NTNU
Norwegian University of
Science and Technology

Ranking Mechanisms for Image Retrieval based on Coordinates, Perspective, and Area

Sindre Skjønsberg

Master of Science in Informatics

Submission date:	June 2010
Supervisor:	Trond Aalberg, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Problem Description

Searching for photos taken of places can largely be based on coordinates (or place names translated to coordinates). In sizable image collections containing many pictures from the same area, such queries can generate a large number of hits, requiring ranking mechanism for displaying the potentially most interesting pictures are listed first in the result set. Assuming a user indicates an area or a point on a map as a search criterion, it is possible to rank the images by comparing the image’s geographical coverage with this criterion.

In this assignment the student will map different types of pictures with regard to how they depict geographical areas, and investigate what information could be relevant to store for later use in ranking images based on locality queries. Furthermore, the student will propose ranking techniques and develop an experimental prototype to try out some of these in practice, providing a basis for the evaluation of these techniques.

Abstract

Image retrieval is becoming increasingly relevant as the size of image collections, and amount of image types grow. One of these types is aerial photography, unique in that it can be represented by its spatial content, and through this, be combined with digital maps. Finding good ways of describing this image type with regard to performing queries and ranking results is therefore an important task, and what this study is about.

Existing systems already combine maps and imagery, but does not take the spatial features found within each image into consideration. Instead, more traditional external metadata, e.g. file name, author, and date are used when performing retrieval operations on the objects involved.

A set of requirements for an image retrieval system on aerial photography using spatial features, were suggested. This described the image- and query types one can expect such a system to handle, and how the information found within these could be represented. A prototype was developed based on these requirements, evaluating the performance of single coordinate queries and a relevance calculation using the coverage, perspective, and areas of interest found in each picture.

The prototype evaluation shows that the different characteristics found in aerial photography makes it very difficult to represent and rank all these images in the same way. Especially images taken horizontally, i.e. where the horizon is showing, have other properties than images looking straight down on an area. The evaluation also shows problems related to manual registration of spatial features for images covering large areas, where inaccuracies introduced here can have a damaging effect on ranking.

Suggestions for future work with spatial image retrieval are mentioned, proposing alternatives to the spatial features used in the prototype, improvements for calculating relevance, as well as technologies that might help the feature extraction process.

Preface

This thesis is written as part of the author's Master of Science degree at the Department of Computer and Information Science at the Norwegian University of Science and Technology (NTNU), with a specialization in Information Management. The work done on the thesis was conducted in the autumn of 2009 and spring of 2010.

I would like to thank my supervisor Trond Aalberg for his guidance and feedback throughout the process, and my fellow students at the Mini/Sule computer lab; it wouldn't have been as fun without you.

Trondheim, May 31st, 2010

Sindre Skjønsberg

Acronyms

2-D	–	Two-dimensional
3-D	–	Three-dimensional
AI	–	Artificial Intelligence
AIS	–	ArcGIS Image Server
ADL	–	Alexandria Digital Library
AJAX	–	Asynchronous JavaScript and XML
API	–	Application Programming Interface
CBIR	–	Content-Based Image Retrieval
CGI	–	Computer-Generated Imagery
CSS	–	Cascading Style Sheets
DBMS	–	Database Management System
ESRI	–	Environmental Systems Research Institute
GDB	–	Geodatabase
GPS	–	Geographic Positioning System
GUI	–	Graphical User Interface
HCI	–	Human-Computer Interaction
HTML	–	Hypertext Markup Language
IR	–	Information Retrieval
IT	–	Information Technology
MIR	–	Multimedia Information Retrieval
OBIR	–	Object-Based Image Retrieval
RDBMS	–	Relational Database Management System
RQL	–	Resource Query Language
SQL	–	Structured Query Language
SRID	–	Spatial Reference Identifier
UI	–	User Interface
WMS	–	Web Map Service

XML	–	Extensible Markup Language

Contents

Table of Figures

1 Introduction

Image retrieval is becoming increasingly interesting as the amount of imagery available through the World Wide Web (WWW) and more traditional image collections have skyrocketed; old photography is being digitalized, a large number of satellites constantly provide us with images of the earth, and the introduction of Web 2.0, also known as the social web, allows people to share their own pictures with the world. This makes finding and ranking the most relevant images to a user's request an important and challenging task.

Images taken of physical locations are unique in the sense that they can have metadata describing the location using coordinates or text. Such textual descriptors are used to a large extent today, most commonly found as captions on web sites and similar. The coordinate metadata type, where a location is described using a single coordinate, is already widely used, as some cameras have a built-in Global Positioning System (GPS) chip, automatically tagging each image taken with the location of the camera. A more comprehensive approach is using coordinates to describe the image coverage, i.e. the area seen in the image, instead of the location of the photographer. In both cases the extra information available gives new opportunities in terms of storage, retrieving and ranking results by relevance.

It was but a few years ago that digital maps were something you paid for, and therefore mainly reserved for businesses and professionals. Thankfully this is no longer the case, and today several companies provide a wide variety of map services to the public. Being driven by commercial forces there is a constant need to find new ways to generate profit, resulting in new services and improvements of existing ones being launched continuously. This makes for an exciting situation, but also one where things are out-dated quickly.

1.1 Thesis Description

Assuming one or more collections containing aerial photography, i.e. images taken from the air looking down on the landscape, described by coverage coordinates, it should be possible to perform queries on these collections also using coordinates, either directly or translated from the name of a location or place. If a large number of pictures cover the query location, ranking mechanisms have to be in place to ensure the most interesting images are returned before the less interesting ones. This ranking should be based on how the different images depict an area, looking at features such as coverage, perspective and area.

The thesis consists of two parts, one theoretical and one practical. The theoretical part involves gaining an overview of the work already done with maps and image retrieval, and then looking specifically at which image- and query types exist when working on aerial photography, describing and illustrating these. With this in mind, an evaluation of image features that can be extracted to represent each picture, and later be used in querying the collection and ranking the result sets, is performed. Using these features, suggestions for querying and ranking these images is to be made, with an extra focus on the latter.

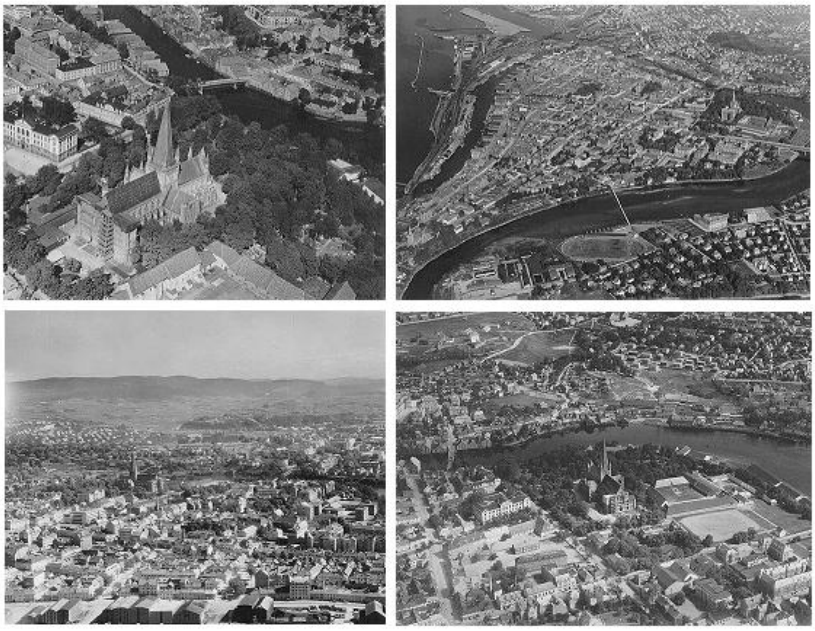


Figure 1-1: Aerial photography examples. Source: Universitetsbiblioteket i Trondheim.

The practical part consists of developing a prototype capable of handling spatial image retrieval of aerial photography. This prototype will implement some of the image retrieval components one might expect to find in such a system, with the purpose of using this functionality to evaluate how the theories presented earlier perform in real life. In addition to the prototype, a technical suggestion for the rest of the components required to make up a complete system is presented.

1.2 Objectives

Images in combination with maps are not a new phenomenon, and several systems in use today mix these two components for various purposes. Likewise, image retrieval has been around for some time, but these three together is a less common sight. Thus the purpose of the thesis is:

- Give an overview of some of the systems dealing with the concepts mentioned above.
- Describe ways to execute the various parts of the image retrieval process for aerial photography:
    - Identifying and expressing the different query types.
    - Categorize the different types of aerial photographs.
    - Describing the spatial content of these pictures through features.
    - Executing spatial queries on image collections.
    - Calculate image relevance based on spatial content features.
- Implement a prototype to test the described retrieval process on real imagery as well as briefly mentioning the remaining system components not included in the prototype.
- Evaluate the prototype’s performance, draw conclusions from this and suggest future research areas.

1.3 Document Structure

This thesis is structured as follows:

Chapter 2: Theory presents the theory required to understand how current systems function and the reasoning used when creating a system suggestion in later chapters.

Chapter 3: State of the Art describes a handful of already existing systems dealing with images, maps and information retrieval.

Chapter 4: Requirements is a list of requirements for an imagined image retrieval system able to handle spatial features. Details on image- and query types found in this kind of software are among the things described here.

Chapter 5: Implementation presents a technical solution fulfilling the requirements described in chapter 4. The prototype part of the thesis is also described here, in much greater detail than the other parts of the system.

Chapter 6: Evaluation presents the results gathered from the prototype using descriptive text and examples to illustrate and explain these.

Chapter 7: Conclusion summarizes and draws conclusions from the evaluation. Suggestions for future research areas, both in general and directly related to the prototype are also listed here.

2 Theory

This chapter contains the theoretical background required to understand the context of the thesis, explaining general information retrieval concepts as well some directly related to geography-systems.

2.1 Multimedia Information Retrieval

Information retrieval is defined as *the part of computer science which studies the retrieval of information (not data) from a collection of written documents. The retrieved documents aim at satisfying a user information need usually expressed in natural language* (Baeza-Yates & Ribeiro-Neto, 1999).By extending this to also include non-written information objects, and allowing for more flexible queries such a query by sound, shape, etc., we have created a definition of what multimedia information retrieval is.

The difference between data- and information retrieval might seem insignificant, but is vital to know when to use one or the other. Retrieval of data is absolute, and the result set returned from performing a query only contain items that match all parts of the query. Those objects in the collection that are different, if only on a single value, from the query, are considered irrelevant. In information retrieval, this either/or situation does not exist. Instead the goal is finding the documents most relevant to the user, regardless of whether these documents match all, parts, or even none of the query parameters.

Typical examples of data retrieval are database query languages, such as Structured Query Language (SQL) and Language Integrated Query (LINQ), languages which can be found in a wide variety of computer systems. Information retrieval systems are not as widespread, especially those handling multimedia data. The most used information retrieval systems found today are online search engines such as Google and Bing, although these are based mainly on text searches, with limited functionality for more complex data types. There are a number of research projects working on dedicated multimedia information retrieval systems, but so far none of these has made their commercial breakthrough.

2.2 Query

For an information retrieval system to be useful there has to be some way for the users to access the information made available through the system. The two main ways of achieving this is by allowing the user to browse, query, or do a combination of the two on the collections containing information. Browsing, as the name implies, describes a scenario where the user looks through the contents of the system casually, without a clearly defined goal. Although browsing is an interesting method, queries is what this chapter (and thesis for that matter) is focused on.

A query is *the expression of the user information need in the input language provided by the information system* (Baeza-Yates & Ribeiro-Neto, 1999). This lack of restraints in query design makes it possible to create query types tailored to the information objects within the system as well as queries which take the level of the user into consideration. Examples of the latter can be queries expressed as regular expressions for advanced users, but also with the option of using free text for those not versed in the intricacies of regex (Goyvaerts, 2010). Other types can include, but is not limited to, Boolean queries, similarity queries, and queries based on example, all of which exist in systems today. There is in reality no limit as to how simple or how complex a query can be, although some types are more common than others.

Queries are entered into the system by means of a user interface, graphical, pure textual, or a hybrid, depending on the system. Commercial systems today often rely on a combination, where the user inputs a text-based query and further specifies the query using a graphical user interface (GUI), typically known as "advanced options". In spatial systems a map interface is commonly used for queries, where locations are marked on the map by the user, and the underlying logic handles the conversion to coordinates, hiding this from the user.

2.3 Features

A feature in information retrieval can be described as *information extracted from an object and used during query processing* (Baeza-Yates & Ribeiro-Neto, 1999). This definition puts no limitations as to what kind of information is considered a feature, nor any restrictions to how this information is gathered from the object. This open approach to features is important, allowing us to cover both features that exist today, and those that may appear in the future, as new query types are introduced and needs change.

Features can be extracted from the object's properties, both internal and external, and the content of the objects. Typical examples of external properties are file size, file type, and information which can be found in the object's metadata, such as author, publisher and date of creation. These features are for the most part easily accessible, and require little effort to extract and prepare for query processing. Internal properties are those found in the content of the object, and we normally separate between two types; high- and low-level (Lu, 1999). High-level features are a description of abstract concepts such as topics, people, places and similar which can be found in the objects. In non-textual media identifying these is a major challenge, as computers are unable to interpret and understand such abstractions on the same level as humans (Datta, Li, & Wang, 2005). This situation is reversed when looking at low-level features. Low-level features covers the basic content building blocks found in objects, such as colour and texture for images, zero-crossing rate in audio and pixel change over time in video. Such features can be derived from numbers alone, something computers excel at.

The fact that computers are good at handling low-level features, and relatively poor on high-level ones, and vice-versa for humans, is a major challenge when designing information retrieval systems. People have a hard time expressing information needs (queries) using low-level attributes, and computers struggle with high level concepts, creating a communication problem between the system and the user, also known as the semantic gap. A solution to this is allowing for queries by example, meaning the user can input an object and the system will find objects similar to this. However, using this type of query is not possible for all systems, and requires the user to already have an object that belongs to the desired result set.

As mentioned there is virtually no limit as to what form a feature has, and as a result of this they can also be represented in a multitude of ways. The most used representations, especially for low-level features, are numerical values contained within a data structure, e.g. an array or a matrix, and text. Both of these are well suited for digital representation, and therefore do not pose a significant challenge for the system handling them. However, complex and less common features like sound or images will more than likely require specially designed solutions.

2.4 Feature Extraction

Feature extraction means, as the name implies, obtaining all the features belonging to an information object. The methods used for this depends on the type of information needs to be extracted and any special conditions, e.g. limited system resources or speed requirements, that may exist. More often than not, several approaches have to be used, as not all features are located in the same place, nor are of the same type. For resource reasons, the feature extraction should be an automatic process, requiring no manual labour.

2.5 Feature Vectors

A mathematical way of representing an information object is by using one or more feature vectors; n-dimensional vectors containing numerical values, each value representing one of the object's features. These feature vectors will then be used when organizing, retrieving and indexing the objects they belong to. Using an object's feature vector instead of the object itself for these operations reduces the resource requirements, due to the fact that the object vector is smaller and less complex than the object itself. By only using numerical values the need for interpretation is removed, as opposed to what would be the case if more abstract data types were used. An additional advantage by using numbers is that it makes including more parameters, such as weights, is trivial.

Feature vectors can be extracted simultaneously with the object, when this it is initially added to the system, when first used, for every query entered into the system, or a combination of these. Which of these methods are chosen is largely dependent on the type of system and the type of objects involved. For instance would a system containing documents that are edited frequently require the feature vectors, or at least part of them, to be updated often to ensure good retrieval performance. In general it is still preferable to have up to date vectors available at all time, to improve system responsiveness.

Feature vectors can also be created for queries, even if the contents of the query are different from the objects it is performed on. To circumvent this problem, a complete feature vector can be created with nil-values replacing those features that are not present in the query. A similarity search can then be executed in the same manner as if we were looking at two objects with the same features.

2.6 Ranking

In almost every case where multiple documents are retrieved in a result set, they will be presented in a ranked manner, with the most relevant document on top. A wide range of techniques and criteria can be used for deciding the ranking, and below a few of the different options are described.

Criteria:

- **Content:** Content in this case describes the internal features which can be extracted from the document. These will typically be used to determine similarity between the query and each of the documents, and is especially useful when performing queries by example, both for multimedia and text.
- **Metadata:** The external features, or metadata, attached to the documents are used for deciding the relevance to the user query. This data is often easily accessible (no calculation required) and can provide unique information such as author, origin and date of creation. Metadata criteria can be used on their own or in combination with content.

Techniques:

- **Boolean**: All documents that match the query exactly are considered relevant while all others are discarded. This is a very simplistic way of ranking, creating only two categories, with no way of separating the documents within each category. This type of ranking is dominant in data retrieval.
- **Single criteria:** A single criteria is used for judging relevance and ranking the different documents according to this relevance. By measuring the distance between a value in the query and the corresponding value in each of the documents, a single measurement of similarity is created. This approach includes documents which would have been considered irrelevant using Boolean ranking, but will in many cases not include enough parameters to properly separate the objects in large result sets.
- **Multiple criteria:** As the name suggests, multiple criteria is an expansion of single criteria. Including more criteria makes it easier to separate different documents, but naturally also increases the complexity. Not only does each of the criteria have to be individually calculated, but they have to be weighted and combined before a ranking can be created.
- **Combination:** When dealing with large collections the different techniques can be combined, supplementing each other. Typically, Boolean ranking will be used initially to weed out all irrelevant documents before one or more criteria are used to rank the relevant ones.

2.7 Evaluation

Evaluating an IR system means determining how the system fulfils a set of given requirements. All aspects of a system can be assessed, grading everything from responsiveness to query results. There are no standards as to what methods are used for the evaluation, and each researcher can freely choose how his or hers system is evaluated. For IR systems the relevance of retrieved documents is always important, and this chapter describes considerations and methods for determining this relevance.

For an evaluation to take place a set of evaluation criteria have to be in place, i.e. some way to tell how relevant a retrieved object is. A number of factors play a part when deciding how to determine these criteria:

- **Complexity:** Closely related to the object type, complexity in this context describes the level of features to be evaluated. Low-level features such as amount of a certain colour, occurrences of a specific word, and wave spectrum require a different approach than abstract features such as objects, topics and speaker.
- **Collection size:** The size of the collection (number of objects) determines if each object can be described automatically, or if a manual method has to be used. For small collections, manually judging and annotating each object is a viable approach that can produce accurate results, but on collections containing thousands of objects this approach is far too time consuming.
- **Time of classification:** The choice between describing all objects beforehand or doing this only for objects which are retrieved. The former might result in describing objects that are not involved in any queries, but is a one-time job, while the latter introduces extra work each time objects are retrieved for the first time, and is subject to changes that might occur over time.

Regardless of which criteria are chosen, a set of guidelines and rules have to be created to ensure all objects are described in the same way. This is especially important when objects are annotated manually.

There are countless IR systems in use today, working with vastly different objects and appurtenant evaluation criteria, with an equal amount of different target user groups. These factors form the basis of which evaluation method is chosen, as the object type often restricts how each individual object can be measured according to relevance, and an evaluation can only be considered valid when performed with the system’s target user group.

A common statistical approach to IR evaluation is using precision and recall. Precision is the fraction of the retrieved documents which are relevant to the user query while recall is the fraction of known relevant documents retrieved from the collection. These two can be used individually, but will more often be combined to create a more meaningful evaluation. One way this can be done is by combining the two values into a curve, showing how precision evolves as recall increases. A variation of this is mean average precision, where recall and precision are calculated and combined each time a relevant retrieved document is encountered, up to a cut-off point.

Recall and precision are not, despite their popularity, suited for all evaluation tasks. To determine maximum recall for a query requires complete knowledge of all documents in the collection, something which is unlikely when working with large collections. Another limitation is that a linear ordering of results is required to make the most of precision and recall, and there is no support for interactive queries. A number of alternatives have therefore been developed, some of these variations upon precision and recall (Harmonic Mean, E-Measure), while others use different measurements (coverage, novelty, etc.) instead.

2.8 Geographic Information Systems

A geographic information system (GIS) is a computer system which uses spatial data in one way or another to provide a service, most commonly providing information. NASA defines such a system as "GIS is *an integrated system of computer hardware, software, and trained personnel linking topographic, demographic, utility, facility, image and other resource data that is geographically referenced*" (Dempsey, 2008). This definition covers all aspects of GIS, and emphasizes the fact that there is more to a GIS than just software providing digitalized maps to their users.

