty to PRF re-ranking of an initial result list is the dynamic estimation of K , which is evaluated for each query separately using an approach formulated in Arampatzis et al. (2009). They conclude that the two-stage approach does not work well with local, but better with global image features. Further CBIR PRF approaches can be found in e.g. Barthel (2008) or Quemada et al. (2009), which all use a similar re-ranking through visual content or some form of PRF. Maillot et al. (2007) show that late fusion (see 2.4.3.1 Result List Fusion) of textual and content result lists cannot improve retrieval, but PRF can. Also Arampatzis et al. (2013) propose their PRF approach as an alternative to fusion procedures (due to fewer problems in deciding on how to *appropriately* combine result lists). However, none of these approaches explicitly incorporates a spatial dimension for improving the *initial* subset of relevant images for re-ranking, but focuses only on textual retrieval. Only Maillot et al. (2007) actively extract location names and even label them accordingly using POS and NER methods, but do not include any geometric or topologic relevance assessments to improve spatial retrieval. Therefore, research on approaches that include a geometric spatial dimension into this process may improve retrieval of spatially relevant images.

Agglomerative Hierarchical Clustering of Images. Unsupervised clustering techniques have been in use since the early days of CBIR to speedup image retrieval in large databases and to improve accuracy as well as automatic image annotation (Datta et al. 2008). The clustering method presented here is *hierarchical clustering* (Table 11, Clustering → Techniques). Specifically, the *agglomerative* version of this clustering method will be examined in more detail (Backhaus et al. 2006). The activity diagram in Figure 11 shows the basic steps for performing hierarchical clustering.

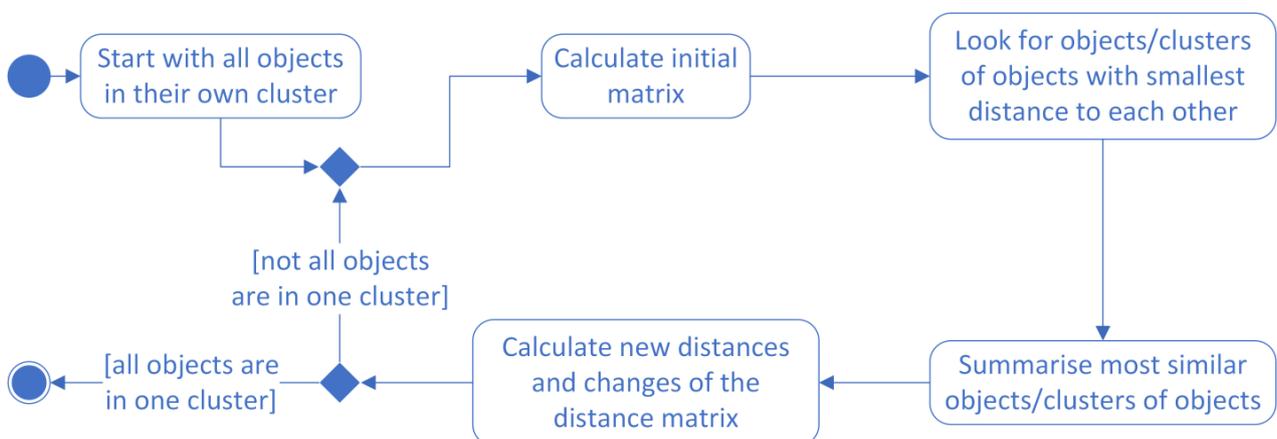


Figure 11: Activity diagram of the agglomerative hierarchical clustering algorithm.

Agglomerative hierarchical clustering is based on the idea that in a first step, all objects (here images) are in an own cluster (see Figure 12, where all images start in red cluster). Iteratively, all objects or clusters of objects closest together (meaning, having the smallest distance or dissimilarity between each other) are merged (the blue coloured boxes in Figure 12). The distance in the dendrogram of Figure 12 is represented by the length of a branch (the shorter, the more similar the objects are). Then, a new distance matrix only having the merged clusters and remaining objects is calculated with the new distance between all of the clusters and objects. This process is repeated until all the images are in one cluster (The biggest light blue box of Figure 12 encompassing all the images). There exist several different methods to calculate the new distance matrix containing the distances between newly formed clusters in each step, some of which are summarised in Table 12. One specific example to illustrate how these measures work is presented in Figure 13.

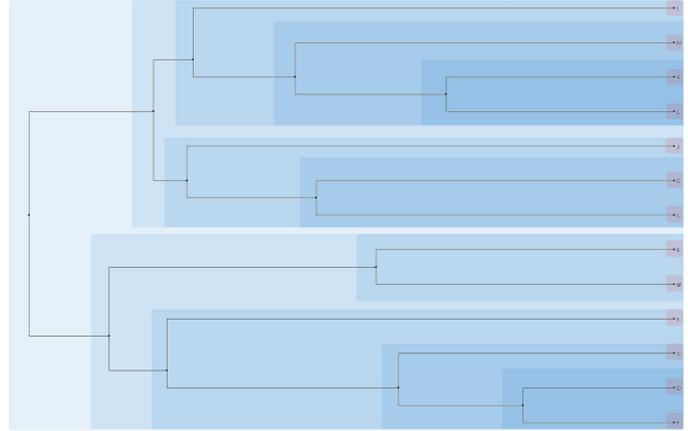


Figure 12: Illustration of agglomerative hierarchical clustering. All items (in this case, images) start in an own cluster (red) and are iteratively summarised until they are all in one cluster (light blue).

Method	Formula
Single Linkage	$d(A, B) = \min \{d(a, b)\}$
Complete Linkage	$d(A, B) = \max \{d(a, b)\}$
Average Linkage (not weighted)	$d(A, B) = \frac{1}{ A B } \sum_{a \in A, b \in B} d(a, b)$
Average Linkage (weighted)	$d(A, B) = \frac{1}{(A + B) + (A + B - 1)} \sum_{x, y \in A \cup B} d(x, y)$
Centroid	$d(A, B) = d(\bar{a}, \bar{b}); (\bar{a}, \bar{b} = \text{Center of clusters } A \text{ and } B)$
Ward	$d(A, B) = \frac{d(\bar{a}, \bar{b})}{\frac{1}{ A } + \frac{1}{ B }}; (\bar{a}, \bar{b} = \text{Center of clusters } A \text{ and } B)$

Table 12: Different methods for calculating the distance between clusters.

In these formulae, because each of the clusters inside a summarised cluster has a certain distance to each of the clusters outside the summarised cluster, an appropriate distance from the summarised cluster to the outside clusters needs to be calculated based on the clusters inside the summarised cluster. In the single linkage case (see Figure 12), for example, the minimal distance found among the inside clusters and an outside cluster is taken to represent the distance between the summarised cluster and the outside cluster. On the other hand, in the average linkage case, the distances of all inside clusters from outside clusters are used to calculate an average distance for the summarised cluster to an outside cluster, and so forth.

All equations are properly formalised on Wikipedia (11.07.2013), but can also be found in Backhaus et al. (2006).

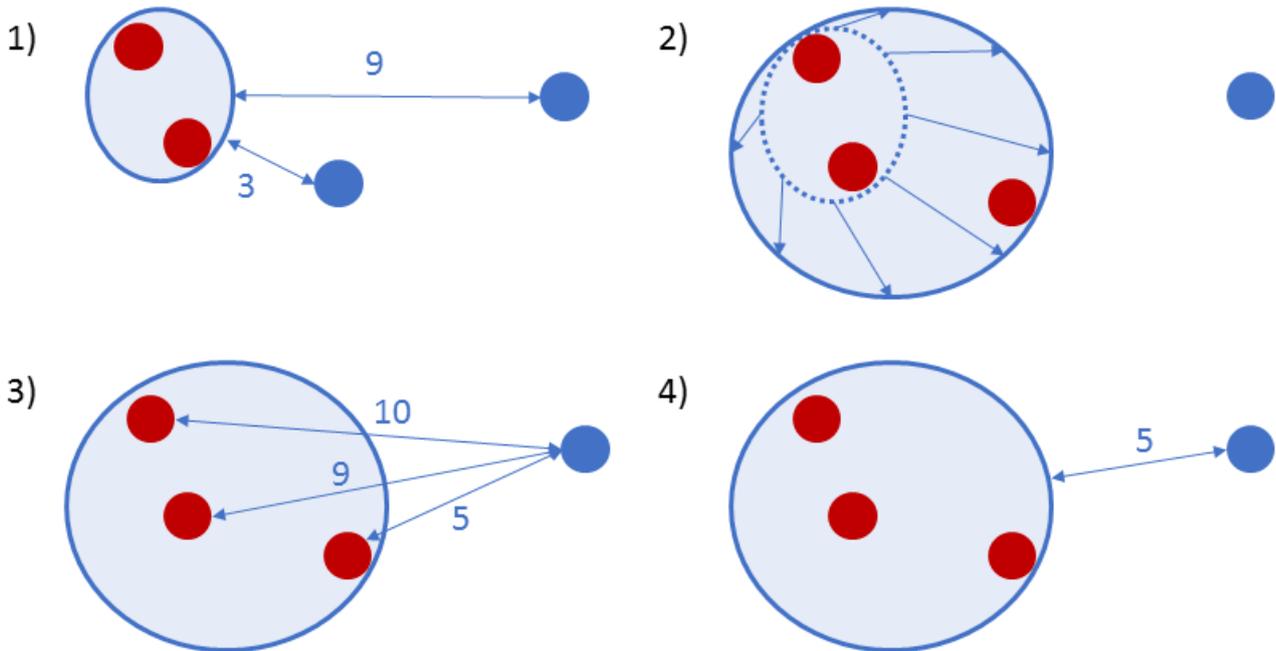


Figure 13: Illustration of single linkage.

There are three clusters, one already containing two smaller clusters. According to 1), the closer cluster of the two outside clusters is the one with distance 3. Therefore, it is added to the initial cluster of two in 2). *Single linkage* dictates that the distance between the large cluster and the remaining outside cluster has to be the *minimal distance* any of the inside clusters has to this outside cluster (3). Therefore, a distance of 5 is added to the distance matrix as the new distance between the large cluster and the remaining outside cluster (4). An example of a distance matrix can be found in Figure 32 a).

2.5 Information Visualisation Process Flow

The first as well as the last step of any retrieval task involves the user. The user types in a text, selects an area on a map, draws a sketch, or chooses an example image, etc. Therefore, the user has to be able to talk to the machine, and the machine has to answer the user by producing results and displaying them appropriately. The concept of a *graphical user interface* (GUI), where the user interacts with windows on a screen using a mouse (and other input devices), has been around for over 4 decades with its introduction by Xerox in 1973. Later on, Apple (1984), as well as Microsoft (Windows 1.0, 1985), adapted this form of interaction with a computer, helping it become a ubiquitous way of *human-computer interaction* (HCI, text based on William Hooper 12.07.2013).

2.5.1 Popular User Interfaces for Image Retrieval

Popular ISEs like Google, Microsoft Bing or Yahoo (see Figure 14) essentially support textual queries (TBIR). Submitting images (query by example, CBIR) is sometimes possible, too. The ordering of the retrieved ranked list of images however is slightly different from usual website search engines (see Figure 14). It usually starts in the upper left corner and proceeds downwards to the lower right corner. Instead of textual descriptions summarising the found images (a common practice in textual search), the ranked list is represented as an assembly of (sometimes differently sized) image thumbnails (small images representing the actual image). Such ordering also corresponds to the way people of Western countries commonly read books and is therefore intuitively understandable for these people. Additionally, all ISEs somehow *highlight* the currently examined thumbnail.



Figure 14: Exemplary popular image search engines on the Web.

2.5.2 User Interfaces in the Context of GIR

Spatially-aware search engines additionally involve some form of the <theme><spatial relationship><location> triplet in their queries. Therefore, a GUI needs to provide facilities to either implicitly (by parsing the textual input query and looking for the triplet actively) or explicitly (by dividing the input fields into the triplet's components) specify such queries. For textual documents, the SPIRIT search engine (Purves et al. 2007) provides split fields (Figure 15 a)). Additionally, as Figure 15 b) shows, users can draw an area on a background map to submit spatial queries without knowing the name of the area. A similar user interface is implemented in Brisaboa et al. (2010), see Figure 16. On retrieval, the results need to be displayed appropriately, taking into account the documents' spatial distribution. To emphasise the spatial context of the documents, they are also linked to a map, showing their actual spatial location (Figure 15 c)). Typically, ISEs require other ways of displaying images compared to the search engines in Purves et al. (2007) and Brisaboa et al. (2010) optimised for the retrieval of websites with spatial references. One ISE concerned with the retrieval and presentation of images in a spatially optimised way was developed during the *Tripod* project (see Sanderson 2009). The GUI presented in Figure 17 positions textual descriptions accompanied by an image on the left side, whereas a map on the right side displays the spatial distribution of the images as dots or circles. The images are densely

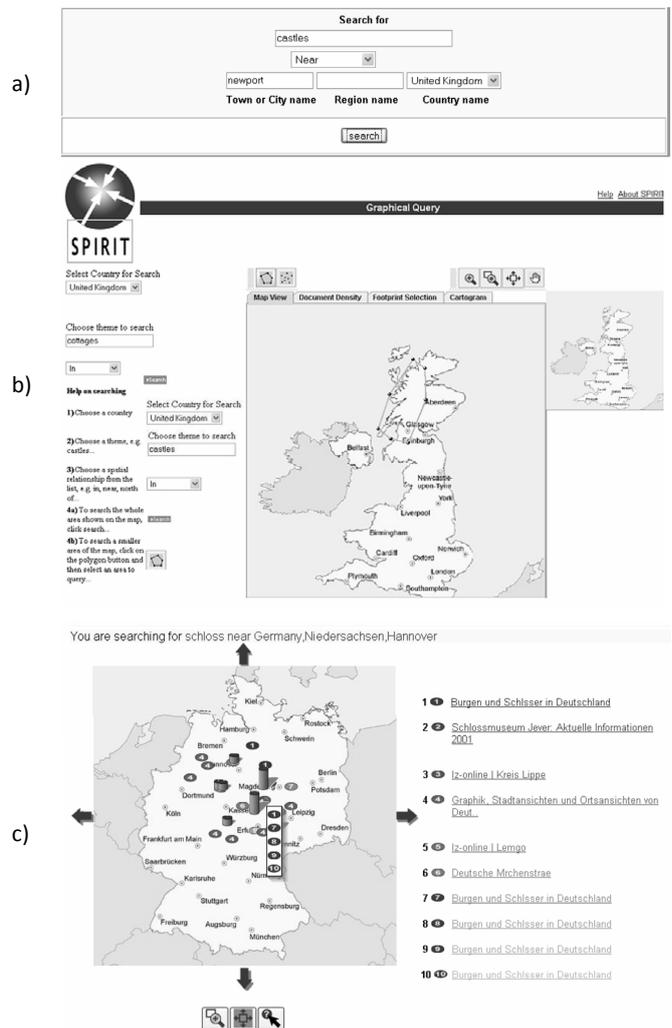


Figure 15: GUI of SPIRIT.

Query/input is shown in a) and b), whereas a result interface is displayed in c).

engines in Purves et al. (2007) and Brisaboa et al. (2010) optimised for the retrieval of websites with spatial references. One ISE concerned with the retrieval and presentation of images in a spatially optimised way was developed during the *Tripod* project (see Sanderson 2009). The GUI presented in Figure 17 positions textual descriptions accompanied by an image on the left side, whereas a map on the right side displays the spatial distribution of the images as dots or circles. The images are densely

covered regions are summarised into larger circles. The latter method corresponds to challenges mentioned in Jones and Purves (2008), where the aggregation of relevant documents, while summarising or filtering duplicate content, is a point to consider when designing user interfaces.

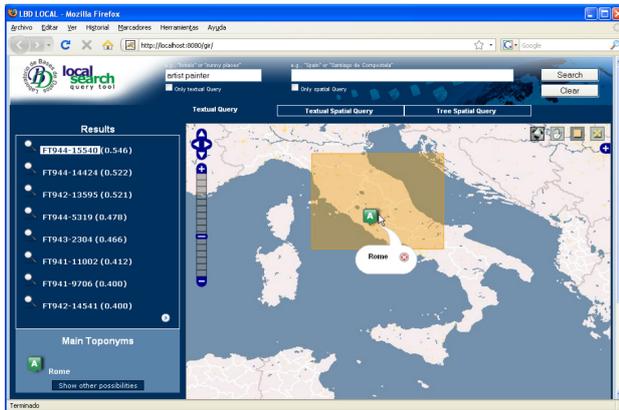


Figure 16: Another example of a GIR GUI.
As seen in Brisaboa et al. (2010).

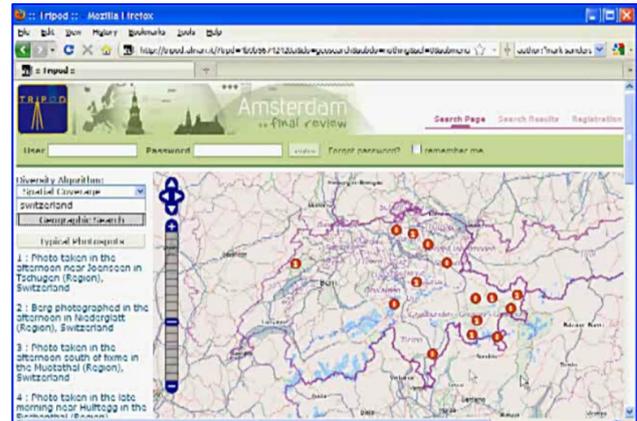


Figure 17: Image Search GUI developed during the Tripod project.

2.5.3 New Interfaces for Image Retrieval

The authors in André et al. (2009) focus on revealing and summarising strategies applied by ISE users to derive new possibilities for searching images and designing ISE interfaces. The most commonly occurring image search behaviour is *rapid browsing* of thumbnails and individual images as well as *enlarging* those images. Informal interviews with 8 researchers and user interaction analyses reveal that image search can be either *exploratory* or *goal-directed*. However, image search typically seems to be more exploratory than web search. Exploration may be needed due to the difficulties of image searchers to appropriately *phrase* their UINs. Furthermore, findings show that users tend to change aims of what they were looking for as a result of what initial searches provided. This means that if users see an image with other features than initially looked for but which catches their attention, they may start looking for similar images. Thus, exploring is more present in image search than in textual search and user interfaces should account for an enhanced exploration experience. Furthermore, facilities to *refine* an initial search are particularly important for goal-directed image search. The main design recommendations can therefore be summarised to:

- 1) Support exploration (rapid browsing, enlarging).
- 2) Fun and aesthetics.
- 3) Goal-directed: query refinement.
- 4) View and save images.

Figure 18 shows some of the interfaces the authors implemented as a result of their findings. They already exhibit rather uncommon ways of presenting and interacting with images. Although the interface in Figure 17 is intended for image search, its result presentation is limited to a textual description and a location presentation possibly more suited for textual document search. However,

images should be displayed in a way that makes them easily accessible and explorable as seen in Figure 18. Unfortunately, none of these prototypes and also no popular ISE provides a map-based interface for spatial queries. Research on an appropriate incorporation of both the technical part of a SPAISE and the visual presentation of results obtained through these techniques is therefore sparse in nature. Although the main aim of this thesis is not to explore new result presentation methods, it is still focused on including updated findings of André et al. (2009) and other authors presented here to enhance retrieval experience for users and to try out different ways of image search and result presentations as well as combinations of methods.

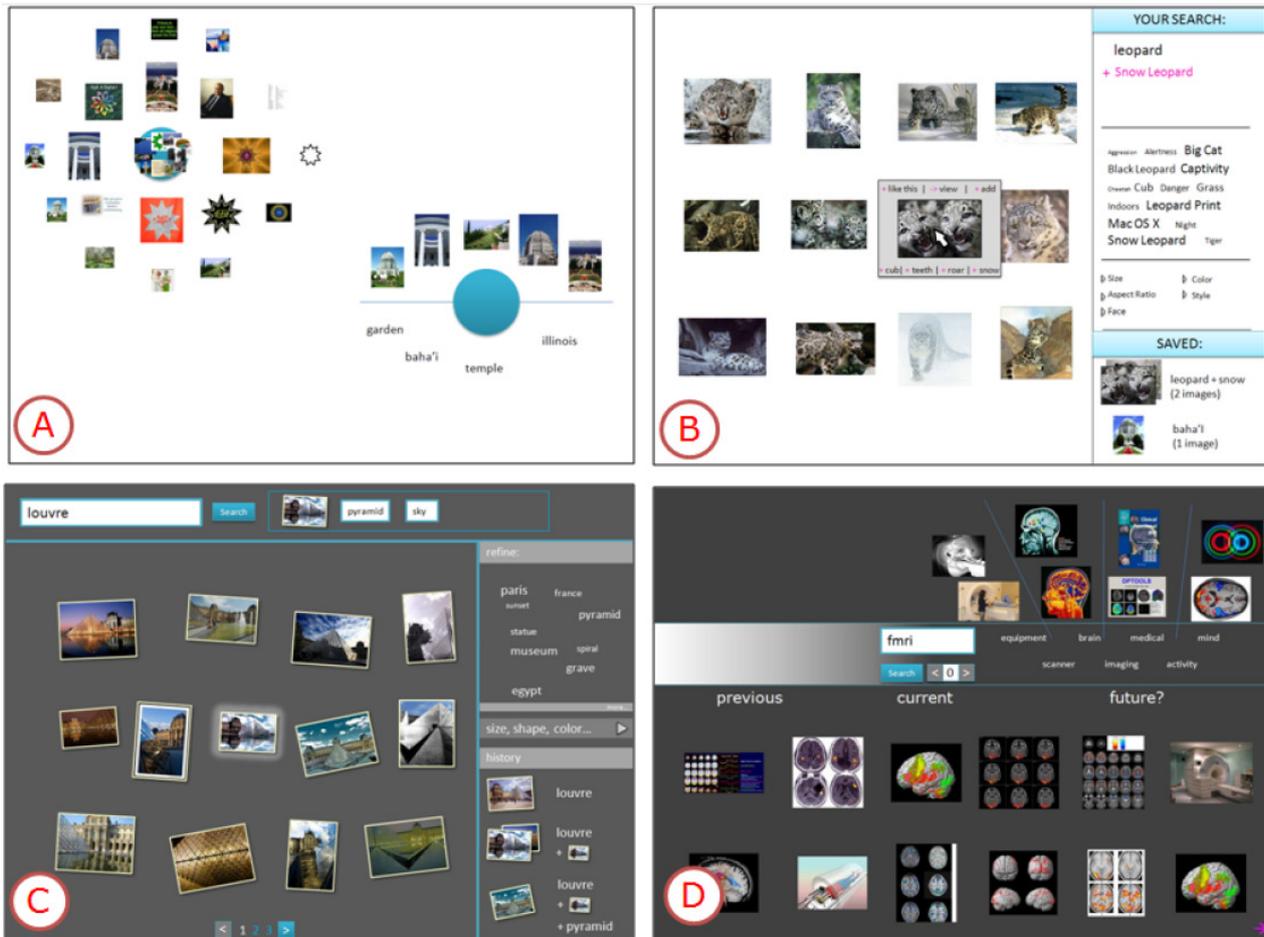


Figure 18: Novel ISE interfaces to enhance user experiences.

2.6 Assessing a Search Engine's Performance

2.6.1 User- and System-Centred Evaluations

Evaluation of search engines is the most crucial part of the search engine development process. If a new method or algorithm is implemented, it needs to be known if it can outperform or is outperformed by existing methods found in the literature. Not only algorithms and system but also users play an important role in evaluating a system. Saracevic (1995) distinguishes 6 levels of evaluation objectives, on an *engineering, input, processing, output, use/user* and *social* level:

- 1) *Engineering level*: handles aspects of technology (computer hardware, networks etc.) to assess issues such as reliability, errors, failures and faults.
- 2) *Input level*: is concerned with assessing inputs and contents of the system to evaluate aspects such as coverage of the document collection.
- 3) *Processing level*: deals with the way inputs are processed to assess aspects such as the performance of algorithms for indexing and retrieval.
- 4) *Output level*: covers interactions with the system and outputs obtained to evaluate aspects like search interactions, feedback and outputs. An example is the evaluation of a system's usability.
- 5) *Use and user level*: Assesses how well the IR system supports people with their searching tasks in a wider context of information seeking behaviour. This includes also the quality of the information returned from the IR system for work tasks.
- 6) *Social level*: Addresses issues of impact on the environment (e.g. within an organisation that uses a search engine) and also includes assessing productivity gains as a result of the introduced system, effects on decision-making, and socio-cognitive relevance.

These levels are not mutually exclusive (Müller et al. 2010). Levels 4 – 6 refer to a more *user-centred evaluation* (UCE), in which users interact with a system within a controlled setting (Carterette and Voorhees 2011). On the other hand, levels 1 – 3 are part of a *system-centred evaluation* (SCE). The main advantage of SCE is that instead of including actual users, they are simulated by an unchanging set of UINs. SCE mainly follows the *Cranfield* paradigm (Cleverdon 1962). The core part is a *test collection*. It encapsulates the experimental environment and is meant to *model* users with realistic UINs (Carterette and Voorhees 2011). Test collections start with a retrieval task (Table 13).

Retrieval task	Description
Ad hoc retrieval	A user wants to find all relevant documents for an arbitrary query.
Filtering¹	A user wants to filter the relevant documents from an incoming stream.
Known-item retrieval²	A user wants to find something that they know exists.
Novel-item retrieval³	A user wants to find new relevant documents.
Diversity retrieval⁴	Different users have different needs for the same query and the system must satisfy them all.

Table 13: Different types of retrieval tasks.

Summarised in Carterette and Voorhees (2011). 1. Robertson and Hull (2000), 2. Beitzel et al. (2003), 3. Harman (2002), 4. Clarke et al. (2009).

Furthermore, they consist of three components (Carterette and Voorhees 2011):

- 1) A *corpus of documents* to search. Such a collection may encompass millions of documents.
- 2) A set of written *UINs*. The ability of a system to satisfy these UINs needs to be evaluated. UINs are typically summarised into *topics*. A topic encompasses a title, a brief description, and a narrative precisely defining the UIN. Around 50 – 150 topics is assumed to be enough for representative evaluations.
- 3) *Relevance Judgements* (RJs) of UINs for each document in the corpus. Relevance is determined by a set of *human judges* and assesses a system's ability to predict a document's relevance to a submitted query. A judge is given a pre-defined topic and each document is manually assessed. The result of this assessment may either be a binary relevance (relevant/irrelevant) or a graded relevance rank (from irrelevant to highly relevant). However, evaluation of large document collections may not be possible due to time and budget constraints. Therefore, an often applied approach to limit the number of documents to evaluate is the *pooling method*: Each topic is submitted to all the systems to evaluate, and only the top-*K* ranked images are pooled together for assessment. Documents retrieved more than once by each system are added only once to the pool of images. Pooling will most likely miss some relevant documents but limits judging efforts to those that are least likely to be irrelevant.

2.6.2 Evaluating a System's Ability to Estimate Relevance

Having retrieved all the RJs for each document, the system's ability to predict relevance needs to be evaluated appropriately. Popular measures are *Precision@n* ($P@n$), *Average Precision* (AP) and *Mean Average Precision* (MAP), and *Normalized Discounted Cumulative Gain* (NDCG). The first two are binary measures both based on *precision*. Precision describes how many of the *retrieved images are relevant*. Besides precision, there also exist *recall* measures. Recall describes how many *relevant images of all the relevant images in the collection* are retrieved (Carterette and Voorhees 2011).

2.6.2.1 Precision and $P@n$

Precision is a binary measure directly estimating how many of the retrieved documents are actually relevant. This is represented as the ratio of *relevant* divided by *all* the retrieved documents. Instead of evaluating all relevant and irrelevant retrieved images, measuring precision at a fixed rank can be undertaken if the assumption is that a user will only choose to examine a fixed number of retrieved results. This also limits the number of documents that need to be evaluated to the first *K* (the top-*K*, e.g. the top-10) documents (pooling approach). The precision is then calculated at this rank and denoted *Precision@n*, $P@n$ or $P(n)$, see Formula XVII (Sanderson 2010).

$$P@n = P(n) = \frac{r(n)}{n}$$

$r(n)$ represents the number of relevant items retrieved in the top n ranks. In the literature, n is usually set to 10, 15 or 20 for initial assessments.

2.6.2.2 AP and MAP

Another measure applied on binary RJs (relevant or irrelevant) and commonly used in the literature is *Average Precision* (Formula XVIII, Harman 1995).

$$\text{XVIII} \quad AP = \frac{\sum_{rn=1}^N (P(rn) \times rel(rn))}{R}$$

N is the overall number of documents retrieved, rn is the rank number, $rel(rn)$ is either 1 or 0, dependent on the relevance of the document at rank rn , $P(rn)$ is the precision measured at rank rn , and R is the total number of *relevant* documents for this particular topic. In other words, this measure calculates the precision at each rank position of each relevant image ($rel(rn) = 1$) and takes the average by dividing the summed-up precision values by the total number of relevant images in the collection. Figure 19 gives an example.

Image ID	H	I	F	L	O	D	Z	W	M	J
Rank	1	2	3	4	5	6	7	8	9	10
RJ	Rel	Rel	Irrel	Rel	Irrel	Irrel	Rel	Rel	Irrel	Rel
Binary	1	1	0	1	0	0	1	1	0	1
Precision	1/1	2/2	2/3	3/4	3/5	3/6	4/7	5/8	5/9	6/10

$$AP = \frac{1 \cdot \frac{1}{1} + 1 \cdot \frac{2}{2} + 0 \cdot \frac{2}{3} + 1 \cdot \frac{3}{4} + 0 \cdot \frac{3}{5} + 0 \cdot \frac{3}{6} + 1 \cdot \frac{4}{7} + 1 \cdot \frac{5}{8} + 0 \cdot \frac{5}{9} + 1 \cdot \frac{6}{10}}{6}$$

$$= \frac{1 + 1 + \frac{3}{4} + \frac{4}{7} + \frac{5}{8} + \frac{6}{10}}{6}$$

$$= 0.7577$$

Figure 19: Example of an AP calculation.

Rank 1 represents the highest and rank 10 the lowest scored images. “Rel/Irrel” correspond to “relevant/irrelevant” and are represented as 1 (relevant) and 0 (irrelevant) in the binary row. Precision is calculated at each rank for calculations of AP.

By taking the average of the APs over all topics, the *Mean Average Precision* can be obtained. *MAP* is one of the most commonly used measures to characterise a system’s *overall performance*, although without statistical tests it can only be seen as an indication for a “better” or “worse” retrieval system.

2.6.2.3 Normalized Discounted Cumulative Gain

Discounted Cumulative Gain (DCG) follows two assumptions (Carterette and Voorhees 2011):

- 1) Highly relevant documents are more useful than marginally relevant documents.
- 2) The lower the ranked position of a relevant document, the less useful it is for the user (because it is less likely to be examined if the result list of images is displayed as a list from most relevant to least relevant).

This measure, originally introduced by Järvelin and Kekäläinen (2002), is sometimes encountered in the evaluation of traditional text-only web search engines but has also been introduced as a measure to evaluate GIRs (Palacio et al. 2011). DCG is defined by a gain function and a discount function:

Gain function. The gain function reflects the value of a particular relevant document to a user. This means that a judge assigns to each document a certain *grade*, not a binary relevant/irrelevant judgement like in P@n or AP. The grade can be any ranking scale, e.g. a four-point scale (0, 1, 2, 3), ranging from irrelevant (0) to highly relevant (3).

$$\text{XIX} \quad g(\text{rel}_k) = \text{rel}_k, \text{rel}_k \in \{0, 1, 2, 3\}$$

Discount function. The discount function represents the patience a user has to proceed down a ranked list. Discounts are assigned to ranks such that they never increase with rank k , and the function is usually logarithmic with base 2 ($\log_2 = ld$, lat. logarithmus dualis, Formula XX).

$$\text{XX} \quad d(k) = \frac{1}{ld(k)}$$

Once the gain function and the discount function are defined, the discounted gain at any rank k can be defined as the ratio of the gain of the document at rank k to the discount of that rank (Formula XXI).

$$\text{XXI} \quad \text{discounted gain @ } k = \frac{g(\text{rel}_k)}{d(k)}$$

A reasonable assumption is to set k to 10, 15 or 20. The **discounted cumulative gain @k** (DCG@k) is then defined as the sum of the discounted gains from ranks 1 to k :

$$\text{XXII} \quad DCG@k = \text{rel}_1 + \sum_{i=2}^k \frac{\text{rel}_i}{ld(i)}$$

The range of DCG depends greatly on the relevant documents known for a topic. If there are many highly relevant documents, DCG can be rather high. This makes averaging DCG over queries problematic, because the best possible performance varies per topic. To address this issue, DCG values are normalised using the *ideal DCG* at the same rank k . The ideal DCG can easily be created by sorting the relevance ranked list from highest rank to lowest rank and then calculating the DCG for this list. Finally, DCG@k values are divided by the ideal DCG@k values to retrieve a comparable number between 0 and 1. Table 14 illustrates NDCG calculations for a list of 10 ranked documents judged on a 0 to 3 relevance scale. Compared to e.g. P@n, NDCG@10 takes into account various degrees of relevance (Sanderson 2010). In P@n, an image is either relevant or it is not. However, consider an image taken inside a church, while a user was looking for an image of a church from outside. Although it is not exactly what the person was looking for, it still may have some relevance compared to an image that shows no church at all (corresponding to the observation of André et al. (2009), that users may not be able to phrase their UIN in the case of images appropriately). P@n cannot account for situations, where nuances in perception decide the relevance of an image. However, P@n and AP are in much wider use than NDCG throughout literature.

Explanation	Calculations
1) 10 ranked documents judged from 0 to 3 (Formula XIX)	{ 3, 2, 3, 0, 0, 1, 2, 2, 3, 0 }
2) Discounted gain (Formula XXI)	$\left\{ 3, \frac{2}{1}, \frac{3}{1.59}, \frac{0}{2}, \frac{0}{2.32}, \frac{1}{2.59}, \frac{2}{2.81}, \frac{2}{3}, \frac{3}{3.17}, \frac{0}{3.32} \right\}$ $=$ { 3, 2, 1.89, 0, 0, 0.39, 0.71, 0.67, 0.95, 0 }
3) DCG at each rank k (Formula XXII)	{ 3, 5, 6.89, 6.89, 6.89, 7.28, 8.66, 9.61, 9.61 }
4) Ideal DCG (perfect ranking)	{ 3, 3, 3, 2, 2, 1, 0, 0, 0 }
5) Ideal DCG at each rank k	{ 3, 6, 7.89, 8.89, 9.75, 10.52, 10.88, 10.88, 10.88, 10.88 }
6) Normalized DCG (NDCG) at each rank k	$\left\{ \frac{3}{3}, \frac{5}{6}, \frac{6.89}{7.89}, \frac{6.89}{8.89}, \frac{6.89}{9.75}, \frac{7.28}{10.52}, \frac{8.66}{10.88}, \frac{9.61}{10.88}, \frac{9.61}{10.88}, \frac{9.61}{10.88} \right\}$ $=$ { 1, 0.83, 0.87, 0.76, 0.71, 0.69, 0.73, 0.8, 0.88, 0.88 }
7) NDCG@10	0.88

Table 14: Example NDCG@10 calculation.

2.6.2.4 Student's paired-samples t-test

For statistical comparisons of any two systems' performances, Student's *t*-test for paired samples is predominantly used throughout the literature. It is a test of statistical significance that compares the mean of two samples by taking into account their variation. If the null hypothesis is supported, the test statistic *t* follows a Student's *t* distribution. If this *t*-statistic surpasses a critical value, two samples are considered to have different mean values and therefore are coming from two different populations. In literature, most often APs of two systems are tested this way where each system's mean value corresponds to the MAP. Formula XXIII shows the two-tailed paired-samples t-test (according to Carterette and Voorhees 2011).

$$\text{XXIII} \quad t = \frac{\hat{\mu}}{\sqrt{\hat{\sigma}^2/n}}$$

Here, $\hat{\mu}$ is the mean of the differences of two sample values and $\hat{\sigma}^2$ is the variance of the differences. The mean is calculated according to Formula XXIV.

$$\text{XXIV} \quad \bar{x} = \frac{1}{n} \cdot \sum_{i=1}^n (x_i)$$

The variance follows Formula XXV. By taking the square root of the variance, the so-called standard deviation can be obtained (which is more intuitive to understand, because it has the same units as the mean, not squared units as the variance). Both variance and standard deviation describe the average scattering of the points around the mean. *n* is the sample size and \bar{x}_i represents the difference of two sample points, e.g. two AP values of two systems.

XXV

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (\bar{x}_i - \hat{\mu})^2$$

An assumption that needs to hold for the t -test is that the samples are normally distributed. However, Hull (1993) stated that the t -test does also perform well even when this assumption is violated. See Sanderson (2010) for a thorough discussion.

2.6.3 Evaluation Goals

Important goals in SCE, besides the evaluation of the performance of search engines, are *repeatability* and *reliability* (Blanco et al. 2011). These two factors assure that achieved results may be re-evaluated by other authors as well. In the Cranfield paradigm, RJs depend on some few human judges that manually assess documents for their topical relevance (Cleverdon 1991). However, gathering RJs is a very time consuming venture and single judges may not be willing to evaluate thousands or even millions of documents on their own. Thus, repeatability may be limited due to the variation of judgements conducted by different experts evaluating the retrieved documents for a task (Harter 1996) and due to the fact that the original judges are often not available for a repetition of the experiments. Recently, stochastic evaluation algorithms have reduced the number of such assessments needed for evaluation, but assessment remains expensive (Lease and Yilmaz 2012). Reliability means that the judges should be expected to produce reliable ground truth (RJs) for evaluation. However, literature shows that RJs of different judges (ranging from judges that know the topic well to those that have no knowledge of the topic) can greatly vary (Bailey et al. 2008). Thus, researchers are looking for adequate ways to create repeatable and reliable, but also fast and cheap evaluation campaigns. Recent advancements have shown that *crowdsourcing* (CS) may provide a solution to these problems (e.g. Urbano et al. 2010, Blanco et al. 2011, Foncubierta-Rodríguez and Müller 2012, Lease and Yilmaz 2012, Sabbata 2013). CS lets a crowd of (presumably) laymen evaluate a topic. Several platforms for CS evaluation have emerged recently, like Amazon's Mechanical Turk (mturk.com) or CrowdFlower (crowdfower.com). The basic idea is that researchers create well-formulated, rather simple tasks. Such tasks are then uploaded onto a website that distributes them to an audience. A person of the audience conducts a task and gets a certain amount of money. Dependent on how complex the task is, more money may be assigned to a task. These rewards are typically very small. Alonso and Mizzaro (2009) for example only pay around 0.02 \$ US-Dollars. Although monetary appeals may attract unfaithful judges that only accomplish tasks to gather money, Alonso and Mizzaro (2009) and Urbano et al. (2010) both suggest that CS is indeed a viable alternative to classical RJs with only few judges, provided the tasks are thought through thoroughly. However, no such CS RJs have ever been collected to evaluate a SPAISE. Due to the spatial dimension intended to be included in topics, exploratory analyses need to identify potential problems. Issues may surface during development of tasks and task descriptions that may have impacts on the relevance assessment of crowd judges. Furthermore, the task design needs to be sophisticated enough to easily discard unfaithful assessors only participating in the evaluation to gather money. However, barely any of the

examined literature shows methods to accomplish this (e.g. Zhu and Carterette 2010). Last but not least, only few papers include CrowdFlower in their CS evaluations (e.g. Sabbata 2013). Most use Mechanical Turk for this task. Therefore, there is much more to explore and contribute to the literature in the field of CS RJ gathering.

2.7 Research Gaps and Research Questions

The state of the art introduced different perspectives on how to implement different kinds of search engines and also on how to adequately evaluate them. However, some gaps could be identified, which will be summarised in the following chapter and synthesised to research questions afterwards.

From a GIR point of view, most research has focused on indexing and retrieving textual internet documents and extracting spatial features from text. Only the Tripod project explicitly focused on the spatial dimension to assess the semantic gap of images. However, the intention was to experiment on possibilities to assign captions to enrich images with spatial information. Therefore, what the GIR literature lacks is a SPAISE that incorporates geometrical/topological image features additionally to the terms assigned to an image. It is an especially valuable research endeavour because it has been proven several times already (e.g. Palacio et al. 2011, Purves et al. 2007) that an additional spatial dimension can improve retrieval performance for queries with spatial content. Also, the effectiveness and applicability of online services like YPM or GN for extracting spatial footprints has not been explicitly evaluated in the context of a SPAISE, therefore leaving the question of their usability in such a setting unanswered.

In the context of CBIR, although many extractable features - global or local - exist, none of them can bridge the semantic gap and each measure is suited best for another retrieval purpose. Furthermore, high computational costs for indexing and retrieval make CBIR not applicable for large image collections. As a consequence, it has been shown that other measures like PRF can improve retrieval performance and applicability of CBIR. The examined literature, however, only focuses on an initial textual query (TBIR), and only this textually relevant result list is then used for re-ranking. An explicit incorporation of spatial features with geometrical spatial relevance assessment measures could not be found. Only Maillot et al. (2007) extract spatial names from text for retrieval purposes but miss the opportunity to treat those textual terms in a spatially geometric or topologic way. However, Kamahara et al. (2012) incorporate both CBIR and geometric spatial methods for the retrieval of identical images taken from different perspectives, but do not provide a textual input. This limits searching capabilities to query by example, which does not correspond to people's common searching behaviour to query by text. Thus, neither could there be found literature examining the possibilities of adding a spatial dimension for spatial image search nor identified any work where PRF was used in combination with a spatial index.

In the context of SPAISE evaluation, traditional evaluation campaigns following the Cranfield paradigm require both human and time resources not necessarily available in the case of smaller research works like a thesis. Crowdsourcing may be a solution to these problems, though only few papers have yet examined such approaches in all their facets. Issues like designing tasks for RJ gathering with unknown assessors and dealing with malicious judges only participating to make a monetary profit need to be solved more thoroughly. The latter points are especially important because Alonso and Mizzaro (2009) could already show that crowdsourced judges may perform as well as normal judges, making them obsolete. Furthermore, the inclusion of an explicit spatial perspective into the evaluation process raises questions about how to formulate UINs appropriately to fully account for the added dimension. However, to be able to create repeatable and reliable SPAISE evaluations, there first of all and most importantly needs to exist an image collection large enough to provide reasonable results. Such a collection could not be identified from the literature.

Aforementioned research gaps lead to the following *research questions* (RQs):

A SPAISE should treat a potential spatial dimension similarly to the textual dimension and additionally include low-level features in an applicable way, so that thousands of images can be searched fast and accurately. For each RQ, a testable *hypothesis* (H) is provided.

Retrieval is the most important part in any search engine. Therefore, RQ₁ focuses on a suitable incorporation of spatial features into a text-only ISE:

RQ₁ *Can an approach combining textual and spatial features outperform a text-only approach for retrieving images of queries with spatial relevance?*

H₁ *A combination of textual and spatial dimensions leads to better retrieval results in the case of images.*

This hypothesis relates to results found in the literature that an explicit distinction between textual and spatial features can increase retrieval performance. This has already been proven for textual documents and websites. RQ₁ therefore investigates a possible verification of the hypothesis in the context of image retrieval.

Moreover, to account for the fact that an image itself is not a textual entity but an array of colour values, a further research interest is the effective incorporation of low-level features for retrieving more relevant images if the initial text-spatial retrieval was noisy. The second RQ is therefore focusing on an appropriate re-ranking of an initially retrieved set of images:

RQ₂ *Can a PRF re-ranking approach, which uses hierarchical clustering and low-level global image features and is applied on a result list retrieved through textual and spatial methods, outperform both text-only and text-spatial-only approaches for retrieving images for spatial queries?*

H₂ *By incorporating low-level global image feature, the retrieval performance of spatial queries can be increased even more than by text- or text-spatial-only retrieval methods because a third relevance dimension, especially important for images, is included.*

This second RQ aims at revealing if CBIR, although researched on heavily in recent years, can actually increase retrieval performance or if the focus of spatially relevant image retrieval should stay on retrieving images only according to their textual and spatial features.

Last but not least, to be able to answer RQ₁ and RQ₂, performance evaluation methods found in the literature may not be enough for the appropriate evaluation of a SPAISE. The identified research gaps lead to the following RQ₃:

RQ₃ *Can relevance judgements gathered through crowdsourcing be combined with traditional evaluation techniques (e.g. P@10) to act as a valuable replacement of human assessors for the evaluation of a SPAISE?*

H₃ *Relevance judgements gathered through crowdsourcing are a viable, quick and inexpensive replacement for known assessors to evaluate a SPAISE using traditional measures, provided certain quality measures are applied.*

The answer of RQ₃ shall cover suggestions on a useful task design, comments on the reliability of judges with implications on how to increase judgment quality as well as an analysis of user comments submitted on task completion.

3 Design and Implementation

3.1 Technology and Design Principles

3.1.1 Development Facilities

Java (version 1.7) is used in this thesis to develop the SPAISE prototype. It inherently provides all the facilities needed to connect to databases like *MySQL* or *PostgreSQL* (using *Java Database Connectivity* JDBC) or to parse XML files. Additionally, there exist many open-source libraries designed especially for Java. Code is written and compiled in the *integrated development environment* (IDE) *Eclipse*.

3.1.2 Design Considerations

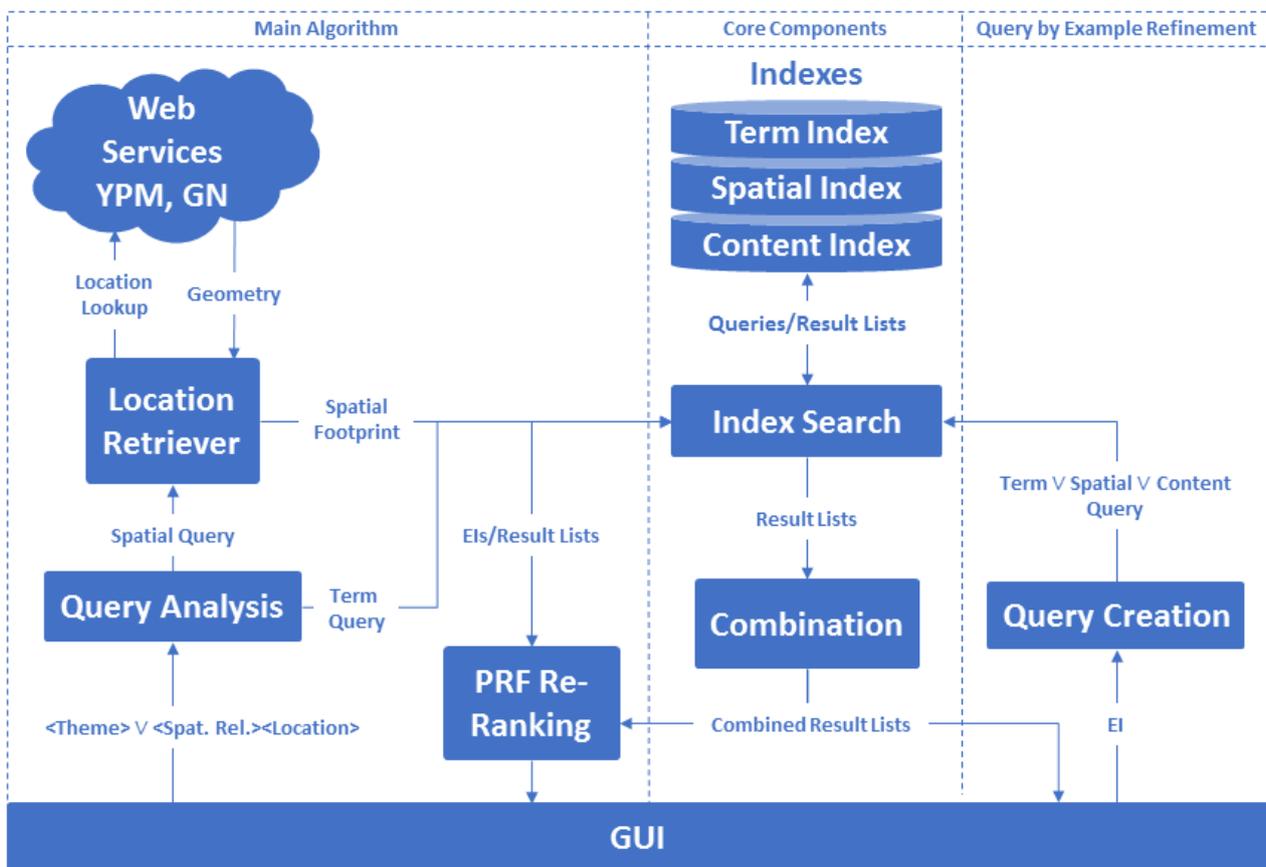


Figure 20: Basic concept and functionalities needed in the intended SPAISE.

Figure 20 gives a conceptual overview of the components needed to build up the SPAISE. The system is designed with extensibility in mind. Therefore, a modular way of implementation is pursued. In Java, it is common to structure code into different *classes*, which are then combined into logically related sets called *packages*. Packages provide a way to split up the code into different *modules*. A module consists of several classes that fulfil only one specified *task* each, and a module combines classes into a system providing one specific *service* for the SPAISE. Primarily, it is focused on choosing clear names for classes that describe the task they accomplish. Wherever possible, *design patterns* (DP) (Gamma 2011) using interface inheritance (subtyping), forwarding (delegation) and template/hook methods are used

instead of implementation-inheritance (sub-classing) (Gruntz 2012). These concepts are also part of the SOLID software design principles (*single responsibility, open-closed, Liskov substitution, interface segregation and dependency inversion*):

- 1) *The single responsibility principle states that every class should have only one single, encapsulated responsibility (Martin 2002).*

Consequently, instead of using one large class having all the functionality, it is focused on implementing many small, rather simple classes. This way, it is easier to adapt the new system to new algorithms.

- 2) *The “Open/Closed Principle” (Meyer 1988) dictates classes to be open for extension, but closed for modification.*

Naturally, prototype implementations can barely follow these principles thoroughly. Some classes or modules may show more, some less interdependency. However, it is tried to avoid as many unwanted dependencies as possible through the use of DPs.

- 3) *The Liskov substitution principle (Liskov and Wing 1994) refers to the fact that an object in a program should be replaceable with instances of their subtypes without altering the correctness of that program.*

This principle is followed through the use of base types instead of subtypes for variables (e.g. `List<T>` instead of `ArrayList<T>`). The advantage of this principle is that base types can be replaced by objects of any subtype if another implementation is needed (e.g. a `LinkedList<T>` or `Vector<T>` instead of an `ArrayList<T>`).

- 4) *The interface segregation principle (Martin 2002) suggests that many (small) client-specific interfaces are better than one (large) general-purpose interface.*

Again, through the use of DPs, this principle can be followed rather easily. However, it is also focused on not having too many small interfaces limiting the overview.

- 5) *The dependency inversion principle (Martin 2002) advises one to depend upon abstractions, not upon concretions.*

This principle actually aims at avoiding the problem of having a class with high-level tasks (e.g. a class that composes lower-level components to an executable algorithm) to be only used in the specific context but not able to be used in any other. It can be accomplished through the use of specific DPs (e.g. Template Method, introduced in a later description) and through the incorporation of *passing* low-level instances to high-level objects instead of *initialising* them directly within the high-level class on creation.

Besides these basic principles of reusable and extensible software design, the integration of existing open-source frameworks is prioritised. Additional online services for spatial lookups and map displays are included as well.

The terms *function* and *method* describe the abilities of a class and are used interchangeably. The same applies to *member* or *instance variables*, which describe a class’s properties (the data it holds).

The description of the system follows a bottom-up approach starting on the lowest level (individual components) and continues to synthesise the system to a high level SPAISE. Therefore, the following description starts with components responsible for creating indexes and continues with components needed to query, retrieve and display result lists of images stored in those indexes. Additional illustrations and theoretical details not introduced in the state-of-the-art chapter are provided.

3.2 SPAISE Components for Indexing Images and Metadata

For an effective retrieval of images based on three different dimensions - text, space and image content - an efficient indexing structure is imperative. The system makes use of separate indexes for each dimension. Although Vaid et al. (2005) showed that this approach can lead to higher retrieval times the modular building allows more flexibility to try out different index structures if intended. Retrieval speed, however, is not the most important point of consideration for this prototypic implementation.

Index structures are only built once and stored in the file system (on the hard disk). Figure 21 gives an overview of classes involved in the index creation process.

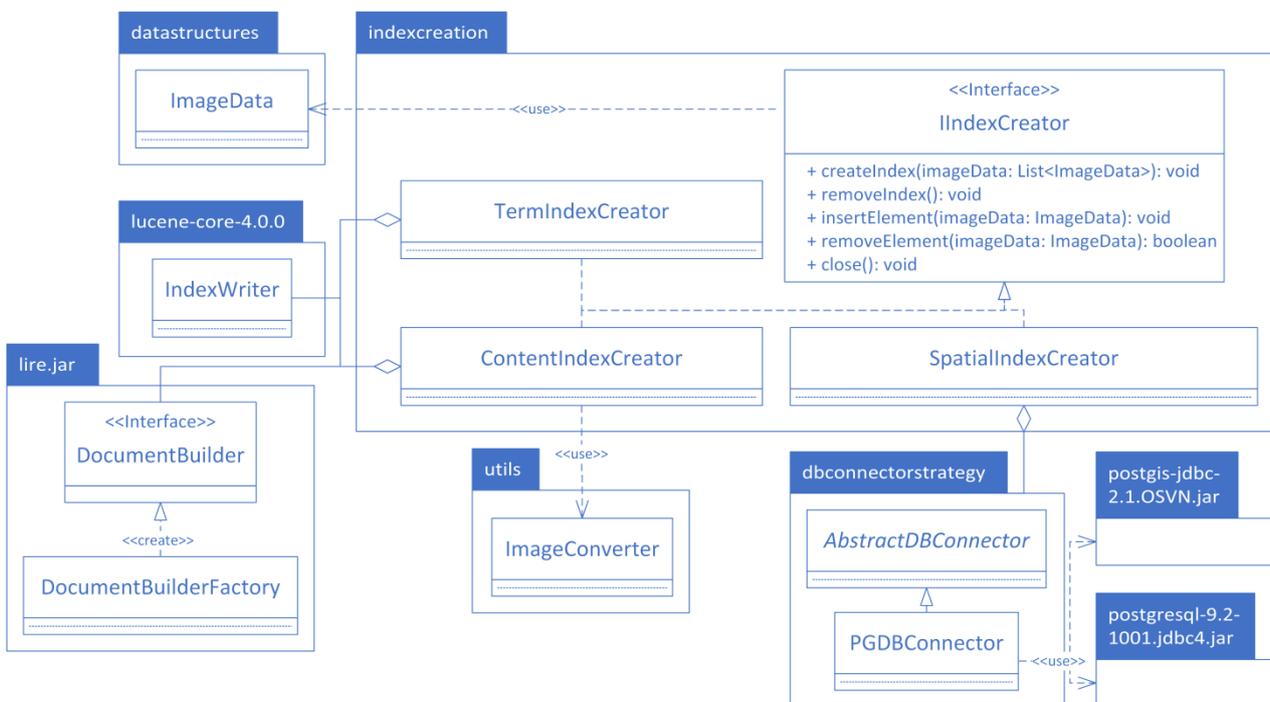


Figure 21: Basic overview of the classes involved in the creation of indexes.

The main classes involved are contained within the indexcreation package: TermIndexCreator, SpatialIndexCreator and ContentIndexCreator. They all implement the Interface

`IIndexCreator` and therefore provide methods for creation and removal of an index and corresponding elements. Any new class intended to create another index structure has to be derived from this `IIndexCreator` interface to be added to the system.

3.2.1 Term Index Implementation

Building up a term index requires several functionalities. To avoid time consuming implementations, a well-maintained and widely used Java-based Library, *Apache Lucene* (or simply *Lucene*, available on lucene.apache.org) assumes responsibility for indexation of terms extracted from image title and description. At the time of writing, Lucene was already available in its version 4.4, showing the high updating rate. It is therefore a valuable choice for term indexing. Even popular pages like Wikipedia (Wikipedia 16.07.2013) use Lucene-based indexing and searching facilities at their core. Lucene builds an inverted index based on a hash table (see 2.3.1 Indexes for Terms and Image Content), allowing very high retrieval speeds. Indexing follows descriptions provided in chapter 2.2.1 Textual Information Extraction. `TermIndexCreator` uses Lucene's `IndexWriter` to build up and insert documents into an index (see Appendix A).

3.2.2 Spatial Index Implementation

Not as simple as indexing terms is the implementation of a spatial index structure. Data structures for storing spatial indexes are commonly implemented in databases as mentioned in 2.3.2 Indexes for Spatial Features. There exist several different implementations of so-called spatial databases, some of which are open-source (e.g. *Oracle MySQL* and *PostgreSQL*), others being closed-source (e.g. *Oracle PL/SQL*). Closed-source implementations are not considered in this thesis. MySQL and PostgreSQL both provide facilities to index spatial data. Both are easy to integrate into Java programs using the JDBC framework. However, PostGIS needs to be added to PostgreSQL explicitly because it does not provide spatial access functions directly, which MySQL does. On the other hand, MySQL only provides R-tree index structures whereas PostgreSQL builds up its R-tree on top of a *Generalized Search Tree* (GiST). A GiST increases search speed when conducting spatial calculations, see PostGIS (26.01.2012) and Hellerstein et al. (04.06.2001). Therefore, and because of the well-tested and mature PostGIS spatial extension, PostgreSQL is favoured over MySQL. Appendix A demonstrates the creation of a spatial index with PostgreSQL/PostGIS.

Geographic indexing in this system is concerned with indexing point coordinates in the *World Geodetic System 1984* (WGS 84) format used by the *Global Positioning System* (GPS). Most digital cameras derive their point locations while taking a picture. WGS 84 coordinates are represented as latitude and longitude and use degrees as units. 0° longitude is the IERS Reference Meridian that passes through Greenwich, England, around 100 meters east of the Greenwich meridian (Paul 25.02.2010, Royal Museums Greenwich 15.08.2005). WGS 84 bounds range from -180° to 180° longitude and from -90 to 90° latitude (Butler et al. 17.07.2013). A detailed description of WGS 84 is given by the European

Organization for the Safety of Air Navigation and Institute of Geodesy and Navigation (12.02.1998). Appendix A shows the insertion of spatial features.

3.2.3 Content Index Implementation

The Java-based LiRE library (Lux and Chatzichristofis 2008) able to extract image features from raster images and to store them in a Lucene index for search and retrieval, is included in the developed SPAISE. The version of LiRE used in this system is 0.93, which indexes the content of images in a Lucene 4.0.0 index. Thus, it uses the same inverted index structure like the term index.

Although LiRE provides extraction and indexing of state-of-of-the-art local features in the form of SIFT and SURF, this system employs visual information retrieval based on global features because of the decreased indexing and computation time. Additionally, as was shown in Arampatzis et al. (2013), local features may not work well with the intended two-stage retrieval process this system is going to employ. However, the ability of the system to replace existing implementations makes it easy to add local features if intended.

Features	MAP	P@10	Error rate	Indexing time (s)
CH	0.450	0.704	0.191	15.27
ACC	0.475	0.725	0.171	413.06
CLD	0.439	0.610	0.309	17.77
EH	0.333	0.500	0.401	20.69
CEDD	0.506	0.710	0.178	47.55
FCTH	0.499	0.703	0.209	60.83
JCD	0.510	0.719	0.177	$\text{Time}_{\text{JCD}} \propto (\text{Time}_{\text{CEDD}}, \text{Time}_{\text{FCTH}})$

Table 15: Performances of different global image descriptors provided by LiRE.
Abbreviations correspond to those given to the global features in Table 5.

To select an appropriate image descriptor, Table 15 is consulted (composed from Lux and Chatzichristofis 2008 and Lux 17.07.2013). Additionally, Dr. Mathias Lux provided help on choosing an appropriate image feature. CEDD, FCTH and JCD show the highest scores in terms of MAP. CEDD and JCD furthermore show almost the lowest error rate, only being undercut by ACC. Although indexing time is relatively high, JCD was finally chosen as the low-level feature to index over ACC, especially because of the much higher assumed indexing times of ACC compared to CEDD and FCTH (on which JCD depends). Documents containing JCD features are created with the `DocumentBuilderFactory` of the LiRE library and stored in a Lucene index created through Lucene's `IndexWriter` (Appendix A).

3.3 SPAISE Components for Image Search, Retrieval and Presentation

3.3.1 Main Retrieval Algorithm

After indexing the images as described in the section before, the system has to be able to retrieve them according to user queries. This process involves formulating a query, submitting it to the system, extracting the features needed from the query to match them against the features stored in the indexes, retrieve and appropriately display the final result list of matching images.

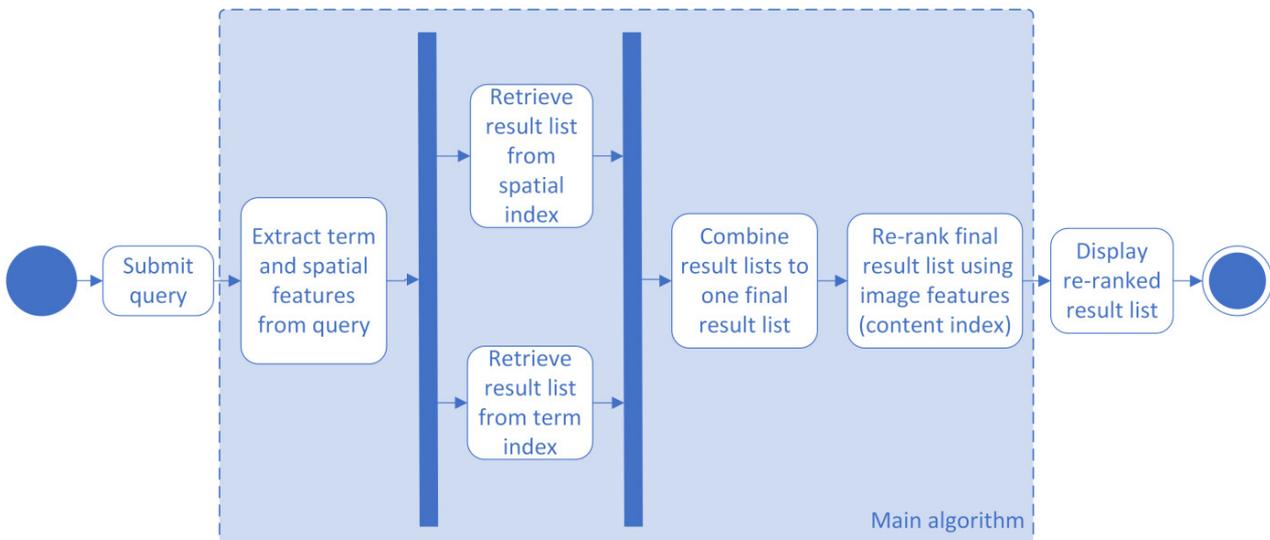
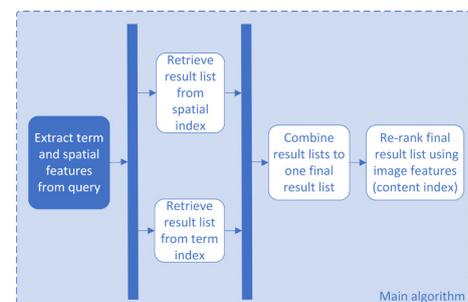


Figure 22: Activity diagram of the main algorithm.

In Figure 22, the main algorithm proposed is displayed in the shaded region. Submitting a query and displaying the final results are part of the user interface and thus will be described thereafter. Each subpart of the main algorithm is analysed in the following section to see which classes interact in what way with each other to retrieve images following the <theme><spatial relationship><location> pattern. For each subchapter describing the main algorithm, a small image of which subpart of the main algorithm is currently being investigated is displayed in dark blue shading in the beginning. An example of how to implement the main algorithm can be found in Appendix B.

3.3.1.1 Extracting Features from a Query

A text query submitted to the search engine will first be separated into a thematic and a spatial part. The thematic part is <theme> in the triplet and is processed exactly same as the indexed terms. The spatial part corresponds to <spatial relationship><location>. To extract features, only <theme> and <location> are needed. <spatial relationship> specifies *how* the spatial query footprints are set into relation to the indexed spatial features, which are exclusively point locations. The chosen spatial similarity function, therefore, depends on the specified spatial relationship. More complex than processing terms of an input query is



the creation of spatial query footprints. The first step is the extraction and disambiguation of the location submitted in the <location> part of the triplet using POS and NER methods. The second step comprises retrieving a geometric representation of the spatial footprint. As demonstrated in chapter 2.2.2.1 Spatial Features, there exist plenty of possibilities to represent spatial features. MBRs are chosen to represent spatial locations in this system. Although being simple representations, they approximate the area already quite well and are also used in many different GIR applications (e.g. Gaio et al. 2008). Although Frontiera et al. (2008) point out the loss of accuracy when using MBRs instead of more accurate representations like convex hulls or more detailed polygons, Cai (2011) observe that the state-of-the-art suggests MBR approximations of spatial footprints to be the favourable choice for computing spatial similarity.

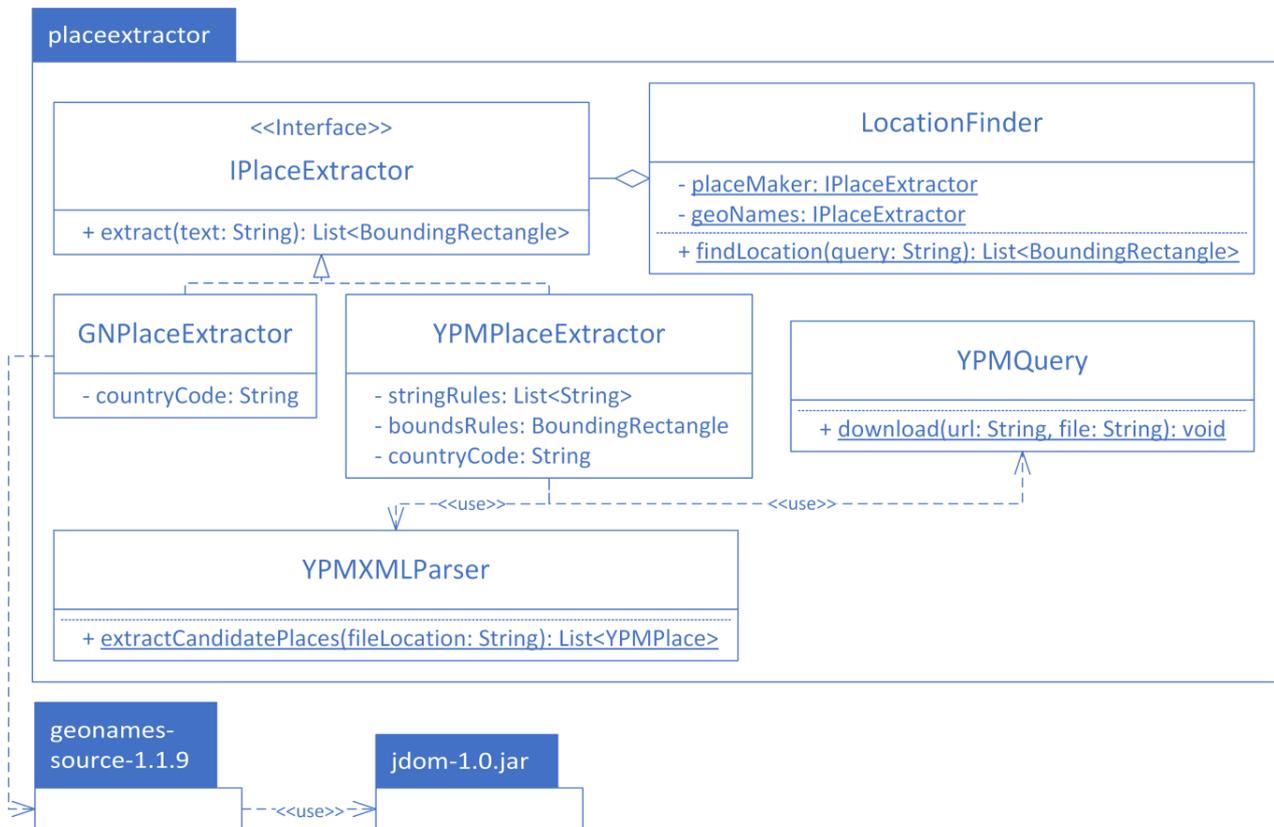


Figure 23: Module for looking up and retrieving a geometric spatial footprint.

Figure 23 shows the module concerned with submitting, receiving and returning a spatial footprint according to a textual location query. LocationFinder holds two different implementations of the interface IPlaceExtractor, GNPlaceExtractor and YPMPlaceExtractor. GNPlaceExtractor uses GN and YPMPlaceExtractor includes YPM for spatial footprint lookup. LocationFinder thus has the purpose of querying and retrieving place names in the following order:

- 1) YPM is queried to retrieve a spatial footprint from the query's <location> part.
- 2) If no location could be found with YPM (no spatial footprint could be retrieved), GN is queried.

- 3) If no location could be found with GN (no spatial footprint could be retrieved), either, no location is returned and the querying process for spatial properties is interrupted.

GN is queried only as a second choice because it cannot return more complex geometrical representations for locations than *points*, whereas YPM is able to retrieve *MBRs*. On the other hand, experiments carried out before implementation showed that for some locations, either GN provided *more* alternatives or YPM did not retrieve *any* place. However, the more accurate representation is favoured over more possible locations.

What both GN and YPM have in common is that a standing online connection is required to be able to query place names, and both systems return XML files containing additional information about the retrieved place (e.g. the country it is located in) besides the actual place geometry. For GN, a Java library is already available to parse the retrieved XML file. None such library exists for YPM. Therefore, two additional classes presented in Figure 23, `YPMQuery` and `YPMXMLParser`, are responsible for handling YPM retrieved XML files. `YPMQuery` sends a location name to the online YPM service and retrieves an XML file (provided by Dr. Palacio, see Appendix C for an YPM XML file snippet). `YPMXMLParser` parses the XML file to extract southwest and northeast coordinates needed to form an MBR. These possibly multiple places are then returned as *candidates* to `YPMPlaceExtractor`. Before the extracted places are returned as an MBR, they need to be validated to assure that the right locations are returned. Several rules can be defined for this purpose:

- 1) **stringRules**: may define a list of place names that should occur in the XML file. If one of the names occurs, a place name is considered valid.
- 2) **boundsRules** may specify an MBR, e.g. the MBR of the United Kingdom, to make sure that retrieved places are located within the UK's extents.

Both rules (1 and 2), neither rules or only one of the two rules can be specified.

- 3) Additionally, a **countryCode** can help both online services already *at query time* to identify the right locations.

Afterwards, the validated southwest and northeast coordinates of a place form the MBR used as spatial query footprint in the further spatial relevance assessment.

Approximating Place Area from Population. If YPM fails to find the specified location name, but GN does, the query footprint will only be a *point*. GN does not provide any area size information to approximate a two-dimensional spatial footprint of a location (e.g. a circle or a rectangle). However, at least the inside relationship requires an area approximation, and it would also be favourable to have such an estimation to derive the relevant extents for the near and directional relationships (e.g. north of). A first simple idea to overcome this problem is to just draw a circular buffer of *constant* radius

around this point. Naturally, this is only a solution in some few exceptional cases. An alternative and to some extent more reasonable way is displayed in Appendix D. GN provides the population size of the place name retrieved. This is the only quantitative value describing anything connected to a place. Therefore, the assumption is:

→ The more people live in a place, the larger its extents have to be.

Of course, such an assumption may also lead to a very rough approximation of the actual area extents, but it is still a better solution than assigning a constant value. As a consequence, a univariate linear regression was calculated for places found throughout the UK, where the area of a place depends on the population retrieved by GN. Formula XXVI should not be considered to be generally applicable, because it is only based on 37 cities and R^2 of the linear regression is very low (only 27%). But as a rough estimation, it may be the best and only way to get an approximation of the area of a point location from the data provided by GN.

$$\text{XXVI} \quad \text{area} = 318.491 * \text{population} + 62685293.12$$

The slope of the regression line is 318.491 and the constant is 62685293.12. The area itself is assumed to be circular because places are considered to have expanded from the centre to the outskirts. Visual inspections of places in maps have shown circular shapes for many cities and villages. Therefore, the radius is calculated from the area using simple circle calculations. Because the system should not need to know if the place was extracted via YPM or GN, an MBR is calculated from the circle. It can be derived by taking the point location retrieved by GN and adding or subtracting the circles radius (because a circle's MBR is a square). Trigonometric formulae are used to convert the radius from meters to degree distances, see Formula XXVII.

$$\text{XXVII} \quad \begin{aligned} \text{latitude}_{NorthEast} &= \text{latitude}_{GN} + \left(\frac{180}{\pi} * \frac{\text{radius}}{\text{radius}_{Earth}} \right) \\ \text{longitude}_{NorthEast} &= \text{longitude}_{GN} + \left(\frac{180}{\pi} * \frac{\text{radius}}{\cos(\text{latitude}_{GN})} \right) \end{aligned}$$

$$\begin{aligned} \text{latitude}_{SouthWest} &= \text{latitude}_{GN} - \left(\frac{180}{\pi} * \frac{\text{radius}}{\text{radius}_{Earth}} \right) \\ \text{longitude}_{SouthWest} &= \text{longitude}_{GN} - \left(\frac{180}{\pi} * \frac{\text{radius}}{\cos(\text{latitude}_{GN})} \right) \end{aligned}$$

Formula XXVII contains several formulae derived and tested from Stéphane (17.07.2013). In these formulae, radius_{Earth} is the semi-major axis of the WGS 84 ellipsoid representing the radius of the earth at the equator (≈ 6378137 meters). radius is the circular extent of a city, and latitude_{GN} and

$longitude_{GN}$, respectively, are the coordinates of the centroid of the location retrieved by GN. With the four formulae, the south west and north east locations of the MBR can be calculated.

3.3.1.2 Retrieving Result Lists

The previous step extracted term features and spatial footprints from a textual input query. The following step comprises querying the corresponding indexes with those items. Figure 24 shows the two core components for conducting image searches in the form of the <theme><spatial relationship> <location> triplet. Note that both indexes are queried separately, using one of the specified `IIndexSearcher` implementations (`TermIndexSearcher` for term queries and `SpatialIndexSearcher` for spatial queries). Both classes will be introduced in the following section.

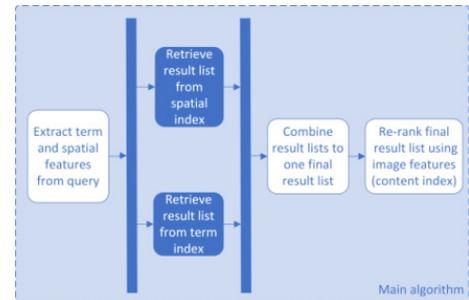


Figure 24: Core index search facilities for term and spatial queries.

3.3.1.2.1 Retrieving Result List from Term Index

Figure 25 illustrates the classes involved in retrieving images based on terms. `TermIndexSearcher` contains an `IndexSearcher` and a `QueryParserBase`, both classes provided by the Lucene library. `IndexSearcher` performs the actual searches on the Lucene index and uses a proper similarity function for query matching. The function has to be the one already used on indexing an image’s texts. In this configuration, `DefaultSimilarity`, a direct subclass of `TFIDFSimilarity` (Lucene 4.0.0 API 2012) is used. It combines the *Boolean model* (BM) with the *Vector Space Model* (VSM) of Information Retrieval, which means that documents first need to be approved by BM before they are scored using VSM. Therefore, only images where at least one of the terms (Boolean OR) or all terms (Boolean AND) occur are retrieved. The similarity measure used is a refined cosine similarity (see 2.4.2.1 Textual Relevance and Similarity). Each implementation of `IIndexSearcher` needs to have its corresponding derivation of `AbstractIndexQuery`. This class acts similarly to a visitor in the *visitor* DP (Gamma 2011): when submitted through the `queryIndex()` method of the specified `IIndexSearcher`, its `scoreList` is filled with the corresponding result list containing scores and identifiers of images for this term query. Furthermore, `TermIndexQuery` holds a character string representing the textual query submitted to the system. An additional `Operator` specifies whether the Boolean OR or AND

should be applied to the query as explained before. The query itself is always conducted on both title and description of an image using an instance of `MultiFieldQueryParser`.

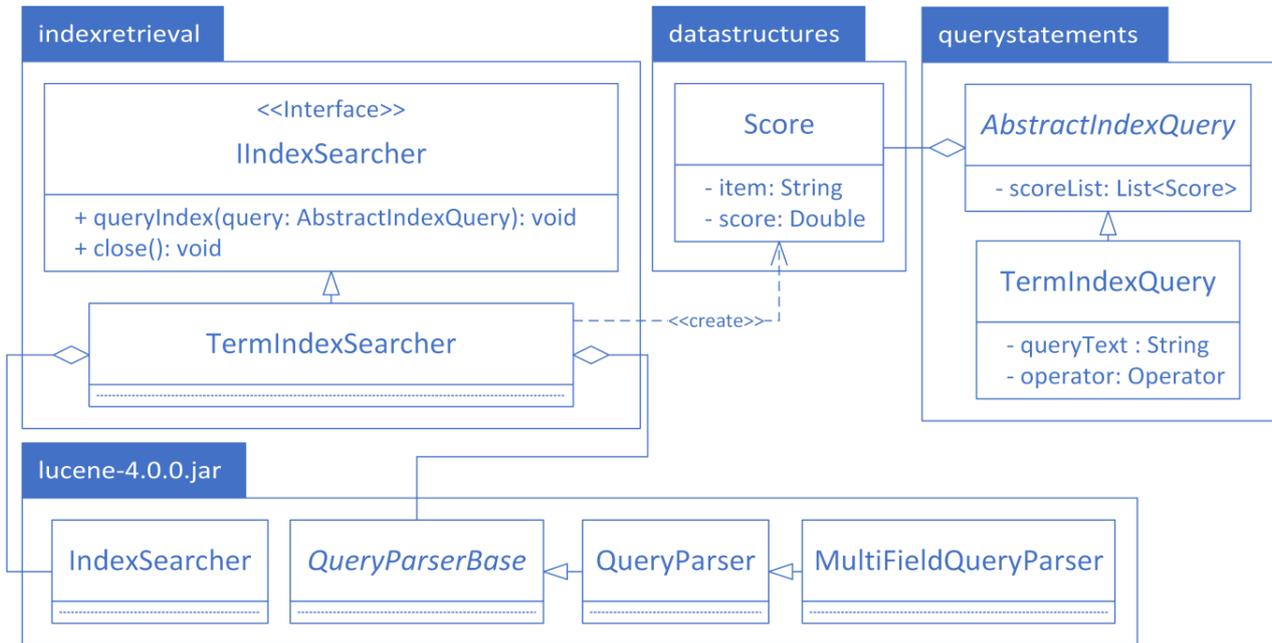


Figure 25: Overview of packages and classes involved in querying the term index.

3.3.1.2.2 Retrieving Result List from Spatial Index

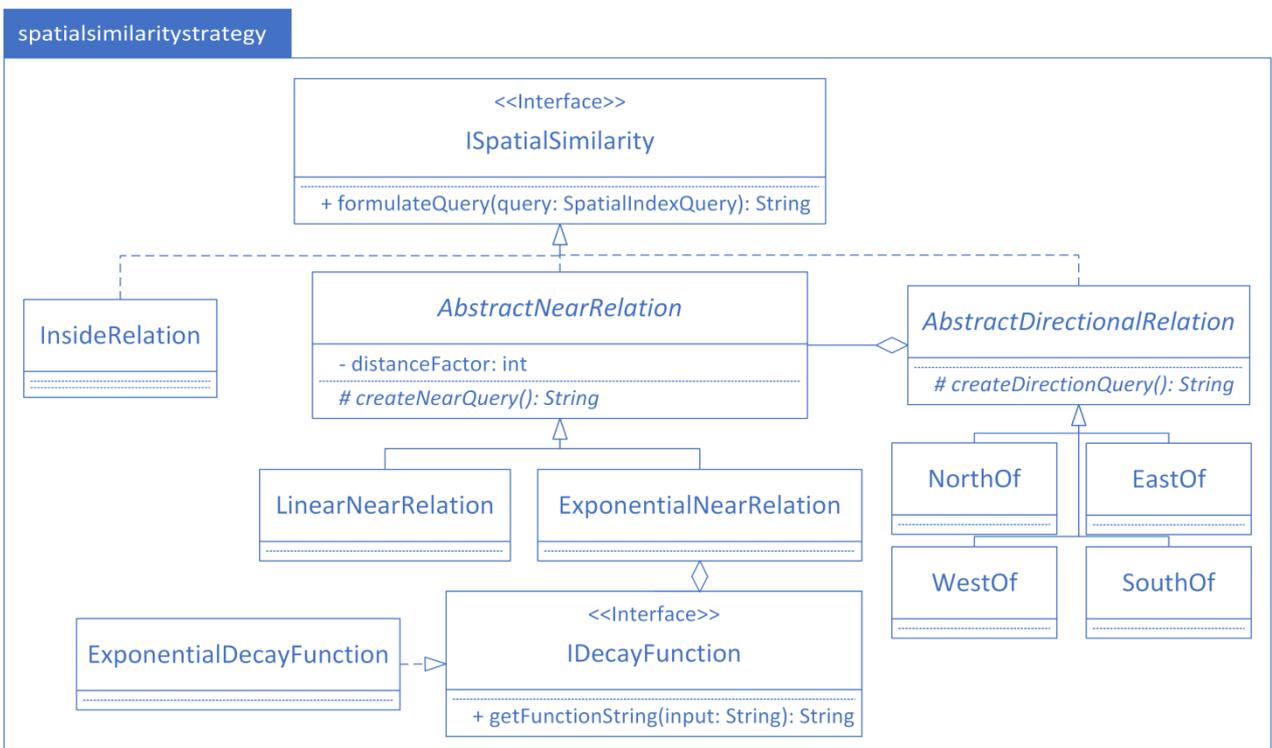


Figure 26: Overview of spatial similarity functions.

Implemented relationships in this system follow mainly the approaches presented in 2.4.2.2 Geographical Relevance and Similarity. The methods introduced there are reasonable choices to assess spatial similarity. More elaborated techniques, e.g. probabilistic approaches (Frontiera et al. 2008), are

not favoured because they need training data beforehand. A tile-based approach optimised for documents containing a possible set of spatial references as shown in Palacio et al. (2012) is not applicable, either because only one location is indexed in the case of images here. Furthermore, the simple representations of locations as points make geometric and topologic approaches more than sufficient solutions. The spatial similarity functions have to be implemented one by one and an overview can be seen in Figure 26.

The modular building of these functions makes it very easy to add new similarity measures, if intended. For example, a relationship like `NorthEastOf` could be added by extending the `AbstractDirectionalRelation` and implementing the provided hook methods. Hook methods (i.e. `createNearQuery()` or `createDirectionQuery()`) provide a convenient way to ensure that implementations in the base class do not have to be changed. Only the hook method needs to be implemented in a subclass. This procedure corresponds to the *Template Method* DP (Gamma 2011). It makes sure that the ordering of the algorithm in the base class, meaning, *when* a hook method is called *within* the template method (`ISpatialSimilarity`'s `formulateQuery()` is the template method), stays the same throughout all subclasses. Subclasses adapt these hook methods to their specifications. This is especially useful in the case of different directional relationships, which have almost everything in common except some minor calculations not simply adaptable by passing different parameters.

Another important DP used here is the *Strategy* (Gamma 2011), as the name of the package already indicates. The pattern is most suitable for this task, because many different algorithms with similar behaviour need to be implemented, but only one at a time is used to retrieve scores. Additionally, it is very simple to add a new spatial similarity measure to the system by implementing the `ISpatialSimilarity` interface and specifying its `formulateQuery()` method. Strategies are used throughout this SPAISE and will be encountered in different parts again, usually indicated by the "strategy" name convention of the packages they are contained in.

In the following part, theory from chapter 2.4.2.2 Geographical Relevance and Similarity is taken up again and refined to implement spatial similarity measures.

InsideRelation. `InsideRelation` implements a binary operator, where a point location of an image is either inside (score = 1) or outside (score = 0) the area retrieved by YPM (an MBR) or GN (an approximated MBR based on the population of the retrieved place).

ExponentialNearRelation. `ExponentialNearRelation` extends the hook method `createNearQuery()` of `AbstractNearRelation` by implementing the function introduced in Formula VII. However, it was only used in experiments due to its high computational requirements. Implementation details can be found in Appendix E.

LinearNearRelation. LinearNearRelation implements AbstractNearRelation's createNearQuery() using the linear function formulated in Formula IX. The main advantage of this implementation over an exponential function is its simplicity, which also decreases retrieval times noticeably. It describes the concept of near proportionally to the MBR's half diagonal of the query location found by either YPM or GN. The maximum score of 1.0 is achieved at the MBR's centroid. It decreases linearly to the borders of the circle with a radius of the half diagonal of the MBR. The radius can additionally be multiplied by a distance factor, which increases or decreases the range considered relevant to the query. Thus, it describes a circular near relationship, see Figure 27.

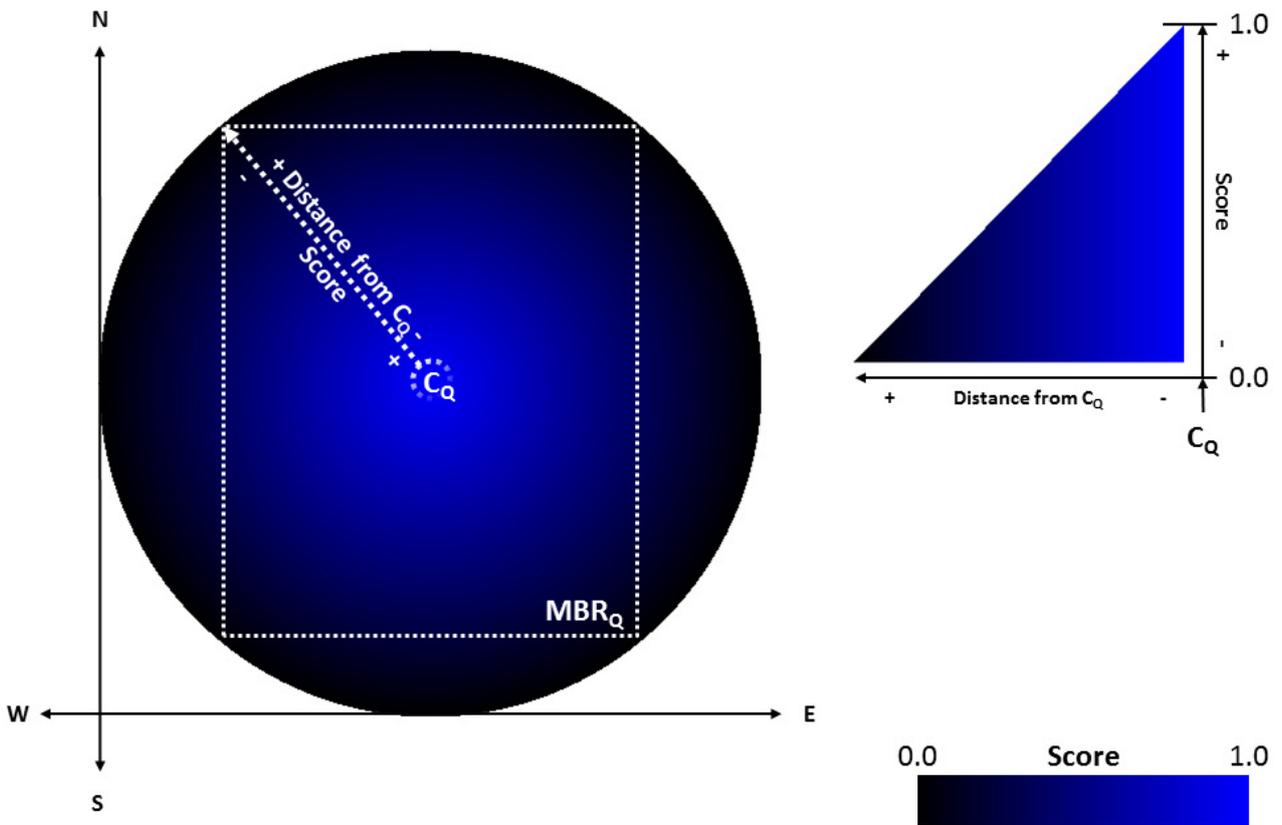


Figure 27: Illustration of the linear near relationship.

C_Q represents the Query footprints centroid, MBR_Q is the query footprint. The left image illustrates the two-dimensional score distribution in the geographic space. The right figure represents the *linear* decrease from C_Q to its maximum extents. The distance factor in this case equals 1, because the circle's radius corresponds exactly to the half diagonal of the MBR_Q .