From this description we can identify and describe the four components that make up a GIS:

- **Data:** Without data there would be no content in the system, making it useless. At least part of these needs to be spatial, meaning it is geographically references. Normally the spatial data will then be combined with other data sources, making adding several layers of information on the same map possible.
- **Software:** As in all computer systems, a GIS requires software to perform its given task. This software is required for creating, storing and managing data used by the system.
- **Hardware:** Hardware includes all the equipment needed, not only to run and host the software and data, but also that which gathers the initial data.
- **People:** Without people with the knowledge on how to use the GIS software, adding data or analysing results, the benefit gained for such systems would be minimal.

We can separate between two typical usage scenarios; digital mapping and information gathering, although the same system can be used for both. The former involves using the GIS exclusively as a source for geographic information, for example systems used in navigation, locating places and similar, whilst the latter incorporates data from other sources to a much larger degree, creating multi-layered information sets. These layers can then be combined and analysed, returning results that can be used for decision making or other research.

There is virtually no limit to which domains a GIS can be applied. For consumers, the most common is as a pure geographical tool, for example in car navigation and address location. Another area GIS has seen much use is medicine, where the combination of health related information and maps is used to gain a greater understanding of not only where problems exist, but also why they do. An example of this is the Biomedical Research Network[1], which provide researchers with a framework where they can share and combine each other's data. Another good example on the variety of fields GIS can be used is the Coastal Atlas[2], a project which collects a variety of map-related information made available to help local businesses make decisions.

[1] http://www.birncommunity.org/

[2] http://www.coastalatlas.net/

Figure 2-1: GIS in car navigation. Source: http://electronicpro.org/

2.9 Geotagging

Geotagging is adding geographic data to a picture or other type of media. There is no limitation on what sort of data could be added, nor in what way it is be represented. In most cases the geotag is the longitude and latitude coordinates where the picture was taken, or a textual description of the location or area. Other data which can be added are altitude, bearing, and similar. The type of geographic data also dictates how it is added to the picture, coordinates can be added automatically using the Global Positioning System, or manually, either by hand or assisted by a computer system.

Geotagging is used to extract more information from a given picture than what can be gathered from just looking at it, information which later could be used to aid the retrieval process. By placing a photograph in a geographic setting it gets another dimension. This new dimension also gives users another way to organize photos, making it possible to keep a record of which places they have visited, and when. Such tagging also contributes to the digital mapping of the world, something which can be a great help in areas such as virtual tourism.

2.10 Georeferencing

Georeferencing is an umbrella term for the process of transforming non-geographic information into geographic information, meaning information that has a valid geographic reference (Goldberg, 2008). This allows non-geographic information to become viable in spatial analysis. An example of this is GPS, which can determine coordinates for a location on earth based on a system of satellites and calibrated ground stations.

Another method for georeferencing is geocoding. Geocoding describes *the act of transforming aspatial locationally descriptive text into a valid spatial representation using a predefined process* (Goldberg, 2008). This definition puts no restrictions on the type of text used as input in the geocoding process, and it opens up for a number of different output formats, not just latitude- and longitude coordinates. That being said, the most common commercial application of geocoding today is using postal addresses as input and getting the map location of this address as a result. This feature is possible in most internet map services such as Bing Maps described in chapter 3.3.

Reverse geocoding is, unsurprisingly, going from a spatial representation to a non-spatial descriptive text. Such services have up until recently not been available to the public due to the amount of data and calculations required, but it now starting to appear, for instance in Google Maps[3].

Geocoding and reverse geocoding can suffer from a lack of accuracy. This can be a result of errors in the calculation of the spatial representation, such as incorrect latitude- and longitude coordinates, or errors generated from the ambiguity inherent in addresses. The latter is caused by addresses and locations not being unique and the fact that they can be expressed differently depending on language further complicates the issue. A common solution to this issue is asking the user to provide more information to the lookup, such as postal address and country, to help identify what he or she is looking for.

Another concern is that of privacy, where the georeferenced data is combined with other sources to reveal otherwise hidden information. By reverse engineering spatial information presented as part of surveys and similar, it is possible to identify the location of the participants down to a residential level. This information can then be used to identify the person(s) who were involved in the survey. An experiment performed using geospatial data on mortality locations after Hurricane Katrina shows how confidential information can be extracted from point level information (Curtis, Mills, & Leitner, 2006).

[3] http://maps.google.com/

3 State of the Art

This chapter gives an introduction to what research and existing systems can be found in the image retrieval domain and GIS in general. Some of these systems are closed, meaning their internal workings are not publicly known and therefore impossible to describe in detail, but their relevance to the thesis objectives have made them worth including anyway.

3.1 Content Based Image Retrieval

There are two image retrieval frameworks used today; text- and content-based image retrieval (CBIR) (Wang, Zhang, & Zhang, 2008) with the former being the more traditional of the two. Here images are annotated by humans and then searched based on these annotations. This approach, although able to describe the contents, especially the abstract concepts, of images to a high degree, requires too much human effort to be viable for large collections of images, such as the WWW. As the size of image collections has increased, researchers have shifted their focus to CBIR-systems (Datta, Li, & Wang, 2005), in addition to attempts of bridging the two approaches, also known as object-based image retrieval (OBIR) (Zheng & Gao, 2008).

The biggest challenge for CBIR researchers today is the distance from low-level features to high-level concepts found in images, a problem known as the semantic gap (Wang, Zhang, & Zhang, 2008). This problem also exists in other forms of information retrieval, such as text and audio, but here the gap can be made smaller by using techniques such as ontologies (Ganguly, Rabhi, & Ray, 2002) and relevance feedback. Similar efforts have been attempted for bridging the gap in image retrieval, but with limited success (Hare, Lewis, Enser, & Sandom, 2006). The reason for this is that there are several layers between the raw media and the full semantic description, e.g. a CBIR can identify that the image contains buildings and people, but not that the picture is taken of Times Square in New York City. A complete overview of the hierarchical levels found in image retrieval can be seen in Figure 3-1.

Although much work remains before the complete semantic meaning in an image can be extracted automatically, researchers both in information retrieval and artificial intelligence are working on closing the gap both from below, going from descriptors to objects, and the top, using ontologies. When these approaches are combined sometime in the future, we can hopefully begin to see the outline of a semantic bridge forming.

The semantic gap is not as big an issue in image retrieval based on text, where images ideally are annotated by domain experts able to identify objects as well as describe abstract concepts. A potential problem here is the difference between how the experts express themselves compared to the intended users, but this is a text retrieval issue, and will not be investigated further. Using a text description instead of internal content features can also provide a challenge when ranking the relevant images, as the text annotation will usually be short, providing very little material to separate similar images and create a ranking. Metadata can be used as an additional component in such a case, but only if such data exists and is of sufficient quality, neither of which are a certainty.

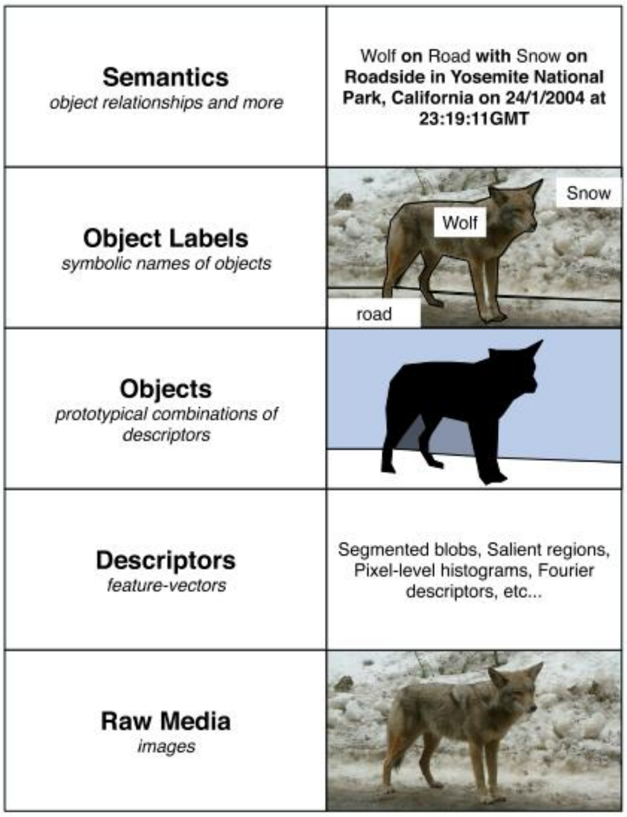

Semantics
object relationships and more
Wolf on Road with Snow on Roadside in Yosemite National Park, California on 24/1/2004 at 23:19:11GMT
Object Labels
symbolic names of objects
Snow
Wolf
road
Objects
prototypical combinations of descriptors
Descriptors
feature-vectors
Segmented blobs, Salient regions, Pixel-level histograms, Fourier descriptors, etc...
Raw Media
images

Figure 3-1: Levels between raw media and semantics (Hare, Lewis, Enser, & Sandom, 2006).

The lack of metadata in combination with the semantic gap leaves much to be desired when managing images, especially for ranking. Being able to extract and use the high-level content of objects as a basis for ranking is today only viable on textual documents and small image- and multimedia collections. The latter case, where one or more content features are used as ranking criteria could be applied to larger collections, but this method depends on the documents being described and indexed before the query is performed, or the delay between query and receiving the results will be too large for actual use. This amount of pre-processing is not possible when working with huge image collections such as the World Wide Web, where searches are limited to using metadata in combination with a few simple content features. For example is the Google Image Search[4] only able to separate between coloured- and black and white images. In comparison there are is a large number of external metadata options for searching; size, usage rights and relation to surrounding text to name a few.

Multimedia information retrieval (MIR) has seen a shift from a technology-centered to a human-centered view over the last decade (Lew, Sebe, Djeraba, & Jain, 2006). Systems designed prior to this had very little, if any, focus on user friendliness and were in reality only useable by other scientists, not by the intended users of the systems. The main goal of this human-centered computing is satisfying the user's need for information when querying or browsing the media collection. This means considering user behaviour and emotion and creating systems that adapt to the user, instead of the other way around. A couple ways of doing this is by using relevance feedback on queries and

[4] http://images.google.no/advanced_image_search?hl=en

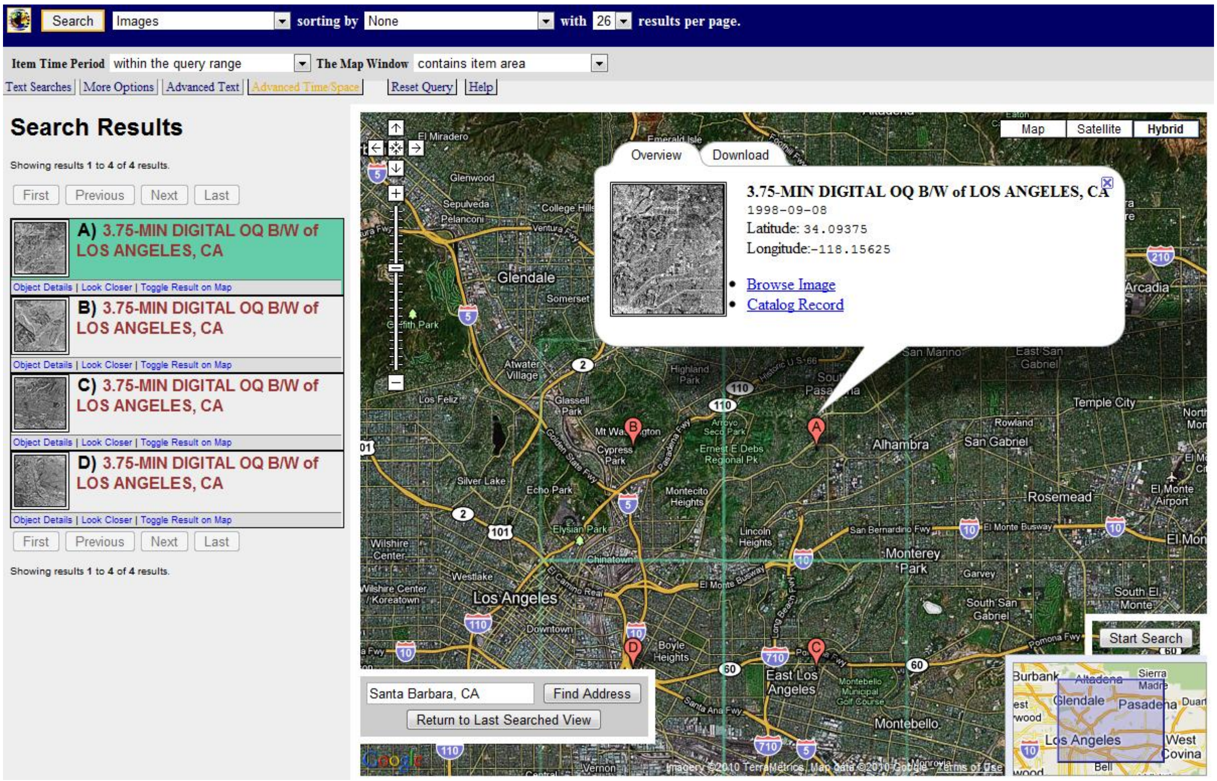
design systems that utilize artificial intelligence to gain an understanding of user behaviour, adjusting the system thereafter.

How MIR systems are being evaluated has also been affected by this shift in research, a result of systems doing well in laboratories did but not showing any significant commercial success (Fluhr, Moëllic, & Hede, 2006). For example were a number of campaigns launched in France, all based on actual user needs, where the success was judged based on usage, not technical merit. Among the campaigns launched was recognition of transformed images, combined text- and image search, as well as text detection in images.

3.2 Georeferenced Resources

Several digital library systems also include geospatial and georeferenced information as an addition to their traditional digital documents. In this chapter a few of these libraries will be described, looking especially at how they handle user queries and ranking of results.

One of the more famous collections of georeferenced materials is held by the Alexandria Digital Library (ADL), a project headquartered at the University of California (University of California, 2004). ADL contains several geographically referenced collections, with more in the process of being added. Some of these collections span globally, but most are focused on a smaller area such as regions or single countries. Regardless of their origin, they can all be accessed through the same web portal, and queries can be executed on some or all collections depending on the user's choice.



Search: Images sorting by None with 26 results per page.

Item Time Period: within the query range The Map Window: contains item area

Search Results

Showing results 1 to 4 of 4 results.

A) 3.75-MIN DIGITAL OQ B/W of LOS ANGELES, CA

B) 3.75-MIN DIGITAL OQ B/W of LOS ANGELES, CA

C) 3.75-MIN DIGITAL OQ B/W of LOS ANGELES, CA

D) 3.75-MIN DIGITAL OQ B/W of LOS ANGELES, CA

Showing results 1 to 4 of 4 results.

3.75-MIN DIGITAL OQ B/W of LOS ANGELES, CA
1998-09-08
Latitude: 34.09375
Longitude:-118.15625

- Browse Image
- Catalog Record

Santa Barbara, CA

Figure 3-2: ADL Globetrotter client, showing the results of an image search. Source: Screen capture.

Queries in ADL are performed using the Globetrotter[5] web client which combines the ADL search engine and Google Maps, the latter providing maps, navigation options and query result visualization. Searches are expressed as text, either single words or phrases, with additional parameters like (but not limited to) time period, spatial limitations (overlaps, outside, etc.), and what collection the query should be performed on. All queries are also limited to the area the user is currently watching, e.g. executing a query when being zoomed in on Trondheim will only return results georeferenced to Trondheim. ADL also feats a simple metadata ranking mechanism, where results can be sorted based on similarity to the query text or by date. The client does not support any kind of relevance calculating based on image content apart from the area restriction already mentioned.

G-Portal is another digital library project focusing on georeferenced information, but unlike ADL, which uses a distributed architecture with several separate collections, G-Portal has a centralized storage system for all metadata (Liu, Lim, Ng, & Goh, 2003). This metadata is represented in Extensible Markup Language (XML), a language well suited for storing semi-structured data (something most metadata certainly is). G-Portal uses a number of XML schemas to describe the resources contained in the collection, an approach that makes it possible to properly annotate different types of documents while maintaining some basic elements required for identification and classification.

Spatial data, if it exists, is stored as a XML location element. This element can be anything from a simple point to a set of polygons, depending on how much information is available. The location element comes with a set of spatial predicates which can be used when performing spatial queries, for example cover() (true if the resource has a location containing a given geometry object) and disjoint() (true if the location of the resource is disjoint with a given geometry object).These predicates could also, if their researchers wanted, be used to set the relevance of individual documents in a result set, forming a simple ranking technique on spatial content, but currently no work has been to add result ranking to the system.

For performing queries in G-Portal, a modification of XQuery, Resource Query Language (RQL), is used. RQL follows the XQuery syntax closely, but has increased support for spatial attributes, allowing both non-spatial queries, spatial queries and a mix between the two. The performance and success of these queries is one of the topics the G-Portal researchers are still working on, and no definite results have been published so far. But judging by previous research on adding relevance, and henceforth producing a ranking of XML documents, much work remains before a solution that both performs and scales well is found.

Not all georeferenced collections have any sort of quality control, and as such cannot be considered digital libraries. GeoNames Reverse Geocoding Services[6] is such as collection, where latitude- and longitude coordinates can be linked to different types of information such as postal codes, oceans, cities, as well as purely descriptive labels (ski resort, avalanche danger, etc.). This information is added in a collaborative fashion by users and GeoNames themselves, using a wiki interface.

[5] http://clients.alexandria.ucsb.edu/globetrotter/

[6] http://www.geonames.org/export/reverse-geocoding.html

The GeoNames database is free to use, either as a standalone download or through various web services. In addition to the data itself, a number of tools are available for creating advanced queries as well as customized ranking of results. These tools are mainly for text-based queries, but in addition to traditional IR, some spatial featured such as distance and area can be included in queries / ranking. As the database is being provided as a service, it is possible for users to add custom functionality, for instance by taking advantage of the built-in classification system, adding their own program features on top. As such there is no default ranking, but with the spatial tools it is possible to creating ranking mechanisms using both content and metadata.

3.3 Internet Map Services

The World Wide Web has redefined how maps are being used. From being something restricted to physical media, maps are now digital entities, which can be transmitted and delivered to the user almost instantly, only requiring an internet connection (Peterson, 2005). This digital revolution has not only increased the availability, but also changed what kind of information we can display on a map. No longer limited to static topographical information found in traditional atlases, internet map services can provide static, interactive and animated maps, often kept up-to-date with frequent updates (e.g. several times a day for weather maps).

Interactive maps describe a map service where the user not only can browse the data as he or she wishes to, but also can control which layers of information is shown, and even contribute with their own information layers. The most basic layer selection is that between satellite images, topographical, or street maps (see Figure 3-3), but in theory there is no limit to what information can be displayed. Maybe more interestingly is the possibility for the users to add their own information to a map, such as geographically referenced pictures, videos and text, something which is a typical Web 2.0 phenomenon.

Even though the map services contain a number of different data types, queries can only be expressed as text, with a list containing the results being presented, as well as their placement on the map (if applicable). The exact formula for deciding the relevance for each document is not known, these all being closed systems, but based on observation some guesses can be made; Most internet map services are part of a search engine, and as such share algorithms. A spatial criterion is added to all objects, if such a thing exists, labelling results in, or close to, the area described in the query as more relevant than results located far away. This spatial relevance is then combined with the other algorithms in the search engine, creating the final ranking which is displayed to the user. It is also possible to limit results to certain categories such as locations, businesses and real-estate, something which not only narrows things down, but has an effect on the other criteria, spatial ones included, although the exact details of this are unknown.

One of the things that separate these maps services from georeferenced libraries is that the map services only have one set of satellite imagery, i.e. a location can only be seen in one photograph. Outdated imagery is most likely placed in storage, but is not made available to users, something which simplifies the retrieval process considerably, never having to choose between several images when a location query is performed.