NorthOf/EastOf/WestOf/SouthOf. Formula XXVIII is an adaption of Formula X. It is slightly altered as shall be explained hereafter.

$$XXVIII \quad angular - score(a, b) = \begin{cases} 1 - \frac{\alpha}{45^\circ} & \text{if } \alpha < 45^\circ \\ 0 & \text{if } \alpha \geq 45^\circ \end{cases}$$

The angle (α) points to the left and right side of the angular direction. This means that if α is 0° , the score is 1.0 (see Figure 28 a). Point a for these calculations is the centroid of the query footprint retrieved by either YPM or GN and represented in the illustration as C_Q , and b corresponds to a location of an indexed image. Compared to Formula X, the largest opening of the angle is 90° , not 180°

anymore. Although it is semantically not 100% correct to do so, because everything above the middle line of the MBR would be considered somehow to be in the north or to the north of a location, an overlapping seemed not correct either (if all directional relationship have a cone of 180°, then each half of the cone would either be one direction, e.g. north of, or the other direction, either west of or east of (seen from north of). Therefore, non-overlapping cones are implemented instead. In Figure 28 a), the north axis (representing the north direction from the query point a at its origin) has the highest score value 1.0. With increasing angle to each side, the achievable score of point b is decreased. At 45° to each side, the score value is 0.0. Therefore, any angle larger than or equal to 45° is not considered relevant anymore. This means that an image lying on the exact north axis seen from a query location a takes the value 1.0, and all the points on (hypothetical) northeast and northwest axes (or further) receive the score 0.0. Besides the directional component, Tobler’s first law of geography (see chapter 2.4.2.2 Geographical Relevance and Similarity) is incorporated into the equation to provide reasonable restriction to what is considered north of (e.g. an image found in Glasgow would be still considered north of London, but reasonably, a person looking for images would not consider this to be relevant to the query). The visual representation of the near relationship can be found in Figure 28 b). Figure 28 c) finally shows the score distribution after multiplying both the scores of directional and near relationships. A consequence hereof is that point b with distance d_1 from a may have a higher score in the end than point c with distance d_2 , although point c lies more to the north of a than b , just because c is *closer* to a than b . S/W/E of relationships are implemented accordingly.

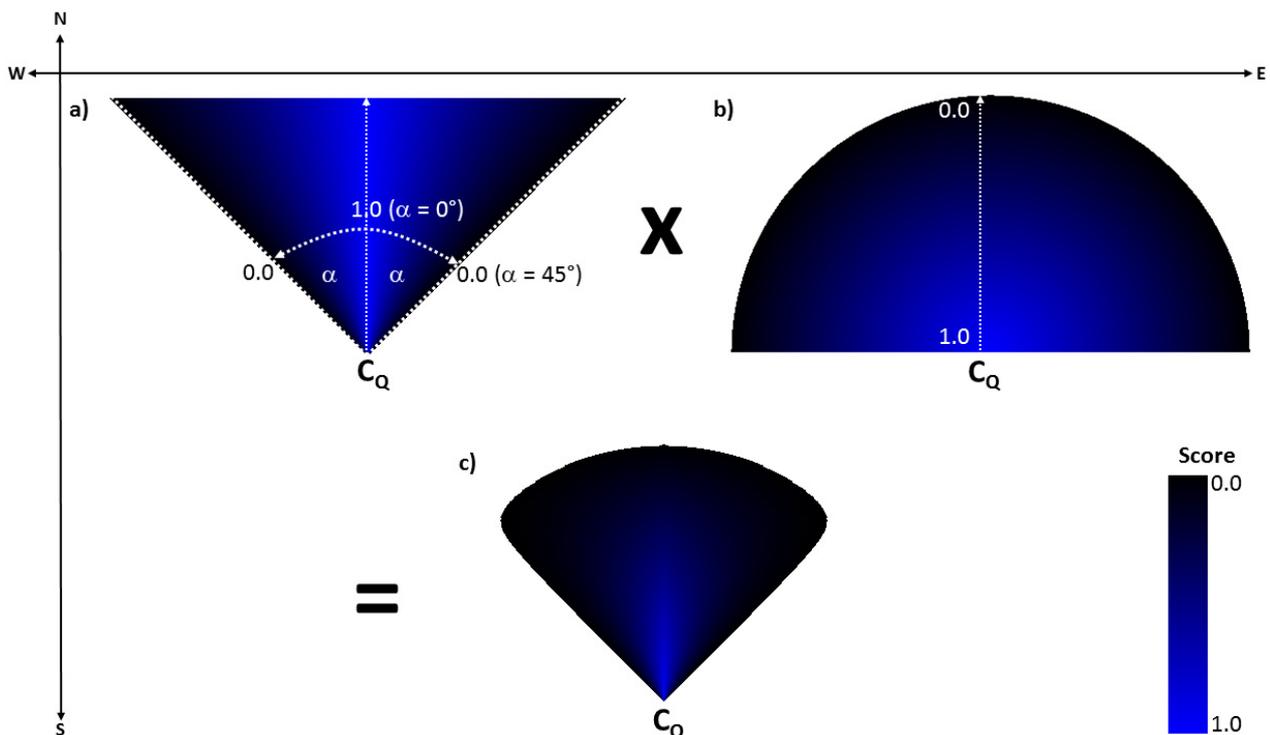


Figure 28: Visualisation of the “north of” spatial relationship.

Multiplying a) (north of) with b) (near) results in the cone shown in c) with decreasing score value from the query footprint’s centroid (C_Q). As shown in Figure 27, the near relationship in b) depends on the query’s MBR extents (not shown), effectively restricting the distance to the north of C_Q still considered being in the relevant range.

In the `spatialsimilaritystrategy` package, `AbstractDirectionalRelation` is the base class for all the directional relationships. It contains a method `createDirectionQuery()`, which has to be implemented by the derived classes `NorthOf`, `SouthOf`, `EastOf`, and `WestOf`. This method acts as a hook method of the *Template* pattern as already described for the `AbstractNearRelation`. The final implementation multiplies the directional relationship with the `LinearNearRelation` to retrieve the final score as shown in Figure 28 c).

Spatial Retrieval. The advantages of designing similarity measures with the Strategy DP become especially prominent when `SpatialScoreRetrieval` comes into play in Figure 29. `SpatialIndexQuery` has an object of type `ISpatialSimilarity` added on query creation. This way, each query has its specified similarity measure, allowing the system to actively change retrieval strategy without even knowing the current spatial relationship. Any new implementation of a spatial similarity measure could be provided without changing any of the existing code. Database communication is provided through `AbstractDBConnector`. For Retrieval, `SpatialIndexSearcher` makes use of `SpatialScoreRetrieval`, which accesses PostGIS functionalities for similarity assessment. SQL query strings generated by any implementation of `ISpatialSimilarity` are submitted to this class. `SpatialIndexSearcher` accepts only one specific type of `AbstractIndexQuery`: `SpatialIndexQuery`. It holds query-specific data about the used spatial relationship, the query place name and footprint (MBR) in WGS 84 coordinates derived therefrom.

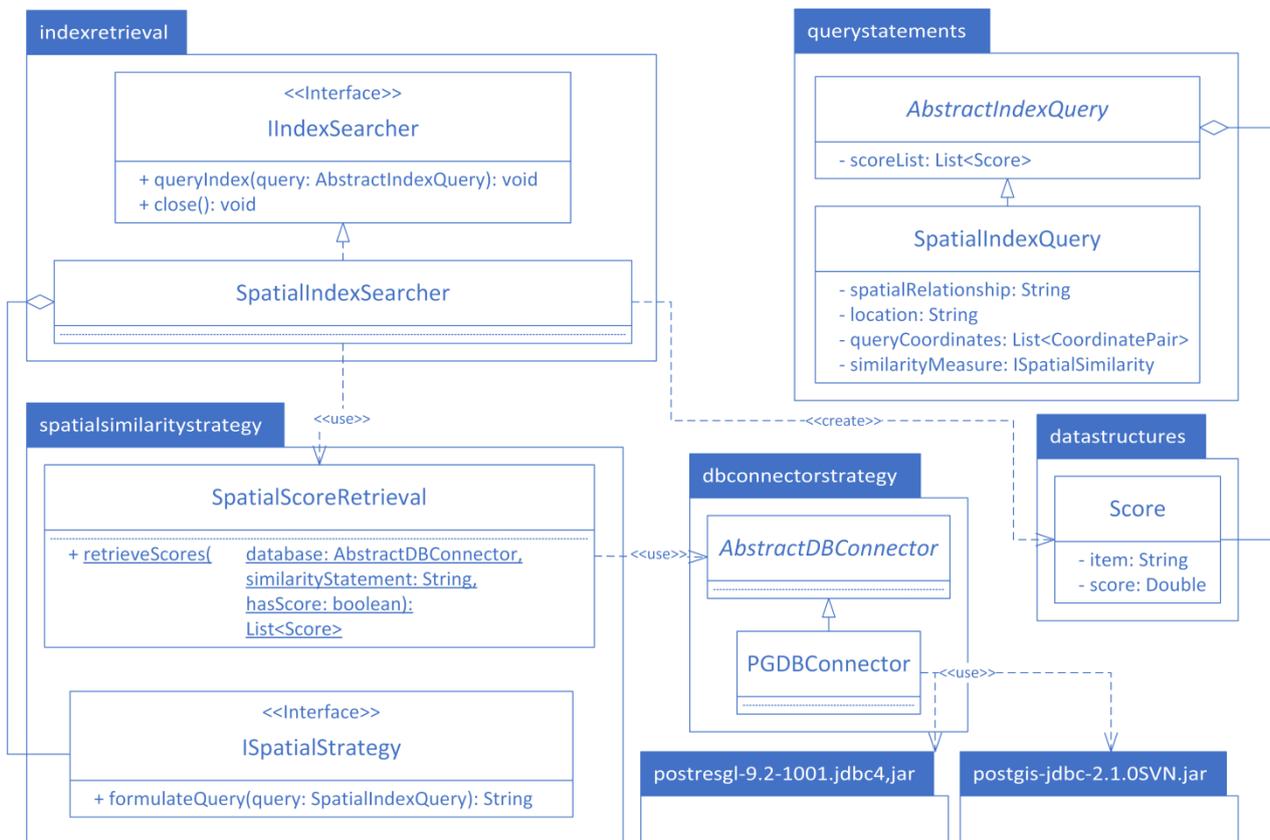
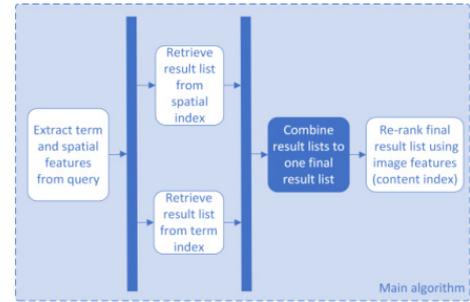


Figure 29: Overview of the classes needed for processing spatial queries.

3.3.1.3 Combining Result Lists

The third part of the main algorithm aims at combining the retrieved term and spatial result lists into a single list. Figure 30 shows all the classes involved in combining initially retrieved score lists from different dimensions. Fusion of an arbitrary number of initial score lists is divided into two packages.



In `scorecombinationstrategy`, `ScoreCombinationBuilder`

is concerned with creating a data structure `ScoreCombination` that holds the score list retrieved for every dimension (e.g. for spatial and term dimensions). Again, the Strategy DP provides the framework for creating different combination strategies by implementing the `ISCBuilder` interface. `UnionSCBuilder` builds up `ScoreCombination` objects that *do not need* to have a score in all the possible dimensions. All images having *at least one* score in one dimension are preserved. On the other hand, `IntersectionSCBuilder` makes sure that only those images are retrieved where a score could be assigned to *each* dimension. The first makes sense if e.g. there are only few matching images to a query in a collection and one wants to retrieve any image that has at least slight reference to the query. The latter is a stronger retrieval strategy, which assures that all dimensions of the query are, at least to some extent, considered in the final result list. This makes sense in large collections of many thousands of documents, where most likely several relevant hits are retrieved.

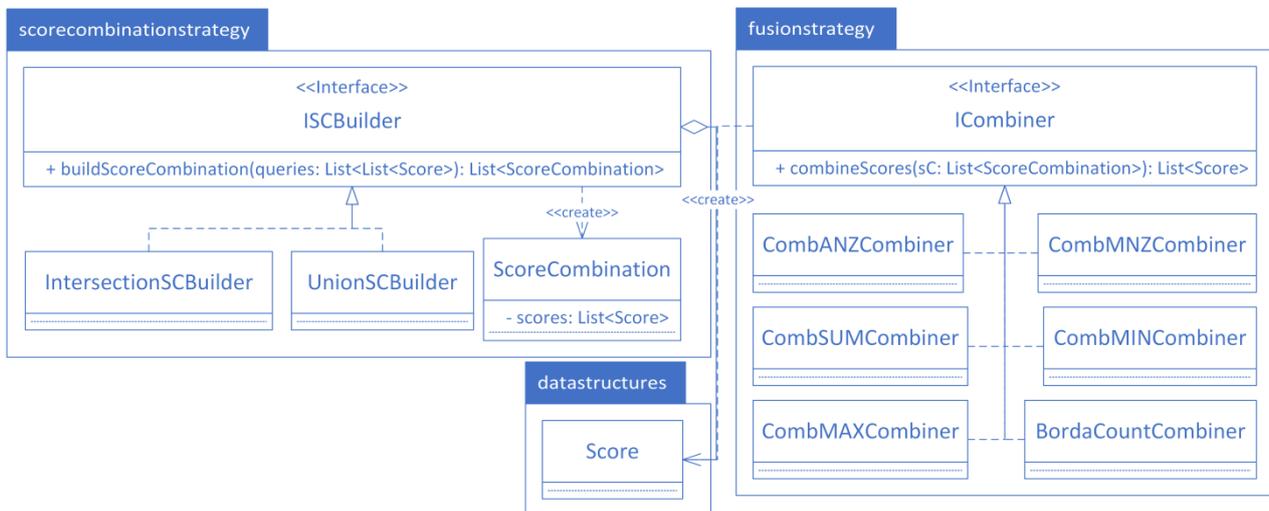
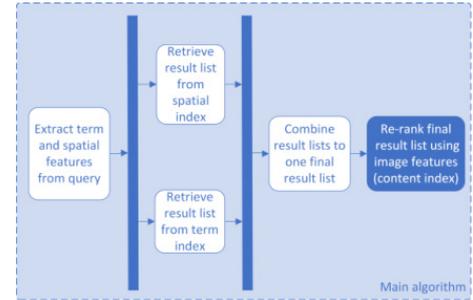


Figure 30: Classes concerned with fusing possibly various scores into a single score.

The package `fusionstrategy` combines these score combinations properly. The strategy interface `ICombiner` provides a method `combineScores()` to implement the desired way in which the beforehand gathered, possibly multiple scores of an image, should be fused. Various combination strategies introduced in 2.4.3.1 Result List Fusion are implemented for experimental reasons.

3.3.1.4 Re-Ranking Final Result List with Image Features

The last step of the main algorithm covers re-ranking of the combined result list consisting of term and spatial scores using a third relevance assessment based on the images' low-level features. The extracted features used in this implementation are JCD. Lux and Chatzichristofis (2008) implement the Tanimoto coefficient as similarity measure for the CEDD (Chatzichristofis and Boutalis 2008a) and FCTH (Chatzichristofis and Boutalis



2008b) features on which JCD is actually based (JCD fuses the result lists retrieved from CEDD and FCTH searches in the end. Therefore, the similarity measure is applied twice, once for CEDD and once for FCTH). How can these low-level features now be used to adequately implement a re-ranking strategy able to improve a possibly noisy initial result list based on only term- and spatial dimensions?

3.3.1.4.1 Basic Re-Ranking Algorithm

Arampatzis et al. (2013) describe a way of implementing a re-ranking algorithm in Formula XXIX.

$$\text{XXIX} \quad \text{rerankedscore} = \max_i JCD_i$$

The index i runs over the K *example images* (EI) selected for re-ranking. For choosing a suitable number K of initially highest ranked images, either a static number (3 in Maillot et al. 2007), a percentage (30% in Popescu et al. 2009), or a dynamic estimation (Arampatzis et al. 2009, employed in Arampatzis et al. 2013) is suggested. Each of the chosen K images is used to re-rank the initial result list according to the similarity of the subset images to the K EIs. If K equals 10, the subset is re-ranked ten times, resulting in ten differently ranked lists, all containing the same images as the initial list. However, the ranking is based on the similarity between the low-level features of the K EIs and the low-level features of all the images in the initially retrieved list. Logically, for each list, the first image has to be the same as the EI with score 1.0 because it actually *is* the same image (EIs were taken from the initial list). As Formula XXIX suggests, for every image of the initial list, its maximum score found in *any* of the K score lists is finally assigned to the image. The set of images with a now new, re-assigned score is then sorted in decreasing order. As a consequence, the first K images used for re-ranking will also achieve the maximum score of 1.0. If the ten first images are taken, these ten images will still be the top-10, all having the same maximum score of 1.0. Only images on ranks higher than K may change. But what if the first K images of the initial result list are already contaminated with noisy, irrelevant results?

3.3.1.4.2 Reducing Noise through Clustering

To reduce noise introduced by an initial result list (a combination of term and spatial scores) and to make re-ranking as described above more robust to images that possibly do not show the topic desired by a user, a classification using hierarchical clustering according to Agglomerative Hierarchical

Clustering of Images is conducted *before* the actual re-ranking algorithm (Formula XXIX) is applied to the initial result list. Main clustering functionality is provided by Lars Behnke (25.07.2013). Figure 31 gives an overview of the basic steps of the re-ranking algorithm, which will be explained in the next section. Although such methods have been used thoroughly in CBIR, there was not found a work that combined the approaches exactly the same way as presented here.

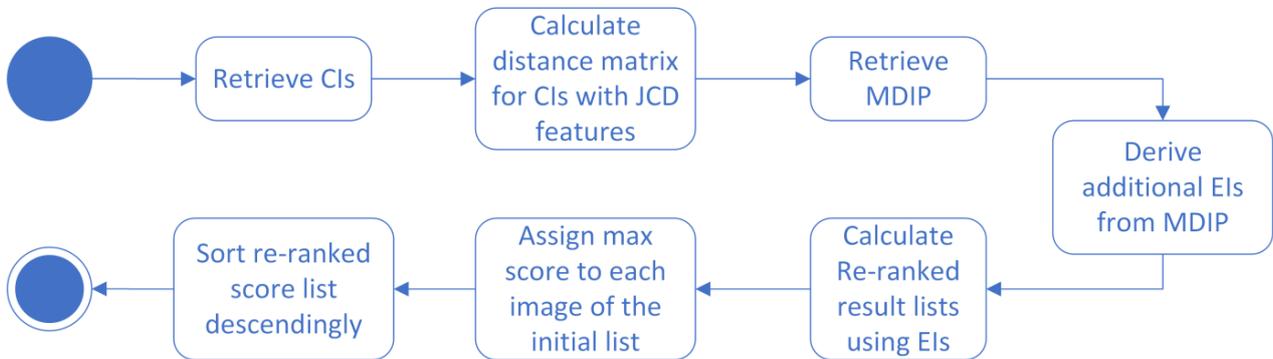


Figure 31: Basic steps of the proposed re-ranking algorithm.

CI: candidate image; MDIP: most desired image pair; EI: example image.

Several assumptions are made for this algorithm:

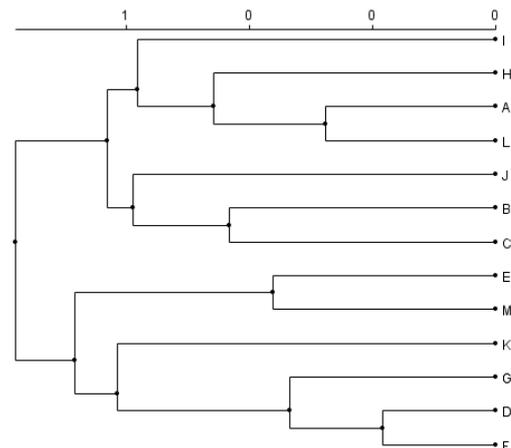
- 1) *Highest ranked images are also the ones most representative for the query.* This assumption corresponds to other literature (e.g. Maillot et al. 2007, Popescu et al. 2009, or Arampatzis et al. 2013). The here proposed algorithm only considers the first $M = 20$ images to be highly relevant (a constant value). These M images are labelled as *candidate images* (CI), which are possible choices for the K *example images* (EI) used in the re-ranking process. If less than M images are retrieved, all retrieved images are considered candidates for EIs. The choice of only a small static number has also to do with the way the system will be evaluated, as shall be seen in chapter Evaluation.
- 2) *Images with similar low-level image features are also close together when classified.* Although more of a fact, this assumption is made to actually enable building up a distance matrix for clustering the CIs into different categories. The distance between every CI to every other CI is estimated by using JCD features. Because the similarity estimation assesses the distance between two images, the conclusion can be drawn that the higher the score between two images, the smaller the distance between them. Therefore, the similarity (normalised to values between 0.0 and 1.0) of two CIs is subtracted from 1.0 to retrieve the distance instead, as Formula XXX describes.

$$\text{XXX} \quad \text{distance}_{i,j} = 1.0 - (\text{normalised_JCD_similarity_score}_{i,j})$$

An example of a distance matrix (for $M = 13$ images) can be seen in Figure 32 a).

Img	A	B	C	D	E	F	G	H	I	J	K	L	M
A	0												
B	0.6	0											
C	0.6	0.5	0										
D	0.9	0.8	0.7	0									
E	0.8	0.7	0.6	0.8	0								
F	0.9	0.7	0.7	0.1	0.7	0							
G	0.9	0.7	0.6	0.3	0.6	0.3	0						
H	0.4	0.7	0.6	0.9	0.5	1	0.9	0					
I	0.6	0.7	0.5	0.9	0.8	1	0.9	0.7	0				
J	0.5	0.4	0.6	0.8	0.7	0.9	0.9	0.7	0.7	0			
K	0.9	0.8	0.8	0.7	0.7	0.7	0.6	0.9	0.9	1	0		
L	0.3	0.6	0.4	0.8	0.7	0.9	0.9	0.5	0.5	0.6	1	0	
M	0.8	0.8	0.7	0.8	0.3	0.8	0.6	0.6	0.9	1	0.7	0.9	0

a)



b)

Figure 32: Distance matrix and resulting dendrogram.

The distance matrix in a) shows the distance between all the CIs. Only the lower part is shown because the upper part (blue) is identical. In b), a dendrogram is displayed resulting from a hierarchical clustering of the distance matrix in a).

A third assumption is made to define which CIs of the hierarchical clustering are now selected as EIs for re-ranking:

- 3) *The more clusters an image is contained in, the more relevant it is.*

Naturally, there is always an image *pair* that has to be selected in the first step, because this defines the first cluster that is not simply the image itself. The pair chosen as a starting point is the one with most parent nodes, corresponding to the earlier summarisation of images into a cluster. If there are more image pairs having the same number of parents, the one with the smaller pair distance is taken as the **most desired image pair** (MDIP) for re-ranking. In Figure 32 b), the chosen image pair thus is either (D, F) or (A, L), both contained within 5 clusters. However, because (D, F) are merged earlier (they are more similar to each other), they are chosen as MDIP. To have a better selection of images, the CIs in close proximity of the MDIP are added to the set of EIs for re-ranking. The number of EIs for re-ranking can be set to any number. Here, less than or equal to $K = 5$ EIs from the $M = 20$ initial CIs are used for re-ranking to allow certain variability, but at the same time remove unwanted noisy images as effectively as possible. $K = 5$ is set due to the way in which the system is going to be evaluated (only the first ten images of each ranked list will be used for evaluation. If e.g. 10 EIs were chosen, it could happen that these 10 EIs are also the 10 highest ranked images in the initially ranked list. An additional re-ranking would then not lead to any other ranking, making the evaluation useless as shall be clarified later on). To retrieve this number of images, the parent clusters of the cluster containing the MDIP are recursively visited. The parent cluster having the same (or less) number of children as the number of images chosen for re-ranking supplies the final set of EIs for re-ranking. Figure 33 illustrates how the dendrogram is recursively traversed to retrieve the final images for re-ranking.

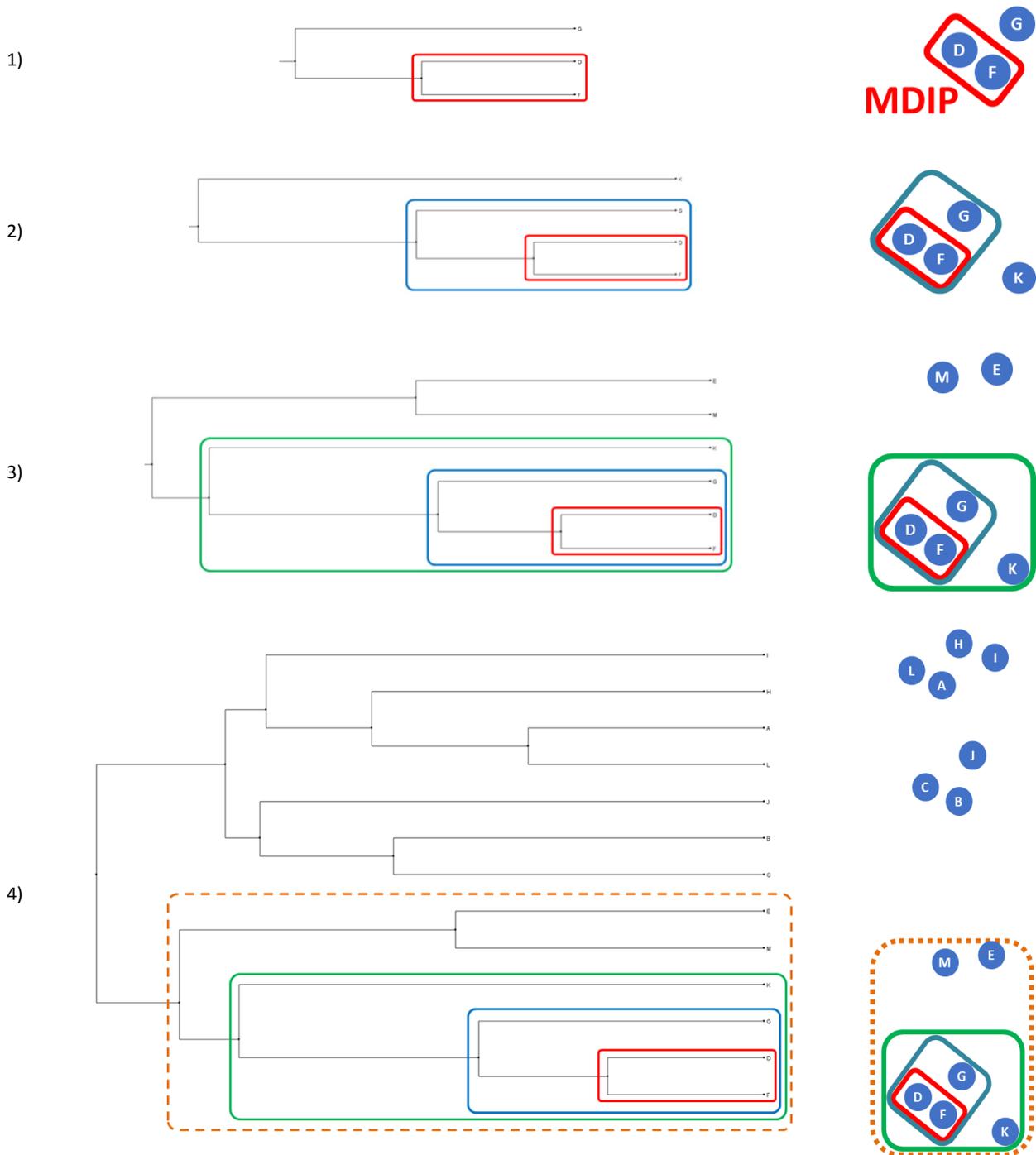


Figure 33: Recursive collection of EIs for final re-ranking from an initial set of CIs.

The left image indicates the actual retrieval conducted by the algorithm. The right image depicts a possible distribution according to the distance between images, and each circle corresponds to an image. Observe that E and M in 4) are not added anymore, because this would lead to more than 5 EIs. However, the cluster is still examined to decide if one more image could be added.

The MDIP in this example consists of D and F (in Figure 32 a), their distance is only 0.1, the smallest distance between any pair of images). Those two images define the starting cluster MDIP. The algorithm continues to recursively search through clusters of higher hierarchy, until the number of EIs specified for re-ranking is retrieved. In 2), G is added to the set. The number of images needed is 5, so the algorithm continues recursively to the next parent cluster, resulting in 3), where K is added to the set. Still 1 image is needed, so the algorithm looks at the parent of this cluster, where two images E and

M could be added to the set of EIs. Adding these two images, however, would exceed the specified maximum number of images for re-ranking (5). Therefore, the final set of EIs used for re-ranking contains only 4 images (D, F, G, K). Such an approach is very likely to reduce diversity, but as could be seen in Popescu et al. (2009), diversity may not necessarily lead to more relevant images to users. Therefore, as an initial start, considerations on diversity are left aside in this work.

As Figure 31 shows, after retrieving all EIs, they are used in the algorithm of Formula XXIX to retrieve the maximum similarity of each image of the initially retrieved result list to any of the K EIs. Then, its maximally achieved score is assigned to that image and all the images are sorted from highest to lowest score, resulting in the final re-ranked result list.

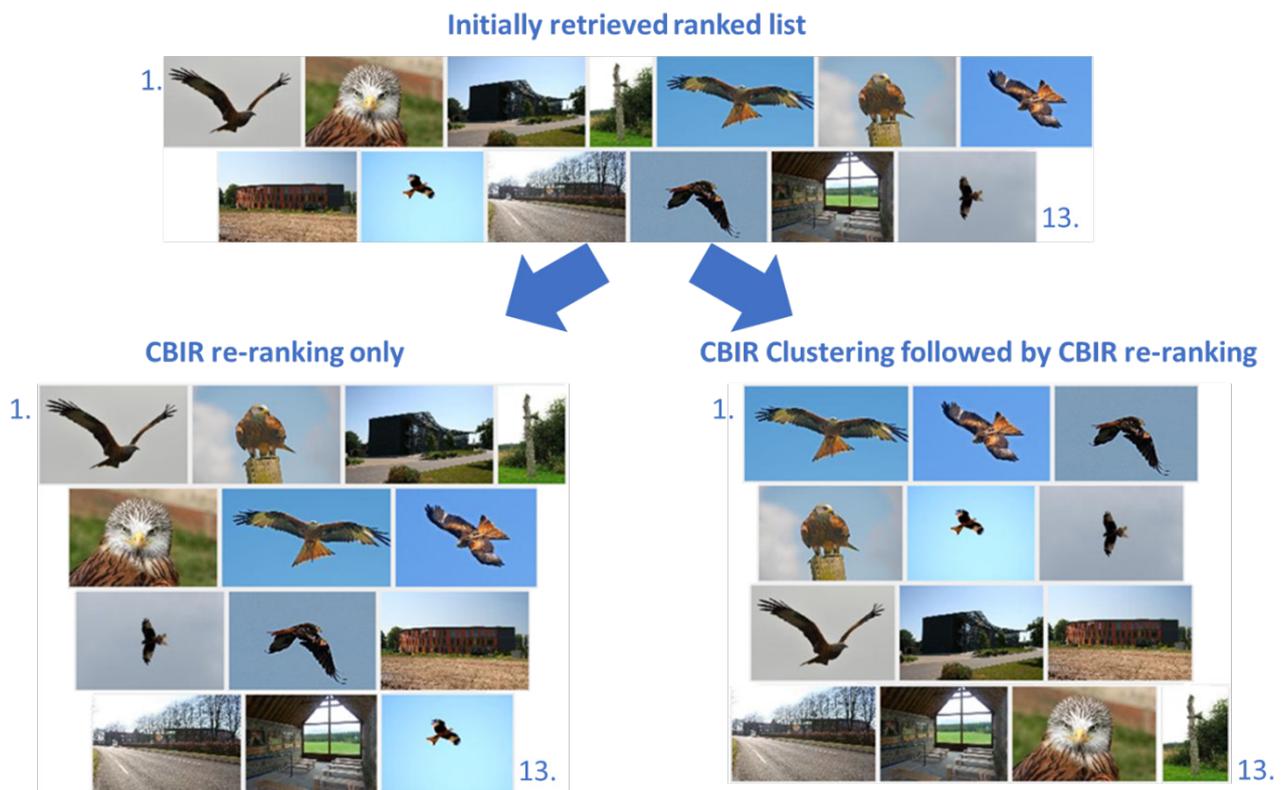


Figure 34: Re-ranking without clustering vs. Re-ranking with clustering.

The submitted query was "red kites". Images rank from highest in the upper left corner to lowest in the lower right corner.

Initial experiments using such a clustering step before re-ranking look promising, as Figure 34 indicates. The result list on the left side uses 5 images of the initial list for re-ranking but no clustering. This leads to a very noisy re-ranking, although desirable images now often are grouped together but possibly scattered throughout the result list. On the right side, the proposed clustering pre-processing step is utilised. Only one image of a red kite is not recognised and therefore given a lower rank. This small experiment indicates that relevant EIs can be distinguished more effectively from the non-relevant ones through such a clustering.

3.3.1.4.3 System Implementation Details of the Re-Ranking Algorithm

Figure 35 shows that IReranker defines a Strategy interface for re-ranking an initial score list. One class is derived from this interface: AbstractMaxScoreReranker. This class uses a SearchHitsFilter of LiRE to re-rank the subset of images retrieved in the initial result list using the low-level features stored in the content index. reorderScores() implemented in the subclasses defines how EIs are chosen for the re-ranking procedure. The way in which the EIs for re-ranking are collected is defined in two subclasses, MaxScoreReranker and ClusterMaxScoreReranker. The first class only provides simple re-ranking capabilities according to Formula XXIX using a fixed number of first K images. The second class additionally employs the aforementioned clustering before re-ranking on the M first CIs, selecting maximally K of those CIs as EIs. Hierarchical agglomerative clustering with average linkage criteria is carried out using DefaultClusteringAlgorithm in combination with AverageLinkageStrategy.

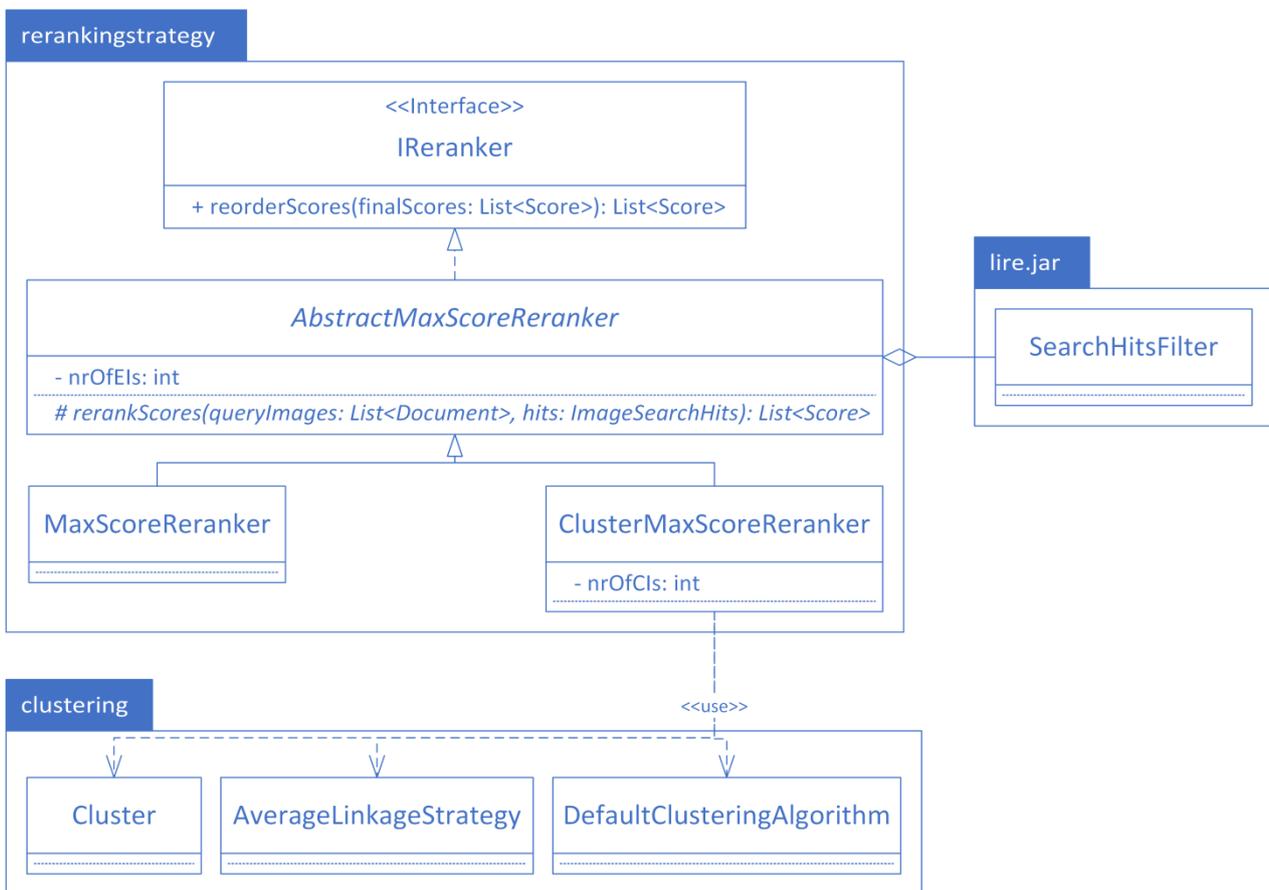


Figure 35: Overview of packages and classes involved in re-ranking.

3.3.2 Query-by-Example Refinement Algorithm

A second possibility is to query by an available example image. Although this could have been directly implemented as main search functionality, it was decided to only use this type of querying as *refinement* functionality for the user after having retrieved a list of images using the main algorithm. Refinement possibilities are also proposed in André et al. (2009) and described in chapter 2.5.3 New Interfaces for Image Retrieval. It is therefore a secondary algorithm, making use of an image that was

retrieved by a user initially. Thus, the system knows the image’s title, description, location, and content, providing much more information compared to an example image that may be uploaded directly by a user. Figure 36 shows the algorithm, which will be thoroughly examined in the sections afterwards.

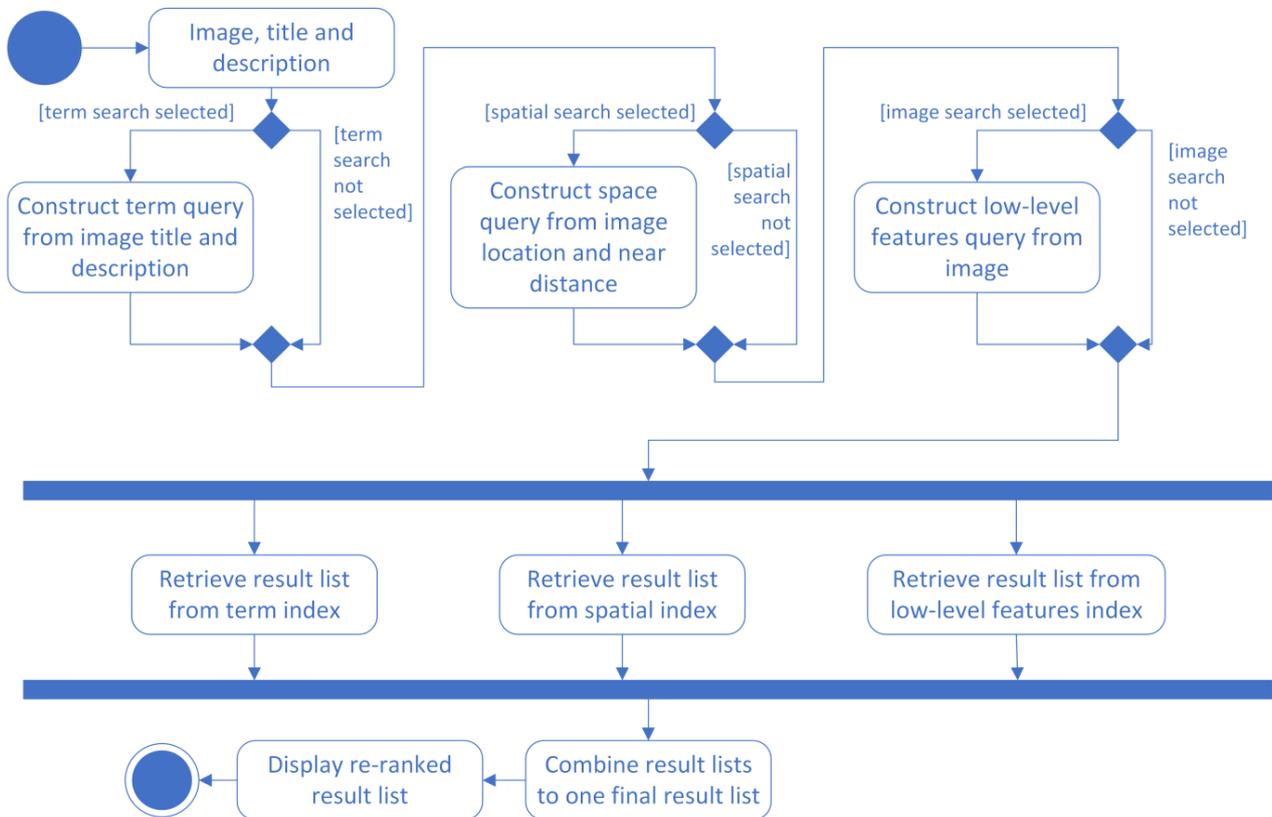
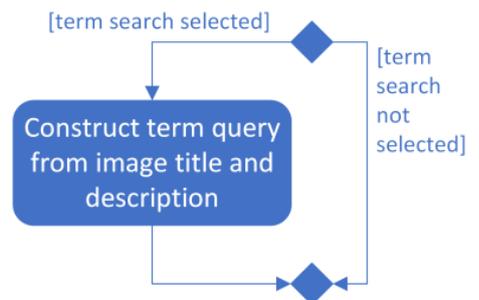


Figure 36: Secondary algorithm designed to refine an initial search result.

3.3.2.1 Term Query from Image

Title and description assigned to an image are used to construct a query similar to a term query, where a user types in keywords or a sentence, which is further processed as described in chapter 2.2.1 Textual Information Extraction. Although TermIndexSearcher is used in the same way as already shown for the main algorithm, title and description undergo an additional pre-processing step.

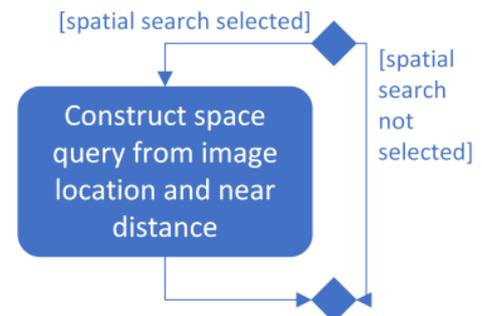
Image-describing texts are prone to contain plenty of “unwanted” words, and the AND operator used in TermIndexSearcher’s Lucene index retrieves only images containing *all* query words. Experiments thus show that, even if stop words are removed, only one image can be retrieved (the same used for querying). On the other hand, using the OR operator results in the retrieval of several images having barely any reference to the query image. Therefore, the pre-processing step aims at reducing the words to a reasonable number and at making sure that only important, theme-describing words are used as input to a term query. To do so, NLP is incorporated, which extracts all *nouns* from the input string by using a POS recognition algorithm. Any other word



type is discarded. In Figure 47 in the lower left corner, the package responsible for POS processing is depicted (posextractor). The class concerned with extracting nouns from title and description is `SimpleNounExtractor`, an implementation of `IPOSExtractor`. Tagging is supported by OpenNLP, an open-source NLP library (opennlp.apache.org). There are different sets for POS identification models available. The POS tagger uses the `en-pos-maxent.bin` Maxent model, the tokenizer the `en-token.bin` model, which can be downloaded from tinyurl.com/opennlp-models. `SimpleNounExtractor` implements exactly one method of `IPOSExtractor`, `extractPOS()`. Firstly, this method uses YPM to find any location descriptions in the submitted sentence consisting of title and description of an image and removes them. This is intended, because the system should strictly distinguish between term and spatial query. Thus, although space terms can occur, they are not part of the term query. A query “sentence” is built by concatenating the remaining nouns and submitted to the `TermIndexSearcher` as a normal term query.