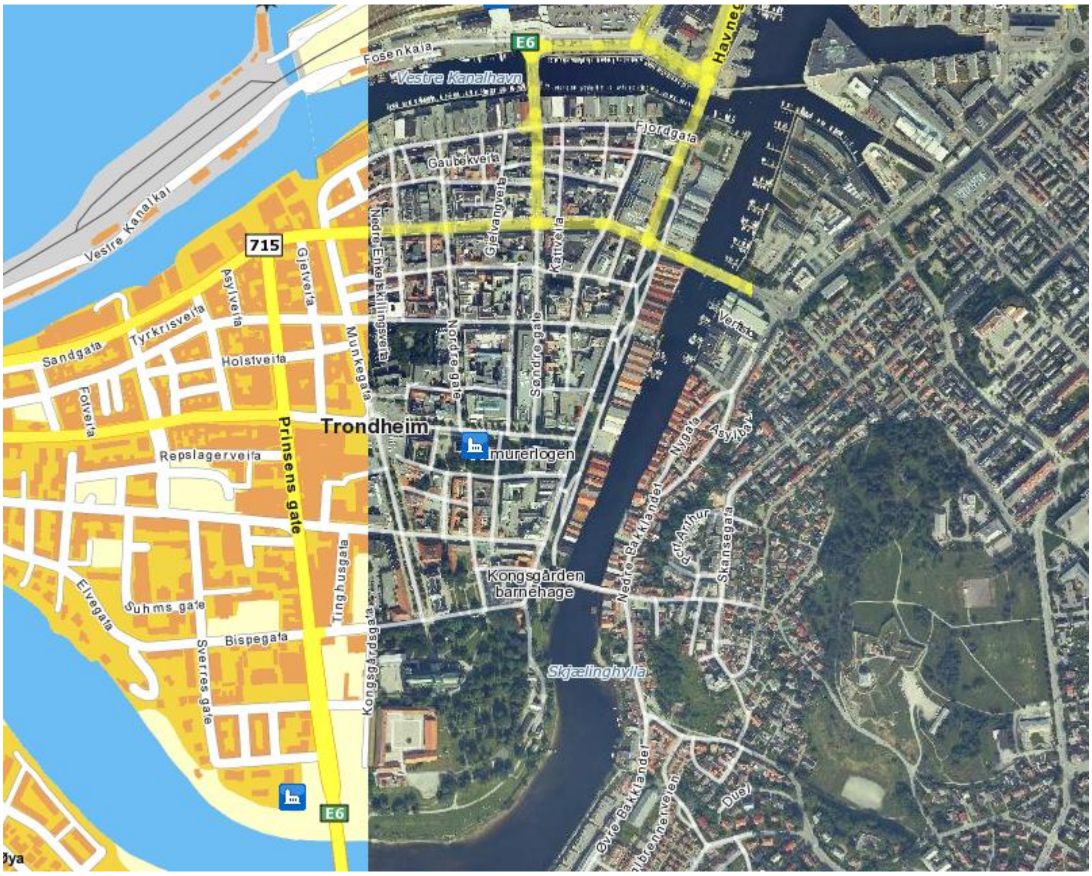


Figure 3-3: The three different map views available in the Gule Sider map service[7], from the left: map, hybrid and satellite photo. Source: Screen capture.

There are several major companies providing map services today, Google[8], Microsoft[9], and Yahoo[10] being perhaps the most well-known. Although there are differences between these, especially in terms of look, feel, and layout, they share most characteristics. With this in mind, the map service by Microsoft, Bing Maps, is used as basis for the description below, but this description should also be relevant for the other providers, with minor alternations.

Bing Maps use satellite images as the base for their service, and as such have maps covering the entire world, barring areas or locations considered too sensitive by the state or government to be shown. The quality of the satellite images are of varying quality, where densely populated areas usually are depicted by more detailed images than those with little or no population. In addition to satellite images, terrain and road maps are also available. Bing Maps also combines satellite- and road maps, creating a hybrid view, as seen in Figure 3-4. This also demonstrates the strong link between latitude- and longitude coordinates and real world locations, such as streets, buildings and landmarks, found in the system. As Bing Maps retains all this information, it is able to convert from coordinates and locations and vice-versa, extending query possibilities.

In addition to the standard map view, Bing Maps feats a great number of applications to further increase the viewing experience (Microsoft, 2010). These vary from those updated in real time, such

[7] http://kart.gulesider.no/
[8] http://maps.google.com/
[9] http://www.bing.com/maps
[10] http://maps.yahoo.com/

as traffic- and weather information, to tourist aids such as hotel- and restaurant finders within selected parts of the map. In addition to displaying businesses and similar, Bing Maps also gathers information on local events such as concerts and exhibitions, displaying both location, date and other relevant facts.

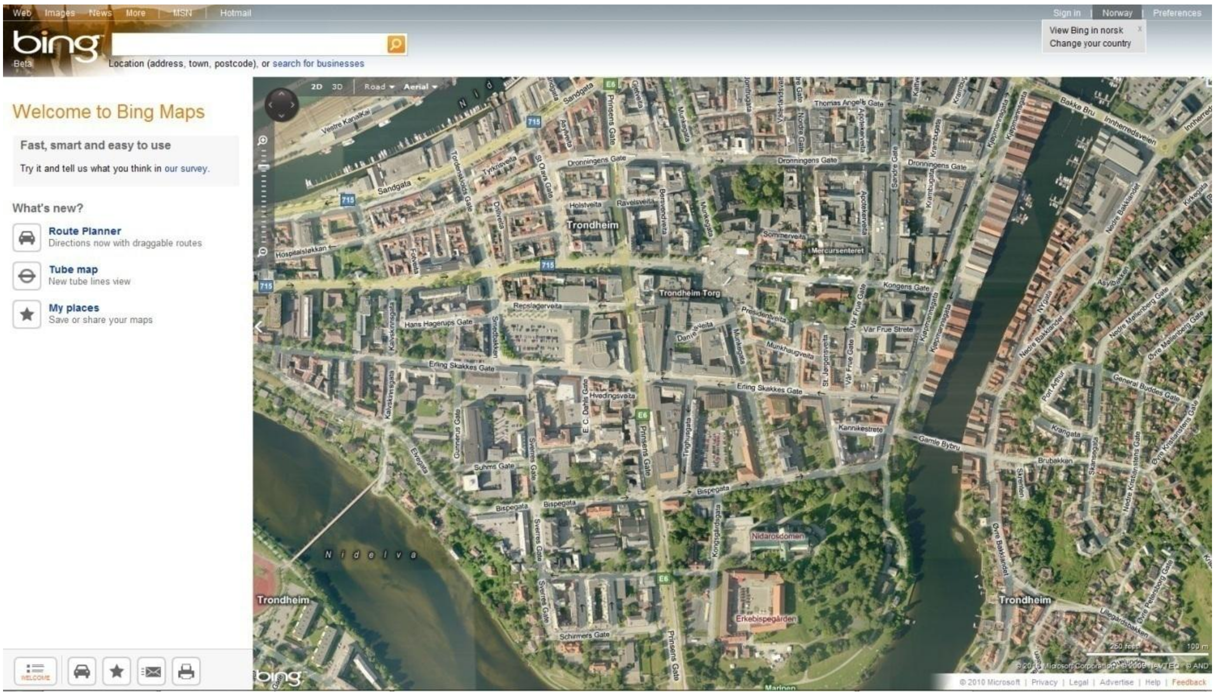


Figure 3-4: Bing Maps showing a hybrid view of the Trondheim city centre. Source: screen capture.

Another interesting feature that has emerged the last couple of years is street view, known as streetside in Bing. Street view is basically pictures taken by a car driving along streets (limited to major cities for the time being), combining these photos into a 360 degree close-up view of the car's surroundings. The location of the car is registered for each set of pictures taken, mapping them to the rest of the map service, allowing the user the images on street level. Microsoft is currently developing an addition to street maps where photographs can be added on top of the street view layer (Microsoft, 2010). Although this only as a technical preview, the idea is that users can add their own geo-tagged pictures, and the system will stretch and rotate the photo to fit the streetside view, as seen in Figure 3-5. This technology combines images, pattern matching, and maps in a fairly unique way.

Arguable, one of the main reasons for the success of internet map services is that they provide their services to other websites, allowing these sites to not only use the maps, but also add their own content on top if they so choose. A simple example of this is adding map markers on all the cities you have visited on your personal home page, each marker containing a small summary of your visit, with a link to pictures. This is made possible by having Application Programming Interfaces (APIs) for each of the map services in the most commonly used web programming languages, allowing developers to expand and adapt the service to fit their own specific needs.

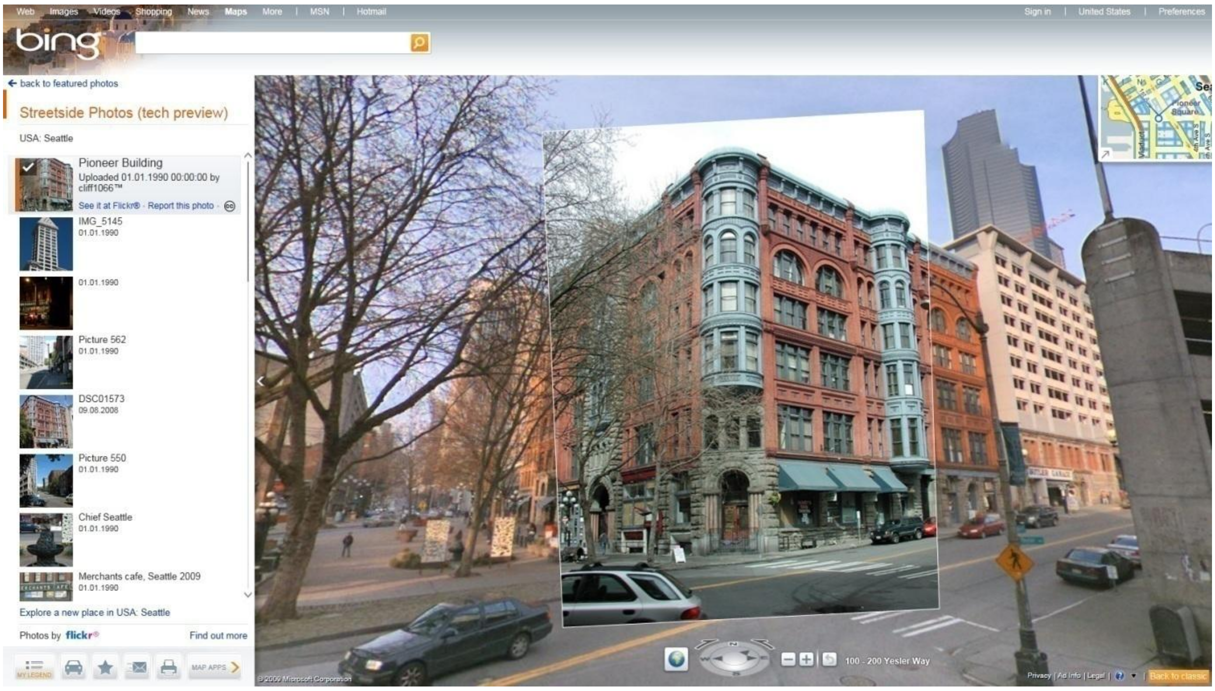


Figure 3-5: Bing streetside with a photograph added on top. Source: Screen capture.

3.4 ArcGIS

There are several enterprise systems for managing and using geographic knowledge; ArcGIS, Quantum GIS and GRASS to name a few. Although different in terms of functionality and technical implementation, they all illustrate what commercially available geographic information systems can and cannot do. Here the ArcGIS system developed and maintained by the Environmental Systems Research Institute, ESRI, will be described more in detail, as this system is used extensively and as a result is heavily documented with both white-papers and books.

ArcGIS is a set of software products, all developed by ESRI, which together form a complete package meant to cover all the GIS needs a business or organization may have (ESRI, 2010). Their solution is divided into three separate main components; Desktop, Server and Web. Respectively, these represent authoring, publishing and using geographic data, and with several smaller modules within each of these components this allows for some customization when delivering a solution to the customer. In addition to the platform itself, ESRI can also provide data, e.g. digital maps, from external providers as part of their product.

ArcGIS is geared towards business and organizations, and as such focuses on solving complex problems which may require lots of data as well as heavy computation. In this way it covers a different market segment than the internet map services, which mainly address individuals. The features included in ArcGIS falls into three toolsets; visualisation, data management, and spatial analysis. These tools can be combined with each other, as well as external systems, e.g. data providers, to aid business operations like risk analysis, monitoring, tracking, data collection and decision support. Here we will look into the components which are most relevant to the topic of this thesis; the geodatabase (GDB), image handling, and tools for geometric analysis.

The GDB is the ArcGIS data storage and management framework, a relational database management system (RDBMS) containing all spatial- and attribute data used by the application (ESRI, 2010). It supports a large number of spatial data types, meaning objects that describes or supports spatial data. Examples of this are satellite images, raster data, coordinates, survey measurements, and 3-D-models. There is no uniform standard for describing these kinds of objects, and the geodatabase therefore supports spatial objects from most of the major database management system (DBMS) providers like Microsoft, Oracle, IBM, and PostgreSQL. This degree of interoperability ensures that GDB can be used as a single point of storage for all spatial data an organization may have.

More interestingly, GDB allows for relationships between data types, for instance linking topology with satellite imagery. This allows for more advanced GIS analysis, as the combination of two or more datasets can provide a better insight into real life situations than viewing each of them separately. Some of the more common dataset combinations consists of geometric networks, terrain and topology, although others exist.

The ArcGIS approach for dealing with different types of spatial data is introducing new and more complex data objects than what we find in regular database systems. These range from relatively simple objects such as annotations and tables, to more advanced objects like network datasets (topologically connected network elements), dimension (lengths and distances on a map) and toolboxes (data- and workflow processes). In addition, GDB groups objects of the same type (lines, polygons, etc.) into classes to simplify storage and handling of these objects.

The foundation for most mapping and GIS technology is geospatial imagery, and unsurprisingly the ArcGIS Image Server (AIS) employs several techniques to make the handling of such images as easy as possible (ESRI, 2008). It is here important to note that this imagery is not limited to pictures taken by satellites and scanned maps, but also includes films, digital photography as well as digital terrain models. The amount of imagery used in GIS today is increasing, both in volume and assortment, and tools to handle these new types is something users expect to find in GIS software.

Compared to working with vector data, there are a few distinct challenges related to imagery (Pichler & Hogeweg, 2009). Firstly the size of imagery collections can be enormous, especially when dealing with photographs. To describe large amounts of images in a good way requires a great deal of metadata, both for the collection as a whole, but also for each unique object within the collection. This metadata would not only have to be generated/fetched when the object is added to the GDB, but also modified when it is updated, a job which often is too comprehensive to be done manually, and therefore needs to be handled by the system itself. Finally, there is the ever present problem of copyright, as not all images may be distributed freely, but this will not be discussed further.

As described in the previous paragraph, the size and complexity of imagery data sets is one of the things a modern GIS system has to deal with. AIS have several ways to tackle this challenge:

- **Centralization:** Imagery is stored in one place only, avoiding redundancy and having to update more than one repository.
- **Scalable:** No upper bound in terms of collection size, neither for number of objects or size in bytes.

- **Mosaicking:** As a collection may contain lots of imagery describing the same location, AIS grants the user control over which of this imagery should be prioritized, for instance the one with the highest quality.
- **Standards:** Supports a number of communication standards such as SOAP (not an acronym) and Web Map Service (WMS), making it possible to exchange data with other systems. This can be especially useful when gathering metadata.

Performance is also an important factor when dealing with imagery, images in particular. Users have grown accustomed to almost instantaneous feedback from the system they are requesting data from, for example seamless navigation in geospatial images (zooming, panning), free from any loading screens. Another benefit, in addition to satisfied users, is that the improved accessibility of the imagery in itself increases the imagery value. To achieve satisfactory performance in this regard, AIS does the following:

- **Server side processing:** The server on which the imagery is stored pre-processes all imagery as it is updated or new items are added, metadata included. The end result being that all information is up-to-date and ready for delivery when requested by the user.
- **Multiple services:** AIS allows several services to run in parallel, spreading the load and improving responsiveness.
- **Caching:** All, or parts, of the imagery viewed by the user can be stored on his or hers computer, removing the need to request these objects in the future, assuming no new versions have been added to the central repository.

Analysis is one of the key features in the ArcGIS package and as such the system comes with a large number of spatial analysis tools to be used with the various spatial objects. These can be divided into three groups, examples in the brackets; overlay (union, intersect), proximity (distance) and surface (slope). These toolkits cover most spatial analysis needs a user may have, and is well suited for ranking spatial objects, e.g. aerial imagery, from their content features alone. It is still important to note that ArcGIS does not have any default ranking method as those found in other systems; instead it provides all the tools needed to create your own method.

Even though ArcGIS is a commercial software package, it tries to provide open access to geographic data and functionality within the program by providing APIs and using relevant GIS and IT standards. The APIs allow developers to customize the use of the software to their specific needs, and use ArcGIS as an integrated part of a larger business system. This compatibility and interoperability with other enterprise systems is made even easier by ArcGIS supporting most IT standards, ranging from operating systems to web services such as XML and SOAP.

ArcGIS is but one of several enterprise GIS systems, and here two others will be mentioned very briefly; Quantum GIS[11] and GRASS GIS[12]. Both are open source alternatives to ArcGIS, released under the GNU General Public Licence, but neither have the same amount of features as their commercial competitor, especially in terms of spatial analysis. However, the fact that they are open source whilst maintaining a basic set of GIS functionality can make them better alternatives for many organizations.

[11] http://www.qgis.org

[12] http://grass.itc.it

3.5 3-D Maps

Mapping in three dimensions is a way to display visual information digitally in a more realistic way than what can be achieved using only two dimensions. It allows us to view something as we would in real life, making it easier to extract information and gain understanding of what we are looking at. 3-D-mapping is something used in many disciplines, and is especially known from medicine, where models of internal organs frequent TV-shows on a regular basis. Below we here look at how 3-D-mapping is used in GIS.

3-D maps have been used to display topographic information in geographic systems for some time. Topographic information from an area can easily be converted into a 3-D model, with the possibility of adding satellite images or other information layers on top. Depending on the data sources used to create the model, this has the potential of creating a good representation of the real world. However, there are two major drawbacks with this approach; it is typically based on top-down imagery and data, such as that taken by satellites, something which can severely limit the level of detail. And secondly, for the same reason, achieving a detailed 3-D environment requires a huge amount of imagery on which to base the view on, something which proves challenging on both hardware and software.

Google Earth[13], created by Google, is a desktop application which allows the user to freely browse geographical, political and social data. It is similar to the internet based map services described in chapter 3.3, but contains much more non-geographic information than these, in addition to displaying geographic content in 3-D. This is made possible by using several patented techniques for image processing (for a more in-depth analysis of these, see (Bar-Zeev, 2007)). But even with these there is a lack of detail and most buildings are displayed as boxes with muddy textures. For an example of how this looks, see Figure 3-6. It is possible to manually create and add 3-D structures to the system, but probably due to the volunteer effort this requires, mostly famous landmarks and buildings have been added so far.

Disregarding the fact that a certain degree of 3-dimensionality is involved, commercial 3-D maps behave similarly to their 2-D counterparts. They are often just another layer on top of the already existing map service, and therefore share functionality such as query and relevance ranking implementation.

Another approach for mapping physical environments is called robotic mapping. The idea here is using an autonomous robot to map a given environment, resulting in a 3-D model of that environment. This research area has gotten lots of attention in the field of artificial intelligence (AI) for more than two decades, and several working solutions for indoor use exist (Thrun, 2003). Over the last years there has been a growing interest in expanding robotic mapping to work in large-scale environments like cities. To achieve this, mobile robots are equipped with lasers for length measurement, cameras for taking pictures or video of the surroundings, and a GPS device for resolving the location. Experiments conducted have resulted in 3-D models with a much higher degree of detail than what satellite imagery is capable of (Montemerlo & Thrun, 2005) (Howard, Wolf, & Sukhatme, 2004). There is obviously a huge amount of data involved, and the scaling of such modelling is one of the challenges that lie ahead.