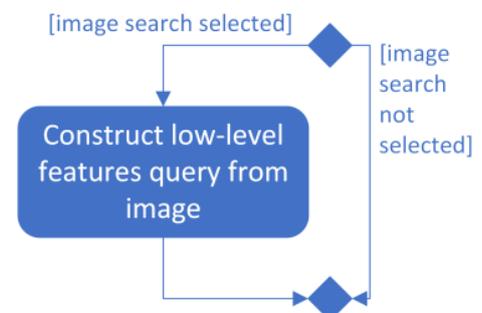
3.3.2.2 Spatial Query from Image

The second part of the retrieval algorithm encompasses the possibility to find images that are in spatial proximity to the example image. To accomplish this, a spatial query is generated using one of the provided near relationships as described in chapter 3.3.1.2.2 Retrieving Result List from Spatial Index, and an additional distance defines the circular extents (a buffer) of what is considered near (and, therefore, relevant) around the example image. This distance, however, has to be provided from outside by the user and can be altered at will. The ranked list is then obtained by submitting the so formed buffer query footprint to `SpatialIndexSearcher`.



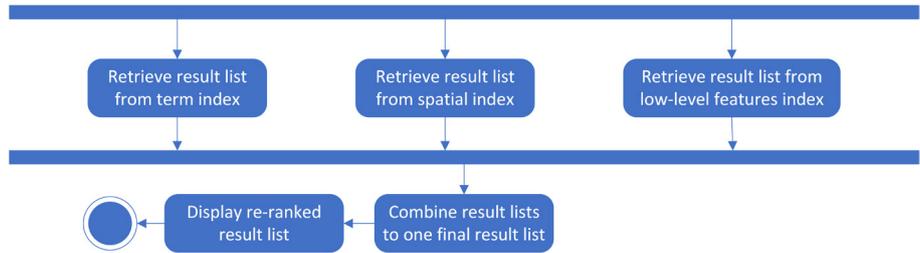
3.3.2.3 Low-Level Features Query from Image

Before the extraction of low-level features can be conducted, a first step limits the collection images to a reasonable number. This has to be done first because of heap space overflow problems occurring when ranking all images with global features so that the application can still be run on the used hardware. Thus, to make query by example applicable in this system, a pre-step taken uses only the union of the score lists retrieved in steps 3.3.2.1 and 3.3.2.2 if available. Although some relevant images may not be returned, it drastically reduces the needed computational power, making it applicable for tests. If no initial score list is provided (it is only queried for images with similar content to the example image), a term query as described in 3.3.2.1 is formulated and submitted to the term index, and low-level feature matching is applied only on the image subset retrieved through this term query.



3.3.2.4 Retrieval and Combination in Query by Example

Querying and combining result lists is similar to the procedure described in the main algorithm. However, a new type of `IIndexSearcher` called



`ContentIndexSearcher` has to be defined to make use of the low-level features index. It is shown in Figure 37. Similar to the other implementations of `IIndexSearcher`, `ContentIndexSearcher` requires to have an own implementation of `AbstractIndexQuery` (`ContentIndexQuery`), holding several important features to conduct a content query. A search for low-level features is accomplished using the `SearchHitsFilter` provided in the LiRE library, instead of `ImageSearcher`. The latter would rank the whole collection, making it not applicable for the used hardware as described before. Consequently, `ImageSearcher` is replaced by a normal Lucene `IndexSearcher`, which is able to only rank a specified subset of images.

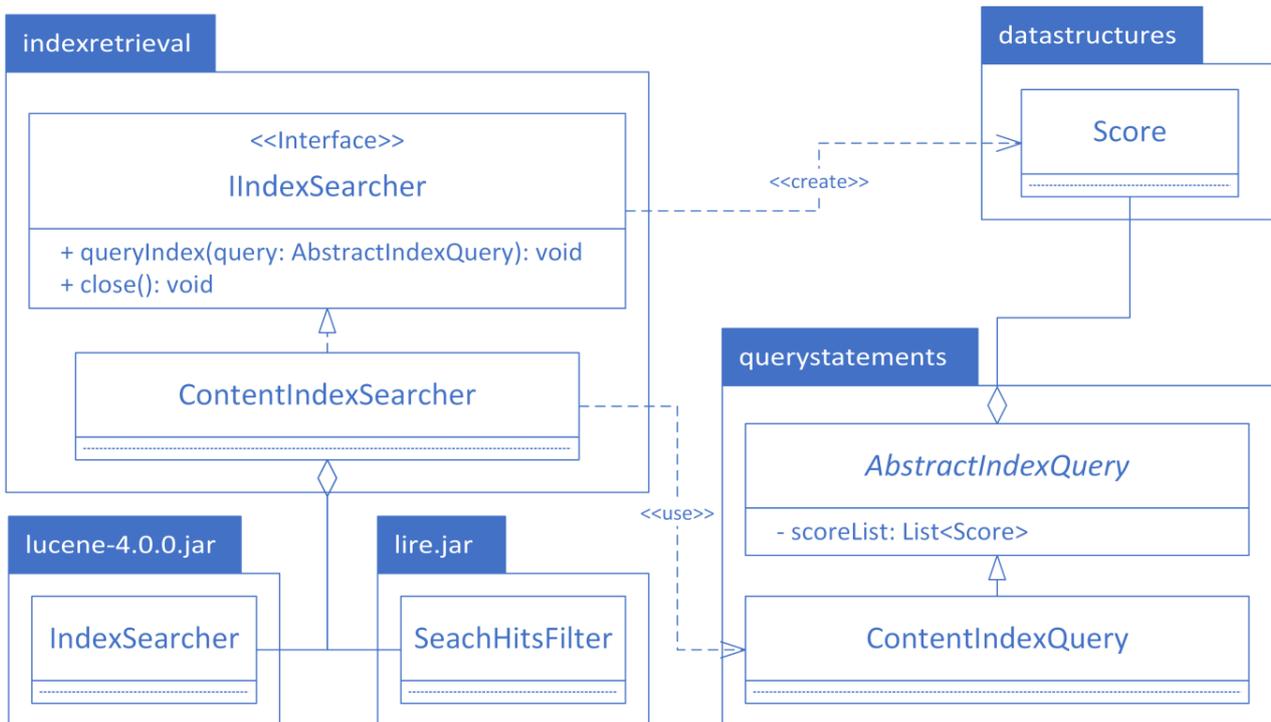


Figure 37: Packages and classes involved in querying the content index.

Querying indexes is conducted in parallel to increase retrieval speed. CombMNZ combines the result lists if more than one is retrieved. There can be either 1 (term, spatial or content query), 3 (term, spatial, and content query) or any combination ([term, spatial], [term, content], [spatial, content]) of result lists, depending on the user's chosen refinements.

3.3.3 Facilities for User Interaction with the System

The last part of the implementation focuses on submitting queries and displaying results in a user-friendly way. Principles are taken up from chapter 2.5 Information Visualisation Process Flow and elaborated. As explained there, one of the most common ways of user interaction with software is through the use of a GUI. Figure 38 shows the GUI of the developed SPAISE as it appears after submitting a query and receiving result images. It consists of three visible parts: the textual input on the upper side, a result list showing the retrieved images on the lower left side, and a map on the lower right side, displaying the locations of where the images are situated. All parts can be resized separately to allow a user control over a specific part of the GUI.

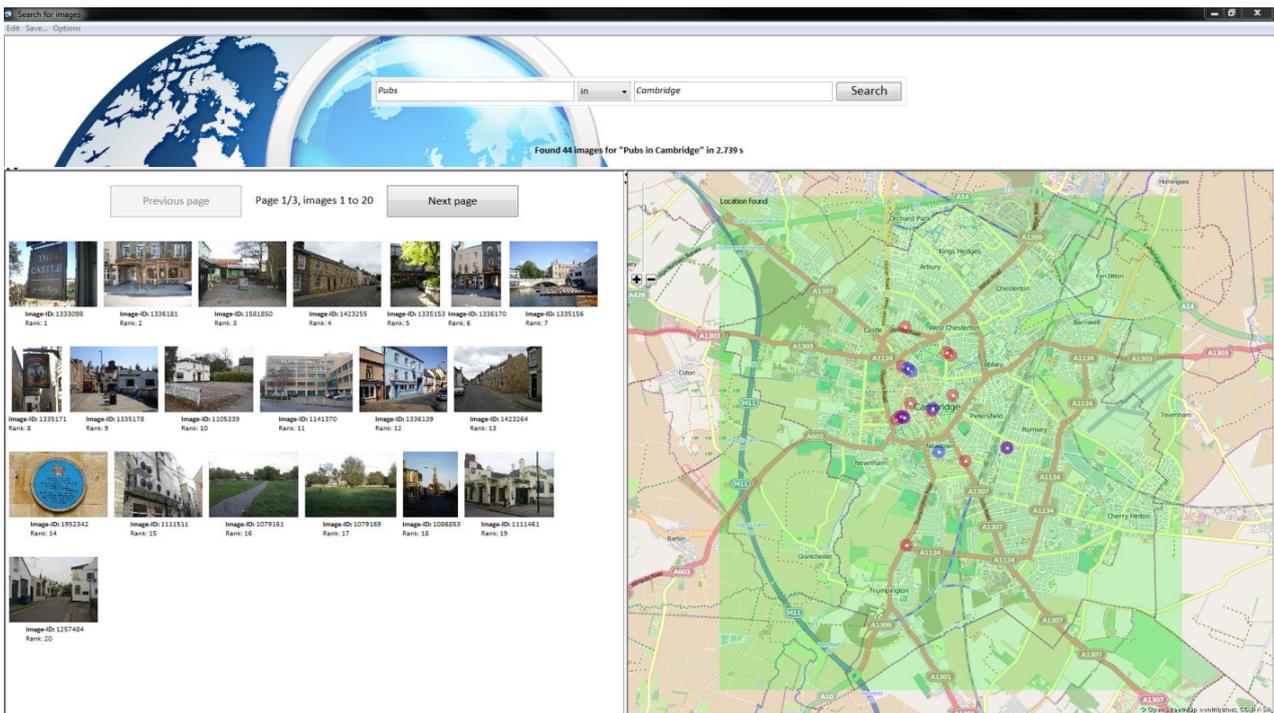


Figure 38: Screenshot of the GUI after submission of a query.

Corresponding to André et al. (2009), the interface provides ways for goal-specific, but also exploratory search. Goal-specific search is implemented by the textual input query field, see Figure 39.

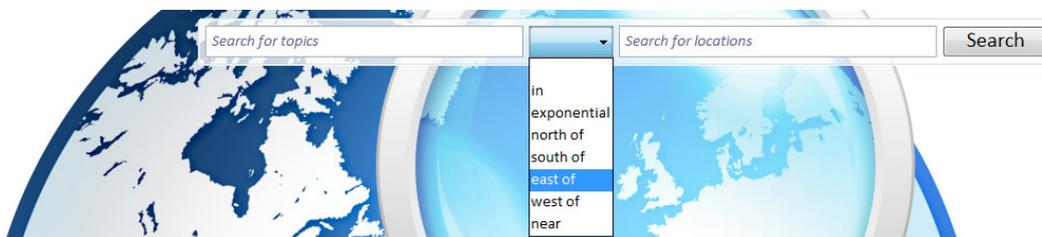


Figure 39: Query input in the form of <theme><spatial relationship><location>.

Search for topics: <theme>; drop-down menu: <spatial relationship>; Search for locations: <location>. “exponential” corresponds to Formula VII, whereas “near” is an implementation of Formula IX.

It exhibits basic input capabilities in the form of the <theme><spatial relationship><location> triplet. The drop-down field allows specification of the desired spatial relationship. Search information is

provided after image retrieval below the query field, indicating how many images were found and what the input query was. Queries submitted here undergo the procedure explained in chapter 3.3.1 Main Retrieval Algorithm to retrieve images.

Retrieved images are presented to the user in a result list. The result list displays thumbnails of images to users, as can be seen in Figure 40. The images are ordered according to their relevance from highest score in the upper left corner to lowest score in the lower right corner. Only an adjustable, limited set of images is displayed per page to increase retrieval speed (reading and writing from a hard disk are computationally expensive operations). Two buttons (“Previous page” and “Next page”) provide switching functionality between different thumbnail pages. The first page shows images with highest scores, whereas the last page contains images with lowest scores. It also displays information about how many pages are there altogether, which page the user is currently located on, as well as how many and which images are displayed on this page.

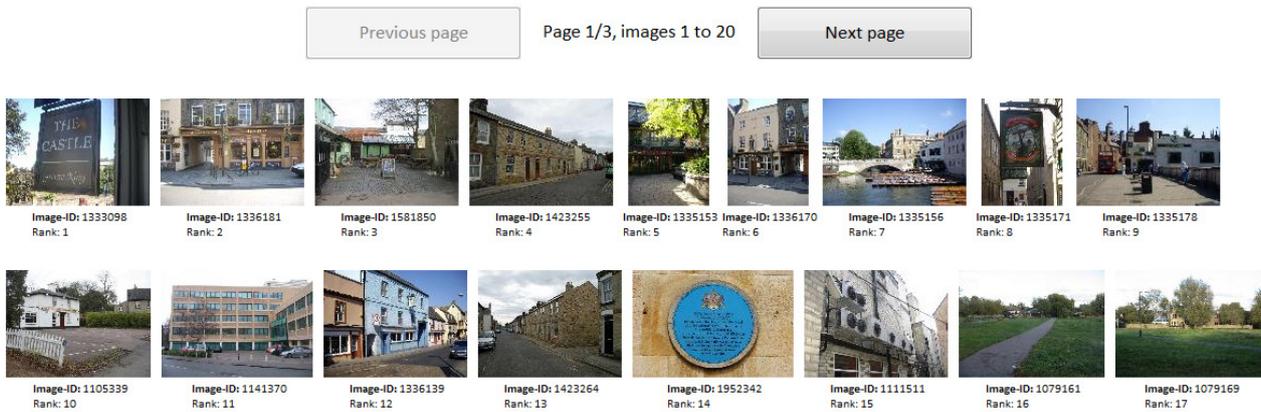


Figure 40: Representation of retrieved images.

Three pages exist, and the user is currently looking at the first page (1/3). 20 images are shown per page.

The way in which a user can interact with an initially retrieved list of images is both goal-specific and exploratory. If a user is interested in the title and descriptions of an image, its thumbnail in the result list can be clicked on as displayed in Figure 41.



Figure 41: Hover functionality on user interaction with image thumbnails.

If a user hovers over an image, its background colour turns from white to blue. A single click on the image turns the background from blue to yellow and opens a new window containing all the information stored for this image.

Hover effects indicate that the user is able to interact with the image thumbnail. A thumbnail in the result list shows already two informatory items: its identifier and achieved rank. No actual relevance score is displayed, because it barely adds any additional information to a user. A click on an image opens a new window, in which users find additional thematic information about the image, namely its indexed title and description, as shown in Figure 42. Spatial information is provided as well in the form of a map, where the image being currently inspected is highlighted using a black dot inside a yellow buffer. Not only *thematic* exploration is provided. User can also *spatially* explore an image as well as its surrounding images on maps either in the main window of the GUI or highlighted in the window opened when an image is clicked on. This map implements zoom and drag capabilities common in all map applications found throughout the internet. All images visible in the result list are displayed on the maps, having a buffer coloured according to the score they achieved. The colour scheme chosen is blue (high score) to red (low score). Hits between highest and lowest scores result in a gradual mixture of blue and red (purple). Besides the locations of the images, a green, half-transparently filled rectangle shows the query spatial footprint (MBR). Additionally, when clicking on an image depicted in the map, it also opens the same image information window shown in Figure 42. On query submission, the map automatically zooms to the retrieved location, if only one query footprint was retrieved. Else, the system makes sure that all the query footprints are visible by taking the MBR around all these query footprints.

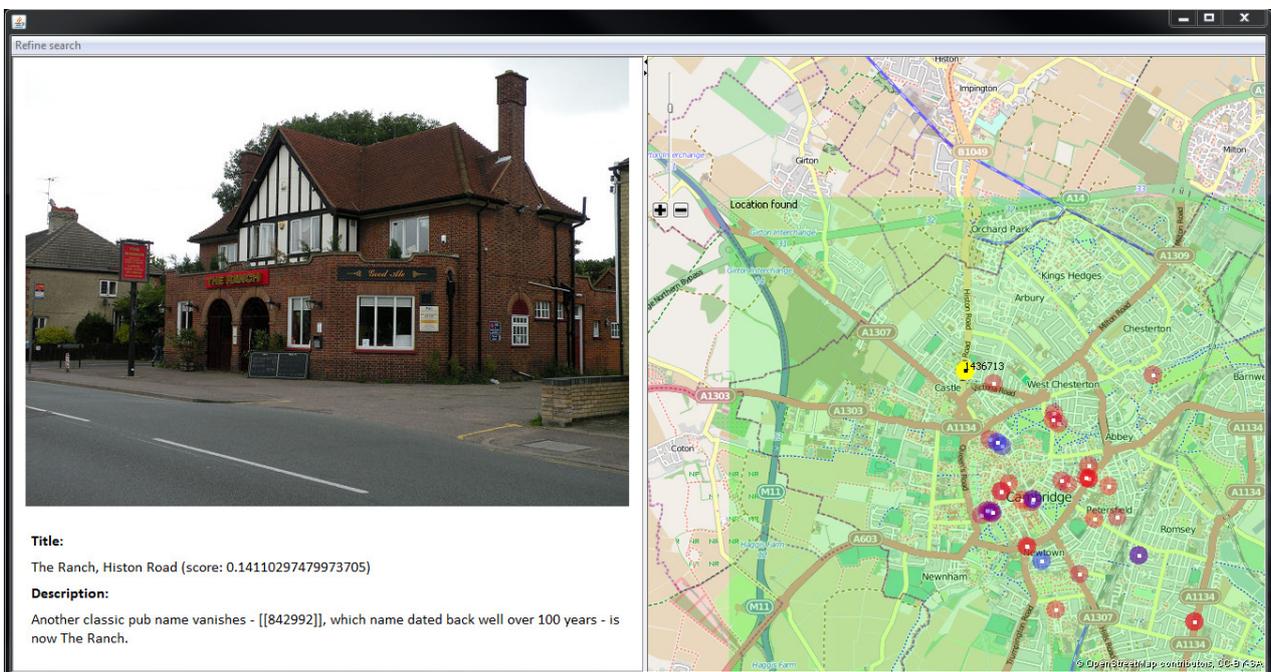


Figure 42: An image window.

Triggered, when a user clicks on an image in the result list.

When the SPAISE is started, it is initially centred at the centroid of the United Kingdom (WGS 84 coordinates: 54° 2' 41" N, 2° 46' 46" W, GeoHack 17.07.2013). The underlying map is based on *Open Street Map* (openstreetmap.org), a freely available, open-source map search application. Figure 43 provides a detailed look on the map.

A brief look at the classes involved in creating the GUI gives an insight into how these components work together. In Figure 45, `View` belongs to the actual large window. This `View` contains several parts, the `QueryPanel` (Figure 39), the `ResultPanel` (Figure 40) and the `MapPanel` (Figure 43). These are the integral parts of the GUI. If a user clicks onto an image, an `ImageWindow` (Figure 42) is opened, containing an `ImageInfoPanel` with the image in full size, the title and description as well as a `MapPanel` (Figure 43), showing the location assigned to the image. The refine checkboxes window in Figure 44 is part of the `ImageWindow` and therefore not separately listed. A part worth mentioning is `JMapView`, a framework built to use OSM maps in Java applications (`JMapView` 09.07.2013). This class provides the whole map interaction abilities. `MapFactory` therefore can create different map types based on `JMapView`. `createSpecificMap()` function returns a map designed to highlight an image as used in Figure 42, whereas `createOverviewMap()` defines a general map like the one incorporated into `MapPanel` in Figure 38.

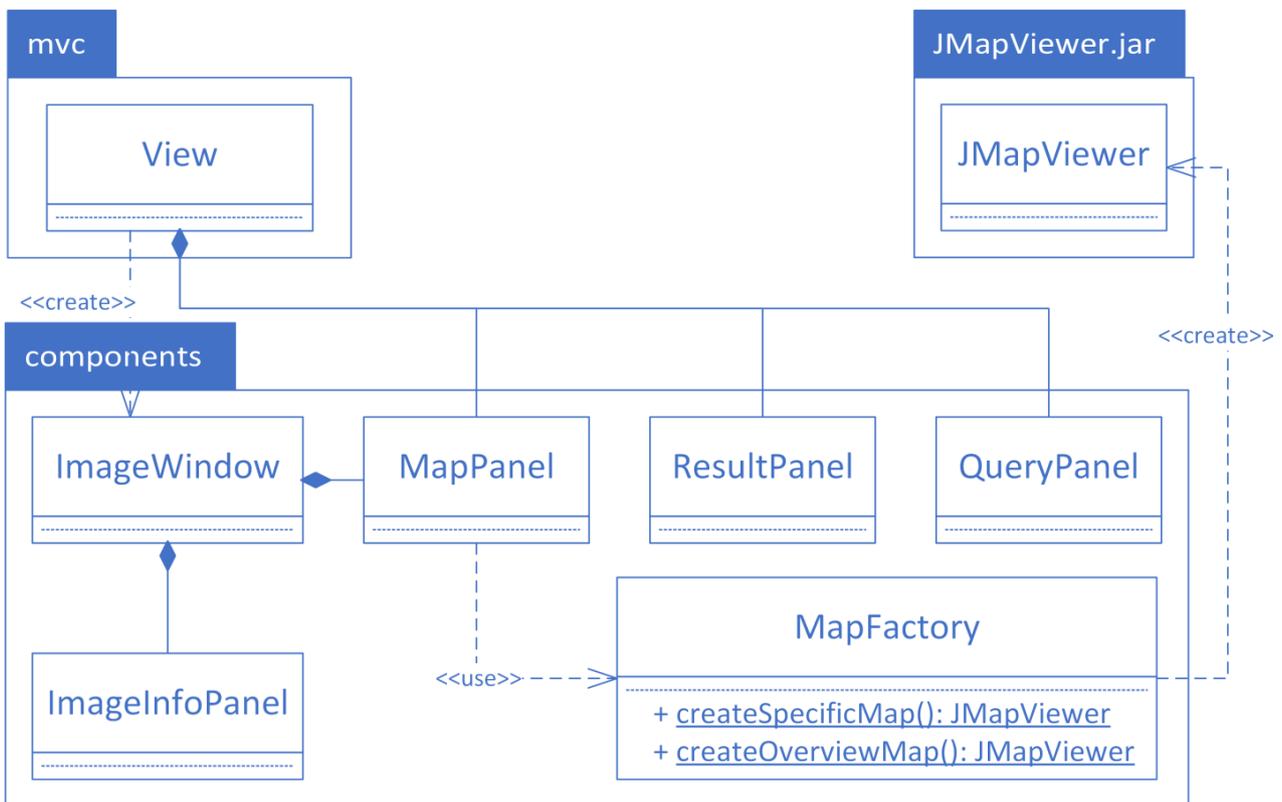


Figure 45: Components used to build up the GUI's main parts.

3.3.4 System Architecture and Interactions

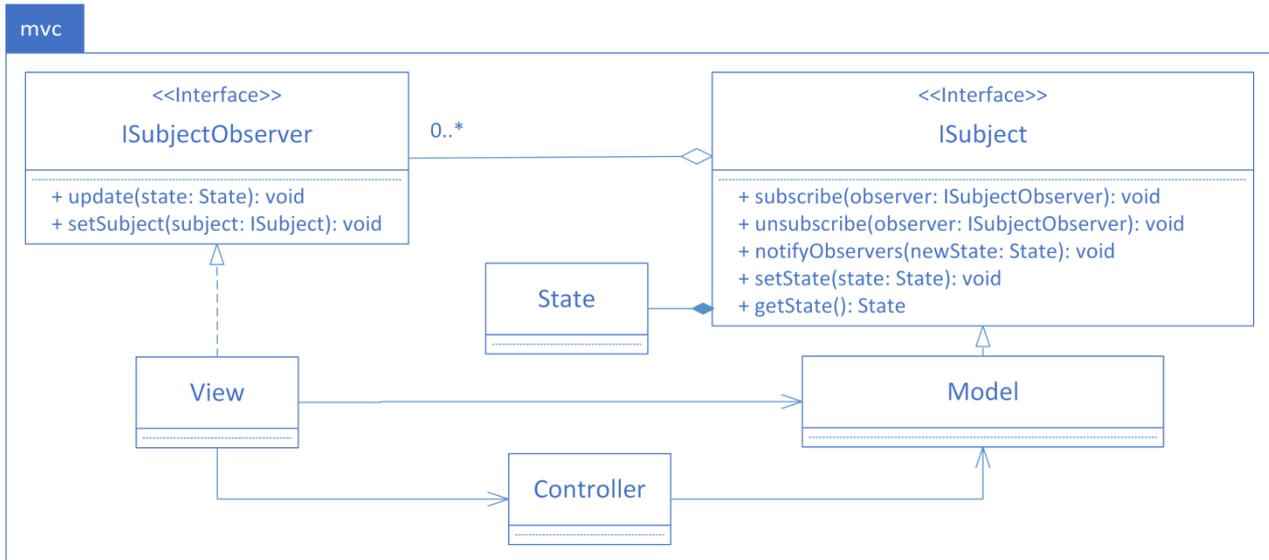


Figure 46: Main classes involved in the MVC architecture.

The architecture used as a base for this GUI is oriented towards the so-called *model-view-controller* design (MVC, described in Fowler 10.07.2013). MVC is a widely used concept that enforces a separation between the *view* of the GUI, meaning all the objects visible on the screen, the *controller* for each of these objects of the view, and the *model*, which stores all data of the currently active state of the GUI. Essentially, view and controller observe the model and are notified by the model when its *state* changes. There can be different views and controllers for the same model at the same time. The word choice of *oriented towards MVC* is intended. It is not a classical MVC, because this architecture mainly refers to the way the GUI, not the functionality concerned with processing inputs, is implemented. In the case of this SPAISE, the whole system is an “MVC”, where the controller acts more like a central query processing unit than a controller of the separate view components (see Figure 46). The `Controller` class represents the “brain” of the whole system and implements all the algorithms for conducting image searches.

A DP that lends itself to implement an MVC-like architecture is the *Observer* (Gamma 2011), one of the most popular DP. What makes it so popular and widely used is its loose coupling. `ISubject` represents an interface for an observable class. It provides methods to subscribe and unsubscribe `ISubjectObserver`. The essential point is that this `ISubject` provides a `State`, an object which stores the system’s current properties and which is interesting for the observing subscribers. Whenever the state of the subject changes, all its observers are notified through the `notifyObserver()` method, and all observers then update their state accordingly using their `update()` method. There are no restrictions on how many subscribers observe a subject. As an example, consider a radio station (the subject) that sends its program over some frequency throughout the air. It does not care how many people listen to it. Furthermore, it does not even care *who* is listening. On the other hand, although all listeners (observers) hear the program (are notified),

they can simply choose to ignore changes in a state if it does not concern them, or instead, act accordingly (update their state, e.g. if radio station listeners planned to go on a hike, but the weather forecast predicts heavy rains).

Any query submitted through the View will be forwarded to the Controller. It then executes the queries, retrieves the results and updates the State of the Model. A change in the State is primarily caused by the submission of a new query, but also changes in the GUI's properties (e.g. number of displayed images per page) can cause State updates. This procedure guarantees that all the views are always in sync with the current state of the GUI. State itself is a very simple class holding many updatable members.

Summarised, the core of this system consists of the *Model* holding all the data to operate the SPAISE within a *State* object, the *View* concerned with displaying the data in an appropriate form, and the *Controller*, which conducts all search operations.

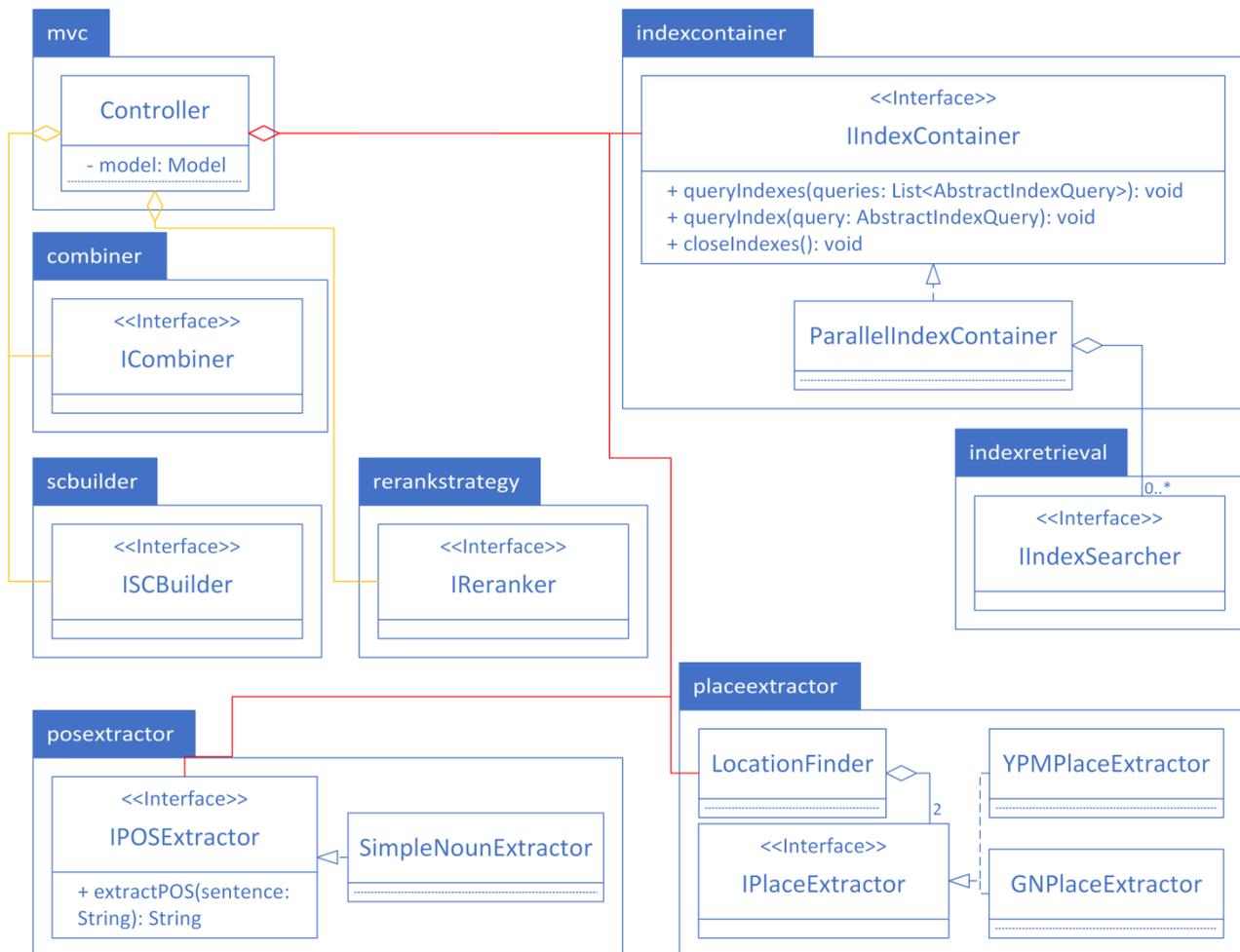


Figure 47: Main functionality synthesised as classes in Controller.
 Red denotes constant, whereas yellow means variable component at runtime.

Finally, Figure 47 shows how the different classes for retrieving images are connected to the Controller. Yellow aggregation lines indicate components exchangeable at runtime, whereas red

lines indicate components not exchangeable during search engine execution. All components connected with yellow lines are vital for retrieval. Of the modules connected with red lines, `indexcontainer` stores all the `IIndexSearcher` and therefore is responsible for querying indexes and retrieving result lists. The `Controller` can directly interact with an instance of `IIndexContainer`. For the sake of performance, the container is implemented as `ParallelIndexContainer`, querying all the indexes simultaneously. Another red aggregation occurs between `Controller` and `LocationFinder`, because this class implements the functionality explained before to first query YPM, and only then query GN to retrieve locations, where the ordering of the query matters. Lastly, `posextractor` provides the functionalities needed to construct a term query as explained in 3.3.2.1 Term Query from Image.

4 Evaluation

An effective assessment of the initially posed research questions requires selecting and combining different evaluation levels (see 2.6.1 User- and System-Centred Evaluations). However, the main research interest in focus is the evaluation of the processing level (3) of the SPAISE. Fortunately, the intended task design will also lead to the creation of user reactions to retrieval results, providing a base for UCEs evaluations.

4.1 Creating a Test Collection

As shown in chapter 2.6 Assessing a Search Engine's Performance, the core part of a SCE based search engine evaluation is the *test collection*. However, there is no test collection available that has a large enough corpus of documents to fulfil the needs of a SPAISE. Furthermore, there does not exist a set of standardised topics for the evaluation of a SPAISE. Therefore, the next section focuses on the creation of a suited corpus of documents and UINs (topics).

4.1.1 Creating a Corpus of Documents

Documents to evaluate are images. Images have been taken from Geograph.org.uk. It is a project where any person can upload geo-referenced images to cover as much of the whole of the United Kingdom and Ireland as possible. Therefore, all the titles and descriptions annotated to images are submitted by "laymen" and not necessarily accurate or even to the point (meaning, not all the images' annotations actually refer to what can be seen on the image, but may instead represent an opinion or remark of the person who took the image). The whole area of the islands is divided into squares, and 267692 of 331960 of these squares have been photographed so far. 11810 members contributed 3.5 Million images, equalling an average of 13 images per square. Besides the vast number of images, the data set is especially suited for the task of evaluating a SPAISE due to the fact that the distribution of the images is much more



Figure 48: Distribution of images around UK and Ireland.

uniform compared to other geographically annotated image collections like e.g. Flickr. In other collections, by the majority, images tend to be concentrated inside and around settlements. Outside such settlements there are often only few images to be found. Also, many of the Geograph images are annotated rather correctly, because there is an actual purpose/task behind collecting images other image collections may not have.

Several pre-processing steps have to be undertaken to create an utilisable image collection ready for indexing and evaluation explained in Appendix F. All images of Geograph.org.uk are contributed by Barry Hunter. Most images are accompanied by a title, description and the approximated location of where the objects shown in the images are located (note: this is not the same as the position of where the photographer was located on capturing an image, although these locations are available, too). The spatial locations are represented as WGS 84 latitude and longitude coordinates. Image metadata (including titles, descriptions, and locations of images) can be downloaded as MySQL relations from data.geograph.org.uk/dumps. Images not having all the needed data are discarded, resulting in 2,255,301 utilisable images. Figure 48 displays the distribution of a subset of around 700,000 images. Each image is represented as a half-transparent black dot. The darker the region, the more images are located there. Although there can be seen some very dark spots around larger city centres like London, it is still notable that most of the UK is represented. This enables queries for rather unknown locations as well. The Irish island though is relatively poorly covered with images. Only the northern parts show some very densely covered areas where images were taken. Therefore, the main focus (also for topic creation) is given to the larger island of Great Britain.

4.1.2 Creating Topics for Relevance Judgements

Although literature suggests around 50 – 150 topics be considered a sufficient number for a reliable evaluation (Carterette and Voorhees 2011), only 25 topics are defined for this thesis due to limited resources. It was shown that already as few as 25 topics can be used to compare the relative effectiveness of different retrieval systems with great confidence (Voorhees 1998). Several problems need to be avoided in the process of creating topics (Müller 2010):

- 1) There need to be (enough) relevant images in the collection for the topic.
- 2) Topics should not only work well for the researcher's system.
- 3) Topics should not be solely based on technical possibilities (ignoring an actual "real world" application of the system).

To overcome the first problem, all the three systems are queried with the topics and it is assured that at least 10 images per system can be retrieved. To avoid the second problem, only the numbers of found images, not the actual images themselves, are stored and inspected. There is therefore no data available that indicates the performance of any of the systems, which could lead to possible biases and

recreation of topics if examined beforehand. Averting the third problem is especially difficult because only few search systems have been evaluated so far that explicitly include a spatial dimension, let alone actual SPAISEs. Therefore, topic creation is inspired by *existing* topics found in e.g. Purves et al. (2007) and Müller (2010) already used for the purpose of evaluating search engines.

Incorporating the spatial dimension properly into the newly defined topics is crucial. All six spatial relationships (in, near, N/S/W/E of) implemented in the TS and TSCR systems are balanced throughout the 25 topics, as can be seen in Table 16. It is assured that each of the possibilities is more or less equally represented. Figure 49 shows an example topic. All topics can be found in Appendix G. The XML-like topic structure is borrowed from TREC-like topics like the ones found in Sanderson (2010). Additionally, the topics defined here explicitly incorporate a spatial part into the narrative, which is separated from the theme part of what the image is about. The intention is to make it clear that the topic actually *has* a spatial part and to account for the abilities of the systems. Furthermore, such a structure emphasises the assumption that users submitting queries to such systems have UINs with a spatial component.

Spatial Relationship	Count
In	4
Near	4
North of	4
South of	5
East of	4
West of	4
<i>Total # queries</i>	<i>25</i>

Table 16: Number of each spatial relationship used in the topics.

```
<topic>
  <number>24</number>
  <title>Cemetery in Chester</title>
  <description>What images of cemeteries in Chester can be found?
</description>
  <narrative>
    <theme>The image has to show a cemetery and it should be clear from
      the image that what somebody looks at is part of a cemetery. </theme>
    <spatial>The cemetery has to be located within or on the border, but
      not outside Chester (the city in Cheshire, England). Use the map to
      determine the location of the image. </spatial>
  </narrative>
</topic>
```

Figure 49: A topic defined for evaluation of a SPAISE.

4.2 Experimental Setup

4.2.1 SPAISE Hardware

Table 17 gives a brief overview of the hardware used for conducting queries on the SPAISE.

Component	Description
<i>Model</i>	Asus X53SJ-SX148V
<i>Processor</i>	Intel Core i7-2630 QM Sandy Bridge quad core 2.0 GHz (2.9 GHz Max Frequency), 6 MB Cache
<i>RAM</i>	4 GB
<i>System type</i>	64 Bit Windows 7 Professional operating system
<i>Secondary Storage</i>	Samsung <i>Solid State Drive</i> (SSD) 830 (520 MBps read/320 MBps write), 128 GB
<i>External Hard Disk Drive</i>	Seagate Backup Plus Desktop 3.4 TB
<i>USB</i>	2.0

Table 17: Hardware used to conduct experiments.

Indexes are stored on the internal SSD drive, leading to fast index access times. However, the image collection to index is too large for the internal, fast SSD drive to store (around 70 GB). Thus, a larger but also slower external HDD needs to be used to store the images. Moreover, indexing cannot make use of the USB 3.0 interface the external HDD provides because the USB port of the employed computer only offers USB 2.0, leading to very high indexing times for image content.

4.2.2 SPAISE configurations

Table 18 gives an overview of the settings used within the SPAISE for the evaluation. Three systems need to be tested: T, TS and TSCR.

T acts as a baseline. It uses only text indexes to retrieve images and does not process spatial information separately. Therefore, spatial information is treated like any other textual information and assessed using tf-idf weighting and cosine similarity.

TS, on the other hand, both uses the term index for processing thematic information and retrieves query footprints for spatial information with an additional processing of the query according to the specified spatial relationship (in, near, N/S/W/E of).

TSCR finally uses all the functionalities explained in 3.3.1 Main Retrieval Algorithm, thus conducts additional re-ranking using clustered EIs.

As a consequence, only the main algorithm is evaluated. Neither the query-by-example algorithm nor interactions of users with the system are being assessed. Only few initial restrictions hold for this first evaluation (see Table 18). No weights are assigned to the different dimensions, so they all equally contribute to the estimated relevance of an image (indicated by weights of “1”, so multiplication with these weights has no effect on the result list’s ordering). The linear distance factor is smaller for the actual near relationship compared to the directional relationships (N/S/W/E). This has to do with the

assumption that, when someone is looking for locations near another location, these places should also be located close to each other. However, for directional relationships, these restrictions are not that strict due to the fact that they do not per se specify any limitation to nearby places. Furthermore, no threshold restrictions are given. This means that even if an image only has a similarity score of 0.01, it is still returned. The main advantage is that more images may be returned in each dimension. Especially for TS and TSCR this may prove vital, because in the end, CombMNZ will only be calculated for images having scores in each dimension (intersection), drastically reducing the number of relevant images compared to the union that is applied in the T case. It was decided to use CombMNZ for result list fusion in the TS and TSCR systems because of its good performances in Palacio et al. (2011) in the context of combining geographic dimensions. T uses a Boolean OR for evaluating terms, because it is very likely that some place names and spatial relationships may not occur in a title/description of an image. A Boolean AND operator in that case would not retrieve any results (because all the terms specified in the query need to be apparent in the retrieved image descriptions), making evaluation through comparisons with TS and TSCR senseless.

a)	T	TS	TSCR
	-Term index only (T). - <i>Baseline</i> for the evaluation.	- Term and spatial indexes (TS). - Term index like T (adaption in i)).	-Term and spatial indexes, content-based re-ranking (TSCR). -Initial results like TS.
b)	-Term index only: see 3.3.1.2.1 Retrieving Result List from Term Index -No spatial query processing.	- Spatial index and spatial relevance methods: see 3.3.1.2.2 Retrieving Result List from Spatial Index.	-Addition: re-ranks result list retrieved with TS (clustered EIs). -Main algorithm: see 3.3.1 Main Retrieval Algorithm.
c)	-	CombMNZ	CombMNZ
d)	-	{1, 1}	{1, 1, 1}
e)	-	Linear, distance factor 1.1	Linear, distance factor 1.1
f)	-	Linear, distance factor 1.5	Linear, distance factor 1.5
g)	Term: 0.0	Term: 0.0 Spatial: 0.0 Combined: 0.0	Term: 0.0 Spatial: 0.0 Combined: 0.0 Re-Rank: 0.0
h)	All	All	All
i)	Boolean OR	Boolean AND	Boolean AND
j)	-	Intersection	Intersection

a) System abbreviation.
b) System description.
c) Score combination strategy in the case of multidimensional queries.
d) Dimension weighting in the case of multidimensional queries.
e) Distance factor of the near relationship in the case of multidimensional queries.
f) Distance factor of the directional near relationships (N/S/W/E).
g) Minimum similarity score (threshold) an image needs to achieve to be retrieved.
h) Maximum number of returned images.
i) Term index's term results combination strategy.
j) Result set combination strategy in case of multidimensional queries.

Table 18: Evaluation settings of the systems.

4.2.3 Indexed Images and Metadata

676,016 images or around 30% of the 2,255,301 images of the created image corpus are used for the evaluation (image identifiers 1000003 to 1999998 without all the removed images). Each image is accompanied by a title, description, and an approximated location in WGS 84 coordinates of where the object in the image is located. Table 19 summarises facts about the indexed images. Indexing times are especially high for content indexing, because each image is retrieved using the slow USB 2.0 external HDD. Therefore, parallel indexing cannot be exploited to its theoretical limits.

Index type	Indexing time	Index size	Number of indexed features
Term index	13 s	182 MB	<i>Terms</i>
			From titles: 85583 From descriptions: 231001
Spatial index	395 s (6 min 35 s)	54 MB	<i>Coordinates</i> 676016
Content index	68332 s (18 h 58min 52 s)	894 MB	<i>JCD</i> 676016

Table 19: Various key performance indicators for indexing.

4.2.4 Image Pool

All 25 topic titles are treated as queries and submitted to the three system configurations (T, TS, and TSCR). Only the top-10 retrieved images are pooled together, resulting in 250 images per system or altogether 750 images. 138 Images occurred multiple times and are therefore removed, resulting in a final set of 612 images to evaluate.

4.2.5 Tasks

For human assessors to be able to evaluate the relevance of an image, a well formulated task needs to be generated. A task states the topic, provides an image and its metadata for evaluation, and gives a clear and unambiguous description of what the assessor has to do. In this evaluation, one task comprises 2 to 5 units. A unit is an image with corresponding title and description as well as a map showing its location. Furthermore, a topic (title, description, and narrative) explains what the judge needs to assess (see Figure 49). Lastly and most importantly, a task description is provided at the beginning of a task. The purpose of the research and a description of the task's content are stated, too. Additionally, it has to be made sure that assessors understand the distinction of the thematic and spatial part of a topic, so that they actively distinguish these parts in their evaluation endeavour. Each image needs to be judged on a four-point relevance scale that can be found in Table 20. It is based on the four-point scale described in Järvelin and Kekäläinen (2002), but altered and adapted to fit relevance assessment of a SPAISE. An additional “not sure” ranking option gives the judges the freedom to skip a ranking task if they are not able to determine an image's relevance to a topic, although this option does not contribute to the RJs at all. Task creation is explained in Appendix H together with an example subtask.