[13] http://earth.google.com

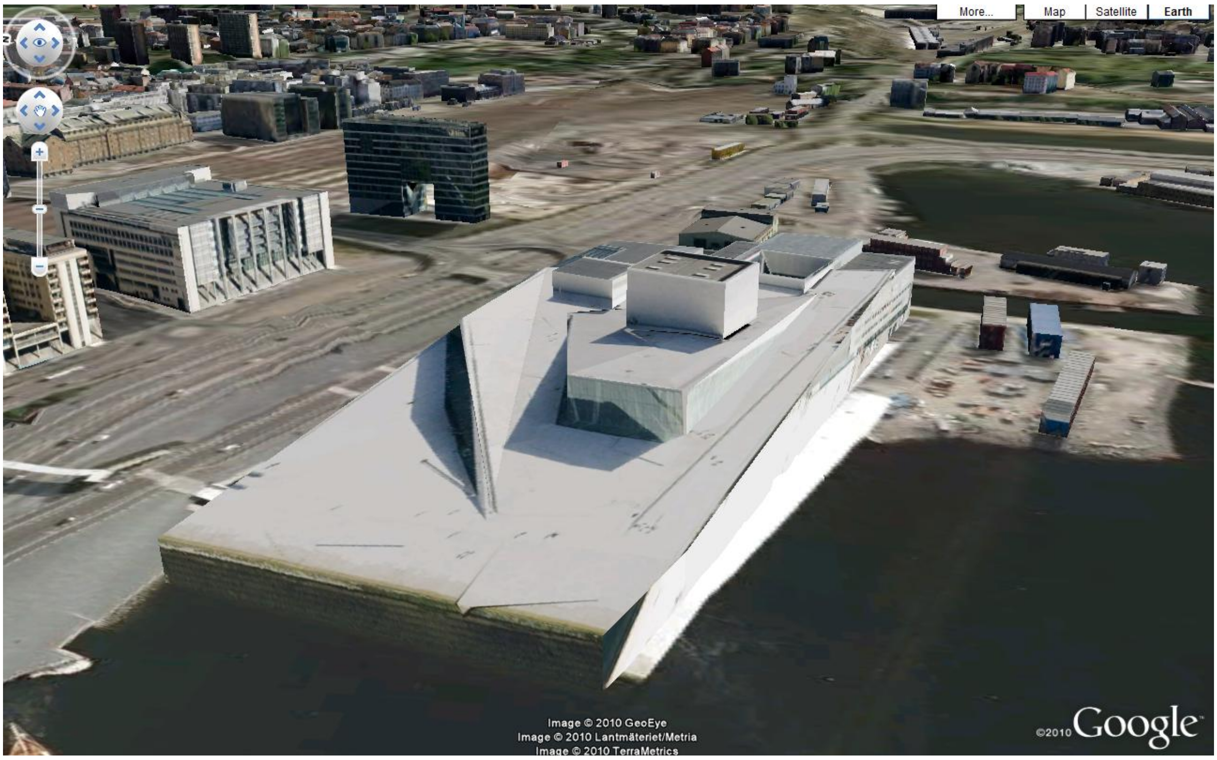


Figure 3-6: Google Earth, showing the Norwegian Opera & Ballet with surrounding buildings. Source: screen capture.

3.6 Augmented Reality

Augmented reality (AR) is a technology defined by three characteristics (Azuma, 1997):

1) Combines real and virtual
2) Interactive in real time
3) Registered in 3-D

This definition does not classify films with computer-generated imagery (CGI) objects (dinosaurs, spaceships, etc.) as AR, seeing as these are not interactive. 2-D live video overlays are not included either, as these do not include the real world in 3-D. It does, however, open for monitor-based interfaces, monocular systems, see-through digital media, and various other technologies. AR can be used in many areas, but it is most commonly found in medicine, military aircraft (heads-up display in fighter planes for instance), and visualisation.

Another field where AR has made an impact is in annotating urban environments. The typical approach to this is using the camera on a mobile phone and stream video to the phone's display, creating an illusion of having a see-through phone. Information of what you are looking at is then added as a layer on top of the live video. GPS is used in combination with cardinal point, functions which are available on most modern smart phones, to determine what the user is looking at.

One example of this is the Gule Sider® Live[14] application developed by Agens AS for the iPhone mobile phone. It works exactly like the description in the previous paragraph, with descriptions of various businesses appearing when you look at them with your mobile camera, in addition to

[14] http://www.gulesider.no/info/gulesiderlive/

displaying a mini map showing the location of various points of interest based on where you are standing. The information displayed through this application is highly commercial, such as advertisements and special offers, but in theory there is no limit to what information one can include in such a system.

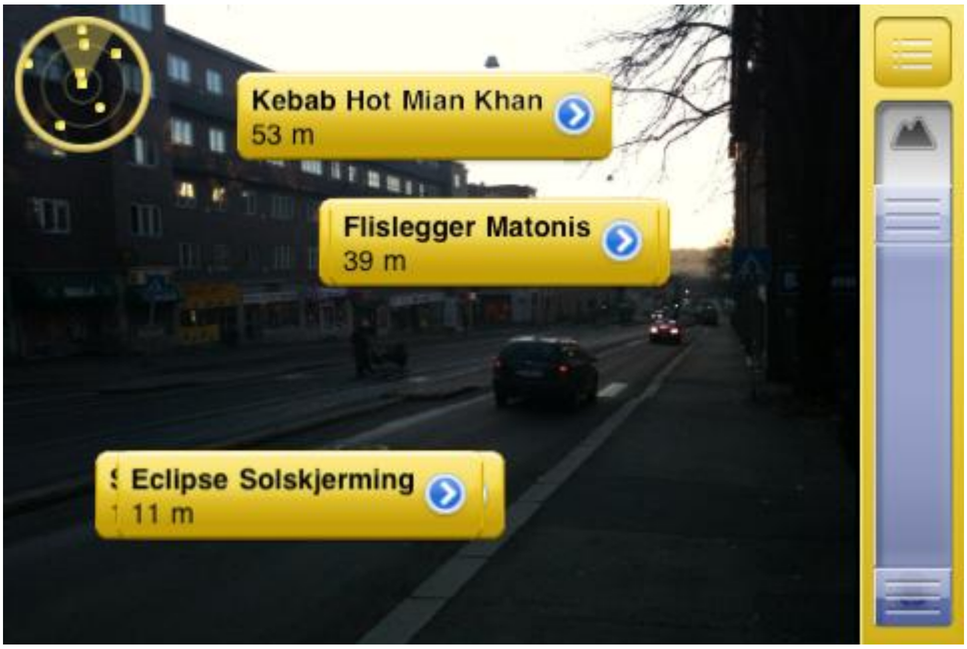


Figure 3-7: Screen capture from Gule Sider® Live, showing a street in Oslo with various business information as well as a mini map. Source: http://www.gulesider.no/info/gulesiderlive/

Augmented reality can be seen as a variation upon the other types of internet map services, but instead of providing an overview, it focuses only on what is in the user’s vicinity. This limit in scope has a huge impact on deciding what content to display, and simple solutions such as the one seen in Figure 3-1 are used; fetching the location and information on local businesses from an underlying search engine and using distance to decide what gets shown and what does not.

4 Requirements

The overall goal for all computer systems should be fulfilling the need it was designed for. This chapter outlines the requirements for a complete coordinate information retrieval system, covering most aspects but focusing on those relevant to data storage and the query process. These requirements are used as basis for the prototype described in chapter 5.

This particular system is centered on the retrieval of photos taken from airplanes and similar. These images have a bird's eye view, looking down at the landscape at a more or less tilted angle, and the image's coverage can be represented as a polygon on a 2-dimensional map. The length of the edges and the corner angles, however, will vary depending on the altitude and perspective of the picture. A very general view of such a system is illustrated in Figure 4-1.

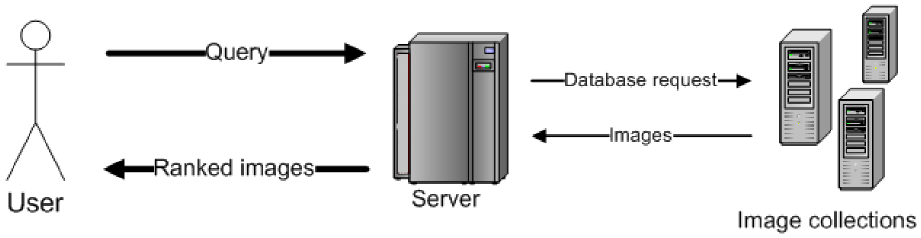


Figure 4-1: System information flow.

4.1 Query Types

As explained in chapter 2.2 there is no limit to the query types which can be used in an information retrieval system, but not all of them are equally suited for the current task. With this in mind, there are three query types relevant for this kind of system; point, shape, and text. Following is a more thorough description of these:

- **Point:** The user provides a single coordinate indicating a location on earth of which he or she wants to retrieve pictures of. This coordinate belongs to one of the Spatial Reference System Identifiers (SRID), and to improve flexibility the system should support multiple SRID systems, and do coordinate transformations between these when required. Example query, expressed as text: *"Find images covering the coordinate 63.23523 , 49.29278, SRID WGS84: 4326"*
- **Shape / area:** Instead of using a single coordinate, a set of three or more coordinates forming a geometric shape can be inputted as a query. This can be seen as an extension on the point based search, and requires the system to handle shapes with an unknown number of corners, making it a significantly more complex data type than a single coordinate. Another challenge with shape queries is that the type of shape also affects the retrieval process, not just the area they cover. Here we can separate between two different shapes:
  - **Circle:** The circle is the shape most resembling a point, and can in this situation be described as a "fuzzy coordinate". This means that the area closest to

the circle centre in most cases will be more relevant to the user than those areas close to the outer edge, and is something the system should take into consideration. Example query, expressed as text: *"Find images covering all, or parts, of a circle with the centre at coordinate 64.7823 , 40.214, SRID WGS84: 4326 and a radius of 2 kilometres."*

  - **Polygon:** A polygon can have anything from 3 to an unlimited number of edges, and it is this number which determines how the area within the polygon should be interpreted. The assumption here is that the more edges a polygon has, the more specific is the user's request is, e.g. in a polygon described with four edges the centre of the shape is the most important, while in a polygon with 14 edges the entire area is of equal importance. Example query, expressed as text: *"Find images within the area limited by coordinates 63.0 , 49.2 – 62.9 , 47.0 – 64.2 , 43.0, SRID WGS84:4326."*
- **Text:** A free text search where the name of a location, building or other geographic entity is inputted and the system retrieves pictures of this location. This type of search is made possible by having maps that contain both a spatial coordinate system (typically latitude longitude) and geographic objects with metadata referencing them to this coordinate system. Examples of this would be linking the Nidarosdomen cathedral to coordinates x and y, while the municipality of Trondheim covers the area between k, l, m and n. With this information included in the system, text queries can be converted to sets of coordinates, and a point- or shape based query can be performed. Example query, expressed as text: *"Find all images covering Gløshaugen campus, Trondheim"*.

Additionally there should be a number of advanced search options allowing the user to further describe his or hers needs. These parameters are optional, and pre-set default values will be used if they are left blank. Which advanced options exist depends on intended use and the amount and type of data stored in the collection(s), but some relevant parameters could be:

- **Area category:** The type of area coverage considered relevant. Three possible categories are described below.
  - **Close-up:** Pictures that cover a small area are preferred. Useful when looking for imagery of a specific building or landmark, with no interest in the surrounding area.
  - **Medium:** Images that cover a larger area, ranging from a city block up to city-wide coverage. These are typically taken from vantage points high up in the air, usually airplanes or skyscrapers, provides a better overview while still maintaining a decent level of detail.
  - **Large:** Every image that displays areas larger than a city. This is mostly imagery taken by satellite or high-altitude airplanes, with little to no details, but instead providing an excellent overview of how the different objects in the picture are placed in relation to each other.
- **Temporal quantifiers:** Limiting or preferring results from a given time period, affecting the ranking of results and/or discarding all images which does not fulfil this requirement. When working with large collections the option to limit results by time can severely reduce the amount of documents which has to be considered, improving both performance and the

quality of the results. This option uses the metadata belonging to each image, and might therefore not always be available.

- **Source:** The author of the image, or the collection it belongs to. Similar to temporal quantifiers in many ways, entirely dependent on metadata being available, but can if used properly improve the query process radically.
- **Advanced text:** Options directly related to textual queries. The advanced options govern if any special considerations need to be made when the query text is interpreted by the system. Examples of this is whether to consider the text as a phrase, single words, if stop words are to be part of the query, and if similar words should be considered.

4.2 Image Types

Images are the single most important component in an image retrieval system, as without any image collections to search in, the system has no value. With images originating from different sources it is almost guaranteed that they are stored in different ways. Such a system must therefore have methods in place for dealing with a wide variety of pictures, both from a technical and an information retrieval point of view.

The technical aspect is first and foremost being able to encode and decode the various image formats available. One part of this is having a basic understanding of the strength and weaknesses of each format so operations such as scaling and generating thumbnails can be optimized. Additionally, ways of managing attached metadata independent of format needs to be present, e.g. having the ability to both fetch and modify the metadata fields that each format possess, as information can be gathered from these as well as from the image’s contents. Also there should be no limitations to image size, neither in terms of bytes or pixels, as both can vary greatly.

From an image retrieval point of view we can separate between several image types based on their contents. Exactly how this classification is determined will depend on the system, but in this case a categorization is made from looking at the perspective/angle of the image, as well as the physical area coverage. The system should be able to identify these types and evaluate them accordingly. With this in mind, the relevant image types of aerial photography are as follows (illustrations from Google Maps):

- **Straight down, small area:** The image is taken from the air, looking fairly straight down on a small area (city block or smaller). This type covers a small, almost quadratic area.

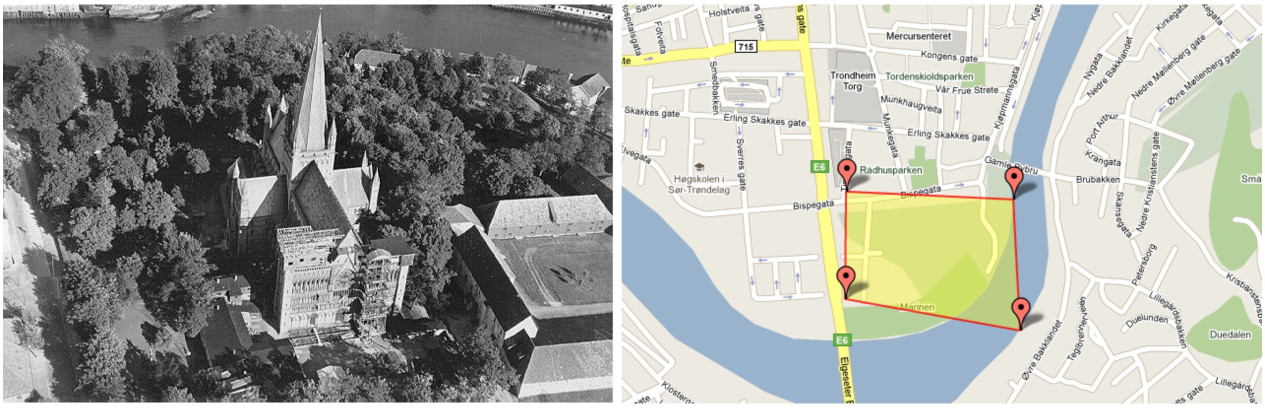


Figure 4-2: Straight down, small area image type.

- **Straight down, large area:** Similar to the previous type, but covers areas larger than a single city block. When seen on the map forms a large, almost quadratic, area.

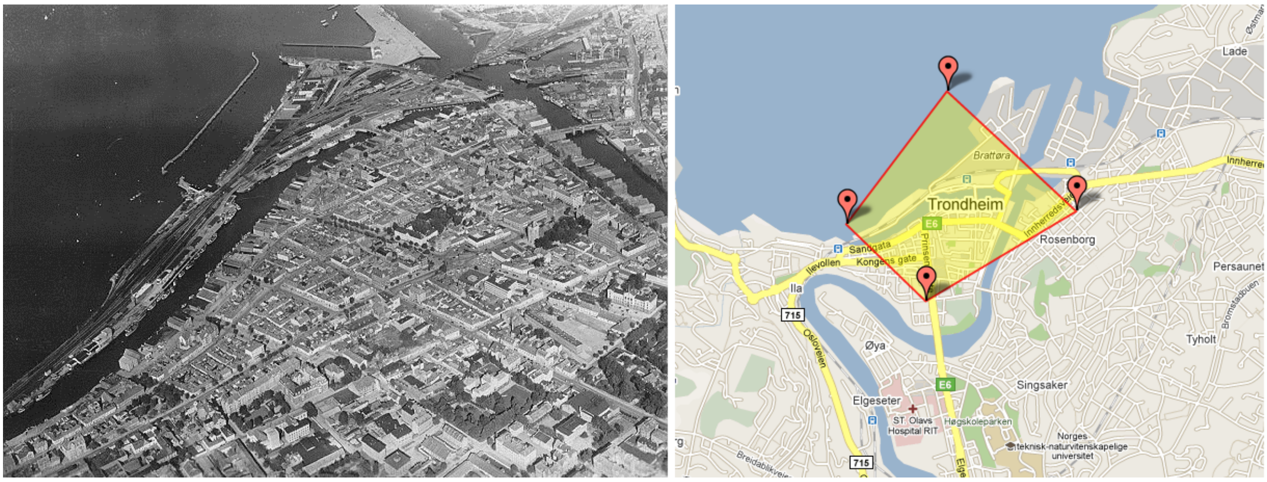


Figure 4-3: Straight down, large area image type.

- **Fairly horizontal, small area:** The photograph is taken at an angle, resulting in an image where the distance between objects in the picture and the point of photography differs more than in straight-down images, even though this type only covers a small area. This typically forms a small trapezoid-shaped coverage shape, with the edge closest to the photographer being shorter than the others.

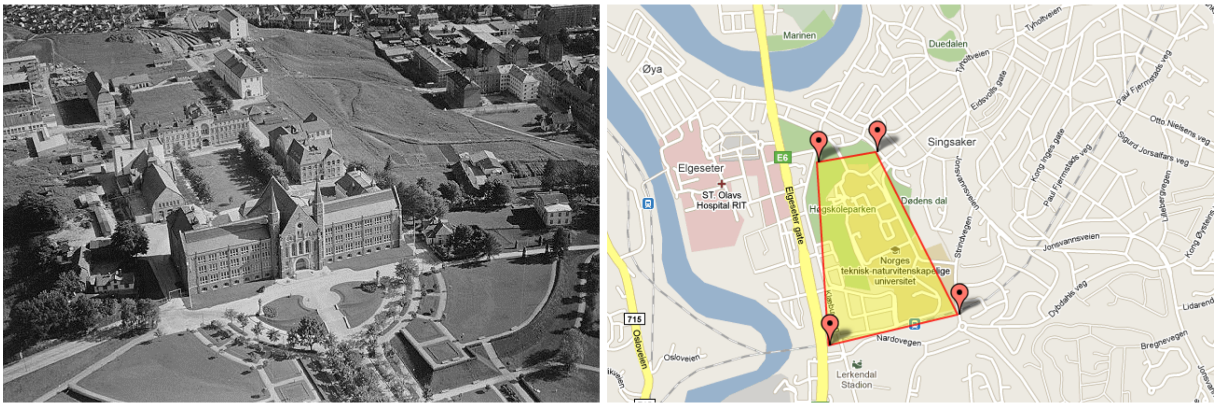


Figure 4-4: Fairly horizontal, small area image type.

- **Fairly horizontal, large area:** Same principle as the previous type, but covering larger areas. The increase in size makes the difference in distance even more significant, objects ranging from detailed in the front to indistinguishable in the back. Like its smaller counterpart, this image type forms a trapezoid when seen on the map.

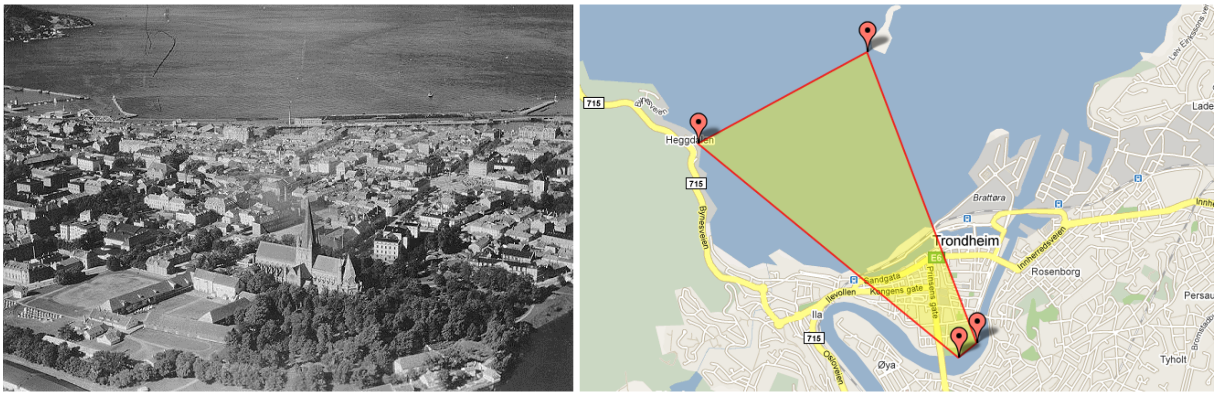


Figure 4-5: Fairly horizontal, large area image type.

- **Horizontal, horizon showing:** A more skewed variation of the previous type, as the image here “stretches” into the horizon. The differences in edge length and level of detail can be extreme, and the total area covered will almost certainly be very large.

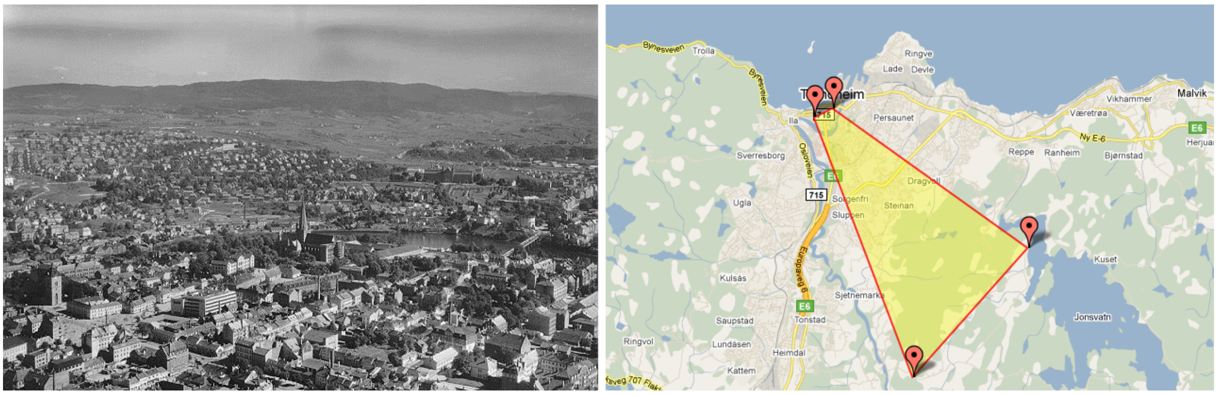


Figure 4-6: Horizontal, horizon showing image type.

- **Horizontal, no horizon:** Same as the type with a visible horizon in every way, but with mountains or similar blocking the horizon, providing an extra challenge when representing the area seen in the picture, as the edge farthest away from the point of photography might not be accurately representing the actual coverage.

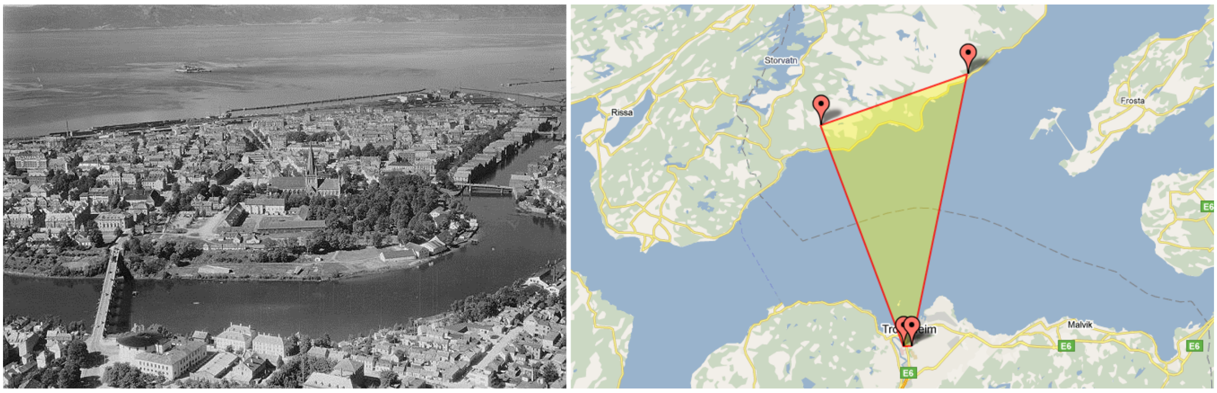


Figure 4-7: Horizontal, no horizon image type.

4.3 Storage

Storage is an overall term for all storage systems involved in handling the different kinds of data and information; the images themselves, metadata, and content features. We can separate these into two distinct storage categories; text and non-text. Images belong in the latter, and should be stored in a file- or database system specifically designed for this purpose. There is, in fact, no requirement having images and metadata/features stored at the same location, as long as the relations between them are maintained.

Both metadata and features can be stored as text, but doing this for all fields is inefficient when dealing with spatial objects. Some of the features extracted from images are geometric, for example coverage (polygon) and image centre (point), and can therefore be stored as geometric objects instead of having to convert from text each time we perform an operation on them. For this reason a database system which supports spatial data should be used, which not only can represent features more completely compared to what the feature describes in the real world, but also allows us to move parts of the business logic down to the database level. This can reduce the amount of programming code we have to write and also yield a performance boost by using such a specialized system that a database is for some of the calculations.

Even though some features can be described as geometric objects, the majority of features, especially the external metadata, are simple text- or numeric fields. These data types are something most traditional databases excel at handling, and finding suitable solutions for the storage of these is only a matter of selecting the text- or numeric data type that fill requirements for accuracy and length.

Regardless of how features and metadata is stored, all values should be as accurate as possible. This might seem obvious, but spatial coordinates are often represented by more digits than your average data type can handle. Which data type is best suited for the task depends on the programming language used, and it is up to the individual developer to make sure an appropriate choice is made. The downside to using a very high degree of accuracy is the effect it might have on query performance (in terms of speed), but this can be circumvented by rounding off values when required, while keeping the original value in storage.

4.4 Result Ranking

Whenever the number of returned documents exceeds what can be displayed simultaneously, typically on the same screen, the system needs a way to sort the retrieved documents, placing the most relevant on top making them more accessible. This situation is bound to occur more often as the collection increases and a large number of documents match even narrowest queries.

The ranking is based on the parameters passed on from the query, or default values for those parameters left blank. An implication of this is that the amount of information the user provides directly affects the quality of the ranking. In an ideal world where users happily fill out all fields this would not be a problem, but in reality this scenario is an exception more than the rule and often only the minimum amount of search options are used. This makes selecting good default values an important task.

Based on the query- and images types described earlier in the chapter, several ways to rank the results from a query can be identified:

- **Metadata:** Ignoring image contents, metadata alone is used for ranking. This means fields such as filename, date, and image size in pixels are used to determine relevance. This approach is used when searching for images in web search engines, and in part by ADL Globetrotter (described in chapter 3.2) where results can be sorted by date.
- **Textual content:** Here the contents of each image have been described using free text, e.g. listing the name or addresses of all buildings/landmarks seen. Each picture is then ranked on how well they match up to a textual query, using text retrieval techniques to determine relevance.
- **Distance:** The physical distance (on the map) between the query coordinate and the centre of each image in the collection. Works for both point- and area based queries, assuming centres are calculated for the latter.

- **Coverage:** A ranking is done based on how much of the spatial query (point or area) is contained within each image. In other words relevance is decided by how much area the query and each of the images share, more being better. Using a point-query this method returns a Boolean value, true if the point is found in the picture, false if not.
- **Area:** The size of the area covered in each image governs how relevant it is to the user's query. For this metric to be meaningful it has to be combined with other factors like coverage, distance or other search parameters.
- **Relevant areas:** Where in an image the query coordinate or area is located decides how relevant the image is. An interesting area can be fuzzily described as "the visual focus of the picture", and therefore differs from image type to image type. For example is the area close to the edges in an image taken straight down in less focus than the centre, while horizontally taken pictures have their most relevant areas closest to the photographer, regardless of edges. How these relevant areas are placed compared to the spatial query can be used for ranking.
- **Combinations:** Two or more of the approaches described above can be combined.

As this task specifically deals with spatial features, it is expected that at least one of the ranking methods involving the internal features of the document, i.e. not metadata or textual content, are used in calculating relevance.

4.5 User Interface

It does not matter how brilliant a system is, if no one is able to use it. Due to the visual nature of images a graphical user interface (GUI) is almost always required, allowing the users to add new images and perform queries on the collection. The GUI layout is best left for human-computer interaction (HCI) experts, but we can still identify some elements that should be included. Most importantly there has to be a digital map tool where the users can mark coordinates and draw polygons, used both when annotating images that are to be added and when querying the collection. This map tool transforms these points and areas into coordinates understood by the system logic.

Even with a visual point-and-click way of using the system, a textual alternative should be included. Using text ensures accuracy, and is undeniably simpler in those cases where the coordinates are known beforehand. For some operations, e.g. searching for a landmark using text, or adding additional metadata to an image, text is vastly superior to visual approaches, and therefore the user should be allowed to choose.

Another key element is the result browser. The system returns a number of images as the result of a query, ranked by relevance, and there has to be a way for the user to see these results. An image browser of some sort will be suited for this task, allowing navigation between retrieved images, as well as other functionality (e.g. saving, sharing, finding similar, etc.), depending on system.

4.6 Performance

For a search engine to be useful it needs to answer queries within a reasonable timeframe. A possible way to achieve this is installing more hardware, but this approach is inconvenient and expensive. Instead there are a number of design choices which can help accomplish the same thing, and the system can employ all or some of these, depending on the speed requirements.

As mentioned in chapter 4.3, some image features are be stored as spatial objects in the database, and indexing these fields will improve performance. There are several data structures suited for indexing geometries, R-tree and Quadtree to mention a couple, the choice ultimately being governed by database solution.

The responsiveness when administering the image collections (i.e. adding, editing, and deleting images) is not nearly as reliant on lightning performance as the performance when querying the system, something which can be taken advantage of. As much work as possible should be calculated when an image is added to the system, e.g. extracting features and updating indexes, to avoid having to do this when the system is in "normal" use. Pre-processing as much information as possible reduces distributes the computations costs in a more sensible way, therefore reducing the delay attached to each query.

These are but a couple of the techniques that improve performance, and many more can be applied as needed, caching and load distributions being examples of this. Often developer ingenuity is the limitation here, as well as actual need. Not all systems require lightning quick response time, making it harder to justify investing time and money into performance boosts.

4.7 Other

Depending on usage scenarios, an image retrieval system such as this can have a number of smaller modules. Typical examples are text query interpreters, coordinate transformers, access control mechanisms to ensure copyright, and various tools to maintain interoperability with other applications. These modules can in fact be vital for the success of the system, even if they are not part of the retrieval process itself.

5 Implementation

In this chapter a prototype for an aerial photography image retrieval system is presented, listing and describing the different component included in the software. The purpose of this prototype is being a way to witness the behaviour of image- and query types by running queries on an actual image collection. The prototype is described in chapter 5.3. Also, a technical outline of a complete system conforming to the requirements described in chapter 4 is found in chapter 5.2, an outline which the prototype derives from.

5.1 Design Mindset

As with all things designed by people, software is affected by those creating it, and this technical description is not an exception. The design decisions are based not only on what technology will get the job, but is also affected by my personal preferences on what is important. These preferences are described below, making it easier to understand why some seemingly less suited options have been chosen over others.

I feel the main purpose for a system like this should be making it publicly available, meaning it is designed in such a way that regular users, and not just experts, find it easy and appealing to use. In many ways this is a more commercial approach than what is common in research environments, but I feel this kind of software is useful to the general public and therefore needs to be designed with this in mind.

Making the system accessible to as many people as possible is the first step towards commercial success. In practice this means not limiting the system to a certain operating system, browser, internet service provider, country, etc., as well as steering clear of complex installations and configurations. Another aspect of this is keeping the threshold for use low, with advanced functionality being hidden unless the user chooses otherwise.

The amount of involvement from the user is also something that should be kept at a minimum, making an impact on feature extraction in particular. Even though forcing the users or system administrators to manually annotate images can result in excellent descriptions well suited for relevance ranking, this is also a sure way of excluding a large part of the user base. A common denominator for the most successful search engines in use today is that they require very little effort from the user. The same mentality will be used here, relying on automatic approaches over manual labour.

When choosing between equally suited technologies, the most familiar of these will be chosen. This serves two purposes; firstly, when used in the GUI, it introduces a number of elements the users might have seen or worked with before, something which might reduce the steepness of the learning curve. An example of this is the map component, where using something well-known like Google Maps means the users does not have to be retaught how to navigate or perform other basic map operations. Secondly this approach makes it possible for developers to see what the chosen technology is capable of by looking at what others have done before, as well as being inspired by this.

In general there is no reason to reinvent the wheel if a suitable component for the system already exists. Even if these components have a price-tag attached, i.e. cannot be used freely, the savings made from not having to develop and maintain your own custom solution will often more than make up for it. If possible open source alternatives should be chosen. Firstly because I like the principle of open source, but also because they tend to be more secure and often come with an active community attached. This community can be an invaluable source for advice and help on how to solve problems, should some occur.

For the image retrieval part of this system to be a success, three factors are considered more important than all others; it needs to return relevant documents, the speed of retrieval needs to be fast, and it has to scale well. With the amount of geographic imagery ever increasing (as nothing is ever deleted) the collection is bound to become large, and with it, a drop in retrieval speed. Even so, the response time has to be kept low enough to keep the user base from giving up or choosing other systems. Taking performance measures that lowers overall retrieval quality is also acceptable, as having a good, fast system that people use is better than having an great but slow piece of software lying around unused. This however, should be a last resort.

5.2 Technical Outline

Figure 5-1 shows a possible technical solution for an image retrieval system fulfilling the requirements in chapter 4, and this subchapter briefly explains the different parts shown in this figure.

5.2.1 Structure

The system is divided into two parts; web and server. This approach is chosen so that the system can be accessed through a web browser, avoiding potential problems with the user's operating system and hardware. It does require the users to have an internet connection, but seeing that it is very unlikely that the image collections are stored locally in the first place, this is not a major concern.

The two parts are in turn divided into smaller modules; each assigned to do a specific task. These are grouped by what kind of operations (logic, storage, UI) they perform. Communication between modules is carried out between their respective interfaces, allowing the inner workings of each module to be changed as long as the interface is kept intact. This lack strongly connected components is something which eases maintenance and upgrades. On an even more detailed level the modules again are divided into classes, again with interfaces, following an object oriented mentality.

5.2.2 Components

The user interface is made with Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript, and Google Maps. HTML and CSS take care of the visual aspects, providing fields to formulate queries and an image browser for displaying results, while JavaScript handles the business logic situated in the UI. Most of this consists of passing values to and from Google Maps, the selected map provider. Google Maps is chosen for its mature JavaScript API, where adding your own map markers and polygons is a breeze, and the fact that it is a well-known service, already being used in several other web applications.

Storing all images in the same database as their features and metadata is a simple and effective way of doing things, but it also introduces a number of constraints. It increases the load on the server each time an image is retrieved, having to fetch both image data and features from the same database, decoding as needed. More importantly this approach assumes storing all images in the same location is legally possible, something which is highly unlikely in this age of copyright infringements.

With this in mind the images are separated from their content data, storing the images at a different, possibly external, location, with the content maintaining a field with the location of the image data. Images are fetched from external collections when required, either directly from a file structure or through a portal, depending on how the external source is configured. This eliminates the problems mentioned in the preceding paragraph, although it complicates the system, for instance in terms of consistency and not having control over the external collection.

Information is passed between the web components and the server using asynchronous JavaScript and XML (AJAX). AJAX allows for the creation of dynamic web applications as the server and client can work independent of each other, giving the developer full control of what is shown, and when. Messages, mainly identifiers and features, are sent as XML between the JavaScript in the web client and a Java servlet on the server.

The business logic is written in the Java programming language. Java is widely used and works well with other components, e.g. database systems and web technologies, making it well suited for this kind of system. In short the business logic covers all functionality not directly related to UI or the actual storing of information. This logic is divided into four separate modules:

- **Storage manager:** The database module which handles the database connection and passes queries to the database for execution, as well as receiving the results from these queries.
- **Feature extractor:** Used when adding pictures to the collection. Receives metadata and the most basic features from the UI, and based on this generates the rest of the content features. An insertion query is formulated and handed over to the storage manager.
- **Query executer:** Receives user queries, expressed as text, points or polygons, which are to be performed on the collection from the UI. These are interpreted, for instance by performing operations on text, and formatted queries are created and sent to the storage manager.
- **Result handler:** A result set containing image metadata and features is received from the storage manager. If this result set is from an insertion query, a message is passed to the UI informing of this. In the case of a result set containing images matching a query, relevance is calculated for each of the objects before they are sorted and passed to the UI, where they will be displayed to the user.

PostgreSQL with the PostGIS extension is used for storing features and metadata. PostgreSQL is an open source object-relational database system supporting most major operating systems and programming languages. The main reason for choosing it however, is PostGIS, an extension that spatially enables PostgreSQL, i.e. adding a set of functionality for working with spatial objects. This makes the combination an excellent backend database for geographic information systems, like the one described in this chapter.

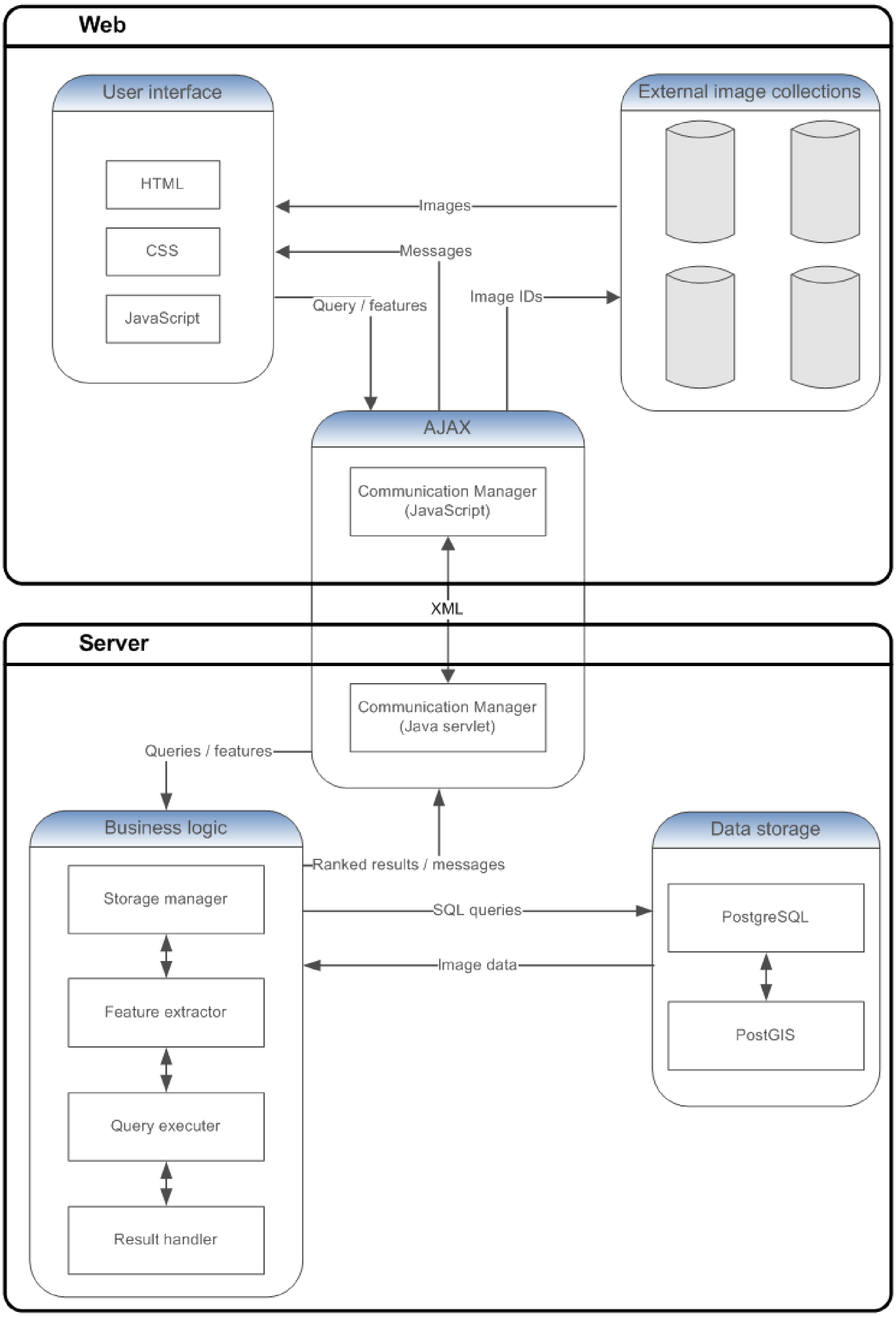


Figure 5-1: Architecture suggestion for a complete aerial photography image retrieval system.