Rank	Verbal description	Explanation
-	Not sure	<i>You're not sure if the image matches the topic. In this case it is particularly important that you add some text explaining why you were not sure.</i>
0	Irrelevant image	<i>The image (together with its texts and location) doesn't fulfil any requirement stated in the topic.</i>
1	Marginally relevant image	<i>The image (together with its texts and location) fulfils only one of the requirements stated in the topic.</i>
2	Fairly relevant image	<i>The image (together with its texts and location) fulfils most, but not all, of the requirements stated in the topic.</i>
3	Highly relevant image	<i>The image (together with its texts and location) fulfils all the requirements stated in the topic.</i>

Table 20: Definition of the four-point relevance scale.

An important part of the creation of tasks is to ensure that judges not interested in contributing trustworthy RJs can be removed safely before any further processing of the data is conducted. Two *traps* are incorporated into each task to accomplish this requirement:

- 1) Each task contains one image irrelevant to a topic (as stated in Zhu and Carterette 2010).
- 2) An image to judge is accompanied by a compulsory text field, where judges need to fill in the reason for choosing a certain rank for an image.

Altogether, 163 jobs are created for CS RJ gathering. The effectiveness of these measures will be evaluated in the results part. Additionally, the compulsory text field provides a valuable source of user comments for UCE.

4.2.6 Platform

The CS platform used for evaluation is CrowdFlower. CrowdFlower enables the creation of tasks explained before that will be distributed through channels to assessors. These tasks are called jobs. A job is an evaluation task a judge has to complete to receive a reward in monetary form.

4.2.7 Participants

Participants are judges gathered through various channels provided by CrowdFlower. For images with an easily distinguishable location (e.g. through inspection of the provided map), judges from various countries are allowed to participate in the relevance assessments. A selection of judges originating in several English-speaking as well as some other countries located close to the UK (see Table 21) may submit RJs.

Australia	United Kingdom	Netherlands	Liechtenstein	USA
Austria	Canada	Ireland	Norway	Switzerland
Sweden	New Zealand	Germany		

Table 21: Countries allowed participation in the evaluation.

For images with clearly and/or easily distinguishable location information.

It is assumed that judges from these countries will most likely provide reliable RJs, due to either their understanding of the language or their knowledge of the places mentioned in the queries. Some of the images' locations are difficult to spot on a map and/or are not restricted by a border (for example the Scottish Highlands). Therefore, jobs containing queries with such ambiguous place extents are only submitted to participants from the United Kingdom or Ireland. The assumption is that they have more locational knowledge and therefore provide better judgements than people not living in this area. It is assumed that gathering judgements from only some few countries like the UK and Ireland takes longer compared to gathering RJs of participants from all over the world. Therefore, only 20 different judgements need to be collected for such a job, whereas a "normal" job submitted to many countries has to be judged by 27 assessors.

4.2.8 Experiment Realisation

The previously generated 163 tasks/jobs are uploaded onto CrowdFlower and distributed to judges of the specified countries. A job costs 1.98\$ (for 27 judgements and 5 images, equalling 1.5 cents per judged image). A smaller job with only 20 judgements costs 1.49\$. Altogether, the experiment costs 298.18\$. The assessment lasts 7 days.

4.3 Pre-processing of Raw Relevance Judgements

4.3.1 Aggregating Relevance Judgements

After conducting the experiment, all the 163 jobs holding the individual RJs have to be downloaded and aggregated. However, the raw data material first needs to be aggregated and cleaned from untrustworthy contributions. This procedure is explained in the following section.

4.3.1.1 Central Tendency: Arithmetic Mean, Median and Mode

For each image, up to 27 RJs may be submitted by CrowdFlower judges. To be able to apply any of the performance measures though, only one RJ per image is required. However, different assessors may judge an image differently. Therefore, the set of RJs submitted for each image needs to be appropriately aggregated, so that an *average RJ* or *rank* for each image can be derived. One way of aggregation is the use of measures of central tendency. Some of these measures commonly used in statistics are *arithmetic mean* (AM), *median*, and *mode*. The three formulae and descriptions can be found in Toutenburg and Heumann (2008).

AM is the standard average, mostly also simply called mean (Formula XXIV). AM can be used for interval (e.g. temperature in degrees Celsius with no absolute zero point) and ratio scales (e.g. values in per cent, temperature in degrees Kelvin with an absolute zero point). AM may be calculated on ordinal (ranks) data, too, but defining the meaning of such an AM may be difficult due to the fact that ranks do not have a defined interval width (1st rank may be defined *to be better* than 2nd, but *how many times better* is not determined). Besides the normal arithmetic mean, there also exist *weighted means*,

where a weight is assigned to the data points (x_i). Additionally it is also possible to cut off a certain percentage of highest and lowest values and calculate a so-called *trimmed mean* to account for outliers.

Another measure of central tendency more robust to outliers than simple AM is the *median*. It is defined as the value in the middle of a sorted list of digits. If the sorted list has an odd number of digits, the middle value is straight forward (e.g., the sorted list of digits {1, 2, **2**, 3, 4} has a median of 2). In a sorted list with an even number of digits, the median is the AM of the two values occurring in the middle (e.g. {3, **4**, **7**, 10} has the median $(4+7)/2 = 5.5$). This procedure is formalised in Formulae XXXI. The median can be calculated for ordinal, interval and ratio scales.

$$\text{XXXI} \quad \tilde{x}_{0.5} = \begin{cases} x_{((n+1)/2)} & \text{if } n \text{ is odd} \\ 0.5 * (x_{(n/2)} + x_{((n/2)+1)}) & \text{if } n \text{ is even} \end{cases}$$

The last measure for central tendency introduced is the mode. It is defined as that value in a list of items that occurs the most (e.g. in {1,1,1,1,2,2,2,2,3,3,3,4,4,4,4,4,4, 1 occurs 4 times, 2 occurs 5 times, 3 occurs 3 times, and 4 occurs 7 times, making 4 the mode of this list with a maximum of 7 occurrences). Unlike median or AM, the mode can also be calculated for nominal (categorical) data (e.g. a fruit basket holding apples, pears, strawberries, etc.). The mode is shown in Formula XXXII, where a_j is a feature characteristic (e.g. a fruit) and n_j is the number of occurrences of this feature characteristic (e.g. the quantity of this specific fruit in a basket).

$$\text{XXXII} \quad \bar{x}_M = a_j \Leftrightarrow n_j = \max\{n_1, n_2, \dots, n_k\}$$

4.3.1.2 Pre-Processing Procedure

Aggregation of all jobs into one big job for easier processing is accomplished using a Java-based script. Further processing is carried out with Microsoft Excel and Visual Basics for Applications (VBA). All judges that failed to identify the fake image (i.e. those that assigned a rank different from “irrelevant” to the fake image) are entirely removed from the file. After removal, between 11 and 24 of the initial 27 RJs (or between 10 and 19 out of 20 RJs for topics with a place name that required locational knowledge) are obtained and further aggregated. Aggregation measures are calculated (mode, median, AM). Because the aggregated RJs corresponded to ranks, it was chosen to use the median, which is the natural measure of central tendency for ranks. If the list contains an even number of digits leading to a median with a trailing “.5”, the value is rounded to the next higher number. Comparisons of median values with mode and rounded mean values reveal no large divergence. The so calculated final “median” ranks are then assigned to each image in one list for further processing.

4.3.2 Assessing Trustworthiness of CS Judges

Although many malicious judges may be removed with the aforementioned procedure, it does not mean that *all* of them can be eliminated. The problem therefore remains how the trustworthiness of

judges can be assessed *before* using the data for performance analysis. An idea is to measure the *correlation* between CS RJs and a trustworthy person's RJs. If there is a strong positive correlation, it may be assumed that the CS RJs are trustworthy, because they correspond to the trustworthy person's RJs. Because of resource limitations, the author of this thesis acts as a trustworthy person and assigns to each of the 612 images a rank between 0 and 3. Carterette and Voorhees (2011) even propose that the judge of the documents *ideally* is the person that created the topics. A short PHP website with a very similar layout based on a CrowdFlower job thus is created and provides almost the same environment for judgement like the assessors have. The only difference is that the website shows all the images at the same time, whereas one job in CrowdFlower only shows up to 5 images. Fake images are left aside for obvious reasons. Evaluation results are stored in an online MySQL database. The assessment site can be found on tinyurl.com/crowflow.

Furthermore, an appropriate correlation measure needs to be introduced to be able to evaluate the assumed relation. Correlation describes a relationship between two variables, although this relationship does not need to be causal. The data to evaluate is of ordinal scale (ranks). Spearman's rho rank correlation is a suitable method to assess a relationship between ranks (see Formula XXXIII, Toutenburg and Heumann 2008).

$$\text{XXXIII} \quad R = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)}$$

In Formula XXXIII, d_i is the difference of the ranks R_{x_i} and R_{y_i} of a pair (x_i, y_i) taken from two samples X and Y . n represents the number of data points. If R is 1, a perfect positive correlation (the more of X , the more of Y) is given and if R is -1, a perfect negative correlation (the more of X , the less of Y) occurs. What needs to be tested is if for each image, the same or a similar rank is assigned. This corresponds to a positive correlation, meaning if the crowd assigns a high rank to an image, so does the trusted person (same applies to low ranks). A negative correlation would be a problem, because it indicates that, if a high rank is assigned to an image by the crowd, the trusted person assigns a low rank, and the other way round. Therefore, the aim is to receive a high positive correlation close to 1. This would indicate that the crowd can indeed be trusted after removing bad judges and calculating an average rank as proposed before. Results will be presented in the next chapter.

4.4 System- and User-Centered Evaluations

After aggregation, the data is ready for analysis. A job for worldwide judges in average took around 2 hours to be finished (27 judgements for each image in the job, 2 to 5 images per job). The UK- and Ireland-only jobs, although requiring only 20 judgements per image and per job, took up to 3 days to complete each. The aggregated data can be found in Appendix I. The origin of the judges can be found in Appendix J. Before performance of the three different systems (T, TS, and TSCR) can be evaluated in

an SCE, the data *quality* needs to be assessed reasonably. In a last part, results contributing more to UCE than SCE shall be presented.

4.4.1 Data Quality Assessment

Figure 50 shows how useful the data obtained is in terms of how many contributors succeeded in finding the fake image. In average, 70.87% of the contributed RJs can be used for further evaluation after removing those contributors that failed to find the fake image.

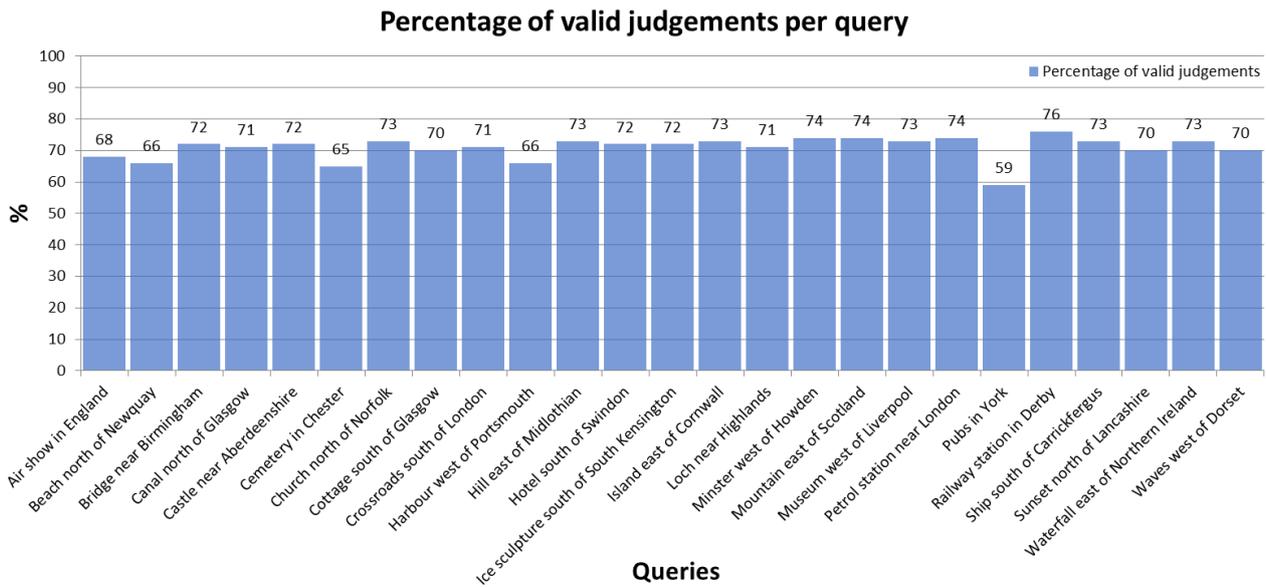


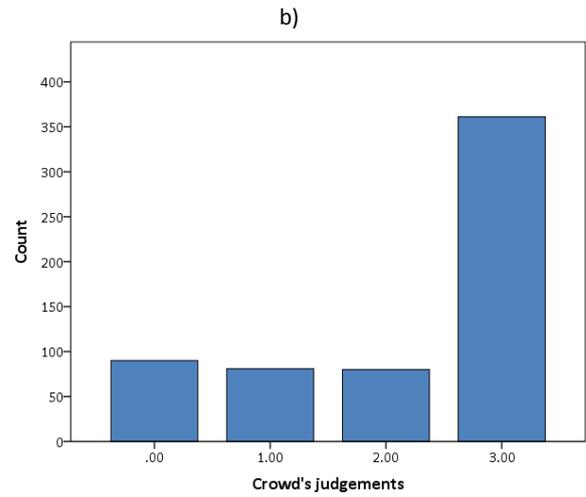
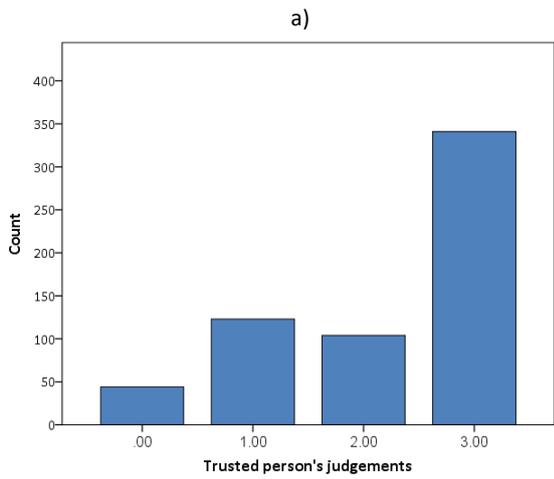
Figure 50: Percentage of assumed valid retrieved judgements per query/topic.

A further assessment is the correlation analysis based on Spearman’s rho rank correlation. Correlation coefficients are calculated in IBM SPSS Statistics 20. Table 22 shows the resulting correlation. The first part of Table 22 displays Spearman’s rho correlation coefficient, which depicts a high positive correlation of 0.872 (1.0 is the maximum) between RJs of the crowd and the trusted person. This means that if an image got a high rank by the trusted person, it also got a high rank by the crowd and the other way round if an image got a low rank by the trusted person, the crowd often assigned a low rank, too. The result is statistically significant, even on a 0.01 significance level (H_0 : there is no association between the variables is rejected). a), b), and c) show the distribution of the values, c) in a three-dimensional representation, which also shows the strong correlation. As can be seen in this image, most of the images’ rankings are the same. 165 out of 612 images were judged with a rank diverging by only 1 step from the trusted person’s ranking. No judgement of an image differed in more than 1 rank. This result indicates that, after removal of the obviously wrong answers of untrustworthy judges, the averaged RJs provided by CS judges can be used for further analysis.

Correlations

Spearman's rho		Trusted person's judgements	Crowd's judgements
	Correlation Coefficient	1.000	.872**
Trusted person's judgements	Sig. (2-tailed)		.000
	N	612	612
	Correlation Coefficient	.872**	1.000
Crowd's judgements	Sig. (2-tailed)	.000	
	N	612	612

** Correlation is significant at the 0.01 level (2-tailed).



c)

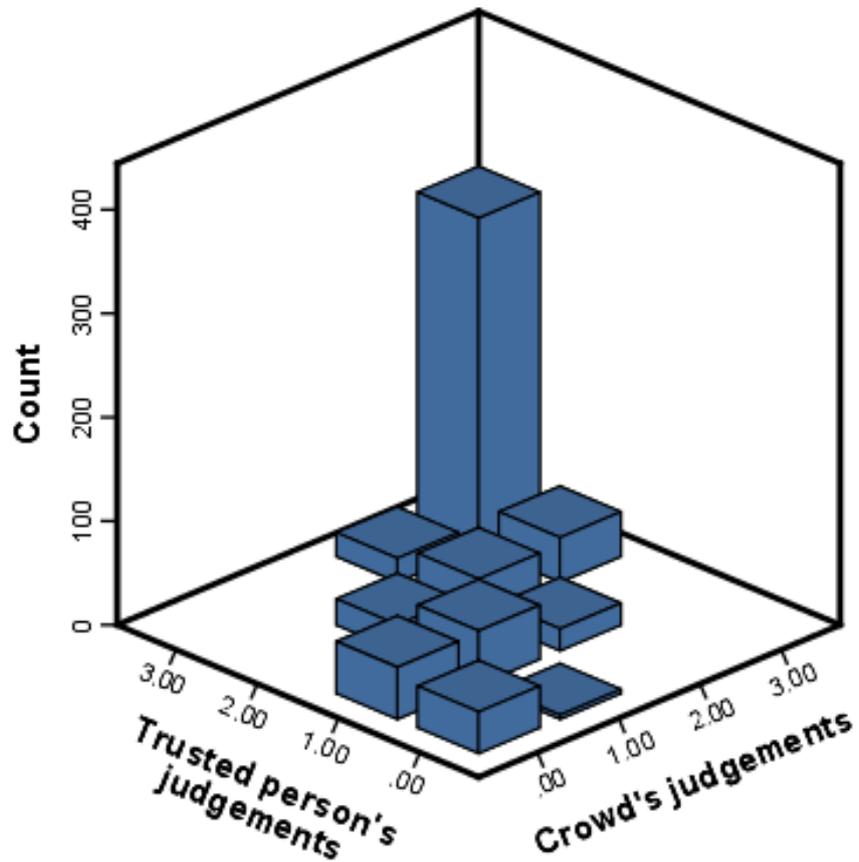


Table 22: Correlation analysis between median CS RJs and a trusted ranking.

See Appendix G for the complete aggregated RJ data set used to calculate the correlation.

4.4.2 Performance Assessment

4.4.2.1 Indexing and Retrieval Performance of the Main Algorithm

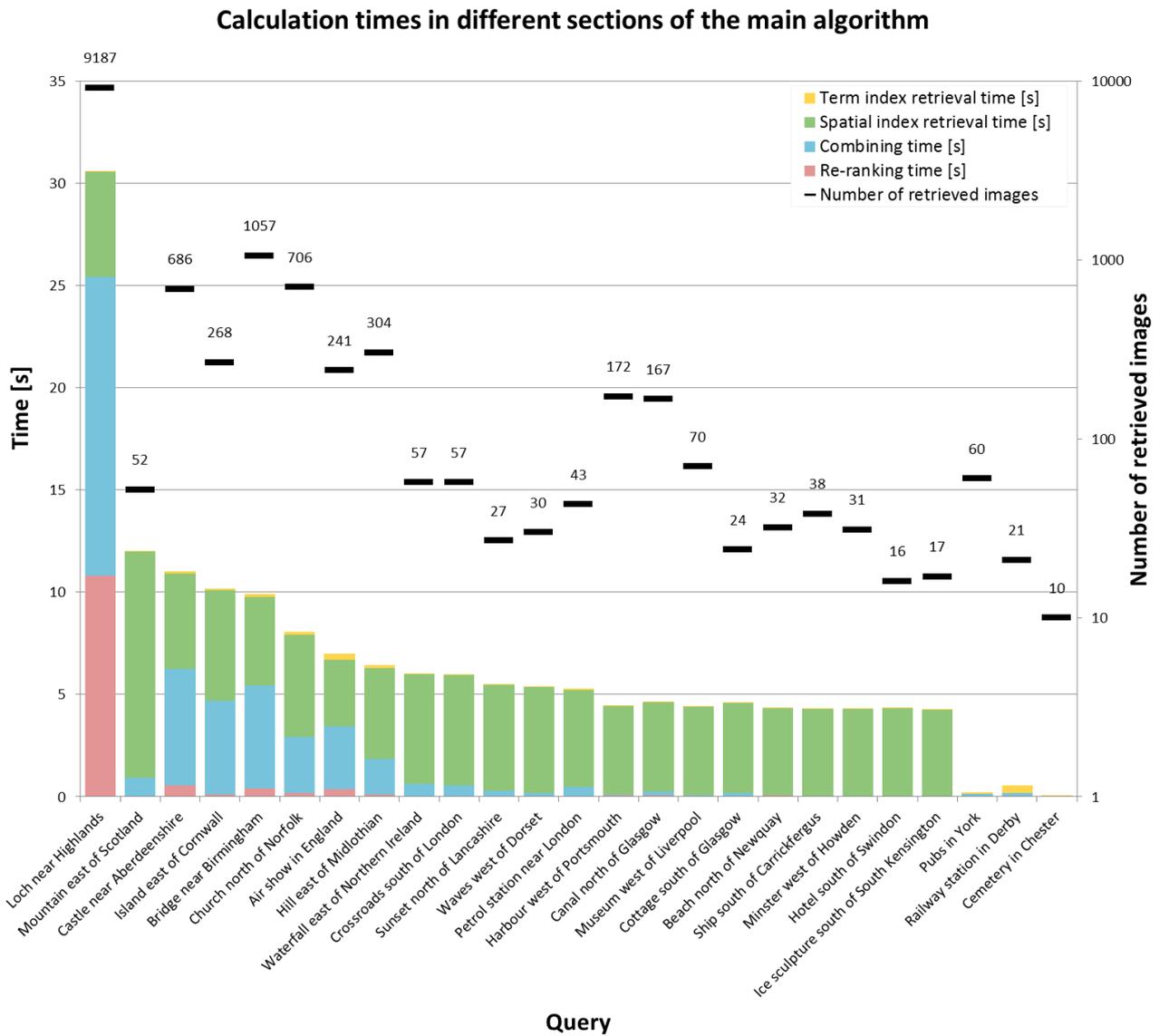


Figure 51: Calculation times extracted from the main algorithm.

The left vertical axis depicts times of the different algorithm parts in seconds, whereas the right vertical axis represents the number of retrieved images on a logarithmic scale with base 10. Black numbers correspond to the number of retrieved images as a visual aid. The horizontal axis denotes all the submitted queries. Therefore, one bar of four different colours represents the complete retrieval time for one submitted query (lacking some minor, insignificant calculations needed for retrieval).

A first performance analysis shall focus on the actual main algorithms query processing times. Although an algorithm may produce a perfect retrieval result, if it is not applicable due to very high processing times, it may never go operational. Figure 51 gives an insight into calculation times of the main algorithm. The algorithm’s times are exactly the same for TS and TSCR, the only difference being the additional time needed to re-rank the images in the TSCR system. First of all, the Lucene-based term index querying does scarcely increase retrieval time. It is always below half a second. This corresponds also to the time needed for retrieving images in the T system, which only uses such a term

index. On the other hand, spatial retrieval time using PostgreSQL/PostGIS indexes heavily depends on the used algorithm and query footprint size, although most of the calculations take around 5 seconds to be conducted. Inside relationships are simple to calculate. Therefore, in 3 out of 4 queries using the inside relationship, spatial retrieval time is below one hundredth of a second. One exception is the query for “air show in England”, which has a large query footprint representing England. The same can be seen for the other relationships. Although there is barely any difference between calculation times of linear near and directional relationships (N/S/W/E of), the query “mountain east of Scotland” has the highest spatial retrieval time of 11 seconds (and also a large query footprint encompassing whole Scotland). Combining times of term and spatial results using intersection depends on the number of images retrieved by both indexes: the more images in *both* lists, the longer the combining time. Consequently, in terms of retrieval times, it is desirable to have one index, either T or TS, to retrieve only *few* relevant images, resulting in lower intersection times. Also dependent on the number of images retrieved are the re-ranking times. However, comparably, they are quite low. This has to do with the drastically decreased image set on which the CBIR querying needs to be conducted. The decreased image set also has implications on the re-ranking: neither the cluster algorithm nor the re-ranking itself increase re-ranking times noticeably. In summary, most time is spent on spatial retrieval, followed by combining and re-ranking times. No weight carries term index retrieval.

4.4.2.2 Performance Evaluation of the Systems’ Ability to Estimate Relevance

Three systems are being evaluated, one of which (T) acts as a baseline to enable a relative performance evaluation to compare this baseline to the other systems (TS and TSCR). Due to the fact that there exists no adequate test collection for a SPAISE with RJs for all images, only the *top-10* ranked images of each system are pooled together and evaluated. Therefore, precision, AP/MAP and NDCG (introduced in chapter 2.6.2 Evaluating a System’s Ability to Estimate Relevance) are all limited to the first 10 images. This corresponds to an often applied rank cut-off, especially for precision. n of $P@n$ equals 10, resulting in $P@10$. Although such a procedure may miss relevant images occurring later in the ranking, it still makes sense to do so: the hope is that TSCR will especially increase the relevance of the top ranked images compared to TS. Thus, already in the first 10 images, a tendency should be visible towards better or worse retrieval results. This is also the reason why only maximally 5 EIs were used for re-ranking, so that not only the images used for re-ranking are represented in the top-10 because they most likely are similar to those of the TS system (the top-20 images of the TS system are used as CIs to choose EIs from). The same procedure is applied to AP/MAP and NDCG: to all of these measures, a rank cut-off of 10 is assigned. Thus, these measures will be called AP@10/MAP@10 and NDCG@10. As a consequence of not evaluating all images’ relevance, no recall measures can be extracted.

P@10. To be able to calculate P@10, the ranks of the initial RJs (irrelevant = 0, marginally relevant = 1, fairly relevant = 2, highly relevant = 3) need to be mapped to (0 = irrelevant, 1 = relevant), because

precision is a binary measure as described in 2.6.2.1 Precision and P@n. Three mapping schemes may be derived therefrom, dependent on how strict or loose the assignment is intended to be. The three possibilities are listed in Table 23.

	Mapping	Explanation
1)	(1, 2, 3) = (1), (0) = (0)	Least restrictive mapping. Assumes all images to be relevant, if at least one aspect of the image is considered relevant.
2)	(2, 3) = (1), (0, 1) = (0)	Takes small variations in RJs into account, so that also images not a hundred per cent correct are considered relevant.
3)	(3) = (1), (0, 1, 2) = (0)	Most restrictive mapping. Assumes only highly relevant images to be considered relevant.

Table 23: Possible mappings from (1, 2, 3, 4) to (0, 1).

Table 24 shows P@10 values for the three systems and the 25 topics. It indicates differences in statistical parameters AM, *standard deviation* (SD, the square root of the variance), range, minimum and maximum values of P@10 calculated with different mappings from Table 23.

	AM	SD	Range	Minimum	Maximum
P@10 T with ranks 1,2,3	.83	.30	1.00	0.00	1.00
P@10 TS with ranks 1,2,3	.97	.08	.32	.68	1.00
P@10 TSCR with ranks 1,2,3	.98	.06	.21	.79	1.00
P@10 T with ranks 2 and 3	.70	.29	1.00	0.00	1.00
P@10 TS with ranks 2 and 3	.89	.14	.43	.57	1.00
P@10 TSCR with ranks 2 and 3	.92	.12	.48	.52	1.00
P@10 T with rank 3 only	.38	.33	1.00	0.00	1.00
P@10 TS with rank 3 only	.73	.26	.90	.10	1.00
P@10 TSCR with rank 3 only	.77	.24	.80	.20	1.00

Table 24: Statistical values calculated for the different variations of P@10.

The statistical parameters indicate the following: systems do not change in AM or SD in relation to each other with different mapping schemes. AM in system T is always much lower than in system TS and TSCR, and TS is always lower than TSCR. Conversely, TSCR has the lowest SD, meaning that 68.2% of all values are closer around the mean value for TSCR than is the case for systems TS and T. Again, in the middle is system TS, having a SD that is always higher than that of system TSCR but also always lower than the SD of system T. The maximum values for each system are 1.0, which shows that each system is able to retrieve at least for one topic the full number of 10 relevant images in the first 10 hits. The minimum, on the other hand, is 0.0 in each configuration for system T, meaning at least one topic/query did not retrieve any relevant image in the top-10 list of T. Therefore, system T always has a maximum range of 1.0. This is also indicated by the high SD being always around 0.3. Better performances show systems TS and TSCR. Their SDs increase steadily with smaller numbers of images

considered relevant. Although TSCR always has a smaller SD than TS, the difference is only 0.02. If the minimum value of each TS and TSCR configuration is taken into account, it can be seen that for configuration 1 and 3, TSCR has a smaller range (0.1 or 10% less than TS). Only for configuration 2, TS (0.43) has a smaller range than TSCR (0.52, a difference of 0.05 or 5% smaller). Naturally, range increases for both systems TS and TSCR with fewer images considered relevant per topic. Consequently, P@10 decreases when reducing the number of images that are considered relevant.

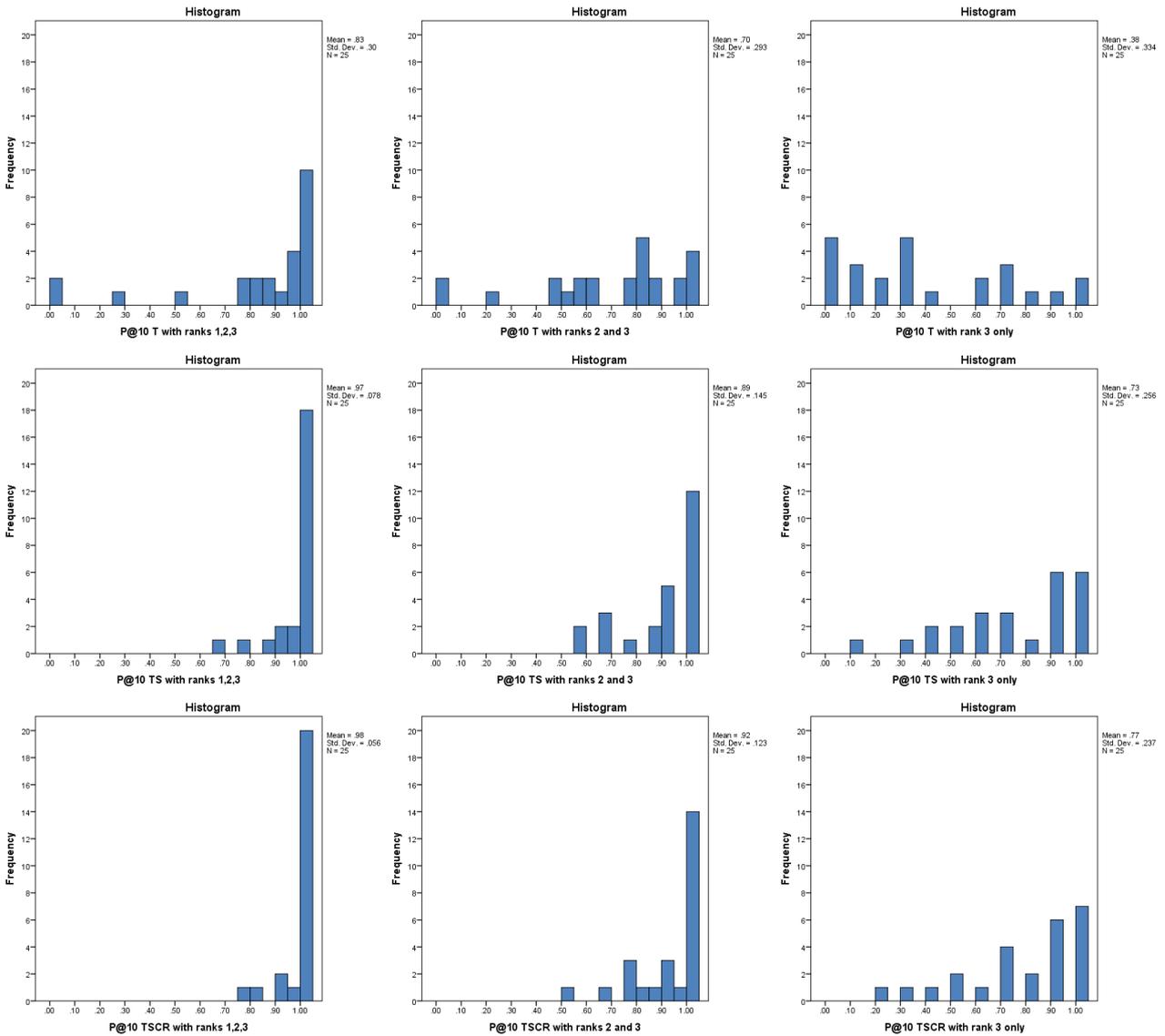


Figure 52: Histograms for P@10 values of the three systems (T, TS and TSCR).
 Bars summarise values of P@10 in a 0.05 range.

Visualising the data with histograms reveals additional information about the three mapping schemes of P@10 in Figure 52. The illustration shows how P@10 highly varies for the term-only system with decreasing number of images considered relevant. The same can be seen for the other systems, but not as much as the term-only system. Additionally, the TSCR system shows a slightly steeper and less varying distribution of P@10 values compared to the TS system when considering only relevant images that have an assigned rank of 3 (highly relevant).

AP@10/MAP@10. AP@10 is derived from P@10 values and calculated for the strongest cut-off of P@10s according to Table 23 3). This assumption follows an evaluation conducted by Bailey et al. (2008). Table 25 provides AP@10 statistics.

	MAP@10	SD	Range	Minimum	Maximum
AP@10 T with rank 3	.57	.39	1.00	0.00	1.00
AP@10 TS with rank 3	.81	.18	.66	.34	1.00
AP@10 TSCR with rank 3	.85	.21	.71	.29	1.00

Table 25: Statistical values calculated for the different variations of AP@10.

Similar relative relationships between the three systems as seen for P@10 (configuration 3 in Table 23) can be observed. TSCR has a higher MAP@10 than TS, which in turn has a higher MAP@10 compared to T. T’s SD is also double as high (0.39) as the SD of the two other systems (0.18 and 0.21 respectively). TSCR has a slightly higher SD than TS (0.18 compared to 0.21). Therefore, also the range of TSCR is slightly higher than the one of TS (0.29 to 1.0, resulting in a range of 0.71 compared to TS’s range of 0.66 in the interval of 0.34 to 1.0). Also, the range of T varies much more compared to the other two systems (from 0.0 to 1.0).

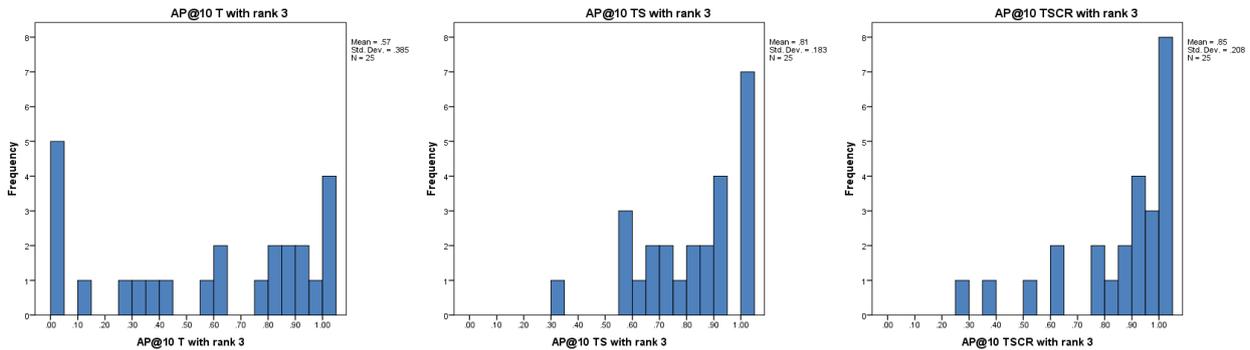


Figure 53: Histograms for AP@10 values of the three systems (T, TS and TSCR).

The histograms in Figure 53 visualise the distribution of the AP@10 values. It can be seen that T has many AP@10 values between 0.0 and 0.04, indicating low retrieval performance. TS and TSCR show more AP@10 values of 1.0, and again the same steeper curve with more high values of TSCR compared to TS can be observed. TSCR on the other hand shows a slightly wider range, but more AP@10 values of 1.0 compared to TS. No values in the range of 0.95 to 0.99 can be found for TS, however.

NDCG@10. NDCG, as described before, puts emphasis on retrieving highly relevant documents. If NDCG@10 is close to 1.0, the observed ranking of the images is very close to a potential *ideal* ranking of the retrieved images. Table 26 shows the AM as an overall value to describe the effectiveness of a system, which is also called MANDCG elsewhere (Palacio et al. 2011). All NDCG@10 values show high AMs, indicating high correlations between ideal and observed ranking. Again, $AM_{TSCR} > AM_{TS} > AM_T$, and $SD_{TSCR} < SD_{TS} < SD_T$, as could already be seen for P@10. SD of T is rather high (0.28 or almost 1/3 of

its range), whereas the SD of TS and TSCR are very small (only 0.073 and 0.066, respectively). T’s NDCG@10 values ranges from 0.0 to 1.0 as expected from P@10 and AP@10. Ranges of TSCR and TS only show slight differences of 0.02 and are rather narrow (from 0.75 to 1.0 (0.25) and from 0.77 to 1.0 (0.23), respectively).

	AM	SD	Range	Minimum	Maximum
NDCG@10 T	.81	.28	1.00	0.00	1.00
NDCG@10 TS	.94	.073	.25	.75	1.00
NDCG@10 TSCR	.96	.066	.23	.77	1.00

Table 26: Statistical values calculated for the different variations of AP@10.

Comparing the histograms in Figure 54, the highest range for NDCG@10 values can be found for T, but only few values actually range so far below where no image could be found at all. Again, a steeper curve around 0.9 to 1.0 can be observed for TSCR compared to the TS. This corresponds to most of the findings in P@10 and AP@10.

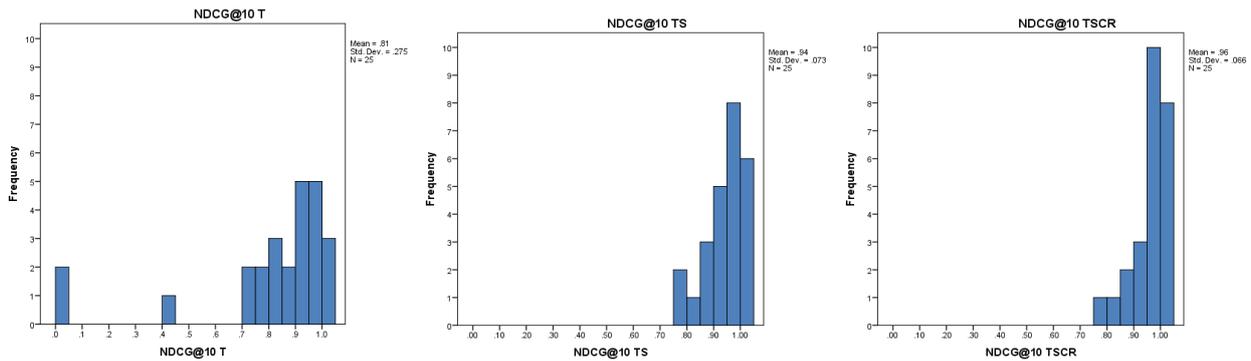


Figure 54: Histograms for NDCG@10 values of the three systems (T, TS and TSCR).

Table 27 summarises the mean values of all each measure for all systems. As indicated before, all mean values of the different measures are highest for TSCR. However, TS achieves only marginally smaller mean values. T is always lower than both TS and TSCR, having mean values that are at least smaller by the amount of 0.13. The next subsection will evaluate the statistical significance of the observed differences between the mean values.

Measure	T	TS	TSCR
Mean P@10 (3)	0.38	0.73	0.77
Mean P@10 (2, 3)	0.70	0.89	0.92
Mean P@10 (1, 2, 3)	0.83	0.97	0.98
MAP@10 (3)	0.57	0.81	0.85
NDCG@10	0.81	0.94	0.96

Table 27: Summary of all mean measures for all the systems.

Bold numbers indicate the system with the highest value found for this measure.

T-tests. As could be seen in the previous part, all measures show very similar relative results for the 3 systems. To check for statistical significance therefore, only one measure is chosen to apply mean comparisons. The measure selected is AP@10 with only “highly relevant” images (rank 3), because it directly compares MAP@10. MAP is one of the most common measures used in information retrieval literature. It therefore makes sense to compare these values, although it has to be kept in mind that here only AP for the top-10 images are calculated, not overall AP values. Direct comparability with literature values thus may be limited.

One-sample Kolmogorov-Smirnov tests reveal all the three samples of MAP@10 (for T, TS and TSCR) to be normally distributed, thus Student’s paired-samples *t*-test (2.6.2.4 Student’s paired-samples *t*-test) can be applied safely without violating any prerequisites. Each system is compared to each other system, resulting in 3 paired-samples *t*-tests. A 0.05 significance level needs to be undercut for statistical significance. H_0 for each test is: the difference between MAP@10 of both systems differs from 0 (two-tailed). Table 28 shows the results. T and TS ($0.012 < 0.05$) as well as T and TSCR ($0.004 < 0.05$) significantly differ in terms of MAP@10. TSCR’s MAP@10 differs from T’s MAP@10 significantly even on a 0.01 significance level. Consequently, T has a smaller MAP@10 than the other two systems and is therefore not as effective in retrieving relevant images. No statistical significance, however, can be found between the TS and TSCR in terms of MAP@10 ($0.533 > 0.05$), although a higher MAP@10 for TSCR compared to TS could be observed. Therefore, the higher MAP@10 (meaning, better performance/effectiveness for retrieving relevant images) of TSCR may be completely arbitrary.

		Paired Samples Test							
		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Dev.	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	AP@10 T – AP@10 TS	-0.241	0.441	0.088	-0.422	-0.059	-2.727	24	0.012
Pair 2	AP@10 T – AP@10 TSCR	-0.271	0.426	0.085	-0.447	-0.096	-3.194	24	0.004
Pair 3	AP@10 TS – AP@10 TSCR	-0.031	0.245	0.049	-0.132	0.070	-0.633	24	0.533

Table 28: Paired-samples *t*-test applied to each pair of systems.

4.4.3 Qualitative Evaluations

4.4.3.1 Topic-Wise Analysis using P@10

A further interesting point to evaluate is for which of the queries which system performed best. Such an analysis is intended to give insights into system-specific strengths and weaknesses and therefore, suggestions on when to use which system shall be derived. The analysis is based on P@10 values calculated using the mapping scheme of Table 23 3) summarised in Table 29.

Topic/Query	T	TS	TSCR
Railway Station in Derby	0.7	0.6	0.5
Minster west of Howden	1	0.9	0.7
Museum west of Liverpool	0.7	1	0.5
Church north of Norfolk	0.3	1	0.9
Crossroads south of London	0.2	0.6	0.3
Petrol station near London	0.3	1	0.7
Air show in England	0.4	0.7	1
Canal north of Glasgow	0.1	0.5	0.9
Cottage south of Glasgow	0.1	0.5	0.9
Harbour west of Portsmouth	0.8	0.9	1
Hill east of Midlothian	0.3	0.4	0.6
Hotel south of Swindon	0.1	0.7	0.8
Mountain east of Scotland	0	0.1	0.2
Ship south of Carrickfergus	0.7	0.8	1
Sunset north of Lancashire	0.2	0.9	1
Waterfall east of Northern Ireland	0	0.3	0.8
Waves west of Dorset	0.6	0.6	0.7
Beach north of Newquay	0.9	1	1
Castle near Aberdeenshire	0.6	0.9	0.9
Cemetery in Chester	0	0.7	0.7
Ice sculptures south of South Kensington	0	0.9	0.9
Island east of Cornwall	0.3	0.4	0.4
Loch near Highlands	0	1	1
Pub in York	0.3	0.9	0.9
Bridge near Birmingham	1	1	1

Legend

- T performed best.
- TS performed best.
- TSCR performed best.
- TS and TSCR performed same and best.
- T, TS and TSCR performed same.

Table 29: Comparison of topic performance in terms of P@10 for T, TS and TSCR.

Bold numbers indicate which of the systems performed best in retrieving images for this query/topic.

Colours separate different rankings to visually indicate which system retrieved most relevant images.

Where T performed best. In two cases, P@10 is highest for T. The topics are “railway station in Derby” and “minster west of Howden”. In the first query, T clearly profits from a railway station that was actually named after the location. Therefore, the word “Derby” can be found quite often within the assigned title/description. Because the other systems do not consider “Derby” in their textual

retrieval, but only “railway station”, the TS and TSCR do not necessarily retrieve the right railway station in Derby, but any railway station, abandoned or not, which may diminish the relevance for a user. TSCR may additionally re-rank irrelevant images already retrieved by TS, leading to a worse re-ranking of the initial result list. The same happens in the case of “Minster west of Howden”. There *is* already a very famous minster in the west of Howden. Thus, T does not need to evaluate “west of” at all, but can only retrieve all the images containing “minster” and “Howden” in their textual descriptions. TS, however, has to look for minsters located in the region defined as being west of Howden (dependent on the system’s implementation of that directional relationship). Furthermore, there are sometimes only parts of the minster visible or even nothing at all, and the re-ranking of TSCR consequently changes the TS’s ranking for the worse. Therefore, clearly described, specific topics with well-known, larger and unambiguous locations are, in the case of the used configurations and image collection, easier to retrieve by only using a simple term-based index compared to more sophisticated approaches.

Where TS performed best. Examples typical for TS to perform better than the other two systems include unspecific topics with a visually hard to distinguish image theme. Notable are “Crossroads south of London” and “Petrol station near London”, but also “Museum west of Liverpool”, all having a difference larger than 0.3. T in this case may identify images of crossroads and such that are taken in London, but can barely tie those words together with “south”. “Of” as part of “south of” is discarded anyways because it is a stop word. This leads to many images only partly fulfilling the query. However, such partly relevant images are discarded afterwards when only highly relevant images (rank 3) are considered relevant for P@10 calculations. On the other hand, TSCR accomplishes content-based re-ranking with global features based on colour and texture and is therefore barely able to distinguish a crossroad on an image (due to its unspecific colour and texture properties). Consequently, re-ranking of the images retrieved by TS does not lead to better results with such queries. The TS system performs better than TSCR if the content of the query is not clearly distinguishable in the image and the topic is general, not specific.

Where TSCR performed best. Examples of TSCR performing better than the other systems with relatively large differences (more than 0.3) in terms of P@10 can be found for the topics “Air show in England”, “Canal north of Glasgow”, “Cottage south of Glasgow” and “Waterfall east of Northern Ireland”. In all cases, the images retrieved have relatively clearly distinguishable, similar colour and/or texture features visible in the image, which can effectively outperform textual only descriptions of T, but also TS. TS e.g. retrieved an image showing a path to a waterfall. Thus, the term index part of TS ranked this image high due to the term “waterfall”. However, only TSCR is able to find images of actual waterfalls, not only of waterfall descriptions, as a result of the application of the clustering and re-ranking algorithms. Because various images actually show a typical waterfall but are not ranked that high because their textual and local descriptions did not distinguish them to be *that* relevant to the

query, the unsupervised cluster analysis can group those images together. Because these waterfall images are closely related to each other in terms of visual features, the following re-ranking places them into a higher rank than the other two systems. Thus, images with visually clearly distinguishable and characteristic features are retrieved well by the TSCR system.

Where TS and TSCR performed same and best. Some rankings did not get worse or better by applying the CBIR re-ranking. This is the case for queries retrieving many relevant images (e.g. “Pubs in York”, “Beach north of Newquay”, or “Loch near Highlands”), or very few relevant images (e.g. “Cemetery in Chester”, “Ice sculptures south of South Kensington”). Additionally, for these queries, T performs often much worse. Examples are e.g. “Pubs in York”, where many images containing pubs are returned that are not located in York, or for “Cemetery in Chester”, many images of cemeteries are returned that are not located in Chester. This has to do with the used Boolean OR operation, which unites the results of each term of the query. Ambiguous place names like “Aberdeenshire” or “Highlands” barely mentioned in the images’ textual descriptions, fall into this category, too, where the T system could not compete with the other two systems.

Where T, TS and TSCR performed same. Only one example, namely “Bridge near Birmingham”, allowed all the systems in question to perform equally well. There are many bridges near Birmingham, and therefore, both words “bridge” and “Birmingham” appear many times together in an image’s textual descriptions. Furthermore, any kind of bridge is considered relevant, and also barely any restriction is made to how close or far away a bridge has to be located in the picture (close bridges are not necessarily considered more relevant than images of bridges further away). Therefore, a topic like this, which is so multi-faceted and barely retrieves any irrelevant images, should have been discarded beforehand and is an example of a badly formulated topic.

4.4.3.2 Analysis of CS RJs comments

To get an idea of how CS judges assess images, the comments provided on submission of a CrowdFlower job are summarised and analysed below.

“The image fulfils the topic better than another image in the same job”

This comment indicates a certain ranking which was not intended while creating the topics. To what extent this “learning effect” biases the results cannot be estimated in retrospect. It may not have had such an influence on the people’s judgements in general, because it only occurs in very few comments.

“Although the image fulfils the theme and location, it is too far away to be highly relevant”
“The image is located northeast, not north of XY”

Both comments indicate flaws with the used algorithms. The first points to the parameter settings that did not fulfil the user need, indicating wrongly chosen distance factors (see Table 18). Furthermore,

some judges made a clear distinction between e.g. north and northeast, bespeaking the need of a finer distinction between those directional relationships.

“Zooming into the photo may reveal different aspects relevant to the topic”

Such comments occur e.g. for the topic “Bridges near Birmingham”, where only a sign is visible, but on closer inspection, the sign mentions something about a bridge, making it more relevant to the topic.

“It is a sign of a XY”

This answer occurs often together with the queries “Bridges near Birmingham”, “Cottages south of Glasgow” and “Pubs in York”. Examination of the assigned ranks reveals that for some judges, this comment means that the picture is highly relevant because it belongs to the object looked for in the query. On the other hand, some assessors judge such an image to be completely irrelevant due to the fact that the image shows a sign, not the actual object in question.

“There is no XY visible, but the text says there is one”

Such comments point out the trustworthiness of judges, because they actually read the text to judge the image and not only rank the image according to what they see in it.

“Right area, but wrong subject”

“Right location, but only a part of the subject in question”

“The subject/location is wrong”

These comments show that judges make the desired and also clearly indicated distinction between an image’s theme and its location mentioned in the task description. Before conducting the experiment, it was assumed that many people would disregard this distinction. Therefore, the comments prove the structure (with a map and locational information) and task description to be effective in at least the examinable cases. Judges also distinguish between the various ways a subject may be displayed in an image, as the second comment reveals. Therefore, they examine the images thoroughly to evaluate if the topic’s theme part is covered completely or if nothing at all is visible, and only the text mentions it. However, a possible flaw in the job design emerges sometimes together with those comments: judges may consider an image to be completely irrelevant if only one part of the topic (either theme or location) is wrong, which is clearly stated in the task description to be judged with a lower ranking (e.g. marginally relevant) and not with irrelevant.

"Test question!"

At least two judges explicitly recognised that one of the images planted in a job had intentionally nothing to do with the topic and was placed there only to trick unreliable judges not thoroughly reading the task description.

5 Discussion

It is time to go back to the initially stated *research questions* (RQ) to evaluate if the proposed hypotheses hold or if the results lead to different perspectives on the problems. Each RQ will be individually elucidated and discussed.

5.1 Research Question 1

RQ₁ *Can an approach combining textual and spatial features outperform a text-only approach for retrieving images of queries with spatial relevance?*

H₁ *A combination of textual and spatial dimensions leads to better retrieval results in the case of images.*

5.1.1 Performance Analysis

Results have shown that in terms of mean P@10, MAP@10 as well as mean NDCG@10, an approach explicitly incorporating the spatial dimension can increase retrieval performance for queries having a spatial part. In terms of AP@10, this result is also statistically significant on a 0.05 significance level ($0.012 < 0.05$). H₁ therefore can be verified: *an additional incorporation of the spatial dimension leads to better retrieval results*. This outcome corresponds to results found in the literature, where an explicit spatial dimension increases retrieval performance for websites (Purves et al. 2007) and text documents (Palacio et al. 2011) with spatial relevance. Although there are many images with titles or descriptions defining regions or place names, in the case of such a descriptor, text-only methods fail to retrieve any spatially relevant image for a query. Indexing and retrieval in the case of images can therefore definitely profit from the addition of a spatial index with spatial retrieval methods.

5.1.2 Performance of Online Location Retrieval Tools

However, images without GPS coordinates cannot profit from such a spatial index. Online services like YPM and GN were shown to effectively geo-parse and -code place names from texts. In the case of YPM, its abilities to relatively precisely extract and disambiguate place names and to assign an MBR to them work very well for query processing. Even vernacular place names like “Highlands” without well-defined extents retrieve relevant spatial footprints, at least for the investigated locations in the UK and Ireland. Furthermore, online querying for these geometries does not noticeably increase retrieval time ($t_{\text{Retrieval}} < 0.5\text{s}$). Although Tobin et al. (2010) showed that YPM can be relatively inaccurate (also compared to GN), according to the gathered RJs, this inaccuracy may not have a large impact on a user’s judgement of an image’s relevance to a spatial query.

Although only used in experiments, GN can retrieve various locations having any relation to the location name. However, the fact that GN lacks the ability to retrieve an MBR makes it less suitable for query processing, although there could be calculated a buffer around the retrieved point locations. An MBR, though, provides a much more accurate base for estimating near distances and inside

relationships. GN is better suited to extract and geo-reference place names from texts, where any ambiguous notation may indicate the true identity of a location. Unfortunately, GN's MBR approximation algorithm based on a linear regression could not be tested in the CS evaluation.

5.1.3 Performance of Spatial Footprints and Algorithms

MBRs provide a suitable approximation for spatial query footprints in many cases. However, as already shown in Frontiera et al. (2008), comments submitted by judges indicate that some images are, although closely located, not actually *inside* the query place in the case of the inside relationship. The consequence is a trade-off between geometrical accuracy, acceptable storage costs, and computational power requirements. However, CS RJs and comments indicate that MBRs (as concluded in Cai 2011) seem to provide a suitable geometrical approximation for query footprints, and also for retrieving spatially relevant images.

Some comments indicate a revision of the chosen *near* relationship/its distance factor. Although the distance considered near in this system already depends on the size of the location found in the query (on the half diagonal from the centre point to an edge of the MBR, similar to suggestions in Purves et al. 2007), the distance factor with which this near relationship is multiplied is chosen relatively arbitrarily (1.1 for the near relationship, 1.5 for the directional relationships). A more empirically grounded estimation may provide better approximations of what users consider to be near.

Several assessors, though, suggest that *more* directional relationships are desirable, e.g. northwest, adding a finer resolution of where an image may be located. Judges, and therefore most likely also users, seem to not view "first order" directional relationships (N/S/E/W) as a superset of the "second order" directional relationships (NE/NW/SE/SW), but consider them equally relevant. Such implementations could be added to existing code rather easily, making geometrical directional estimations again suitable for this task.

Retrieval times seen in Figure 51 are a point of concern. Most of the overall retrieval time resulted from querying the spatial PostgreSQL/PostGIS index. It is therefore advised to carefully implement and test such algorithms to avoid the loss of retrieval speed. Retrieval times for combining results from term and spatial indexes using an intersection strategy drastically increase with the number of retrieved images. This observation corresponds to what already has been demonstrated for separate indexes in Vaid et al. (2005). However, one should not forget that it may be a strategy to *not* intersect, but unite two sets, especially if too few images were retrieved by either the term or spatial index. Furthermore, separate index structures provide the possibility to experiment on different combination strategies, as Martins et al. (2005) already suggested. Such rather simple exchanges of the combining strategies may be difficult in the case of interwoven term and spatial indexes. Moreover, it was not the goal to study efficiency here, but to increase effectiveness of retrieval results.

5.2 Research Question 2

RQ₂ *Can a PRF re-ranking approach, which uses hierarchical clustering and low-level global image features and is applied on a result list retrieved through textual and spatial methods, outperform both text-only and text-spatial-only approaches for retrieving images for spatial queries?*

H₂ *By incorporating low-level global image features the retrieval performance of spatial queries can be increased even more than by text- or text-spatial-only retrieval methods because a third relevance dimension, especially important for images, is included.*

5.2.1 Performance Analysis

In terms of mean P@10, MAP@10 and NDCG@10, the re-ranking strategy outperforms both text-only and text-spatial retrieval strategies. However, although TSCR outperforms the T system significantly ($0.004 < 0.01$ significance level) in terms of MAP@10, no such significance can be observed for a comparison between TSCR and TS. Therefore, the re-ranking may only perform better by chance. As a consequence, H₂ cannot be verified. Nevertheless, a *tendency* towards better retrieval results when using TSCR is still apparent if Table 29 is examined. There it is clearly visible that more queries performed best with TSCR, followed by TS. Furthermore, both TS and TSCR clearly provide a better subset of retrieved images from the whole image collection than the T system.

5.2.2 Problem Identification

A possible problem may be the top-*K* example images chosen for re-ranking. Although 20 candidate images are evaluated and only a maximum of five of those images are finally elected to be used for re-ranking, there is no guarantee that the number of example images is not too small. A more dynamic approach for estimating the number of candidate or example images, like 30% in Popescu et al. (2009) or the approach presented in Arampatzis et al. (2009), could provide a better subset for re-ranking. Further parameter tuning therefore may lead to better re-ranking results.

Moreover, the definition of directional relationships may contain flaws. If an image is assigned a higher rank through re-ranking, although it is spatially not as relevant to the query as another one (e.g., because it is positioned on the borders of the N/E/W/S cones defining the directional relationship), people may experience this image to be *less* relevant. It was already shown in section 5.1 Research Question 1 that judges make a distinction between finer resolutions of cardinal directions. The re-ranking therefore favours images that are similar by means of low-level features, but discards the relevance assessed by textual and spatial queries (due to the semantic gap as described in the introduction). Having more cardinal directions may therefore limit the decrease of spatial (as well as thematic) relevance but, on the other hand, may also decrease the number of retrieved relevant images. However, the decreased angle of directional relationships from 180° to 90° maximum limits also the set of images not considered *that* spatially relevant.

From a CBIR point of view, however, approaches where global low-level feature analysis is only applied on a drastically reduced and thematically more relevant image subset (as shown here and in Arampatzis et al. 2009, Popescu et al. 2009, and Maillot et al. 2007), proves to be an efficient way to avoid the problem of re-ranking all images of the collection and simulating a “term-index-like” behaviour, where also only the relevant terms are used for ranking, not all terms.

5.2.3 Suggestions on Improving TSCR

Although it was not statistically significant, the result is also not completely crestfallen. TSCR still tends to perform better than the other two systems. Therefore, possible adaptations to the algorithm may reveal significantly better retrieval results. What could be done is:

- 1) Parameter tuning: differently chosen distance factors for spatial near relationships, weightings for different dimensions and different numbers of candidate and example images, dynamically or statically estimated, may alter the result already significantly.
- 2) While Arampatzis et al. (2009) did not recommend their re-ranking approach to be used with local low-level features it may still be an idea to try it anyways due to the fact that they did not incorporate a spatial dimension. This additional dimension may alter their results to the better or the worse. The question remaining is if local features are applicable for such a large image collection. Indexing global features in the form of JCD already took more than 18 hours (Table 19). However, as Figure 51 shows, also local features may heavily profit from the reduced image set retrieved through the term and spatial dimensions, which may limit retrieval times to an applicable minimum.
- 3) A last idea is to provide re-ranking, but to not automatically re-rank the initially retrieved images. Even more, instead of PRF, normal RF (see e.g. Carbonell et al. 1997) could be provided, where the user may choose some of the images from an initial result list for re-ranking. Consequently, the clustering algorithm implemented would become obsolete. However, the actual idea behind PRF is that users *do not* interact with the system, but the system itself automatically achieves an improvement of the initial results.

5.3 Research Question 3

RQ₃ *Can relevance judgements gathered through crowdsourcing be combined with traditional evaluation techniques (e.g. P@10) to act as a valuable replacement of human assessors for the evaluation of a SPAISE?*

H₃ *Relevance judgements gathered through crowdsourcing are a viable, quick and inexpensive replacement for known assessors to evaluate a SPAISE using traditional measures, provided certain quality measures are applied.*

5.3.1 Dealing with a Crowd

First of all, traps within the tasks submitted to CrowdFlower in form of images that have nothing to do with the topic to evaluate are a very useful addition to rule out malicious judges. This was already shown to be effective in Zhu and Carterette (2010) for Amazon's Mechanical Turk. However, it would be much easier if CrowdFlower provided the possibility to flag a unit simply as a trap and to ban users that fail to find this trap from participation. Neither is it possible to ban certain IP addresses after retrieving results of a trial job. This results in a monetary overkill:

- 1) Trap images are units. Therefore, 163 images (one per job, resulting in $163 \cdot 0.05 = 8.15$ \$ US-Dollars) are paid for although they will be removed in the data pre-processing stage.
- 2) On average, only 70.87% of the submitted RJs could be used for further processing. Thus, around 90\$ US-Dollars of the raised 298.18\$ US-Dollars went to judges that did not actually *earn* the money.

Consequently, around $\frac{1}{3}$ of the money spent on CrowdFlower is lost.

However, after removing these obviously bad RJs, CS with CrowdFlower provides a useful RJ base for evaluation, which was also concluded by e.g. Nowak and Ruger (2010) or Blanco et al. (2011). As the authors conclude there, even with such a high number of RJs per image, CS RJs are still relatively inexpensive. Although Zhu and Carterette (2010) point out that some judges may participate reliably in one task but not in another, using various RJs per image and applying the averaging strategy can most likely eliminate this problem. This is supported by the high positive correlation (0.872) between CS RJs and trusted RJs.

Additionally, there are also those judges that did not contribute a valuable explanation of *why* they had chosen a certain RJ. However, these RJs were not a priori discarded. Moreover, the correlation analysis revealed most ranks given by judges to correspond to the ranks a trusted person would assign to an image. In general, judges make less use of the finer granularity of the ranking scale. An image is either completely valid or not. If either the thematic part or the spatial part did not match the topic, many assessors assigned an image the rank 0 (irrelevant), although they could have judged it to be e.g. marginally relevant. This result indicates, therefore, that the task design is not yet completely mature, and more research needs to be done to effectively incorporate all dimensions into the tasks.

5.3.2 Task Design and Data Pre-Processing

What the section above indicates is that a good task design is vital for retrieving valuable RJs. The quite long but also informative task description turned out to work rather well. Additionally, having at least around 10 valid RJs per image proves to be a viable strategy to calculate an average rank, reliably approximating the “real” rank of an image in relation to a topic. However, as Blanco et al. (2011) suggest, around three judges may already be enough for a reliable result. Generally speaking, for CS tasks (where the judges have no responsibility towards the researchers and will not suffer any consequences if they act untrustworthy), retrieving more than one RJ and calculating an average RJ is imperative.

Trustworthiness of judges and a suitable task design can also be derived from the submitted comments. For example, many judges clearly distinguish between thematic and spatial parts of the query. In these subparts, they even distinguish well visible themes from those only mentioned in the text or barely visible in the image. Additionally, when an image was located *near* but not *inside* an area and the query asked for images *inside the boundaries* of the location, some judges with either locational knowledge or those who examined the provided map thoroughly, distinguished images lying inside or outside the location in question. Therefore, the provided map proves to be a vital addition to the tasks. Some judges however would have liked the possibility to zoom into an image to better evaluate the relevance of an image to a topic. This may be an addition worth considering in further task designs. An unwanted effect was that assessors judged images’ relevance in comparison to other images of the same job. This is not wanted and adds a certain bias because the units of a job are evaluated in relation to each other. However, although not examined in detail, averaging RJs may ease the impact of such a bias.

Some assessors even got used to the fact that there exist trap images. A similar observation was made by Zhu and Carterette (2010). Therefore, it may happen that also those judges only looking to earn as much money as possible learn how to avoid such questions. This would be a problem especially if the same judges could assess the same job more than once, or if they evaluated all the jobs that are submitted by the same researcher and always look out for the fake image to avoid it. Again, averaging RJs diminishes the influence of such missed bad judges.

From a time constraint point of view, the CrowdFlower experiment lasted around 6 to 7 days. Experiments with no preliminary training of the judges reported in Alonso and Mizzaro (2009) were completed within 2 days. Due to the different settings of the experiments, the times can still be considered comparably short, which supports the fact that CS RJs are, besides their relatively high usability, very fast to obtain once the tasks are created.

6 Conclusions

6.1 Achievements

This thesis investigated several research gaps concerning algorithms for retrieving images from queries with spatial relevance. Moreover, a complete SPAISE prototype was implemented and evaluated. The evaluation explored and analysed how CS with CrowdFlower can be effectively used to supply viable RJs for SPAISE performance evaluations. The overarching research question was the following:

Overarching Research Question

How can methods from GIR and CBIR efficiently be combined for the purpose of retrieving spatially relevant images and also effectively favouring thematically highly relevant images while discarding images with minor relevance to the submitted query; and how can this performance be assessed?

This question was assessed both from an algorithmic and an evaluation point of view.

The algorithmic part revealed that, in terms of P@10, AP@10, and NDCG@10, both an explicit incorporation of the spatial dimension into the retrieval process and an additional CBIR PRF re-ranking are able to enhance the retrieval performance compared to a text-only retrieval system. However, together with the configurations, topics and image set used here, it is not clear if an additional CBIR re-ranking can improve retrieval results compared to a system only having term and spatial dimensions.

The experiment on evaluation measures revealed that only small adaptations in topic design (an explicit addition of the spatial dimension) compared to existing topics provide a useful task description for judges to evaluate the relevance of images. Furthermore, CS RJs for evaluating a SPAISE provide a viable replacement for traditional RJs using only some few, known judges, as long as several protection measures are built into the tasks. Although at least 30% of the retrieved RJs had to be removed before performance assessments, a correlation analysis between CS RJs and RJs of a trusted person revealed that an average rank calculated from the remaining set of around 70% RJs provides a good base for assessing the performance of a SPAISE.

6.2 Implications

6.2.1 Implications on Retrieval Methods

ISE should be able to understand spatial relationships explicitly if it is intended to conduct spatial queries. Online services for disambiguating location names and retrieving spatial query footprints work well, but only provide point locations (GN) or MBRs (YPM). In most cases however, an MBR proved to be more than enough, especially together with the point-based geometrical, spatial

relationships. It thus can also be concluded that geometric, mostly point-based approaches with restrictions dependent on the spatial query footprint size, are able to perform well for images.

Furthermore, a good SPAISE design may not necessarily need the incorporation of an image content dimension to effectively retrieve images. The semantic gap may rather be *avoided* instead of solved for image queries with spatial relevance. However, if the computational power allows it, such a re-ranking can still provide better retrieval results, dependent on the submitted query. Table 30 conclusively summarises the identified query types, where each system performed better compared to the other two systems. An assured fact is that CBIR should only be applied on an already filtered, thematically more relevant subset of images. This helps bridging and limiting the semantic gap and reduces the computational needs by avoiding the problem of ranking all images of a subset. This, on the other hand, requires precise annotations in form of titles, descriptions, and coordinates.

System	Preferred topic types in terms of precision@10
T	<ul style="list-style-type: none"> - Specific topics. - Well-known, often used place names with no specific directional preference.
TS	<ul style="list-style-type: none"> - General topics. - Any (ambiguous or specific) place name and direction. - Hard to distinguish image content/low-level features.
TSCR	<ul style="list-style-type: none"> - General topics. - Any (ambiguous or specific) place name and direction. - Characteristic or easily distinguishable image content/low-level features.

Table 30: Summary of query types and systems suited best to assess them.

Additionally, the high P@10, AP/MAP@10 and NDCG@10 values indicate that the coordinates assigned to the images of the collection are accurate, making the Geograph database a very useful choice for the purpose of evaluating a SPAISE.

6.2.2 Implications on Crowdsourcing Evaluations

Several suggestions can be summarised for a good task design and processing of RJs retrieved through CS on CrowdFlower. These findings involve:

- 1) Keep the task as simple as possible. Describe the task as clearly as possible.
- 2) Each task description needs to incorporate a brief summary of the task’s purpose and then clearly explain, in subsections, how judges are supposed to evaluate the topic in question.
- 3) If more than one dimension needs to be judged (e.g. a spatial), all these dimensions need to be clearly separated and flagged to be easily recognised as such.
- 4) Trap images are an effective measure to rule out untrustworthy judges.

- 5) Mandatory text answers provide interesting insights into the reasoning of the judges and can be used to assess the quality of the RJs.
- 6) Testing the task design with a handful of trusted people beforehand that have no knowledge about the task may reveal flaws the researchers have not thought about.

However, it has to be kept in mind that an average of around 30% of the RJs cannot be used for further analysis. Therefore, to retrieve more reliable RJs, the following measures should be taken:

- 7) Gather plenty of RJs for each image (at least 5 to 10).
- 8) Calculate an average RJ from these image RJs.

Different average measures (at least AM, median and mode) barely alter the average RJ.

6.3 Future Work

First of all, the algorithms used may be adapted. In this thesis, it could not be evaluated if a different weighting of the two (three) dimensions could improve retrieval performance. Connected to the latter point is the question of how many images per dimension should be retrieved and what the appropriate thresholds for retrieval scores (e.g. no images retrieved with a similarity score below 0.5) are. Additionally, different combinational strategies (different Comb strategies, Borda fusion instead of Comb) may prove to provide more relevant retrieval results.

An evaluation of the accuracy of the proposed MBR estimation from a linear regression formula using GN's population data (and possible adaptations through the use of other point fitting curves) may reveal this method to be a simple and useful alternative for situations where no MBR is available but needed. Furthermore, including more precise spatial query footprints (e.g. convex hulls instead of MBRs) may improve retrieval accuracy, especially in the case of the inside relationship. Also, effective ways to extract the spatial part from an input query string should be tested on, so that the user does not have to explicitly choose a spatial relationship (free text query). Different distance factors for the near relationship could be empirically evaluated to find a more general description of what is still considered to be near. Further directional relationships (NE, NW, SE, and SW) may increase retrieval accuracy as suggested by judges' comments. This system included coordinates of where the object in the image is located. However, normally only the location of where the photographer stood can be provided. Therefore, the system's performance should be tested with such more realistic coordinates.

The latter suggestion may also have effects on the CBIR PRF re-ranking, making it more useful in the situation of noisy coordinates. The same applies to terms. A more general set of text annotations e.g. from Flickr may be experimented on to estimate the applicability of the re-ranking algorithm in a presumably more realistic setting. Altering parameters and methods of the PRF re-ranking algorithms (number of candidate and example images, cluster linkage algorithms) may, furthermore, lead to

better retrieval performance. Using example images of several clusters of candidate images may add more variability. Also, other global low-level image features like ACC or local features like SURF could be tested to see if they may increase retrieval effectiveness of the PRF algorithm.

On the evaluation side, more than 25 topics need to be formulated to assess retrieval performance more accurately. Moreover, all images of the collection should be assigned an RJ (or at least the top-20 of each system). In this work, neither the real MAP over all retrieved images nor any recall measures could be calculated. It is possible that at least an extension to the top-20 images could already show different results between the systems. Judging the whole image collection and incorporating more than only the 30% of images used in this thesis would also provide a base to reproducibly and comparably evaluate SPAISEs, because to date, no such collection suited for spatial image search evaluations containing enough images exists. A thorough evaluation of the RJs obtained here through crowdsourcing also needs to be contrasted with expert evaluations to completely reveal which of the set of judges performs better overall. A further study could also evaluate the already gathered RJs to find out how many RJs are actually needed to retrieve a reliable average rank for an image. Additionally, the defined topics need to be evaluated. Too many images retrieved were highly relevant. Either the queries submitted were too simple even for the T system or there are too many relevant images for these topics. It may also have to do with the number of evaluated images as mentioned before. Queries, therefore, may be formulated that focus more on revealing the *weaknesses* of the systems, not their *strengths*.

Bibliography

- Alonso, O., Mizzaro, S., 2009. Can we get rid of TREC Assessors? Using Mechanical Turk for Relevance Assessment., In: Geva, S., Kamps, J., Peters, C., Sakai, T., Trotman, A., Voorhees, E.M. (Eds.). *Proceedings of the SIGIR 2009 Workshop Future of IR Evaluation*, IR Publications, Amsterdam.
- Aly, M., Welinder, P., Munich, M., Perona, P., 2009. Automatic discovery of image families: Global vs. local features. *16th IEEE International Conference on Image Processing (ICIP)*, pp. 777 – 780. doi: 10.1109/ICIP.2009.5414235.
- André, P., Cutrell, E., Tan, D.S., Smith, G., 2009. Designing Novel Image Search Interfaces by Understanding Unique Characteristics and Usage, In: Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M.Y., Weikum, G., Gross, T., Gulliksen, J., Kotzé, P., Oestreicher, L., Palanque, P., Prates, R.O., Winckler, M. (Eds.) *Human-Computer Interaction – INTERACT 2009*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 340–353.
- Arampatzis, A., Kamps, J., Robertson, S., 2009. Where to stop reading a ranked list? In: *Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval - SIGIR '09*, ACM Press, p. 524.
- Arampatzis, A., Zagoris, K., Chatzichristofis, S.A., 2013. Dynamic two-stage image retrieval from large multimedia databases. *Information Processing & Management*, 49(1), 274–285. doi:10.1016/j.ipm.2012.03.005.
- Armitage, L.H., Enser, P.G., 1997. Analysis of user need in image archives. *Journal of Information Science*, 23(4), 287–299. doi:10.1177/016555159702300403.
- Backhaus, K., Backhaus-Erichson-Plinke-Weiber, Erichson, B., Plinke, W., Weiber, R., 2006. *Multivariate Analysemethoden: Eine anwendungsorientierte Einführung*, 11th edn., Springer, Berlin [u.a.], VII, 830 S.
- Baeza-Yates, R., Ribeiro, B., 1999. *Modern information retrieval*. Addison-Wesley, Harlow [u.a.], XX, 513 S.
- Bailey, P., Craswell, N., Soboroff, I., Thomas, P., Vries, A.P. de, Yilmaz, E., 2008. Relevance Assessment: Are Judges Exchangeable and Does it Matter? In: *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '08*, ACM Press, pp. 667–674.
- Barthel, K.U., 2008. Improved Image Retrieval Using Automatic Image Sorting and Semi-automatic Generation of Image Semantics. *2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services*, IEEE, pp. 227–230.

- Bay, H., Tuytelaars, T., Gool, L., 2006. SURF: Speeded Up Robust Features, In: Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M.Y., Weikum, G., Leonardis, A., Bischof, H., Pinz, A. (Eds.) *Computer Vision – ECCV 2006*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 404–417.
- Beitzel, S.M., Jensen, E.C., Chowdhury, A., Grossman, D., Frieder, O., 2003. Using manually-built web directories for automatic evaluation of known-item retrieval. In: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval - SIGIR '03*, ACM Press, p. 373.
- Bentley, J.L., 1975. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9), pp. 509–517. doi:10.1145/361002.361007.
- Berg, M. de, Cheong, O., van Kreveld, M., Overmars, M., 2008. *Computational Geometry: Algorithms and Applications*, Springer Berlin Heidelberg, Berlin, Heidelberg.
- Blair, D.C., 1979. Information Retrieval, 2nd ed. C.J. Van Rijsbergen. London: Butterworths; 1979: 208 pp. *Journal of the American Society for Information Science*, 30(6), 374–375. doi:10.1002/asi.4630300621.
- Blanco, R., Halpin, H., Herzig, D.M., Mika, P., Pound, J., Thompson, H.S., 2011. Repeatable and reliable search system evaluation using crowdsourcing. In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information - SIGIR '11*, ACM Press, pp. 923–932.
- Böhm, C., Klump, G., Kriegel, H.-P., 1999. XZ-Ordering: A Space-Filling Curve for Objects with Spatial Extension, In: Güting, R.H., Papadias, D., Lochovsky, F. (Eds.) *Advances in Spatial Databases*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 75–90.
- Borda, J.-C. de, 1781. Mémoire sur les élections au scrutin Histoire de l'Académie Royale des Sciences.
- Brisaboa, N.R., Luaces, M.R., Places, Á.S., Seco, D., 2010. Exploiting geographic references of documents in a geographical information retrieval system using an ontology-based index. *GeoInformatica*, 14(3), pp. 307–331. doi:10.1007/s10707-010-0106-3.
- Butler, H., Schmidt, C., Springmeyer, D., Livni, J. EPSG Projection 4326 - WGS 84, tinyurl.com/bo8czxo, [accessed 17 July 2013].
- Cai, G., 2011. Relevance ranking in Geographical Information Retrieval. *SIGSPATIAL Special*, 3(2), 33–36. doi:10.1145/2047296.2047304.
- Carbonell, J.G., Yang, Y., Frederking, R.E., Brown, R.D., Geng, Y., Lee, D., 1997. Translingual information retrieval: A comparative evaluation, In: Pollack, M.E. (Ed.). *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

- Carterette, B., Voorhees, E.M., 2011. Overview of Information Retrieval Evaluation, In: Lupu, M., Mayer, K., Tait, J., Trippe, A.J. (Eds.). *Current Challenges in Patent Information Retrieval*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 69–85.
- Chatzichristofis, S.A., Arampatzis, A., 2010. Late fusion of compact composite descriptors for retrieval from heterogeneous image databases. In: *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '10*, ACM Press, p. 825.
- Chatzichristofis, S.A., Boutalis, Y.S., 2008a. CEDD: Color and Edge Directivity Descriptor: A Compact Descriptor for Image Indexing and Retrieval, In: Gasteratos, A., Vincze, M., Tsotsos, J.K. (Eds.) *Computer Vision Systems*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 312–322.
- Chatzichristofis, S.A., Boutalis, Y.S., 2008b. FCTH: Fuzzy Color and Texture Histogram - A Low Level Feature for Accurate Image Retrieval. *2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services*, IEEE, pp. 191–196.
- Chi, Z., Yan, H., Pham, T., 1996. *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition*. World Scientific, River Edge, NJ, Singapore.
- Clarke, C.L.A., Craswell, N., Soboroff, I., 2009. Overview of the TREC 2009 Web track, In: Voorhees, E.M., Buckland, L.P. (Eds.). *Proceedings of the 18th text retrieval conference (TREC 2009)*, Gaithersburg.
- Cleverdon, C.W., 1962. Report on the testing and analysis of an investigation into the comparative efficiency of indexing systems. Technical Report, Cranfield, USA.
- Cleverdon, C.W., 1991. The significance of the Cranfield tests on index languages. In: *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '91*, ACM Press, pp. 3–12.
- Clough, P.D., Sanderson, M., Joho, H., 04.01.2004. Spatially-Aware Information Retrieval on the Internet: Extraction of semantic annotations from textual web pages, http://www.geospirit.org/publications/SPIRIT_WP6_D15_geomarkup_revised_FINAL.pdf, [accessed 23 Jul 2013].
- Dalal, N., Triggs, B., 2005. Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, pp. 886–893.
- Datta, R., Joshi, D., Li, J., Wang, J.Z., 2008. Image retrieval: Ideas, Influences, and Trends of the New Age. *ACM Computing Surveys*, 40(2), 1–60. doi:10.1145/1348246.1348248.
- Deselaers, T., Keysers, D., Ney, H., 2008. Features for image retrieval: an experimental comparison. *Information Retrieval*, 11(2), 77–107. doi:10.1007/s10791-007-9039-3.
- Eakings, J., Graham, M., 1999. *Content-based Image Retrieval: A Report to the JISC Technology Applications Programme*, University of Northumbria, Newcastle.

- Egenhofer, M.J., Mark, D.M., 1995. Naive Geography, In: Goos, G., Hartmanis, J., Leeuwen, J., Frank, A.U., Kuhn, W. (Eds.). *Spatial Information Theory A Theoretical Basis for GIS*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–15.
- Elmasri, R., Navathe, S., 2011. Fundamentals of database systems, 6th edn., Addison-Wesley, Boston, xxvii, 1172.
- Enser, P., 2000. Visual image retrieval: seeking the alliance of concept-based and content-based paradigms. *Journal of Information Science*, 26(4), 199–210. doi:10.1177/016555150002600401.
- Enser, P.G., Sandom, C.J., Hare, J.S., Lewis, P.H., 2007. Facing the reality of semantic image retrieval. *Journal of Documentation* 63, 465–481. doi:10.1108/00220410710758977.
- European Organization for the Safety of Air Navigation, Institute of Geodesy and Navigation, 12.02.1998. WGS 84 Implementation Manual, <http://tinyurl.com/kg7g2>.
- Faloutsos, C., Christodoulakis, S., 1984. Signature files: an access method for documents and its analytical performance evaluation. *ACM Transactions on Information Systems*, 2(4), 267–288. doi:10.1145/2275.357411.
- Finkel, R.A., Bentley, J.L., 1974. Quad trees a data structure for retrieval on composite keys. *Acta Informatica* 4, 1–9. doi:10.1007/BF00288933.
- Foncubierta-Rodríguez, A., Müller, H., 2012. Ground truth generation in medical imaging. In: *Proceedings of the ACM multimedia 2012 workshop on Crowdsourcing for multimedia - CrowdMM '12*, ACM Press, p. 9.
- Fowler, M., 10.07.2013. GUI Architectures: Model View Controller, <http://tinyurl.com/namwdp3>, [accessed 17 July 2013].
- Fox, E., Shaw, J., 1993. Combination of Multiple Searches., In: Harman, D. (Ed.). *TREC-1: Proceedings of the First Text Retrieval Conference*, Gaithersburg, MD, USA, pp. 243–252.
- Frontiera, P., Larson, R., Radke, J., 2008. A comparison of geometric approaches to assessing spatial similarity for GIR. *International Journal of Geographical Information Science*, 22(3), 337–360. doi:10.1080/13658810701626293.
- Gaede, V., Günther, O., 1998. Multidimensional access methods. *ACM Computing Surveys*, 30(2), 170–231. doi:10.1145/280277.280279.
- Gaio, M., Sallaberry, C., Etcheverry, P., Marquesuzaa, C., Lesbegueries, J., 2008. A global process to access documents' contents from a geographical point of view. *Journal of Visual Languages & Computing*, 19(1), 3–23. doi:10.1016/j.jvlc.2007.08.010.
- Gamma, E., 2011. Design patterns: Elements of reusable object-oriented software, 39th edn., Addison-Wesley, Boston [u.a.], XV, 395 S.

- GeoHack - Centre of United Kingdom, <http://tinyurl.com/jwcfhw9>, [accessed 17 July 2013].
- Goodchild, M.F., 1999. Future Directions in Geographic Information Science. *Annals of GIS*, 5(1), 1–8. doi:10.1080/10824009909480507.
- Goodrich, M.T., Tamassia, R., Mount, D.M., 2004. Data structures and algorithms in C++, Wiley, Hoboken, NJ, xv, 683.
- Gruntz, D., 2012. Object-Oriented Software Design (using Java), Zurich.
- Gustafson, D., Kessel, W., 1978. Fuzzy clustering with a fuzzy covariance matrix. *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, IEEE, pp. 761–766.
- Guttman, A., 1984. R-trees: A dynamic index structure for spatial searching. In: *Proceedings of the 1984 ACM SIGMOD international conference on Management of data - SIGMOD '84*, ACM Press, pp. 47–57.
- Haar, A., 1910. Zur Theorie der orthogonalen Funktionensysteme. *Mathematische Annalen*, 69(3), 331–371. doi:10.1007/BF01456326.
- Harman, D., 1995. Overview of the Second Text Retrieval Conference (TREC-2). *Information Processing & Management*, 31(3), 271–289. doi:10.1016/0306-4573(94)00047-7.
- Harman, D., 2002. Overview of the TREC 2002 novelty track, In: Voorhees, E.M. (Ed.). *Proceedings of the 11th Text Retrieval Conference (TREC 2002)*, Gaithersburg, pp. 46–55.
- Harter, S.P., 1996. Variations in relevance assessments and the measurement of retrieval effectiveness. *Journal of the American Society for Information Science*, 47(1), 37–49. doi:10.1002/(SICI)1097-4571(199601)47:1<37::AID-ASI4>3.3.CO;2-I.
- He, D., Wu, D., 2008. Toward a Robust data fusion for document retrieval. *2008 International Conference on Natural Language Processing and Knowledge Engineering*, IEEE, pp. 1–8.
- Heiddorn, P., 1999. Image Retrieval as Linguistic and Nonlinguistic Visual Model Matching. *Library Trends* 48, 303–325.
- Hellerstein, J., Kornacker, M., Shah, M., Thomas, M., Papadimitriou, C., 04.06.2001. The GiST Indexing Project: GiST: Generalized Search Tree, <http://gist.cs.berkeley.edu/>, [accessed 17 July 2013].
- Hill, L.L., 2006. Georeferencing: The geographic associations of information, MIT Press, Cambridge, Mass, xiii, 260.
- Hjørland, B., 2009. The foundation of the concept of relevance. *Journal of the American Society for Information Science and Technology*, 61(2). doi:10.1002/asi.21261.
- Hsu, W.H., Kennedy, L.S., Chang, S.-F., 2007. Reranking Methods for Visual Search. *IEEE Multimedia*, 14(3), 14–22. doi:10.1109/MMUL.2007.61.

- Huang, J., Kumar, S., Mitra, M., Wei-Jing Zhu, Zabih, R., 1997. Image indexing using color correlograms. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Comput. Soc, pp. 762–768.
- Hull, D., 1993. Using statistical testing in the evaluation of retrieval experiments. In: *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '93*, ACM Press, pp. 329–338.
- Iqbal, K., Odetayo, M.O., James, A., 2012. Content-based image retrieval approach for biometric security using colour, texture and shape features controlled by fuzzy heuristics. *Journal of Computer and System Sciences*, 78(4), 1258–1277. doi:10.1016/j.jcss.2011.10.013.
- Järvelin, K., Kekäläinen, J., 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4), 422–446. doi:10.1145/582415.582418.
- JMapView - OpenStreetMap Wiki, <http://wiki.openstreetmap.org/wiki/JMapView>, [accessed 17 July 2013].
- Jones, C.B., Purves, R.S., 2008. Geographical information retrieval. *International Journal of Geographical Information Science*, 22(3), 219–228. doi:10.1080/13658810701626343.
- Kamahara, J., Nagamatsu, T., Tanaka, N., 2012. Conjunctive ranking function using geographic distance and image distance for geotagged image retrieval. In: *Proceedings of the ACM multimedia 2012 workshop on Geotagging and its applications in multimedia - GeoMM '12*, ACM Press, p. 9.
- Kraaij, W., Vries, A.P. de, Clarke, C.L.A., Fuhr, N., Kando, N., Farah, M., Vanderpooten, D., 2007. An outranking approach for rank aggregation in information retrieval. In: *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '07*, ACM Press, p. 591.
- Kraft, D.H., Croft, W.B., Harper, D.J., Zobel, J., Aslam, J.A., Montague, M., 2001. Models for metasearch. In: *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '01*, ACM Press, pp. 276–284.
- Kuropka, D., 2004. Modelle zur Repräsentation natürlichsprachlicher Dokumente: Ontologie-basiertes Information-Filtering-und-Retrieval mit relationalen Datenbanken, 1st edn., Logos Verlag Berlin, Berlin, xix, 242.
- Kuropka, D., 20.07.2012. Information Retrieval Models, <http://tinyurl.com/qyzymee>, [accessed 13 July 2013].
- Lars Behnke. Implementation of an agglomerative hierarchical clustering algorithm in Java.: lbehnke/hierarchical-clustering-java · GitHub, <http://tinyurl.com/ljaxv xv>, [accessed 25 July 2013].
- Larson, R.R., Frontiera, P., 2004. Spatial Ranking Methods for Geographic Information Retrieval (GIR) in Digital Libraries, In: Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C.,

- Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M.Y., Weikum, G., Heery, R., Lyon, L. (Eds.) *Research and Advanced Technology for Digital Libraries*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 45–56.
- Lease, M., Yilmaz, E., 2012. Crowdsourcing for information retrieval. *ACM SIGIR Forum*, 45(2), 66. doi:10.1145/2093346.2093356.
- Liew, A.W.-C., Law, N.-F., 2008. Content-Based Image Retrieval, In: Khosrow-Pour, M. (Ed.) *Encyclopedia of Information Science and Technology*, Second Edition, IGI Global, pp. 744–749.
- Liskov, B.H., Wing, J.M., 1994. A behavioral notion of subtyping. *ACM Transactions on Programming Languages and Systems*, 16(6), 1811–1841. doi:10.1145/197320.197383.
- Liu, B., 2007. Information Retrieval and Web Search, In: Web Data Mining, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 183–236.
- Lowe, D., 1999. Object recognition from local scale-invariant features. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*, IEEE, pp. 1150–1157 vol.2.
- Lucene 4.0.0 API, TFIDFSimilarity, <http://tinyurl.com/anzjcqr>, [accessed 17 July 2013].
- Lux, M. News on LIRE performance, <http://www.semanticmetadata.net/2012/12/12/news-on-lire-performance/>, [accessed 17 July 2013].
- Lux, M., Chatzichristofis, S.A., 2008. Lire: lucene image retrieval: an extensible java CBIR library. In : *Proceedings of the 16th ACM international conference on Multimedia – MM '08*, ACM press, pp. 1085 – 1088.
- Maillot, N., Chevallet, J.-P., Lim, J.H., 2007. Inter-media Pseudo-relevance Feedback Application to ImageCLEF 2006 Photo Retrieval, In: Peters, C., Clough, P., Gey, F.C., Karlgren, J., Magnini, B., Oard, D.W., Rijke, M., Stempfhuber, M. (Eds.). *Evaluation of Multilingual and Multi-modal Information Retrieval*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 735–738.
- Manber, U., Myers, G., 1993. Suffix Arrays: A New Method for On-Line String Searches. *SIAM Journal on Computing*, 22, 935–948. doi:10.1137/0222058.
- Manjunath, B., Ohm, J.-R., Vasudevan, V., Yamada, A., 2001. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6), 703–715. doi:10.1109/76.927424.
- Manning, C.D., Raghavan, P., Schütze, H., 2008. Introduction to information retrieval, Cambridge University Press, New York, xxi, 482.
- Margaret Rouse. Thumbnail: Definition: Multimedia and graphics glossary, <http://tinyurl.com/p9z55nr>, [accessed 20 July 2013].
- Maron, M.E., Kuhns, J.L., 1960. On Relevance, Probabilistic Indexing and Information Retrieval. *Journal of the ACM*, 7(3), 216–244. doi:10.1145/321033.321035.