5.3 Prototype

A prototype adds a practical dimension to research, making it possible to study actual behaviour and not rely on theory alone. Part of this thesis was creating such a prototype, a subset of the complete spatial image retrieval system described in chapter 5.2. The prototype was created to investigate the image retrieval aspects of the system, and the remainder of the chapter describes the relevant components in detail.

5.3.1 Limitations

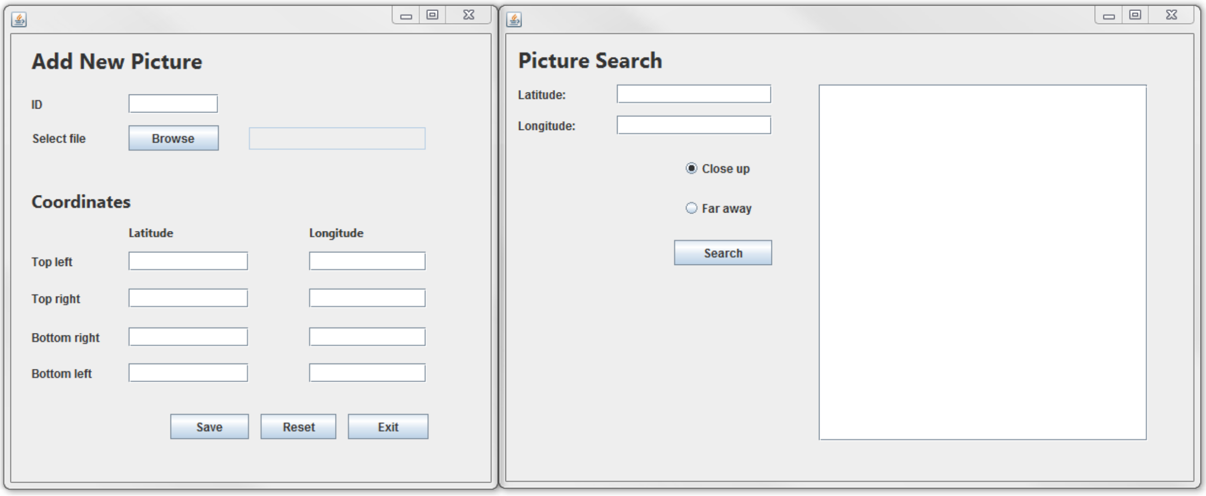
Implementing a complete system would be beyond the scope of this thesis, and the prototype therefore only covers a small part of the functionality one can expect to find in a real system. The focus of this paper is the image retrieval aspects of the system and with this in mind the business logic and database are the only parts included in the prototype, and even these are simplified.

The prototype is limited to point-based queries only. This query type, although a simple one, is a good way of seeing how the different image types (see chapter 4.2) affect the query and ranking process, as well as sharing some of the characteristics also found in shape queries. By excluding textual queries there is no need to store and extract metadata, nor have tools for interpreting text strings. Even though these are interesting topics, they fall outside the main focus of this thesis.

An outcome of the limitations is that the parts of the complete system located on the web have been moved, or left out entirely from the prototype. A result of this is that the image collection is merged with its features and stored in the same database. Also a replacement GUI written in Java has been created, making inserting images and performing queries possible.

5.3.2 Graphical User Interface

The GUI consists of two separate windows, one for inserting images and one for expressing point queries on the collection as well as displaying the results of these queries. Unlike what would be the case in a complete system, there is no map component; images are registered by manually setting ID, corner coordinates, and attaching the image file itself. Similarly the point query is expressed by entering the latitude and longitude coordinates in text fields, with no option of marking them on a map. Advanced search options are reduced to selecting a preferred area size of the retrieved images, choosing between “Close up” (small) and “Far Away” (large).



Add New Picture

ID

Select file

Coordinates

	Latitude	Longitude
Top left		
Top right		
Bottom right		
Bottom left		

Picture Search

Latitude:

Longitude:

● Close up

○ Far away

Figure 5-2: The two windows in the prototype GUI.

5.3.3 Feature Extractor

As described in the previous paragraph the only information the system receives about the image's content are the four coordinates representing the area shown in the picture. All other content features, barring distance (see the following chapter for a description of features), are calculated based on these corners, pre-processing each image when it is added. This removes the need for more user interaction, and also reduces the calculation for each query to a minimum. The picture ID and image file are also inputted, but neither is used in the feature extraction process. When the extraction is concluded the features along with the ID and file are sent to the storage manager.

5.3.4 Included Features

This subchapter lists the different features which are extracted from the pictures in the system. The purpose of these is to determine the relevance used in the result ranking. As the coverage of each image can be rendered on a 2D plane, mostly geometric functions have been used, with parameters represented in the BigDecimal data type. BigDecimal was chosen to make sure all features were expressed as accurately as possible, something which was not feasible using the primitive data types in Java. A description of included features, and how these were calculated, can be found below. Relevant, yet excluded, features are also mentioned in a separate list.

Included features:

- **Outer perimeter:** The four corners of the pictures, each of these expressed as longitude- latitude coordinates. This feature represents the entire area displayed in the picture, and is the only feature which has to be manually entered when adding the picture to the database. No further calculations are required for this. The original photography and the outer perimeter feature (displayed in Google Maps) are shown in Figure 5-3.

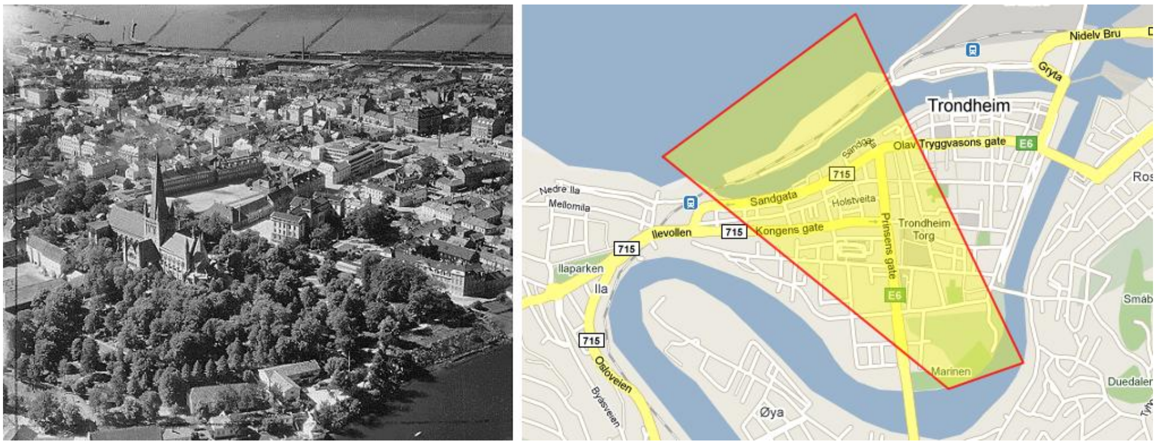


Figure 5-3: Outer perimeter feature and original picture.

- **Centre:** The centre point of the area displayed in the photograph, represented as a single coordinate. This is calculated by finding the middle of the outer perimeter edges, drawing lines between opposing middle points, making the intersection the area centre of the image. This process is displayed in Figure 5-4, along with the resulting feature.

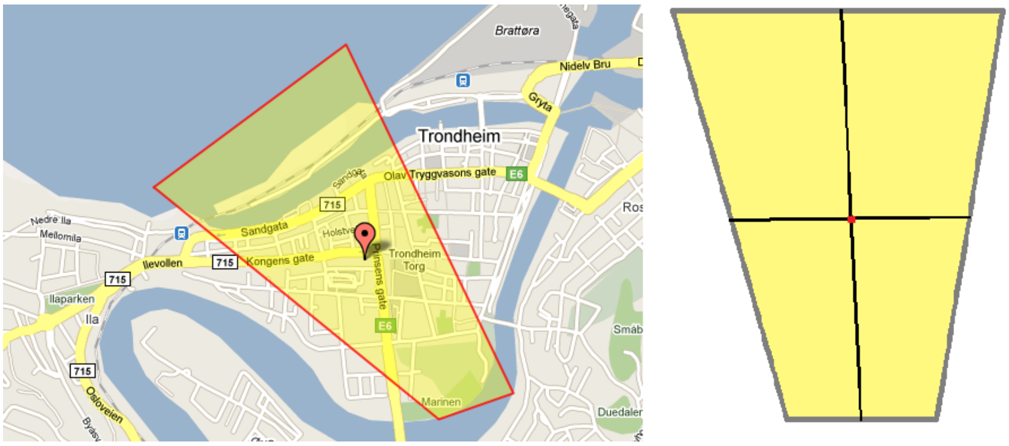


Figure 5-4: Centre feature and calculation illustration.

- **Inner perimeter:** Four points indicating the corners of a smaller quadrilateral placed in the centre of the outer perimeter. The inner perimeter is an attempt to quantify the most interesting part of the picture without requiring additional human input. The perimeter use the centre point and outer perimeter to calculate four new corner coordinates that are 25% closer to the centre than the outer perimeter, and these coordinates makes up the inner perimeter, as shown in Figure 5-5.

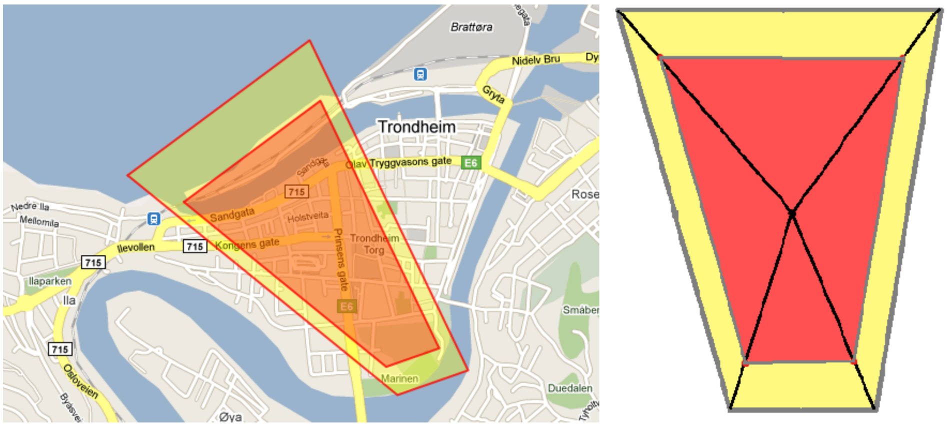


Figure 5-5: Inner perimeter and calculation illustration.

- **Angle/perspective:** In an attempt to quantify the image perspective by comparing edge lengths, we can make an estimate of how much the camera was tilted when taking the photograph, i.e. if two edges are a lot shorter than the other two, we can assume the camera was held fairly horizontal. The value is calculated by dividing the average length of the two edges adjacent to the shortest edge, with the length of the shortest edge. A high number indicates the photo was taken fairly horizontally, while a low number is a sign of the opposite.

  Formula (see Figure 5-6): $Angle = \frac{(S4+S2)*0.5}{S3}$

- **Area:** Using the outer perimeter feature the total area seen in the photograph can be estimated. The formula for determining area with four known vertices is used for this. Area tells us, in combination with angle, which image type (chapter 4.2) we are dealing with. On its own, area indicates the level of detail we can expect to find in the image, going by the assumption that an increase in area results in a detail reduction. Formula (see Figure 5-6): $Area = \left|\frac{x1*y0-x3*y0-x0*y1+x2*y1-x1*y2+x3*y2+x0*y3-x2*y3}{2}\right|$

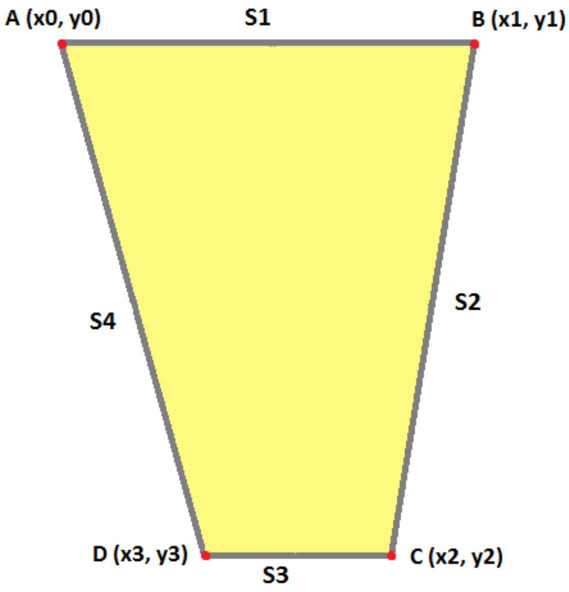


Figure 5-6: Annotated outer perimeter.

- **Skewed centre:** A lower perimeter is created, using the end coordinates from the shortest edge and the middle points of the two adjacent edges, effectively cutting the image in two. Diagonal lines are drawn between the corners, and the skewed centre is the point where these lines intersect. This is mainly useful for horizontal images, as an alternative to the centre feature. Here the most interesting area is that close to the shortest edge, and the skewed centre is the centre of this. The skewed centre is shown in Figure 5-7.

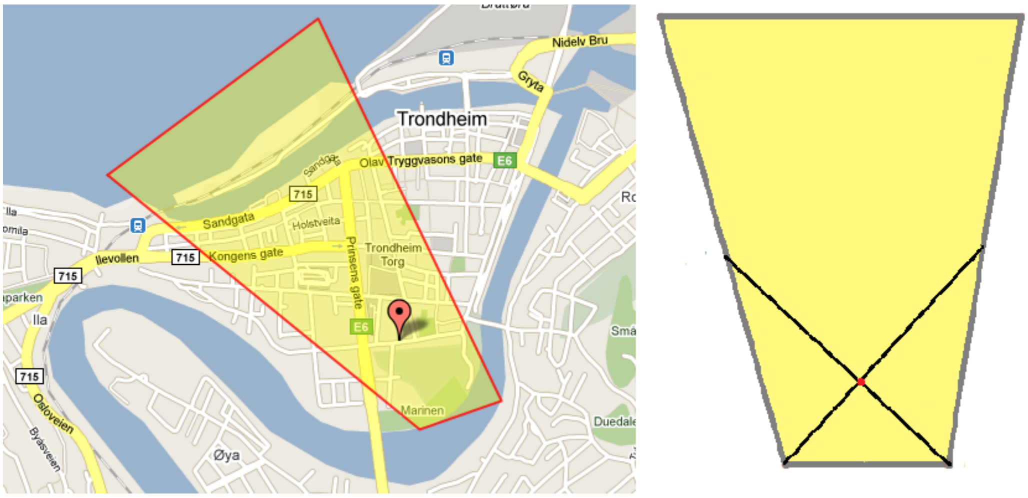


Figure 5-7: Skewed centre and calculation illustration.

- **Close perimeter:** Using the same formulas as with the inner perimeter, but using the skewed centre and lower perimeter as parameters. Similar to the skewed centre, the close perimeter is intended as an inner perimeter replacement for horizontally taken pictures. See Figure 5-8.

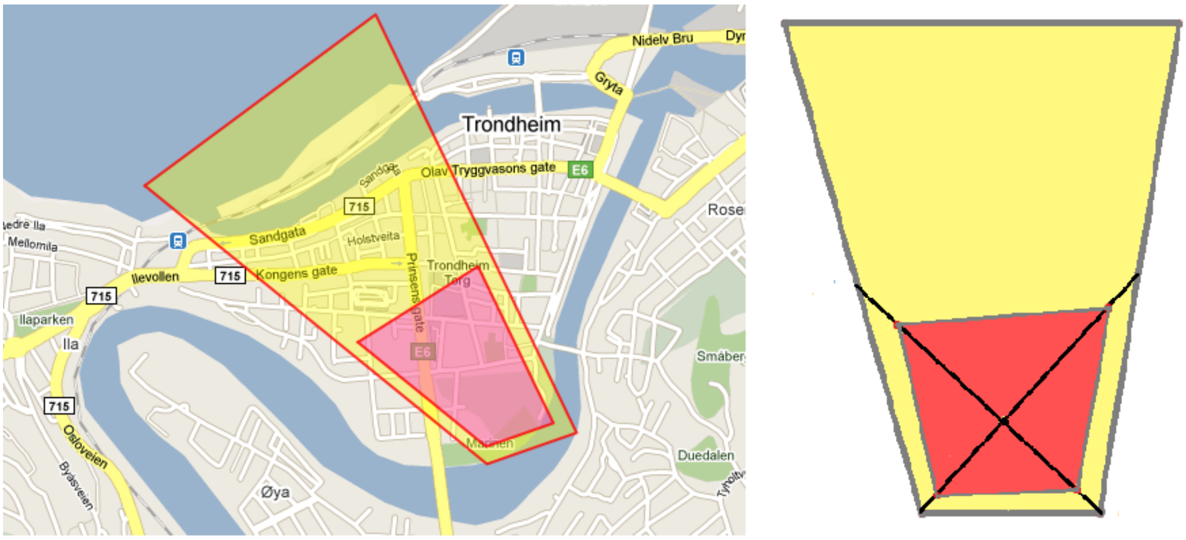


Figure 5-8: Close perimeter and calculation illustration.

- **Distance from photograph:** The distance from the query (point- or area-based), to the centre of the photograph. Unlike the other included features, this depends on the query before it can be calculated, and as such cannot be pre-processed. It is also the only included feature able to rank images that does not contain the query coordinate in a meaningful way.

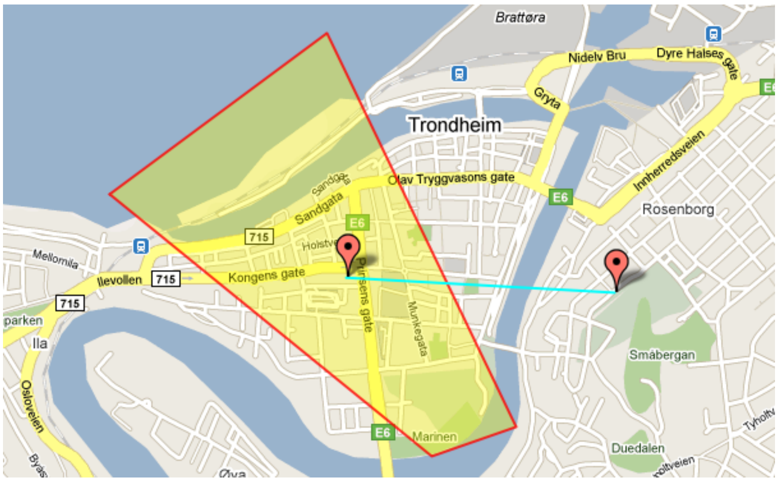


Figure 5-9: Distance, right marker indication query coordinate.

Excluded features:

- **Areas of interest:** Allowing the user to manually enter the areas he or she considers to be the most visually interesting in an image. The advantage of this is that human perception is good at identifying visual objects, regardless of image type, even though it introduces some subjectivity to the system. However, the main reason for excluding this feature is the extra labour that goes with manually selecting these areas, something which I do not think we can expect a user to bother with, especially when a large number of images require annotating. Figure 5-10 shows an example of how this might look on the example photograph, having two areas of interest for illustration purposes.

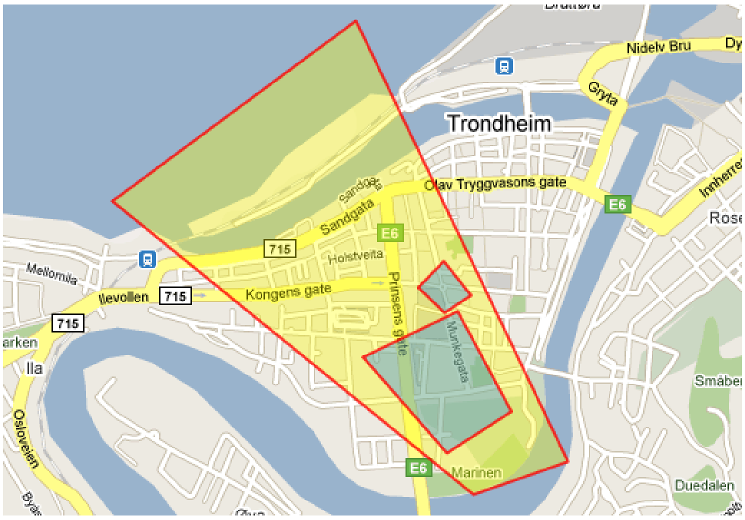
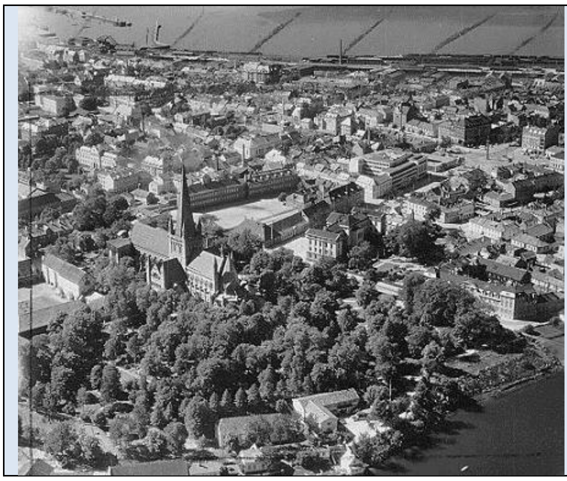


Figure 5-10: User annotated areas of interest.

- **Landmarks:** Describing pictures with the name and descriptions of the landmarks, areas and similar found on it. This could be done either manually or automatically using a database containing geospatial relations between coordinates and objects. As the prototype only supports spatial queries this feature would not be used, and was therefore excluded.



- Nidarosdomen
- Trondheimsfjorden
- Trondheim square
- Trondheim city hall
- Prince Carl’s bastion

Figure 5-11: Original photo with listing a handful of landmarks.

- **Location of photographer:** By combining area, edge lengths and angle, the location of the photographer can be estimated. However, no formula was found to calculate this, and the usefulness of the feature considered minor seeing as this feature says very little about the image contents. An approximation of this feature is shown in Figure 5-12.

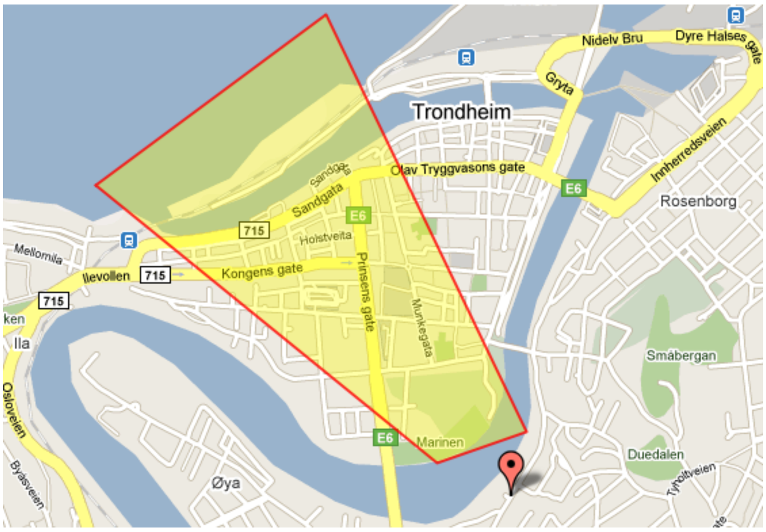


Figure 5-12: Location of photographer.

5.3.5 Storage

A PostgreSQL server with the PostGIS spatial extension is set up locally, removing any network latency when interacting with the database. As mentioned in the introduction to the prototype are the images themselves also stored here. All data is saved in a single table, described in Table 1, and an insertion query example can be seen in Figure 5-13, using example numerical values and replacing all coordinates with letters for readability. ST_GeomFromText is a unique PostGIS function that takes in a set of coordinates and an SRID, creating a geometric object.

```
INSERT INTO    geoimage ( id , area , angle , centre , outerperim , innerperim , scenter , closeperim ,
                         imagefile )
VALUES         (1234 , 0.56, 2.3 , ST_GeomFromText(a b , 4326) , ST_GeomFromText(c d e f , 4326) ,
               ST_GeomFromText(g h i j , 4326) , ST_GeomFromText(k l , 4326),
               ST_GeomFromText(m n o p , 4326) , bytes[] )
```

Figure 5-13: SQL insertion query.

Table 1: Prototype database table.

Name	Data type	Description
ID	Integer	The unique image identifier and primary key of the table.
Area	Numeric	The area coverage feature, stored as an exact numeric value with a precision of up to 1000 digits.
Angle	Double precision	The angle feature represented as an 8-byte floating-point number.
Centre	Geometry: Point	The centre stored as a point, one of the spatial data types made available through PostGIS. Specified to only allow 2-dimensional points.
Outerperim	Geometry: Polygon	The outer perimeter stored a polygon, a PostGIS spatial data type. Specified to 2 dimensions, but an unlimited amount of corners.
Innerperim	Geometry: Polygon	Stores the inner perimeter, otherwise identical to Outerperim.
Scenter	Geometry: Point	Stores the skewed centre, otherwise identical to Centre.
Closeperim	Geometry: Polygon	Stores the close perimeter, otherwise identical to Outerperim.
Imagefile	Bytea	The image's file itself, stored in a byte array. Can store images up to a max size of 1GB.

None of the columns have been indexed. Indexing can help improve performance, but there are not enough images in the test database to do any meaningful experiments in this area, and indexes have therefore not been included.

5.3.6 Query Execution

As mentioned only point queries are supported in the prototype, and each query takes three parameters; latitude coordinate, longitude coordinate and a Boolean value indicating if close-up or overview pictures are preferred. An SQL query containing the coordinates is formulated and executed on the database, a query which is shown below (the latitude coordinate is replaced with x and the longitude with y for readability).

```
SELECT      id,
            area,
            angle,
            ST_Distance(ST_GeomFromText( 'POINT(x y)', 4326), center),
            point('x', 'y') <@ innerperim,
            point('x', 'y') <@ closeperim
            imagefile,
FROM        geoimage
WHERE       point('x', 'y') <@ outerperim
```

Figure 5-14: SQL image query.

The query consists of three parts:

1. **SELECT:** The values requested from each of the images in the collection that matches the query's WHERE clause. Each of these is described below, with the retrieved data type in the parenthesis.
   - **Id:** Image identifier (integer)
   - **Area:** Area covered by the image (double).
   - **Angle:** Angle indicating the image perspective (double).
   - **Distance:** Distance between the query coordinate and image centre calculated using a PostGIS function (double, converted from string).
   - **Within inner:** Whether or not the query coordinate is located within the inner perimeter (Boolean).
   - **Within close:** Whether or not the query coordinate is located within the close perimeter (Boolean).
   - **Imagefile:** The file itself (array containing bytes).
2. **FROM:** The table(s) which are involved in the query. This prototype has all necessary information stored in one table; *geoimage*.
3. **WHERE:** The requirements that have to be fulfilled for a document in the collection to be considered relevant to the query, and therefore returned. In effect this acts as a filter, excluding the majority of the available documents before performing the calculations described in the SELECT part of the query. Only one such requirement is used here:
   - **Within outer:** For an image to be considered relevant it has to contain the query coordinate, i.e. the point is visible in the image. As coordinates are absolute, two assumptions are made here; the user has no interest in results that does not depict what he or she is looking for, and secondly that the query coordinates and outer perimeter feature are accurate.

5.3.7 Ranking

All documents returned from the database are run through a scoring function, where each image is given a score based on how well their features match up to the user's information need expressed through the query. The higher the score, the more relevant is the image, and when all the retrieved documents have been scored a sorted list is sent to the GUI, where it is displayed to the user.

The scoring system uses four of the five included features when determining relevance; inner perimeter, close perimeter, area, and angle. Distance turned out not to be needed for scoring, and was therefore not used. For estimating relevance/score, three questions are asked:

- **Where is the query coordinate located in the image?** Here there are only two possibilities; inside one of the interesting areas (inner or close perimeter), or outside. If the coordinate is located within one of the interesting areas a number is added to the score, while the score remains unchanged if the coordinate is outside these areas.
- **How much area is seen in the picture?** A large area results in less detail but a greater overview, with the opposite being true for images covering a small area. The size of the image impacts the score in one or the other direction depending on what type of images the user wants (given by the preferred image type option).
- **What angle/perspective does the image have?** The angle feature does not directly affect the score, but determines the effect the area feature has on score. This weighting is required to compensate for the difference in area distribution between image types. For instance will a picture taken horizontally have the vast majority of its area located far away from the point it was taken, while the area will be evenly distributed in an image taken straight down, in effect making area a less important factor for horizontal images.

These questions described as the specific functions that determine score / relevance are described and explained in Table 2. The weighting constants are a result from trying various values until which performed acceptable were found.

Table 2: Description of scoring functions.

Number	Name	Description	Score change
1	**Within inner**	The query coordinate is within the inner perimeter.	+ 0.01
2	**Within close**	The query coordinate is within the close perimeter AND angle is larger than 4. This requirement prevents quadratic images from scoring double, as both perimeters cover the same area for this image type.	+ 0.02
3	**Area for horizontal images**	The query coordinate is within the close perimeter AND angle is larger than 10 AND close-up images are preferred.	- 10 * area
4	**Area for fairly horizontal images**	The query coordinate is within the close perimeter AND the angle is larger than 4 AND close-up images are preferred.	- 25 * area
5	**Area for top-down images**	All other retrieved documents where close-up images are preferred.	- 50 * area
6	**Area**	All retrieved documents, far-away (overview) images preferred.	+ 40 * area

5.3.8 Other

The prototype is created for testing purposes only, and can be considered a proof of concept more than anything. It is not intended for future development or usage in any setting, and the quality of code and documentation reflects this. Factor such as usability, maintainability, security, etc., have not been prioritized, and the system should be treated with this in mind.

6 Evaluation

In this chapter the image retrieval parts of the prototype are evaluated. The purpose of this evaluation is not to see how well the prototype performs as an image retrieval system, but to discover and describe challenges and behaviour related to the different image types. Especially issues related to ranking and content features are evaluated, using the prototype as a way to illustrate and describe the various phenomenons and drawing on the theory presented earlier where possible.

6.1 Data Source

All experiments have been performed on an image collection containing 50 aerial photos of Trondheim and its surrounding area. These images are a selection from an image collection hosted by Universitetsbiblioteket i Trondheim[15], containing black and white photographs taken in the 1940s. Even though the area has changed much since then, the city layout and most landmarks remains the same. The selected images are a mix of all the image types described in chapter 4.2, with a slight overrepresentation of the types covering small areas.

6.2 Image Features

It turned out that even for images taken of an urban area, where landmarks are fairly easy to recognize, manually annotating the outer perimeter of each image was a difficult and time-consuming task. Even by having landmarks and a detailed 2-dimensional map to go by, accurately determining the coordinate of each corner proved to be quite challenging. The mistakes introduced here affected all the content features attached to an image, although not to such a degree that they made the prototype useless.

The difficulty of annotating images was mainly governed by the size of the area seen in the picture; large areas resulted in more significant inaccuracies, especially in images taken at horizontal angle. This was partly due to the difficulty in making out details in the landscape far away from the point of photography, but also from topographical elements such as hills resulting in areas that could not be represented using only four edges.

The features and values stored with each image were sufficient to rank query results in an insightful way. However, the choice to store all values as accurately as possible was not equally successful, due to the queries themselves not taking this into consideration (more on this in chapter 6.3). With the inaccuracies introduced when selecting coordinates (both for the outer perimeter and the queries themselves), approximate values may have done a better job of representing the user's input.

[15] http://www.ntnu.no/ub/

Not all content features stored in, and later fetched from the database ended up being used in the relevance ranking. Seeing as the choice was made to only return images which contained the search coordinates, the centre and skewed centre features did not have any apparent contribution, and therefore remained unused. However, both were used in determining and storing other features in the extraction process, and therefore had no detriment effect other than the extra storage space required. This approach should have been extended to include other metrics only used in the feature extraction, in the case of them becoming useful at some point, but was not.

6.3 Ranking

As mentioned, the ranking generated by the prototype was for evaluation purposes more than actual use, but these two are obviously closely related. A description of the ranking performance is described in this chapter, with examples supporting the descriptions. Due to images being a complex and visual type of information object, the evaluation of individual images has been done manually, using my own perception to decide to what degree, and why, an image is relevant. A consequence of this is that images are split into groups of similar images, as opposed to being viewed as individual units. This is due to the difficulty in judging the effect minor differences has on ranking, something a computer would be much better at. However, no automatic evaluation tool was available for this task.

Coordinates are unambiguous in the sense that there are no partial matches, and with the way the prototype handles queries this means all retrieved documents from the collection should be relevant to the query (the coordinate is located within the image). However, this was not the case due to the feature inaccuracy described in chapter 6.2, and in some cases images where the query coordinate in reality was located just outside the outer perimeter were retrieved, and to a much lesser degree images where the query coordinate was slightly off target and some small images would not be retrieved. Even though this accuracy issue is a source for concern, it did not have any impact on the vast majority of queries, as it did not affect the top half of the retrieved documents. The reason for this is that this problem is found exclusively in images covering a large area where the query coordinate is far away from the interesting areas. This type of image is not considered a particularly good match by any of the queries supported in the prototype.

The ranking method used in the prototype favours images where the query coordinate is located within one of the interesting areas, and the image area is either small or large depending on the query. For images taken straight down or a slightly horizontal, this method works quite well, and the most relevant images are given the highest score. The constants used when calculating the area modifier, i.e. those decided by perspective, makes it possible for images where the query coordinate is outside the interesting area to get a higher score and rank than images with the coordinate inside the interesting area, but with an area size in the opposite end compared to the user's preference. This behaviour balances the query parameters to a certain extent, but struggles as the images become more horizontal.

Even though the ranking of top-down imagery works well, there are some noticeable issues that affect performance in a negative way:

- **Interesting areas:** The score gained from the inner- and close perimeter features can be a too significant factor for images covering a large area. This was evident when requesting overview images where the query coordinates can be close to the outer edges without losing any real relevance, as the surrounding areas can be equally important to the user.
- **Area constants:** Currently the constants used for weighting the area feature have been selected through trying and failing, arriving at values that work fairly well with this particular collection, but which would have to be made more flexible as the amount of images increases. These values are decided solely by the angle of the image, a feature that has no connection to what matters for an actual user; how large or small an area have to be before the image itself is irrelevant. This problem is very obvious in this prototype, seeing as requested images should either be as small or as large as possible, instead of having clearly defined size categories. With the way area constants are implemented this means the largest and the smallest images sometimes get a higher relevance than they should.

Scoring of pictures taken horizontally is not nearly as successful as for straight-down images. The trend here is that the more horizontal an image is, the less representative its features stored in the database are. These image types suffer from the same problems mentioned as top-down images, but also come with a few additional ones of their own:

- **Close perimeter:** The close perimeter feature was intended to cover the interesting area in horizontal images, but is only partially successful in doing so. It is better than the inner perimeter, but fails to reflect how the interesting area of the image changes along with its perspective. As the image becomes more horizontal, the interesting area moves closer to the shortest edge (and the point of photography), ending up practically on the edge in the most extreme cases. Having the focus of the image so close to the border is unique for these kinds of images, and is something the prototype does not account for, resulting in inaccurate relevance calculations.
- **Area:** In the straight-down image types where the distance from all points in the picture to the point of photography is fairly similar, the area feature tells us approximately the level of detail we can expect to see in the image. When the image is tilted, this is no longer the case. They often cover a very large area, but the level of detail in the front of the picture is much higher than at the back, making the area feature meaningless. The prototype tries taking this into consideration by determining area constants from angle, but as the examples later in this chapter will show, this does not come close to fixing the problem.

Only the image type extremities have been described in detail in this chapter, but this does not mean the other types have been ignored or considered irrelevant. They are all affected by the problems mentioned previously to some extent, depending on how their own characteristics compare to the extremities. Having an individual evaluation of each would make little sense, seeing as they all share the same properties, although at different degrees, and the borders between the different types are easy definable.

6.3.1 Evaluation of Example Queries

To illustrate the various challenges and aspects found in the ranking, three evaluation example queries are shown below. Not all the retrieved images are shown, as this would take up too much space, instead some of the more interesting ones are shown and explained.

Query 1: Munkholmen, close up

- **Description:** Munkholmen is an islet located in Trondheimsfjorden, just outside the Trondheim city centre. Three images in the test collection cover this location, and all of these are taken at a horizontal angle, covering a large area.
- **Results:** The database returned five images matching the query, all those who actually contain it, and two more where the location was just outside their outer perimeter. Several of the images had the location within their inner-, close-, or both perimeters, but the area feature was the dominating factor in this query. This was especially evident for the one picture where Munkholmen was located at the front of the picture, but the huge size of the area brought the score down, ranking the photo 4th out of 5. See Figure 6-1.

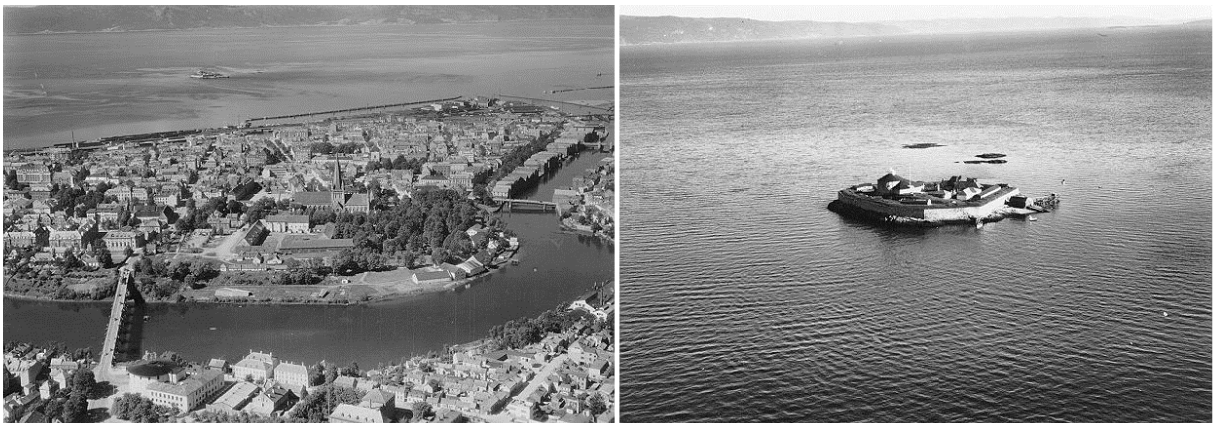


Figure 6-1: Munkholmen query, 1st ranked image on the left, 4th on the right.

- **Evaluation:** This query shows the effect inaccurate representation of images can have on ranking, where an image not even containing the query gets a better score than images that do. It also shows how the image a human would consider to be the most relevant is considered the opposite by the system, due to the combination of the coordinate being outside the interesting areas and it covering a huge area compared to the other documents in the result set.

Query 2: Nidarosdomen, close up

- **Description:** Nidarosdomen is a cathedral in the middle of the Trondheim city centre and can be seen in 22 images. Approximately half of these cover a small area, i.e. only containing the cathedral and its closest neighbouring buildings, with neither of these images having an extreme horizontal angle. The remaining half of the images is distributed between the other image types.
- **Results:** A total of 28 images were retrieved; the 22 containing the coordinate, 4 where it was located just outside the outer perimeter, and 2 where it was outside by a more considerably margin. 10 out of the top 13 results were close-up pictures of the cathedral, either alone or with some surrounding buildings. The remaining 3 have Nidarosdomen in the centre of the image, but are taken further away, and therefore cannot be considered quite as

good. The bottom half of the results are made up of various large area images, most which cover a very large area looking straight down, but also some horizontal images having the query coordinate in the foreground. See Figure 6-2.