- Martin, R.C., 2002. Agile software development: Principles, patterns, and practices, Prentice Hall/Pearson Education, Upper Saddle River, NJ, XXII, 529 p.
- Martins, B., Calado, P., 2010. Learning to rank for geographic information retrieval, In: Purves, R., Clough, P., Jones, C. (Eds.). *Proceedings of the 6th Workshop on Geographic Information Retrieval - GIR '10*, ACM Press, p. 1.
- Martins, B., Silva, M.J., Andrade, L., 2005. Indexing and ranking in Geo-IR systems. In: *Proceedings of the 2005 workshop on Geographic information retrieval - GIR '05*, ACM Press, p. 31.
- Meyer, B., 1988. Object-oriented software construction, 2nd edn., Prentice Hall, Upper Saddle River, NJ, XXVII, 1254 S.
- Mikolajczyk, K., Schmid, C., 2005. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10), 1615–1630. doi:10.1109/TPAMI.2005.188.
- Müller, H., 2010. Creating Realistic Topics for Image Retrieval Evaluation, In: Müller, H., Clough, P., Deselaers, T., Caputo, B. (Eds.) *ImageCLEF*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 45–61.
- Müller, H., Clough, P., Deselaers, T., Caputo, B., 2010. *ImageCLEF*, Springer Berlin Heidelberg, Berlin, Heidelberg.
- Nievergelt, J., Hinterberger, H., Sevcik, K.C., 1984. The Grid File: An Adaptable, Symmetric Multikey File Structure. *ACM Transactions on Database Systems*, 9(1), 38–71. doi:10.1145/348.318586.
- Nowak, S., Rüger, S., 2010. How reliable are annotations via crowdsourcing. In: *Proceedings of the international conference on Multimedia information retrieval - MIR '10*, ACM Press, pp. 557–566.
- OpenGeo, 19.07.2013a. Introduction to PostGIS: Section 8: Geometries, <http://tinyurl.com/d43wd7v>, [accessed 20 July 2013].
- OpenGeo, 19.07.2013b. Introduction to PostGIS: Section 15: Projecting Data, <http://tinyurl.com/ckmjdnq>, [accessed 20 July 2013].
- Ounis, I., Ruthven, I., Macdonald, C., Christoforaki, M., He, J., Dimopoulos, C., Markowetz, A., Suel, T., 2011. Text vs. space: Efficient Geo-Search Query Processing. In: *Proceedings of the 20th ACM international conference on Information and knowledge management - CIKM '11*, ACM Press, p. 423.
- Palacio, D., Cabanac, G., Sallaberry, C., Hubert, G., 2011. On the evaluation of Geographic Information Retrieval systems. *International Journal on Digital Libraries*, 11(2), 91–109.
- Palacio, D., Sallaberry, C., Gaio, M., 2012. Normalizing Spatial Information to Improve Geographical Information Indexing and Retrieval in Digital Libraries, In: Yeh, A.G., Shi, W., Leung, Y., Zhou, C. (Eds.). *Advances in Spatial Data Handling and GIS*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 65–78.

- Panofsky, E., 1982, c1955. *Meaning in the visual arts*, University of Chicago Press, Chicago, xviii, 364.
- Paul, J., 25.02.2010. History of the Prime Meridian - Past and Present, <http://gpsinformation.net/main/greenwich.htm>, [accessed 17 July 2013].
- Popescu, A., Moëllic, P.-A., Kanellos, I., Landais, R., 2009. Lightweight web image reranking. In: *Proceedings of the seventeen ACM international conference on Multimedia - MM '09*, ACM Press, p. 657.
- Porter, M., 1980. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3), 130–137. doi:10.1108/eb046814.
- PostGIS: Chapter 4. Using PostGIS. 1.3.7 SVN Manual, <http://tinyurl.com/bw49v4c>, [accessed 17 July 2013].
- Purves, R.P., Edwards, A., Fan, X., Hall, M., Tomko, M., 2010. Automatically generating keywords for georeferenced images. In: *Proceedings of the international conference on Multimedia information retrieval*, ACM Press.
- Purves, R.S., Clough, P., Jones, C.B., Arampatzis, A., Bucher, B., Finch, D., Fu, G., Joho, H., Syed, A.K., Vaid, S., Yang, B., 2007. The design and implementation of SPIRIT: a spatially aware search engine for information retrieval on the Internet. *International Journal of Geographical Information Science*, 21(7), 717–745. doi:10.1080/13658810601169840.
- Quemada, J., León, G., Maarek, Y., Nejdl, W., van Leuken, R.H., Garcia, L., Olivares, X., van Zwol, R., 2009. Visual diversification of image search results. In: *Proceedings of the 18th international conference on World wide web - WWW '09*, ACM Press, p. 341.
- Raper, J., 2007. Geographic relevance. *Journal of Documentation*, 63(6), 836–852. doi:10.1108/00220410710836385.
- Robertson, S.E., 1995. Okapi at TREC-3, British Library Research and Development Department, London, 27 pp.
- Robertson, S., Hull, D., 2000. The TREC-9 filtering track final report, In: Voorhees, E.M., Harman, D. (Eds.). *Proceedings of the 9th text retrieval conference (TREC-9)*, Gaithersburg.
- Royal Museums Greenwich, 15.08.2005. The Longitude of Greenwich, <http://tinyurl.com/kubojwo>, [accessed 17 July 2013].
- Rui, Y., Huang, T., 1999. Image retrieval: Current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 10. 39 - 62.
- Sabbata, S. de, 2013. Assessing Geographic Relevance for Mobile Information Services. Dissertation, University of Zurich, Zurich.

- Samet, H., 2006. Foundations of multidimensional and metric data structures, Elsevier/Morgan Kaufmann, Amsterdam [u.a.], XXVII, 993 S.
- Sanderson, M., 09.07.2009. Project Tripod: Automating caption creation, <http://tripod.shef.ac.uk/>, [accessed 16 July 2013].
- Sanderson, M., 2010. Test Collection Based Evaluation of Information Retrieval Systems. *Foundations and Trends® in Information Retrieval*, 4(4), 247–375. doi:10.1561/1500000009.
- Saracevic, T., 1995. Evaluation of evaluation in information retrieval, In: Fox, E., Ingwersen, P., Fidel, R. (Eds.). *Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval - SIGIR '95*, ACM Press, pp. 138–146.
- Shatford, S., 1986. Analyzing the Subject of a Picture: A Theoretical Approach. *Cataloging & Classification Quarterly*, 6(3), 39–62. doi:10.1300/J104v06n03_04.
- Shekhar, S., Chawla, S., 2003. Spatial databases: A tour, Prentice Hall, Upper Saddle River, N.J., xxiii, 262.
- Sikora, T., 2001. The MPEG-7 visual standard for content description-an overview. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6), 696–702. doi:10.1109/76.927422.
- Skarbek, W. (Ed.), 2001. Computer Analysis of Images and Patterns, Springer Berlin Heidelberg, Berlin, Heidelberg.
- Smeulders, A., Worring, M., Santini, S., Gupta, A., Jain, R., 2000. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12), 1349–1380. doi:10.1109/34.895972.
- Spatial Reference, 2013. WGS 84: EPSG Projection, <http://spatialreference.org/ref/epsg/4326/>, [accessed 20 July 2013].
- Stéphane. Android - Adding distance to a GPS coordinate - Stack Overflow, <http://tinyurl.com/kms5n4a>, [accessed 17 July 2013].
- Swain, M.J., Ballard, D.H., 1991. Color indexing. *International Journal of Computer Vision*, 7(1), 11–32. doi:10.1007/BF00130487.
- Tamura, H., Mori, S., Yamawaki, T., 1978. Textural Features Corresponding to Visual Perception. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(6), 460–473. doi:10.1109/TSMC.1978.4309999.
- Tobin, R., Grover, C., Byrne, K., Reid, J., Walsh, J., 2010. Evaluation of georeferencing. In: *Proceedings of the 6th Workshop on Geographic Information Retrieval - GIR '10*, ACM Press, p. 1.
- Tobler, W.R., 1970. A Computer Movie Simulating Urban Growth in the Detroit Region. *Economic Geography*, 46(2), 234. doi:10.2307/143141.
- Toutenburg, H., Heumann, C., 2008. Deskriptive Statistik, Springer Berlin Heidelberg, Berlin, Heidelberg.

- Urbano, J., Morato, J., Marrero, M., Martin, D., 2010. Crowdsourcing preference judgments for evaluation of music similarity tasks, In: *Proceedings of the ACM SIGIR 2010 Workshop on Crowdsourcing for Search Evaluation (CSE 2010)*, Geneva, Switzerland, pp. 9–16.
- Vaid, S., Jones, C.B., Joho, H., Sanderson, M., 2005. Spatio-textual Indexing for Geographical Search on the Web, In: Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M.Y., Weikum, G., Bauzer Medeiros, C., Egenhofer, M.J., Bertino, E. (Eds.) *Advances in Spatial and Temporal Databases*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 218–235.
- van Kreveld, M., Reinbacher, I., Arampatzis, A., van Zwol, R., 2005. Multi-Dimensional Scattered Ranking Methods for Geographic Information Retrieval*. *GeoInformatica*, 9(1), 61–84. doi:10.1007/s10707-004-5622-6.
- van Rijsbergen, C.J., 1979. *Information retrieval*, 2nd edn., Butterworths, London, Boston, ix, 208.
- Wikipedia, 11.07.2013. Hierarchische Clusteranalyse, <http://tinyurl.com/l52f3wq>, [accessed 16 July 2013].
- Wikipedia, 16.07.2013. MediaWiki - Wikipedia, the free encyclopedia, <http://tinyurl.com/mgem5pw>, [accessed 17 July 2013].
- William Hooper, 12.07.2013. A Short History of the GUI and the Microsoft vs Apple Debate, <http://tinyurl.com/lqmkb5>, [accessed 16 July 2013].
- Yuen-C, T., Shin, Q.H., 2009. How Google search work, tinyurl.com/howgooglesearchworks, [accessed 13 July 2013].
- Zhu, D., Carterette, B., 2010. An analysis of assessor behavior in crowdsourced preference judgments, In: *Proceedings of the ACM SIGIR 2010 Workshop on Crowdsourcing for Search Evaluation (CSE 2010)*, Geneva, Switzerland, pp. 21–26, URL: <http://tinyurl.com/jvny4tt>.
- Zobel, J., Moffat, A., 2006. Inverted files for text search engines. *ACM Computing Surveys*, 38(2), 6–es. doi:10.1145/1132956.1132959.

Appendix A

Code Snippets for Creating Indexes and Inserting Images

Insertion of an Image's Title and Description into a Lucene Term Index. A 1 shows the classes implemented in `TermIndexCreator` used to create and insert terms into a document.

Class	Description
IndexWriter	<ul style="list-style-type: none">- Main class for indexing textual documents.- Creates and maintains an index.- <code>IndexWriterConfig</code> specifies configurations of <code>IndexWriter</code> for indexation.
IndexWriterConfig	<ul style="list-style-type: none">- Holds all the configurations of <code>IndexWriter</code> (e.g. used <code>Analyzer</code>).
Analyzer	<ul style="list-style-type: none">- Breaks up texts into tokens.- Performs operations on these tokens (e.g. down casing, synonym insertion, stemming, etc.).
EnglishAnalyzer	<ul style="list-style-type: none">- Tokenisation, stop word removal, down-casing and stemming (using Porter 1980).- No linguistic pre-processing or normalisation (see 2.2.1.1 Tokens and Terms).

A 1: Classes used for indexing terms with Lucene.

Insertion of a document follows the procedure shown in A 1.

```
Document doc = new Document();
doc.add(new StringField("imageID", imageID, Field.Store.YES));
doc.add(new TextField("title", title, Field.Store.YES));
doc.add(new TextField("description", description, Field.Store.YES));
indexWriter.addDocument(doc);
```

A 2: Code used to insert an image's title and description into a Lucene index.

A `Document` represents a container in the term index, where an image's identifier, title and description are stored. A `StringField` is a field that should not be tokenised, whereas a `TextField` is a field that should be. Therefore, only title and descriptions are tokenised (and analysed using the `EnglishAnalyzer`). In a last step, an instance of type `IndexWriter` adds a document to the index.

Creation of a Spatial Index. A 3 shows the SQL code used to create a table in a PostgreSQL database able of holding WGS 84 point coordinates.

```
1) CREATE TABLE locations ( image_id text );
2) SELECT ADDGEOMETRYCOLUMN('locations', 'geometry', 4326, 'GEOMETRY', 2);
3) CREATE INDEX locations_spatial_index ON locations USING GIST(geometry);
```

A 3: Code used to create a table able to store point locations and to conduct spatial queries and fast retrieval.

In 1), a table called `locations` is created which has one column `image_id` of type `text`. As the name suggests, this column holds an image identifier (a unique number representing the image). In 2), a spatial geometry column (`'geometry'`) is added to the beforehand created table. The third parameter

of this function, 4326, specifies the WGS 84 spatial reference system (Spatial Reference 2013). To be able to use this reference system, the database needs to hold the `spatial_ref_sys` table. This table defines all the spatial reference systems known to PostGIS (OpenGeo 19.07.2013b). 'GEOMETRY', the fourth parameter of this function, represents the used geometry type. In the case of point locations, a simple 'POINT' may be used, but 'GEOMETRY' is also able to store any other kind of geometry like 'LINE', 'POLYGON' etc. (OpenGeo 19.07.2013a). If any implementation of the coordinates of an image changed, e.g. to store the line from where the image was taken to the point where the actual object in the image was located, this implementation would not need to be adapted. The last parameter 2 specifies the number of dimensions. In the planar case, this means 2 dimensions with X and Y coordinates (OpenGeo 19.07.2013a). To enable fast spatial retrieval, 3) adds the actual GiST-index to the geometry column with name `locations_spatial_index`. Only three lines of code are needed to build up a spatially indexed database table in PostgreSQL/PostGIS.

Insertion of an Image's Coordinates into a Spatial Index. Insertion of image locations follows common SQL conventions. A 4 shows that first an image's identifier ('19') and secondly, the geometry is inserted using the PostGIS function `ST_GEOMFROMTEXT` which converts text to WGS 84 coordinates. This function takes the type of geometry as an input (here 'POINT') with its corresponding latitude and longitude coordinates and again as a second parameter the spatial reference system (4326). This insertion statement is repeated for *all* the images to be indexed.

```
INSERT INTO locations VALUES(
    '19',
    ST_GEOMFROMTEXT('POINT(longitude, latitude)', 4326)
);
```

A 4: Code used to insert a new point location.

In Java, an instantiation of `AbstractDBConnector` is passed to `SpatialIndexCreator`, which in this case is of type `PGDBConnector` (PG means PostgreSQL). The class is able of connecting to and querying a PostgreSQL database holding the specified spatial index. `PGDBConnector` uses JDBC to connect to a database. To be able to conduct SQL queries, a `java.sql.Statement` has to be created from a `java.sql.Connection` and then a well-formed query string (like the one in A 4) needs to be passed to this `Statement` via the `executeQuery()` method. This simple procedure can be seen in A 5 for the insertion of a new point location.

```
Statement statement = connection.createStatement();
String query = "INSERT INTO locations VALUES('19',
    ST_GEOMFROMTEXT('POINT(longitude, latitude)',4326))";
statement.executeUpdate(query);
```

A 5: Code used to insert a new point location in the `SpatialIndexCreator`.

Insertion of an Image into a LiRE Image Content Index. An instance of `IndexWriter` creates and inserts instances of type `Document`. A `Document` consists of the image's identifier and the extracted features. The `DocumentBuilder` is provided through the `DocumentBuilderFactory`, a factory method (Gamma 2011) which implements all the global features possible to extract with LiRE. Inserting into the index, therefore, is very similar to inserting into a Lucene index as can be seen in A 6. In contrast to indexing of terms into a Lucene index, image feature extraction and indexing is encapsulated within the `DocumentBuilder`. Only the actual image (as a `BufferedImage`) and the image identifier need to be passed. The created `Document` is then simply added to the index by using an instance of Lucenes `IndexWriter`.

```
Document doc = documentBuilder.createDocument(imageFile, imageID);
indexWriter.addDocument(doc);
```

A 6: Code used to insert an image's global features into a Lucene index provided by LiRe.

Appendix B

Basic Implementation of the main algorithm with the provided classes

```
public static void main(String[] args) {
    //An initial list of scores from different dimensions, e.g. terms and space
    List<List<Score<String>>> scoreLists = getAllScoreLists();

    //Normalise all scores using min-max normalization
    List<List<Score<String>>> normSCs = new ArrayList<List<Score<String>>>();
    List<Score<String>> normSC = null;

    for(List<Score<String>> scoreList: scoreLists) {
        GenericScoreFunctions.normalizeScoreListMinMax(scoreList);
        normSCs.add(normSC);
    }

    //Build up the scores for each image
    ISCBUILDER builder = SCFactory.createSCBuilder(SCBuilderType.INTERSECTED);
    List<ScoreCombination> sCs = builder.buildScoreCombination(normSCs);

    //Fuse the scores for each image
    CombinerType c = CombinerType.COMBMNZ;
    ICombiner scoreCombinator = CombinerFactory.createScoreCombiner(c, null);
    List<Score<String>> cS = scoreCombinator.combineScores(sCs);

    //Best to normalise these scores again, because they can exceed 1
    List<Score<String>> normCS = GenericScoreFunctions.normalizeScoreListMinMax(cS);

    //Re-ranking
    String imagePath = "where/the/images/are/stored/";
    String imageExtension = ".jpg";
    String contentIndexPath = "where/the/content_index/is/stored/";
    Class contentIndexClass = JCD.class;
    String contentIndexFieldName = DocumentBuilder.FIELD_NAME_JCD;

    IReranker reranker = RerankerFactory.createReranker(
        RerankerType.CLUSTER_AND_MAXIMUM_SCORE,
        imagePath,
        imageExtension,
        contentIndexPath,
        contentIndexClass,
        contentIndexFieldName,
        5,
        20
    );

    List<Score<String>> finalScores = reranker.reorderScores(normCS);

    //Displaying
    for(Score<String> score: finalScores) {
        System.out.println(score);
    }
}
```

B 1: Classes needed for a basic implementation of the main algorithm.

Appendix C

Example of an XML File with Locational Information Retrieved by YPM

```
<localScopes>
  <localScope>
    <woeId>15829</woeId>
    <type>Town</type>
    <name><![CDATA[Chester, England, GB (Town)]]></name>
    <centroid>
      <latitude>53.1973</latitude>
      <longitude>-2.89373</longitude>
    </centroid>
    <southWest>
      <latitude>53.1643</latitude>
      <longitude>-2.94378</longitude>
    </southWest>
    <northEast>
      <latitude>53.2303</latitude>
      <longitude>-2.84368</longitude>
    </northEast>
    <ancestors>
      <ancestor>
        <woeId>56616837</woeId>
        <type>District</type>
        <name><![CDATA[Cheshire West and Chester]]></name>
      </ancestor>
      <ancestor>
        <woeId>12602157</woeId>
        <type>County</type>
        <name><![CDATA[Cheshire]]></name>
      </ancestor>
      <ancestor>
        <woeId>24554868</woeId>
        <type>Country</type>
        <name><![CDATA[England]]></name>
      </ancestor>
      <ancestor>
        <woeId>23424975</woeId>
        <type>Country</type>
        <name><![CDATA[United Kingdom]]></name>
      </ancestor>
    </ancestors>
  </localScope>
</localScopes>
```

C 1: An excerpt of an XML file downloaded for the location “Chester, Cheshire, GB” using YPM.

Part of the XML file retrieved from YPM. Southwest to northeast ranges can be used to form an MBR.

Appendix D

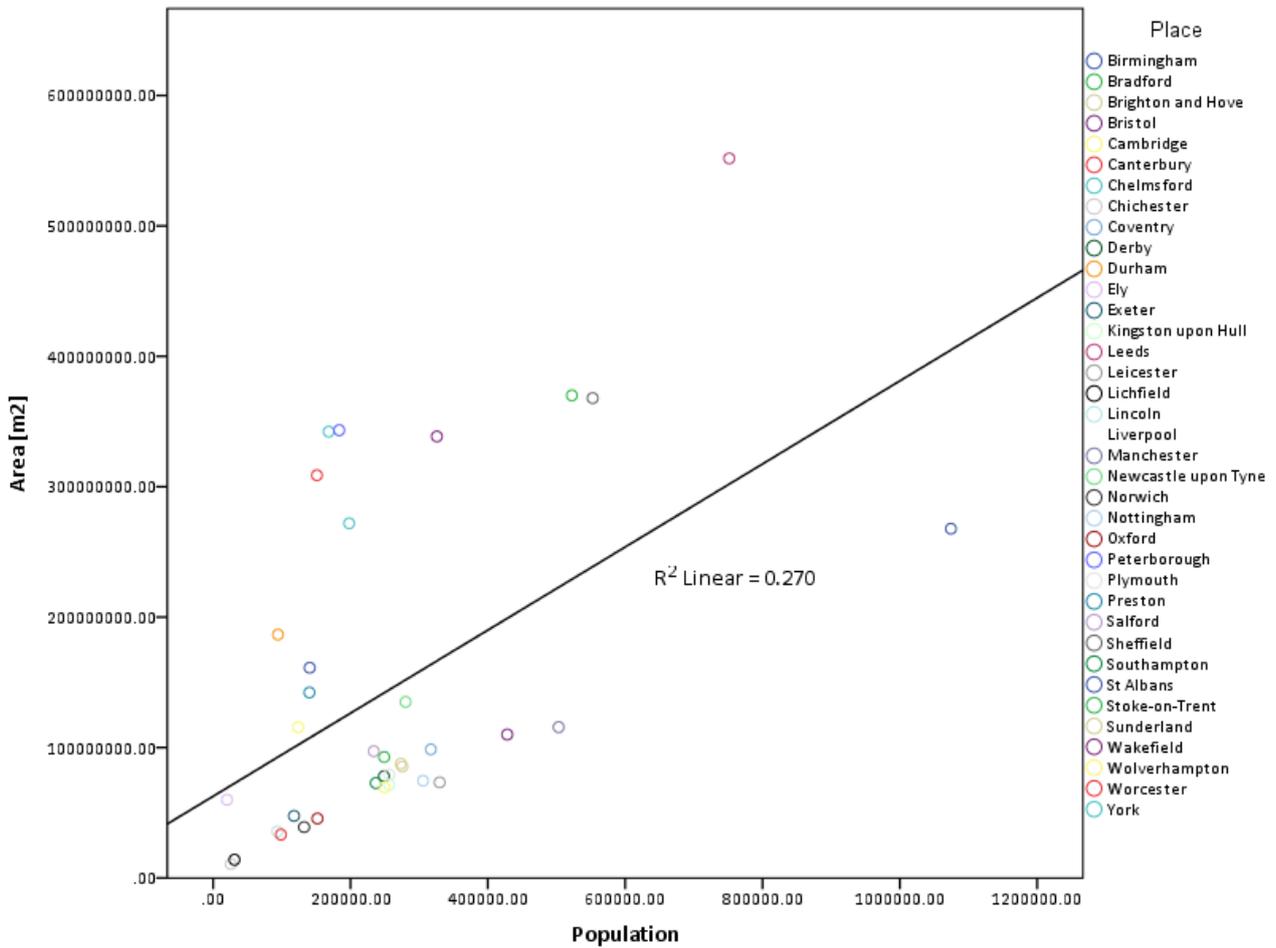
Calculation of a Linear Regression to Estimate Area from Population

Place	Population	Area [m ²]	Place	Population	Area [m ²]
Ely	20240	60000000	Stoke-on-Trent	249008	92740000
Canterbury	151145	308840000	Worcester	98768	33280000
Chelmsford	168310	342240000	Brighton and Hove	273369	87540000
Durham	94375	186680000	Derby	248752	78030000
Peterborough	183631	343380000	Coventry	316915	98640000
York	198051	271940000	Sunderland	275506	85456000
St Albans	140644	161180000	Plymouth	256384	79290000
Wakefield	325837	338600000	Southampton	236882	72800000
Preston	140202	142220000	Oxford	151906	45590000
Cambridge	123867	115650000	Norwich	132512	39020000
Leeds	751485	551720000	Kingston upon Hull	256406	71450000
Bradford	522452	370000000	Wolverhampton	249470	69440000
Sheffield	552698	367940000	Bristol	428234	110000000
Newcastle upon Tyne	280177	135000000	Birmingham	1074300	267770000
Lichfield	31068	14020000	Nottingham	305680	74610000
Salford	233933	97190000	Liverpool	466415	111840000
Chichester	25749	10670000	Manchester	503127	115650000
Exeter	117773	47600000	Leicester	329839	73320000
Lincoln	93541	35690000			

D 1: Data used for the linear regression.

Procedure for linear regression follows descriptions in Backhaus et al. (2006).

Appendix D



D 2: Scatter plot showing all 37 places with their corresponding population and area.

D 2 does not show a clear linear dependency between the two variables population and area. However, there should still be some correlation to estimate. Therefore, correlation measures are shown in D 3.

Correlations

		Population	AreaSquareMeter
Spearman's rho	Population	1.000	.501**
	Correlation Coefficient		
	Sig. (2-tailed)		.002
	N	37	37
AreaSquareMeter	AreaSquareMeter	.501**	1.000
	Correlation Coefficient		
	Sig. (2-tailed)	.002	
	N	37	37

** . Correlation is significant at the 0.01 level (2-tailed).

D 3: Spearman correlation for area and population.

D 3 reveals a slight positive correlation of 0.501, making the data at least partially useful for linear regression calculations. Spearman correlation is chosen because the area is not normally distributed. In D 4, the model summary can be seen. It reveals that only 0.27 or 27% (adjusted: 24.9%) of the variance are explained by the model, a very weak result.

Model Summary^b

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-Watson
1	.519 ^a	.270	.249	111129340,6	.515

a. Predictors: (Constant), Population

b. Dependent Variable: AreaSquareMeter

D 4: Model summary for the calculated linear regression.

The next measures taken only show the applicability of the model outside the sample. First of all, the regression coefficient needs to be different from 0. This can be calculated using the *t*-tests in D 5. H_0 : The regression coefficient is different from 0 in the whole population.

Coefficients^a

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	95.0% Confidence Interval for B	
		B	Std. Error	Beta			Lower Bound	Upper Bound
1	(Constant)	62685293,12	30135981,89		2.080	.045	1505997.355	123864588,9
	Population	318.491	88.600	.519	3.595	.001	138.624	498.359

a. Dependent Variable: AreaSquareMeter

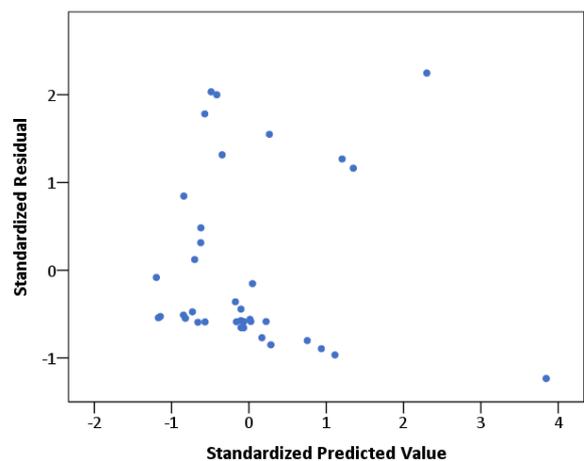
D 5: Linear regression with t-tests and confidence intervals.

If a significance level of 5% is assumed, the regression coefficient (slope) from population is different from 0. Therefore, also from D 5, a linear model with equation D 6 can be formed.

$$D\ 6 \quad \text{area} = 318.491 * \text{population} + 62685293.12$$

Preconditions that need to hold for the model to be used outside the sample are (apart from some initially holding preconditions):

1. Standardised residues are normally distributed with expectancy value 0 and SD 1 → normal distribution is NOT given (0.01) with a 5% significance level.
2. No autocorrelation in the residues visible. Durbin Watson test (D 5) is used for testing. Upper and lower bounds for $K = 37$ and $J = 1$ are: $d_L: 1.419$, $d_U: 1.530$. H_0 : the sample values are not autocorrelated. Durbin Watson value is 0.515, being far below the 1.419 lower bound. This means that H_0 has to be refused. There is a positive autocorrelation.
3. Homoscedasticity: Heteroscedasticity is a state where the variance of residues is not constant. This means that the residues depend on the independent variable and from the order of



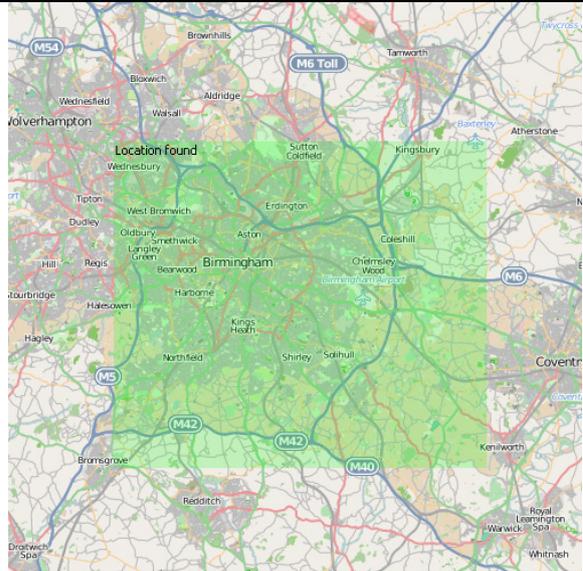
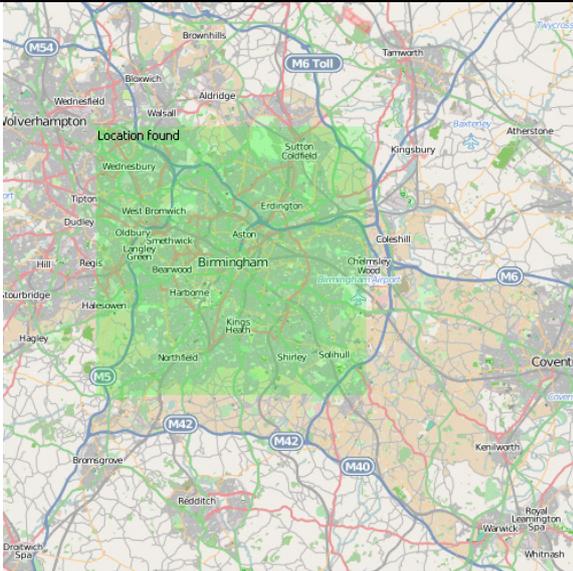
D 7: Linear regression with t-tests and confidence intervals.

occurrences. D 7, however, does not show a clear triangular pattern, which would indicate heteroscedasticity (increasing or decreasing residuals with increasing predicted value). Therefore, although it was not tested, it is assumed that homoscedasticity is given in this model.

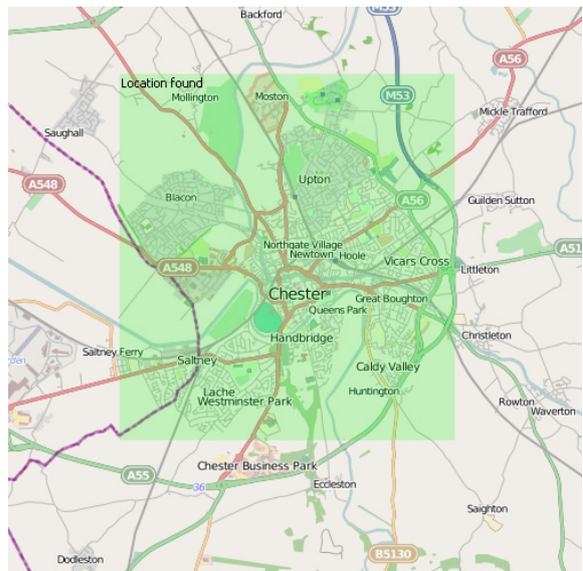
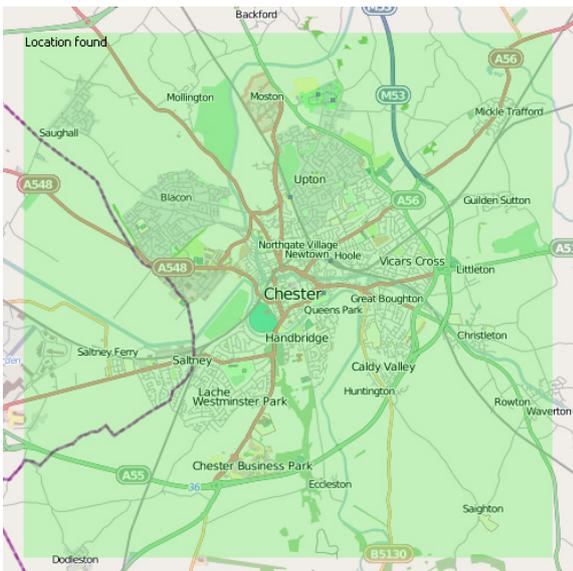
Thus, because two of three preconditions do not hold, the linear model should not be used outside the samples' extents, as was already expected. However, as a better solution than a constant value, it is still used for the estimation of areas from the population size, and D 8 gives some example MBR calculations compared to the actual MBRs provided by YPM.

GeoNames approximation

Yahoo! Placemaker (MBR)

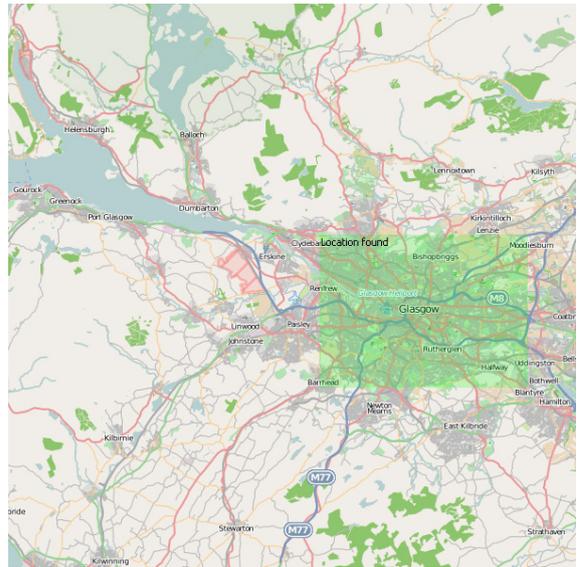
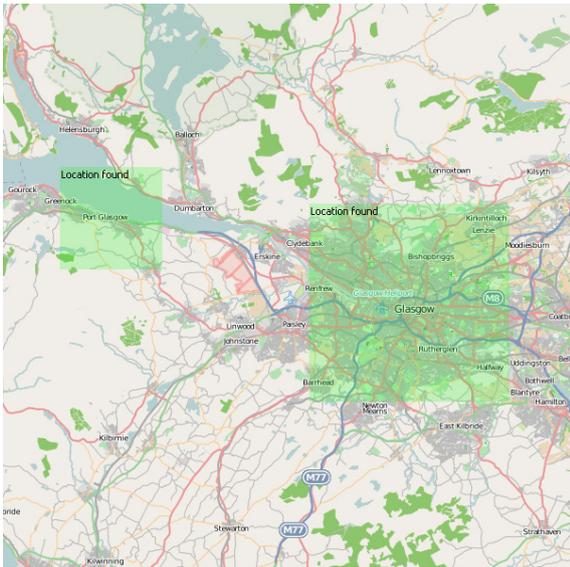


Birmingham

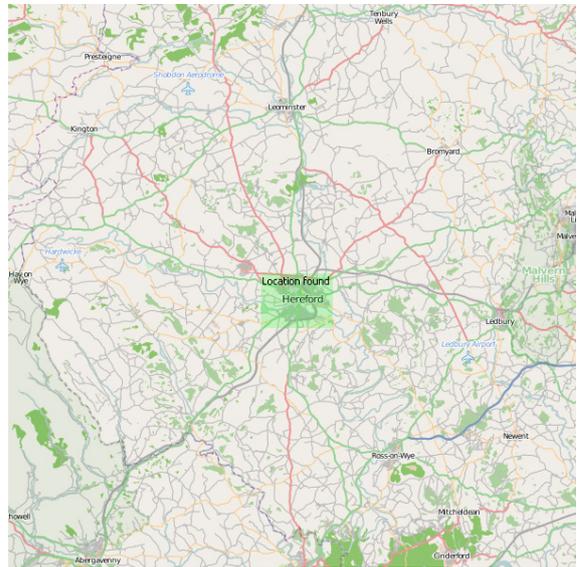
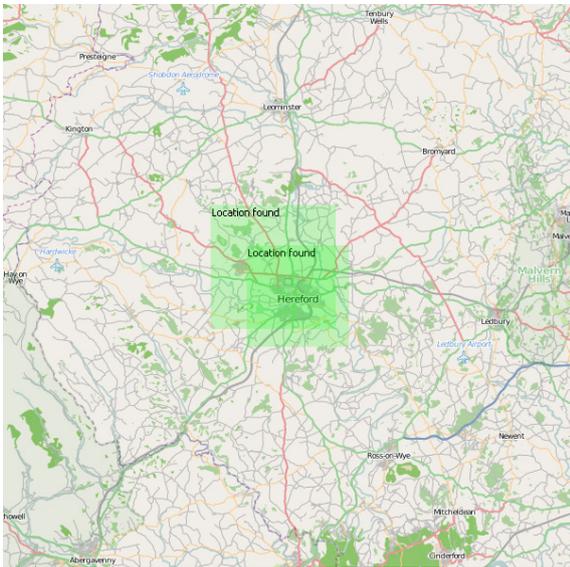


Chester

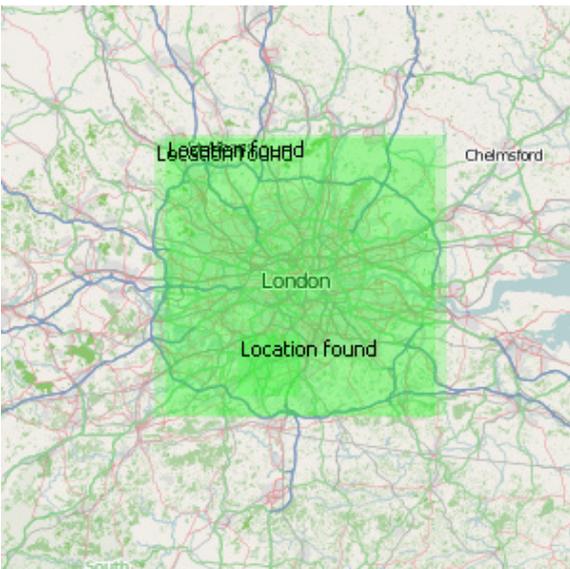
Appendix D



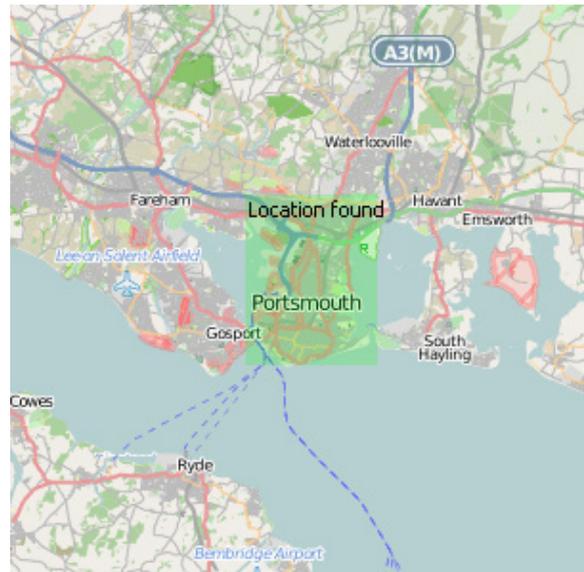
Glasgow



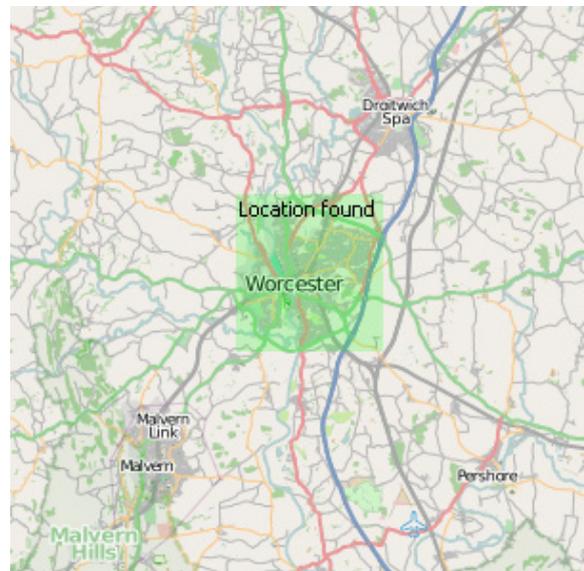
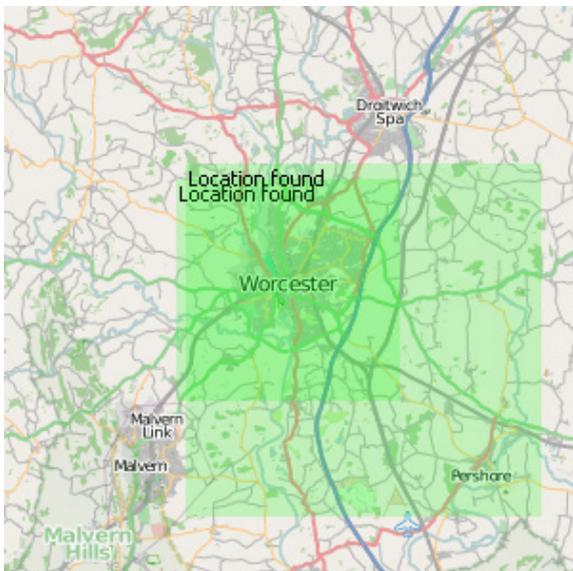
Hereford



London



Portsmouth



Worcester

D 8: Example areas calculated using the linear regression.

For many locations retrieved with GN, one advantage is visible: more than only one possible location is returned. Because a user searching the system may not only want to obvious answer but also other examples of smaller places with the same name, GN would provide the possibility to also account for such less prominent locations in queries. YPM on the other hand does much more disambiguation, leading to only one MBR per query. However, YPM provides the actual extents of a place, which GN cannot.