- **Evaluation:** In general the ranking performs well; favouring close-up images with the query coordinate centered, as it should, although with some irregularities. Pictures where the location is within both the inner- and close perimeters get a slightly better score than they deserve, as illustrated by the 6th ranked image in Figure 3-1. As with the Munkholmen query we can see inaccuracies and problems with horizontal images here as well, although they are not as obvious due to the number of highly relevant images coming out on top.

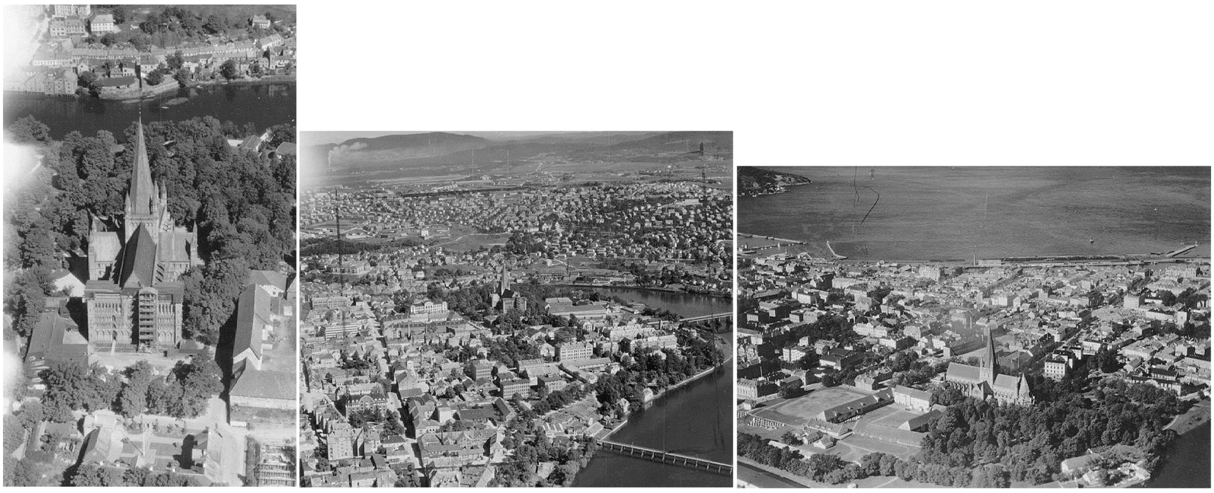


Figure 6-2: Nidarosdomen, close-up query, from left to right; 1st, 6th and 21st ranked images.

Query 3: Nidarosdomen, far away

- **Description:** Same as query 2.
- **Results:** Unsurprisingly the same amount of images was retrieved as in query 2, although ranked quite differently. Horizontal images covering a large area all do very well, along with top-down images covering most of Trondheim city (the biggest top down images in the collection). The lower half of the result list is comprised of small and medium sized images, typically those which got a good score in query 2, where those having the coordinate close to the borders being ranked the lowest.
- **Evaluation:** Again a ranking that makes sense when judged by human perception. The highest scoring images are all overview images, although with different perspectives, and as such fulfil the query parameters. Interestingly, the 6th ranked image in query 2 is ranked 6th here as well, another indication of the effect being inside both inner perimeters can have on scoring. Apart from this particular case, area is the main factor for determining relevance in this query. Images covering a large area get the highest scores, regardless of where the coordinate is within the picture. An unfortunate result of this is the 2nd ranked image, a fairly horizontal image covering a huge area, which does not contain Nidarosdomen.

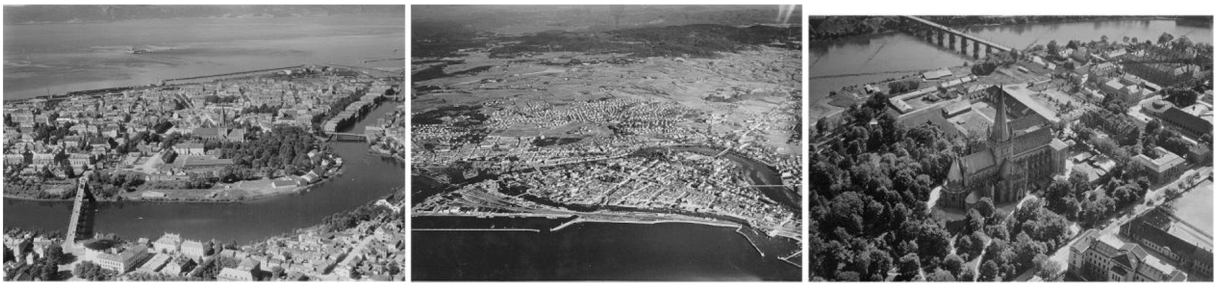


Figure 6-3: Nidarosdomen, far away query, from left to right; 1st, 4th and 21st ranked images.

6.4 Other

The prototype is designed to evaluate the information retrieval process, nothing more. But due to a noticeable delay when performing queries some tests were ran to find the source. These were executed directly on the database, comparing the time used when retrieving only the ID (numerical value), to retrieving ID and the image file (byte array). The results can be seen in the table below.

Table 3: Query execution time

Number of items	ID only (milliseconds)	ID and image file (milliseconds)
1	11	91
5	12	507
10	12	1227
25	15	2832
50	12	6159

Considering the scope and environment of this test no definite conclusions can be drawn, but the results indicate that fetching the actual image data is extremely time-consuming compared to returning numerical values. The trend shown in the test also shows the gap between the two increases exponentially, making this storage solution unfit for large-scale systems.

7 Conclusion

In this chapter the evaluation is summarized and conclusions are presented. In addition, suggestions for future work within the field of retrieving images based on spatial content, both in general and directly related to the prototype. A glimpse into the future of image retrieval based on spatial content, as judged by the author, is also offered as part of the conclusion.

7.1 Conclusion

The prototype exemplifies searching and ranking images based on their spatial content instead of regular metadata, the latter being the dominating method in other systems combining maps and images. Due to this lack of other similar projects, the points made in this chapter are based mostly on tests conducted using the prototype, and are grouped in a way that resembles the flow found here; i.e. starting with issues related to inserting images and ending with the result ranking.

Manual annotating and inserting images as done in this prototype was neither fast nor accurate enough to be used in a real world situation. Both drawbacks are caused by the difficulty of associating what is shown on a picture with a 2-dimensional map. Several factors contribute to this:

- **Landmarks:** Being unable to recognize features on either the image or the map.
- **Temporal factors:** The map and images are taken at different times or in different seasons.
- **Cardinal point:** Digital maps are locked to a direction, typically up equalling north; images can be taken facing any direction.
- **Distance:** The further away from the photographer parts of an image is, the harder it is to identify what exactly one is looking at.
- **Dimension:** The map being 2-dimensional, all imagery is taken looking straight down, showing only rooftops. This is not the case for aerial photography.

These factors only apply when using a digital map to help find coordinates. Ignoring the issue of acquiring the coordinates in the first place, inputting these values directly avoids the factors mentioned above, but requires knowledge of which SRIDs are used, and a way to convert between these if required.

The content features extracted from each image were chosen for their ability to define the image type without requiring user input, and for describing aspects of the image similar to human perception, for example by having interesting areas (inner and close perimeter). The included features partly accomplish this. Combining area and angle turned out to be a good way to separate between small and large, straight down and horizontal, although the lack of predefined category boundaries somewhat limited the actual usefulness. Also, for the image types covering a small area and those taken straight down or slightly horizontal, the features defined the visually interesting areas close to what a human would do. With the extreme horizontal image types, horizon showing or not, the situation was the complete opposite. The lack of any topographic information not only made it impossible to separate between the two types, as their angle and area features were indistinguishable, but neither the close- nor inner perimeter feature was able to identify the same interesting areas a person would, effectively making them useless for ranking in these cases. The close perimeter features was included solely to handle this particular issue, and its complete failure when dealing with these two types of images was a disappointment, yet informative.

Coordinates are unambiguous in their nature, and therefore well suited to be handled by data retrieval systems like the PostgreSQL and PostGIS combination used in the prototype. The spatial queries, both for inserting and retrieving images, were all written in SQL syntax which should be instantly familiar to those who have worked with this technology previously. Not having to learn a new query language from scratch can be seen as a major benefit for those tasked with creating and maintaining the system.

One of the strengths of storing information in a SQL database, the accuracy and absoluteness, is also its biggest weakness. In theory this should not be a concern, but as seen in the prototype evaluation, inaccuracies exist. Some fuzziness therefore needs to be included in the system, for instance in the database or the methods which formulates the queries.

The ranking performance is a result of all the underlying factors already described. When designing the system, the intention was giving the highest relevance to images where the query coordinate was inside one of the interesting areas (close- or inner perimeter), and ranking these on area size. This method proved too simple, not taking the difference between image types into account. The findings can be summarized as follows, grouped by image property:

- **Small size:** Images covering a small area, looking either straight down or horizontally, get the most accurate ranking of all the image types. They are not affected by inaccuracies to the same extent as the other image types, and are either the most or least relevant of all retrieved documents, depending on the query.
- **Medium size:** Not one of the previously defined characteristics, but mentioned here as it behaves differently from its smaller and larger counterparts. This type of image scores higher than it should when the query coordinate is within both the inner- and close perimeter. Larger areas are not affected in the same way due to the sometimes dominant nature of the area feature.
- **Large size:** Images covering a large area are the most prone to being inaccurate, with the situation worsening as the image tilts. In addition the area score was often too significant compared to the score gained from being inside the interesting areas. The combination of these two problems would in some cases rank an image not containing the coordinates over smaller images having the query point dead centre, assuming the query asks for overview images.
- **Straight down:** The easiest image type to rank and represent, regardless of size. The inner perimeter closely represents what a human considers interesting, making it relatively easy to find the most relevant images.
- **Fairly horizontal:** Slightly more complicated than straight down, the ranking of these images can still be considered relatively good. The visually interesting area shifts from the centre of the image, moving closer to the shortest edge, something which is picked up by the

close perimeter feature and therefore can be scored. Problems with ranking become more frequent as the area and angle increase.

- **Horizontal:** The prototype was not able to capture the visually interesting areas, as they were located too close to the edge to fall within the inner or close perimeter features. This image type also suffers from the area feature not representing the level of detail in the image, making the ranking of horizontal images perform very poorly.

The prototype was never intended to solve the problem of storing, searching and ranking images based on their spatial features, something it is far from accomplishing. However, it has uncovered several challenges related to representing and ranking images in such a system. Perhaps the most interesting of these and what should be the focus of future research, is the fact that the variation found in aerial photography image types is too large for all of them to be treated in the same way. Especially the horizontal pictures did not play by the same rules at the other image types, creating huge problems when representing and ranking them.

7.2 Future Work

This being an investigative study, suggestions for future research can be of equal importance as the results themselves. The following two subchapters outline the author's ideas for what should be focused on in the future using the theory, state of the art, and prototype results presented in this thesis as a foundation. These suggestions are divided in two; and those concerning large and complex problems and those dealing with the development of a working prototype.

7.2.1 In General

Accurate registration of images is one of the more pressing concerns, and several approaches involving existing technology can be explored to achieve this. A possibility is switching from static 2-D maps to interactive maps in 3-D when determining perimeter coordinates, something which could make it easier to correlate image content with maps of the real world. Other UI changes should also be looked into.

Removing all manual factors from image registration is another way to improve accuracy, in other words having a computer automatically decide the coverage of the image. In many ways this can be considered an extension of Bing streetside described in chapter 3.3, where pattern matching is used for mapping images to locations. To make possible on larger areas, detailed 3-D maps have to be available, with possible sources for this being Google Earth and robotic mapping projects. Another aspect of this approach is looking at the metadata requirements necessary for it to work, and specifically look at how spatial coordinates or textual descriptors attached images can help the pattern matching process by narrowing the area considered.

As opposed to increasing accuracy, decreasing it on purpose is also an avenue which could be explored. Although this goes against much of what has been written earlier, having a system which is aware of this fact, treating data accordingly, could be a viable approach. The major benefit to this is not relying on users inputting exact values when inserting and querying images, but it does require a clearly defined set of rules for rounding off values and similar.

In this prototype the outer perimeter of images has been represented using only four coordinates, something which was not always able to captures the entire coverage. Attempts should therefore be made using polygons having five or more corners, to see if the benefits gained in accuracy are worth the added complexity in and possible effect this change may have on performance.

Free text queries were listed as one of the requirements for a fully functioning spatial image retrieval system. To make this possible a database containing relations between landmarks and coordinates has to be available. Two possibilities for performing text queries have been identified and should be explored further;

- **Text-to-coordinate:** All text-queries are translated into coordinates, and the coordinate is used for querying the collections. Text-queries can therefore be treated just like those involving coordinates in all parts of the system.
- **Pre-processing:** Whenever a new image is added to the system, the coverage (gathered from its coordinates) is combined with the relations database, extracting all landmarks within the given area and storing these along with the image. Can possibly improve browsing, but is vulnerable to landmark changes, and comes with a set of ranking challenges, unless the position within the image is stored as well, in which case it will be very similar to the other query types.

Regardless of how text queries are handled, techniques and tools from IR should be applied where possible to avoid recreating solutions that already exist.

Optimal storage and indexing of spatial- and image data is only briefly mentioned in this paper, but is something that needs to be evaluated and implemented to ensure scalability. There are obviously other areas with room for optimization, but these will be closely related to project-specific details such as implementation language, and is therefore not as relevant here.

As shown in the evaluation, the visually interesting area in images shifts when the perspective and area of the image changes. Finding a way to express the this shift mathematically is a key challenge for describing and ranking aerial photography, making it possible to pinpoint the interesting areas for all image types without requiring human input. This can be seen as a further development of the inner- and close perimeter features in the prototype.

Another ranking challenge is finding a way to normalize the results from each image type, so that comparisons can be made between the different types. With the current ranking technique the score varies too much between categories, some image types being more likely to score higher than others independent of their relevance to the query. The main reason for this is the area feature having too much of an effect, and ways to handle this without losing any of its meaning should be explored.

The goal with an image retrieval system such as this is putting it to actual use, and to achieve this, the intended users have to be included to some extent in all the suggestions mentioned above. This is especially important in all work concerning the relevance ranking of the results. With this in mind, rapid prototyping and usability workshops should be used to make sure the system developed is the correct one.

7.2.2 Prototype

The future work suggestions already presented will at some point all affect the system prototypes. When, if, and how depends on which subject(s) the individual research teams decide to focus on, as they span too widely to all be tackled at the same time. Some work, however, can be almost directly applied to the prototype functionality described in this paper to make it more useful, either by expanding existing code or building a new prototype from scratch.

The GUI was one of the things that were not prioritized in this project, but due to the extensive testing a system such as this should be put through, a working UI should be implemented. Unlike the current prototype, where a simple placeholder GUI is used, the goal should be implementing something resembling the desired final result. This means connecting the system to a digital map service, e.g. Google Maps, and allowing linking to external images, to name a couple of the desired components.

Such a system needs to support more than single coordinate queries, and adding functionality for shape and text queries should be a top priority. Not only are these important features in themselves, but they also allow us to study ranking behaviour unique to these query types, similar to what the prototype currently does for single coordinates queries.

Bibliography

Azuma, R. T. (1997). A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments 6*, 355-385.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval.* Harlow: Pearson Education Limited.

Bar-Zeev, A. (2007, July 3). *How Google [Really] Works*. Retrieved March 19, 2010, from Reality Prime: http://www.realityprime.com/articles/how-google-earth-really-works

Curtis, A. J., Mills, J. W., & Leitner, M. (2006). Spatial confidentiality and GIS: re-engineering mortality locations from published maps about Hurricane Katrina. *International Journal of Health Geographics*, 5-44.

Datta, R., Li, J., & Wang, J. Z. (2005). Content-Based Image Retrieval - Approaches and Trends of the New Age. *MIR'05* (pp. 11-12). Singapore: ACM.

Dempsey, C. (2008, May 1). *What is GIS?* Retrieved April 9, 2010, from GIS Lounge: http://gislounge.com/what-is-gis/

ESRI. (2008). *Understanding and Implementing ArcGIS Image Server.* New York: ESRI.

ESRI. (2010). *ArcNews Winter 2008/2009 Issue -- The Geodatabase: Medeling and Managing Spatial Data*. Retrieved March 16, 2010, from ESRI: http://www.esri.com/news/arcnews/winter0809articles/the-geodatabase.html

ESRI. (2010). *ESRI | The GIS Software Leader*. Retrieved March 11, 2010, from ArcGIS: A Complete Integrated System: www.esri.com/arcgis

Fluhr, C., Moëllic, P.-A., & Hede, P. (2006). Usage-oriented Multimedia Information Retrieval Technological Evaluation. *MIR'06.* Santa Barbara: ACM.

Ganguly, P., Rabhi, F. A., & Ray, P. K. (2002). Bridging Semantic Gap. *Conferences in Research and Practice in Information Technology, Vol. 13.* Melbourne: Australian Computer Society, Inc.

Goldberg, D. W. (2008). *A Geocoding Best Practices Guide.* North American Association of Central Cancer Registries, Inc.

Goyvaerts, J. (2010). *Regular-Expressions.info*. Retrieved March 5, 2010, from Regular-Expressions.info: http://www.regular-expressions.info/

Hare, J. S., Lewis, P. H., Enser, P. G., & Sandom, C. J. (2006). Mind the Gap: Another look at the problem of the semantic gap in image retrieval. *Proceedings of SPIE.* SPIE.

Howard, A., Wolf, D. F., & Sukhatme, G. S. (2004). Towards 3D Mapping in Large Urban Environments., (pp. 419-424). Sendai.

Lew, M. S., Sebe, N., Djeraba, C., & Jain, R. (2006, February). Content-based Multimedia Information Retrieval: State of the Art and Challenges. *ACM Transactions on Multimedia Computing, Communications and Applications*, pp. 1-19.

Liu, Z., Lim, E.-P., Ng, W.-K., & Goh, D. H. (2003). On Querying Geospatial and Georeferenced Metadata Resources in G-Portal. *Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries* (pp. 245-255). Houston, Texas: IEEE Computer Society.

Lu, G. (1999). *Multimedia Database Management Systems.* Norwood: Artech House, Inc.

Microsoft. (2010). *Map apps*. Retrieved March 15, 2010, from Bing Maps: http://www.bing.com/maps/explore/

Microsoft. (2010). *New Bing Maps Application: Streetside Photos*. Retrieved March 15, 2010, from Bing Community: http://www.bing.com/toolbox/blogs/maps/archive/2010/02/11/new-bing-maps-application-streetside-photos.aspx

Montemerlo, M., & Thrun, S. (2005). Large-Scale Robotic 3-D Mapping of Urban Structures.

Peterson, M. P. (2005). Maps and the Internet: An Introduction. In M. P. Peterson, *Maps and the Internet.* Amsterdam, USA: Elsevier.

Pichler, G., & Hogeweg, M. (2009). Improving Access and Use of Imagery using Open and Interoperable Off-the-shelf Technologies. *International Symposium on Remote Sensing of Environment.* Stresa.

Thrun, S. (2003). Robotic Mapping: A Survey. In *Exploring Artificial Intelligence in the New Millennium* (pp. 1-35). San Francisco: Morgan Kaufmann Publishers Inc.

University of California. (2004). *Alexandria Digital Library Operations*. Retrieved March 24, 2010, from Alexandria Digital Library Project: http://www.alexandria.ucsb.edu/adl/

Wang, C., Zhang, L., & Zhang, H.-J. (2008). Learning to Reduce the Semantic Gap in Web Image Retrieval and Annotation*. *SIGIR'08* (pp. 20-24). Singapore: ACM.

Zheng, Q.-F., & Gao, W. (2008). Constructing Visual Phrases for Effective and Efficient Object-Based Image Retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications*.

Zhitomirsky-Geffet, M., & Dagan, I. (2009). Bootstrapping Distributional Feature Vector Quality. *Computational Linguistics*.