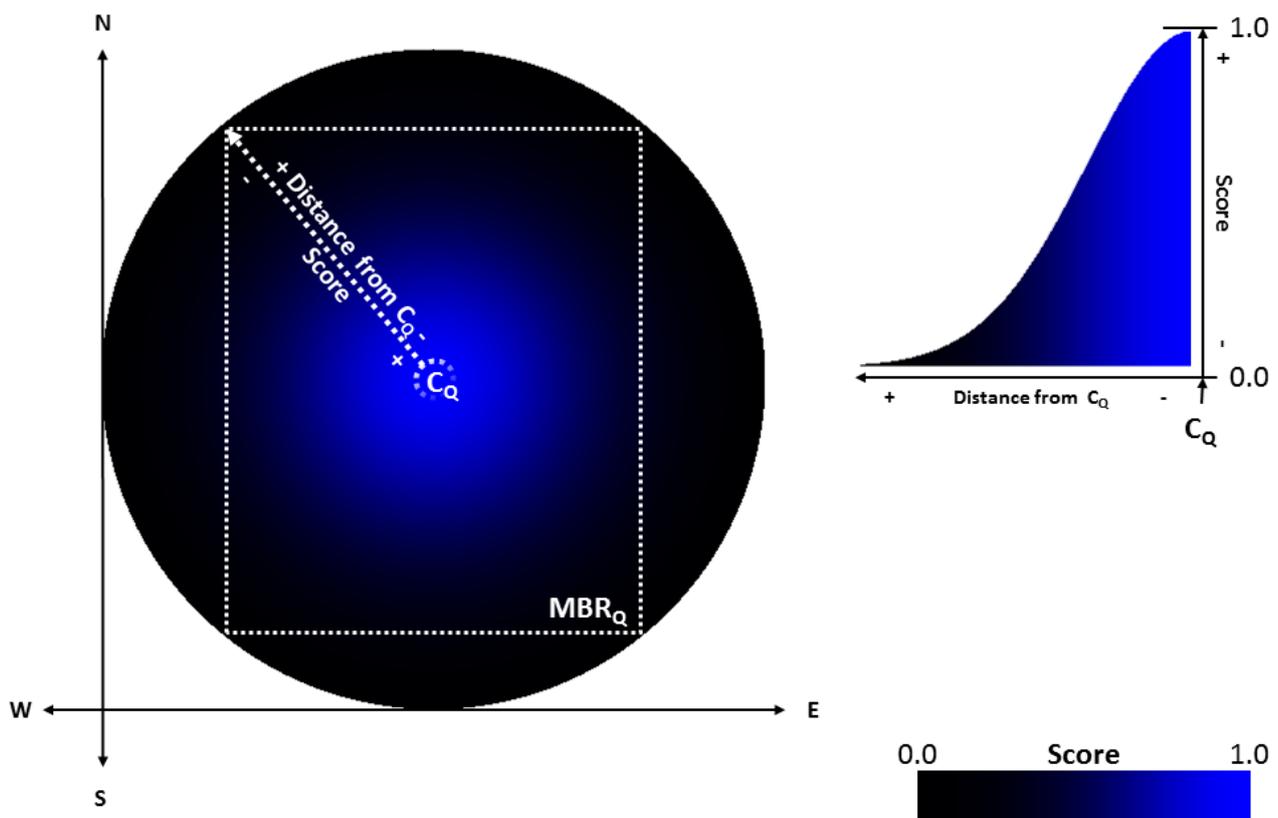
Appendix E

Exponential near relationship

An appropriate function for L has to be chosen. Experiments carried out make use of Formula E 1.

$$E\ 1 \quad L = \frac{d(P_q, P_c)}{(factor_{distance} * half\ diagonal)^2}$$

ExponentialNearRelation describes a circular near relationship around the centroid of a query location's MBR, as illustrated in E 2.



E 2: Illustration of the exponential near relationship.

C_Q represents the Query footprints centroid, MBR_Q is the query footprint. The left image illustrates the two-dimensional score distribution in the geographic space, whereas the right figure shows the actual exponential decrease from C_Q to its maximum extents (here, a distance factor smaller than one is chosen, resulting in a rather steep curve with only small values at the borders of the circle).

The score maximum is at the MBR's centroid (1.0). The score decreases towards the borders. In Formula E 1, $d(P_q, P_c)$ is again the same distance between the spatial footprints centroid and an image's point location as described in Formula VII. This way, $d(P_q, P_c)$ is squared in Formula E 1, resulting in a normal-distribution-like decay function. This curve is adapted to the footprints size by dividing it by the squared half diagonal, restricting the scattering around the MBR's centroid to a reasonable extent. Because an exponential decay can never reach 0.0, only image locations where $d(P_q, P_p) \leq half\ diagonal$ holds are assigned a score. A distance factor ($factor_{distance}$) may

alter the steepness of the curve. If it is below 1, the steepness will increase, resulting in smaller score values and eventually also decreasing the extents of what is considered to be near. On the other hand, a distance factor larger than 1 will decrease the steepness, resulting in higher scores. This way, near can be altered without being completely arbitrary. Naturally, L could also be chosen to alter $d(P_q, P_c)$ linearly, cubically, or in any other power, especially because near is not an easily described concept. However, evaluating other exponential adaptations is out of the scope of this work.

Appendix F

Data Pre-Processing Before Indexing

Database Creation. Although the provided MySQL relations could be combined using a JOIN-command each time the system needs access to the data, instead, images not containing all the needed information are discarded beforehand. Thematic and geographic data required for indexing are: title, description, and latitude and longitude coordinates in WGS 84 (units: degrees). Most of the data needed, namely image identifiers, titles and WGS 84 coordinates, can be found in the `gridimage_base` relation (downloadable here: tinyurl.com/gibase). The only missing parts are the descriptions. Descriptions are stored in the `gridimage_text` relation (downloadable here: tinyurl.com/gitext). Altogether, these relations hold textual data of 3228188 images. After joining the two relations into one, an additional field for inserting the file system path to the actual image file is added. Only images having description, title, and coordinates are added to the new relation, which finally holds data of 2255301 images.

Renaming. Before the images can be indexed, they need to be renamed. Else, matching image identifiers stored in the Geograph relations against the ones actually assigned to the images could not be accomplished. Names of images contain a trailer which prevents people from hot linking the images directly from the Geograph.org.uk servers. An example is `048003_a0241913.jpg`, where the actual image identifier stored in the relation is `48003` and everything after the underscore until the dot before the format specification is an automatically created safety string. Rather than always recalculating this safety string while inserting the images into the indexes, the trailer is removed completely.

Thumbnails Creation. A thumbnail is a small image representing the actual, mostly larger image (Margaret Rouse 20.07.2013). Such thumbnails are used in this system instead of the actual image as previews to create overviews of the retrieved image set and to save transfer time from input/output operations. All thumbnails are resized to have an equal height of 200 pixels. The thumbnail's name is the same as the name for the actual image with an additional suffix “_thumb” at the end of the image name. The new size of the images is calculated using formula F 1.

$$F\ 1 \quad Width_{Thumbnail} = Width_{Original\ image} * \left(\frac{200}{Height_{Original\ image}} \right)$$

Appendix G

Topics for Evaluation in CrowdFlower

```
<topics>
  <topic>
    <number>1</number>
    <title>Air show in England</title>
    <description>What images of air shows in England can be found?</description>
    <narrative>
      <theme>The image has to show air-show-typical scenes, like airplanes
        performing manoeuvres in the air, starting or still on the ground.
        If no airplanes are visible, it has to be clear from e. g. the
        textual description that the image was taken as part of an air show. </theme>
      <spatial>The image showing an air show has to be located within or on
        the border, but not outside England (the country). Use the map to
        determine the location of the image. </spatial>
    </narrative>
  </topic>
  <topic>
    <number>2</number>
    <title>Railway Station in Derby</title>
    <description>What images of railway stations in the city of Derby can
      be found?</description>
    <narrative>
      <theme>The image has to show a railway station, either from in- or
        outside. It should be clear from the image that what can be seen on
        the image is part of a railway station. </theme>
      <spatial>The railway station has to be located within or on the
        border, but not outside the city of Derby (in the East Midlands
        region of England). Use the map to determine the location of the
        image. </spatial>
    </narrative>
  </topic>
  <topic>
    <number>3</number>
    <title>Castle near Aberdeenshire</title>
    <description>What images of castles near Aberdeenshire can be found?</description>
    <narrative>
      <theme>The image has to show one or more castles. A castle is a
        private fortified residence of a lord or noble. Despite the actual
        meaning, similar structures like palaces (which are not fortified)
        are also considered relevant to this topic. The age of a castle does
        not matter. A church is not considered a castle, although they can
        look similar. </theme>
      <spatial>The castle has to be located inside or on the borders of
        Aberdeenshire (the unitary authority in Scotland) or 'close' around
        it. Because of the apparent vagueness of the meaning of 'near', it
        is for the assessor to decide what is still considered near
        Aberdeenshire. Use the map to determine the location of the image. </spatial>
    </narrative>
  </topic>
  <topic>
    <number>4</number>
    <title>Loch near Highlands</title>
    <description>What images of Lochs near the Scottish Highlands can be
      found?</description>
    <narrative>
      <theme>The image has to show a loch. A loch is the Scottish Gaelic
        and Irish word for a lake or a sea inlet. </theme>
      <spatial>The loch has to be located inside or on the borders of the
        Scottish Highlands or 'close' around it. Because of the apparent
        vagueness of the meaning of 'near', it is for the assessor to decide
        what is still considered near the Scottish Highlands. Use the map to
        determine the location of the image. </spatial>
    </narrative>
  </topic>
  <topic>
    <number>5</number>
    <title>Bridge near Birmingham</title>
    <description>What images of bridges near Birmingham can be found?</description>
    <narrative>
      <theme>The image has to show a bridge. There are no restrictions on
        what kind of a bridge it has to be. Large or small bridges are
        considered relevant. </theme>
      <spatial>The bridge has to be located inside or on the borders of
        Birmingham (the city and metropolitan borough in the West Midlands
        of England) or 'close' around it. Because of the apparent vagueness
        of the meaning of 'near', it is for the assessor to decide what is
        still considered near Birmingham. Use the map to determine the
        location of the image. </spatial>
    </narrative>
  </topic>
</topics>
```

Appendix G

```
<topic>
  <number>6</number>
  <title>Petrol station near London</title>
  <description>What images of petrol stations near London can be found?</description>
  <narrative>
    <theme>The image has to show a petrol station, where people can buy
      fuel for their cars, motorcycles etc. </theme>
    <spatial>The petrol station has to be located inside or on the
      borders of London (the capital city of England and the United
      Kingdom) or 'close' around it. Because of the apparent vagueness of
      the meaning of 'near', it is for the assessor to decide what is
      still considered near London. Use the map to determine the location
      of the image. </spatial>
  </narrative>
</topic>
<topic>
  <number>7</number>
  <title>Church north of Norfolk</title>
  <description>What images of churches in the north or to the north of
    Norfolk can be found?</description>
  <narrative>
    <theme>The image has to show one or more church, either from in- or
      from outside. A church is a Christian religious institution or
      building. </theme>
    <spatial>The church has to be located north of Norfolk (the county in
      the east of England) (either in- or outside of Norfolk, the
      requirement is that it can be regarded as certainly not in any other
      direction of Norfolk (west, east, or south)). </spatial>
  </narrative>
</topic>
<topic>
  <number>8</number>
  <title>Beach north of Newquay</title>
  <description>What images of beaches in the north or to the north of
    Newquay can be found?</description>
  <narrative>
    <theme>The image has to show a beach. A beach is a landform along the
      shorelines of an ocean, sea, lake or river. It does not need to be a
      certain type of beach like sand beach, but can also be made of
      gravel, shingle, pebbles, cobblestone, etc. </theme>
    <spatial>The beach has to be located north of Newquay (the town in
      Cornwall, England), either in- or outside of Newquay. The
      requirement is that it can be regarded as certainly not in any other
      direction of Newquay (west, east, or south). Use the map to
      determine the location of the image. </spatial>
  </narrative>
</topic>
<topic>
  <number>9</number>
  <title>Canal north of Glasgow</title>
  <description>What images of canals in the north or to the north of
    Glasgow can be found?</description>
  <narrative>
    <theme>The image has to show a water canal. A canal is a man-made
      channel for water, either waterways or aqueducts. It therefore may
      look similar to a river. </theme>
    <spatial>The canal has to be located north of Glasgow (the largest
      city in Scotland), either in- or outside of Glasgow. The requirement
      is that it can be regarded as certainly not in any other direction
      of Glasgow (west, east, or south). Use the map to determine the
      location of the image. </spatial>
  </narrative>
</topic>
<topic>
  <number>10</number>
  <title>Ice sculptures south of South Kensington</title>
  <description>What images of ice sculptures in the south or to the
    south of South Kensington can be found?</description>
  <narrative>
    <theme>The image has to show one or more ice-made sculptures. The raw
      material of the sculptures has to be ice. The sculptures can be
      abstract or realistic. </theme>
    <spatial>The image showing ice sculptures has to be located south of
      South Kensington (in London, England), either in- or outside of
      South Kensington. The requirement is that it can be regarded as
      certainly not in any other direction of South Kensington (west,
      east, or north). Use the map to determine the location of the image. </spatial>
  </narrative>
</topic>
```

Appendix G

```
<topic>
  <number>11</number>
  <title>Ship south of Carrickfergus</title>
  <description>What images of ships in the south or to the south of
    Carrickfergus can be found?</description>
  <narrative>
    <theme>The ship has to be the main topic of the image. There can be
      more than one ship in an image. Boats are also considered relevant. </theme>
    <spatial>The ship has to be located south of Carrickfergus (the town
      in Northern Ireland), either in- or outside of Carrickfergus. The
      requirement is that it can be regarded as certainly not in any other
      direction of Carrickfergus (west, east, or north). Use the map to
      determine the location of the image. </spatial>
  </narrative>
</topic>
<topic>
  <number>12</number>
  <title>Crossroads south of London</title>
  <description>What images of crossroads in the south or to the south of
    London can be found?</description>
  <narrative>
    <theme>An image has to show a crossroads where cars cross. A
      crossroads is an intersection or road junction, where two or more
      roads either meet or cross. </theme>
    <spatial>The crossroads has to be located south of London (the
      capital city of England and the United Kingdom), either in- or
      outside of London. The requirement is that it can be regarded as
      certainly not in any other direction of London (west, east, or
      north). Use the map to determine the location of the image. </spatial>
  </narrative>
</topic>
<topic>
  <number>13</number>
  <title>Hotel south of Swindon</title>
  <description>What images of hotels in the south or to the south of
    Swindon can be found?</description>
  <narrative>
    <theme>The image has to show a hotel. Anybody who is looking for a
      hotel and is given the image should see that there is a hotel in the
      image. </theme>
    <spatial>The hotel has to be located south of Swindon (the town in
      the ceremonial county of Wiltshire, in South West England), either
      in- or outside of Swindon. The requirement is that it can be
      regarded as certainly not in any other direction of Swindon (west,
      east, or north). Use the map to determine the location of the image. </spatial>
  </narrative>
</topic>
<topic>
  <number>14</number>
  <title>Museum west of Liverpool</title>
  <description>What images of museums in the west or to the west of
    Liverpool can be found?</description>
  <narrative>
    <theme>The image has to show a museum. All kinds of museums are
      considered relevant (e. g. scientific, artistic, cultural,
      historical, etc. ). </theme>
    <spatial>The museum has to be located west of Liverpool (the city in
      North West England), either in- or outside of Liverpool. The
      requirement is that it can be regarded as certainly not in any other
      direction of Liverpool (north, east, or south). Use the map to
      determine the location of the image. </spatial>
  </narrative>
</topic>
<topic>
  <number>15</number>
  <title>Harbour west of Portsmouth</title>
  <description>What images of harbours in the west or to the west of
    Portsmouth can be found?</description>
  <narrative>
    <theme>The image has to show a harbour, either seen from a ship/boat
      or from land. A harbour is a body of water where ships, boats, etc.
      can seek shelter from weather or are stored for future use. The
      harbour can be artificial or natural. </theme>
    <spatial>The harbour has to be located west of Portsmouth (the city
      in the ceremonial county of Hampshire on the south coast of
      England), either in- or outside of Portsmouth. The requirement is
      that it can be regarded as certainly not in any other direction of
      Portsmouth (north, east, or south). Use the map to determine the
      location of the image. </spatial>
  </narrative>
</topic>
```

Appendix G

```
<topic>
  <number>16</number>
  <title>Minster west of Howden</title>
  <description>What images of minsters in the west or to the west of
    Howden can be found?</description>
  <narrative>
    <theme>The image has to show the outside or the inside of a minster.
      It is also allowed to only show part of the minster, but it has to
      be clear for people looking at the image that this could be part of
      a minster. A minster is a church. Minster is an honorific title
      given to particular churches in England. </theme>
    <spatial>The minster has to be located west of Howden (the small town
      and civil parish in the East Riding of Yorkshire, England), either
      in- or outside of Howden. The requirement is that it can be regarded
      as certainly not in any other direction of Howden (north, east, or
      south). Use the map to determine the location of the image. </spatial>
  </narrative>
</topic>
<topic>
  <number>17</number>
  <title>Cottage south of Glasgow</title>
  <description>What images of cottages in the south or to the south of
    Glasgow can be found?</description>
  <narrative>
    <theme>The image has to show a cottage. A cottage is a modest
      dwelling, typically located in rural or semi-rural sites. It tends
      to be of traditional build, but can also be modern, sometimes
      imitating traditional dwellings. </theme>
    <spatial>The image showing cottages has to be located south of
      Glasgow (the largest city in Scotland), either in- or outside of
      Glasgow. The requirement is that it can be regarded as certainly not
      in any other direction of Glasgow (west, east, or north). Use the
      map to determine the location of the image. </spatial>
  </narrative>
</topic>
<topic>
  <number>18</number>
  <title>Mountain east of Scotland</title>
  <description>What landscape images of mountains in the east or to the
    east of Scotland can be found?</description>
  <narrative>
    <theme>The image has to show landscape with mountains. Also smaller,
      less steep hills are considered relevant, but not as relevant as
      actual mountains. </theme>
    <spatial>The image of mountains has to be located east of Scotland
      (the country), either in- or outside of Scotland. The requirement is
      that it can be regarded as certainly not in any other direction of
      Scotland (west, north, or south). Use the map to determine the
      location of the image. </spatial>
  </narrative>
</topic>
<topic>
  <number>19</number>
  <title>Waves west of Dorset</title>
  <description>What images of waves in the west or to the west of Dorset
    can be found?</description>
  <narrative>
    <theme>The image has to show a water wave on any liquid surface,
      showing characteristics of waves as for example white foam and/or
      ripples. </theme>
    <spatial>The image of waves has to be located west of Dorset (The
      county in South West England on the English Channel coast), either
      in- or outside of Dorset. The requirement is that it can be regarded
      as certainly not in any other direction of Dorset (north, east, or
      south). Use the map to determine the location of the image. </spatial>
  </narrative>
</topic>
<topic>
  <number>20</number>
  <title>Sunset north of Lancashire</title>
  <description>What images of sunsets in the north or to the north of
    Lancashire can be found?</description>
  <narrative>
    <theme>The image has to show a Sunset. A sunset is the daily
      disappearance of the Sun below the western half of the horizon. </theme>
    <spatial>The image of the sunset has to be located north of
      Lancashire (the non-metropolitan county in the North West of
      England), either in- or outside of Lancashire. The requirement is
      that it can be regarded as certainly not in any other direction of
      Lancashire (west, east, or south). Use the map to determine the
      location of the image. </spatial>
  </narrative>
</topic>
```

Appendix G

```
<topic>
  <number>21</number>
  <title>Hill east of Midlothian</title>
  <description>What images of hills in the east or to the east of
    Midlothian can be found?</description>
  <narrative>
    <theme>The image has to show a landscape with hills. A hill is a
      landform that extends above the surrounding terrain. A hill is
      generally considered lower than a mountain, but this interpretation
      may be subjective. </theme>
    <spatial>The image of hills has to be located east of Midlothian (one
      of the 32 council areas in Scotland), either in- or outside of
      Midlothian. The requirement is that it can be regarded as certainly
      not in any other direction of Midlothian (west, north, or south).
      Use the map to determine the location of the image. </spatial>
  </narrative>
</topic>
<topic>
  <number>22</number>
  <title>Waterfall east of Northern Ireland</title>
  <description>What images of waterfalls in the east or to the east of
    Northern Ireland can be found?</description>
  <narrative>
    <theme>The image has to show a waterfall. A waterfall is a place
      where water flows over a vertical drop in the course of a stream or
      river. </theme>
    <spatial>The waterfall has to be located east of Northern Ireland
      (the country), either in- or outside of Northern Ireland. The
      requirement is that it can be regarded as certainly not in any other
      direction of Northern Ireland (west, north, or south). Use the map
      to determine the location of the image. </spatial>
  </narrative>
</topic>
<topic>
  <number>23</number>
  <title>Island east of Cornwall</title>
  <description>What images of islands in the east or to the east of
    Cornwall can be found?</description>
  <narrative>
    <theme>The image has to show an island. An island or isle is any
      piece of sub-continental land that is surrounded by water. </theme>
    <spatial>The image of the island has to be located east of Cornwall
      (the unitary authority and ceremonial county of England), either in-
      or outside of Cornwall. The requirement is that it can be regarded
      as certainly not in any other direction of Cornwall (west, south, or
      south). Use the map to determine the location of the image. </spatial>
  </narrative>
</topic>
<topic>
  <number>24</number>
  <title>Cemetery in Chester</title>
  <description>What images of cemeteries in Chester can be found?</description>
  <narrative>
    <theme>The image has to show a cemetery and it should be clear from
      the image that what somebody looks at is part of a cemetery. </theme>
    <spatial>The cemetery has to be located within or on the border, but
      not outside Chester (the city in Cheshire, England). Use the map to
      determine the location of the image. </spatial>
  </narrative>
</topic>
<topic>
  <number>25</number>
  <title>Pub in York</title>
  <description>What images of pubs in York can be found?</description>
  <narrative>
    <theme>The image has to show one or more pubs. A pub is a drinking
      establishment fundamental to the culture of Britain. Synonyms to pub
      (e. g. arms) are also considered relevant to this topic. Pubs can be
      part of a building or buildings themselves. </theme>
    <spatial>The pub has to be located within or on the border, but not
      outside of York (the city in North Yorkshire, England). Use the map
      to determine the location of the image. </spatial>
  </narrative>
</topic>
</topics>
```

Appendix H

3 Steps to CrowdFlower Task/Job Creation

1st step: obtaining images from the search systems. Images are pooled together by submitting each of the 25 queries to all of the three systems (T, TS, and TSCR) and saving the first 10 images retrieved by each system. Each image is represented by one line of information, separated by commas and saved in a CSV-(Comma Separated Values)-file. The header format for these files can be seen in H 1.

	id	title	description	realname	latitude	longitude
	internetlink	maplink	querytitle	querydescr	querynarrative	
Legend	<i>id</i>	An image's unique identifier.				
	<i>title</i>	Title assigned by the person who took the image.				
	<i>description</i>	Text describing the image, assigned by the person who took it.				
	<i>realname</i>	The name of the person that took the image.				
	<i>internetlink</i>	Link to the image's location on the web.				
	<i>maplink</i>	Link to the image's map location on the web.				
	<i>querytitle</i>	The query submitted to the system that retrieved this image.				
	<i>querydescr</i>	A brief description of what the image's topic should be about.				
	<i>querynarrative</i>	A more elaborate and thorough description of the requirements an image should meet to be considered relevant.				

H 1: Header row of an image's data.

Additionally, the ranks or order in which the 10 images appear on retrieval is saved in a separate text file. Altogether, 750 images are obtained of which 138 occur more than once. These 138 doubly occurring images are removed so that finally, 612 different images are pooled together for all three systems and 25 queries for evaluation.

2nd step: generating random tasks. All the generated CSV files have to be split into separate jobs. A job consists of 4 valid images, each image belonging to the same topic (there is no topic mixture in one job). Each job has a random selection of image data lines to assure that the images do not all come from the same system.

3rd step: adding a fake image. *Traps* have to be set for those judges that do not want to submit honest judgements. One of these traps is an additional fake image in each job (resulting in 5 images to judge for each job). H 2 displays the 4 used fake images. The images are randomly assigned to a job and occur at different positions within it. Neither theme nor locations of these images are anyhow related to the topic to judge. The fake images are all taken especially for the thesis not to violate any possible copyrights. For all the 612 images obtained by querying the 3 systems with 25 queries, 163 Jobs are created, each containing between 1 and 4 "real" images to judge and 1 fake image.



H 2: The 4 fake images used to identify untrustworthy judges.
Each job contained one of these randomly assigned images.

H 3 shows the job description as displayed to the CrowdFlower judges who assess the images' relevance to a topic. In H 4, a unit of a job is displayed. For space reasons, the actual image and map are cut off (indicated by the white triangular trailer of the image and map).

Task Description

A) Introduction

Our research is concerned with developing systems which retrieve images in response to a textual query. Your task in this job is to judge the quality of image search results for a search engine, which also takes account of where an image was taken.

B) Task: Relevance judgement

Your task is to judge the relevance of an image.

To do this, you need to do the following:

1. First, read the topic description. A topic consists of:
 - o A **query**. This is what was submitted to the search engine.
 - o A **description**. A brief summary of what the actual image retrieved should show.
 - o A **narrative** that describes in more detail what is expected as a response to this query.

For every image on this page, the same topic can be used.

2. Having read the topic, you should judge the relevance of the images on this page according to the following scale:
 - o **Highly relevant image**
The image (together with its texts and location) fulfils **all** the requirements stated in the topic.
 - o **Fairly relevant image**
The image (together with its texts and location) fulfils **most, but not all**, of the requirements stated in the topic.
 - o **Marginally relevant image**
The image (together with its texts and location) fulfils **only one** of the requirements stated in the topic.
 - o **Irrelevant image**
The image (together with its texts and location) **doesn't fulfil** any requirement stated in the topic.
 - o **Not sure**
You're not sure if the image matches the topic. In this case it is particularly important that you add some text (see below) explaining why you were not sure.

IMPORTANT NOTE: judge theme **AND** location of an image

- o An image having the right theme, but a wrong location, should not be considered irrelevant
- o An image having the right location, but a wrong theme, should not be considered irrelevant, either
- o An image should only be considered irrelevant if neither theme nor location of the image have any connection to the topic

3. After judging the relevance of the images, we'd ask you to explain, in your own words, how you made your decision. This text is very valuable for our research, because it may help us understand differences between different annotators. **This explanation can be very short.** Remember that **there are no right or wrong answers!!**

C) Example relevance judgement

For each image you are asked to judge, you will see the following:

H 3: Job description displayed before every job.

Topic:

Title:

Ship south of Carrickfergus

Description:

What images of ships in the south or to the south of Carrickfergus can be found?

Narrative:

The ship has to be the main topic of the image. There can be more than one ship in an image. Boats are also considered relevant. The ship has to be located south of Carrickfergus (the town in Northern Ireland), either in- or outside of Carrickfergus. The requirement is that it can be regarded as certainly not in any other direction of Carrickfergus (west, east, or north). Use the map to determine the location of the image.

The image to judge for this topic:

Image:



(c) Copyright David Hawgood and licensed under this [Creative Commons Licence](#)

Image title:

Radio control tower, Carrickfergus harbour

Image description:

The tower is now derelict, but it was used to control shipping within Belfast Lough. See NI Towns website http://ni_towns.tripod.com/carrickfergus/carrickfergus-harbour.html .

Image location:



To see a zoom- and panable map, click on the map above

Choose one

- Highly relevant image
- Fairly relevant image
- Marginally relevant image
- Irrelevant image
- Don't know

Please briefly explain why you chose this rank for the image above. This is very important for our research.

H 4: Example of a job to judge by a CrowdFlower assessor.

Appendix I

Aggregated RJs of a Trusted Person and Averaged RJs of the Crowd Used in the Correlation Analysis

1	T	C	2	T	C	3	T	C	4	T	C	5	T	C	6	T	C	7	T	C	8	T	C	9	T	C	10	T	C
1022042	1	0	1928301	1	1	1223408	3	3	1066526	1	1	1729358	3	3	1223838	1	0	1868170	3	3	1527675	3	3	1336570	1	1	1116507	3	3
1041441	2	2	1659426	2	3	1685253	3	3	1415099	3	3	1729398	3	3	1144374	3	3	1936963	1	1	1758789	3	3	1547067	3	3	1117482	3	3
1041459	2	3	1087000	3	3	1496285	1	0	1233901	3	3	1719576	3	3	1661715	3	3	1384558	3	2	1226397	2	2	1515105	1	2	1120331	3	3
1320847	3	3	1059485	1	2	1943378	3	3	1489117	3	3	1369664	2	3	1766336	3	3	1565860	3	3	1160924	3	3	1778286	1	0	1607373	1	0
1366365	3	3	1091233	2	1	1566773	3	3	1246865	2	3	1369659	2	3	1721098	3	2	1959743	2	1	1228598	3	3	1778291	1	0	1120260	3	3
1376013	3	3	1087048	3	3	1364515	3	3	1850974	3	3	1725102	3	3	1785115	1	1	1547395	3	3	1759506	3	3	1177267	1	2	1113996	1	0
1376248	3	3	1099384	3	3	1690889	3	3	1256364	2	3	1730424	3	3	1974764	3	3	1547774	3	3	1228611	3	3	1172947	3	3	1607368	1	0
1379690	3	3	1095078	3	3	1945271	1	1	1197197	3	3	1729415	3	3	1445812	3	3	1547410	3	3	1246387	3	3	1627147	3	3	1117400	3	2
1379695	3	3	1659388	2	2	1871327	3	3	1589597	3	3	1720379	3	3	1975023	3	3	1242999	3	3	1758797	3	3	1171428	3	3	1721420	1	0
1411394	3	2	1659398	2	2	1872762	3	3	1113154	0	0	1255672	3	3	1517103	1	2	1547781	3	3	1527509	3	3	1171452	1	0	1117767	3	3
1411408	3	3	1659438	2	2	1684320	3	3	1361331	1	1	1369670	2	3	1913769	1	2	1936773	2	1	1758958	3	3	1336583	3	3	1448684	1	0
1417131	2	2	1699631	1	0	1223490	3	3	1831201	3	3	1725353	3	3	1806771	3	3	1545518	3	3	1759344	3	3	1781413	1	0	1569254	1	0
1417148	2	3	1087011	3	3	1864575	3	3	1831217	3	3	1255685	3	3	1013303	3	3	1545516	3	3	1759828	3	3	1300385	1	0	1117610	3	3
1426512	3	3	1794236	1	0	1778282	3	3	1775524	1	2	1726152	3	3	1822411	1	2	1269073	3	3	1759042	3	3	1780173	1	0	1324592	1	0
1426516	3	3	1087057	3	3	1176982	3	3	1857765	1	1	1720434	3	3	1482188	1	1	1547785	3	3	1226698	2	3	1379328	0	0	1120326	3	3
1426526	3	3			1275711	3	3	1336467	1	0	1730321	3	3	1508578	3	3	1566507	3	3	1759461	3	3	1518223	3	3	1116510	3	3	
1427755	3	3			1685320	3	3	1379871	2	3	1369673	3	3	1066382	3	3	1045398	3	3	1199051	3	3	1780176	1	1	1120334	3	3	
1468509	3	3			1871448	3	3	1343431	3	3	1726015	3	3	1119800	1	1	1869873	1	2	1758985	3	3	1171402	3	3	1739923	1	0	
1468516	3	2			1966672	1	1	1815269	2	2	1730530	3	3	1597422	3	3	1968231	1	0	1672653	3	3	1171389	3	3	1116226	3	3	
1667599	1	0			1039468	3	3	1415114	3	3	1725127	3	3	1092923	3	3	1945479	3	3	1759025	3	3	1171430	3	3	1116702	3	3	
1871597	3	3			1482685	3	3	1143031	3	3	1725118	3	3	1597619	1	0	1547408	3	3	1226836	2	3	1176464	3	3	1575037	0	0	
1871599	3	3			1319440	1	1	1376327	2	3	1369665	3	3	1407576	3	3	1873976	3	3	1527528	3	3	1336575	3	3	1120325	3	3	
1889274	3	3			1871738	3	3	1415177	3	3	1727591	3	3	1508697	2	2	1055990	3	3	1758935	3	3	1171405	3	3	1117754	3	3	
					1342967	1	1	1035198	1	1	1369667	3	3	1814383	3	3	1555169	3	2			1171461	1	1	1116878	3	3		
					1566794	2	2	1199495	1	1	1190646	3	3	1633979	3	3	1269203	3	3			1173062	3	3	1862113	0	0		
								1867584	3	3	1732209	3	3			1936954	2	2			1171355	3	3						
								1887075	2	2	1730664	3	3			1547393	3	3			1336565	2	1						
										1725094	3	3			1243020	3	3			1787359	2	3			1810397	0	0		
															1879078	3	3												

Appendix I

11	T	C	12	T	C	13	T	C	14	T	C	15	T	C	16	T	C	17	T	C	18	T	C	19	T	C	20	T	C
1449986	3	3	1345115	3	3	1123597	1	0	1162336	3	3	1990451	3	3	1623734	3	3	1484253	3	3	1405146	0	1	1228287	3	3	1863502	3	3
1628252	1	1	1935145	1	1	1199976	2	1	1162266	1	1	1368366	3	3	1202079	3	3	1642648	3	3	1060189	1	1	1790508	1	2	1746861	3	3
1449985	3	3	1532613	3	3	1199978	3	3	1639545	3	3	1598020	3	3	1266114	3	3	1279133	0	1	1752518	3	3	1184319	3	3	1101459	3	3
1450015	3	3	1080619	2	3	1333129	2	2	1850383	3	3	1416116	3	3	1326038	1	1	1219820	1	0	1616454	1	1	1371801	0	0	1029976	2	3
1449071	3	3	1345124	3	3	1482371	3	3	1850377	3	3	1990672	3	3	1624978	3	3	1535335	1	0	1466954	1	2	1195470	3	3	1928551	0	0
1449988	3	3	1572466	2	1	1527229	2	1	1158649	2	2	1986007	2	1	1635104	2	3	1200184	2	1	1037667	1	1	1447989	2	2	1540567	3	3
1585700	3	3	1658824	1	2	1626586	3	2	1002744	1	1	1037655	2	1	1623733	3	3	1450290	3	3	1370196	2	2	1594399	2	3	1904254	1	2
1860054	1	0	1030185	0	0	1705339	0	0	1738411	3	3	1309742	3	3	1623721	3	3	1005827	2	3	1063238	2	2	1779476	3	3	1664269	3	3
1072863	3	3	1680763	3	3	1716234	1	0	1020938	3	3	1346164	3	3	1202071	3	3	1678568	0	0	1284919	1	2	1287478	2	3	1220045	2	3
1040476	3	3	1409249	1	0	1737315	0	0	1433339	1	0	1508049	3	3	1265450	2	2	1657830	3	3	1769235	1	2	1160628	3	3	1736454	3	3
1861010	1	0	1079034	2	3	1843414	0	0	1850238	0	0	1255956	3	3	1623729	3	3	1692093	1	0	1566505	1	2	1661490	1	1	1904155	1	0
1629961	3	3	1754520	2	1	1857076	1	1	1158953	1	0	1272867	3	3	1265456	2	2	1869336	1	0	1827036	2	1	1694812	0	0	1167629	3	3
1583961	1	1	1795126	1	2	1857180	1	2	1433355	1	0	1013921	3	3	1202098	2	3	1821585	3	3	1197790	3	2	1359437	3	3	1912011	1	1
1450158	3	3	1787736	2	1	1896581	3	3	1021246	3	3	1037684	3	3	1618326	3	3	1464762	3	3	1723243	2	2	1183868	2	2	1712810	1	0
1450169	3	3	1935147	1	0	1896582	3	3	1147393	2	1	1156559	3	3	1624996	3	2	1443744	0	0	1822482	1	1	1369832	0	0	1101116	3	3
1450155	3	3	1332438	2	2	1896583	3	3	1202099	3	3	1511041	3	3	1264617	3	3	1484238	3	2	1752425	2	1	1591503	2	3	1134455	2	3
1449993	3	3	1103953	2	2	1896584	3	3	1162485	2	3	1598031	3	3	1201160	3	3	1778592	1	1	1712775	3	3	1039483	3	3	1664352	3	3
1450192	3	3	1140829	2	3	1896585	3	3	1162653	2	3	1115938	3	3	1265471	3	3	1687065	1	1	1719871	1	2	1034695	2	3	1220052	2	3
1044810	3	3	1658621	1	1	1896588	3	3	1988043	3	3	1255906	3	3				1871177	3	3	1826527	2	2	1068627	0	0	1134082	2	3
1025474	3	3	1409343	1	1	1899188	3	3	1040190	3	3	1322143	3	3				1897646	3	3	1377062	1	0	1295804	2	3	1766293	1	1
1727068	3	3	1871383	1	1	1899238	3	3	1162301	3	3	1306319	3	3				1480585	3	3	1964353	1	1	1195879	3	3	1709978	3	3
1449983	3	3	1597292	1	2	1899254	3	3				1930525	2	3				1875424	3	3	1719254	1	1	1215235	0	0	1276132	0	0
			1860596	1	1	1949577	1	1				1037666	3	3				1492214	1	1	1526712	2	2	1591466	2	3	1583344	3	3
			1745779	2	1							1041254	2	1				1484301	1	0	1295254	2	3	1491026	0	0	1674818	2	3
			1675804	3	3							1037643	3	3				1464800	3	2	1827041	3	2			1186939	1	1	
																				1506004	2	2			1904161	1	1		
																										1211403	3	3	

Appendix I

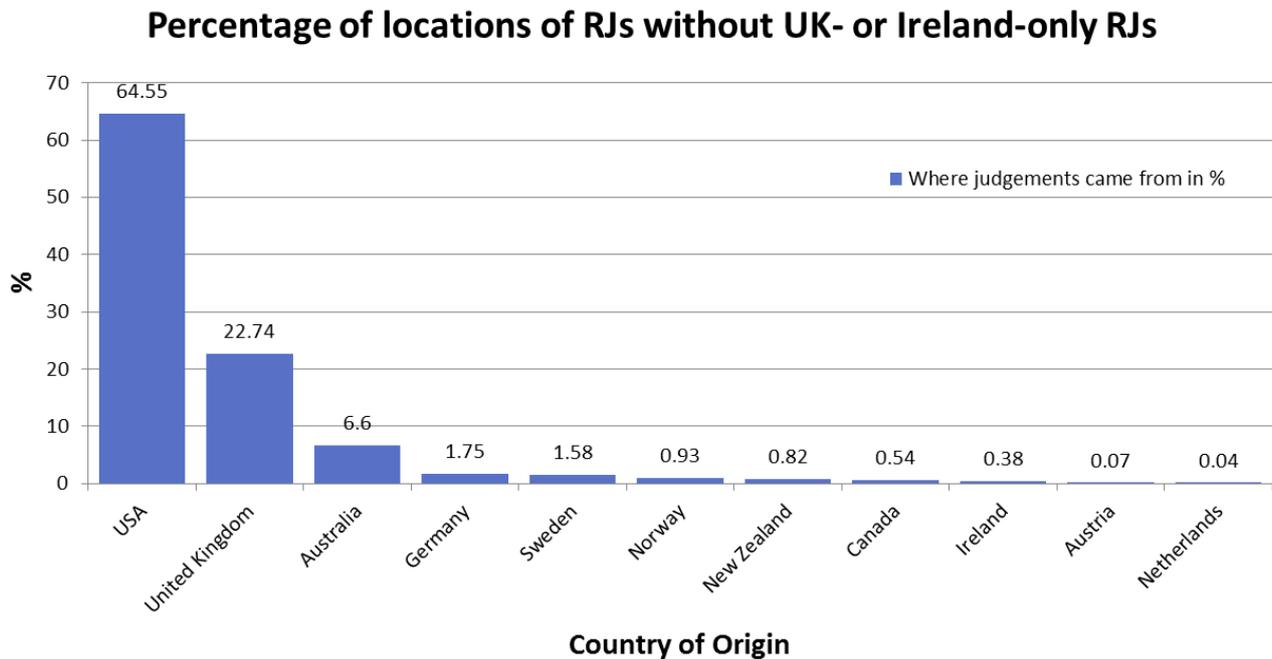
21	T	C	22	T	C	23	T	C	24	T	C	25	T	C
1035816	3	3	1373293	3	3	1086056	3	3	1650466	3	3	1988821	3	3
1198158	2	3	1308015	3	3	1086057	2	1	1834120	1	2	1982630	2	3
1913189	3	2	1158741	0	0	1086058	2	3	1650471	3	3	1988829	3	3
1934199	3	3	1349780	0	0	1091342	2	1	1916237	0	0	1164073	3	3
1458271	3	2	1001744	2	1	1096269	3	3	1650457	3	3	1295959	2	1
1186970	3	2	1001723	3	3	1144919	2	3	1700776	0	0	1164742	3	3
1177527	1	1	1622987	0	0	1144982	2	3	1650477	3	3	1156924	3	2
1049134	2	3	1366105	1	0	1178282	2	3	1548851	0	0	1162448	1	1
1836794	3	3	1086546	3	3	1180748	2	2	1652065	0	0	1516853	3	3
1974143	0	0	1984425	1	0	1196013	3	3	1535611	2	2	1975072	0	1
1001809	2	2	1984407	1	0	1226698	2	2	1650454	3	3	1515333	3	3
1458275	2	2	1001761	2	1	1237997	3	3	1246201	0	0	1164732	3	3
1597279	3	3	1158918	0	0	1242110	3	3	1650462	3	3	1091480	3	3
1715281	3	3	1623053	0	0	1242816	1	0	1834111	1	1	1183075	3	3
1035831	3	3	1306481	3	3	1310089	2	2	1197391	1	0	1988795	3	3
1458565	3	3	1288679	3	3	1313783	3	2	1336317	3	2	1702677	1	1
1067954	2	1	1033397	3	3	1475147	1	1	1194329	0	0	1183241	3	3
1068427	3	2	1001692	3	3	1476006	2	1	1828915	1	0	1987552	2	2
1173830	1	1	1366151	1	1	1530857	1	0	1073642	1	0	1515286	3	3
1068248	3	3	1740739	0	0	1551518	2	3	1650447	3	3	1975085	1	1
1470420	1	1	1100035	3	3	1759292	2	2				1219333	1	2
1001663	1	0	1261218	0	0	1759595	2	2				1129805	3	3
1001814	0	1	1510652	0	0	1773677	2	2				1717714	3	3
1467389	3	2	1553379	0	0	1810129	0	0						
1068470	3	2	1055424	3	2	1844774	3	2						
1229317	3	3	1477820	0	0	1845543	3	3						
1438262	3	3	1824992	0	0	1877203	0	0						
1670168	2	2				1935177	2	1						
						1951918	1	0						

I 1: Aggregated RJs from a trusted person and average RJs from CrowdFlower.

In the uppermost row, the number indicates the topic, T represents the trusted person's vote, and C the average rank calculated as median from the CrowdFlower judges.

Appendix J

Countries of Origin of the Relevance Judges



J 1 Where the judges originated from without the jobs only judged by judges from the UK or Ireland.

This figure does not encompass all the judgements. It only shows the distribution for judgements that could be submitted from judges all over the world.

In J 1, it can be seen where the judges originated from over the jobs that could be judged by all the countries. Judges from the United States of America are represented by over 64% of the judgements, meaning that $\frac{2}{3}$ of the RJs actually originated in the USA. The next larger portion belongs to the United Kingdom with almost 23%. The third rank goes to Australia with 6.6% of the judgements. The remaining 6% are divided by the other countries. Only Germany and Sweden contribute more than 1% of RJs to the evaluation. Therefore, CrowdFlower is not that popular in other than English-speaking countries. Thus, it makes sense to split the jobs in such that can only be judged by certain countries, although the judgement procedure may take longer.

Country	# of RJs	In %
USA	9977	50.04
United Kingdom	7817	39.21
Australia	1020	5.12
Germany	270	1.35
Sweden	244	1.22
Norway	238	1.19
New Zealand	144	0.72
Canada	127	0.64
Ireland	84	0.42
Austria	11	0.06
Netherlands	6	0.03
Sum	19938	100

J 2 Origins of the RJs for all jobs.

J 2 Summarises the origin countries for *all* the RJs. Naturally, there are now more judgements submitted by people from the United Kingdom. However, only few more people from Ireland contributed to the RJs, meaning that CrowdFlower is not very popular in Ireland. Almost 20000 RJs were submitted for the complete evaluation of the 612 images of the 25 topics.

Personal Declaration

I hereby declare that the submitted thesis is the result of my own, independent, work. All external sources are explicitly acknowledged in the thesis.

29th August 2013

Oliver F. M. Zihler